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. The range for the number of days of the month is from 1 to 31. The range for the years is from 1993 to 2035.
duration *duration value*	(Optional) Determines the lifetime of the key in seconds. The range is from 1-2147483646.
infinite	(Optional) Specifies that the key never expires after it becomes valid.
end-time	(Optional) Time, in *hh:mm:ss day month year* format, after which the key expires. The range is from 0:0:0 to 23:59:59.

Command Default

None

Command Modes

Key configuration

Command History

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 13	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: • Plain-text key strings—Minimum of 1 character and a maximum of 32. • Encrypted key strings—Minimum of 4 characters and no maximum.

Command Default

The default value is clear.

Command Modes

Keychain-key configuration

Command History

Release	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. The range for the number of days of the month to start is from 1 to 31. The range for the years is from 1993 to 2035.
duration *duration value*	(Optional) Determines the lifetime of the key in seconds.
infinite	(Optional) Specifies that the key never expires once it becomes valid.
end-time	(Optional) Time, in *hh:mm:ss day month year* format, after which the key expires. The range is from 0:0:0 to 23:59:59

Command Default

No default behavior or values

Command Modes

Keychain-key configuration

Command History

Release	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
```