CHAPTER 7

Configuring MPLS LDP Lossless MD5 Session Authentication

This chapter describes how to configure Multiprotocol Label Switching (MPLS) Label Distribution Protocol (LDP) lossless MD5 session authentication on Cisco NX-OS devices.

This chapter includes the following sections:

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information" chapter or the Feature History table below.

Information About MPLS LDP Lossless MD5 Session Authentication

The MPLS LDP lossless MD5 session authentication feature enables an LDP session to be password protected without tearing down and reestablishing the LDP session.

The following topics provide information about the LDP lossless MD5 session authentication feature:

- How Messages Are Exchanged in MPLS LDP Lossless MD5 Session Authentication, page 7-79
- Benefits of MPLS LDP Lossless MD5 Session Authentication, page 7-79
- Keychain Use with MPLS LDP Lossless MD5 Session Authentication, page 7-80
- Application of Rules to Overlapping Passwords, page 7-81
- Resolving LDP Password Problems, page 7-81

How Messages Are Exchanged in MPLS LDP Lossless MD5 Session Authentication

MPLS LDP messages (discovery, session, advertisement, and notification messages) are exchanged between LDP peers through two channels:

- LDP discovery messages are transmitted as User Datagram Protocol (UDP) packets to the well-known LDP port.
- Session, advertisement, and notification messages are exchanged through a TCP connection established between two LDP peers. These messages can be protected against spoofed TCP segments by using the TCP MD5 signature option.

The MPLS LDP lossless MD5 session authentication feature allows an LDP session to incur a password change without tearing down and reestablishing the LDP session.

Benefits of MPLS LDP Lossless MD5 Session Authentication

MPLS LDP MD5 session authentication allows you to set up password requirements for a set of LDP neighbors to help prevent unauthorized peers from establishing LDP sessions and to block spoofed TCP messages.

The MPLS LDP lossless MD5 session authentication feature provides these benefits:

- Enables you to specify peers for which password protection is required in order to prevent the establishment of LDP sessions with unexpected peers.
- Enables you to activate or change LDP MD5 session authentication without interrupting the LDP session.
- Enables you to configure multiple passwords so one password can be used now and other passwords later.

Note LDP passwords cannot be configured on interfaces. You can configure one password per peer or per peer group. To configure multiple passwords, you must use keychains. The **key-chain** command allows different key strings to be used at different times according to the keychain configuration.

- Enables you to configure asymmetric passwords, which allows one password to be used for incoming TCP segments and a different password to be used for outgoing TCP segments.
- Enables you to configure passwords so that they overlap for a period of time. This functionality is beneficial when the clocks on two LSRs are not synchronized.

- If the neighboring nodes support graceful restart, then LDP sessions are gracefully restarted. The LDP MD5 password configuration is checkpointed to the standby route processors (RPs). The LDP MD5 password is used by the router when the new active RP attempts to establish LDP sessions with neighbors after the switchover.

Note Passwords can be configured to change over time, but they are not guaranteed to be lossless unless keychains are used with overlapping send and accept lifetimes for the transmit and receive keys.

Keychain Use with MPLS LDP Lossless MD5 Session Authentication

The MPLS LDP lossless MD5 session authentication feature allows keychains to be used to specify different MD5 keys to authenticate LDP traffic exchanged in each direction.

In the following example, three passwords are configured:

- Key 1 specifies the lab password. The **send-lifetime** command enables the lab password to authenticate the outgoing TCP segments from April 2, 2010, at 10:00:00 a.m. until May 2, 2010, at 10:00:00 a.m. The **accept-lifetime** command is configured so that the lab password is never used to authenticate incoming TCP segments. The **accept-lifetime** command enables the lab password for 1 second on January 1, 1970. By setting the date to the past and by enabling a duration of 1 second, the password for incoming TCP segments immediately expires. If the **accept-lifetime** command is omitted from the keychain configuration, then the password is always valid for incoming TCP segments.
- Key 2 and key 3 specify the lab2 and lab3 passwords, respectively. The **send-lifetime** commands enable the passwords for 1 second on January 1, 1970. By setting the date to the past and by enabling a duration of 1 second, the passwords for outgoing TCP segments immediately expire. If the **send-lifetime** commands are omitted from the keychain configuration, the passwords are always valid for outgoing TCP segments. The **accept-lifetime** commands for key 2 and key 3 enable the passwords to authenticate the incoming TCP segments from April 2, 2010, at 10:00:00 a.m. until April 17, 2010, at 10:00:00 a.m. and from April 17, 2010, at 10:00:00 a.m. until May 2, 2010, at 10:00:00 a.m., respectively.

```
switch(config)# ip prefix-list nbrp1 permit 10.0.0.0/32
switch(config)# key chain KeyChain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string lab
switch(config-keychain-key)# send-lifetime 10:00:00 Apr 2 2010 10:00:00 May 2 2010
switch(config-keychain-key)# accept-lifetime 00:00:00 Jan 1 1970 duration 1
switch(config-keychain-key)# exit
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string lab2
switch(config-keychain-key)# send-lifetime 00:00:00 Jan 1 1970 duration 1
switch(config-keychain-key)# accept-lifetime 10:00:00 Apr 2 2010 10:00:00 Apr 17 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 3
switch(config-keychain-key)# key-string lab3
switch(config-keychain-key)# send-lifetime 00:00:00 Jan 1 1970 duration 1
switch(config-keychain-key)# accept-lifetime 10:00:00 Apr 17 2010 10:00:00 May 2 2010
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password required for nbrp1
switch(config-ldp)# password option 1 for nbrpl key-chain KeyChain1
```

Application of Rules to Overlapping Passwords

Overlapping passwords can be useful when two LSRs have clocks that are not synchronized. The overlapping passwords provide a window to ensure that TCP packets are not dropped. The following rules apply to overlapping passwords:

- If the send-lifetime value for the next password begins before the send-lifetime value of the current password expires, the password with the shorter key ID is used during the overlap period. The send-lifetime value of the current password can be shortened by configuring a shorter send-lifetime value. Similarly, the send-lifetime value of the current password can be lengthened by configuring a longer send-lifetime value.
- If the accept-lifetime value for the next password begins before the accept-lifetime value of the current password expires, both the next password and the current password are used concurrently. The next password information is passed to TCP. If TCP fails to authenticate the incoming segments with the current password, it tries authenticating with the next password. If TCP authenticates a segment using the new password, it discards the current password and uses the new password from that point on.
- If a password for incoming or outgoing segments expires and no additional valid password is configured, one of the following actions occurs:
  - If a password is required for the neighbor, LDP drops the existing session.
  - If a password is not required for the neighbor, LDP attempts to roll over to a session that does not require authentication. This attempt also fails unless the password expires on both LSRs at the same time.

Resolving LDP Password Problems

LDP displays error messages when an unexpected neighbor attempts to open an LDP session or the LDP password configuration is invalid.

When a password is required for a potential LDP neighbor but no password is configured for it, the LSR ignores LDP hello messages from that neighbor. When the LSR processes the hello message and tries to establish a TCP connection with the neighbor, it displays the error message and stops establishing the LDP session with the neighbor. The error is rate-limited and has the following format:

```
2010 Sep 9 09:59:43.274519 ldp: MD5 protection is required for peer 3.3.3.3:0(default),
but no password is configured.
```

The output of the **show sockets connection detail** command shows a summary of TCP connection failures.

Licensing Requirements for MPLS LDP Lossless MD5 Session Authentication

Product	License Requirement
Cisco NX-OS	MPLS LDP lossless MD5 session authentication requires an MPLS license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Prerequisites for MPLS LDP Lossless MD5 Session Authentication

MPLS LDP lossless MD5 session authentication has the following prerequisites:

- You must configure static or dynamic routing for the LSR.

Guidelines and Limitations for MPLS LDP Lossless MD5 Session Authentication

MPLS LDP lossless MD5 session authentication has the following configuration guidelines and limitations:

- Lossless MD5 session authentication is supported between Cisco NX-OS and Cisco IOS devices.

Default Settings for MPLS LDP Lossless MD5 Session Authentication

Table 7-1 lists the default settings for MPLS LDP lossless MD5 session authentication parameters.

Table 7-1 Default MPLS LDP Lossless MD5 Session Authentication Parameters

Parameters	Default
MPLS LDP lossless MD5 session authentication	Disabled

Configuring MPLS LDP Lossless MD5 Session Authentication

This section includes the following topics:

- Configuring MPLS LDP Lossless MD5 Session Authentication Using a Keychain, page 7-82
- Configuring a Fallback Password within a Keychain, page 7-88
- Enabling the Display of MPLS LDP Password Changes, page 7-93

Configuring MPLS LDP Lossless MD5 Session Authentication Using a Keychain

You configure MPLS LDP lossless MD5 session authentication using a keychain. Keychains allow a different key string to be used at different times according to the keychain configuration. MPLS LDP gives TCP the keychain information, and TCP queries the appropriate keychain to obtain the current live key and key ID for the specified keychain.

If the sessions to be protected are not already using your keychain, the configuration changes take effect the next time that each session is reestablished. For sessions already using this keychain, the configuration changes take effect immediately. For LDP sessions not already using the keychain, the preexisting authentication remains in effect until the next session is reestablished. A session reestablishment might be forced (with a temporary loss of label switching if LDP graceful restart is not enabled on the session) by using the **clear mpls ldp neighbor** *ip-address* command.

If you are not already using authentication, you must make all the required changes on all peers and then force them into action with the **password required for** *prefix-list* command so that the sessions using the specified the *prefix list* are reestablished using the lossless MD5 session authentication you have defined in your configurations.

Prerequisites

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that MPLS LDP is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **ip prefix-list** *prefix-list* **permit** *network/length*
3. key chain *keychain-name*
4. **key** *key-id*
5. key-string *key*
   (If you plan to configure a fallback keychain in Step 13, repeat Steps 3 through 5 to configure a backup keychain.)
6. **accept-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds* | *end-time* | **infinite**}
7. **send-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds* | *end-time* | **infinite**}
8. exit
9. exit
10. mpls ldp configuration
11. **(Optional) password required** [**for** p*refix-list*]
12. **password option** *number* **for** *prefix-list* **key-chain** *keychain-name*
13. **(Optional)** password fallback key-chain *keychain-name*
14. **(Optional) show mpls ldp neighbor** [*ip-address* | *interface slot/port*] [**detail**]
15. **(Optional) copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`configure terminal`**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**`ip prefix-list`** *`prefix-list`* **`permit`** *`network/length`*
	Example:
	`switch(config)# ip prefix-list p1 permit 10.0.0.0/32`
		Creates an IP prefix list and specifies the prefixes permitted by the prefix list. The *prefix-list* argument can be up to 63 characters.

Step 3	**`key chain`** *`keychain-name`*
	Example:
	`switch(config)# key chain KeyChain1`
	`switch(config-keychain)#`
		Identifies a group of authentication keys and enters keychain configuration mode.

Step 4	**`key`** *`key-id`*
	Example:
	`switch(config-keychain)# key 1`
	`switch(config-keychain-key)#`
		Identifies an authentication key on a keychain and enters keychain key configuration mode.
		The *key-id* argument must be a numeral from 0 to 65535.

Step 5	**`key-string`** *`key`*
	Example:
	`switch(config-keychain-key)# key-string pwd1`
		Specifies the authentication string for a key.
		The *string* argument can be from 1 to 80 uppercase or lowercase alphanumeric characters. The first character cannot be a numeral.
		Note If you plan to configure a fallback keychain in Step 13, repeat Steps 3 through 5 to configure a backup keychain.

	Command	Purpose
Step 6	**accept-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds* | *end-time* | **infinite**} **Example:** switch(config-keychain-key)# accept-lifetime 10:00:00 Jan 13 2010 10:00:00 Jun 13 2010	Specifies the time period during which the authentication key on a keychain can be used for verifying incoming TCP segments. The *start-time* argument identifies the time to start, and the **local** *start-time* argument identifies the time to start in the local time zone. Both arguments have the same parameters: • *hh:mm:ss* is the time format. • Enter the number of days from 1 to 31. • Enter the name of the month. • Enter the year from the present to 2035. **Note** The time reference depends on the clock time zone configuration on the router. If it is configured, the local time zone is used (for example, EST, PST, or so on). Once the start time is entered, select from the following: • The **duration** keyword sets the key lifetime duration in seconds. • The *end-time* argument sets the time to stop. These parameters are the same as those used for the *start-time argument*. • The **infinite** keyword allows the accept-lifetime period to never expire. **Note** If the **no accept-lifetime** value is defined, the associated receive password is valid for authenticating incoming TCP segments.

	Command	Purpose
Step 7	**send-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds* | *end-time* | **infinite**} **Example:** switch(config-keychain-key)# send-lifetime 10:00:00 Jan 13 2010 10:00:00 Jun 13 2010	Specifies the time period during which the authentication key on a keychain can be used for verifying outgoing TCP segments. The *start-time* argument identifies the time to start, and the **local** *start-time* argument identifies the time to start in the local time zone. Both arguments have the same parameters: • *hh***:***mm***:***ss* is the time format. • Enter the number of days from 1 to 31. • Enter the name of the month. • Enter the year from 1993 to 2035. **Note** The time reference depends on the clock time zone configuration on the router. If it is configured, the local time zone is used (for example, EST, PST, or so on). Once the start time is entered, select from the following: • The **duration** keyword sets the send lifetime duration in seconds. • The *end-time* argument sets the time to stop. These parameters are the same as those used for the *start-time argument*. • The **infinite** keyword allows the send lifetime period to never expire. **Note** If the **no send-lifetime** value is defined, the associated send password is valid for authenticating outgoing TCP segments.
Step 8	exit **Example:** switch(config-keychain-key)# exit switch(config-keychain)#	Exits keychain key configuration mode.
Step 9	exit **Example:** switch(config-keychain)# exit switch(config)#	Exits keychain configuration mode.
Step 10	**mpls ldp configuration** **Example:** switch(config)# mpls ldp configuration switch(config-ldp)#	Enters LDP configuration mode.

	Command	Purpose
Step 11	**password required** [**for** *prefix-list*] **Example:** `switch(config-ldp)# password required for p1`	(**Optional**) Specifies that LDP must use a password when establishing a session between LDP peers. The **for** *prefix-list* keyword-argument pair names a prefix list, which specifies that a password is mandatory only for LDP sessions with neighbors whose LDP router IDs are permitted by the list.
Step 12	**password option** *number* **for** *prefix-list* **key-chain** *keychain-name* **Example:** `switch(config-ldp)# password option 25 for p1 key-chain KeyChain1`	Configures an MD5 password for LDP sessions with neighbors whose LDP router IDs are permitted by a specified prefix list. • The *number* argument defines the order in which the prefix lists are evaluated in the determination of a neighbor password. The valid range is from 1 to 32767. • The **for** *prefix-list* keyword-argument pair specifies the name of the prefix list that includes the LDP router IDs of those neighbors for which the password applies. • The **key-chain** *keychain-name* keyword-argument pair specifies a keychain of multiple MD5 keys to be used for the specified LDP sessions.
Step 13	**password fallback key-chain** *keychain-name* **Example:** `switch(config-ldp)# password fallback key-chain KeyChainBackup`	(**Optional**) Configures a backup MD5 keychain for peers that have no keychain configured in Step 12. The **key-chain** *keychain-name* keyword-argument pair specifies a keychain of multiple MD5 keys to be used for the LDP sessions.

	Command	Purpose
Step 14	**show mpls ldp neighbor** [*ip-address* | *interface slot/port*] [**detail**] **Example:** `switch(config-ldp)# show mpls ldp neighbor detail`	**(Optional)** Displays the status of LDP sessions. • The *ip-address* argument identifies the neighbor with the IP address for which password protection is configured. • The *interface* argument lists the LDP neighbors accessible over this interface. • The **detail** keyword displays password information for this neighbor. Here are the items displayed: – An indication as to whether a password is mandatory for this neighbor (required or not required). – The password source (neighbor, fallback, or option number). – An indication as to whether the latest configured password or keychain for this neighbor is used by the TCP session (in use) or the TCP session uses an old password or keychain (stale). A keychain is always considered stale when compared with a simple password, even when the keychain may at the moment lead to using the same simple password.
Step 15	**copy running-config startup-config** **Example:** `switch(config-ldp)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring a Fallback Password within a Keychain

You can change MD5 passwords for LDP session authentication without having to close and reestablish an existing LDP session by configuring a fallback password within a keychain.

This addition of a fallback password is nondisruptive when done with peers already using the keychain.

Prerequisites

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Ensure that MPLS LDP is enabled.

SUMMARY STEPS

1. **configure terminal**
2. key chain *keychain-name*
3. **key** *key-id*
4. **key-string** *key*

5. **send-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds* | *end-time* | **infinite**}
6. **accept-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds* | *end-time* | **infinite**}
7. **exit**
8. **key** *key-id*
9. **key-string** *key*
10. **(Optional) show mpls ldp neighbor** [*ip-address* | *interface slot/port*] [**detail**]
11. **(Optional) copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**key chain** *keychain-name*
	Example:
	`switch(config)# key chain KeyChain1`
	`switch(config-keychain)#`
		Identifies a group of authentication keys and enters keychain configuration mode.

Step 3	**key** *key-id*
	Example:
	`switch(config-keychain)# key 1`
	`switch(config-keychain-key)#`
		Identifies an authentication key on a keychain and enters keychain key configuration mode.
		The *key-id* argument must be a numeral from 0 to 65535.

	Command	Purpose

Step 4	**`key-string`** *`key`*
	`Example:`
	`switch(config-keychain-key)# key-string pwd1`
		Specifies the authentication string for a key.
		The *string* argument can be from 1 to 80 uppercase or lowercase alphanumeric characters. The first character cannot be a numeral.

Step 5	**`send-lifetime`** `{`*`start-time`* `|` **`local`** *`start-time`*`}` `{`**`duration`** *`seconds`* `|` *`end-time`* `|` **`infinite`**`}`
	`Example:`
	`switch(config-keychain-key)# send-lifetime 10:00:00 Jan 13 2010 10:00:00 Jun 13 2010`
		Specifies the time period during which the authentication key on a keychain can be used for verifying outgoing TCP segments. The *start-time* argument identifies the time to start, and the **local** *start-time* argument identifies the time to start in the local time zone. Both arguments have the same parameters:
		• *hh***:***mm***:***ss* is the time format.
		• Enter the number of days from 1 to 31.
		• Enter the name of the month.
		• Enter the year from 1993 to 2035.
		Note The time reference depends on the clock time zone configuration on the router. If it is configured, the local time zone is used (for example, EST, PST, or so on).
		Once the start time is entered, select from the following:
		• The **duration** keyword sets the send lifetime duration in seconds.
		• The *end-time* argument sets the time to stop. These parameters are the same as those used for the *start-time argument.*
		• The **infinite** keyword allows the send lifetime period to never expire.
		Note If the **no send-lifetime** value is defined, the associated send password is valid for authenticating outgoing TCP segments.

	Command	Purpose
Step 6	**accept-lifetime** {*start-time* | **local** *start-time*} {**duration** *seconds* | *end-time* | **infinite**} **Example:** switch(config-keychain-key)# accept-lifetime 10:00:00 Jan 13 2010 10:00:00 Jun 13 2010	Specifies the time period during which the authentication key on a keychain can be used for verifying incoming TCP segments. The *start-time* argument identifies the time to start, and the **local** *start-time* argument identifies the time to start in the local time zone. Both arguments have the same parameters: • *hh:mm:ss* is the time format. • Enter the number of days from 1 to 31. • Enter the name of the month. • Enter the year from the present to 2035. **Note** The time reference depends on the clock time zone configuration on the router. If it is configured, the local time zone is used (for example, EST, PST, or so on). Once the start time is entered, select from the following: • The **duration** keyword sets the key lifetime duration in seconds. • The *end-time* argument sets the time to stop. These parameters are the same as those used for the *start-time argument*. • The **infinite** keyword allows the accept-lifetime period to never expire. **Note** If the **no accept-lifetime** value is defined, the associated receive password is valid for authenticating incoming TCP segments.
Step 7	exit **Example:** switch(config-keychain-key)# exit switch(config-keychain)#	Exits keychain key configuration mode.
Step 8	**key** *key-id* **Example:** switch(config-keychain)# key 65535 switch(config-keychain-key)#	Identifies an authentication key on a keychain and enters keychain key configuration mode. The *key-id* argument must be a numeral from 0 to 65535.
Step 9	**key-string** *key* **Example:** switch(config-keychain-key)# key-string fallback-password	Specifies the authentication string for a key. The *string* argument can be from 1 to 80 uppercase or lowercase alphanumeric characters. The first character cannot be a numeral.

	Command	Purpose
Step 10	`show mpls ldp neighbor [ip-address	interface slot/port] [detail]` **Example:** `switch(config-ldp)# show mpls ldp neighbor detail`	**(Optional)** Displays the status of LDP sessions. • The *ip-address* argument identifies the neighbor with the IP address for which password protection is configured. • The *interface* argument lists the LDP neighbors accessible over this interface. • The **detail** keyword displays password information for this neighbor. Here are the items displayed: – An indication as to whether a password is mandatory for this neighbor (required or not required) – The password source (neighbor, fallback, or option number) – An indication as to whether the latest configured password for this neighbor is used by the TCP session (in use) or the TCP session uses an old password (stale)

Step 11	`copy running-config startup-config`
	Example:
	`switch(config-ldp)# copy running-config startup-config`
		(Optional) Copies the running configuration to the startup configuration.

Enabling the Display of MPLS LDP Password Changes

You can enable the display of events related to password configuration changes and rollover events.

Note When a password is required for a neighbor but no password is configured for the neighbor, an error message is logged as shown in the "Resolving LDP Password Problems" section on page 7-81.

Prerequisites

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. **configure terminal**
2. mpls ldp configuration
3. **logging password configuration** [**rate-limit** *number*]
4. **logging password rollover** [**rate-limit** *number*]
5. **(Optional) copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**`configure terminal`**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**`mpls ldp configuration`**
	Example:
	`switch(config)# mpls ldp configuration`
	`switch(config-ldp)#`
		Enters LDP configuration mode.

Step 3	**`logging password configuration [rate-limit`** *`number`***`]`**
	Example:
	`switch(config-ldp)# logging password configuration rate-limit 20`
		Enables the display of events related to password configuration changes. The output displays events when a new password is configured or an existing password has been changed or deleted. A rate limit of 1 to 60 messages a minute can be specified.

Step 4	**`logging password rollover [rate-limit`** *`number`***`]`**
	Example:
	`switch(config-ldp)# logging password rollover rate-limit 10`
		Enables the display of events related to password rollover events. Events are displayed when a new password is used for authentication or when authentication is disabled. A rate limit of 1 to 60 messages a minute can be specified.

Step 5	**`copy running-config startup-config`**
	Example:
	`switch(config-ldp)# copy running-config startup-config`
		(Optional) Copies the running configuration to the startup configuration.

Verifying the MPLS LDP Lossless MD5 Session Authentication

To display the MPLS LDP lossless MD5 session authentication, perform one of these tasks:

Command	Purpose
show mpls ldp neighbor [*ip-address* | *interface slot/port*] **detail**	Displays the status of LDP sessions, including detailed neighbor information.
show mpls ldp neighbor [*ip-address* | *interface slot/port*] **password**	Displays the status of LDP sessions, including password information for the neighbor.

For detailed information about the fields in the output from this command, see the *Cisco Nexus 7000 Series NX-OS MPLS Command Reference*.

Configuration Examples for MPLS LDP Lossless MD5 Session Authentication

This section provides configuration examples for MPLS LDP lossless MD5 session authentication and includes the following topics:

- Examples: Configuring MPLS LDP Lossless MD5 Session Authentication Using a Keychain, page 7-95
- Examples: Using a Fallback Password within a Keychain, page 7-96
- Examples: Common Misconfigurations When Changing an MPLS LDP Lossless MD5 Session Authentication Password, page 7-97

Examples: Configuring MPLS LDP Lossless MD5 Session Authentication Using a Keychain

The following example shows how to configure two peer LSRs that use symmetrical MD5 keys:

LSR1 (with Router ID 10.1.1.1)

```
switch# configure terminal
switch(config)# ip prefix-list 10 permit 10.2.2.2/32
switch(config)# key chain KeyChain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string pwd1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:00:00 Feb 1 2010
switch(config-keychain-key)# accept-lifetime 09:00:00 Jan 1 2010 11:00:00 Feb 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password required for 10
switch(config-ldp)# password option 1 for 10 key-chain KeyChain1
```

LSR2 (with Router ID 10.2.2.2)

```
switch# configure terminal
switch(config)# ip prefix-list 10 permit 10.1.1.1/32
switch(config)# key chain KeyChain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string pwd1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:00:00 Feb 1 2010
switch(config-keychain-key)# accept-lifetime 09:00:00 Jan 1 2010 11:00:00 Feb 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password required for 10
switch(config-ldp)# password option 1 for 10 key-chain KeyChain1
```

The following example shows how to configure two peer LSRs that use asymmetrical MD5 keys:

LSR1 (with Router ID 10.1.1.1)

```
switch# configure terminal
switch(config)# ip prefix-list 10 permit 10.2.2.2/32
switch(config)# key chain KeyChain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string pwd1
switch(config-keychain-key)# accept-lifetime 00:00:00 Jan 1 2010 duration 1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:00:00 Feb 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string pwd2
switch(config-keychain-key)# accept-lifetime 09:00:00 Jan 1 2010 11:00:00 Feb 1 2010
switch(config-keychain-key)# send-lifetime 00:00:00 Jan 1 2010 duration 1
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password required for 10
switch(config-ldp)# password option 1 for 10 key-chain KeyChain1
```

LSR2 (with Router ID 10.2.2.2)

```
switch# configure terminal
switch(config)# ip prefix-list 10 permit 10.1.1.1/32
switch(config)# key chain KeyChain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string pwd2
switch(config-keychain-key)# accept-lifetime 00:00:00 Jan 1 2010 duration 1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:00:00 Feb 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string pwd1
switch(config-keychain-key)# accept-lifetime 09:00:00 Jan 1 2010 11:00:00 Feb 1 2010
switch(config-keychain-key)# send-lifetime 00:00:00 Jan 1 2010 duration 1
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password required for 10
switch(config-ldp)# password option 1 for 10 key-chain KeyChain1
```

Examples: Using a Fallback Password within a Keychain

The following example shows how to configure a fallback password within a keychain. For example, because 65535 is the largest key value allowed for a keychain, the “SampleKeyChain” keychain provides a fallback send and receive password of “fallback-password” for times outside the specified send and accept lifetimes for the specified keystrings (passwords).

```
switch# configure terminal
switch(config)# key chain SampleKeyChain
switch(config-keychain)# key 10
switch(config-keychain-key)# key-string lab1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:30:00 Jan 1 2010
switch(config-keychain-key)# accept-lifetime 10:00:00 Jan 1 2010 10:45:00 Jan 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 20
switch(config-keychain-key)# key-string lab2
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:30:00 Jan 1 2010
switch(config-keychain-key)# accept-lifetime 10:00:00 Jan 1 2010 10:45:00 Jan 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 65535
switch(config-keychain-key)# key-string fallback-password
```

Examples: Common Misconfigurations When Changing an MPLS LDP Lossless MD5 Session Authentication Password

The following examples show common misconfigurations that can occur when the MD5 password is migrated in a lossless way. Misconfigurations can lead to undesired behavior in an LDP session.

- Example: Incorrect Keychain LDP Password Configuration, page 7-97
- Example: Reconfiguring a Keychain to Prevent TCP Authentication and LDP Session Failures, page 7-99
- Avoiding Prefix List Configuration Problems, page 7-100

Example: Incorrect Keychain LDP Password Configuration

Possible misconfigurations can occur when keychain-based commands are used with the **password option for key-chain** command. If the **accept-lifetime** and **send-lifetime** commands are not specified in this configuration, then a misconfiguration can occur when more than two keys are in a keychain.

The following example shows an incorrect keychain configuration with three passwords for LSR A and LSR B in the keychain:

LSR A Incorrect Keychain LDP Password Configuration

```
switch# configure terminal
switch(config)# ip prefix-list 10 permit 10.11.11.11
switch(config)# key chain KeyChain1
switch(config-keychain)# key 10
switch(config-keychain-key)# key-string lab1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:30:00 Jan 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 11
switch(config-keychain-key)# key-string lab2
switch(config-keychain-key)# send-lifetime 10:30:00 Jan 1 2010 10:30:00 Feb 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 12
switch(config-keychain-key)# key-string lab3
switch(config-keychain-key)# send-lifetime 10:30:00 Feb 1 2010 10:30:00 Mar 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password option 5 for 10 key-chain KeyChain1
```

LSR B Incorrect Keychain LDP Password Configuration

```
switch# configure terminal
switch(config)# ip prefix-list 10 permit 10.10.10.10
switch(config)# key chain KeyChain1
switch(config-keychain)# key 10
switch(config-keychain-key)# key-string lab1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:30:00 Jan 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 11
switch(config-keychain-key)# key-string lab2
switch(config-keychain-key)# send-lifetime 10:30:00 Jan 1 2010 10:30:00 Feb 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 12
switch(config-keychain-key)# key-string lab3
switch(config-keychain-key)# send-lifetime 10:30:00 Feb 1 2010 10:30:00 Mar 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password option 5 for 10 key-chain KeyChain1
```

In the examples above for LSR A and LSR B during the period of the third **send-lifetime 10:30:00 Feb 1 2010 10:30:00 Mar 1 2010 command**, all three configured keys are valid as receive keys, and only the last configured key is valid as a transmit key. The keychain resolution rules dictate that keys 10 and 11 are used as receive keys, and only the last key (key 12) can be used as the transmit key. Because the transmit and receive keys are mismatched, the LDP session will not stay active.

Note When more than two passwords are configured in a keychain, the configuration needs to have both the **accept-lifetime** and **send-lifetime** commands configured correctly.

The following example shows the correct keychain configuration with multiple passwords in the keychain:

LSR A Correct Keychain LDP Password Configuration

```
switch# configure terminal
switch(config)# ip prefix-list 10 permit 10.11.11.11
switch(config)# key chain KeyChain1
switch(config-keychain)# key 10
switch(config-keychain-key)# key-string lab1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:30:00 Jan 1 2010
switch(config-keychain-key)# accept-lifetime 10:00:00 Jan 1 2010 10:45:00 Jan 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 11
switch(config-keychain-key)# key-string lab2
switch(config-keychain-key)# send-lifetime 10:30:00 Jan 1 2010 10:30:00 Feb 1 2010
switch(config-keychain-key)# accept-lifetime 10:15:00 Jan 1 2010 10:45:00 Feb 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 12
switch(config-keychain-key)# key-string lab3
switch(config-keychain-key)# send-lifetime 10:30:00 Feb 1 2010 10:30:00 Mar 1 2010
switch(config-keychain-key)# accept-lifetime 10:15:00 Feb 1 2010 10:45:00 Mar 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password option 5 for 10 key-chain KeyChain1
```

LSR B Correct Keychain LDP Password Configuration

```
switch# configure terminal
switch(config)# ip prefix-list 10 permit 10.10.10.10
switch(config)# key chain KeyChain1
switch(config-keychain)# key 10
switch(config-keychain-key)# key-string lab1
switch(config-keychain-key)# send-lifetime 10:00:00 Jan 1 2010 10:30:00 Jan 1 2010
switch(config-keychain-key)# accept-lifetime 10:00:00 Jan 1 2010 10:45:00 Jan 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 11
switch(config-keychain-key)# key-string lab2
switch(config-keychain-key)# send-lifetime 10:30:00 Jan 1 2010 10:30:00 Feb 1 2010
switch(config-keychain-key)# accept-lifetime 10:15:00 Jan 1 2010 10:45:00 Feb 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 12
switch(config-keychain-key)# key-string lab3
switch(config-keychain-key)# send-lifetime 10:30:00 Feb 1 2010 10:30:00 Mar 1 2010
switch(config-keychain-key)# accept-lifetime 10:15:00 Feb 1 2010 10:45:00 Mar 1 2010
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# mpls ldp configuration
switch(config-ldp)# password option 5 for 10 key-chain KeyChain1
```

In the examples above for LSR A and LSR B during the period of the third **send-lifetime 10:30:00 Feb 1 2010 10:30:00 Mar 1 2010** command, only the last key (key 12) is valid as the transmit and receive key. Therefore, the LDP session remains active.

Example: Reconfiguring a Keychain to Prevent TCP Authentication and LDP Session Failures

If the configuration needs to specify the last key in the keychain to always be valid, then configure the keychain to have at least two keys. Each key must be configured with both the send and accept lifetime period.

```
switch(config)# key chain KeyChain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string firstpass
switch(config-keychain-key)# accept-lifetime 01:03:00 Sep 10 2010 01:10:00 Sep 10 2010
switch(config-keychain-key)# send-lifetime 01:05:00 Sep 10 2010 01:08:00 Sep 10 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string secondpass
switch(config-keychain-key)# accept-lifetime 01:06:00 Sep 10 2010 01:17:00 Sep 10 2010
switch(config-keychain-key)# send-lifetime 01:08:00 Sep 10 2010 01:15:00 Sep 10 2010
switch(config-keychain-key)# exit
switch(config-keychain)# key 3
switch(config-keychain-key)# key-string thirdpass
```

If the configuration needs to specify the first keychain for the time interval, then use the second key forever after that interval. You can do so by configuring the start time for the second key to begin shortly before the end time of the first key and by configuring the second key to be valid forever after that interval. For example:

```
switch(config)# key chain KeyChain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string firstpass
switch(config-keychain-key)# accept-lifetime 00:03:00 Sep 10 2010 01:10:00 Sep 10 2010
switch(config-keychain-key)# send-lifetime 00:05:00 Sep 10 2010 01:08:00 Sep 10 2010
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string secondpass
switch(config-keychain-key)# accept-lifetime 01:06:00 Sep 10 2010 infinite
switch(config-keychain-key)# send-lifetime 01:08:00 Sep 10 2010 infinite
```

If the configuration needs to specify the two keys in the order of the second key, first key, and second key again, then specify three keys in that order. For example:

```
switch(config)# key chain KeyChain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string firstpass
switch(config-keychain-key)# accept-lifetime 00:03:00 Sep 10 2010 01:10:00 Sep 10 2010
switch(config-keychain-key)# send-lifetime 00:05:00 Sep 10 2010 01:08:00 Sep 10 2010
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string secondpass
switch(config-keychain-key)# accept-lifetime 01:06:00 Sep 10 2010 01:17:00 Sep 10 2010
switch(config-keychain-key)# send-lifetime 01:08:00 Sep 10 2010 01:15:00 Sep 10 2010
switch(config-keychain)# key 3
switch(config-keychain-key)# key-string firstpass
switch(config-keychain-key)# accept-lifetime 01:13:00 Sep 10 2010 infinite
switch(config-keychain-key)# send-lifetime 01:15:00 Sep 10 2010 infinite
```

Avoiding Prefix List Configuration Problems

Use caution when modifying or deleting a prefix list. Any empty prefix list implies “permit any” by default. When you use the **password option for key-chain** command for MPLS LDP lossless MD5 session authentication, and if the prefix list specified in the command becomes empty as a result of a modification or deletion, then all LDP sessions on the router expect a password. This configuration might cause undesired behavior in LDP sessions. To avoid this scenario, ensure that the proper prefix list is specified for each LSR.

Additional References for MPLS LDP Lossless MD5 Session Authentication

For additional information related to implementing MPLS LDP lossless MD5 session authentication, see the following sections:

- Related Documents, page 7-100
- MIBs, page 7-100

Related Documents

Related Topic	Document Title
CLI commands	*Cisco Nexus 7000 Series NX-OS MPLS Command Reference*
Cisco IOS MPLS LDP lossless MD5 session authentication	MPLS LDP—Lossless MD5 Session Authentication

MIBs

MIB	MIBs Link
MPLS-LDP-STD-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

Feature History for MPLS LDP Lossless MD5 Session Authentication

Table 7-2 lists the release history for this feature.

Table 7-2 Feature History for MPLS LDP Lossless MD5 Session Authentication

Feature Name	Releases	Feature Information
MPLS LDP lossless MD5 session authentication	5.2(1)	This feature was introduced.