# System Security Command Reference for Cisco 8000 Series Routers

**First Published:** 2020-03-01

**Last Modified:** 2024-03-01

**Americas Headquarters**
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

# Preface

This guide describes the configuration procedure and examples for system security in Cisco ASR 9000 Series Routers Cisco 8000 Series Routers.



# 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***

| Date | Summary |
|---|---|
| March 2024 | Republished for Release 24.1.1 |
| August 2023 | Republished for Release 7.10.1 |
| November 2022 | Republished for Release 7.8.1 |
| July 2022 | Republished for Release 7.7.1 |
| May 2022 | Republished for Release 7.3.4 |
| April 2022 | Republished for Release 7.5.2 |
| November 2021 | Republished for Release 7.5.1 |
| October 2021 | Republished for Release 7.3.2 |
| May 2021 | Republished for Release 7.3.15 |
| February 2021 | Republished for Release 7.3.1 |
| August 2020 | Republished for Release 7.0.14 |
| August 2020 | Republished for Release 7.2.1 |
| August 2020 | Republished for Release 7.1.2 |

| Date | Summary |
|---|---|
| March 2020 | Initial release of this document. |

# 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.

# Trustworthy Systems Commands

This module describes the commands related to trustworthy systems on Cisco IOS XR7 software.

For detailed information about the key components that form the trustworthy security systems, see the *Implementing Trustworthy Systems* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# platform security device-ownership

To configure secure device ownership for the router, use the **platform security device-ownership** command in EXEC modeXR EXEC mode.

**platform security device-ownership** *ownership-voucher-path* **location** { *location* | **all** }

## Syntax Description

| | |
|---|---|
| *ownership-voucher-path* | Path to the .tar file containing the Ownership Vouchers (OV) and Authenticated Variable (AV) to securely transfer device ownership |
| **location** { *location* \| **all** } | Applies AV to a specific location or all locations |

## Command Default

None

## Command Modes

EXECXR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | This command was introduced. |

## Usage Guidelines

A power cycle of the node is required for the extended ownership transfer to take affect.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

This example shows how to configure the device ownership on the router:

```
Router#platform security device-ownership /harddisk:/multiple-ov.tar.gz location all
Thu Feb 23 16:42:19.207 UTC
Successfully applied ownership voucher in node0_RP0_CPU0.
Successfully applied ownership voucher in node0_1_CPU0
Power-cycle of the node is required for the dual ownership transfer to take affect.
```

# platform security variable customer

To configure the secure variable for certificate storage of customer variables, use the **platform security variable customer** command in EXEC modeXR EXEC mode.

**platform security variable customer** { **zeroize** *authenticated-variable-file-path* **GUID** *av-customer-guid* | **append** *key authenticated-variable-file-path* | **update** *key authenticated-variable-file-path* } **location** { *location* | **all** }

## Syntax Description

| | |
|---|---|
| **zeroize** | Clears the entire certificate store using Authenticated Variable (AV). Use this variable with caution |
| **append** *key* | Appends certificates or hashes to Extensible Firmware Interface (EFI) to one of the following keys:<br>• KEKCustomer—Key Exchange Key Customer<br>• PKCustomer—Platform Key Customer<br>• dbCustomer—Signature and key database Customer<br>• dbxCustomer—Forbidden signature and key database Customer |
| **update** *key* | Removes or replace certificates or hashes in EFI for one of the following keys:<br>• KEKCustomer—Key Exchange Key Customer<br>• PKCustomer—Platform Key Customer<br>• dbCustomer—Signature and key database Customer<br>• dbxCustomer—Forbidden signature and key database Customer |
| *authenticated-variable-file-path* | Path to the AV file |
| **GUID** *av-customer-guid* | Cisco-provided Global Unique Identification number (GUID) |
| **location** { *location* \| **all** } | Applies AV to a specific location or all locations |

## Command Default

None

## Command Modes

EXECXR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | This command was introduced. |

## Usage Guidelines

Use the zeroize command with caution as the entire certificate store using authenticated variable can be cleared. After you use the command, a reboot is required immediately for the changes to take effect.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

This example shows how to update the KEKCustomer key for all nodes on the router using a sample `sonic-kek-release-update.auth` file that is created and stored in the harddisk: of the router:

```
Router#platform security variable customer update KEKCustomer
/harddisk:/sonic-kek-release-update.auth location all
Fri Feb 24 05:15:35.765 UTC
Performing operation on all nodes..
=========================
Location : 0/RP0/CPU0
=========================
Successfully applied AV /harddisk:/sonic-kek-release-update.auth for KEKCustomer
* WARNING *: Immediate reboot is recommended to avoid system instability!
=========================
Location : 0/1/CPU0
=========================
Successfully applied AV /harddisk:/sonic-kek-release-update.auth for KEKCustomer
* WARNING *: Immediate reboot is recommended to avoid system instability!
```

# show platform security boot mode

To display the security boot mode for the router, use the **show platform security boot mode** command in EXEC modeXR EXEC mode.

**show platform security boot mode location** { *location* | **all** }

**Syntax Description**

| | |
|---|---|
| **location** { *location* \| **all** } | Specifies a specific location or all locations |

**Command Default**

None

**Command Modes**

EXECXR EXEC

**Command History**

| Release | Modification |
|---|---|
| Release 7.10.1 | This command was introduced. |

No specific guidelines impact the use of this command.

**Task ID**

| Task ID | Operations |
|---|---|
| system | read, write |

**Examples**

This example shows how to view the secure boot mode of the router. In this example, the mode is `Generic Mode`:

```
Router#show platform security boot mode location all
Tue Feb 21 16:40:16.207 UTC
Performing operation on all nodes...
=========================
Location  :  0/RP0/CPU0
=========================
Aikido mode: Generic Mode
Aikido mode value: 43

=========================
Location  :  0/1/CPU0
=========================
Aikido mode: Generic Mode
Aikido mode value: 43
```

This example shows the mode in `Customer Mode`:

```
Router#show platform  security boot mode location all
Tue Feb 21 16:40:16.207 UTC
Performing operation on all nodes..
=========================
Location : 0/RP0/CPU0
```

```
========================

Aikido mode: Customer Mode
Aikido mode value: 127
========================
Location : 0/2/CPU0
========================

Aikido mode: Customer Mode
Aikido mode value: 127
========================
Location : 0/1/CPU0
========================

Aikido mode: Customer Mode
Aikido mode value: 127
```

# show platform security integrity log

To display the security integrity logs for the router, use the **show platform security integrity log** command in EXEC modeXR EXEC mode.

**show platform security integrity log** { **boot location** *location-name* | **runtime** *file-location* | **secure-boot status location** *location-name* }

## Syntax Description

| | |
|---|---|
| **boot** | Displays boot integrity logs |
| **runtime** | Displays integrity measurement architecture (IMA) logs |
| **secure-boot** | Displays information related to secure boot |

## Command Default

None

## Command Modes

EXECXR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | The command was modified to include the secure boot status. |
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

If the router does not support this secure boot verification functionality, then the status is displayed as *Not Supported*.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

This example shows how to verify the secure boot status of the router:

```
Router#show  platform security integrity log secure-boot status
Wed Aug 10 15:39:17.871 UTC

+--------------------------------------+
   Node location: node0_RP0_CPU0
+--------------------------------------+
Secure Boot Status: Enabled
Router#
```

# show platform security variable customer

To verify that the customer key certificate is active and registered for PKCustomer, KEKCustomer, dbCustomer and dbxCustomer variables, use the **show platform security variable customer** command in EXEC modeXR EXEC mode.

**show platform security variable customer** *key* **[detail] location** { *location* | **all** }

## Syntax Description

| | |
|---|---|
| *key* | Specifies the type of variable to which the customer key certificate is added—PKCustomer, KEKCustomer, dbCustomer and dbxCustomer |
| **detail** | Displays full certificate details for a specific location or all nodes |
| **location** *location-name* | Specifies a specific location or all locations |

## Command Default

None

## Command Modes

EXECXR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | This command was introduced. |

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

This example shows how to view the secure variables for KEKCustomer certificate for all the locations on the router:

```
Router#show platform security variable customer KEKCustomer location all
Fri Feb 24 05:16:56.365 UTC
Performing operation on all nodes..
=========================
Location : 0/RP0/CPU0
=========================

Variable : KEKCustomer
+--------------------

Signature List # 0
 GUID : f79d17d1-88d4-40dd-aff8-9f9da3c30e9e
 Extension type : X509

 Entry # 0
 Owner GUID : f79d17d1-88d4-40dd-aff8-9f9da3c30e9e
 Size : 1211
```

```
  Serial Number  : BA:5C:D4:5E:F3:D4:D0:4C
  Subject:
        O=Cisco,OU=RELEASE,CN=IOSXR-WHITEBOX-KEK
  Issued By      :
        O=Cisco,OU=RELEASE,CN=IOSXR-WHITEBOX-KEK
  Validity Start : 10:03:18 UTC Wed Feb 23 2022
  Validity End   : 10:03:18 UTC Tue Feb 18 2042

  CRL Distribution Point
        http://www.cisco.com/security/pki/crl/crcakekdtxr.crl
  SHA1 Fingerprint:
         AE4DFD35EB8486FC5707609C93A5C44CDB579126

Total Signature Lists # 1
Total Certificates # 1
=========================
Location : 0/1/CPU0
=========================

Variable : KEKCustomer
+--------------------

Signature List # 0
 GUID : f79d17d1-88d4-40dd-aff8-9f9da3c30e9e
 Extension type : X509

 Entry # 0
 Owner GUID : f79d17d1-88d4-40dd-aff8-9f9da3c30e9e
 Size : 1211

  Serial Number  : BA:5C:D4:5E:F3:D4:D0:4C
  Subject:
        O=Cisco,OU=RELEASE,CN=IOSXR-WHITEBOX-KEK
  Issued By      :
        O=Cisco,OU=RELEASE,CN=IOSXR-WHITEBOX-KEK
  Validity Start : 10:03:18 UTC Wed Feb 23 2022
  Validity End   : 10:03:18 UTC Tue Feb 18 2042

  CRL Distribution Point
        http://www.cisco.com/security/pki/crl/crcakekdtxr.crl
  SHA1 Fingerprint:
         AE4DFD35EB8486FC5707609C93A5C44CDB579126

Total Signature Lists # 1
Total Certificates # 1
```

# 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 ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# aaa accounting

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

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

## Syntax Description

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

EXEC modeXR EXEC mode

## Command History

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

## 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

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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 Global Configuration modeXR 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:<br>• **group tacacs+**—Uses the list of all TACACS+ servers for accounting.<br>• **group radius**—Uses the list of all RADIUS servers for accounting.<br>• **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

Global Configuration modeXR Config mode

## Command History

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

## 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

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

## Examples

This example shows how to cause a "start accounting" record to be sent to a TACACS+ server when a router initially boots. A "stop accounting" record is also sent when a router is shut down or reloaded.

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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 Global Configuration modeXR Config mode. To disable the system accounting for rp-failover, use the **no** form of this command.

**aaa accounting system rp-failover** {*list_name* {**start-stop** | **stop-only**} | **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

Global Configuration modeXR Config mode

## Command History

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

## 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:

```
RP/0/RP0RSP0/CPU0:router(config)# aaa accounting system rp-failover default start-stop none
```

## Related Commands

| Command | Description |
|---|---|
| aaa attribute format | Create an AAA attribute format name. |

# aaa accounting update

To enable periodic interim accounting records to be sent to the accounting server, use the **aaa accounting update** command in the Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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** { **dot1x** { *list-name* | **default** } **group** { *server-group-name* | **radius** } [ **group** *server-group-name* ] | **login** | **ppp**} {**default***list-name* | **remote**} *method-list*

## Syntax Description

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

## Command Default

Default behavior applies the local authentication on all ports.

## Command Modes

Global configuration

## Command History

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

## 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.

## Task ID

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa authentication login default group tacacs+
```

This example shows how to set the AAA authentication lists for dot1x to use list of all RADIUS hosts:

```
Router#configure
Router(config)#aaa authentication dot1x default group radius
Router(config)#commit
```

## Related Commands

| Command | Description |
|---|---|
| aaa accounting, on page 13 | Creates a method list for accounting. |
| aaa authorization, on page 21 | Creates a method list for authorization. |
| aaa group server radius, on page 28 | Groups different RADIUS server hosts into distinct lists and distinct methods. |
| aaa group server tacacs+, on page 30 | Groups different TACACS+ server hosts into distinct lists and distinct methods. |
| login authentication, on page 48 | Enables AAA authentication for logins. |
| tacacs-server host, on page 107 | Specifies a TACACS+ host. |

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

## Syntax Description

| | |
|---|---|
| **commands** | Configures authorization for all EXEC shell commands. |
| **eventmanager** | Applies an authorization method for authorizing an event manager (fault manager). |
| **exec** | Configures authorization for an interactive ( EXEC) session. |
| **network** | Configures authorization for network services, such as PPP or Internet Key Exchange (IKE). |
| **subscriber** | Sets the authorization lists for the subscriber. |
| **nacm** | Enables the nacm functionality. |
| **default** | Uses the listed authorization methods that follow this keyword as the default list of methods for authorization. |
| *list-name* | Character string used to name the list of authorization methods. |
| **none** | Uses no authorization. If you specify **none**, no subsequent authorization methods is attempted. However, the task ID authorization is always required and cannot be disabled. |
| **local** | Uses local authorization.<br>While this method of authorization is already supported, it is available for command authorization only from Cisco IOS XR Software Release 7.5.1 and later. |
| **group tacacs+** | Uses the list of all configured TACACS+ servers for authorization. |
| **group radius** | Uses the list of all configured RADIUS servers for authorization. This method of authorization is not available for command authorization. |
| **group** *group-name* | Uses a named subset of TACACS+ or RADIUS servers for authorization as defined by the **aaa group server tacacs+** or **aaa group server radius** command. |

## Command Default

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

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| Release 7.5.1 | The command was modified to make the **local** option available for command authorization as well. |

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

## Usage Guidelines

Use the **aaa authorization** command to create method lists defining specific authorization methods that can be used on a per-line or per-interface basis. You can specify up to four methods in the method list.

**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.

To know more about command authorization using local user account feature which was introduced in Cisco IOS XR Software Release 7.5.1, see the *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

## Task ID

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa authorization commands listname1 group tacacs+
```

The following examples show how to configure command authorization using local user account:

```
Router#configure
Router(config)#aaa authorization commands default group tacacs+ local
Router(config)#commit
```

or

```
Router(config)#aaa authorization commands default local
Router(config)#commit
```

## Related Commands

| Command | Description |
| --- | --- |
| aaa accounting, on page 13 | Creates a method list for accounting. |

# aaa display-login-failed-users

To display username for failed authentication, use the **aaa display-login-failed-users** command in Global Configuration modeXR Config mode. To remove the configuration, use the **no** form of this command.

**aaa display-login-failed-users**

## Syntax Description

This command has no keywords or arguments.

## Command Default

Disabled, by default

## Command Modes

Global configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | The command was introduced to make the **display-login-failed-users** option available to display user ID for failed user login attempts. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| aaa | read, write |

This example shows how to enable the functioanlity to display the username for a failed authentication:

```
Router#Configure
Router(config)# aaa display-login-failed-users
Router(config)#commit
```

# aaa default-taskgroup

To specify a task group for both remote TACACS+ authentication and RADIUS authentication, use the **aaa default-taskgroup** command in the Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

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

## Task ID

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa default-taskgroup taskgroup1
```

# aaa enable-cert-authentication

To enable certificate-based authentication for users in the TACACS+ Server or Server Groups, use the **aaa enable-cert-authentication** command in the XR-Config mode.

**aaa enable-cert-authentication**

## Syntax Description

This command has no keywords or arguments.

## Command Default

Certificate-based user authentication using TACACS+ server is disabled.

## Command Modes

XR-Config mode.

## Command History

| Release | Modification |
|---|---|
| Release 7.5.4 | This command was introduced. |

## Usage Guidelines

Enable AAA authorization using **aaa authorization exec** command.

## Task ID

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

## Examples

The following example shows how to configure certificate-based authentication for users configured in the TACACS+ Server or Server Groups:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa enable-cert-authentication
RP/0/RP0RSP0/CPU0:router(config)# aaa authorization exec default group tacacs+ local
RP/0/RP0RSP0/CPU0:router(config)# commit
```

# aaa group server radius

To group different RADIUS server hosts into distinct lists, use the **aaa group server radius** command in the Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server radius radgroup1
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server 10.0.0.5 auth-port 1700 acct-port 1701
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server 10.0.0.10 auth-port 1702 acct-port 1703
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server 10.0.0.20 auth-port 1705 acct-port 1706
```

**Note** If the **auth-port** *port-number* and **acct-port** *port-number* keywords and arguments are not specified, the default value of the *port-number* argument for the **auth-port** keyword is 1645 and the default value of the *port-number* argument for the **acct-port** keyword is 1646.

# aaa group server tacacs+

To group different TACACS+ server hosts into distinct lists, use the **aaa group server tacacs+** command in the Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server tacacs+ tacgroup1
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs)# server 192.168.200.226
```

```
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs)# server 192.168.200.227
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs)# server 192.168.200.228
```

# aaa password-policy

To define a AAA password security policy, use the **aaa password-policy** command in Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |
| Release 7.2.1 | The command options (except a few mentioned in the usage guidelines section) were extended to user secret as well. |

## 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 ASR 9000 Series Routers 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 *Configuiring FIPS Mode* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

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

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

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

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

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

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

| Command Variables | Default Value | Maximum Value | Minimum Value |
|---|---|---|---|
| *policy-name* | None | 253 | 1 |
| *max-length* | 253 | 253 | 2 |
| *min-length* | 2 | 253 | 2 |
| *special-char* | 0 | 253 | 0 |
| *upper-case* | 0 | 253 | 0 |
| *lower-case* | 0 | 253 | 0 |
| *numeric* | 0 | 253 | 0 |
| For **lifetime** :<br>**years**<br>**months**<br>**days**<br>**hours**<br>**minutes**<br>**seconds** | 0<br>0<br>0<br>0<br>0<br>0 | 99<br>11<br>30<br>23<br>59<br>59 | 1<br>1<br>1<br>1<br>1<br>1 |

| Command Variables | Default Value | Maximum Value | Minimum Value |
|---|---|---|---|
| *min-char-change* | 4 | 253 | 0 |
| *authen-max-attempts* | 0 | 24 | 1 |
| For **lockout-time** : | 0 | 255 | 1 |
| **days** | 0 | 23 | 1 |
| **hours** | 0 | 59 | 1 |
| **minutes** | 0 | 59 | 1 |
| **seconds** | | | |

## Task ID

| Task ID | Operation |
|---|---|
| aaa | read, write |

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

```
RP/0/RP0RSP0/CPU0:router(config)#aaa password-policy test-policy
RP/0/RP0RSP0/CPU0:router(config-aaa)#min-length 8
RP/0/RP0RSP0/CPU0:router(config-aaa)#max-length 15
RP/0/RP0RSP0/CPU0:router(config-aaa)#lifetime months 3
RP/0/RP0RSP0/CPU0:router(config-aaa)#min-char-change 5
RP/0/RP0RSP0/CPU0:router(config-aaa)#authen-max-attempts 3
RP/0/RP0RSP0/CPU0:router(config-aaa)#lockout-time days 1
```

## Related Commands

| Command | Description |
|---|---|
| username, on page 123 | |

# 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 EXEC modeXR EXEC mode shell commands. |
| **exec** | Enables accounting of EXEC modeXR 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

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

## 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

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# line template configure
RP/0/RP0RSP0/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 EXEC modeXR 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

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# line template configure
RP/0/RP0RSP0/CPU0:router(config-line)# authorization commands listname4
```

# clear tacacs counters

To clear AAA counters for all the TACACS+ servers in the system, use the **clear tacacs counters**command in the EXEC modeXR EXEC mode.

**clear tacacs counters**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.5.4 | This command was introduced. |

## Usage Guidelines

Use the **clear tacacs counters** command to clear all AAA counter statistics for all the TACACS+ server configured in the system.

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

The following is sample output from the **clear tacacs counters** command:

```
Router:ios# show tacacs counters
TACACS+ Server:   10.105.236.101/4010 [global]

  Authentication:
   10 requests, 4 accepts, 3 failure, 2 error, 1 timeout

  Exec Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout

  Command Authorization:
   6 requests, 6 accepts, 0 denied, 0 error, 0 timeout

  Exec Accounting:
   0 requests, 0 accepts, 0 fail, 0 error, 0 timeout

  Command Accounting:
   6 requests, 6 accepts, 0 fail, 0 error, 0 timeout

TACACS+ Server:   10.105.236.101/2201 [private] vrf = default

  Authentication:
   0 requests, 0 accepts, 0 failure, 0 error, 0 timeout

  Exec Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout
```

```
  Command Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout

  Exec Accounting:
   0 requests, 0 accepts, 0 fail, 0 error, 0 timeout

  Command Accounting:
   0 requests, 0 accepts, 0 fail, 0 error, 0 timeout

Router:ios# clear tacacs counters
Router:ios# show tacacs counters
TACACS+ Server:   10.105.236.101/4010 [global]

  Authentication:
   0 requests, 0 accepts, 0 failure, 0 error, 0 timeout

  Exec Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout

  Command Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout

  Exec Accounting:
   0 requests, 0 accepts, 0 fail, 0 error, 0 timeout

  Command Accounting:
   0 requests, 0 accepts, 0 fail, 0 error, 0 timeout


TACACS+ Server:   10.105.236.101/2201 [private] vrf = default

  Authentication:
   0 requests, 0 accepts, 0 failure, 0 error, 0 timeout

  Exec Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout

  Command Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout

  Exec Accounting:
   0 requests, 0 accepts, 0 fail, 0 error, 0 timeout

  Command Accounting:
   0 requests, 0 accepts, 0 fail, 0 error, 0 timeout
```

# 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

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

## 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.

## Task ID

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server 10.1.1.1 auth-port 1645 acct-port 1646
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server 10.2.2.2 auth-port 2000 acct-port 2001
RP/0/RP0RSP0/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

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# taskgroup alpha
RP/0/RP0RSP0/CPU0:router(config-tg)# description this is a sample taskgroup
```

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# usergroup alpha
RP/0/RP0RSP0/CPU0:router(config-ug)# description this is a sample user group
```

# group (AAA)

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

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

## Syntax Description

| | |
|---|---|
| **cisco-support** | Adds the user to the predefined Cisco support personnel group.<br>**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

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

## Usage Guidelines

Use the **group** command in username configuration mode. To access username configuration mode, use the username, on page 123 command in Global Configuration modeXR 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.

## Task ID

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

## Examples

The following example shows how to assign the user group operator to the user named user1:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# username user1
RP/0/RP0RSP0/CPU0:router(config-un)# group operator
```

# inherit taskgroup

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

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

## Syntax Description

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

## Command Default

None

## Command Modes

Task group configuration

## Command History

| Release | Modification |
|---|---|
| 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# taskgroup tg1
RP/0/RP0RSP0/CPU0:router(config-tg)# inherit taskgroup tg2
RP/0/RP0RSP0/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**

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

**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**

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

**Examples**

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# usergroup purchasing
RP/0/RP0RSP0/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-keyclear-text-key*}
**no key** {**0** *clear-text-key* | **7** *encrypted-keyclear-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

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

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

## Examples

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

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# aaa group server radius group1
RP/0/RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0/RSP0/CPU0:router(config-sg-radius-private)# key anykey
```

## Related Commands

| Command | Description |
|---|---|
| aaa group server tacacs+, on page 30 | Groups different RADIUS server hosts into distinct lists. |
| radius-server key, on page 60 | Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon. |

| Command | Description |
| --- | --- |
| radius-server retransmit, on page 61 | Specifies the number of times a RADIUS request is resent to a server if the server is not responding or is responding slowly. |
| server-private (RADIUS), on page 71 | Configures the IP address of the private RADIUS server for the group server. |
| timeout (RADIUS), on page 120 | Specifies the number of seconds the router waits for the RADIUS server to reply before retransmitting. |

# 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-keyauth-key*}
**no key** {**0** *clear-text-key* | **7** *encrypted-keyauth-key*}

## Syntax Description

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

## Command Default

None

## Command Modes

TACACS host configuration

## Command History

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226
RP/0/RP0RSP0/CPU0:router(config-tacacs-host)# key anykey
```

## Related Commands

| Command | Description |
|---|---|
| tacacs-server host, on page 107 | Specifies a TACACS+ host. |
| tacacs-server key, on page 111 | Globally sets the authentication encryption key used for all TACACS+ communications between the router and the TACACS+ daemon. |

# login authentication

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

**login authentication** {**default***list-name*}
**no login authentication**

## Syntax Description

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

## Command Default

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

## Command Modes

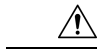
Line template configuration

## Command History

| Release | Modification |
|---|---|
| Release 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.

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read, write |
| tty-access | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
```

```
RP/0/RP0RSP0/CPU0:router(config)# line template template1
RP/0/RP0RSP0/CPU0:router(config-line)# login authentication default
```

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# line template template2
RP/0/RP0RSP0/CPU0:router(config-line)# login authentication list1
```

## Related Commands

| Command | Description |
|---|---|
| aaa authentication, on page 19 | Creates a method list for authentication. |

# 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

| | |
|---|---|
| **0** | (Optional) Specifies that an unencrypted clear-text password follows. |
| **7** | Specifies that an encrypted password follows. |
| *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.<br>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

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

## 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

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

## Examples

The following example shows how to establish the unencrypted password *pwd1* for user. The output from the **show** command displays the password in its encrypted form.

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# username user1
RP/0/RP0RSP0/CPU0:router(config-un)# password 0 pwd1
RP/0/RP0RSP0/CPU0:router(config-un)# commit
RP/0/RP0RSP0/CPU0:router(config-un)# show running-config
Building configuration...
username user1
 password 7 141B1309
```

## Related Commands

| Command | Description |
|---|---|
| group (AAA), on page 41 | Adds a user to a group. |
| usergroup, on page 122 | Accesses user group configuration mode and configures a user group, associating it with a set of task groups. |
| username, on page 123 | Accesses username configuration mode and configures a new user with a username, establishing a password and granting permissions for that user. |
| line | Enters line template configuration mode for the specified line template.<br>For more information, see the Cisco IOS XR *System Management Command Reference.* |

# policy (AAA)

To configure a policy that is common for user password as well as secret, use the **policy** command in username configuration mode. To remove this configuration, use the **no** form of this command.

**policy** *policy-name*

## Syntax Description

| | |
|---|---|
| *policy-name* | Specifies the name of the policy that is common for user password as well as secret. |

## Command Default

None

## Command Modes

username

## Command History

| Release | Modification |
|---|---|
| Release 7.2.1 | This command was introduced. |

## Usage Guidelines

For detailed usage guidelines for this command, see the *Guidelines to Configure Password Policy for User Secret* section in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

## Task ID

| Task ID | Operation |
|---|---|
| aaa | read, write |

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

```
Router#configure
Router(config)#username user1
Router(config-un)#policy test-policy1
Router(config-un)#secret 10
$6$dmwuW0Ajicf98W0.$y/vzynWF1/OcGxwBwHs79VAy5ZZLhoHd7TicR4mOo8IIVriYCGAKW0A.w1JvTPO7IbZry.DxHrE3SN2BBzBJe0
Router(config-un)#commit
```

## Related Commands

| Command | Description |
|---|---|
| aaa password-policy, on page 32 | Defines the FIPS-compliant AAA password security policy. |
| username, on page 123 | |

# 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 to120 seconds. If the *seconds* argument is not configured, the number of seconds ranges from 10 to 60, depending on the transaction rate of the server.

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

## Command Default

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

## Command Modes

Global configuration

## Command History

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

## Usage Guidelines

**Note** 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# radius-server dead-criteria time 5
```

**Related Commands**

| Command | Description |
|---|---|
| radius-server dead-criteria tries, on page 55 | Specifies the number of consecutive timeouts that must occur on the router before the RADIUS server is marked as dead. |
| radius-server deadtime, on page 57 | Defines the length of time, in minutes, for a RADIUS server to remain marked dead. |
| show radius dead-criteria, on page 89 | Displays information for the dead-server detection criteria. |

# 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 |
|---|---|
| Release 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

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# radius-server dead-criteria tries 4
```

**Related Commands**

| Command | Description |
|---|---|
| radius-server dead-criteria time, on page 53 | 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. |
| radius-server deadtime, on page 57 | Defines the length of time, in minutes, for a RADIUS server to remain marked dead. |
| show radius dead-criteria, on page 89 | Displays information for the dead-server detection criteria. |

# 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 Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

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

## 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

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

## 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/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# radius-server deadtime 5
```

# radius-server host

To specify a RADIUS server host, use the **radius-server host** command in Global Configuration modeXR 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.<br>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.<br>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

Global Configuration modeXR Config mode

## Command History

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

## 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.

## Task ID

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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

| Command | Description |
|---|---|
| **aaa accounting subscriber** | Creates a method list for accounting. |
| **aaa authentication subscriber** | Creates a method list for authentication. |
| **aaa authorization subscriber** | Creates a method list for authorization. |
| radius-server key, on page 60 | Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon. |
| radius-server retransmit, on page 61 | Specifies how many times Cisco IOS XR software retransmits packets to a server before giving up. |
| radius-server timeout, on page 62 | Sets the interval a router waits for a server host to reply. |

# 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 Global Configuration modeXR Config mode. To disable the key, use the **no** form of this command.

**radius-server key** {**0** *clear-text-key* | **7** *encrypted-keyclear-text-key*}
**no radius-server key**

## Syntax Description

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

## Command Default

The authentication and encryption key is disabled.

## Command Modes

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

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

## Task ID

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# radius-server key 0 samplekey
```

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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 Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

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

## Task ID

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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 Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

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

## Task ID

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# radius-server timeout 10
```

# restrict-consecutive-characters

To restrict consecutive characters (that includes regular English alphabets, and English alphabets from QWERTY keyboard layout and numbers), for user passwords and secrets, use the **restrict-consecutive-characters** command in *aaa password-policy* configuration mode. To disable the feature, use the **no** form of the command.

**restrict-consecutive-characters** { **english-alphabet** | **qwerty-keyboard** } *num-of-chars* [**cyclic-wrap**]

## Syntax Description

| | |
|---|---|
| **english-alphabet** | Restricts consecutive English alphabets for user passwords and secrets.<br>For example, "abcd", "wxyz", and so on. |
| **qwerty-keyboard** | Restricts consecutive English alphabets from QWERTY keyboard layout and numbers, for user passwords and secrets.<br>For example, "qwer", "mnbv", "7890", and so on. |
| *num-of-chars* | Specifies the number of consecutive characters to be restricted for user passwords and secrets.<br>Range is 2 to 26, for **english-alphabet**.<br>Range is 2 to 10, for **qwerty-keyboard**. |
| **cyclic-wrap** | Restricts cyclic wrapping of the alphabet or the number for user passwords and secrets.<br>For example, "yzab", "opqw", "9012", and so on. |

## Command Default

Disabled, by default.

## Command Modes

aaa password-policy configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 7.7.1 | This command was introduced. |

## Usage Guidelines

All password policies are applicable only to locally configured users.

After creating the password policy, you must explicitly apply that policy to the user profiles to have an effect of that policy in the password and secret configuration.

For more details about the feature and configuration task, see the section *Enhanced Security for User Passwords and Secrets* in *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

## Task ID

| Task ID | Operation |
|---|---|
| aaa | read, write |

This example shows how to configure a AAA password policy that restricts cyclic wrapping of 4 consecutive English alphabets and 6 consecutive characters from QWERTY keyboard.

```
Router(config)#aaa password-policy test-policy
Router(config-pp)#restrict-consecutive-characters english-alphabet 4 cyclic-wrap
Router(config-pp)#restrict-consecutive-characters qwerty-keyboard 6
```

This example shows how to apply the password policy to the user profile, *user1*:

```
Router(config)#username user1
Router(config-un)#policy test-policy
Router(config-un)#commit
```

**Related Commands**

| Command | Description |
|---|---|
| aaa password-policy, on page 32 | Defines the FIPS-compliant AAA password security policy. |

# 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

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

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

## Examples

The following example shows how to set the retransmit value:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# aaa group server radius group1
RP/0/RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0/RSP0/CPU0:router(config-sg-radius-private)# retransmit 100
```

## Related Commands

| Command | Description |
|---|---|
| aaa group server tacacs+ | Groups different RADIUS server hosts into distinct lists. |
| server-private (RADIUS) | Configures the IP address of the private RADIUS server for the group server. |
| timeout (RADIUS), on page 120 | Specifies the number of seconds the router waits for the RADIUS server to reply before retransmitting. |

# 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

| | |
|---|---|
| **0** | (Optional) Specifies that an unencrypted (clear-text) password follows. The password will be encrypted for storage in the configuration using an MD5 encryption algorithm. Otherwise, the password is not encrypted. |
| **5** | Specifies that an encrypted MD5 password (secret) follows. |
| **8** | (Optional) Specifies that SHA256-encrypted password follows. |
| **9** | (Optional) Specifies that scrypt-encrypted password follows. |
| **10** | (Optional) Specifies that SHA512-encrypted password follows. |
| *secret-login* | Text string in alphanumeric characters that is stored as the MD5-encrypted password entered by the user in association with the user's login ID.<br>Can be up to 253 characters in length.<br>**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

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

## 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 EXEC modeXR 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.

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

## Task ID

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

## Examples

The following example shows how to establish the clear-text secret "lab" for the user *user2*:

```
Router# configure
Router(config)# username user2
Router(config-un)# secret 0 lab
Router(config-un)# commit
Router(config-un)# show running-config
Building configuration...
username user2
 secret 5 $1$DTmd$q7C6fhzje7Cc7Xzmu2Frx1
!
end
```

The following examples show how to configure a Type 10 (SHA512) password for the user, *user10*. You can also see the examples and usage of the username, on page 123 command.

You can specify Type as '10' under the **secret** keyword, to explicitly configure Type 10 password.

```
Router#configure
Router(config)#username user10 secret 10
$6$9UvJidvsTEqgkAPU$3CL1Ei/F.E4v/Hi.UaqLwX8UsSEr9ApG6c5pzhMJmZtgW4jObAQ7meAwyhu5VM/aRFJqe/jxZG17h6xPrvJWf1
Router(config-un)#commit
```

You can also use the **enc-type** keyword under the **secret 0** option, to specify Type 10 as the encryption for a password entered in clear text.

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

# server (RADIUS)

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

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

## Syntax Description

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

## Command Default

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

- Authentication port: 1645
- Accounting port: 1646

## Command Modes

RADIUS server-group configuration

## Command History

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

## Usage Guidelines

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

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

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

## Task ID

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server 10.1.1.1 auth-port 1645 acct-port 1646
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server 10.2.2.2 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

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

## 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

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server tacacs+ tac1
RP/0/RP0RSP0/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.<br>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.<br>The *retries* argument specifies the retransmit value. The range is from 1 to 100. If no retransmit value is specified, the global value is used. |
| **key** *string* | (Optional) Specifies the authentication and encryption key that is used between the router and the RADIUS daemon running on the RADIUS server. This key overrides the global setting of the **radius-server key** command. If no key string is specified, the global value is used. |

## Command Default

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

- Authentication port: 1645
- Accounting port: 1646

## Command Modes

RADIUS server-group configuration

## Command History

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

## 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.

## Task ID

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

## 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:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 timeout 5
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 retransmit 3
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 key coke
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0/RP0RSP0/CPU0:router(config-sg-radius-private)# exit
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 timeout 5
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 retransmit 3
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 key coke
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.2.2.2 auth-port 300
RP/0/RP0RSP0/CPU0:router(config-sg-radius-private)#

RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0/RP0RSP0/CPU0:router(config-sg-radius-private)# exit
(config-sg-radius)# server-private 10.2.2.2 auth-port 300
RP/0/RP0RSP0/CPU0:router(config-sg-radius-private)#
```

# server-private (TACACS+)

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

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

## Syntax Description

| | |
|---|---|
| *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

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

## 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

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

## Examples

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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server tacacs+ myserver
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs+)# server-private 10.1.1.1 timeout 5
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs+)# server-private 10.1.1.1 key a_secret
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs+)# server-private 10.1.1.1 port 51
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs-private)# exit
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs+)# server-private 10.2.2.2 timeout 5
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs+)# server-private 10.2.2.2 key coke
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs+)# server-private 10.2.2.2 port 300
RP/0/RP0RSP0/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.<br>**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

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

## 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

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/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/RP0RSP0/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/RP0RSP0/CPU0:router# show aaa taskgroup netadmin

Task group 'netadmin'

Task group 'netadmin' has the following combined set
  of task IDs (including all inherited groups):

Task:                  aaa  : READ
Task:                  acl  : READ    WRITE    EXECUTE    DEBUG
Task:                admin  : READ
Task:                 ancp  : READ    WRITE    EXECUTE    DEBUG
Task:                  atm  : READ    WRITE    EXECUTE    DEBUG
Task:       basic-services  : READ    WRITE    EXECUTE    DEBUG
Task:                bcdl  : READ
Task:                  bfd  : READ    WRITE    EXECUTE    DEBUG
Task:                  bgp  : READ    WRITE    EXECUTE    DEBUG
```

```
Task:                boot  : READ    WRITE    EXECUTE    DEBUG
Task:              bundle  : READ    WRITE    EXECUTE    DEBUG
Task:                 cdp  : READ    WRITE    EXECUTE    DEBUG
Task:                 cef  : READ    WRITE    EXECUTE    DEBUG
Task:                 cgn  : 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
Task:                dwdm  : READ    WRITE    EXECUTE    DEBUG
Task:                 eem  : READ    WRITE    EXECUTE    DEBUG
Task:               eigrp  : READ    WRITE    EXECUTE    DEBUG
Task:    ethernet-services  : READ
Task:           ext-access  : READ    WRITE    EXECUTE    DEBUG
Task:               fabric  : READ    WRITE    EXECUTE    DEBUG
Task:           fault-mgr  : READ    WRITE    EXECUTE    DEBUG
Task:           filesystem  : READ    WRITE    EXECUTE    DEBUG
Task:             firewall  : READ    WRITE    EXECUTE    DEBUG
Task:                   fr  : READ    WRITE    EXECUTE    DEBUG
Task:                 hdlc  : READ    WRITE    EXECUTE    DEBUG
Task:       host-services  : READ    WRITE    EXECUTE    DEBUG
Task:                 hsrp  : READ    WRITE    EXECUTE    DEBUG
Task:           interface  : READ    WRITE    EXECUTE    DEBUG
Task:           inventory  : READ
Task:         ip-services  : READ    WRITE    EXECUTE    DEBUG
Task:                ipv4  : READ    WRITE    EXECUTE    DEBUG
Task:                ipv6  : READ    WRITE    EXECUTE    DEBUG
Task:                isis  : READ    WRITE    EXECUTE    DEBUG
Task:               l2vpn  : READ    WRITE    EXECUTE    DEBUG
Task:                  li  : READ    WRITE    EXECUTE    DEBUG
Task:             logging  : READ    WRITE    EXECUTE    DEBUG
Task:                lpts  : READ    WRITE    EXECUTE    DEBUG
Task:             monitor  : READ
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
Task:             pos-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                                (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
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                                (reserved)
Task:                vlan  : READ    WRITE    EXECUTE    DEBUG
Task:                vrrp  : READ    WRITE    EXECUTE    DEBUG
```

The following sample output is from the **show aaa** command, using the **taskgroup** keyword for an operator. The task group operator has the following combined set of task IDs, which includes all inherited groups:

```
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:                  acl admin  : 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:      config-services           crypto  : READ    WRITE    EXECUTE    DEBUG
Task:              crypto   diag  : READ    WRITE    EXECUTE    DEBUG
Task:                diag  drivers  : READ    WRITE    EXECUTE    DEBUG
Task:             drivers ext-access  : READ    WRITE    EXECUTE    DEBUG
Task:          ext-access     fabric  : READ    WRITE    EXECUTE    DEBUG
Task:              fabric  fault-mgr  : READ    WRITE    EXECUTE    DEBUG
Task:           fault-mgr filesystem  : READ    WRITE    EXECUTE    DEBUG
Task:          filesystem         fr  : READ    WRITE    EXECUTE    DEBUG
Task:                  fr hdlc  : READ    WRITE    EXECUTE    DEBUG
Task:                hdlc  host-services  : READ    WRITE    EXECUTE    DEBUG
Task:        host-services           hsrp  : 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:          ip-services         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:             monitor mpls-ldp  : 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:             network     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:               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
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:

```
RP/0/RP0RSP0/CPU0:router# show aaa userdb

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/RP0RSP0/CPU0:router# show aaa task supported

aaa
acl
admin
atm
basic-services
bcdl
bfd
bgp
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
```

```
ipv6
isis
logging
lpts
monitor
mpls-ldp
mpls-static
mpls-te
multicast
netflow
network
ospf
ouni
pkg-mgmt
pos-dpt
ppp
qos
rib
rip
User group root-systemlr
root-system
route-map
route-policy
sbc
snmp
sonet-sdh
static
sysmgr
system
transport
tty-access
tunnel
universal
vlan
vrrp
```

## Related Commands

| Command | Description |
|---|---|
| show user, on page 101 | Displays task IDs enabled for the currently logged-in user. |

# show aaa password-policy

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

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

## Syntax Description

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

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | 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

| Task ID | Operation |
|---|---|
| aaa | read |

This is a sample out of **show aaa password-policy** command:

```
RP/0/RP0RSP0/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**

| Command | Description |
|---|---|
| aaa password-policy, on page 32 | Defines the FIPS-compliant AAA password security policy. |

# show radius accounting

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

**show radius accounting**

## 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

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/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*

| Field | Description |
|---|---|
| Server | Server IP address/UDP destination port for authentication requests; UDP destination port for accounting requests. |

# show radius authentication

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

**show radius authentication**

## Syntax Description

This command has no keywords or arguments.

## Command Default

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

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/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*

| Field | Description |
|---|---|
| Server | Server IP address/UDP destination port for authentication requests; UDP destination port for accounting requests. |

# show radius

To display information about the RADIUS servers that are configured in the system, use the **show radius** command in the EXEC modeXR 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

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

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

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# show radius

Global dead time: 0 minute(s)

Server: 10.1.1.1/1645/1646  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: 10.2.2.2/1645/1646  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
```

```
    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 4: show radius Field Descriptions***

| Field | Description |
|---|---|
| Server | Server IP address/UDP destination port for authentication requests/UDP destination port for accounting requests. |
| Timeout | Number of seconds the router waits for a server host to reply before timing out. |
| Retransmit limit | Number of times the Cisco IOS XR software searches the list of RADIUS server hosts before giving up. |

# show radius dead-criteria

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

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

## Syntax Description

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

## Command Default

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

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/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***

| Field | Description |
|---|---|
| Server | Server IP address/UDP destination port for authentication requests/UDP destination port for accounting requests. |
| Timeout | Number of seconds the router waits for a server host to reply before timing out. |

| Field | Description |
|---|---|
| Retransmits | Number of times Cisco IOS XR software searches the list of RADIUS server hosts before giving up. |

# 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 EXEC modeXR EXEC mode.

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

## Syntax Description

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

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# show radius server-groups

Global list of servers
  Contains 2 server(s)
    Server 10.1.1.1/1645/1646
    Server 10.2.2.2/1645/1646

Server group 'radgrp1' has 2 server(s)
  Dead time: 0 minute(s) (inherited from global)
  Contains 2 server(s)
    Server 10.1.1.1/1645/1646
    Server 10.2.2.2/1645/1646

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/RP0RSP0/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 10.1.1.1/1645/1646
    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/RP0RSP0/CPU0:router# show radius server-groups radgrp-priv detail

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

This table describes the significant fields shown in the display.

***Table 6: show radius server-groups Field Descriptions***

| Field | Description |
|---|---|
| Server | Server IP address/UDP destination port for authentication requests/UDP destination port for accounting requests. |

# show tacacs

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

**show tacacs**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

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

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# show tacacs

For IPv4 IP addresses:
Server:10.1.1.1/21212 opens=0 closes=0 aborts=0 errors=0
        packets in=0 packets out=0
        status=up single-connect=false

Server:10.2.2.2/21232 opens=0 closes=0 aborts=0 errors=0
        packets in=0 packets out=0
        status=up single-connect=false

For IPv6 IP addresses:
Server: 10.2.3.5/49  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 7: show tacacs Field Descriptions***

| Field | Description |
|---|---|
| Server | Server IP address. |
| opens | Number of socket opens to the external server. |

| Field | Description |
|---|---|
| closes | Number of socket closes to the external server. |
| aborts | Number of tacacs requests that have been terminated midway. |
| errors | Number of error replies from the external server. |
| packets in | Number of TCP packets that have been received from the external server. |
| packets out | Number of TCP packets that have been sent to the external server. |

# show tacacs counters

To display statistics of authentication, executive and command authorization, and executive and command accounting for each TACACS+ servers configured in the system, use the **show tacacs counters** command in the EXEC modeXR EXEC mode.

**show tacacs counters**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.5.4 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

The following is a sample output from the **show tacacs counters** command:

```
RP/0/RP0RSP0/CPU0:router# show tacacs counters

TACACS+ Server:   10.105.236.101/4010 [global]

  Authentication:
   10 requests, 4 accepts, 3 failure, 2 error, 1 timeout

  Exec Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout

  Command Authorization:
   6 requests, 6 accepts, 0 denied, 0 error, 0 timeout

  Exec Accounting:
   0 requests, 0 accepts, 0 fail, 0 error, 0 timeout

  Command Accounting:
   6 requests, 6 accepts, 0 fail, 0 error, 0 timeout

TACACS+ Server:   10.105.236.101/2201 [private] vrf = default

  Authentication:
   0 requests, 0 accepts, 0 failure, 0 error, 0 timeout

  Exec Authorization:
   0 requests, 0 accepts, 0 denied, 0 error, 0 timeout
```

```
Command Authorization:
 0 requests, 0 accepts, 0 denied, 0 error, 0 timeout

Exec Accounting:
 0 requests, 0 accepts, 0 fail, 0 error, 0 timeout

Command Accounting:
 0 requests, 0 accepts, 0 fail, 0 error, 0 timeout
```

# show tacacs details

To display detailed information about the TACACS+ server and server groups that are configured in the system, use the **show tacacs details** command in the EXEC modeXR EXEC mode.

**show tacacs details**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.5.4 | This command was introduced. |

## Usage Guidelines

Use the **show tacacs details** 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

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# show tacacs details

TACACS+ Server                                                  : 10.105.236.101/4010
[Global]
    Family                                                      : IPv4
    Timeout(in secs)                                            : 3
    Connection Opens                                            : 8
    Connection Closes                                           : 8
    Requests sent                                               : 6
    Response received                                           : 6
    Packets Abort                                               : 2
    Server State                                                : Down
    Server On-Hold                                              : True
    Tacacs-Single-Connect                                       : False
    Tacacs-Single-Connect-Idle-Timeout(in secs)                 : 0
    Last Connection Attempted                                   : 08:32:43 UTC Tue Aug
02 2022


TACACS+ Server                                                  : 10.105.236.101/8010
[Private] vrf=default
    Family                                                      : IPv4
    Timeout(in secs)                                            : 3
    Connection Opens                                            : 8
    Connection Closes                                           : 7
```

```
    Requests sent                                                   : 7
    Response received                                               : 7
    Packets Abort                                                   : 0
    Server State                                                    : Up
    Server On-Hold                                                  : False
    Tacacs-Single-Connect                                           : False
    Tacacs-Single-Connect-Idle-Timeout(in secs)                     : 0
    Last Connection Attempted                                       : 08:32:52 UTC Tue Aug
02 2022


TACACS+ Server-groups:

Global list of servers
    Server 10.105.236.101/4010 family=IPv4

Server group 'tac1' has 1 servers
    Servers in this group are under 'default' vrf
    Server 10.105.236.101/8010 [private] family=IPv4

TACACS+ Source-Interface:

Interface                                          VRF Id
IPV4-Address

GigabitEthernet0/0/0/0                            0x60000001
0.0.0.0
MgmtEth0/RP0/CPU0/0                               0x60000000
192.168.122.222


Interface                                           VRF Id
IPV6-Address

GigabitEthernet0/0/0/0                            0x60000001
 ::
MgmtEth0/RP0/CPU0/0                               0x60000000
 ::
```

# 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 EXEC modeXR EXEC mode.

**show tacacs server-groups**

1,1,1

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# show tacacs server-groups

Global list of servers
  Server 192.168.25.61/23456
  Server 192.168.49.12/12345
  Server 192.168.49.12/9000
  Server 192.168.25.61/23432
  Server 10.5.5.5/23456
  Server 10.1.1.1/49
Server group 'tac100' has 1 servers
Server 192.168.49.12
```

This table describes the significant fields shown in the display.

***Table 8: show tacacs server-groups Field Descriptions***

| Field | Description |
|---|---|
| Server | Server IP address. |

# show tacacs source-interface

To display information about the source interface for the TACACS+ server that are configured in the system, use the **show tacacs source-interface** command in the EXEC modeXR EXEC mode.

**show tacacs source-interface**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.5.4 | This command was introduced. |

## Usage Guidelines

Use the **show tacacs source-interface** command to display source interface information about each configured TACACS+ server, including the interface name, vrf-id, and IPv4 and Ipv6 address.

## Task ID

| Task ID | Operations |
|---|---|
| aaa | read |

## Examples

The following is sample output from the **show tacacs source-interface** command:

```
RP/0/RP0RSP0/CPU0:router# show tacacs source-interface
Interface                            VRF Id                          IPV4-Address

MgmtEth0/RP0/CPU0/0                 0x60000000                     192.168.122.222


Interface                            VRF Id                          IPV6-Address

MgmtEth0/RP0/CPU0/0                  0x60000000                           ::
```

# show user

To display all user groups and task IDs associated with the currently logged-in user, use the **show user** command in the EXEC modeXR 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

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

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

## Task ID

| Task ID | Operations |
|---|---|
| none | — |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# show user authentication method

local
```

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

```
RP/0/RP0RSP0/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:

```
RP/0/RP0RSP0/CPU0:router# 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:                  aaa  : READ    WRITE    EXECUTE    DEBUG
Task:                  acl  : READ    WRITE    EXECUTE    DEBUG
Task:                admin  : READ    WRITE    EXECUTE    DEBUG
Task:                  atm  : READ    WRITE    EXECUTE    DEBUG
Task:       basic-services  : READ    WRITE    EXECUTE    DEBUG
Task:                 bcdl  : READ    WRITE    EXECUTE    DEBUG
Task:                  bfd  : READ    WRITE    EXECUTE    DEBUG
Task:                  bgp  : READ    WRITE    EXECUTE    DEBUG
Task:                 boot  : READ    WRITE    EXECUTE    DEBUG
Task:               bundle  : READ    WRITE    EXECUTE    DEBUG
Task:                  cdp  : READ    WRITE    EXECUTE    DEBUG
Task:                  cef  : READ    WRITE    EXECUTE    DEBUG
Task:          config-mgmt  : READ    WRITE    EXECUTE    DEBUG
Task:      config-services  : READ    WRITE    EXECUTE    DEBUG
Task:               crypto  : READ    WRITE    EXECUTE    DEBUG
Task:                 diag  : READ    WRITE    EXECUTE    DEBUG
Task:              drivers  : READ    WRITE    EXECUTE    DEBUG
Task:           ext-access  : READ    WRITE    EXECUTE    DEBUG
Task:               fabric  : READ    WRITE    EXECUTE    DEBUG
Task:           fault-mgr  : READ    WRITE    EXECUTE    DEBUG
Task:           filesystem  : READ    WRITE    EXECUTE    DEBUG
Task:             firewall  : READ    WRITE    EXECUTE    DEBUG
Task:                   fr  : READ    WRITE    EXECUTE    DEBUG
Task:                 hdlc  : READ    WRITE    EXECUTE    DEBUG
Task:        host-services  : READ    WRITE    EXECUTE    DEBUG
Task:                 hsrp  : READ    WRITE    EXECUTE    DEBUG
Task:            interface  : READ    WRITE    EXECUTE    DEBUG
Task:            inventory  : READ    WRITE    EXECUTE    DEBUG
Task:          ip-services  : READ    WRITE    EXECUTE    DEBUG
Task:                 ipv4  : READ    WRITE    EXECUTE    DEBUG
Task:                 ipv6  : READ    WRITE    EXECUTE    DEBUG
Task:                 isis  : READ    WRITE    EXECUTE    DEBUG
Task:              logging  : READ    WRITE    EXECUTE    DEBUG
Task:                 lpts  : READ    WRITE    EXECUTE    DEBUG
Task:              monitor  : READ    WRITE    EXECUTE    DEBUG
Task:              mpls-ldp  : READ    WRITE    EXECUTE    DEBUG
Task:              mpls-static  : READ    WRITE    EXECUTE    DEBUG
Task:              mpls-te  : READ    WRITE    EXECUTE    DEBUG
Task:              multicast  : READ    WRITE    EXECUTE    DEBUG
Task:              netflow  : READ    WRITE    EXECUTE    DEBUG
Task:              network  : READ    WRITE    EXECUTE    DEBUG
Task:                 ospf  : READ    WRITE    EXECUTE    DEBUG
Task:                 ouni  : READ    WRITE    EXECUTE    DEBUG
Task:             pkg-mgmt  : READ    WRITE    EXECUTE    DEBUG
Task:                  ppp  : READ    WRITE    EXECUTE    DEBUG
Task:                  qos  : READ    WRITE    EXECUTE    DEBUG
Task:                  rib  : READ    WRITE    EXECUTE    DEBUG
Task:                  rip  : READ    WRITE    EXECUTE    DEBUG
Task:              root-lr  : READ    WRITE    EXECUTE    DEBUG (reserved)
Task:          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/RP0RSP0/CPU0:router# show user tasks

Task:                  aaa  : READ    WRITE    EXECUTE    DEBUG
Task:                  aaa  : READ    WRITE    EXECUTE    DEBUG
Task:                  acl  : READ    WRITE    EXECUTE    DEBUG
Task:                admin  : READ    WRITE    EXECUTE    DEBUG
Task:                  atm  : READ    WRITE    EXECUTE    DEBUG
Task:       basic-services  : READ    WRITE    EXECUTE    DEBUG
Task:                 bcdl  : READ    WRITE    EXECUTE    DEBUG
Task:                  bfd  : READ    WRITE    EXECUTE    DEBUG
Task:                  bgp  : READ    WRITE    EXECUTE    DEBUG
Task:                 boot  : READ    WRITE    EXECUTE    DEBUG
Task:               bundle  : READ    WRITE    EXECUTE    DEBUG
Task:                  cdp  : READ    WRITE    EXECUTE    DEBUG
Task:                  cef  : READ    WRITE    EXECUTE    DEBUG
Task:          config-mgmt  : READ    WRITE    EXECUTE    DEBUG
Task:      config-services  : READ    WRITE    EXECUTE    DEBUG
Task:               crypto  : READ    WRITE    EXECUTE    DEBUG
Task:                 diag  : READ    WRITE    EXECUTE    DEBUG
Task:              drivers  : READ    WRITE    EXECUTE    DEBUG
Task:           ext-access  : READ    WRITE    EXECUTE    DEBUG
Task:               fabric  : READ    WRITE    EXECUTE    DEBUG
Task:            fault-mgr  : READ    WRITE    EXECUTE    DEBUG
Task:           filesystem  : READ    WRITE    EXECUTE    DEBUG
Task:             firewall  : READ    WRITE    EXECUTE    DEBUG
Task:                   fr  : READ    WRITE    EXECUTE    DEBUG
Task:                 hdlc  : READ    WRITE    EXECUTE    DEBUG
Task:        host-services  : READ    WRITE    EXECUTE    DEBUG
Task:                 hsrp  : READ    WRITE    EXECUTE    DEBUG
Task:            interface  : READ    WRITE    EXECUTE    DEBUG
Task:            inventory  : READ    WRITE    EXECUTE    DEBUG
Task:          ip-services  : READ    WRITE    EXECUTE    DEBUG
Task:                 ipv4  : READ    WRITE    EXECUTE    DEBUG
Task:                 ipv6  : READ    WRITE    EXECUTE    DEBUG
Task:                 isis  : READ    WRITE    EXECUTE    DEBUG
Task:              logging  : READ    WRITE    EXECUTE    DEBUG
Task:                 lpts  : READ    WRITE    EXECUTE    DEBUG
Task:              monitor  : READ    WRITE    EXECUTE    DEBUG
Task:              mpls-ldp  : READ    WRITE    EXECUTE    DEBUG
Task:              mpls-static  : READ    WRITE    EXECUTE    DEBUG
Task:              mpls-te  : READ    WRITE    EXECUTE    DEBUG
Task:              multicast  : READ    WRITE    EXECUTE    DEBUG
Task:              netflow  : READ    WRITE    EXECUTE    DEBUG
Task:              network  : READ    WRITE    EXECUTE    DEBUG
Task:                 ospf  : READ    WRITE    EXECUTE    DEBUG
Task:                 ouni  : READ    WRITE    EXECUTE    DEBUG
Task:              pkg-mgmt  : READ    WRITE    EXECUTE    DEBUG
Task:                  ppp  : READ    WRITE    EXECUTE    DEBUG
Task:                  qos  : READ    WRITE    EXECUTE    DEBUG
Task:                  rib  : READ    WRITE    EXECUTE    DEBUG
Task:                  rip  : READ    WRITE    EXECUTE    DEBUG
```

```
Task:              root-lr  : READ    WRITE    EXECUTE    DEBUG (reserved)
Task:         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
```

# 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

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226
RP/0/RP0RSP0/CPU0:router(config-tacacs-host)# single-connection
```

# single-connection-idle-timeout

To set the idle timeout value for the single TCP connection to the TACACS+ server, use the **single-connection-idle-timeout** command in *tacacs-server host* configuration mode. To remove the configuration or to disable the idle timeout for the single connection, use the **no** form of this command.

**single-connection-idle-timeout** *time-in-seconds*

## Syntax Description

| | |
|---|---|
| *time-in-seconds* | Specifies the single connection idle timeout value, in seconds.<br>The range is:<br>• 500 to 7200 (prior to Cisco IOS XR Software Release 7.3.2)<br>• 5 to 7200 (from Cisco IOS XR Software Release 7.3.2, and later) |

## Command Default

Single connection idle timeout is not set, by default.

## Command Modes

tacacs-server host

## Command History

| Release | Modification |
|---|---|
| Release 7.3.2 | This command was modified to change the single connection idle timeout range. |
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

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

## Examples

This example shows how to set an idle timeout value of 60 seconds for the single TCP connections to the TACACS+ server:

```
RP/0/RP0RSP0/CPU0:router(config)#tacacs-server host 209.165.200.226
RP/0/RP0RSP0/CPU0:router(config-tacacs-host)#single-connection-idle-timeout 60
RP/0/RP0RSP0/CPU0:router(config-tacacs-host)#commit
```

## Related Commands

| Command | Description |
|---|---|
| single-connection, on page 105 | Multiplexes all TACACS+ requests to the server over a single TCP connection. |

# tacacs-server host

To specify a TACACS+ host server, use the **tacacs-server host** command in Global Configuration modeXR 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**]
[ **single-connection-idle-timeout** *time-in-seconds* ]

## Syntax Description

| | |
|---|---|
| *host-name* | Host or domain name or IP address of the TACACS+ server. |
| **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 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.<br>Note: You can use this parameter only in the config-tacacs-host sub-mode. |
| **key** [**0** \| **7**] *auth-key* | (Optional) Specifies an authentication and encryption key shared between the AAA server and the TACACS+ server. The TACACS+ packets are encrypted using this key. This key 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.<br>(Optional) Entering **0** specifies that an unencrypted (clear-text) key follows.<br>(Optional) Entering **7** specifies that an encrypted key follows.<br>The *auth-key* argument specifies the unencrypted key between the AAA server and the TACACS+ server.<br>Note: You can use this parameter only in the config-tacacs-host sub-mode. |
| **single-connection** | (Optional) Multiplexes all TACACS+ requests to this server over a single TCP connection. By default, a separate connection is used for each session.<br>Note: You can use this parameter only in the config-tacacs-host sub-mode. |
| **single-connection-idle-timeout** *time-in-seconds* | (Optional) Specifies the single connection idle timeout value, in seconds.<br>The range is:<br>• 500 to 7200 (prior to Cisco IOS XR Software Release 7.4.1/Release 7.3.2/Release 6.8.1)<br>• 5 to 7200 (from Cisco IOS XR Software Release 7.4.1/Release 7.3.2/Release 6.8.1, and later) |

## 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.

Single connection idle timeout is not set, by default.

**Command Modes**

Global Configuration modeXR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.3.2 | This command was modified to change the range for **single-connection-idle-timeout**. |
| Release 7.0.12 | This command was introduced. |

**Usage Guidelines**

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**

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

**Examples**

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

```
RP/0/RP0RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226
RP/0/RP0RSP0/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/RP0RSP0/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/RP0RSP0/CPU0:router(config)# tacacs-server host host1 port 51
RP/0/RP0RSP0/CPU0:router(config-tacacs-host)# timeout 30
RP/0/RP0RSP0/CPU0:router(config-tacacs-host)# key a_secret
```

# tacacs-server ipv4

To set the Differentiated Services Code Point (DSCP), which is represented by the first six bits in the Type of Service (ToS) byte of the IP header, use the **tacacs-server ipv4** command in Global Configuration modeXR Config mode.

**tacacs-server ipv4 dscp** *dscp-value*

## Syntax Description

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

**Command Default**

None

**Command Modes**

Global Configuration modeXR Config mode

**Command History**

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

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

| Task ID | Operation |
|---|---|
| aaa | read, write |

**Examples**

The following example sets the DSCP value to Assured Forwarding (AF)11:

```
RP/0/RP0RSP0/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 Global Configuration modeXR Config mode. To disable the key, use the **no** form of this command.

**tacacs-server key** {**0** *clear-text-key* | **7** *encrypted-keyauth-key*}
**no tacacs-server key** {**0** *clear-text-key* | **7** *encrypted-keyauth-key*}

## Syntax Description

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

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/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 Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

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

## 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/RP0RSP0/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 Global Configuration modeXR 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.<br>**Note** Use the **show interfaces** command in Global Configuration modeXR Config mode to see a list of all interfaces currently configured on the router.<br>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

Global Configuration modeXR Config mode

## Command History

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

## 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

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

## Examples

The following example shows how to set the IP address of the specified interface for all outgoing TACACS+ packets:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# tacacs source-interface HundredGigabitEthernet 0/0/0/29
vrf abc
```

# task

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

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

## Syntax Description

| | |
|---|---|
| read | Enables read-only privileges for the named task ID. |
| write | Enables write privileges for the named task ID. The term "write" implies read also. |
| execute | Enables execute privileges for the named task ID. |
| debug | Enables debug privileges for the named task ID. |
| *taskid-name* | Name of the task ID. |

## Command Default

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

## Command Modes

Task group configuration

## Command History

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

## Usage Guidelines

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

Task IDs are the base of command authorization. Only users who have the required permissions can execute a particular command on the router. To execute a command, the user must be part of a user group that consists of task group(s) that includes required task IDs and privileges. Cisco IOS XR software supports multiple task IDs. For example, **aaa**, **config-services**, **crypto**, **system**, and so on. To see the list of task IDs available for the user, use the **show user tasks** command.

Likewise, all commands are associated with one or more task IDs, and their corresponding operations (such as **read**, **write**, **execute**, and **debug**) that denote the permissions required to execute those commands. You can use the **describe** command to know the task ID and permissions that are required to execute a particular command.

For example, the following output shows that the user needs **aaa** task ID with **read** and **write** permission to execute the **show run aaa** command. So, users can execute this command if they belong to a user group associated with a task group that includes this **aaa** task ID having read and write privileges.

```
Router# describe show run aaa
The command is defined in aaa_cmds.parser

User needs ALL of the following taskids:

    aaa (READ WRITE) ----------->

It will take the following actions:
```

```
Wed Mar 16 07:58:01.451 UTC
  Spawn the process:
    nvgen "-c" "-q" "gl/aaa/"
Router#
```

Root users (users in **root-lr** or **root-system** user group) have all task IDs, and hence will be able to execute all commands. Also, certain commands might not require any task ID as such to execute it. So, all users will have permission to execute such commands. If you do not have the required permission to execute a command, the command authorization fails. If the user group assignment is preventing you from using any command, contact your AAA administrator for assistance.

A few other examples that describe the commands to list the task ID:

```
Router#describe show interfaces
The command is defined in show_interface.parser

show_interface.parser
User needs ALL of the following taskids:

   interface (READ)---------->

It will take the following actions:
Thu Mar 17 06:42:08.264 UTC
  Spawn the process:
    show_interface "-a"
Router#


Router(config)#describe ssh server
The command is defined in ssh.parser

ssh.parser
User needs ALL of the following taskids:

   crypto (READ WRITE) ---------->

It will take the following actions:
  Create/Set the configuration item:
       Path: gl/crypto/ssh/server/sshd/vrf/default
       Value: packed[ 0x1 <string> <string> ]
Router(config)#
```

For more details, see *Configuring AAA Services* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

## Task ID

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

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# taskgroup taskgroup1
RP/0/RP0RSP0/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 Global Configuration modeXR Config mode. To delete a task group, use the **no** form of this command.

**taskgroup** *taskgroup-name* [{**description** *string* | **task** {**read** | **write** | **execute** | **debug**} *taskid-name* | **inherit taskgroup** *taskgroup-name*}]
**no taskgroup** *taskgroup-name*

## Syntax Description

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

## Command Default

Five predefined user groups are available by default.

## Command Modes

Global Configuration modeXR Config mode

## Command History

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

## 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. For more details on task IDs, see the Usage Guidelines section of the **task** command.

You can use the **show user group** command in Global Configuration modeXR Config mode to know the group(s) that the current user is part of. Similarly, you can use the **show user all** to know the group or task information (such as username, groups, authentication method, task IDs, and so on) of the current user.

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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# taskgroup alpha
RP/0/RP0RSP0/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

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# line template alpha
RP/0/RP0RSP0/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

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

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

## Examples

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

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# aaa group server radius group1
RP/0/RSP0/CPU0:router(config-sg-radius)# server-private 10.1.1.1 auth-port 300
RP/0/RSP0/CPU0:router(config-sg-radius-private)# timeout 500
```

## Related Commands

| Command | Description |
|---|---|
| radius-server timeout, on page 62 | Sets the interval for which a router waits for a server host to reply before timing out. |
| radius-server retransmit, on page 61 | Specifies the number of times a RADIUS request is resent to a server if the server is not responding or is responding slowly. |
| server-private (RADIUS), on page 71 | Configures the IP address of the private RADIUS server for the group server. |

# 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

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

## Usage Guidelines

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

## Task ID

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router(config)# tacacs-server host 209.165.200.226
RP/0/RP0RSP0/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 Global Configuration modeXR 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.

**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

Global Configuration modeXR Config mode

## Command History

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# usergroup alpha
RP/0/RP0RSP0/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 Global Configuration modeXR Config mode. To delete a user from the database, use the **no** form of this command.

**username** *name* [{ **group** *name* | **policy** *name* | [ **password-policy** *name* ] { **password** | **masked-password** } [ *type* ] *password* | { **secret** | **masked-secret** } [{ *type* | **0** [ **enc-type** *type* ] *secret* }] }]
**no username** *name* [{ **group** *name* | **policy** | **password** | **masked-password** | **secret** | **masked-secret** | **password-policy** *name* [ **masked-password** [ *type* ] *password* ] }]

## Syntax Description

| | |
|---|---|
| *name* | Name of the user. The *name* argument can be only one word. Spaces and quotation marks are not allowed.<br>The allowed range for a user-defined username is 2-253 characters. |
| **group** *name* | Enables a user to be associated with a user group, as defined with the **usergroup** command. |
| **policy** *name* | Configures a password policy that is common to user password and secret. |
| **password-policy** *name* | (Optional) Specifies the password policy for cleartext and Type 7 password authentication. |
| **password** | Enables a password to be created for the specified user. |
| **masked-password** | Enables a password to be created for the specified user. When you key in the password, it is not visible on the screen. |

| | |
|---|---|
| *type password* | Specifies the password type and the password to be keyed in.<br><br>Enter 0 or 7 for the *type* argument. 0 specifies a cleartext password, and 7 specifies a Type 7 encrypted password.<br><br>If Type 7 encryption is enabled with the **password** keyword, the password is not visible to the user. The password can be up to 253 characters in length.<br><br>(Optional) *type* argument |
| **secret** | Enables a secret to be created for the specified user. |
| **masked-secret** | Enables a secret to be created for the specified user. When you key in the secret, it is not visible on the screen. |
| *type secret* | Specifies the secret type and the secret to be keyed in.<br><br>Enter 0, or enter 5, 8, 9, or 10, for the *type* argument. Details:<br>• 0 specifies a cleartext secret that will be encrypted for use.<br>• 5 specifies a Type 5 password that uses MD5 hashing algorithm.<br>• 8 specifies a Type 8 password that uses SHA256 hashing algorithm.<br>• 9 specifies a Type 9 password that uses scrypthashing algorithm.<br>• 10 specifies a Type 10 password that uses SHA512 hashing algorithm.<br><br>(Optional) *type* argument. |

| | |
|---|---|
| **0 enc-type** *type secret* | Specifies that you enter a cleartext secret to be encrypted by a specified encryption method.<br>• 0 specifies that you should enter a cleartext secret.<br>• **enc-type** specifies that you enter 5, 8, 9, or 10, for the *type* argument.<br>• Enter the cleartext secret for the *secret* argument.<br>(Optional) **enc-type** *type* keyword-argument combination. |

**Command Default**

No usernames are defined in the system.

**Command Modes**

Global Configuration modeXR Config mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |
| Release 7.2.1 | Added the support for **policy** option to configure policy common to user password and secret. |
| Release 7.3.1 | Password Masking feature options (**masked-password** and **masked-secret**) were added. When you key in a password or secret, it is not displayed on the screen. |

**Usage Guidelines**

**Note**

- 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 Global Configuration modeXR Config mode or username configuration submode. Permissions (task IDs) are assigned by associating the user with one or more defined user groups.

From Global Configuration modeXR 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.

The following are password masking guidelines for various command forms:

- **username** *name* **password** *type password*

  **username** *name* **masked-password** *type password*

  Enter 0 or 7 for the *type* argument. *0* specifies a cleartext password, and *7* specifies a Type 7 encrypted password.

- **secret** *type secret*

  **masked-secret** *type secret*

  Enter 0, or enter 5, 8, 9, or 10, for the *type* argument. 0 specifies a cleartext secret, and 5, 8, 9, and 10 specify a Type 5, Type 8, Type 9, and Type 10 secret, respectively.

- **secret 0 enc-type** *type secret*

  **masked-secret 0 enc-type** *type secret*

  Enter 5, 8, 9, or 10, for the *type* argument.

- **masked-password** *type password*

  **masked-secret** *type secret*

  After specifying the password encryption type, press **Enter** or **return** on your keyboard. The password/secret option appears in the next line. Example:

```
Router(config)# masked-secret 10

Enter secret:
Re-enter secret:
```

## Task ID

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

## Examples

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

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

| | |
|---|---|
| clear | Clear the uncommitted configuration |
| commit | Commit the configuration changes to running |
| describe | Describe a command without taking real actions |
| do | Run an exec command |
| exit | Exit from this submode |
| group | User group in which this user will be a member of |
| no | Negate a command or set its defaults |
| password | Specify the password for the user |
| pwd | Commands used to reach current submode |
| root | Exit to the Global Configuration modeXR Config mode |
| secret | Specify the secure password for the user |
| 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
```

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

```
Router#configure
Router(config)#username user1
Router(config-un)#policy test-policy1
Router(config-un)#secret 10
$6$dmwuW0Ajicf98W0.$y/vzynWF1/OcGxwBwHs79VAy5ZZLhoHd7TicR4mOo8IIVriYCGAKW0A.w1JvTPO7IbZry.DxHrE3SN2BBzBJe0
Router(config-un)#commit
```

The following example shows how to configure a Type 8 (SHA256) password for the user, *user8*. You can also see the examples and usage of the secret, on page 66 command.

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

```
Router#configure
Router(config)#username user8 secret 8
$8$ZYKGl1dZIw73Dl$IUWJOqTLoMyExhsNKoL5vMtvCOYguM5ajXf4uGeQj6I
Router(config-un)#commit
```

This example shows how to configure Type 9 password:

```
Router#configure
Router(config)#username user9 secret 9
$9$/rIQL1B3rplRBL$oS2fLWKFYH6B/kApxkkXmIqbPAHpRZkPEoh3WqGbvwQ
Router(config-un)#commit
```

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

```
Router#configure
Router(config)#username user10 secret 10
$6$9UvJidvsTEqgkAPU$3CL1Ei/F.E4v/Hi.UaqLwX8UsSEr9ApG6c5pzhMJmZtgW4jObAQ7meAwyhu5VM/aRFJqe/jxZG17h6xPrvJWf1
Router(config-un)#commit
```

### Password Masking Examples

The following example shows how to enable password masking for a cleartext password entry:

In this example, for user us3, a cleartext password is entered.

```
Router(config)# username us3 masked-password 0


Enter password:
Re-enter password:

Router(config)#commit
```

In the **show** command output, you can see the encrypted password:

```
Router# show run aaa
..

username us3
 password 7 105A1D0D
```

The encrypted password 105A1D0D is entered in the **Enter password:** and **Re-enter password:** fields, for Type 7 password encryption:

```
Router(config)# username us3 masked-password 7

Enter password:
Re-enter password:


Router(config)#commit
```

If there is a password mismatch between the two entries, an error message is displayed.

The following example shows how to enable password masking for a AAA password policy:

In this example, for user us6, a cleartext password is entered.

```
Router(config)# aaa password-policy security
Router(config)# username us6 password-policy security masked-password 0

Enter password:
```

```
Re-enter password:

Router(config)#commit
```

In the **show** command output, you can see the encrypted password.

```
Router# show run aaa
..

aaa password-policy security
..
username us6
 password-policy security password 7 0835585A
```

The encrypted password 0835585A is entered in the **Enter password:** and **Re-enter password:** fields for Type 7 password encryption.

```
Router(config)# username us6 password-policy test-policy masked-password 7

Enter password:
Re-enter password:


Router(config)#commit
```

# users group

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

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

**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**

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

**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**

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

**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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa authen login vty-authen line
RP/0/RP0RSP0/CPU0:router(config)# commit
RP/0/RP0RSP0/CPU0:router(config)# line template vty
RP/0/RP0RSP0/CPU0:router(config-line)# users group operator
RP/0/RP0RSP0/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

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

## 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

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server radius group1
RP/0/RP0RSP0/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

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

## Usage Guidelines

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

## Task ID

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

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# aaa group server tacacs+ myserver
RP/0/RP0RSP0/CPU0:router(config-sg-tacacs+)# server 9.27.10.6
RP/0/RP0RSP0/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 ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# 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

| | |
|---|---|
| *sa-id* | Identifier for the SA. IPSec supports from 1 to 64,500 sessions. |
| **all** | Deletes all IPSec SAs in the IPSec SADB. |
| **counters** | Clears the counters in the IPSec SADB. |
| **interface** | Clears the interfaces in the IPSec SADB. |
| **tunnel-ipsec** | The range of tunnel-ipsec is <0-4294967295>. |

## Command Default

No default behavior or values

## Command Modes

EXEC

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# clear crypto ipsec sa 100
```

## Related Commands

| Command | Description |
|---|---|
| show crypto ipsec sa, on page 138 | Displays the settings used by current SAs. |

# 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

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| interface | read, write |

## Examples

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

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface tunnel-ip 50000
RP/0/RSP0/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

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

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

```
RP/0/RP0RSP0/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 encrpt fail:0                     #pkts decrpt 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***

| Field | Description |
|---|---|
| SA id | Identifier for the SA. |
| interface | Identifier for the interface. |
| profile | String of alphanumeric characters that specify the name of a security profile. |
| local ident | IP address, mask, protocol, and port of the local peer. |
| remote ident | IP address, mask, protocol and port of the remote peer. |
| outbound esp sas | Outbound ESP SAs. |
| inbound esp sas | Inbound ESP SAs. |
| transform | The transform being used in the SA. |

| Field | Description |
|---|---|
| sa lifetime | The lifetime value used in the SA. |

The following sample output is from the **show crypto ipsec sa** command for the **profile** keyword for a profile named pn1:

```
RP/0/RP0RSP0/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/RP0RSP0/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.

**show crypto ipsec summary**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC

## Command History

| Release | Modification |
|---|---|
| Release 3.7.2 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

The following sample output is from the **show crypto ipsec summary** command:

```
RP/0/RP0RSP0/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
3600/100000000
```

This table describes the significant fields shown in the display.

***Table 10: show crypto ipsec summary Field Descriptions***

| Field | Description |
|---|---|
| SA | Identifier for the security association. |
| Node | Identifier for the node. |
| Local Peer | IP address of the local peer. |
| Remote Peer | IP address of the remote peer. |

| Field | Description |
|---|---|
| FVRF | 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 |
| Mode | Profile mode type. |
| Profile | Crypto profile in use. |
| Transform | Transform in use. |
| Lifetime | Lifetime value, displayed in seconds followed by kilobytes. |

# 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*]

**Syntax Description**

| | |
|---|---|
| *transform-set-name* | (Optional) IPSec transform set with the specified value for the *transform-set-name* argument are displayed. |

**Command Default**

No default values. The default behavior is to print all the available transform-sets.

**Command Modes**

EXEC

**Command History**

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

**Usage Guidelines**

If no transform is specified, all transforms are displayed.

**Task ID**

| Task ID | Operations |
|---|---|
| crypto | read |

**Examples**

The following sample output is from the **show crypto ipsec transform-set** command:

```
RP/0/RP0RSP0/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
```

# 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 the Cisco ASR 9000 Series Router* configuration module in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# 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.<br>The range for the number of days of the month is from 1 to 31.<br>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

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0RSP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0RSP0/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

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

## 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

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0RSP0/CPU0:router(config-isis-keys)# accept-tolerance infinite
```

# clear type6 client

To clear the Type 6 client state in case the primary key update process is stuck at any stage, use the **clear type6** command in EXEC modeXR EXEC mode.

**clear type6 client** { **keychain** | **snmp** }

## Syntax Description

| | |
|---|---|
| **keychain** | Clears the key chain client information. |
| **snmp** | Clears the snmp client information. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

You can track the primary key update operation using the **show type6 server** command output. If the *Master key Inprogress* field in that output displays as *YES*, then you can use **show type6 masterkey update status** command (or, **show type6 clients** command, prior to Cisco IOS XR Software Release 7.0.14) to check which client has not completed the operation. Accordingly, you can clear that particular client using this **clear** command.

## Task ID

| Task ID | Operation |
|---|---|
| system | read, write |

This example shows how to clear the Type 6 client state:

```
Router#clear type6 client keychain
```

## Related Commands

| Command | Description |
|---|---|
| show type6, on page 157 | Displays Type 6 password encryption information. |

# 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

| | |
|---|---|
| **HMAC-MD5** | Configures HMAC-MD5 as a cryptographic algorithm with a digest size of 16 bytes. |
| **HMAC-SHA1-12** | Configures HMAC-SHA1-12 as a cryptographic algorithm with a digest size of 12 bytes. |
| **HMAC-SHA1-20** | Configures HMAC-SHA1-20 as a cryptographic algorithm with a digest size of 20 bytes. |
| **MD5** | Configures MD5 as a cryptographic algorithm with a digest size of 16 bytes. |
| **SHA-1** | Configures SHA-1-20 as a cryptographic algorithm with a digest size of 20 bytes. |
| **HMAC-SHA-256** | Configures HMAC-SHA-256 as a cryptographic algorithm with a digest size of 32 bytes. |
| **HMAC-SHA1-96** | Configures HMAC-SHA1-96 as a cryptographic algorithm with a digest size of 12 bytes. |
| **AES-128-CMAC-96** | Configures AES-128-CMAC as a cryptographic algorithm with a digest size of 12 bytes. |

## Command Default

No default behavior or values

## Command Modes

Keychain-key configuration

## Command History

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

## 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.
- Intermediate System-to-Intermediate System (IS-IS) supports HMAC-MD5, SHA-1, MD5, AES-128-CMAC-96, HMAC-SHA-256, HMAC-SHA1-12, HMAC-SHA1-20, 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.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0RSP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0RSP0/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

Global Configuration modeXR Config mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0RSP0/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

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

## Usage Guidelines

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **key** command:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0RSP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0RSP0/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:<br>• Plain-text key strings—Minimum of 1 character and a maximum of 32.<br>• 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 | Modification |
|---|---|
| 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

50aefd

From Cisco IOS XR Software Release 7.2.1 and later, if you are using any **HMAC-SHA** algorithm for a session, then you must ensure that the configured *key-string* has a minimum length of 14 characters. Otherwise, the session goes down. This guideline is applicable only for FIPS mode.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **keystring** command:

```
RP/0/RP0RSP0/CPU0:router:# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0RSP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0RSP0/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.<br>The range for the number of days of the month to start is from 1 to 31.<br>The range for the years is from 1993 to 2035. |
| **duration** *duration value* | (Optional) Determines the lifetime of the key in seconds. |
| **infinite** | (Optional) Specifies that the key never expires once it becomes valid. |
| *end-time* | (Optional) Time, in *hh:mm:ss day month year* format, after which the key expires. The range is from 0:0:0 to 23:59:59 |

## Command Default

No default behavior or values

## Command Modes

Keychain-key configuration

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain isis-keys
RP/0/RP0RSP0/CPU0:router(config-isis-keys)# key 8
RP/0/RP0RSP0/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**

EXEC modeXR EXEC mode

**Command History**

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

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

| Task ID | Operations |
|---|---|
| system | read |

**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/RP0RSP0/CPU0:router# show key chain isis-keys

Key-chain: isis-keys/ -

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

# show type6

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

**show type6** { **clients** | **masterkey update status** | **server** | **trace server** { **all** | **error** | **info** } [ *trace-server-parameter* ] }

## Syntax Description

| | |
|---|---|
| **clients** | Displays Type 6 client information. |
| **masterkey update status** | Displays Type 6 primary key operation status. |
| **server** | Displays Type 6 server information. |
| **trace server** | Displays Type 6 trace server information. |
| **all** | Displays all Type 6 traces. |
| **error** | Displays Type 6 error traces. |
| **info** | Displays Type 6 information trace entries. |
| *trace-server-parameter* | (Optional) Displays Type 6 trace server information for the specified parameter. Use one from the list of parameters defined in the Usage Guidelines section. |

## Command Default

None.

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |
| Release 7.0.14 | This command was modified to include the **masterkey update status** option. |

## Usage Guidelines

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

The **show type6 clients** command is deprecated with the introduction of **masterkey update status**.

| Trace Server Parameter | Displayed Trace Server Information |
|---|---|
| **file** | The specified file. |
| **hexdump** | Hexadecimal format. |
| **last** | The most recent entries. |
| **location** | Line card location. |

| Trace Server Parameter | Displayed Trace Server Information |
|---|---|
| **reverse** | From the most recent entry to the first entry. |
| **stats** | Statistics information. |
| **tailf** | New traces as they are added. |
| **udir** | Copies trace information from remote locations to the specifed temporary directory. |
| **unique** | Unique entries with counts. |
| **usec** | User security information, with time stamp. |
| **verbose** | Internal debugging information. |
| **wide** | Removes buffer name, node name, and tid information. |
| **wrapping** | Wrapping entries. |

## Examples

The following command displays Type 6 password encryption feature information:

```
Router# show type6 server

Server detail information:
============================
AES config State : Enabled
Masterkey config State : Enabled
Type6 feature State : Enabled
Master key Inprogress : No
```

```
Router# show type6 trace server all

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

```
Router# show type6 clients

Type6 Clients information:

Client Name   MK State
====================
keychain      UNKNOWN
```

This example shows a sample output of the **masterkey update status** command:

```
Router#show type6 masterkey update status
Thu Sep 17 06:50:07.980 UTC
Type6 masterkey operation is inprogress

Masterkey upate status information:
Client Name                Status
====================================
keychain                   INPROGRESS
```

# 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 ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# allow (macsec)

To specify MACsec policy exception to allow packets in clear text, use **allow** command under MACsec policy configuration mode. To remove this configuration, use the **no** form of this command.

**allow** { **lacp-in-clear** | **pause-frames-in-clear** | **lldp-in-clear** }

## Syntax Description

| | |
|---|---|
| **lacp-in-clear** | Allows Link Aggregation Control Plane protocol (LACP) packets in clear text. |
| **pause-frames-in-clear** | Allows Ethernet PAUSE frame packets in clear text. |
| **lldp-in-clear** | Allows Link Layer Discovery Protocol (LLDP) packets in clear text. |

## Command Default

None

## Command Modes

MACsec policy configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 7.11.1 | This command was modified to include the **lldp-in-clear** option. |
| Release 7.3.15 | This command was modified to include the **pause-frames-in-clear** option. |
| Release 7.3.1 | This command was introduced. |

## Usage Guidelines

The **policy-exception lacp-in-clear** command under MACsec policy configuration mode is deprecated. Hence, it is recommended to use the **allow lacp-in-clear** command instead, to allow LACP packets in clear-text format.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

This example shows how to create a MACsec policy exception to allow LACP, LLDP, and Ethernet PAUSE frame packets in clear text:

```
Router#configure
Router(config)#macsec-policy test-macsec-policy
Router(config-macsec-policy)#allow lacp-in-clear
Router(config-macsec-policy)#allow pause-frames-in-clear
Router(config-macsec-policy)#allow lldp-in-clear
Router(config-macsec-policy)#commit
```

# 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:<br>• GCM-AES-128<br>• GCM-AES-256<br>• GCM-AES-XPN-128<br>• 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

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **cipher-suite** command:

```
RP/0/RP0RSP0/CPU0:router# configure t
RP/0/RP0RSP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# cipher-suite GCM-AES-XPN-256
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#commit
```

# conf-offset

Configures 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:<br>• CONF-OFFSET-0 : Does not offset the encryption.<br>• CONF-OFFSET-30: Offsets the encryption by 30 bytes.<br>• CONF-OFFSET-50: Offsets the encryption by 50 bytes. |

## Command Default

Default value is 0.

## Command Modes

MACsec policy configuration.

## Command History

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure t
RP/0/RP0RSP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# conf-offset CONF-OFFSET-30
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

# crypto-sks-kme

To display details of the Quantum Key Distribution (QKD) server, use the **crypto-sks-kme** command in EXEC mode.

**crypto-sks-kme** *profile-name* { **entropy** | **capability** }

## Syntax Description

| | |
|---|---|
| *profile-name* | Specifies the key string in clear-text form. |
| **entropy** | Specifies the key in encrypted form. |
| **capability** | Specifies the key in Type 6 encrypted form. |

## Command Default

None

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.9.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following examples shows how to use the **crypto-sks-kme** command:

```
Router# crypto sks kme remote_qkd_prof1 entropy
Entropy Details:
Key details Dump: 0000 - 406b004c9c7f000000000000000000280c71794fa6f029d0ee2f6c4cd01b46
Key : 406b004c9c7f000000000000000000280c71794fa6f029d0ee2f6c4cd01b46
Entropy Length: 32
```

```
Router# crypto sks kme QkdIP capability
Capability Details:
  Entropy supported : False
  Key supported     : False
  Algorithm         : QKD
  Local identifier  : Alice1
  Remote identifier : Alice1, Bob1,
```

# enable-legacy-fallback

To enable interoperability with peer devices that do not support MACsec active fallback feature, use the **enable-legacy-fallback** command in MACsec policy configuration mode. To remove the configuration, use the **no** form of this command.

**enable-legacy-fallback**

## Syntax Description

This command has no keywords or arguments.

## Command Default

Disabled, by default.

## Command Modes

MACsec policy configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.14 | This command was introduced. |

## Usage Guidelines

For more details on MACsec active fallback feature, see the *Fallback PSK* section in the *Configuring MACsec Encryption* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

## Task ID

| Task ID | Operation |
|---|---|
| system | read, write |

This example shows how to enable interoperability with peer devices that do not support MACsec active fallback feature:

```
Router#configure
Router(config)#macsec-policy P1
Router(config-macsec-policy)#enable-legacy-fallback
Router(config-macsec-policy)#commit
```

# 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 Global Configuration modeXR 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. |
| *location* | Specifies the node location to enable POST KAT. |
| **all** | Enables POST KAT for all nodes. |

## Command Default

Disabled by default

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.14 | This command was introduced. |

## 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

| Task ID | Operation |
|---|---|
| system | read, write |

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
```

## Related Commands

| Command | Description |
|---|---|
| show hw-module macsec-fips-post, on page 184 | Displays the power-on self-test (POST) known answer test (KAT) configurations of nodes in a router. |

# 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 Global Configuration modeXR Config mode mode. To remove this configuration, use the no form of this command.

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

## Syntax Description

| | |
|---|---|
| **location** | Specifies the node location to enable the MACsec mode. |
| **all** | Enables MACsec mode for all nodes. |
| *location* | Enables MACsec mode for a specific node. |

## Command Default

Disabled by default

## Command Modes

Global Configuration modeXR Config mode

## Command History

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

## 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

| Task ID | Operation |
|---|---|
| system | read, write |

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
```

## Related Commands

| Command | Description |
| --- | --- |
| show hw-module macsec-mode, on page 186 | Displays the MACsec mode of a line card and the user action to be performed. |

# 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 | Modification |
|---|---|
| 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 ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **key** command:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router# key chain mac_chain macsec
RP/0/RP0RSP0/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

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **key-server-priority** command:

```
RP/0/RP0RSP0/CPU0:router# configure t
RP/0/RP0RSP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# key-server-priority 16
RP/0/RP0RSP0/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 | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |

**Task ID**

| Task ID | Operations |
|---|---|
| system | read, write |

**Examples**

The following example shows how you can configure a key chain for MACsec encryption:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0RSP0/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.

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

## Syntax Description

| | |
|---|---|
| **clear** | Specifies the key string in clear-text form. |
| **password**password | Specifies the key in encrypted form. |
| **password6** | Specifies the key in Type 6 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:<br>• Plain-text key strings—Minimum of 1 character and a maximum of 32 (128-bit encryption)/64 (256-bit encryption) characters (hexadecimal string).<br>• Encrypted key strings—Minimum of 4 characters and no maximum. |

## Command Default

The default value is clear.

## Command Modes

Key configuration submode under the macsec key chain mode

## Command History

| Release | Modification |
|---|---|
| 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

50aefd

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **key-string** command:

```
! For AES 128-bit encryption

RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
12345678123456781234567812345678 cryptographic-algorithm AES-128-CMAC

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

RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string clear
1234567812345678123456781234567812345678123456781234567812345678 cryptographic-algorithm
AES-256-CMACRP/0/RP0RSP0/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.<br>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

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **lifetime** command:

```
! For AES 128-bit encryption
```

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
12345678123456781234567812345678 cryptographic-algorithm AES-128-CMAC
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february
 2020 12:00:00 30 september 2020
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)#commit

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

RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
1234567812345678123456781234567812345678123456781234567812345678 cryptographic-algorithm
AES-256-CMAC
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february
 2020 duration 2592000
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# commit

! Lifetime specified as infinite:

RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# key chain mac_chain macsec
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec)# key 1234abcd5678
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# key-string
1234567812345678123456781234567812345678123456781234567812345678 cryptographic-algorithm
AES-256-CMAC
RP/0/RP0RSP0/CPU0:router(config-mac_chain-MacSec-1234abcd5678)# lifetime 05:00:00 20 february
 2020 infinite
RP/0/RP0RSP0/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.

**macsec-policy** *policy-name* [{ **allow** { **lacp-in-clear** | **pause-frames-in-clear** } | **cipher-suite** { **GCM-AES-128** | **GCM-AES-256** | **GCM-AES-XPN-128** | **GCM-AES-XPN-256** } | **conf-offset** { **CONF-OFFSET-0** | **CONF-OFFSET-30** | **CONF-OFFSET-50** } | **delay-protection** | **enable-legacy-fallback** | **include-icv-indicator** | **key-server-priority** *priority-value* | **policy-exception lacp-in-clear** | **sak-rekey-interval** { *value-in-minutes* | **seconds** *value-in-seconds* } | **security-policy** { **must-secure** | **should-secure** } | **use-eapol-pae-in-icv** | **vlan-tags-in-clear** *value* | **window-size** *window-size* }]

## Syntax Description

| | |
|---|---|
| *policy_name* | The MACsec policy name with a maximum length of 16. |
| **allow** | Specifies MACsec policy exception to allow packets in clear text. |
| **pause-frames-in-clear** | Allows Ethernet PAUSE frame packets in clear text. |
| **cipher-suite** | Specifies the cipher-suite used for encryption. |
| **conf-offset** | Specifies the confidentiality offset value for encryption. |
| **delay-protection** | Enables data delay protection. |
| **include-icv-indicator** | Includes integrity check value (ICV) indicator parameter set in MACsec Key Agreement PDU (MKPDU). |
| **enable-legacy-fallback** | Enables interoperability with peer devices that do not support MACsec active fallback feature. |
| **key-server-priority** | Specifies the key-server priority for the node. |
| **policy-exception** | Specifies MACsec policy exception to allow packets in clear text. |
| **lacp-in-clear** | Allows Link Aggregation Control Plane protocol (LACP) packets in clear text. |
| **sak-rekey-interval** | Specifies the interval after which the key-server generates a new Secure Association Key (SAK) for a secured session. |
| **security-policy** | Specifies the security policy as must secure or should secure for data encryption. |
| **use-eapol-pae-in-icv** | Enables the use of Extensible Authentication Protocol over LAN (EAPoL) port access entity (POE) address in ICV. |
| **vlan-tags-in-clear** | Specifies the number of vlan-tags in clear (1 or 2). |
| **window-size** | Specifies the window-size used for encryption.<br>The range is from 0 to 1024. |

## Command Default

No default behavior or values.

**Command Modes**

Global configuration

**Command History**

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |
| Release 7.0.14 | This command was modified to include the **enable-legacy-fallback** option. |
| Release 7.3.1 | This command was modified to include the **allow** keyword. |
| Release 7.3.15 | This command was modified to include the **pause-frames-in-clear** option under the **allow** keyword. |

**Usage Guidelines**

The **policy-exception lacp-in-clear** command is deprecated. Hence, it is recommended to use the **allow lacp-in-clear** command instead, to allow LACP packets in clear-text format.

**Task ID**

| Task ID | Operations |
|---|---|
| system | read, write |

**Examples**

This example shows how to configure the **macsec-policy** command:

```
RP/0/RP0RSP0/CPU0:router# configure t
RP/0/RP0RSP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#cipher-suite GCM-AES-128
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#conf-offset CONF-OFFSET-30
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#delay-protection
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# include-icv-indicator
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# key-server-priority 10
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# policy-exception lacp-in-clear
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# sak-rekey-interval seconds 86400
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# security-policy must-secure
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# use-eapol-pae-in-icv
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# vlan-tags-in-clear 1
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

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

```
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# window-size 256
RP/0/RP0RSP0/CPU0:router(config-mac_policy)#
```

This example shows how to create a MACsec policy exception to allow LACP and Ethernet PAUSE frame packets in clear text:

```
Router#configure
Router(config)#macsec-policy test-macsec-policy
Router(config-macsec-policy)#allow lacp-in-clear
Router(config-macsec-policy)#allow pause-frames-in-clear
Router(config-macsec-policy)#commit
```

# 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**

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

**Usage Guidelines**

Enabling the **macsec shutdown** command, brings down all macsec sessions on the MACsec-enabled interfaces and resets ports to non-macsec mode. The already existing MACsec configurations remain unaffected by enabling this feature.

Disabling the **macsec shutdown** command, brings up MACsec sessions for the configured interfaces and enforces MACsec policy on the port.

**Warning** Configuring **macsec shutdown** command disables MACsec on all data ports, system wide. Execute **clear** command to erase cached configuration or **commit** command to continue.

**Task ID**

| Task ID | Operation |
|---|---|
| system | read, write |

**Example**

The following example shows how to enable MACsec shutdown:

```
RP/0/RSP0/CPU0:router# configure terminal
RP/0/RSP0/CPU0:router(config)# macsec shutdown
```

# sak-rekey-interval

To set a timer value to rekey the MACsec secure association key (SAK) at a specified interval, use the **sak-rekey-interval** command in the macsec-policy configuration mode. To disable this feature, use the **no** form of this command.

**sak-rekey-interval** *timer-value*

## Syntax Description

| | |
|---|---|
| *timer-value* | Specifies the timer value, in seconds.<br>Range is 60 to 2592000. |

## Command Default

The timer is set to OFF, by default

## Command Modes

MACsec policy configuration mode.

## Command History

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

This example shows how to set a timer value to rekey the MACsec SAK:

```
Router#configure
Router(config)#macsec-policy test-policy
Router(config-macsec-policy)#sak-rekey-interval 120
Router(config-macsec-policy)#commit
```

# 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 EXEC modeXR 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

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.14 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| system | read |

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
```

```
0/11/CPU0       YES             YES               NONE
```

## Related Commands

| Command | Description |
|---|---|
| hw-module macsec-fips-post, on page 168 | Enables power-on self-test known answer test (KAT) for the physical layer transceiver (PHY) of a line card |

# 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 EXEC modeXR 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

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |
| Release 7.0.14 | This command was modified to include the **all** option. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| 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            NO               RELOAD
```

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

Location       Configured     Running          Action
-------------------------------------------------------------
0/1/CPU0       YES            YES              NONE
```

## Related Commands

| Command | Description |
|---|---|
| hw-module macsec-mode, on page 170 | Enables the MACsec mode for the physical layer transceiver (PHY) of a line card. |

# show crypto sks profile

To display the details or statistics of the Session Key Service (SKS) profiles in the router, use the **show crypto sks profile** command in the EXEC mode.

**show crypto sks profile** { *profile-name* | **all** } [ **stats** ]

## Syntax Description

| | |
|---|---|
| *profile name* | Specifies the name of the SKS profile. |
| **all** | Specifies all the SKS profiles in the router. |
| **stats** | Displays the statistics of the SKS profiles. |

## Command Default

None

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.9.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| system | read |

The following example shows how to view the SKS profile details in a router:

```
Router# show crypto sks profile all
Profile Name         :ProfileR1toR2
Myidentifier        :Router1
Type                :Remote
Reg Client Count      :1

Server
IP                   :192.0.2.35
Port                 :10001
Vrf                  :Notconfigured
Source Interface     :Notconfigured
Status               :Connected
Entropy              :true
Key                  :true
Algorithm            :QKD
Local identifier     :Alice
Remote identifier    :Alice

Peerlist
QKD ID                :Bob
State                 :Connected
```

```
Peerlist
QKD ID                :Alice
State                 :Connected
```

The following example shows how to view the SKS profile statistics in a router:

```
Router# show crypto sks profile all stats
Profile Name            : ProfileR1toR2
My identifier           : Router1
Server
 IP                     : 192.0.2.35
 Port                   : 10001
 Status                 : connected
Counters
 Capability request         : 1
 Key request                : 3
 Key-id request             : 0
 Entropy request            : 0
 Capability response        : 1
 Key response               : 3
 Key-id response            : 0
 Entropy response           : 0
 Total request              : 4
 Request failed             : 0
 Request success            : 4
 Total response             : 4
 Response failed            : 0
 Response success           : 4
 Retry count                : 0
 Response Ignored           : 0
 Cancelled count            : 0
Response time
 Max Time                   : 100 ms
 Avg Time                   : 10  ms
 Min Time                   : 50  ms
Last transaction
 Transaction Id             : 9
 Transaction type           : Get key
 Transaction status         : Response data received, successfully
 Http code                  : 200 OK (200)
```

# show macsec mka summary

To display the Summary of MACsec Sessions, use the **show macsec mka summary** command in EXEC mode.

**show macsec mka summary**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

No default behavior or values.

**Command Modes**

EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

**Usage Guidelines**

The **show macsec mka summary** command is available only with the installation of the k9sec rpm.

**Task ID**

| Task ID | Operation |
|---|---|
| interface | read |

This example shows how to view MACsec mka summary information for a specific interface.

```
 Router# show macsec mka summary
Fri Dec 15 06:41:13.299 UTC

NODE: node0_RP0_CPU0
================================================================================================
   Interface-Name     Status     Cipher-Suite       KeyChain       PSK/EAP       CKN
================================================================================================
     TF0/0/0/24       Secured  GCM-AES-XPN-256         kc1          PRIMARY     1111
     TF0/0/0/25       Secured  GCM-AES-XPN-256         kc1          PRIMARY     1111
     TF0/0/0/26       Secured  GCM-AES-XPN-256         kc1          PRIMARY     1111
     TF0/0/0/27       Secured  GCM-AES-XPN-256         kc1          PRIMARY     1111

Total MACSec Sessions  : 4
    Secured Sessions   : 4
    Pending Sessions   : 0
    Suspended Sessions : 0
    Active  Sessions   : 0
```

# show macsec mka session

To display the detailed Information of MACsec Sessions, use the **show macsec mka session** command in EXEC mode.

**show macsec mka session interface** *interface name* **location** *location name* **detail**

## Syntax Description

| | |
|---|---|
| **interface** | Specifies the interface name to view MACsec details. |
| *interface name* | Enables MACsec mode for a specified interface. |
| **location** | Specifies the node location to enable the MACsec details. |
| *location name* | Enables MACsec mode for a specific node. |
| **detail** | (Optional) Detailed information specific to session. |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec mka session** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows how to view MACsec mka session information for a specific interface.

```
 Router# show macsec mka session
Fri Dec 15 06:31:38.457 UTC

NODE: node0_RP0_CPU0
==============================================================================================

   Interface-Name        Local-TxSCI       #Peers   Status   Key-Server   PSK/EAP      CKN
==============================================================================================

     TF0/0/0/24      ac3a.67ee.281c/0001     1      Secured      YES      PRIMARY     1111
     TF0/0/0/25      ac3a.67ee.281d/0001     1      Secured      YES      PRIMARY     1111
     TF0/0/0/26      ac3a.67ee.281e/0001     1      Secured      YES      PRIMARY     1111
     TF0/0/0/27      ac3a.67ee.281f/0001     1      Secured      YES      PRIMARY     1111
```

# show macsec mka interface detail

To display detailed information on MACsec interfaces, use the **show macsec mka interface detail** command in the EXEC modeXR EXEC mode.

**show macsec mka interface** *interface name* **detail**

## Syntax Description

| | |
|---|---|
| *interface name* | Specifies the name of the interface for which you want to view the MACsec details. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.11.1 | The **lldp-in-clear** counter was introduced. |
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec mka interface detail** command is available only with the installation of the k9sec rpm.

The **show macsec mka interface detail** command displays information about all MACsec-enabled interfaces across all nodes. If you need MACsec information for a specific interface, use the **show macsec mka interface** *interface name* **detail** command.

## Task ID

| Task ID | Operation |
|---|---|
| system | read |

This example shows how to view the MACsec information for a specific interface:

```
 Router# show macsec  mka interface HundredGigE 0/0/0/29 detail
Interface Name : HundredGigE0/0/0/29
    Interface Namestring     : HundredGigE0/0/0/29
    Interface short name     : Hu0/0/0/29
    Interface handle         : 0x3c000008
    Interface number         : 0x3c000008
    MacSecControlledIfh      : 0x3c008110
    MacSecUnControlledIfh    : 0x3c008118
    Interface MAC            : ac4a.6730.0624
    Ethertype                : 888E
    EAPoL Destination Addr   : 0180.c200.0003
    MACsec Shutdown          : FALSE
    Config Received          : TRUE
    IM notify Complete       : TRUE
    MACsec Power Status      : N/A
    Interface CAPS Add       : TRUE
    RxSA CAPS Add            : TRUE
    TxSA CAPS Add            : TRUE
    lldp-in-clear            : TRUE
```

```
Principal Actor          : Primary
MKA PSK Info
  Key Chain Name         : kc
  MKA Cipher Suite       : AES-256-CMAC
  CKN                    : 12 34
MKA fallback_PSK Info
  fallback keychain Name : fb
  MKA Cipher Suite       : AES-256-CMAC
  CKN                    : 99 99
Policy                   : mp
SKS Profile              : N/A
Traffic Status           : Protected
Rx SC 1
  Rx SCI                 : ac3a67ee28240001
  Rx SSCI                : 2
  Peer MAC               : ac:3a:67:ee:28:24
  Is XPN                 : YES
  SC State               : Provisioned
  SAK State[3]           : Provisioned
  Rx SA Program Req[3]   : 2023 Nov 08 10:45:16.000
  Rx SA Program Rsp[3]   : 2023 Nov 08 10:45:16.054
  SAK Data
    SAK[3]               : ***
    SAK Len              : 32
    SAK Version          : 1861
    HashKey[3]           : ***
    HashKey Len          : 16
    Conf offset          : 0
    Cipher Suite         : GCM-AES-XPN-256
    CtxSalt[3]           : 0e 43 04 9b 46 92 b2 5a 56 95 c2 af
    CtxSalt Len          : 12
    ssci                 : 2

Tx SC
  Tx SCI                 : ac4a673006240001
  Tx SSCI                : 1
  Active AN              : 3
  Old AN                 : 2
  Is XPN                 : YES
  Next PN                : 1, 1, 1, 1
  SC State               : Provisioned
  SAK State[3]           : Provisioned
  Tx SA Program Req[3]   : 2023 Nov 08 10:45:16.104
  Tx SA Program Rsp[3]   : 2023 Nov 08 10:45:16.154
  SAK Data
    SAK[3]               : ***
    SAK Len              : 32
    SAK Version          : 1861
    HashKey[3]           : ***
    HashKey Len          : 16
    Conf offset          : 0
    Cipher Suite         : GCM-AES-XPN-256
    CtxSalt[3]           : 0e 43 04 98 46 92 b2 5a 56 95 c2 af
    CtxSalt Len          : 12
    ssci                 : 1
```

# show macsec mka statistics

To display MKA interface and session statistics, use the **show macsec mka statistics** command in EXEC mode.

**show macsec mka statistics** [ **interface** *interface name* | **location** *location name* ]

## Syntax Description

| | |
|---|---|
| **interface** | Specifies the interface name to view MACsec details. |
| *interface name* | Enables MACsec mode for a specified interface. |
| **location** *location name* | (Optional) Location of the node to view global statistics of the MKA instance. |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec mka statistics** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows the output for **show macsec mka statistics**:

```
 Router# show macsec mka statistics location 0/RP0/CPU0
Fri Dec 15 06:43:21.985 UTC


MKA Global Statistics
=====================
MKA Session Totals
   Secured..................... 10
   Reauthentication Attempts.. 0

   Deleted (Secured).......... 6
   Keepalive Timeouts......... 0

CA Statistics
   Pairwise CAKs Derived...... 0
   Pairwise CAK Rekeys........ 0
   Group CAKs Generated....... 0
   Group CAKs Received........ 0

SA Statistics
   SAKs Generated............. 10
   SAKs Rekeyed............... 0
   SAKs Received.............. 0
```

```
   SAK Responses Received..... 10
   PPK Tuple Generated........ 0
   PPK Retrieved.............. 0

MKPDU Statistics
   MKPDUs Validated & Rx...... 480156
      "Distributed SAK"..... 0
      "Distributed CAK"..... 0
      "Distributed PPK"..... 0
      "PPK Capable"......... 0
   MKPDUs Transmitted......... 480167
      "Distributed SAK"..... 10
      "Distributed CAK"..... 0
      "Distributed PPK"..... 0
      "PPK Capable"......... 0
```

# show macsec mka client

To display MACsec MKA client traces, use the **show macsec mka client** command in EXEC mode.

**show macsec mka client [trace {all | errors | events | info}]**

## Syntax Description

| | |
|---|---|
| **all** | (Optional) Show all MACsec MKA client traces for the specified node, or the current node if none is specified. |
| **errors** | (Optional) Show MACsec MKA client error traces for the specified node, or the current node if none is specified. |
| **events** | (Optional) Show MACsec MKA client event traces for the specified node, or the current node if none is specified |
| **info** | (Optional) Show MACsec MKA client info traces for the specified node, or the current node if none is specified |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec mka trace** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows the output for **show macsec mka client trace all**:

```
 Router# show macsec mka client trace all
Tue Dec  5 10:32:14.266 UTC
1 wrapping entries (10432 possible, 192 allocated, 0 filtered, 1 total)
Dec  4 09:56:25.544 macsec_mka/client/events 0/RP0/CPU0 t5544 TP257:aipc, server:driver,
client:default, init from pid:4779
```

# show macsec mka standby

To display MACsec MKA information from hot standby node, use the **show macsec mka standby** command in EXEC mode.

**show macsec mka standby [interface | session | statistics]** { *interface name* **detail** } **[summary]**

## Syntax Description

| | |
|---|---|
| **interface** | Specifies the interface name to view MACsec details. |
| *interface name* | Enables MACsec mode for a specified interface. |
| **detail** | (Optional) detailed information specific to Interface/Session |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec mka standby** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows the output for **show macsec mka standby summary**:

```
 Router# show macsec mka standby summary
Tue Dec  5 10:38:29.004 UTC

Total MACSec Sessions  : 0
    Secured Sessions   : 0
    Pending Sessions   : 0
    Suspended Sessions : 0
     Active  Sessions : 0
```

# show macsec mka trace

To display MACsec MKA traces, use the **show macsec mka trace** command in EXEC mode.

**show macsec mka trace [all | base | config | errors | events | new-errors | new-events ]**

## Syntax Description

| | |
|---|---|
| **all** | (Optional) Show all MACsec MKA traces for the specified node, or the current node if none is specified. |
| **base** | (Optional) Show MACsec MKA base traces for the specified node, or the current node if none is specified. |
| **config** | (Optional) Show MACsec MKA config traces for the specified node, or the current node if none is specified. |
| **errors** | (Optional) Show MACsec MKA error traces for the specified node, or the current node if none is specified. |
| **events** | (Optional) Show MACsec MKA event traces for the specified node, or the current node if none is specified. |
| **new-errors** | (Optional) Show MACsec MKA new-errors traces for the specified node, or the current node if none is specified. |
| **new-events** | (Optional) Show MACsec MKA new-event traces for the specified node, or the current node if none is specified. |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec mka trace** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows the output for **show macsec mka trace all**:

```
 Router# show macsec mka trace all
Fri Dec 15 06:42:04.919 UTC
2385 wrapping entries (8576 possible, 3968 allocated, 0 filtered, 2385 total)
Dec 12 15:12:30.077 macsec_mka/base 0/RP0/CPU0 t10778 TP1002: ********** MacSec MKA(10778)
 init start ********.
Dec 12 15:12:30.077 macsec_mka/new_events 0/RP0/CPU0 t10778 TP1002: ********** MacSec
MKA(10778) init start ********.
```

```
Dec 12 15:12:30.077 macsec_mka/events 0/RP0/CPU0 t10778 TP18: MKA_EVENT: Successfully created
 mka event queue
Dec 12 15:12:30.077 macsec_mka/base 0/RP0/CPU0 t10778 TP10: Timer init Success
Dec 12 15:12:30.077 macsec_mka/base 0/RP0/CPU0 t10778 TP801: process respawn_count:1
Dec 12 15:12:30.080 macsec_mka/base 0/RP0/CPU0 t10778 TP164: platform_capa : macsec:1,
macsec-service:0, macsec-subif:0, if_capa:1, ddp:1, secy_intf:1
Dec 12 15:12:30.080 macsec_mka/base 0/RP0/CPU0 t10778 TP164: platform_capa : ea_ha:0,
driver_ha:1, ea_retry:1, plt_sci:0, persist:0, max_an:3, no_secure_loc:1
Dec 12 15:12:30.080 macsec_mka/base 0/RP0/CPU0 t10778 TP164: platform_capa : issu:0,
ppk_support:1, pl_if_data:0, power_status:0, hot_stdby:0
Dec 12 15:12:30.080 macsec_mka/base 0/RP0/CPU0 t10778 TP1341: HA role: Active
```

# show macsec policy detail

To display details on the MACsec policies configured on the router, use the **show macsec policy detail** command in the EXEC modeXR EXEC mode.

**show macsec policy** *policy name* **detail**

## Syntax Description

| | |
|---|---|
| *policy name* | Specifies the name of the MACsec policy that you want to view. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.11.1 | The **lldp-in-clear** counter was introduced. |
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec policy detail** command is available only with the installation of the k9sec rpm.

The **show macsec policy detail** command displays information about all MACsec policies in the router. If you need details of a specific, use the **show macsec policy** *policy name* **detail** command.

## Task ID

| Task ID | Operation |
|---|---|
| system | read |

This example shows the output for **show macsec policy** *policy name* **detail**:

```
Router# show macsec policy mp detail
Policy Name                 : mp
      Cipher Suite          : GCM-AES-XPN-256
      Key-Server Priority   : 16
      Window Size           : 64
      Conf Offset           : 0
      Replay Protection     : TRUE
      Delay Protection      : FALSE
      Security Policy       : Must Secure
      Vlan Tags In Clear    : 1
      LACP In Clear         : FALSE
      LLDP In Clear         : FALSE
      Pause Frame In Clear  : FALSE
      Sak Rekey Interval    : 60 seconds
      Include ICV Indicator : FALSE
      Use Eapol PAE in ICV  : FALSE
      Disable Suspend On Request    : FALSE
      Disable Suspend For           : FALSE
      Enable legacy fallback        : FALSE
      SKS Profile                   : N/A
```

```
      Max AN                        : 3
```

This example shows the output for **show macsec policy detail**:

```
Router# show macsec policy detail

Total Number of Policies = 2
-----------------------------------------------------------
Policy Name                 : DEFAULT-POLICY
      Cipher Suite          : GCM-AES-XPN-256
      Key-Server Priority   : 16
      Window Size           : 64
      Conf Offset           : 0
      Replay Protection     : TRUE
      Delay Protection      : FALSE
      Security Policy       : Must Secure
      Vlan Tags In Clear    : 1
      LACP In Clear         : FALSE
      LLDP In Clear         : FALSE
      Pause Frame In Clear  : FALSE
      Sak Rekey Interval    : OFF
      Include ICV Indicator : FALSE
      Use Eapol PAE in ICV  : FALSE
      Disable Suspend On Request    : FALSE
      Disable Suspend For           : FALSE
      Enable legacy fallback        : FALSE
      SKS Profile                   : N/A
      Max AN                        : 3

Policy Name                 : mp
      Cipher Suite          : GCM-AES-XPN-256
      Key-Server Priority   : 16
      Window Size           : 64
      Conf Offset           : 0
      Replay Protection     : TRUE
      Delay Protection      : FALSE
      Security Policy       : Must Secure
      Vlan Tags In Clear    : 1
      LACP In Clear         : FALSE
      LLDP In Clear         : FALSE
      Pause Frame In Clear  : FALSE
      Sak Rekey Interval    : 60 seconds
      Include ICV Indicator : FALSE
      Use Eapol PAE in ICV  : FALSE
      Disable Suspend On Request    : FALSE
      Disable Suspend For           : FALSE
      Enable legacy fallback        : FALSE
      SKS Profile                   : N/A
      Max AN                        : 3
```

# show macsec secy

To display Interface based MACsec dataplane (SecY) statistics, use the **show macsec secy** command in EXEC mode.

**show macsec secy** [ **stats** { **interface** *interface name* **sc** } ]

## Syntax Description

| | |
|---|---|
| *interface name* | MACsec enabled Interface to be specified.. |
| **sc** | (Optional) Display Secure Channel Statistics for both Rx-SC,SA and Tx-SC,SA specific to the given interface |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec secy** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows the output for **show macsec secy**:

```
 Router# show macsec mka secy stats interface HundredGigE 0/0/0/29 sc
Interface Stats
    InPktsUntagged     :  0
    InPktsNoTag        :  0
    InPktsBadTag       :  0
    InPktsUnknownSCI   :  0
    InPktsNoSCI        :  0
    InPktsOverrun      :  0
    InOctetsValidated  :  0
    InOctetsDecrypted  :  3510182
    OutPktsUntagged    :  0
    OutPktsTooLong     :  0
    OutOctetsProtected :  0
    OutOctetsEncrypted :  1827580
```

# show macsec ea

To display MACsec programming details for each interface, use the **show macsec ea** command in EXEC mode.

**show macsec ea [ idb** { **interface** *interface name* | | **location** *location name* } **| trace {all | errors | events| base}**

## Syntax Description

| | |
|---|---|
| **interface** | Specifies the interface name to view MACsec details. |
| *interface name* | Enables MACsec mode for a specified interface. |
| **location** | Specifies the node location to enable the MACsec details. |
| *location name* | Enables MACsec mode for a specific node. |
| **all** | (Optional) Show **all** MACsec EA traces for the specified node, or the current node if none is specified. |
| **base** | (Optional) Show MACsec EA **base** traces for the specified node, or the current node if none is specified. |
| **errors** | (Optional) Show MACsec EA **error** traces for the specified node, or the current node if none is specified. |
| **events** | (Optional) Show MACsec EA **event** traces for the specified node, or the current node if none is specified. |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec ea** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows how to view MACsec information for a specific interface located on location 0/RP0/CPU0.

```
 Router# show macsec ea idb location 0/RP0/CPU0
Mon Dec  4 03:59:07.481 UTC
```

```
IDB Details:
  if_sname                : TF0/0/0/23
  if_handle               : 0x3c000068
  MacSecControlledIfh     : 0x3c008120
  MacSecUnControlledIfh   : 0x3c008128
  Replay window size      : 64
  Local MAC               : ac:4a:67:30:06:1b
  Rx SC Option(s)         : Validate-Frames Replay-Protect
  Tx SC Option(s)         : Protect-Frames Always-Include-SCI
  Security Policy         : MUST SECURE
  Delay Protection        : FALSE
  Sectag offset           : 0
  db_init Req             : 2023 Dec 03 09:36:22.656
  db_init Rsp             : 2023 Dec 03 09:36:22.662
  if_enable Req           : 2023 Dec 03 09:36:22.663
  if_enable Rsp           : 2023 Dec 03 09:36:23.127
  Rx SC 1
    Rx SCI                : ac3a67ee281b0001
    Peer MAC              : ac:3a:67:ee:28:1b
    Stale                 : NO
    SAK Data
      SAK[2]              : ***
      SAK Len             : 32
      SAK Version         : 1
      HashKey[2]          : ***
      HashKey Len         : 16
      Conf offset         : 0
      Cipher Suite        : GCM-AES-XPN-256
      CtxSalt[2]          : e8 5c ca 8f b3 7a 9d 65 2a 35 ac f8
      ssci                : 2
      Rx SA Program Req[2]: 2023 Dec 03 09:36:27.632
      Rx SA Program Rsp[2]: 2023 Dec 03 09:36:27.712
```

This example shows how to view events associated with the MACsec ea command.

```
Router#show macsec ea trace events

Mon Dec  4 03:57:58.463 UTC
59 wrapping entries (18496 possible, 320 allocated, 0 filtered, 59 total)
Dec  3 09:36:02.903 macsec_ea/events 0/RP0/CPU0 t6945 TP155: ********** MacSec EA(0x1b21)
process START ********.
Dec  3 09:36:02.926 macsec_ea/events 0/RP0/CPU0 t6945 TP180: macsec_ea_programming_conn_up_cb
 received.
Dec  3 09:36:02.966 macsec_ea/events 0/RP0/CPU0 t6945 TP191: macsec_ea_platform_init success
Dec  3 09:36:03.050 macsec_ea/events 0/RP0/CPU0 t6945 TP208: ea_plat_cb_evq:
event_async_attach success, pulse_code:0x7c
Dec  3 09:36:03.050 macsec_ea/events 0/RP0/CPU0 t6945 TP211: ea_plat_cb_evq: created
successfully
Dec  3 09:36:03.083 macsec_ea/events 0/RP0/CPU0 t6945 TP121: ********** Started MacSec
EA(0x1b21) Successfully ********.
```

# show macsec open-config

To display Open-config MACSEC traces, use the **show macsec open-config** command in EXEC mode.

**show macsec opwn-config trace**

**Syntax Description**

This command has no keywords or arguments.

**Command Default**

No default behavior or values.

**Command Modes**

EXEC mode

**Command History**

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

**Usage Guidelines**

The **show macsec open-config** command is available only with the installation of the k9sec rpm.

**Task ID**

| Task ID | Operation |
|---|---|
| cisco-support | read |

This example shows the output for **show macsec open-config trace**:

```
 Router#show macsec open-config trace
Fri Dec 15 09:08:37.760 UTC
20 wrapping entries (320 possible, 64 allocated, 0 filtered, 20 total)
Dec 12 12:42:43.823 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_edm_open:313, Successful
Dec 12 12:42:43.823 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_mka_oper_gl_sysdb_bind:173,
sysdb_bind successful
Dec 12 12:42:43.823 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_if_sysdb_bind:315, sysdb bind
 successful
Dec 12 12:42:43.827 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_mka_sysdb_bind:343, sysdb
bind: success
Dec 12 12:42:43.827 oc_macsec/all 0/RP0/CPU0 t16252
oc_macsec_mka_gl_stats_oper_sysdb_bind:372, sysdb_bind success
Dec 12 12:42:43.847 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_reg_cfg_notif:250, Successful
Dec 12 15:12:31.317 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_20: notif macsec_if_config, create/update
Dec 12 15:13:52.560 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_21: notif macsec_if_config, create/update
Dec 12 15:16:41.447 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_22: notif macsec_if_config, create/update
Dec 12 15:18:12.700 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_23: notif macsec_if_config, create/update
Dec 12 15:47:30.887 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TenGigE0_0_0_24: notif macsec_if_config, create/update
Dec 13 08:39:35.878 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TenGigE0_0_0_24: notif macsec_if_config, delete
Dec 13 08:46:15.995 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_20: notif macsec_if_config, delete
Dec 13 08:46:15.995 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
```

```
TwentyFiveGigE0_0_0_21: notif macsec_if_config, delete
Dec 13 08:46:15.995 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_22: notif macsec_if_config, delete
Dec 13 08:46:15.995 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_23: notif macsec_if_config, delete
Dec 13 09:25:40.478 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_24: notif macsec_if_config, create/update
Dec 13 09:27:59.242 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_25: notif macsec_if_config, create/update
Dec 13 09:29:32.355 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_26: notif macsec_if_config, create/update
Dec 13 09:31:03.658 oc_macsec/all 0/RP0/CPU0 t16252 oc_macsec_notify_if_macsec:74,
TwentyFiveGigE0_0_0_27: notif macsec_if_config, create/update
```

# show macsec platform hardware

To display hardware-specific details for MACsec on each interface, use the **show macsec platform hardware** command in EXEC mode.

**show macsec platform hardware [flow | sa | stats]** { **interface** *interface name* | **location** *location name* }

## Syntax Description

| | |
|---|---|
| **interface** | Specifies the interface name to view MACsec details. |
| *interface name* | Enables MACsec mode for a specified interface. |
| **location** | Specifies the node location to enable the MACsec details. |
| *location name* | Enables MACsec mode for a specific node. |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec platform hardware** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows how to view MACsec platform hardware information for a specific interface located on location 0/RP0/CPU0.

```
 Router# show macsec platform hardware flow location 0/RP0/CPU0
Wed Dec 20 08:39:18.958 UTC
------------------------------------------
Interface : TwentyFiveGigE0_0_0_27


------------------------------------------
Interface : TwentyFiveGigE0_0_0_26


------------------------------------------
Interface : TwentyFiveGigE0_0_0_25


------------------------------------------
```

```
Interface : TwentyFiveGigE0_0_0_24
```

# show macsec platform idb

To display interface database (IDB) details specific to MACsec, use the **show macsec platform idb** command in EXEC mode.

**show macsec platform idb** { **interface** *interface name* | **location** *location name* }

## Syntax Description

| | |
|---|---|
| **interface** | Specifies the interface name to view MACsec details. |
| *interface name* | Enables MACsec mode for a specified interface. |
| **location** | Specifies the node location to enable the MACsec details. |
| *location name* | Enables MACsec mode for a specific node. |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec platform idb** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows how to view MACsec platform idb information for a specific interface located on location 0/RP0/CPU0.

```
 Router# show macsec platform idb location 0/RP0/CPU0
Wed Dec 20 08:55:47.745 UTC

------------------------------------------------------------------
        EA IDB Details:
------------------------------------------------------------------
    IF Handle             : 0x3c000048
    IF Name               : TF0/0/0/27

------------------------------------------------------------------
        EA IDB Details:
------------------------------------------------------------------
    IF Handle             : 0x3c000050
    IF Name               : TF0/0/0/26

------------------------------------------------------------------
        EA IDB Details:
```

```
------------------------------------------------------------------
    IF Handle             : 0x3c000058
    IF Name               : TF0/0/0/25

------------------------------------------------------------------
        EA IDB Details:
------------------------------------------------------------------
    IF Handle             : 0x3c000060
    IF Name               : TF0/0/0/24
```

# show macsec platform stats

To display MACsec platform statistics, use the **show macsec platform stats** command in EXEC mode.

**show macsec platform stats** { **interface** *interface name* | **location** *location name* }

## Syntax Description

| | |
|---|---|
| **interface** | Specifies the interface name to view MACsec details. |
| *interface name* | Enables MACsec mode for a specified interface. |
| **location** | Specifies the node location to enable the MACsec details. |
| *location name* | Enables MACsec mode for a specific node. |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec platform stats** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows how to view MACsec platform statistics information for a specific interface located on location 0/RP0/CPU0.

```
 Router# show macsec platform stats location 0/RP0/CPU0
Wed Dec 20 08:56:13.285 UTC

------------------------------------------
Interface : TwentyFiveGigE0_0_0_27

------------------------------------------------------------------
    Global Statistics: Ingress
------------------------------------------------------------------
    Rx Ctrl Pkts                          : 47300
    Rx Ctrl Octets                        : 6905732
    Rx Data Pkts                          : 13
    Rx Data Octets                        : 894
    Rx OverSized Pkts                     : 0
    Rx Pkts Bad Tag                       : 0
    Rx Pkts No SCI                        : 0
    Rx Pkts No Tag                        : 0
    Rx Pkts Tagged                        : 0
    Rx Pkts Untagged                      : 0
```

```
    Rx Pkts Unknown SCI                     : 0
    Rx Pkts Untagged Miss                   : 0
    Rx Transform Error Pkts                 : 0
    Rx Pkts SA Not In Use                   : 0

-----------------------------------------------------------------
    Global Statistics: Egress
-----------------------------------------------------------------
    Tx Ctrl Pkts                            : 47308
    Tx Ctrl Octets                          : 6906216
    Tx Data Pkts                            : 16
    Tx Data Octets                          : 894
    Tx Pkts SA Not In Use                   : 0
    Tx Untagged Pkts                        : 0
    Tx Transform Error Pkts                 : 0

-----------------------------------------------------------------
    SA Statistics:Ingress
-----------------------------------------------------------------
    Index                                   : 0
    SCI                                     : ac3a67ee281f0001
    Current AN                              : 0
    Port                                    : 27
    Rx Data Pkts Decrypted                  : 13
    Rx Data Octets Decrypted                : 894
    Rx Pkts Delayed                         : 0
    Rx Pkts Invalid                         : 0
    Rx Pkts Late                            : 0
    Rx Pkts Not Using SA                    : 0
    Rx Pkts Not Valid                       : 0
    Rx Pkts Unchecked                       : 0
    Rx Pkts Untagged Hit                    : 0
    Rx Pkts Unused SA                       : 0
```

# show macsec platform trace

To display MACsec platform trace logs, use the **show macsec platform trace** command in EXEC mode.

**show macsec platform hardware trace [all | detail | errors | events]** { **interface** *interface name* | **location** *location name* }

## Syntax Description

| | |
|---|---|
| **interface** | Specifies the interface name to view MACsec details. |
| *interface name* | Enables MACsec mode for a specified interface. |
| **location** | Specifies the node location to enable the MACsec details. |
| *location name* | Enables MACsec mode for a specific node. |
| **all** | (Optional) Show **all** MACsec Platform traces for the specified node, or the current node if none is specified. |
| **detail** | (Optional) Show MACsec Platform**detail** traces for the specified node, or the current node if none is specified. |
| **errors** | Optional) Show MACsec Platform **error** traces for the specified node, or the current node if none is specified. |
| **events** | (Optional) Show MACsec Platform **event** traces for the specified node, or the current node if none is specified. |

## Command Default

No default behavior or values.

## Command Modes

EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

The **show macsec platform trace** command is available only with the installation of the k9sec rpm.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

This example shows how to view MACsec platform trace information for a specific interface located on location 0/RP0/CPU0.

```
 Router# show macsec platform trace detail location 0/RP0/CPU0
Wed Dec 20 08:57:03.178 UTC
2023-12-19:06.28.09.556530212:34390:secydrv_client_commu_ipc_common_fvt_init:COMMU_IPC_DET_36:secydrv_client_commu_ipc_common_fvt_init
```

```
 called
2023-12-19:06.28.09.556530980:34390:secydrv_client_commu_ipc_fvt_init:COMMU_IPC_DET_53:secydrv_client_commu_ipc_fvt_init
 called
2023-12-19:06.28.09.558317574:34390:secydrv_commu_ipc_platform_init:COMMU_IPC_DET_83:secydrv_commu_ipc_platform_init
 called
2023-12-19:06.28.10.579426302:34390:secydrv_commu_ipc_resync_start:COMMU_IPC_DET_106:secydrv_commu_ipc_resync_start
 called
2023-12-19:06.28.10.596378984:34390:secydrv_commu_ipc_resync_stop:COMMU_IPC_DET_129:secydrv_commu_ipc_resync_stop
 called
2023-12-19:06.28.19.598852376:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.28.29.598939886:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.28.39.599043710:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.28.49.599136368:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.28.59.599221556:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.29.09.599315246:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.29.19.599396186:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.29.29.599470492:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.29.39.599542858:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.29.49.599616712:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.29.59.599691262:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.30.09.599768752:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.30.19.599842944:34390:macsec_ea_platform_poll_pn_exceeded:EAPD_DET_3192:PN
Threshold Check:No active sessions
2023-12-19:06.30.27.011625732:34390:macsec_ea_platform_idb_init:EAPD_DET_1026:IDB Init:
ifh: 0x3c000060, if_name TF0/0/0/24, slot 0
2023-12-19:06.30.27.011632184:34390:secydrv_commu_ipc_if_init:COMMU_IPC_DET_151:secydrv_commu_ipc_if_init
 called
```

# vlan-tags-in-clear

To configure the number of VLAN tags to be unencrypted (in clear) in MACsec, use the **vlan-tags-in-clear** command in the MACsec policy configuration mode.

**vlan-tags-in-clear** *number*

## Syntax Description

| | |
|---|---|
| *number* | Specifies the number of VLAN tags in clear.<br>For single VLAN tag with 802.1q encapsulation, the value is 1.<br>For double VLAN tags with 802.1ad outer tag and 802.1q inner tag encapsulation, the value is 2. |

## Command Default

Default value is 1.

## Command Modes

MACsec policy configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 7.11.1 | This command was introduced. |

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **vlan-tags-in-clear** command:

```
Router# configure
Router(config)# macsec-policy mac_policy
Router(config-mac_policy)# vlan-tags-in-clear 1
Router(config-mac_policy)# commit
```

# 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

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to use the **window-size** command:

```
RP/0/RP0RSP0/CPU0:router# configure t
RP/0/RP0RSP0/CPU0:router(config)# macsec-policy mac_policy
RP/0/RP0RSP0/CPU0:router(config-mac_policy)# window-size 64
```

# URPF Commands

This module describes the commands used in enabling the Unicast Reverse Path Forwarding (uRPF).

For detailed information about FIPS configuration tasks, and examples, see the *Configuring FIPS Mode* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# hw-module profile cef unipath-surpf

To configure uRPF on strict mode, use the **hw-module profile cef unipath-surpf** command in the Global configuration mode.

**hw-module profile cef unipath-surpf enable**

## Syntax Description

| | |
|---|---|
| **enable** | Enables uRPF in strict mode. |

## Command Default

Strict mode in uRPF is disabled.

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| Release 7.9.1 | This command was introduced. |

## Usage Guidelines

You must configure both IPv4 and IPv6 commands to enable uRPF in strict mode.

You must reload the router after using the **hw-module profile cef unipath-surpf** command.

To disable the strict mode in uRPF use the **no** form of the **hw-module profile cef unipath-surpf** command.

## Task ID

| Task ID | Operation |
|---|---|
| acl | read, write |
| network | read, write |
| ipv4 | read, write |
| ipv6 | read, write |

### Example

This example shows how to configure uRPF in strict mode on the router:

```
Router# configure
Router(config)# hw-module profile cef unipath-surpf enable
Router(config-if)# commit
```

# ipv4/ipv6 verify unicast source reachable-via

To configure uRPF, use the **ipv4 verify unicast source reachable-via** command in the Interface configuration mode.

{ **ipv4** | **ipv6** } **verify unicast source reachable-via** { **any** | **rx** } [**allow-default**]

## Syntax Description

| | |
|---|---|
| **any** | (uRPF Loose Mode) Configures a source that is reachable any interface. |
| **rx** | (uRPF Strict Mode) Configures a source that is reachable onl the interface that ia same the interface used to transmit the back to source. |
| **allow-default** | Enables the matching of default routes. |

## Command Default

None

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| Release 7.9.1 | The **rx** keyword was introduced. |
| Release 7.3.15 | This command was introduced. |

## Usage Guidelines

You must configure both IPv4 and IPv6 commands to enable uRPF.

## Task ID

| Task ID | Operation |
|---|---|
| acl | read, write |
| network | read, write |
| ipv4 | read, write |
| ipv6 | read, write |

### Example

This example shows how to configure uRPF in loose mode on the router along with the default address.:

```
Router# configure
Router(config)# interface hundredGigE 0/0/0/0
Router(config-if)# ipv4 verify unicast source reachable-via any allow-default
Router(config-if)# ipv6 verify unicast source reachable-via any allow-default
Router(config-if)# commit
```

This example shows how to configure uRPF in strict mode on the router along with the default address.:

```
Router# configure
Router(config)# hw-module profile cef unipath-surpf enable
Router(config)# interface hundredGigE 0/0/0/0
Router(config-if)# ipv4 verify unicast source reachable-via rx allow-default
Router(config-if)# ipv6 verify unicast source reachable-via rx allow-default
Router(config-if)# commit
```

# 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 ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers* .

# 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:<br>• IPv4—*A.B.C.D/length*<br>• IPv6—*X.X:X.X* |

## Command Default

If no specific peer is configured, all peers are allowed.

## Command Modes

Interface peer configuration

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to configure the peer address for management traffic:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/CPU0:router(config-mpp)# inband
RP/0/RP0RSP0/CPU0:router(config-mpp-inband)# interface all
RP/0/RP0RSP0/CPU0:router(config-mpp-inbandoutband-all)# allow all peer
RP/0/RP0RSP0/CPU0:router(config-telnettftp-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

| | |
|---|---|
| *peer-ip-address* | Peer IPv6 address in which management traffic is allowed on the interface. This address can effectively be the source address of the management traffic that is coming in on the configured interface. |
| s*peer ip-address/length* | Prefix of the peer IPv6 address. |

## Command Default

If no specific peer is configured, all peers are allowed.

## Command Modes

Interface peer configuration

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to configure the peer IPv6 address 33::33 for management traffic:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/CPU0:router(config-mpp)# inband
RP/0/RP0RSP0/CPU0:router(config-mpp-inband)# interface HundredGigabitEthernet 0/1/1/2
RP/0/RP0RSP0/CPU0:router(config-mpp-inband-if)# allow TFTP peer
RP/0/RP0RSP0/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:<br>• HTTP(S)<br>• NETCONF (version 1.1 protocol)<br>• SNMP (also versions)<br>• Secure Shell (v1 and v2)<br>• TFTP<br>• Telnet<br>• 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

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

## 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 | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to configure all management protocols for all inband interfaces:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/CPU0:router(config-mpp)# inband
RP/0/RP0RSP0/CPU0:router(config-mpp-inband)# interface all
RP/0/RP0RSP0/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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl-mpp)# inband interface all allow xml peer address ipv4
 172.10.10.1
```

# control-plane

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

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

Use the **control-plane** command to enter control plane configuration mode.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to enter control plane configuration mode using the **control-plane** command:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/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

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

## Usage Guidelines

Use the **inband** command to enter management plane protection inband configuration mode.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to enter management plane protection inband configuration mode using the **inband** command:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/CPU0:router(config-mpp)# inband
RP/0/RP0RSP0/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.<br>**Note** Use the **show interfaces** command in EXEC mode to see a list of all interfaces currently configured on the router.<br>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 | Modification |
|---|---|
| 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.

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to configure all inband interfaces for MPP:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
```

```
RP/0/RP0RSP0/CPU0:router(config-mpp)# inband
RP/0/RP0RSP0/CPU0:router(config-mpp-inband)# interface all
RP/0/RP0RSP0/CPU0:router(config-mpp-inband-all)#
```

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/CPU0:router(config-mpp)# out-of-band
RP/0/RP0RSP0/CPU0:router(config-mpp-outband)# interface all
RP/0/RP0RSP0/CPU0:router(config-mpp-outband-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

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

## Usage Guidelines

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to enter management plane protection configuration mode using the **management-plane** command:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/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

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

## 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

| Task ID | Operations |
|---|---|
| system | read, write |

## Examples

The following example shows how to enter management plane protection out-of-band configuration mode using the **out-of-band** command:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/CPU0:router(config-mpp)# out-of-band
RP/0/RP0RSP0/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.<br>**Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router.<br>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

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

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

## Task ID

| Task ID | Operations |
|---|---|
| system | read |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# show mgmt-plane
```

```
Management Plane Protection

inband interfaces
----------------------

interface - HundredGigabitEthernet0_1_1_0
        ssh configured -
                All peers allowed
        telnet configured -
                peer v4 allowed - 10.1.0.0/16
        all configured -
                All peers allowed
interface - HundredGigabitEthernet0_1_1_0
        telnet configured -
                peer v4 allowed - 10.1.0.0/16

interface - all
        all configured -
                All peers allowed

outband interfaces
----------------------
interface - HundredGigabitEthernet0_1_1_0
        tftp configured -
                peer v6 allowed - 33::33
```

The following sample output displays the Virtual Private Network (VPN) routing and forwarding (VRF) reference of an out-of-band interface:

```
RP/0/RP0RSP0/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

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

## 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 |
|---|---|
| system | read |

## Examples

The following example shows how to configure the VRF:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# vrf my_out_of_band
RP/0/RP0RSP0/CPU0:router(config-vrf)# address-family ipv4 unicast
RP/0/RP0RSP0/CPU0:router(config-vrf-af)# exit
RP/0/RP0RSP0/CPU0:router(config-vrf)# address-family ipv6 unicast
RP/0/RP0RSP0/CPU0:router(config-vrf-af)# commit
RP/0/RP0RSP0/CPU0:router(config-vrf-af)# end
RP/0/RP0RSP0/CPU0:router#
```

The following example shows how to configure the VRF definition for MPP:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# control-plane
RP/0/RP0RSP0/CPU0:router(config-ctrl)# management-plane
RP/0/RP0RSP0/CPU0:router(config-mpp)# out-of-band
RP/0/RP0RSP0/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 ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# auto-enroll

To specify the duration after which the router request for automatic renewal of a PKI certificate from the CA, , use the **auto-enroll** command in trustpoint configuration mode. To disable the automatic renewal of the certificate after the said period, use the **no** form of this command.

**auto-enroll** *percentage*

## Syntax Description

| | |
|---|---|
| *percentage* | Percentage of the certificate validity after which the router will request for a new certificate from the CA. The range is from 1 to 99. |

## Command Default

None

## Command Modes

Trustpoint configuration

## Command History

| Release | Modification |
|---|---|
| Release 7.5.3 | This command was introduced. |

## Usage Guidelines

This command is applicable only for Cisco IOS XR 64-bit Software.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to configure auto renewal of PKI certificate in the router:

```
Router#configure
Router(config)#crypto ca trustpoint system-trustpoint
Router(config-trustp)#auto-enroll 30
Router(config-trustp)#commit
```

# 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** | **ed25519** | **rsa** } *key-pair-label*

## Syntax Description

| | |
|---|---|
| *key-pair-label* | Specifies the key pair label for the respective key signature algorithm (DSA, ECDSA, Ed25519 or RSA). |

## Command Default

None

## Command Modes

Trustpoint configuration

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |
| Release 7.3.1 | The command was modified to include the **ed25519** option. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

This example shows how to create the key pair for the root certificate on the router:

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

## Related Commands

| Command | Description |
|---|---|
| keypair, on page 284 | Creates the key pair for the leaf certificate on the router. |

# 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 EXEC modeXR EXEC mode.

**clear crypto ca certificates** *trustpoint*

## Syntax Description

| | |
|---|---|
| *trustpoint* | Trustpoint name. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## 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.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

The following example shows how to clear the certificates associated with trustpoints that no longer exist in the configuration file:

```
RP/0/RP0RSP0/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 EXEC modeXR EXEC mode.

**clear crypto ca crl**

## Syntax Description

This command has no keywords or arguments.

## Command Default

No default behavior or values

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## 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.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

The following example shows how to clear all CRLs stored on the router:

```
RP/0/RP0RSP0/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/RP0RSP0/CPU0:router# clear crypto ca crl
RP/0/RP0RSP0/CPU0:router# show crypto ca crls
RP/0/RP0RSP0/CPU0:router#
```

# crl optional (trustpoint)

To allow the certificates of other peers to be accepted without trying to obtain the appropriate CRL, use the **crl optional** command in trustpoint configuration mode. To return to the default behavior in which CRL checking is mandatory before your router can accept a certificate, use the **no** form of this command.

**crl optional**
**no crl optional**

## Syntax Description

This command has no keywords or arguments.

## Command Default

The router must have and check the appropriate CRL before accepting the certificate of another IP security peer.

## Command Modes

Trustpoint configuration

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0RSP0/CPU0:router(config-trustp)# enrollment url http://ca_server
RP/0/RP0RSP0/CPU0:router(config-trustp)# enrollment retry period 20
RP/0/RP0RSP0/CPU0:router(config-trustp)# enrollment retry count 100
RP/0/RP0RSP0/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 EXEC modeXR EXEC mode.

**crypto ca authenticate** {*ca-name* | **system-trustpoint**}

## Syntax Description

| | |
|---|---|
| *ca-name* | Name of the CA Server. |
| **system-trustpoint** | Generates self-signed root certificate. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | execute |

## 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
  Serial Number  : 06:A5:1B:E6:4F:5D:F7:83:41:11:D5:F9:22:7F:95:23
  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 EXEC modeXR EXEC mode.

**crypto ca cancel-enroll** *ca-name*

## Syntax Description

| | |
|---|---|
| *ca-name* | Name of the certification authority (CA). |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## 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 291 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.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

The following example shows how to cancel a current enrollment request from a CA named **myca**:

```
RP/0/RP0RSP0/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 EXEC modeXR 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

EXEC modeXR EXEC mode

## Command History

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

## 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 291 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

| Task ID | Operations |
|---|---|
| crypto | execute |

## 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 you password will not be saved in the configuration.
% Please make a note of it.
%Password
re-enter Password:
    Fingerprint: 4F35ADC9 2791997A CE211437 AFC66CF7
RP/0/RPO/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 fqdn-check ip-address allow

To avoid server certificate (leaf certificate) failure in the router, resulting from the IP addresses in the Subject Alternate Name (SAN) field of the certificates instead of Fully Qualified Domain Names (FQDNs) when the certificate extension type doesn't specifies the IP address, use the **crypto ca fqdn-check ip-address allow** command in Global Configuration mode.

**crypto ca fqdn-check ip-address allow**

## Syntax Description

This command has no keywords or arguments.

## Command Default

When the certificate extension type doesn't specifies the IP address, the certificates with IP addresses in the SAN field don't function properly.

## Command Modes

Global Configuration

## Command History

| Release | Modification |
|---|---|
| Release 7.4.2 | This command was introduced. |

## Usage Guidelines

In Cisco IOS XR Routers, to use an IP address in the SAN field in server certificates, the certificate extension type is IP addresses. The router rejects certificates that don't meet this criterion. To prevent such failures when an IP address is present in the SAN field, configure the **crypto ca fqdn-check ip-address allow** command. This command enables the router to validate and accept server certificates with IP addresses in the SAN field without the IP addresses certificate extension type.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

This example shows how to run the command for the router to accept server certificates with ip-address in the SAN field:

```
Router# config
Router(config)# crypto ca fqdn-check ip-address allow
```

# 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 EXEC modeXR EXEC mode.

**crypto ca import** *name* **certificate**

## Syntax Description

| | |
|---|---|
| *name* **certificate** | Name of the certification authority (CA). This name is the same name used when the CA was declared with the crypto ca trustpoint, on page 253 command. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| 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/RP0RSP0/CPU0:router# crypto ca import myca certificate
```

# crypto ca http-proxy

To fetch the Certificate Revocation List (CRL) through the http proxy server, use the **crypto ca http-proxy** command in the Global Configuration modeXR Config mode. Use the **no** form of this command to disable the proxy server.

**crypto ca http-proxy** *proxy-server-IP-address* **port** *port-number*
**no crypto ca http-proxy** *proxy-server-IP-address* **port** *port-number*

## Syntax Description

| | |
|---|---|
| **http-proxy** *proxy-server-IP-address* | Specifies the proxy server IP address. |
| **port** *port-number* | Specifies the proxy server port number. The range is from 1-65535. |

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

### Example

This example shows how to configure the proxy server to enable communication with the certification authority to retrieve the Certificate Revocation List (CRL).

```
Router#configure
Router(config)#crypto ca http-proxy 10.10.10.1 port 1
```

# crypto ca crl request

To fetch the latest CRL from a specific CDP (CRL Distribution point), use the **crypto ca crl request** command in the EXEC modeXR EXEC mode.

**crypto ca crl request** *cdp-url* [ **http-proxy** *ip-address* **port** *port-number* ]

## Syntax Description

| | |
|---|---|
| *cdp-url* | Specifies the CDP URL. |
| **http-proxy** *proxy-server-IP-address* | Specifies the proxy server IP address. |
| **port** *port-number* | Specifies the proxy server port number. The range is from 1-65535. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.1 | This command was modified. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

### Example

This example shows how to fetch the latest CRL from a specific CDP.

```
Router#crypto ca crl request http://zxy-w2k.cisco.com/CertEnroll/zxy-w2k-root.crl
Certificate Revocation List (CRL):
        Version 2 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: /C=US/ST=NC/L=RTP/O=Cisco/OU=GCT/CN=ca-root
        Last Update: Jan 29 11:43:50 2019 GMT
        Next Update: Jan 26 11:43:50 2029 GMT
        CRL extensions:
            xyz321v3 CRL Number:
                292
Revoked Certificates:
    Serial Number: 0138
        Revocation Date: Feb 17 01:01:55 2017 GMT
    Serial Number: 0139
        Revocation Date: Feb 17 01:22:28 2017 GMT
    Serial Number: 013A
        Revocation Date: Feb 17 03:04:32 2017 GMT
    Serial Number: 013B
```

# 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 Global Configuration modeXR Config mode.

**crypto ca trustpoint** {*ca-name* | **system-trustpoint**}

## Syntax Description

| | |
|---|---|
| *ca-name* | Name of the CA. |
| **system-trustpoint** | Specifies the default system trustpoint. |

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

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

## 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.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## 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
```

| Command | Description |
|---|---|
| ca-keypair, on page 240 | Creates the key pair for the root certificate on the router. |
| crl optional (trustpoint), on page 243 | Allows the certificates of other peers to be accepted without trying to obtain the appropriate CRL. |
| enrollment retry count, on page 276 | Specifies how many times a router resends a certificate request. |
| enrollment retry period, on page 277 | Specifies the wait period between certificate request retries. |
| enrollment terminal, on page 278 | Specifies manual cut-and-paste certificate enrollment. |
| enrollment url, on page 279 | Specifies the URL of the CA. |
| ip-address (trustpoint), on page 281 | Specifies a dotted IP address that is included as an unstructured address in the certificate request. |
| key-usage, on page 282 | Specifies the key usage field for the self-enrollment certificate. |
| keypair, on page 284 | Creates the key pair for the leaf certificate on the router. |
| lifetime (trustpoint), on page 287 | Configures the lifetime for self-enrollment of certificates. |
| message-digest, on page 288 | Configures the message digest hashing algorithm for the certificates. |
| query url, on page 289 | Specifies the LDAP URL of the CRL distribution point.<br>Required only if your CA supports Lightweight Directory Access Protocol (LDAP). |
| rsakeypair, on page 291 | Specifies a named RSA key pair for this trustpoint. |
| serial-number (trustpoint), on page 292 | Specifies a router serial number in the certificate request. |
| sftp-password (trustpoint), on page 293 | Secures the FTP password. |
| sftp-username (trustpoint), on page 294 | Secures the FTP username. |
| subject-name (trustpoint), on page 308 | Specifies a subject name in the certificate request. |

# crypto ca trustpool import url

To manually update certificates in the trust pool if they are not current, are corrupt, or if certain certificates need to be updated, use the **crypto ca trustpool import url** command in EXEC modeXR EXEC mode.

**crypto ca trustpool import url** { **clean** *URL* }

## Syntax Description

| | |
|---|---|
| **clean** | (Optional) Manually remove all downloaded certificate authority (CA) certificates. |
| *URL* | Specify the URL from which the CA trust pool certificate bundle must be downloaded. This manually imports (downloads) the CA certificate bundle into the CA trust pool to update or replace the existing CA certificate bundle.<br>This parameter can either be the URL of an external server or the local folder path (**/tmp**) in the router where the certificate is available. |

## Command Default

The CA trust pool feature is enabled. The router uses the built-in CA certificate bundle in the CA trust pool which is updated automatically from Cisco.

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

The CA trust pool feature is enabled by default and uses the built-in CA certificate bundle in the trust pool, which receives automatic updates from Cisco. Use the **crypto ca trustpool import url** command to manually update certificates in the trust pool if they are not current, are corrupt, or if certain certificates need to be updated.

You can also specify a local folder path (**/tmp**) in the router as the *URL* parameter for **crypto ca trustpool import url** command. This is useful in scenarios where the router does not have connectivity to an external server to download the certificate. In such cases, you can download the certificate from an external server to elsewhere, and then copy it to the **/tmp** folder in the router.

**Note** The local folder path in the router has to be **/tmp** itself; no other folder paths are allowed.

The format of the certificate can .pem, .der, or .p7b(bundle).

For example,

**crypto ca trustpool import url /tmp/certificate.pem**

**crypto ca trustpool import url /tmp/certificate.der**

**crypto ca trustpool import url /tmp/pki_bundle_tmp.p7b**

## Task ID

| Task ID | Operation |
|---|---|
| crypto | execute |

This example shows how to run the command to manually update certificates in the trust pool if they are not current, are corrupt, or if certain certificates need to be updated. The certificate is directly downloaded from an external server, in this case.

```
Router#crypto ca trustpool import url http://www.cisco.com/security/pki/trs/ios.p7b
```

This example shows how to import a certificate that resides in the local **/tmp** folder in the router:

```
Router#crypto ca trustpool import url /tmp/certificate.der
```

## Related Commands

| Command | Description |
|---|---|
| show crypto ca trustpool policy, on page 299 | Displays the CA trust pool certificates of the router in a verbose format. |

# crypto key generate authentication-ssh

To generate the cryptographic key pair for public key-based authentication of logged-in users on Cisco IOS XR routers that are configured as SSH clients, use the **crypto key generate authentication-ssh** command in EXEC modeXR EXEC mode.

**crypto key generate authentication-ssh rsa**

## Syntax Description

| | |
|---|---|
| **rsa** | Generates RSA key pairs for signing and encryption of packets for SSH public key-based authentication. |

## Command Default

None

## Command Modes

EXECXR EXEC

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | This command was introduced. |

## Usage Guidelines

Remote AAA servers such as RADIUS and TACACS+ servers do not support public key-based authentication. Hence this functionality is available only for users who are configured locally on the router and not for users who are configured remotely.

To delete the RSA key of a user, use the **crypto key zeroize authentication-ssh rsa username** command in EXEC modeXR EXEC mode.

A user with root privileges has permission to create and delete keys for other users.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

This example shows how to generate an RSA key pair for public key-based authentication of SSH clients on Cisco IOS XR routers:

```
Router#crypto key generate authentication-ssh rsa
Wed Dec 21 10:02:57.684 UTC
The name for the keys will be: cisco
  Choose the size of the key modulus in the range of 512 to 4096. Choosing a key modulus
greater than 512 may take a few minutes.

How many bits in the modulus [2048]:
Generating RSA keys ...
Done w/ crypto generate keypair
[OK]

Router#
```

# crypto key generate dsa

To generate Digital Signature Algorithm (DSA) key pairs, use the **crypto key generate dsa** command in XR EXEC mode and XR Config mode.

**crypto key generate dsa** [{**system-enroll-key** | **system-root-key**}]

## Syntax Description

| | |
|---|---|
| **system-enroll-key** | Specifies key pair generation for the leaf certificate.<br>Note: Crypto key generation in XR Config Mode does not support this option. |
| **system-root-key** | Specifies key pair generation for the root certificate.<br>Note: Crypto key generation in XR Config Mode does not support this option. |

## Command Default

None

## Command Modes

XR EXEC mode and XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.2 | This command was introduced in XR Config mode |
| Release 7.0.12 | This command was introduced in XR EXEC mode |

## 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 in XR Config mode, use **no** form of this command in XR Config mode.

To remove the DSA key generated in XR EXEC mode, use the **crypto key zeroize dsa** command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## 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:

```
Router#crypto key generate dsa system-enroll-key
```

The following example shows how to generate a 512-bit DSA key-pair in XR Config mode:

```
Router#conf t
Router(config)#crypto key generate dsa 512
Router(config)#commit
```

This example shows how to delete a DSA key-pair in XR Config mode:

```
Router# conf t
Router(config)#no crypto key generate dsa 512
Router(config)#commit
```

# crypto key generate ecdsa

To generate an Elliptic Curve Digital Signature Algorithm (ECDSA) key pair, use the **crypto key generate ecdsa** command in XR EXEC mode and XR Config mode.

**crypto key generate ecdsa** [{**nistp256** | **nistp384** | **nistp521**}] [{**system-enroll-key** | **system-root-key**}]

## Syntax Description

| | |
|---|---|
| **nistp256** | Generates an ECDSA key of curve type nistp256, with key size 256 bits. |
| **nistp384** | Generates an ECDSA key of curve type nistp384, with key size 384 bits. |
| **nistp521** | Generates an ECDSA key of curve type nistp521, with key size 521 bits. |
| **system-enroll-key** | Specifies key pair generation for the leaf certificate.<br>Note: Crypto key generation in XR Config Mode does not support this option. |
| **system-root-key** | Specifies key pair generation for the root certificate.<br>Note: Crypto key generation in XR Config Mode does not support this option. |

## Command Default

None

## Command Modes

XR EXEC mode and XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.2 | This command was introduced in XR Config mode |
| Release 7.0.12 | This command was introduced in XR EXEC mode |

## Usage Guidelines

To remove the ECDSA key generated in XR Config mode, use **no** form of this command in XR Config mode.

To remove an ECDSA key generated in XR EXEC mode, use the **crypto key zeroize ecdsa** command.

## Task ID

| Task ID | Operation |
|---|---|
| crypto | execute |

## Examples

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

```
Router# crypto key generate ecdsa nistp384
Wed Mar 28 12:53:57.355 UTC
% You already have keys defined for the_default
Do you really want to replace them? [yes/no]: yes
Generating ECDSA keys ...
Done w/ crypto generate ECDSA keypair
[OK]
```

This example shows how to generate a ECDSA key pair for the root certificate:

```
Router#crypto key generate ecdsa system-root-key
```

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

```
Router#crypto key generate dsa system-enroll-key
```

The following example shows how to generate an ECDSA key-pair in XR Config mode:

```
Router#conf t
Router(config)#crypto key generate ecdsa nistp256
Router(config)#commit
```

This example shows how to delete en ECDSA key-pair in XR Config mode:

```
Router# conf t
Router(config)#no crypto key generate ecdsa nistp256
Router(config)#commit
```

# crypto key generate ed25519

To generate Ed25519 crypto key pairs as part of supporting the Ed25519 public-key signature system, use the **crypto key generate ed25519** command in XR EXEC mode and XR Config mode.

**crypto key generate ed25519** [{ **system-enroll-key** | **system-root-key** }]

## Syntax Description

| | |
|---|---|
| **system-enroll-key** | Specifies key pair generation for the leaf certificate.<br>Note: Crypto key generation in XR Config Mode does not support this option. |
| **system-root-key** | Specifies key pair generation for the root certificate.<br>Note: Crypto key generation in XR Config Mode does not support this option. |

## Command Default

None

## Command Modes

XR EXEC mode and XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.2 | This command was introduced in XR Config mode |
| Release 7.3.1 | This command was introduced in XR EXEC mode. |

## Usage Guidelines

This command is applicable only for Cisco IOS XR 64-bit platforms.

To remove the Ed25519 key generated in XR Config mode, use **no** form of this command in XR Config mode.

To remove the Ed25519 key generated in XR EXEC mode, use the **crypto key zeroize ed25519** command.

You can generate the crypto keys either with an empty label or with two predefined labels (**system-root-key** and **system-enroll-key**). In case of empty label, the system generates the key pair against the default label. The key pairs with the predefined labels are used to integrate Cisco IOS XR with Cisco Crosswork Trust Insights.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

This example shows how to generate a Ed25519 crypto key pair:

```
Router# crypto key generate ed25519

Mon Nov 30 07:03:17.058 UTC
The name for the keys will be: the_default
Generating ED25519 keys ...
Done w/ crypto generate keypair
```

```
[OK]
```

This example shows how to generate a Ed25519 crypto key pair for the root certificate:

```
Router#crypto key generate ed25519 system-root-key
```

This example shows how to generate a Ed25519 crypto key pair for the leaf certificate:

```
Router#crypto key generate ed25519 system-enroll-key
```

The following example shows how to generate an Ed25519 key-pair in XR Config mode:

```
Router#conf t
Router(config)#crypto key generate ed25519
Router(config)#commit
```

This example shows how to delete en Ed25519 key-pair in XR Config mode:

```
Router# conf t
Router(config)#no crypto key generate ed25519
Router(config)#commit
```

## Related Commands

| Command | Description |
|---|---|
| crypto key zeroize ed25519, on page 273 | Deletes Ed25519 crypto key pairs from the router. |
| show crypto key mypubkey ed25519, on page 304 | Displays the Ed25519 public keys of the router. |

# 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 and XR Config mode. .

**crypto key generate rsa** [{**usage-keys** | **general-keys** | **system-enroll-key** | **system-root-key**}] [*keypair-label*]

## Syntax Description

| | |
|---|---|
| usage-keys | (Optional) Generates separate RSA key pairs for signing and encryption. |
| general-keys | (Optional) Generates a general-purpose RSA key pair for signing and encryption. |
| *keypair-label* | (Optional) RSA key pair label that names the RSA key pairs. |
| **system-enroll-key** | Specifies key pair generation for the leaf certificate.<br>Note: Crypto key generation in XR Config Mode does not support this option. |
| **system-root-key** | Specifies key pair generation for the root certificate.<br>Note: Crypto key generation in XR Config Mode does not support this option. |

## 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 and XR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.2 | This command was introduced in XR Config mode |
| Release 7.0.12 | This command was introduced in XR EXEC mode. |

## 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 generated in XR Config mode, use **no** form of this command in XR Config mode.

To remove an RSA key generated in XR EXEC mode, use the **crypto key zeroize rsa** command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## 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
```

The following example shows how to generate an RSA key-pair in XR Config mode:

```
Router#conf t
Router(config)#crypto key generate rsa user1 general-keys 2048
Router(config)#commit
```

This example shows how to delete en RSA key-pair in XR Config mode:

```
Router# conf t
Router(config)#no crypto key generate rsa user1 general-keys 2048
Router(config)#commit
```

# 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 EXEC modeXR EXEC mode.

**crypto key import authentication rsa** [ **username** *name* ] [ **WORD** | **second** | **third** | **fourth** ]

## Syntax Description

| | |
|---|---|
| **rsa** | Imports the RSA public key on the router. |
| **username** | (Optional) Imports the RSA public key for the user *name*. |
| *name* | Specifies the name of the user for which the RSA public key is imported.<br>If you do not specify a *name*, the RSA public key for the currently logged-in user is imported. |
| **WORD** | (Optional) Specifies the path (`harddisk:/` or `disk0:/` or `tftp`) to the RSA public key file. |
| **second** | (Optional) Imports the second RSA public key for a user. |
| **third** | (Optional) Imports the third RSA public key for a user. |
| **fourth** | (Optional) Imports the fourth RSA public key for a user. |

## Command Default

- The **crypto key import authentication rsa** command imports the first RSA public key for the currently logged-in user if you do not specify the **WORD**, **second**, **third**, or **fourth** option.
- The **crypto key import authentication rsa username** *name* command imports the first RSA public key for the user *name* if you do not specify the **second**,**third**, or **fourth** option.

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.11.1 | This command was modified to include the **second**,**third**,and **fourth** options. |
| Release 3.9.0 | This command was introduced. |

## Usage Guidelines

1. Use shh-keygen generation mechanism to generate keys using either a LINUX or UNIX client. This creates two keys: one public and one private.
2. Remove the comment and other header tag from the keys, except the base64encoded text.
3. Decode the base64encoded text, and use the for authentication.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

This example shows how to import the second RSA public key for the currently logged-in user.

```
RP/0/RP0/CPU0:OC_router1#crypto key import authentication rsa harddisk:/id_rsa_key2.pub
Thu Nov  9 20:43:19.568 IST
RP/0/RP0/CPU0:Nov  9 20:43:19.740 IST: cepki[129]: %SECURITY-CEPKI-6-KEY_INFO : crypto key
 RSA(public key authentication) generated, label:cafyauto, modBits:4096
RP/0/RP0/CPU0:OC_router1#RP/0/RP0/CPU0:Nov  9 20:43:20.964 IST: cepki[129]:
%SECURITY-CEPKI-6-INFO : key database updated successfully
RP/0/RP0/CPU0:OC_router1#
```

This example shows how to import the third RSA public key for the currently logged-in user by manually copy-pasting the key.

```
RP/0/RP0/CPU0:OC_router1#crypto key import authentication rsa third
Thu Nov  9 20:51:52.599 IST
Enter the public key
ssh-rsa
[redacted]

RP/0/RP0/CPU0:Nov  9 20:52:38.122 IST: cepki[129]: %SECURITY-CEPKI-6-KEY_INFO : crypto key
 RSA(public key authentication) generated, label:cafyauto, modBits:4096
RP/0/RP0/CPU0:OC_router1#
```

This example shows how to import the fourth RSA public key for user *test*.

```
RP/0/RP0/CPU0:OC_router1#crypto key import authentication rsa username test fourth
harddisk:/id_rsa_key4.pub
Thu Nov  9 20:55:02.586 IST
RP/0/RP0/CPU0:Nov  9 20:55:02.757 IST: cepki[129]: %SECURITY-CEPKI-6-KEY_INFO : crypto key
 RSA(public key authentication) generated, label:test, modBits:4096
RP/0/RP0/CPU0:OC_router1
```

# crypto key zeroize authentication-ssh

To delete the cryptographic key pair on the router that was generated for public key-based authentication of SSH clients, use the **crypto key zeroize authentication-ssh** command in EXEC modeXR EXEC mode.

**crypto key zeroize authentication-ssh rsa** [ **username** *name* ]

## Syntax Description

| | |
|---|---|
| **rsa** | Deletes the RSA key pair on the router. |
| **username** *name* | Specifies the name of the user whose RSA key pairs are to be deleted from the router. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | This command was introduced. |

## Usage Guidelines

If the **username** is not specified, then the command deletes the key for the user who is currently logged in.

A user with root privileges has permission to create and delete keys for other users.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

This example shows how to delete the RSA key pair that was generated for public key-based authentication of SSH clients.

```
Router#crypto key zeroize authentication-ssh rsa username user1
```

# crypto key zeroize authentication rsa

To delete a public key imported on the router using the Rivest, Shamir, and Adelman (RSA) method, use the **crypto key zeroize authentication rsa** command in EXEC modeXR EXEC mode.

**crypto key zeroize authentication rsa** [ **username** *name* ] [ **all** | **second** | **third** | **fourth** ]

## Syntax Description

| | |
|---|---|
| **rsa** | Deletes the RSA public key on the router. |
| **username** | Deletes the RSA public key for the user specified in the *name*. |
| *name* | (Optional) Specifies the name of the user for which the RSA public key is deleted.<br>If you do not specify a *name*, the RSA public key for the currently logged-in user is deleted. |
| **all** | Deletes all imported RSA public keys. |
| **second** | Deletes second imported RSA public key. |
| **third** | Deletes third imported RSA public key. |
| **fourth** | Deletes fourth imported RSA public key. |

## Command Default

- The **crypto key zeroize authentication rsa** command deletes the first imported RSA public key if you do not specify the **all**, **second**, **third**, or **fourth** option.
- The **crypto key zeroize authentication rsa username** *name* command deletes the first imported RSA public key for the user *name* if you do not specify the **second**,**third**, or **fourth** option.

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.11.1 | This command was modified to include the **second**,**third**, and **fourth** options. |
| Release 7.2.1 | This command was introduced. |

## Usage Guidelines

If the **username** is not specified, then the command deletes the first imported RSA public key for the currently logged-in user.

A user with root privileges can create and delete keys for other users.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

This example shows how to delete the first imported RSA public key for the currently logged-in user *test1*.

```
RP/0/RP0/CPU0:OC_router1#crypto key zeroize authentication rsa

Wed Oct 25 18:32:30.421 IST
% Keys to be removed are named test1
Do you really want to remove these keys ?? [yes/no]: yes

RP/0/RP0/CPU0:OC_router1#
```

This example shows how to delete the fourth imported RSA public key for the currently logged-in user *test1*.

```
RP/0/RP0/CPU0:OC_router1#crypto key zeroize authentication rsa fourth

Wed Oct 25 21:18:04.336 IST
% Keys to be removed are named test1
Do you really want to remove these keys ?? [yes/no]: yes

RP/0/RP0/CPU0:OC_router1#
```

This example shows how to delete the first imported RSA public key for user *test2*.

```
RP/0/RP0/CPU0:OC_router1#crypto key zeroize authentication rsa username test2

Wed Oct 25 18:54:34.153 IST
% Keys to be removed are named test2
Do you really want to remove these keys ?? [yes/no]: yes

RP/0/RP0/CPU0:OC_router1#
```

This example shows how to delete the second imported RSA public key for user *test3*.

```
RP/0/RP0/CPU0:OC_router1#crypto key zeroize authentication rsa username test3 second

Wed Oct 25 18:54:34.153 IST
% Keys to be removed are named test3
Do you really want to remove these keys ?? [yes/no]: yes

RP/0/RP0/CPU0:OC_router1#
```

This example shows how to delete all imported RSA public keys on the router in EXEC mode.

```
RP/0/RP0/CPU0:OC_router1#crypto key zeroize authentication rsa all

Wed Oct 25 18:32:58.007 IST
Do you really want to remove all these keys ?? [yes/no]: yes

RP/0/RP0/CPU0:OC_router1#
```

# crypto key zeroize dsa

To delete the Digital Signature Algorithm (DSA) key pair from your router, use the **crypto key zeroize dsa** command in EXEC modeXR EXEC mode.

**crypto key zeroize dsa**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

Use the **crypto key zeroize dsa** command to delete the DSA key pair that was previously generated by your router.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

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

```
RP/0/RP0RSP0/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

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

## Usage Guidelines

None

## Task ID

| Task ID | Operation |
|---|---|
| crypto | execute |

### 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
```

# crypto key zeroize ed25519

To delete the Ed25519 crypto key pair from the router, use the **crypto key zeroize ed25519** command in EXEC modeXR EXEC mode.

**crypto key zeroize ed25519**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

This example shows how to delete Ed25519 crypto key pairs from your router:

```
Router# crypto key zeroize ed25519
% Keys to be removed are named the_default
Do you really want to remove these keys? [yes/no]: yes
```

## Related Commands

| Command | Description |
|---|---|
| crypto key generate ed25519, on page 262 | Generates Ed25519 crypto key pairs. |
| show crypto key mypubkey ed25519, on page 304 | Displays the Ed25519 public keys of your router. |

# crypto key zeroize rsa

To delete all Rivest, Shamir, and Adelman (RSA) keys from the router, use the **crypto key zeroize rsa** command in EXEC modeXR 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

EXEC modeXR EXEC mode

## Command History

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

## 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 247 command CA.
- Manually remove the certificates from the configuration using the **clear crypto ca certificates** command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## Examples

The following example shows how to delete the general-purpose RSA key pair that was previously generated:

```
RP/0/RP0RSP0/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

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

## Usage Guidelines

Use the **description** command in the trustpoint configuration mode to create a description for a trustpoint.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to create a trustpoint description:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0RSP0/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

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

## 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

| Task ID | Operations |
|---|---|
| crypto | 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0RSP0/CPU0:router(config-trustp)# enrollment url http://ca_server
RP/0/RP0RSP0/CPU0:router(config-trustp)# enrollment retry period 10
RP/0/RP0RSP0/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

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to declare a CA and change the retry period to 5 minutes:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0RSP0/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

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## 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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0RSP0/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).<br><br>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

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

## 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 11: Certificate Enrollment Methods***

| Enrollment Method | Description |
|---|---|
| SFTP | Enroll through SFTP: file system |
| TFTP[1] | Enroll through TFTP: file system |

[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

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

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

```
 RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)#
               crypto ca trustpoint myca
 RP/0/RP0RSP0/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

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

## Usage Guidelines

Use the **ip-address** command to include the IP address of the specified interface in the certificate request or to specify that an IP address should not be included in the certificate request.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to include the IP address of the Ethernet-0 interface in the certificate request for the trustpoint frog:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint frog
RP/0/RP0RSP0/CPU0:router(config-trustp)# enrollment url http://frog.phoobin.com
RP/0/RP0RSP0/CPU0:router(config-trustp)# subject-name OU=Spiral Dept., O=tiedye.com
RP/0/RP0RSP0/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/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0RSP0/CPU0:router(config-trustp)# enrollment url http://10.3.0.7:80
RP/0/RP0RSP0/CPU0:router(config-trustp)# subject-name CN=subject1, OU=PKI, O=Cisco Systems,
 C=US
RP/0/RP0RSP0/CPU0:router(config-trustp)# ip-address none
```

# key-usage

To specify the key usage field for the self-enrollment certificate, use the **key-usage** command in trustpoint configuration mode. To remove this configuration, use the **no** form of this command.

**key-usage** {**ca-certificate** {**crlsign** | **digitalsignature** | **keycertsign** | **nonrepudiation**} | **certificate** {**dataencipherment** | **digitalsignature** | **keyagreement** | **keyencipherment** | **nonrepudiation**}}

## Syntax Description

| | |
|---|---|
| **ca-certificate** | Specifies the key usage field for the CA certificate. |
| **certificate** | Specifies the key usage field for the leaf certificate. |
| **crlsign** | Asserts **cRLSign** (bit 6) for the key usage field to verify signatures on certificate revocation list (CRL). |
| **digitalsignature** | Asserts **digitalSignature** (bit 0) for the key usage field.<br>This is used when the subject public key is used with a digital signature mechanism to support security services other than certificate signing (bit 5), or CRL signing (bit 6). |
| **keycertsign** | Asserts **keyCertSign** (bit 5) for the key usage field when the subject public key is used for verifying a signature on public key certificates. |
| **nonrepudiation** | Asserts **nonRepudiation** (bit 1) for the key usage field when the subject public key is used to verify digital signatures that is used to provide a non-repudiation service. |
| **dataencipherment** | Asserts **dataEncipherment** (bit 3) for the key usage field when the subject public key is used for enciphering user data, other than cryptographic keys. |
| **keyagreement** | Asserts **keyAgreement** (bit 4) for the key usage field when the subject public key is used for key agreement. |
| **keyencipherment** | Asserts **keyEncipherment** (bit 2) for the key usage field when the subject public key is used for key transport. |

## Command Default

None

## Command Modes

Trustpoint configuration

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

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** | **ed25519** | **rsa** } *key-pair-label*

## Syntax Description

| | |
|---|---|
| *key-pair-label* | Specifies the key pair label for the respective key signature algorithm (DSA, ECDSA, Ed25519 or RSA). |

## Command Default

None

## Command Modes

Trustpoint configuration

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |
| Release 7.3.1 | The command was modified to include the **ed25519** option. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

This example shows how to create the key pair for the leaf certificate on the router:

```
Router#configure
Router(config)#crypto ca trustpoint system-trustpoint
Router(config-trustp)#keypair rsa system-enroll-key
Router(config-trustp)#commit
```

## Related Commands

| Command | Description |
|---|---|
| ca-keypair, on page 240 | Creates the key pair for the root certificate on the router. |

# keystring

To import the RSA public key in SSH format into the router for authenticating a user, use the **keystring** command in the SSH user key configuration mode. To remove the imported public key, use the **no** form of this command.

**keystring** [ **second** | **third** | **fourth** ] *key*

## Syntax Description

| | |
|---|---|
| **second** | (Optional) Imports the second RSA public key. |
| **third** | (Optional) Imports the third RSA public key. |
| **fourth** | (Optional) Imports the fourth RSA public key. |
| *key* | Specifies the key in SSH format. |

## Command Default

The command imports the first RSA public key into the router if none of the options are specified.

## Command Modes

SSH user key configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 7.11.1 | This command was modified to include the **second**,**third**, and **fourth** options. |
| Release 7.2.1 | This command was introduced. |

## Usage Guidelines

This command imports the first RSA public key if you do not specify the **second**,**third**, or **fourth** option.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

This example shows how to import the first RSA public key specified in SSH format for user *test*.

```
RP/0/RP0/CPU0:OC_router1#conf t
Tue Nov  7 20:28:58.585 IST
RP/0/RP0/CPU0:OC_router1(config)#ssh server username test
RP/0/RP0/CPU0:OC_router1(config-user-key)#keystring ssh-rsa
[redacted]
RP/0/RP0/CPU0:OC_router1(config-user-key)#commit
Tue Nov  7 20:29:19.109 IST
RP/0/RP0/CPU0:OC_router1(config-user-key)#
```

This example shows how to import the third RSA public key specified in SSH format for user *test*.

```
RP/0/RP0/CPU0:OC_router1#conf t
Tue Nov  7 20:28:58.585 IST
RP/0/RP0/CPU0:OC_router1(config)#ssh server  username test
```

```
RP/0/RP0/CPU0:OC_router1(config-user-key)#keystring third ssh-rsa
[redacted]

RP/0/RP0/CPU0:OC_router1(config-user-key)#commit
Tue Nov  7 20:30:51.892 IST
RP/0/RP0/CPU0:OC_router1(config-user-key)#
```

# 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.<br>The range is from 30 to 5474 days. |

## Command Default

None

## Command Modes

Trustpoint configuration

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| 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

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | 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).<br>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

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

## Usage Guidelines

LDAP is a query protocol used when the router retrieves the Certificate Revocation List (CRL). The certification authority (CA) administrator should be able to tell you whether the CA supports LDAP; if the CA supports LDAP, the CA administrator can tell you the LDAP location where certificates and certificate revocation lists should be retrieved.

To change the query URL, repeat the **query url** command to overwrite the previous URL.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0RSP0/CPU0:router(config-trustp)# query url ldap://my-ldap.domain.com
```

# renewal-message-type

Allows you to configure the request type from the router to the CA for automatic PKI certificate renewal.

**renewal-message-type** { **pkcsreq** | **renewalreq** }

## Syntax Description

| | |
|---|---|
| **pkcsreq** | The router uses Public Key Cryptography Standards (PKCS) requests for automatic PKI certificate renewal. |
| **renewalreq** | The router uses Renew requests for automatic PKI certificate renewal. |

## Command Default

By default, the PKCS request is available in the router.

## Command Modes

Trustpoint configuration

## Command History

| Release | Modification |
|---|---|
| Release 7.5.3 | This command was introduced. |

## Usage Guidelines

This command is applicable only for Cisco IOS XR 64-bit Software.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

This example shows how to use this command in the router:

```
Router#configure
Router(config)#crypto ca trustpoint system-trustpoint
Router(config-trustp)# renewal-message-type renewalreq
Router(config-trustp)# keypair rsa system-enroll-key
Router(config-trustp)# commit
```

# 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

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to specify the named RSA key pair key1 for the trustpoint myca:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint myca
RP/0/RP0RSP0/CPU0:router(config-trustp)# rsakeypair key1
```

# serial-number (trustpoint)

To specify whether the router serial number should be included in the certificate request, use the **serial-number** command in trustpoint configuration mode. To restore the default behavior, use the **no** form of this command.

**serial-number** [**none**]
**no serial-number**

## Syntax Description

| | |
|---|---|
| none | (Optional) Specifies that a serial number is not included in the certificate request. |

## Command Default

You are prompted for the serial number during certificate enrollment.

## Command Modes

Trustpoint configuration

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to omit a serial number from the root certificate request:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint root
RP/0/RP0RSP0/CPU0:router(config-trustp)# enrollment url http://10.3.0.7:80
RP/0/RP0RSP0/CPU0:router(config-trustp)# ip-address none
RP/0/RP0RSP0/CPU0:router(config-trustp)# serial-number none
RP/0/RP0RSP0/CPU0:router(config-trustp)# subject-name ON=Jack, OU=PKI, O=Cisco Systems,
C=US
```

# sftp-password (trustpoint)

To secure the FTP password, use the **sftp-password** command in trustpoint configuration mode. To disable this feature, use the **no** form of this command.

**sftp-password** {*clear text* | **clear** *text* | **password** *encrypted string*}
**no sftp-password** {*clear text* | **clear** *text* | **password** *encrypted string*}

## Syntax Description

| | |
|---|---|
| *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

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to secure the FTP password in an encrypted form:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint msiox
RP/0/RP0RSP0/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

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to secure the FTP username:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto ca trustpoint msiox
RP/0/RP0RSP0/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 EXEC modeXR EXEC mode.

**show crypto ca certificates**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | This command was modified to include the **Trusted Certificate Chain** field in the output as part of supporting multi-tier CA for trustpoint authentication. |
| Release 7.0.12 | This command was introduced. |

## 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

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

The following sample output is from the **show crypto ca certificates** command:

```
RP/0/RP0RSP0/CPU0:router# show crypto ca certificates
Trustpoint       : msiox
================================================
CAa certificate
  Serial Number  : 06:A5:1B:E6:4F:5D:F7:83:41:11:D5:F9:22:7F:95:23
  Subject:
    Name: CA2
    CN= CA2
  Issued By      :
        cn=CA2
  Validity Start : 07:51:51 UTC Wed Jul 06 2005
  Validity End   : 08:00:43 UTC Tue Jul 06 2010
  CRL Distribution Point
        http://10.56.8.236/CertEnroll/CA2.crl
Router certificate
```

```
  Status         : Available
  Key usage      : Signature
  Serial Number  : 38:6B:C6:B8:00:04:00:00:01:45
  Subject:
    Name: tdlr533.cisco.com
    IP Address: 3.1.53.3
    Serial Number: 8cd96b64
  Issued By      :
        cn=CA2
  Validity Start : 08:30:03 UTC Mon Apr 10 2006
  Validity End   : 08:40:03 UTC Tue Apr 10 2007
  CRL Distribution Point
        http://10.56.8.236/CertEnroll/CA2.crl
Associated Trustpoint: MS-IOX
Router certificate
  Status         : Available
  Key usage      : Encryption
  Serial Number  : 38:6D:2B:A7:00:04:00:00:01:46
  Subject:
    Name: tdlr533.cisco.com
    IP Address: 3.1.53.3
    Serial Number: 8cd96b64
  Issued By      :
        cn=CA2
  Validity Start : 08:31:34 UTC Mon Apr 10 2006
  Validity End   : 08:41:34 UTC Tue Apr 10 2007
  CRL Distribution Point
        http://10.56.8.236/CertEnroll/CA2.crl
Associated Trustpoint: msiox
```

The following is a sample output with multi-tier CA. The command output displays the **Trusted Certificate Chain** field if there is one or more subordinate CAs involved in the hierarchy.

```
Router#show crypto ca certificates test-ca
Mon Feb  6 09:03:53.019 UTC

Trustpoint       : test-ca
==================================================
CA certificate
  Serial Number  : 10:01
  Subject:
            CN=SUB_CA_CERT,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Issued By      :
            CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Validity Start : 12:31:40 UTC Sun Jun 14 2020
  Validity End   : 12:31:40 UTC Wed Jun 12 2030

  CRL Distribution Point
            http://10.105.236.78/crl_akshath_two_level_ca/crl.der
  SHA1 Fingerprint:
             D8E0C11ECED96F67FDBC800DB6A126676A76BD62
Trusted Certificate Chain
  Serial Number  : 0F:A0:06:7A:C9:5E:A9:E7:61:A2:B9:2B:27:D1:D6:8F:3D:51:43:3B
  Subject:
            CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Issued By      :
            CN=TWO-LEVEL-CA,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Validity Start : 13:12:32 UTC Sun Jun 07 2020
  Validity End   : 13:12:32 UTC Sat Jun 02 2040

  CRL Distribution Point
            http://10.105.236.78/crl_akshath_two_level_ca/crl.der
  SHA1 Fingerprint:
             08E71248FB7578614442E713AC87C461D173952F
```

```
Router certificate
  Key usage      : General Purpose
  Status         : Available
  Serial Number  : 28:E5
  Subject:
           CN=test
  Issued By      :
           CN=SUB_CA_CERT,OU=SPBU,O=CSCO,L=BGL,ST=KA,C=IN
  Validity Start : 08:49:54 UTC Mon Feb 06 2023
  Validity End   : 08:49:54 UTC Wed Mar 08 2023
  SHA1 Fingerprint:
            6C8644FA67D9CEBC7C5665C35838265F578835AB
Associated Trustpoint: test-ca
```

# show crypto ca crls

To display information about the local cache Certificate Revocation List (CRL), use the **show crypto ca crls** command in EXEC modeXR EXEC mode.

**show crypto ca crls**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

The following sample output is from the **show crypto ca crls** command:

```
RP/0/RP0RSP0/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
Next Update : [UTC] Thu Jan 17 13:21:14 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 EXEC modeXR 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

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

Use the command to display the CA trust pool certificates of the router in a verbose format.

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read |

### Example

This example shows you how to run the command to view details of your CA certificate trust pool policy.

```
RP/0/RP0RSP0/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 authentication-ssh

To display the cryptographic keys that are used for the public key-based authentication of SSH clients on the router, use the **show crypto key mypubkey authentication-ssh** command in EXEC modeXR EXEC mode.

**show crypto key mypubkey authentication-ssh rsa** [{ **all** | **username** *name* }]

## Syntax Description

| | |
|---|---|
| **rsa** | Displays the RSA key of the user. |
| **username** *name* | Specifies the name of the user whose RSA key is to be displayed. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.10.1 | This command was introduced. |

## Usage Guidelines

If the **username** is not specified, then the command displays the key for the currently logged-in user.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

This example shows how to display the RSA key used for public key-based authentication of SSH clients on Cisco IOS XR routers:

```
Router#show crypto key mypubkey authentication-ssh rsa
Wed Dec 21 10:24:34.226 UTC
Key label: cisco
Type     : RSA Authentication
Size     : 2048
Created  : 10:02:59 UTC Wed Dec 21 2022
Data     :
 30820122 300D0609 2A864886 F70D0101 01050003 82010F00 3082010A 02820101
 00A292B0 E45ACBB9 47B9EDA8 47E4664E 58FC3EA5 CE0F6B7A 3C6B7A73 537E6CEB
 .
 .
 .
 FF6BAF95 D9617CF6 65C058CC 7C6C22A9 9E48CC43 FDFF0EB7 ABADEB77 55A274DB
 15020301 0001

OpenSSH Format:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCikrDkWsu5R7ntqEfkZk5Y/.../2uvldlhfPZlwFjMfGwiqZ5IzEP9/w63q63rd1WidNsV

Router#
```

The key value starts with *ssh-rsa* in the above output.

# 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 EXEC modeXR EXEC mode.

**show crypto key mypubkey dsa**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

The following sample output is from the **show crypto key mypubkey dsa** command:

```
RP/0/RP0RSP0/CPU0:router# show crypto key mypubkey dsa

Key label: mykey
Type : RSA General purpose
Size : 1024
Created : 17:33:23 UTC Thu Sep 18 2003
Data :
3081F230 81AA0605 2B0E0302 0C3081A0 02020200 024100C8 A36B6179 56B8D620
1F77595C 32EF3004 577A9F79 0A8ABDA4 89FB969D 35C04E7E 5491ED4E 120C657C
610576E5 841696B6 0948846C C92F56E5 B4921458 70FC4902 1500AB61 5C0D63D3
EB082BB9 F16030C5 AA0B5D1A DFE50240 73F661EA 9F579E77 B413DBC4 9047B4F2
10A1CFCB 14D98B57 3E0BBA97 9B5120AD F52BBDC7 15B63454 8CB54885 92B6C9DF
7DC27768 FD296844 42024945 5E86C81A 03430002 4071B49E F80F9E4B AF2B62E7
AA817460 87EFD503 C668AD8C D606050B 225CC277 7C0A0974 8072D7D7 2ADDDE42
329FE896 AB015ED1 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**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC

## Command History

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

## Usage Guidelines

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read |

### 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 ed25519

To display the Ed25519 crypto public keys of your router, use the **show crypto key mypubkey ed25519** command in EXEC modeXR EXEC mode.

**show crypto key mypubkey ed25519**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

This example shows the sample output of the **show crypto key mypubkey ed25519** command:

```
Router# show crypto key mypubkey  ed25519

Mon Nov 30 07:05:06.532 UTC
Key label: the_default
Type : ED25519
Size : 256
Created : 07:03:17 UTC Mon Nov 30 2020
Data :
FF0ED4E7 71531B3D 9ED72C48 3F79EC59 9EFECCC3 46A129B2 FAAA12DD EE9D0351
```

## Related Commands

| Command | Description |
|---|---|
| crypto key generate ed25519, on page 262 | Generates Ed25519 crypto key pairs. |
| crypto key zeroize ed25519, on page 273 | Deletes all Ed25519 keys from the router. |

# 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 EXEC modeXR EXEC mode.

**show crypto key mypubkey rsa**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

The following is sample output from the **show crypto key mypubkey rsa** command:

```
RP/0/RP0RSP0/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
5BFCA055 DA4D164D F6EDB78B 926B1DDE 0383027F BA71BCC6 9D5592C4 5BA8670E
35CD19B7 1C973A46 62CC5F8C 82BD596C F292410F 8E83B753 4BA71BAC 41AB6B60
F34A2499 EDE11639 F88B4210 B2A0CF5F DD678C36 0D8B7DE1 A2AB5122 9ED947D5
76CF5BCD D9A2039F D02841B0 7F8BFF97 C080B791 10A9ED41 00FB6F40 95020301
0001

Key label: the_default
Type : RSA General purpose
Size : 512
Created  : 07:46:15 UTC Fri Mar 17 2006
Data :
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00C7DE73 7B3EA447
CCE8F3DF DD1327D8 C1C30C45 2EEB4981 B1B48D2B 1AF14665 178058FB 8F6BB6BB
E08C6163 FA0EE356 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 EXEC modeXR EXEC mode.

**show platform security integrity dossier** [**include** {**packages** | **reboot-history** | **rollback-history** | **running-config** | **system-integrity-snapshot** | **system-inventory**}] [**nonce** *nonce-value*]

## Syntax Description

| | |
|---|---|
| **packages** | Displays active package(s) installed. |
| **reboot-history** | Displays reboot history of the node. |
| **rollback-history** | Displays rollback history of the node. |
| **running-config** | Displays the currently committed running configuration on the node, as displayed by **show running configuration** command. |
| **system-integrity-snapshot** | Displays the system integrity snapshot. |
| **system-inventory** | Displays the system inventory. |
| **nonce** | Specifies the nonce to generate the signature. |
| *nonce-value* | Specifies the nonce value in hexadecimal string format. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

The output of this command is displayed in JSON format.

## Task ID

| Options | Task ID | Operations |
|---|---|---|
| **packages** | pkg-mgmt | read |
| **reboot-history** | system | read |
| **rollback-history** | config-services | read |
| **running-config** | NA (available to all users) | read |
| **system-integrity-snapshot** | basic-services | read |
| **system-inventory** | sysmgr | read |

## Examples

This example shows the usage of **show platform security integrity dossier** command with various selectors:

```
Router#show platform security integrity dossier include packages reboot-history
rollback-history system-integrity-snapshot system-inventory nonce 1580 | utility sign nonce
 1580 include-certificate
```

# 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

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## 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**

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

**Usage Guidelines**

No specific guidelines impact the use of this command.

**Task ID**

| Task ID | Operations |
|---|---|
| crypto | execute |

**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  Software configuration module in the  *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers.*

# 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

EXEC modeXR EXEC mode

## Command History

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

## 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

| Task ID | Operation |
|---|---|
| config-services | read, write |

### Example

This example shows how to use the **clear netconf-yang agent session** command:

```
RP/0/RP0RSP0/CPU0:router (config) #  clear netconf-yang agent session 32125
```

# clear ssh

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

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

## Syntax Description

| | |
|---|---|
| *session-id* | 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

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

Use the **clear ssh** command to disconnect incoming or outgoing SSH connections. Incoming connections are managed by the SSH server running on the local networking device. Outgoing connections are initiated from the local networking device.

To display the session ID for a connection, use the **show ssh** command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |

## 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/RP0RSP0/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


RP/0/RP0RSP0/CPU0:router# clear ssh 0


RP/0/RP0RSP0/CPU0:router# show ssh

SSH version : Cisco-2.0

id  chan pty     location        state           userid    host                 ver
authentication connection type
-------------------------------------------------------------------------------------------------------------
Incoming sessions

Outgoing sessions
1               0/33/1     SESSION_OPEN         cisco       172.19.72.182        v2
2               0/33/1     SESSION_OPEN         cisco       3333::50             v2
```

# disable auth-methods

To selectively disable the authentication methods for the SSH server, use the **disable auth-methods** command in ssh server configuration mode. To remove the configuration, use the **no** form of this command.

**disable auth-methods** { **keyboard-interactive** | **password** | **public-key** }

## Syntax Description

| | |
|---|---|
| **keyboard-interactive** | Disables keyboard-interactive authentication method for the SSH server |
| **password** | Disables password authentication method for the SSH server |
| **public-key** | Disables publick-key authentication method for the SSH server |

## Command Default

Allows all the authentication methods, by default.

## Command Modes

ssh server

## Command History

| Release | Modification |
|---|---|
| Release 7.8.1 | This command was introduced. |

## Usage Guidelines

If this configuration is not present, you can consider that the SSH server on the router allows all the authentication methods.

The public-key authentication method includes certificate-based authentication as well.

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read, write |

This example shows how to disable the public-key authentication method for the SSH server on the router.

```
Router#configure
Router(config)# ssh server
Router(config-ssh)# disable auth-methods public-key
Router(config-ssh)# commit
```

# netconf-yang agent ssh

To enable netconf agent over SSH (Secure Shell) , use the **netconf-yang agent ssh** command in the global configuration mode. To disable netconf, use the **no** form of the command.

**netconf-yang agent ssh**
**no netconf-yang agent ssh**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

Global Configuration

## Command History

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

## Usage Guidelines

SSH is currently the supported transport method for Netconf.

## Task ID

| Task ID | Operation |
|---|---|
| config-services | read, write |

### Example

This example shows how to use the **netconf-yang agent ssh** command:

```
RP/0/RP0RSP0/CPU0:router (config) #  netconf-yang agent ssh
```

# sftp

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

**sftp** [ *username* **@** *host* **:** *remote-filename* ] *source-filename dest-filename* [**source-interface** *type interface-path-id*] [**vrf** *vrf-name*]

## Syntax Description

| | |
|---|---|
| *username* | (Optional) Name of the user performing the file transfer. The at symbol (@) following the username is required. |
| *hostname:remote-filename* | (Optional) Name of the Secure Shell File Transfer Protocol (SFTP) server. The colon (:) following the hostname is required. |
| *source-filename* | SFTP source, including the path. |
| *dest-filename* | SFTP destination, including the path. |
| **source-interface** | (Optional) Specifies the source IP address of a selected interface for all outgoing SSH connections. |
| *type* | Interface type. For more information, use the question mark (**?**) online help function. |
| *interface-path-id* | Physical interface or virtual interface.<br>**Note** Use the **show interfaces** command in EXEC modeXR EXEC mode to see a list of all interfaces currently configured on the router.<br>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

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

SFTP provides for the secure (and authenticated) copying of files between a router and a remote host. Like the **copy** command, the **sftp** command can be invoked only in EXEC modeXR 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.

From Cisco IOS XR Software Release 7.10.1 and later, you can use public-key based user authentication for Cisco IOS XR routers configured as SSH clients as well. This feature thereby allows you to use password-less authentication for secure file transfer and copy operations using SFTP and SCP protocols.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |
| basic-services | execute |

## Examples

In the following example, user *abc* is downloading the file *ssh.diff* from the SFTP server *ena-view1* to *disk0:*

```
RP/0/RP0RSP0/CPU0:router#sftp abc@ena-view1:ssh.diff disk0
```

In the following example, user *abc* is uploading multiple files from disk 0:/sam_* to /users/abc/ on a remote SFTP server called ena-view1:

```
RP/0/RP0RSP0/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/RP0RSP0/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/RP0RSP0/CPU0:router#dir disk0:/V6copy

Directory of disk0:

70144       -rwx  308413      Sun Oct 16 23:06:52 2011  V6copy

2102657024 bytes total (1537638400 bytes free)
```

In the following example, user *admin* is uploading the file *v6copy* from *disk0:* to *disk0a:/v6back* on a local SFTP server using an IPv6 address:

```
RP/0/RP0RSP0/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/RP0RSP0/CPU0:router#dir disk0a:/v6back

Directory of disk0a:

66016       -rwx  308413      Sun Oct 16 23:07:28 2011  v6back

2102788096 bytes total (2098987008 bytes free)
```

In the following example, user *admin* is downloading the file *sampfile* from *disk0:* to *disk0a:/sampfile_v4* on a local SFTP server using an IPv4 address:

```
RP/0/RP0RSP0/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/RP0RSP0/CPU0:router#dir disk0a:/sampfile_v4

Directory of disk0a:

131520      -rwx  986         Tue Oct 18 05:37:00 2011  sampfile_v4

502710272 bytes total (502001664 bytes free)
```

In the following example, user *admin* is uploading the file *sampfile_v4* from *disk0a:* to *disk0:/sampfile_back* on a local SFTP server using an IPv4 address:

```
RP/0/RP0RSP0/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/RP0RSP0/CPU0:router#dir disk0:/sampfile_back

Directory of disk0:

121765      -rwx  986         Tue Oct 18 05:39:00 2011  sampfile_back

524501272 bytes total (512507614 bytes free)
```

# sftp (Interactive Mode)

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

**sftp** [ *username* **@** *host* **:** *remote-filename* ] [**source-interface** *type interface-path-id*] [**vrf** *vrf-name*]

## Syntax Description

| | |
|---|---|
| *username* | (Optional) Name of the user performing the file transfer. The at symbol (@) following the username is required. |
| *hostname:remote-filename* | (Optional) Name of the Secure Shell File Transfer Protocol (SFTP) server. The colon (:) following the hostname is required. |
| **source-interface** | (Optional) Specifies the source IP address of a selected interface for all outgoing SSH connections. |
| *type* | Interface type. For more information, use the question mark (**?**) online help function. |
| *interface-path-id* | Physical interface or virtual interface.<br>**Note** Use the **show interfaces** command in EXEC modeXR EXEC mode to see a list of all interfaces currently configured on the router.<br>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

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

The SFTP client, in the interactive mode, creates a secure SSH channel where the user can enter any supported command. When a user starts the SFTP client in an interactive mode, the SFTP client process creates a secure SSH channel and opens an editor where user can enter any supported command.

More than one request can be sent to the SFTP server to execute the commands. While there is no limit on the number of 'non-acknowledged' or outstanding requests to the server, the server might buffer or queue these requests for convenience. Therefore, there might be a logical sequence to the order of requests.

The following unix based commands are supported in the interactive mode:

- bye

- **cd** <*path*>
- **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

From Cisco IOS XR Software Release 7.10.1 and later, you can use public-key based user authentication for Cisco IOS XR routers configured as SSH clients as well. This feature thereby allows you to use password-less authentication for secure file transfer and copy operations using SFTP and SCP protocols.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | execute |
| basic-services | execute |

## 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/RP0RSP0/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:/frmRouterdownoad

/auto/tftp-server1-users5/admin/frmRouter
  Transferred 1578 Bytes
  1578 bytes copied in 0 sec (27684)bytes/sec
sftp> put /disk0:/frmRouterdownoad againtoServer

/disk0:/frmRouterdownoad
  Transferred 1578 Bytes
  1578 bytes copied in 0 sec (14747)bytes/sec
sftp>
```

In the following example, user *abc* is downloading and uploading a file from/to an external SFTP server using an IPv4 address:

```
RP/0/RP0RSP0/CPU0:router#sftp abc@2.2.2.2

Connecting to 2.2.2.2...
Password:
sftp> pwd
Remote working directory: /
sftp> cd /auto/tftp-server1-users5/abc
sftp> get frmRouter /disk0:/frmRouterdownoad

/auto/tftp-server1-users5/abc/frmRouter
  Transferred 1578 Bytes
  1578 bytes copied in 0 sec (27684)bytes/sec
sftp> put /disk0:/frmRouterdownoad againtoServer

/disk0:/frmRouterdownoad
  Transferred 1578 Bytes
  1578 bytes copied in 0 sec (14747)bytes/sec
sftp>
```

# show 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

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| config-services | read |

### Example

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

```
RP/0/RP0RSP0/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*

| Field name | Description |
|---|---|
| Client session ID | Assigned session identifier |
| NC version | Version of the Netconf client as advertised in the hello message |
| Client connection time | Time elapsed since the client was connected |
| Last OP time | Last operation time |
| Last OP type | Last operation type |
| Lock (yes or no) | To check if the session holds a lock on the configuration datastore |

# show netconf-yang statistics

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

**show netconf-yang statistics**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| config-services | read |

### Example

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

```
RP/0/RP0RSP0/CPU0:router (config) #  sh netconf-yang statistics
Summary statistics
                        # requests|             total time|   min time per request|   max
 time per request|   avg time per request|
other                             0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
close-session                     4|       0h  0m  0s   3ms|       0h  0m  0s   0ms|
 0h  0m  0s   1ms|       0h  0m  0s   0ms|
kill-session                      0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
get-schema                        0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
get                               0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
get-config                        1|       0h  0m  0s   1ms|       0h  0m  0s   1ms|
 0h  0m  0s   1ms|       0h  0m  0s   1ms|
edit-config                       3|       0h  0m  0s   2ms|       0h  0m  0s   0ms|
 0h  0m  0s   1ms|       0h  0m  0s   0ms|
commit                            0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
cancel-commit                     0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
lock                              0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
unlock                            0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
discard-changes                   0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
```

```
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
validate                          0|       0h  0m  0s   0ms|       0h  0m  0s   0ms|
 0h  0m  0s   0ms|       0h  0m  0s   0ms|
xml parse                         8|       0h  0m  0s   4ms|       0h  0m  0s   0ms|
 0h  0m  0s   1ms|       0h  0m  0s   0ms|
netconf processor                 8|       0h  0m  0s   6ms|       0h  0m  0s   0ms|
 0h  0m  0s   1ms|       0h  0m  0s   0ms|
```

***Table 13: Field descriptions***

| Field name | Description |
|---|---|
| Requests | Total number of processed requests of a given type |
| Total time | Total processing time of all requests of a given type |
| Min time per request | Minimum processing time for a request of a given type |
| Max time per request | Maximum processing time for a request of a given type |
| Avg time per request | Average processing time for a request type |

# 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

EXEC modeXR EXEC mode

## Command History

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

## 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.

The connection type field in the command output of **show ssh** command shows as **port-forwarded local** for SSH port-forwarded sessions.

Use the **show ssh server** command to see the details of the SSH server. The **Port Forwarding** column shows as **local** for the port-forwarded session. Whereas, for a regular SSH session, the field displays as **disabled**.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

The following output is applicable for the **show ssh** command starting release 6.0 and later.

```
RP/0/RP0RSP0/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*

| Field | Description |
|---|---|
| session | Session identifier for the incoming and outgoing SSH connections. |
| chan | Channel identifier for incoming (v2) SSH connections. NULL for SSH v1 sessions. |
| pty | pty-id allocated for the incoming session. Null for outgoing SSH connection. |
| location | 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. |
| state | The SSH state that the connection is currently in. |
| userid | Authentication, authorization and accounting (AAA) username used to connect to or from the router. |
| host | IP address of the remote peer. |
| ver | Specifies if the connection type is SSHv1 or SSHv2. |
| authentication | Specifies the type of authentication method chosen by the user. |
| connection type | Specifies which application is performed over this connection (Command-Line-Interface, Remote-Command, Scp, Sftp-Subsystem, or Netconf-Subsystem) |

The following is a sample output of SSH port-forwarded session:

```
Router#show ssh

Wed Oct 14 11:22:05.575 UTC
SSH version : Cisco-2.0

id chan pty location   state        userid host          ver authentication connection type
-----------------------------------------------------------------------------------------------
Incoming sessions
15 1    XXX 0/RP0/CPU0 SESSION_OPEN admin  192.168.122.1 v2  password
port-forwarded-local

Outgoing sessions

Router#
```

The following is a sample output of **show ssh server** command with SSH port forwarding enabled:

```
Router#show ssh server
Tue Sep  7 17:43:22.483 IST
---------------------
SSH Server Parameters
---------------------

Current supported versions := v2
                  SSH port := 22
                  SSH vrfs := vrfname:=default(v4-acl:=, v6-acl:=)
              Netconf Port := 830
              Netconf Vrfs := vrfname:=default(v4-acl:=, v6-acl:=)

 Algorithms
```

```
---------------
        Hostkey Algorithms :=
x509v3-ssh-rsa,ecdsa-sha2-nistp521,ecdsa-sha2-nistp384,ecdsa-sha2-nistp256,rsa-sha2-512,rsa-sha2-256,ssh-rsa,ssh-dsa,ssh-ed25519

   Key-Exchange Algorithms :=
ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group14-sha1
     Encryption Algorithms :=
aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com
            Mac Algorithms := hmac-sha2-512,hmac-sha2-256,hmac-sha1

 Authentication Method Supported
-----------------------------------
                 PublicKey := Yes
                  Password := Yes
      Keyboard-Interactive := Yes
         Certificate Based := Yes

 Others
------------
                      DSCP  := 0
                 Ratelimit  := 600
              Sessionlimit  := 110
                 Rekeytime  := 30
       Server rekeyvolume  := 1024
  TCP window scale factor  := 1
            Backup Server  := Disabled
          Host Trustpoint  :=
          User Trustpoint  := tes,test,x509user
          Port Forwarding  := local
Max Authentication Limit  := 16
     Certificate username  := Common name(CN) User principle name(UPN)
Router#
```

# show ssh history

To display the last hundred SSH connections that were terminated, use the **show ssh history** command in EXEC modeXR EXEC mode.

**show ssh history**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

The following is sample output from the **show ssh history** command to display the last hundred SSH sessions that were teminated:

```
RP/0/RP0RSP0/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
```

```
9        1     vty0     0/RP0/CPU0      root      10.196.98.106         v2  key-intr
Command-Line-Interface
```

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 EXEC modeXR EXEC mode.

**show ssh history details**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read |

## Examples

The following is sample output from the **show ssh history details** command to display the last hundred SSH sessions that were teminated along with the start and end time of the sessions:

```
RP/0/RP0RSP0/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
hmac-sha2-256   14-02-18 16:22:18       14-02-18 16:22:19
4       ecdh-sha2-nistp256      ssh-rsa               aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256   15-02-18 12:17:44       15-02-18 12:17:46
5       ecdh-sha2-nistp256      ssh-rsa               aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256   15-02-18 12:18:16       15-02-18 12:18:17
6       ecdh-sha2-nistp256      ssh-rsa               aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256   15-02-18 14:44:08       15-02-18 14:44:09
7       ecdh-sha2-nistp256      ssh-rsa               aes128-ctr  aes128-ctr  hmac-sha2-256
hmac-sha2-256   15-02-18 14:50:15       15-02-18 14:50:16
8       ecdh-sha2-nistp256      ssh-rsa               aes128-ctr  aes128-ctr  hmac-sha2-256
```

```
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
hmac-sha2-256   15-02-18 15:31:26       15-02-18 15:31:38
```

This table describes the significant fields shown in the display.

***Table 15: Field Descriptions***

| Field | Description |
|---|---|
| session | Session identifier for the incoming and outgoing SSH connections. |
| key-exchange | Key exchange algorithm chosen by both peers to authenticate each other. |
| pubkey | Public key algorithm chosen for key exchange. |
| incipher | Encryption cipher chosen for the receiver traffic. |
| outcipher | Encryption cipher chosen for the transmitter traffic. |
| inmac | Authentication (message digest) algorithm chosen for the receiver traffic. |
| outmac | Authentication (message digest) algorithm chosen for the transmitter traffic. |
| start_time | Start time of the session. |
| end_time | End time of the session. |

# 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**

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

**Usage Guidelines**

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

**Task ID**

| Task ID | Operations |
|---|---|
| crypto | read |

**Examples**

The following sample output is from the **show ssh rekey** command:

```
 # 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***

| Field | Description |
|---|---|
| Rekey Count | Number of times the ssh rekey is generated. |
| TimeToRekey | Time remaining (in minutes) before the ssh rekey is regenerated based on the value set using the **ssh server rekey-time** command. |
| VolumeToRekey | Volume remaining (in megabytes) before the ssh rekey is regenerated based on the value set using the **ssh server rekey-volume** command. |

# 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

EXEC modeXR EXEC mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read |

## 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/RP0RSP0/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 17: show ssh session details Field Descriptions***

| Field | Description |
|---|---|
| session | Session identifier for the incoming and outgoing SSH connections. |
| key-exchange | Key exchange algorithm chosen by both peers to authenticate each other. |

| 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**

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

**Usage Guidelines**

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

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

**Task ID**

| Task ID | Operations |
|---|---|
| crypto | read |

**Examples**

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

```
RP/0/RP0RSP0/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***

| Field | Description |
|---|---|
| PID | Process ID of the SSL application. |
| Method | Protocol version (sslv2, sslv3, sslv23, or tlsv1). |
| Type | SSL client or server. |
| Peer | IP address of the SSL peer. |
| Port | Port number on which the SSL traffic is sent. |

| 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 EXEC modeXR EXEC mode.

**show tech-support ssh**

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

**Command Default** None

**Command Modes** EXEC modeXR EXEC mode

**Command History**

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

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

**Task ID**

| Task ID | Operations |
|---|---|
| crypto | read |

**Examples** The following is sample output from the **show tech-support ssh** command:

```
RP/0/RP0RSP0/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:turin-sec1#
```

The **show tech-support ssh** command collects the output of these CLI:

| Command | Description |
|---|---|
| **show logging** | Displays the contents of the logging buffer. |
| **show context location all** | |
| **show running-config** | Displays the contents of the currently running configuration or a subset of that configuration. |
| **show ip int brief** | Displays brief information about each interface. |

| Command | Description |
|---|---|
| **show ssh** | Displays all incoming and outgoing connections to the router. |
| **show ssh session details** | Displays the details for all the incoming and outgoing SSHv2 connections, to the router. |
| **show ssh rekey** | Displays session rekey details such as session id, session rekey count, time to rekey, data to rekey. |
| **show ssh history** | Displays the last hundred SSH connections that were terminated. |
| **show tty trace info all all** | |
| **show tty trace error all all** | |

# 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 Global Configuration modeXR 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

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| 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

| Command | Description |
|---|---|
| ssh client enable cipher , on page 344 | Enables CBC mode ciphers on the SSH client. |
| ssh server enable cipher, on page 357 | Enables CBC mode ciphers on the SSH server. |

# ssh client auth-method

To set the preferred order of SSH client authentication methods to be negotiated with the SSH server while establishing SSH sessions, use the **ssh client auth-method** command in the Global Configuration modeXR Config mode. To revert to the default order of SSH client authentication methods, use the **no** form of this command.

```
ssh client auth-method list-of-auth-method
```

## Syntax Description

| | |
|---|---|
| *list-of-auth-method* | Specifies the list of SSH client authentication methods in the respective order.<br>The available options are:<br>• **keyboard-interactive**<br>• **password**<br>• **public-key** |

## Command Default

None

## Command Modes

Global ConfigurationXR Config

## Command History

| Release | Modification |
|---|---|
| Release 7.9.2/Release 7.10.1 | This command was introduced. |

## Usage Guidelines

The default order of SSH client authentication methods on Cisco IOS XR routers is as follows:

- On routers running Cisco IOS XR SSH:
  - **public-key**, **password** and **keyboard-interactive**
- On routers running CiscoSSH (open source-based SSH):
  - **public-key**, **keyboard-interactive** and **password**

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read, write |

This example shows how to set the order of SSH client authentication methods in such a way that public key authentication is negotiated first, followed by keyboard-interactive, and then password-based authentication.

```
Router#configure
```

```
Router(config)#ssh client auth-method public-key keyboard-interactive password
Router(config-ssh)#commit
```

# 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 Global Configuration modeXR 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

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

## 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

| Task ID | Operation |
|---|---|
| crypto | read, write |

## 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

| Command | Description |
|---|---|
| ssh algorithms cipher, on page 341 | Configures the list of supported SSH algorithms on the client or on the server. |
| ssh server enable cipher, on page 357 | Enables CBC mode ciphers on the SSH server. |

# 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

Global Configuration modeXR Config mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following sample output is from the **ssh client knownhost** command:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# ssh client knownhost disk0:/ssh.knownhost
RP/0/RP0RSP0/CPU0:router(config)# commit
RP/0/RP0RSP0/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/RP0RSP0/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.

**ssh client source-interface** *type interface-path-id*
**no ssh client source-interface** *type interface-path-id*

## Syntax Description

| | |
|---|---|
| *type* | Interface type. For more information, use the question mark (?) online help function. |
| *interface-path-id* | Physical interface or virtual interface.<br>**Note** Use the **show interfaces** command to see a list of all interfaces currently configured on the router.<br>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

Global Configuration modeXR Config mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to set the IP address of the Management Ethernet interface for all outgoing SSH connections:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# ssh client source-interface MgmtEth 0/RP0/CPU0/0
```

# ssh client vrf

To configure a new VRF for use by the SSH client, use the **ssh client vrf** command. To remove the specified VRF, use the **no** form of this command.

**ssh client vrf** *vrf-name*
**no ssh client vrf** *vrf-name*

## Syntax Description

| | |
|---|---|
| *vrf-name* | Specifies the name of the VRF to be used by the SSH client. |

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

An SSH client can have only one VRF.

If a specific VRF is not configured for the SSH client, the default VRF is assumed when applying other SSH client-related commands, such as ssh client knownhost, on page 346 or ssh client source-interface, on page 347.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows the SSH client being configured to start with the specified VRF:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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 Global Configuration modeXR 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.<br>**Note** This option is available only for the **server**. |

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read, write |

This example shows how to disable SHA1 HMAC cryptographic algorithm on the SSH client:

```
Router#ssh client disable hmac hmac-sha1
```

This example shows how to disable SHA-2 HMAC cryptographic algorithm on the SSH server:

```
Router#ssh server disable hmac hmac-sha2-512
```

# ssh

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. |
| **cipheraes** | (Optional) Specifies Advanced Encryption Standard (AES) as the cipher for the SSH client connection.<br>**Note** If there is no specification of a particular cipher by the administrator, the client proposes 3DES as the default to ensure compatibility. |
| 128-CBC | 128-bit keys in CBC mode. |
| 192-CBC | 192-bit keys in CBC mode. |
| 256-CBC | 256-bit keys in CBC mode. |
| **source interface** | (Optional) Specifies the source IP address of a selected interface for all outgoing SSH connections. |
| *type* | Interface type. For more information, use the question mark (**?**)online help function. |
| *interface-path-id* | Physical interface or virtual interface.<br>**Note** Use the**showinterfaces** command in EXEC modeXR EXEC mode to see a list of all interfaces currently configured on the router.<br>For more information about the syntax for the router, use the question mark(**?**)online help function. |
| command | (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

EXEC modeXR EXEC mode

## Command History

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

## 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 347 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 347 or ssh client knownhost, on page 346 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

| Task ID | Operations |
|---|---|
| crypto | execute |
| basic-services | execute |

## Examples

The following sample output is from the **ssh** command to enable an outbound SSH client connection:

```
RP/0/RP0RSP0/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.

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

## Syntax Description

| | |
|---|---|
| **vrf** *vrf-name* | Specifies the name of the VRF to be used by the SSH server. The maximum VRF length is 32 characters.<br>**Note** If no VRF is specified, the default VRF is assumed. |
| v2 | Forces the SSH server version to be only 2. |

## 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

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

An SSH server must be configured at minimum for one VRF. If you delete all configured VRFs, including the default, the SSH server process stops. If you do not configure a specific VRF for the SSH client when applying other commands, such as **ssh client knownhost** or **ssh client source-interface,** the default VRF is assumed.

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 369 command.

To verify that the SSH server is up and running, use the **show process sshd** command.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

In the following example, the SSH server is brought up to receive connections for VRF “green”:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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 Global Configuration modeXR Config mode. To remove the configuration, use the **no** form of this command.

**ssh server algorithms host-key** { **dsa** | **ecdsa-nistp256** | **ecdsa-nistp384** | **ecdsa-nistp521** | **ed25519** | **rsa** | **x509v3-ssh-rsa** }

## Syntax Description

| | |
|---|---|
| • **dsa**<br>• **ecdsa-nistp256**<br>• **ecdsa-nistp384**<br>• **ecdsa-nistp521**<br>• **ed25519**<br>• **rsa**<br>• **x509v3-ssh-rsa** | Selects the specified host keys to be offered to the SSH client.<br>While configuring this, you can specify the algorithms in any order. |

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.0.12 | This command was introduced. |
| Release 7.3.1 | The support for **ed25519** and **x509v3-ssh-rsa** algorithms was introduced. |

## Usage Guidelines

This configuration is optional. If this configuration is not present, it is considered that all the SSH host-key pairs are configured. In that case, the SSH client is allowed to connect to the SSH sever 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 the automatic generation of SSH host-key pairs, the **show crypto key mypubkey** command output displays key information of all the keys that are auto-generated. Before its introduction, the output of this command displayed key information of only those host-key pairs that were explicitly configured using the **crypto key generate** command.

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read, write |

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

```
Router#ssh server algorithms host-key ecdsa-nistp521
```

Similarly, this example shows how to select the **ed25519** algorithm:

```
Router(config)#ssh server algorithms host-key ed25519
```

Similarly, this example shows how to select the **x509v3-ssh-rsa** algorithm:

```
Router(config)#ssh server algorithms host-key x509v3-ssh-rsa
```

# ssh server certificate

To configure the certificate-related parameters of SSH server, use the **ssh server certificate** command in Global Configuration modeXR Config mode. To remove the configuration, use the **no** form of this command.

**ssh server certificate username** { **common-name** | **user-principle-name** }

## Syntax Description

| | |
|---|---|
| **username** | Specifies which field in the certificate to be used as the username. |
| **common-name** | Configures the user common name (CN) from the subject name field. |
| **user-principle-name** | Configures the user principle name (UPN) from subject alternate name. |

## Command Default

In the absence of this configuration, the SSH server considers common name (CN) as the username.

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.1 | This command was introduced. |

## Usage Guidelines

The user name must match the user name provided in the CLI.

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read, write |

This example shows how to specify which field in the certificate is to be used as the username. Here, it specifies the user common name to be picked up from the subject name field.

```
Router#configure
Router(config)#ssh server certificate username common-name
Router(config)#commit
```

Here, it specifies the user principle name to be picked up from the subject alternate name field.

```
Router#configure
Router(config)#ssh server certificate username user-principle-name
Router(config)#commit
```

# 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 Global Configuration modeXR 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

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

## 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 | Operation |
|---|---|
| crypto | 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

| Command | Description |
|---|---|
| ssh algorithms cipher, on page 341 | Configures the list of supported SSH algorithms on the client or on the server. |
| ssh client enable cipher , on page 344 | 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

Global Configuration modeXR Config mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows the initiation of an SSH server logging:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# ssh server logging
```

# ssh server max-auth-limit

To configure the maximum number of authentication attempts allowed for SSH connection, use the **ssh server max-auth-limit** command in Global Configuration modeXR Config mode. To remove the configuration, use the **no** form of this command.

**ssh server max-auth-limit** *limit*

## Syntax Description

| | |
|---|---|
| *limit* | Specifies the maximum authentication attempts allowed for SSH connection. The limit ranges from 3 to 20; default being 20 (prior to Cisco IOS XR Software Release 7.3.2, the limit range was from 4 to 20). |

## Command Default

The default authentication limit is 20.

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.2 | The command was modified to change the minimum value of limit range from 4 to 3. |
| Release 7.3.1 | This command was introduced |

## Usage Guidelines

The SSH server limits the number of authentication attempts using the password authentication method to a maximum of 3 due to security reasons. You cannot change this particular limit of 3 by configuring the maximum authentication attempts limit for SSH.

For example, even if you configure the maximum authentication attempts limit as 5, the number of authentication attempts allowed using the password authentication method still remain as 3.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

This example shows how to configure the maximum number of authentication attempts allowed for SSH connection:

```
Router# configure
Router(config)# ssh server max-auth-limit 5
Router(config)# commit
```

# ssh server port-forwarding local

To enable SSH port forwarding feature on SSH server, use the **ssh server port-forwarding local** command in Global Configuration modeXR Config mode. To disable the feature, use the **no** form of this command.

**ssh server port-forwarding local**

## Syntax Description

This command has no keywords or arguments.

## Command Default

Disabled, by default.

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.2 | This command was introduced with CiscoSSH, an OpenSSH-based implementation of SSH. |
| Release 7.3.15 | This command was introduced with Cisco IOS XR SSH. |

## Usage Guidelines

The Cisco IOS XR software supports SSH port forwarding only on SSH server; not on SSH client. Hence, to utilize this feature, the SSH client running at the end host must already have the support for SSH port forwarding or tunneling.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

This example shows how to enable SSH port forwarding feature on SSH server:

```
Router#configure
Router(config)#ssh server port-forwarding local
Router(config)#commit
```

## Related Commands

| Command | Description |
|---|---|
| show ssh, on page 327 | Displays all incoming and outgoing SSH connections on the router. |

# ssh server netconf

To configure a port for the netconf SSH server, use the **ssh server netconf port** in the Global Configuration modeXR Config mode. To disable netconf for the configured port, use the **no** form of the command.

**ssh server netconf** [ **port** *port-number* ]
**no ssh server netconf** [ **port** *port-number* ]

## Syntax Description

| | |
|---|---|
| *port-number* | (Optional) Port number for the netconf SSH server (default port number is 830). |

## Command Default

Default port number is 830.

## Command Modes

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read, write |

### Example

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

```
RP/0/RP0RSP0/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

| | |
|---|---|
| **port** *port-number* | Port number for the netconf SSH server (default port number is 830). |

## Command Default

The default port number is 830.

## Command Modes

Global configuration

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# ssh server netconf port 831
```

## Related Commands

| Command | Description |
|---|---|
| ssh server netconf | Configures the vrf(s), where netconf subsystem requests are to be received. |
| netconf-yang agent ssh | Configures the **ssh netconf-yang backend** for the netconf subsystem (Required to allow the system to service netconf-yang requests).<br>For more information, see the *System Management Command Reference for Cisco ASR 9000 Series RoutersSystem Management Command Reference for Cisco 8000 Series Routers*. |

# ssh server rate-limit

To limit the number of incoming Secure Shell (SSH) connection requests allowed per minute, use the **ssh server rate-limit** command. To return to the default value, use the **no** form of this command.

**ssh server rate-limit** *rate-limit*
**no ssh server rate-limit**

## Syntax Description

| | |
|---|---|
| *rate-limit* | Number of incoming SSH connection requests allowed per minute. Range is from 1 to 120.<br>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

Global Configuration modeXR 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to set the limit of incoming SSH connection requests to 20 per minute:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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

| Release | Modification |
|---|---|
| Release 6.2.1 | This command was introduced. |

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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.<br>**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

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

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

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

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/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

Global Configuration modeXR Config mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to set the limit of incoming SSH connections to 50:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# ssh server session-limit 50
```

# ssh server set-dscp-connection-phase

To set the DSCP marking from TCP connection phase itself for SSH packets originating from Cisco IOS XR routers that function as SSH servers, use the **ssh server set-dscp-connection-phase** command in Global Configuration modeXR Config mode. To remove the configuration and to continue marking the SSH packets from the authentication phase, use the **no** form of this command.

**ssh server set-dscp-connection-phase**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 24.1.1 | This command was introduced. |

## Usage Guidelines

- By default, the DSCP marking for the SSH packets originating from Cisco IOS XR routers with CiscoSSH that function as SSH servers is done from the authentication phase. Whereas, for routers with Cisco IOS XR SSH, the DSCP marking for the SSH packets is done from TCP connection phase itself.
- Although the **ssh server set-dscp-connection-phase** command is available on routers with CiscoSSH and routers with Cisco IOS XR SSH, this configuration is relevant only on routers with CiscoSSH due to the above mentioned reason.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

This example shows how to set the DSCP marking from TCP connection phase itself for SSH server packets originating from Cisco IOS XR routers with CiscoSSH:

```
Router#configure
Router(config)#ssh server set-dscp-connection-phase
Router(config-ssh)#commit
```

# ssh server trustpoint

To configure the trustpoint for SSH certificates, use the **ssh server trustpoint** command in Global Configuration modeXR Config mode. To disable this feature, use the **no** form of this command.

**ssh server trustpoint** { **host** | **user** } *trustpoint-name*

## Syntax Description

| | |
|---|---|
| **host** | Configures the trustpoint from where server takes its certificate. |
| **user** | Configures the trustpoints used for user certificate validation. |
| *trustpoint-name* | Specifies the name of the trustpoint. |

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read, write |

This example shows how to configure the trustpoint from where SSH server takes its certificate:

```
Router#configure
Router(config)#ssh server trustpoint host test-host-tp
Router(config)#commit
```

This example shows how to configure the trustpoint used for user certificate validation:

```
Router#configure
Router(config)#ssh server trustpoint user test-user-tp
Router(config)#commit
```

# 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

Global Configuration modeXR Config mode

## Command History

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

## Usage Guidelines

Only SSHv2 client connections are allowed.

## Task ID

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

The following example shows how to initiate the SSH server version to be only SSHv2:

```
 RP/0/RP0RSP0/CPU0:router#configure
RP/0/RP0RSP0/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

Global Configuration modeXR Config mode

## Command History

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

## 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

| Task ID | Operations |
|---|---|
| crypto | read, write |

## Examples

In the following example, the timeout value for AAA user authentication is set to 60 seconds:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# ssh timeout 60
```

# Secure Logging Commands

This module describes the Cisco IOS XR software commands used to configure secure logging on the Cisco 8000 Series Routers over Transport Layer Security (TLS). TLS, the successor of Secure Socket Layer (SSL), is an encryption protocol designed for data security over networks.

For detailed information about secure logging concepts, configuration tasks, and examples, see the *Implementing Secure Logging* module in the *System Security Configuration Guide for Cisco 8000 Series Routers*.

# address

To configure the syslog server settings with IP address, use the **address** command in logging TLS peer configuration mode. To remove the configuration, use the **no** form of this command.

**address** { **IPv4** *ipv4-address* | **IPv6** *ipv6-address* **}**

## Syntax Description

| | |
|---|---|
| *ipv4-address* | IPv4 address in A:B:C:D format. |
| *ipv6-address* | IPv6 address in X:X::X format. |

## Command Default

None

## Command Modes

Logging TLS peer configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 6.2.1 | This command was introduced. |

## Usage Guidelines

You can use the IPv4 or IPv6 address of the server to access the remote syslog server.

## Task ID

| Task ID | Operations |
|---|---|
| logging | Read, Write |

## Examples

The following example shows how to configure syslog server settings with IPv4 address:

```
Router(config)# logging tls-server TEST
Router(config-logging-tls-peer)# severity debugging
Router(config-logging-tls-peer)# trustpoint tp
Router(config-logging-tls-peer)# address ipv4 10.105.230.83
```

## Related Commands

| Command | Description |
|---|---|
| logging tls-server, on page 373 | Configures syslog over TLS server. |
| trustpoint , on page 376 | Configures the trustpoint for the TLS server. |

# logging tls-server

To configure System Logging over Transport Layer Security (TLS) server, use the **logging tls-server** command in Global Configuration mode. To remove the configuration, use the **no** form of this command.

**logging tls-server** *tls-name*

## Syntax Description

| | |
|---|---|
| *tls-name* | User-defined name for the TLS server. |

## Command Default

None

## Command Modes

Global configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 6.2.1 | This command was introduced. |

## Usage Guidelines

This command enters the logging TLS peer configuration mode, where you can configure the settings to access the remote syslog server.

## Task ID

| Task ID | Operation |
|---|---|
| logging | read, write |

This example shows how to configure a TLS server that enters the logging TLS peer configuration mode:

```
Router#Configure
Router(config)# logging tls-server TEST
Router(config-logging-tls-peer)#
```

# tls-hostname

To configure the syslog server settings with hostname or FQDN of the secure log server, use the **tls-hostname** command in logging TLS peer configuration mode. To remove the configuration, use the **no** form of this command.

**tls-hostname** *hostname*

## Syntax Description

| | |
|---|---|
| *hostname* | Name of the logging host. |

## Command Default

None

## Command Modes

Logging TLS peer configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 6.2.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| logging | Read, Write |

## Examples

The following example shows how to configure syslog server settings with server hostname:

```
Router(config)# logging tls-server TEST
Router(config-logging-tls-peer)# severity debugging
Router(config-logging-tls-peer)# trustpoint tp
Router(config-logging-tls-peer)# tls-hostname xyz.cisco.com
```

## Related Commands

| Command | Description |
|---|---|
| logging tls-server, on page 373 | Configures syslog over TLS server. |
| trustpoint , on page 376 | Configures the trustpoint for the TLS server. |

# tlsv1-disable

To disable Transport Layer Security (TLS) version 1.0, use the **tlsv1-disable** command in Global Configuration modeXR Config mode.

**tlsv1-disable**

## Syntax Description

This command has no keywords or arguments.

## Command Default

None

## Command Modes

Global Configuration modeXR Config mode

## Command History

| Release | Modification |
|---|---|
| Release 7.9.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| system | Read, Write |

## Examples

The following example shows how to disable TLS version 1.0:

```
Router(config)# grpc tlsv1-disable
```

# trustpoint

To configure syslog server settings with a trustpoint for the TLS server, use the **trustpoint** command in logging TLS peer configuration mode. To remove the configuration, use the **no** form of this command.

**trustpoint** *trustpoint-name*

## Syntax Description

| | |
|---|---|
| *trustpoint-name* | Name of the configured trustpoint . |

## Command Default

None

## Command Modes

Logging TLS peer configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 6.2.1 | This command was introduced. |

## Usage Guidelines

Ensure that you have already configured the trustpoint name, using the **crypto ca trustpoint** command.

## Task ID

| Task ID | Operations |
|---|---|
| logging | Read, Write |

## Examples

The following example shows how to configure syslog server settings with trustpoint:

```
Router(config)# logging tls-server TEST
Router(config-logging-tls-peer)# severity debugging
Router(config-logging-tls-peer)# trustpoint tp
```

## Related Commands

| Command | Description |
|---|---|
| logging tls-server, on page 373 | Configures syslog over TLS server. |

# vrf

To configure the VRF option for the TLS server, use the **vrf** command in logging TLS peer configuration mode. To remove the configuration, use the **no** form of this command.

**vrf** *vrf-name*

## Syntax Description

| | |
|---|---|
| *vrf-name* | VPN Routing/Forwarding instance name. |

## Command Default

None

## Command Modes

Logging TLS peer configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 6.2.1 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| logging | Read, Write |

## Examples

The following example shows how to configure a VRF instance:

```
Router(config)# logging tls-server TEST
Router(config-logging-tls-peer)# vrf vrftest
```

## Related Commands

| Command | Description |
|---|---|
| logging tls-server, on page 373 | Configures syslog over TLS server. |

# 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 ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# 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

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

## 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.

You must configure the session with a FIPS-approved cryptographic algorithm. A session configured with non-approved cryptographic algorithm for FIPS (such as, **MD5** and **HMAC-MD5**) does not work. This is applicable from Cisco IOS XR Software Release 7.2.1 and later, for OSPF, BGP, RSVP, ISIS, or any application using key chain with non-approved cryptographic algorithm, and only for FIPS mode (that is, when **crypto fips-mode** is configured).

## Task ID

| Task ID | Operation |
|---|---|
| crypto | read, write |

### Example

This example shows how to configure FIPS:

```
RP/0/RP0RSP0/CPU0:router# configure
RP/0/RP0RSP0/CPU0:router(config)# crypto fips-mode
```

# 802.1X and Port Control Commands

This module describes the 802.1X and port control commands.

For detailed information about port control using MAC Authentication Bypass (MAB), the related configuration tasks, and examples, see the *Implementing MAC Authentication Bypass* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers*.

# authenticator

To configure authenticator parameters and to enter the authenticator configuration sub mode, use the **authenticator** command in dot1x profile configuration sub mode. To remove this configuration, use the **no** form of this command.

**authenticator** { **eap profile** *profile-name* | **host-mode** { **multi-auth** | **multi-host** | **single-host** } | **server dead action** { **auth-fail** | **auth-retry** } | **timer** { **mab-retry-time** *retry-timer-value* | **reauth-time** { *reauth-timer-value* | **server** } } }

## Syntax Description

| | |
|---|---|
| **eap** | Enables local Extensible Authentication Protocol (EAP) server for MACSec. |
| *profile-name* | Specifies the EAP profile name, in WORD. |
| **host-mode** | Sets the host mode for authentication.<br>**Note** Only **single-host** mode is supported. |
| **server dead action** | Sets the action to be taken when the remote AAA server is unreachable. You can set it as either to retry the authentication or to consider it as authentication failure. |
| **timer** | Sets various timers for authentication. |
| **mab-retry-time** | Sets the interval, in seconds, after which the router re-initiates an authentication attempt for the MAC authentication bypass (MAB) clients, in scenarios where previous authentication failed or if the RADIUS server was unreachable.<br>Range is 60 to 300, default being 60. |
| **reauth-time** | Sets the interval, in seconds, after which the router automatically initiates re-authentication process with the RADIUS server.<br>Range is 60 to 5184000 (2 months). |
| **server** | Sets the re-authentication interval on the router as per the value specified by the RADIUS server.<br>Minimum expected value is 60 seconds, default being 1 hour. |

## Command Default

None

## Command Modes

Dot1x profile configuration mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.4<br>Release 7.5.2 | This command was modified to include the **mab-retry-time** timer option as part of the MAB feature. |
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operations |
|---|---|
| config-services | read, write |

## Examples

This example shows how to set the authenticator mode as **single-host**:

```
Router# configure
Router(config)# dot1x profile test_profile
Router(config-dot1x-test_profile)# authenticator host-mode single-host
Router(config-dot1x-test_profile)# commit
```

This example shows how to set the authenticator retry timer for MAB clients:

```
Router#configure
Router(config)#dot1x profile test_mab
Router(dot1xx-test_mab)#authenticator timer mab-retry-time 60
Router(dot1xx-test_mab)#commit
```

## Related Commands

| Command | Description |
|---|---|
| dot1x profile, on page 385 | Configures IEEE 802.1X profile parameters and enters dot1x profile configuration sub mode. |

# clear mab

To clear the MAC authentication bypass (MAB) session or statistics, use the **clear mab** command in the EXEC modeXR EXEC mode.

**clear mab** { **session** *intf-type if-name* [ **client** *mac-address* ] | **statistics** { **interface** *intf-type if-name* | **location** *node* } }

## Syntax Description

| | |
|---|---|
| **session** | Clears MAB session related to a specific interface. |
| **statistics** | Clears MAB statistics |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.4 | This command was introduced. |
| Release 7.5.2 | |

## Usage Guidelines

No specific guidelines impact the use of this command.

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

The following example shows how to clear MAB statistics on an interface:

```
Router#clear mab statistics interface gigabitEthernet 0/0/0/0
```

# dot1x profile

To configure IEEE 802.1X profile parameters and to enter dot1x profile configuration sub mode, use the **dot1x profile** command in Global Configuration modeXR Config mode. To remove this configuration, use the **no** form of this command.

**dot1x profile** *profile-name* { **authenticator** | **mab** | **pae** { **authenticator** | **both** | **supplicant** } | **supplicant eap** [ **profile** *profile-name* ] }

## Syntax Description

| | |
|---|---|
| *profile-name* | Specifies the dot1x profile name, in WORD, with a maximum of 63 characters. |
| **authenticator** | Enters the sub mode for authenticator. |
| **mab** | Enables MAC authentication bypass (MAB) feature. |
| **pae** | Sets 802.1X PAE type |
| **supplicant** | Enters the sub mode for supplicant. |
| **eap** | Configures EAP supplicant parameters. |

## Command Default

None

## Command Modes

Global ConfigurationXR Config

## Command History

| Release | Modification |
|---|---|
| Release 7.3.4<br>Release 7.5.2 | This command was modified to include the **mab** option as part of MAC authentication bypass (MAB) feature. |
| Release 7.0.12 | This command was introduced. |

## Usage Guidelines

Prior to the introduction of MAB feature, the dot1x configuration in these routers was only a key-provider for MACSec functionality, and not a mechanism for port control on the router.

See the *MACSec Using EAP-TLS Authentication* chapter and the *Implementing Port Control* chapter in the *System Security Configuration Guide for Cisco ASR 9000 Series RoutersSystem Security Configuration Guide for Cisco 8000 Series Routers* respectively, for details of 802.1X profile and MAB feature.

## Task ID

| Task ID | Operations |
|---|---|
| config-services | read, write |

## Examples

This example shows how to configure 802.1X profile on the router:

```
Router# configure
```

```
Router(config)# dot1x profile test_profile
Router(config-dot1x-test_profile)# pae both
Router(config-dot1x-test_profile)# authenticator timer reauth-time 3600
Router(config-dot1x-test_profile)# supplicant eap profile test-eap-profile
Router(config-dot1x-test_profile)# commit
```

This example shows how to enable MAB feature to implement port controlling:

```
Router#configure
Router(config)#dot1x profile test_mab
Router(dot1xx-test_mab)#mab
Router(dot1xx-test_mab)#commit
```

## Related Commands

| Command | Description |
|---|---|
| authenticator, on page 382 | Configures authenticator parameters and enters the authenticator configuration sub mode. |

# show mab

To display the MAC authentication bypass (MAB) feature status of the client, use the **show mab** command in the EXEC modeXR EXEC mode.

**show mab** { **detail** [ **location** *node* ] | **interface** *intf-type if-name* [**detail**] | **statistics** { **interface** *intf-type if-name* | **location** *node* } | **summary** [ **location** *node* ] }

## Syntax Description

| | |
|---|---|
| **detail** | Displays detailed MAB information. |
| **interface** | Displays MAB information of the interface. |
| **statistics** | Displays MAB statistics |
| **summary** | Displays summary of the MAB information. |

## Command Default

None

## Command Modes

EXEC modeXR EXEC mode

## Command History

| Release | Modification |
|---|---|
| Release 7.3.4 | This command was introduced. |
| Release 7.5.2 | |

## Usage Guidelines

Based on the client authorization status, the **show mab** command output displays one of these values in the authorization status field:

- Authorizing
- Authorized
- Authorized (Server unreachable)
- Authorized (Server send fail)
- Unauthorized (Server Reject)
- Unauthorized (Server unreachable)
- Unauthorized (Server send fail)

## Task ID

| Task ID | Operation |
|---|---|
| interface | read |

The following examples show how to verify client MAB information at various levels:

```
Router#show mab summary
Fri Apr  1 16:37:32.340 IST

NODE: node0_0_CPU0
=====================================================================
   Interface-Name        Client          Status
=====================================================================
     Gi0/0/0/0         1122.3344.5566   Authorized

Router#


Router#show mab detail
Fri Apr  1 16:37:37.140 IST

NODE: node0_0_CPU0

MAB info for GigabitEthernet0/0/0/0
---------------------------------------------------------------
InterfaceName       : Gi0/0/0/0
InterfaceHandle     : 0x00000060
HostMode            : single-host
PortControl         : Enabled
PuntState           : Stop Success
PuntSummary         : Punt disabled
Client:
  MAC Address       : 1122.3344.5566
  Status            : Authorized
  SM State          : Terminate
  ReauthTimeout     : 60s, Remaining 0 day(s), 00:00:46
  RetryTimeout      : 60s, timer not started yet
  AuthMethod        : PAP (remote)
  LastAuthTime      : 2022 Apr 01 16:37:23.634
  ProgrammingStatus : Add Success

Router#


Router#show mab interface gigabitEthernet 0/0/0/0 detail
Fri Apr  1 16:38:31.543 IST
MAB info for GigabitEthernet0/0/0/0
---------------------------------------------------------------
InterfaceName       : Gi0/0/0/0
InterfaceHandle     : 0x00000060
HostMode            : single-host
PortControl         : Enabled
PuntState           : Stop Success
PuntSummary         : Punt disabled
Client:
  MAC Address       : 1122.3344.5566
  Status            : Authorized
  SM State          : Terminate
  ReauthTimeout     : 60s, Remaining 0 day(s), 00:00:51
  RetryTimeout      : 60s, timer not started yet
  AuthMethod        : PAP (remote)
  LastAuthTime      : 2022 Apr 01 16:38:23.640
  ProgrammingStatus : Add Success
Router#


Router#show mab statistics interface gigabitEthernet 0/0/0/0
Fri Apr  1 16:41:23.011 IST
InterfaceName         : GigabitEthernet0/0/0/0
```

```
------------------------------------------------------------------
MAC Learning:
  RxTotal             : 0
  RxNoSrcMac          : 0
  RxNoIdb             : 0
Port Control:
  EnableSuccess       : 1
  EnableFail          : 0
  UpdateSuccess       : 0
  UpdateFail          : 0
  PuntStartSuccess    : 0
  PuntStartFail       : 0
  PuntStopSuccess     : 1
  PuntStopFail        : 0
  AddClientSuccess    : 1
  AddClientFail       : 0
  RemoveClientSuccess : 0
  RemoveClientFail    : 0
Client                :
  MAC Address         : 1122.3344.5566
  Authentication:
    Success           : 1406
    Fail              : 0
    Timeout           : 0
    AAA Unreachable   : 0
Router#
```