Contents

Keychain commands

accept-lifetime utc

Use **accept-lifetime utc** to set the receiving lifetime for a key of a keychain in absolute time mode.

Use **undo accept-lifetime** to restore the default.

Syntax

accept-lifetime utc *start-time start-date* { **duration** { *duration-value* | **infinite** } | **to** *end-time end-date* }

undo accept-lifetime

Default

The receiving lifetime is not configured for a key of a keychain.

Views

Key view

Predefined user roles

network-admin

Parameters

start-time: Specifies the start time in the HH:MM:SS format. The value range for this argument is 0:0:0 to 23:59:59.

start-date: Specifies the start date in the MM/DD/YYYY or YYYY/MM/DD format. The value range for YYYY is 2000 to 2035.

duration *duration-value*: Specifies the lifetime of the key, in the range of 1 to 2147483646 seconds.

duration infinite: Specifies that the key never expires after it becomes valid.

to: Specifies the end time and date.

end-time: Specifies the end time in the HH:MM:SS format. The value range for this argument is 0:0:0 to 23:59:59.

end-date: Specifies the end date in the MM/DD/YYYY or YYYY/MM/DD format. The value range for YYYY is 2000 to 2035.

Usage guidelines

A key becomes a valid accept key when the following requirements are met:

- A key string has been configured.
- An authentication algorithm has been specified.
- The system time is within the specified receiving lifetime.

If an application receives a packet that carries a key ID, and the key is valid, the application uses the key to authenticate the packet. If the key is not valid, packet authentication fails.

If the received packet does not carry a key ID, the application uses all valid keys in the keychain to authenticate the packet. If the packet does not pass any authentication, packet authentication fails.

An application can use multiple valid keys to authenticate packets received from a peer.

Examples

# Set the receiving lifetime for key 1 of keychain **abc** in absolute time mode.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] key 1
[Sysname-keychain-abc-key-1] accept-lifetime utc 12:30 2015/1/21 to 18:30 2015/1/21
```

accept-tolerance

Use **accept-tolerance** to set a tolerance time for accept keys in a keychain.

Use **undo accept-tolerance** to restore the default.

Syntax

accept-tolerance { *value* | **infinite** }

undo accept-tolerance

Default

No tolerance time is configured for accept keys in a keychain.

Views

Keychain view

Predefined user roles

network-admin

Parameters

value: Specifies a tolerance time in the range of 1 to 8640000 seconds.

infinite: Specifies that the accept keys never expires.

Usage guidelines

After a tolerance time is configured, the start time and the end time configured in the **accept-lifetime utc** command are extended for the period of the tolerance time.

If authentication information is changed, information mismatch occurs on the local and peer devices, and the service might be interrupted. Use this command to ensure continuous packet authentication.

Examples

# Set the tolerance time to 100 seconds for accept keys in keychain **abc**.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] accept-tolerance 100
```

# Configure the accept keys in keychain **abc** to never expire.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] accept-tolerance infinite
```

authentication-algorithm

Use **authentication-algorithm** to specify an authentication algorithm for a key.

Use **undo authentication-algorithm** to restore the default.

Syntax

authentication-algorithm{ **hmac-md5** | **hmac-sha-256** | **md5** }

undo authentication-algorithm

Default

No authentication algorithm is specified for a key.

Views

Key view

Predefined user roles

network-admin

Parameters

hmac-md5: Specifies the HMAC-MD5 authentication algorithm.

hmac-sha-256: Specifies the HMAC-SHA-256 authentication algorithm.

md5: Specifies the MD5 authentication algorithm.

Usage guidelines

If an application does not support the authentication algorithm specified for a key, the application cannot use the key for packet authentication.

Examples

# Specify the MD5 authentication algorithm for key 1 of keychain **abc** in absolute time mode.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] key 1
[Sysname-keychain-abc-key-1] authentication-algorithm md5
```

default-send-key

Use **default-send-key** to specify a key in a keychain as the default send key.

Use **undo default-send-key** to restore the default.

Syntax

default-send-key

undo default-send-key

Default

No key in a keychain is specified as the default send key.

Views

Key view

Predefined user roles

network-admin

Usage guidelines

When send keys in a keychain are inactive, the default send key can be used for packet authentication.

A keychain can have only one default send key. The default send key must be configured with an authentication algorithm and a key string.

Examples

# Specify key **1** in keychain **abc** as the default send key.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] key 1
[Sysname-keychain-abc-key-1] default-send-key
```

display keychain

Use **display keychain** to display keychain information.

Syntax

display keychain [**name** *keychain-name* [**key** *key-id*]]

Views

Any view

Predefined user roles

network-admin

network-operator

Parameters

name *keychain-name*: Specifies a keychain by its name, a case-sensitive string of 1 to 63 characters. If you do not specify a keychain, this command displays information about all keychains.

key *key-id*: Specifies a key by its ID in the range of 0 to 281474976710655. If you do not specify a key, this command displays information about all keys in a keychain.

Examples

# Display information about all keychains.

```
<Sysname> display keychain

 Keychain name          : abc
   Mode                 : absolute
   Accept tolerance     : 0
   TCP kind value       : 254
   TCP algorithm value
     HMAC-MD5           : 5
     MD5                : 3
   Default send key ID  : 2 (Inactive)
   Active send key ID   : 1
   Active accept key IDs: 1 2

   Key ID               : 1
     Key string         : $c$3$vuJpEX3Lah7xcSR2uqmrTK2IZQJZguJh3g==
     Algorithm          : md5
     Send lifetime      : 01:00:00 2015/01/22 to 01:00:00 2015/01/25
     Send status        : Active
     Accept lifetime    : 01:00:00 2015/01/22 to 01:00:00 2015/01/27
     Accept status      : Active
```

```
  Key ID               : 2
    Key string         : $c$3$vuJpEX3Lah7xcSR2uqmrTK2IZQJZguJh3g==
    Algorithm          : md5
    Send lifetime      : 01:00:01 2015/01/25 to 01:00:00 2015/01/27
    Send status        : Inactive
    Accept lifetime    : 01:00:00 2015/01/22 to 01:00:00 2015/01/27
    Accept status      : Active
```

Table 1 Command output

Field	Description
Mode	Time mode for the keychain.
Accept tolerance	Tolerance time (in seconds) for accept keys of the keychain.
TCP kind value	Value for the TCP kind field.
TCP algorithm value	ID of the TCP authentication algorithm.
Default send key ID	ID of the default send key. The status for the key is displayed in parentheses.
Key string	Key string in encrypted form.
Algorithm	Authentication algorithm for the key: • **hmac-md5** • **hmac-sha-256** • **md5**
Send lifetime	Sending lifetime for the key.
Send status	Status of the send key: **Active** or **Inactive**.
Accept lifetime	Receiving lifetime for the key.
Accept status	Status of the accept key: **Active** or **Inactive**.

key

Use **key** to create a key for a keychain and enter its view, or enter the view of an existing key.

Use **undo key** to delete a key and all its configurations for a keychain.

Syntax

key *key-id*

undo key *key-id*

Default

No keys exist.

Views

Keychain view

Predefined user roles

network-admin

Parameters

key-id: Specifies a key ID in the range of 0 to 281474976710655.

Usage guidelines

The keys in a keychain must have different key IDs.

Examples

# Create key 1 and enter its view.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] key 1
[Sysname-keychain-abc-key-1]
```

keychain

Use **keychain** to create a keychain and enter its view, or enter the view of an existing keychain.

Use **undo keychain** to delete a keychain and all its configurations.

Syntax

keychain *keychain-name* [**mode absolute**]

undo keychain *keychain-name*

Default

No keychains exist.

Views

System view

Predefined user roles

network-admin

Parameters

keychain-name: Specifies a keychain name, a case-sensitive string of 1 to 63 characters.

mode: Specifies a time mode.

absolute: Specifies the absolute time mode. In this mode, each time point during a key's lifetime is the UTC time and is not affected by the system's time zone or daylight saving time.

Usage guidelines

You must specify the time mode when you create a keychain. You cannot change the time mode for an existing keychain.

The time mode is not required when you enter the view of an existing keychain.

Examples

# Create keychain **abc**, specify the absolute time mode for it, and enter keychain view.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc]
```

key-string

Use **key-string** to configure a key string for a key.

Use **undo key-string** to restore the default.

Syntax

key-string { **cipher** | **plain** } *string*

undo key-string

Default

No key string is configured for a key.

Views

Key view

Predefined user roles

network-admin

Parameters

cipher: Specifies a key in encrypted form.

plain: Specifies a key in plaintext form. For security purposes, the key specified in plaintext form will be stored in encrypted form.

string: Specifies the key. Its plaintext form is a case-sensitive string of 1 to 255 characters. Its encrypted form is a case-sensitive string of 33 o 373 characters.

Usage guidelines

If the length of a plaintext key exceeds the length limit supported by an application, the application uses the supported length of the key to authenticate packets.

Examples

# Set the key string to **123456** in plaintext form for key 1.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] key 1
[Sysname-keychain-abc-key-1] key-string plain 123456
```

send-lifetime utc

Use **send-lifetime utc** to set the sending lifetime for a key of a keychain in absolute time mode.

Use **undo send-lifetime** to restore the default.

Syntax

send-lifetime utc *start-time start-date* { **duration** { *duration-value* | **infinite** } | **to** *end-time end-date* }

undo send-lifetime

Default

The sending lifetime is not configured for a key of a keychain.

Views

Key view

Predefined user roles

network-admin

Parameters

start-time: Specifies the start time in the HH:MM:SS format. The value range for this argument is 0:0:0 to 23:59:59.

start-date: Specifies the start date in the MM/DD/YYYY or YYYY/MM/DD format. The value range for YYYY is 2000 to 2035.

duration *duration-value*: Specifies the lifetime of the key, in the range of 1 to 2147483646 seconds.

duration infinite: Specifies that the key never expires after it becomes valid.

to: Specifies the end time and date.

end-time: Specifies the end time in the HH:MM:SS format. The value range for this argument is 0:0:0 to 23:59:59.

end-date: Specifies the end date in the MM/DD/YYYY or YYYY/MM/DD format. The value range for YYYY is 2000 to 2035.

Usage guidelines

A key becomes a valid send key when the following requirements are met:

- A key string has been configured.
- An authentication algorithm has been specified.
- The system time is within the specified sending lifetime.

To make sure only one key in a keychain is used at a time to authenticate packets to a peer, set non-overlapping sending lifetimes for the keys in the keychain.

Examples

# Set the sending lifetime for key 1 of keychain **abc** in absolute time mode.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] key 1
[Sysname-keychain-abc-key-1] send-lifetime utc 12:30 2015/1/21 to 18:30 2015/1/21
```

tcp-algorithm-id

Use **tcp-algorithm-id** to set an algorithm ID for a TCP authentication algorithm.

Use **undo tcp-algorithm-id** to restore the default.

Syntax

tcp-algorithm-id { **hmac-md5** | **md5**} *algorithm-id*

undo tcp-algorithm-id { **hmac-md5** | **md5** }

Default

The algorithm ID is 3 for the MD5 authentication algorithm and 5 for the HMAC-MD5 authentication algorithm.

Views

Keychain view

Predefined user roles

network-admin

Parameters

hmac-md5: Specifies the HMAC-MD5 authentication algorithm, which provides a key length of 16 bytes.

md5: Specifies the MD5 authentication algorithm, which provides a key length of 16 bytes.

algorithm-id: Specifies an algorithm ID in the range of 1 to 63.

Usage guidelines

If an application uses keychain authentication during TCP connection establishment, the incoming and outgoing TCP packets will carry the TCP Enhanced Authentication Option. The *algorithm-id* field in the option represents the authentication algorithm ID. The algorithm IDs are not assigned by IANA. They are vendor-specific.

To communicate with a peer device from another vendor, the local device must have the same algorithm ID as the peer device. For example, if the algorithm ID is 3 for the HMAC-MD5 algorithm on the peer device, you must execute the **tcp-algorithm-id hmac-md5** 3 command on the local device.

Examples

# Create keychain **abc** and set the algorithm ID to 1 for the HMAC-MD5 authentication algorithm.

```
<Sysname> system-view
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] tcp-algorithm-id hmac-md5 1
```

tcp-kind

Use **tcp-kind** to set the kind value in the TCP Enhanced Authentication Option.

Use **undo tcp-kind** to restore the default.

Syntax

tcp-kind *kind-value*

undo tcp-kind

Default

The kind value is 254 in the TCP Enhanced Authentication Option.

Views

Keychain view

Predefined user roles

network-admin

Parameters

kind-value: Specifies the kind value in the range of 28 to 255. The default is 254.

Usage guidelines

If an application uses keychain authentication during TCP connection establishment, the incoming and outgoing TCP packets will carry the TCP Enhanced Authentication Option. For a successful packet authentication, the local device and the peer device must have the same kind value setting in the TCP Enhanced Authentication Option.

Examples

# Set the kind value to 252 for keys in keychain **abc** in absolute time mode.

```
<Sysname> system-view
```

```
[Sysname] keychain abc mode absolute
[Sysname-keychain-abc] tcp-kind 252
```