16

Key Management System

Contents

Overview

The switches covered in this guide provide support for advanced routing capabilities. Security turns out to be extremely important as complex networks and the internet grow and become a part of our daily life and business. This fact forces protocol developers to improve security mechanisms employed by their protocols, which in turn becomes an extra burden for system administrators who have to set up and maintain them. One possible solution to the problem is to centralize the mechanisms used to configure and maintain security information for all routing protocols. The Key Management System (KMS) can carry this burden.

KMS is designed to configure and maintain key chains. A key chain is a set of keys with a timing mechanism for activating and deactivating individual keys. KMS provides specific instances of routing protocols with one or more Send or Accept keys that must be active at the time of a request. A *protocol instance* is usually an interface on which the protocol is running.

Feature	Default	Menu	CLI	Web
Generating a Key Chain	n/a	n/a	page 16-3	n/a
Generating a Time-Independent key	n/a	n/a	page 16-4	n/a
Generating a Time-Dependent key	n/a	n/a	page 16-5	n/a

Terminology

- **Key Chain:** A key or set of keys assigned for use by KMS-enabled protocols. A key chain may optionally contain the time to activate and deactivate a particular key.
- **Time-Independent Key:** A key that has no activate or deactivate time associated with it. This type of key does not expire, which eliminates the need for a key chain.
- **Time-Dependent key:** a key that has an activate and deactivate time associated with the Accept and Send processes. Time-Dependent keys expire, which means a key chain is needed to keep the assigned protocols supplied with keys.
- **Key Management System (KMS) Enabled Protocol:** A protocol that uses KMS to store authentication key information.

Configuring Key Chain Management

KMS-Related CLI Commands in This Section	Page
show key-chain < *chain_name* >	page 16-3
[no] key-chain *chain_name*	page 16-3
[no] key-chain *chain_name* key *Key_ID*	page 16-4

The Key Management System (KMS) has three configuration steps:

1. Create a key chain entry.
2. Assign a time-independent key or set of time-dependent keys to the Key Chain entry. The choice of key type is based on the level of security required for the protocol to which the key entry will be assigned.
3. Assign the key chain to a KMS-enabled protocol.

This procedure is protocol-dependent. For information on a specific protocol, refer to the chapter covering that protocol in the *Management and Configuration Guide* for your switch.

Creating and Deleting Key Chain Entries

To use the Key Management System (KMS), you must create one or more key chain entries. An entry can be the pointer to a single time-independent key or a chain of time-dependent keys.

Note The key chain information is copied to the standby management module (if redundancy is enabled and the standby module has passed self-test).

.

Syntax: [no] key-chain < *chain_name* >

Generate or delete a key chain entry. Using the optional **no** *form of the command deletes the key chain. The* < *chain_name* > *parameter can include up to 32 characters.*

show key-chain

Displays the current key chains on the switch and their overall status.

For example, to generate a new key chain entry:

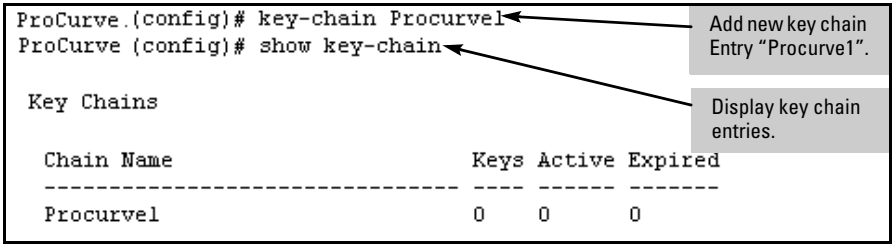

Figure 16-1. Adding a New Key Chain Entry

After you add an entry, you can assign key(s) to it for use by a KMS-enabled protocol.

Assigning a Time-Independent Key to a Chain

A time-independent key has no Accept or Send time constraints. It is valid from boot-up until you change it. If you use a time-independent key, then it is the only key needed for a key chain entry.

Syntax: [no] key-chain < *chain_name* > key < *key_id* >

Generates or deletes a key in the key chain entry < chain_name >. Using the optional **no** *form of the command deletes the key. The < key_id > is any number from 0-255.*

[key-string < *key_str* >]

This option lets you specify the key value for the protocol using the key. The < key_str > can be any string of up to 14 characters in length.

[accept-lifetime infinite] [send-lifetime infinite]

accept-lifetime infinite: *Allows packets with this key to be accepted at any time from boot-up until the key is removed.*

send-lifetime infinite: *Allows the switch to send this key as authorization, from boot-up until the key is removed.*

show key-chain < *chain_name* >

Displays the detail information about the keys used in the key chain named < chain_name >.

For example, to generate a new time-independent key for the Procurve1 key chain entry:

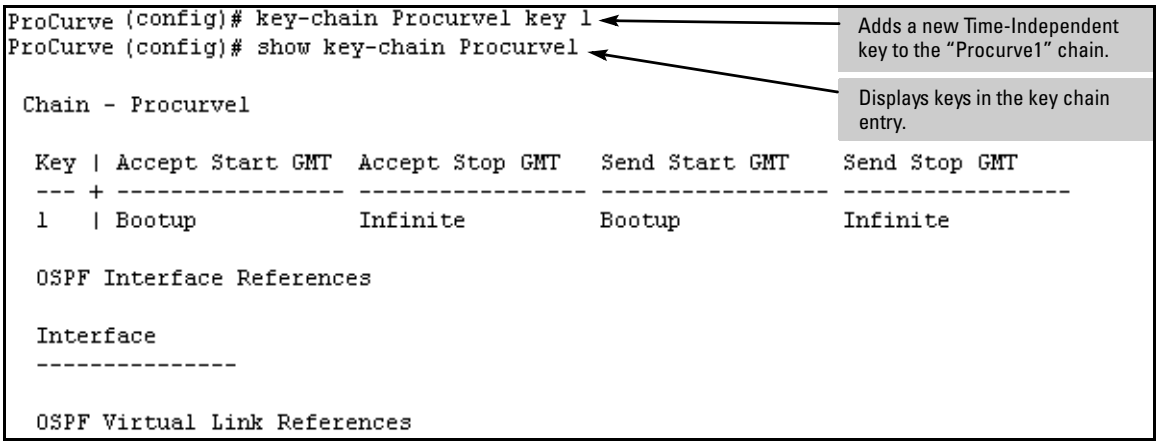

Figure 16-2. Example of Adding and Displaying a Time-Independent Key to a Key Chain Entry

Assigning Time-Dependent Keys to a Chain

A time-dependent key has Accept or Send time constraints. It is valid only during the times that are defined for the key . If a time-dependent key is used, there is usually more than one key in the key chain entry.

Syntax: [no] key-chain < *chain_name* > key < *key_id* >

Generates or deletes a key in the key chain entry < chain_name >. Using the optional **no** *form of the command deletes the key. The < key_id > is any number from 0-255.*

[key-string < *key_str* >]

This option specifies the key value referenced by the protocol using the key. The < key_str > can be any string up to 14 characters in length.

accept-lifetime < *mm/dd/yy* [*yy*] *hh:mm:ss* | now >

*Specifies the **start** date and time of the valid period in which the switch can use this key to authenticate inbound packets.*

duration < *mm/dd/yy* [*yy*] *hh:mm:ss* | *seconds* >

*Specifies the **time period** during which the switch can use this key to authenticate inbound packets. Duration is either an end date and time or the number of seconds to allow after the start date and time (which is the* **accept-lifetime** *setting*).

send-lifetime <*mm/dd/yy[yy] hh:mm:ss* | now>

*Specifies the **start** date and time of the valid period in which the switch can transmit this key as authentication for outbound packets.*

duration < *mm/dd/yy[yy] hh:mm:ss* | *seconds* >

*Specifies the **time period** during which the switch can use this key to authenticate outbound packets. Duration is either an end date and time or the number of seconds to allow after the start date and time (which is the* **accept-lifetime** *setting*).

show key-chain < *chain_name* >

Displays the detail information about the keys used in the key chain named < chain_name >.

Note

Using time-dependent keys requires that all the switches have accurate, synchronized time settings. You can manually set the time or use the Time protocol feature included in the switches. For more information, refer to the chapter covering time protocols in the *Management and Configuration Guide* for your switch.

For example, to add a number of keys to the key chain entry “Procurve2”:

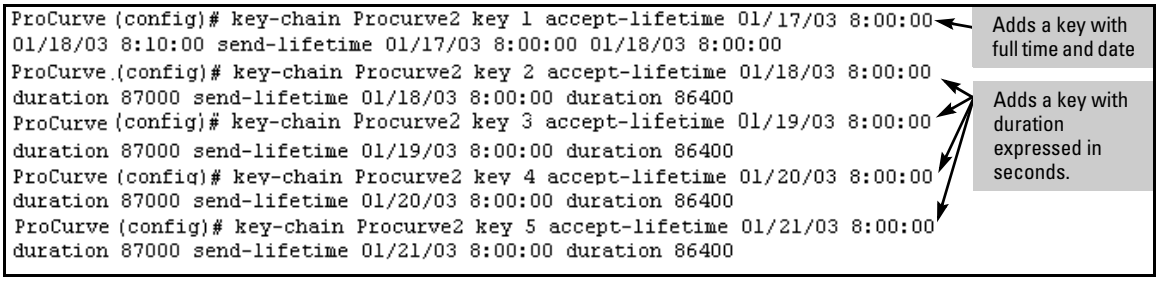

Figure 16-3. Adding Time-Dependent Keys to a Key Chain Entry

Note

Given transmission delays and the variations in the time value from switch to switch, it is advisable to include some flexibility in the Accept lifetime of the keys you configure. Otherwise, the switch may disregard some packets because either their key has expired while in transport or there are significant time variations between switches.

To list the result of the commands in figure 16-3:

```
ProCurve(config)# show key-chain Procurve2

 Chain - Procurve2

  Key | Accept Start GMT  Accept Stop GMT   Send Start GMT    Send Stop GMT
  --- + ----------------- ----------------- ----------------- -----------------
  1   | 01/17/03 08:00:00 01/18/03 08:10:00 01/17/03 08:00:00 01/18/03 08:00:00
  2   | 01/18/03 08:00:00 01/19/03 08:10:00 01/18/03 08:00:00 01/19/03 08:00:00
  3   | 01/19/03 08:00:00 01/20/03 08:10:00 01/19/03 08:00:00 01/20/03 08:00:00
  4   | 01/20/03 08:00:00 01/21/03 08:10:00 01/20/03 08:00:00 01/21/03 08:00:00
  5   | 01/21/03 08:00:00 01/22/03 08:10:00 01/21/03 08:00:00 01/22/03 08:00:00

  OSPF Interface References

  Interface
  ---------------

  OSPF Virtual Link References

  Area/Virtual Link
  ------------------------------
```

Figure 16-4. Display of Time-Dependent Keys in the Key Chain Entry

You can use **show key-chain** to display the key status at the time the command is issued. Using the information from the example configuration in figures 16-3 and 16-4, if you execute **show key-chain** at 8:05 on 01/19/03, the display would appear as follows:

```
ProCurve(config)# show key-chain

 Key Chains

  Chain Name                       Keys Active Expired
  -------------------------------- ---- ------ -------
  Procurve1                        1    1      0
  Procurve2                        5    2      1
```

Figure 16-5. Status of Keys in Key Chain Entry "Procurve2"

The "Procurve1" key chain entry is a time-independent key and will not expire. "Procurve2" uses time-dependent keys, which result in this data:

Expired = 1	Key 1 has expired because its lifetime ended at 8:10 on 01/18/03, the previous day.
Active = 2	Key 2 and 3 are both active for 10 minutes from 8:00 to 8:10 on 1/19/03.

Keys 4 and 5 are either not yet active or expired. The total number of keys is 5.