RSVP Message Authentication

First Published: March 17, 2003
Last Updated: August 6, 2007

The Resource Reservation Protocol (RSVP) Message Authentication feature provides a secure method to control quality of service (QoS) access to a network.

History for the RSVP Message Authentication Feature

Release	Modification
12.2(15)T	This feature was introduced.
12.0(26)S	Restrictions were added for interfaces that use Fast Reroute (FRR) node or link protection and for RSVP hellos for FRR for packet over SONET (POS) interfaces.
12.0(29)S	Support was added for per-neighbor keys.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for RSVP Message Authentication

Ensure that RSVP is configured on one or more interfaces on at least two neighboring routers that share a link within the network.

Restrictions for RSVP Message Authentication

- The RSVP Message Authentication feature is only for authenticating RSVP neighbors.
- The RSVP Message Authentication feature cannot discriminate between various QoS applications or users, of which many may exist on an authenticated RSVP neighbor.
- Different send and accept lifetimes for the same key in a specific key chain are not supported; all RSVP key types are bidirectional.
- Authentication for graceful restart hello messages is supported for per-neighbor and per-access control list (ACL) keys, but not for per-interface keys.
- You cannot use the **ip rsvp authentication key** and the **ip rsvp authentication key-chain** commands on the same router interface.
- For a Multiprotocol Label Switching/Traffic Engineering (MPLS/TE) configuration, use per-neighbor keys with physical addresses and router IDs.

Information About RSVP Message Authentication

To configure RSVP Message Authentication, you need to understand the following concepts:

- Feature Design of RSVP Message Authentication, page 2
- Global Authentication and Parameter Inheritance, page 3
- Per-Neighbor Keys, page 3
- Key Chains, page 4
- Benefits of RSVP Message Authentication, page 4

Feature Design of RSVP Message Authentication

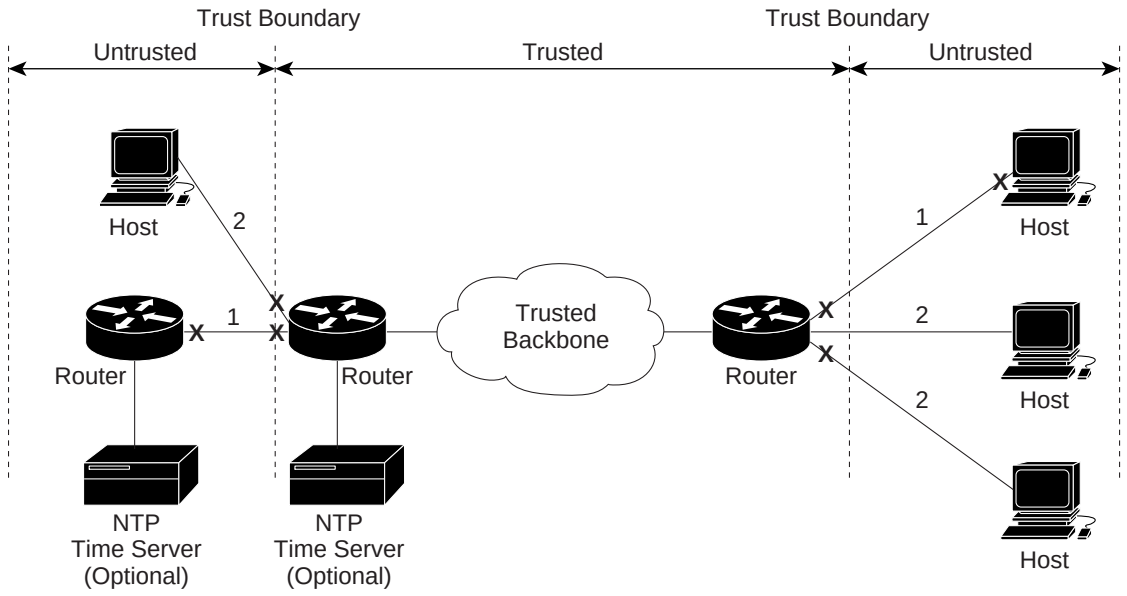
Network administrators need the ability to establish a security domain to control the set of systems that initiate RSVP requests.

The RSVP Message Authentication feature permits neighbors in an RSVP network to use a secure hash to sign all RSVP signaling messages digitally, thus allowing the receiver of an RSVP message to verify the sender of the message without relying solely on the sender's IP address as is done by issuing the **ip rsvp neighbor** command with an ACL.

The signature is accomplished on a per-RSVP-hop basis with an RSVP integrity object in the RSVP message as defined in RFC 2747. This method provides protection against forgery or message modification. However, the receiver must know the security key used by the sender in order to validate the digital signature in the received RSVP message.

Network administrators manually configure a common key for each RSVP neighbor interface on the shared network. A sample configuration is shown in Figure 1.

Figure 1 RSVP Message Authentication Configuration



Global Authentication and Parameter Inheritance

You can configure global defaults for all authentication parameters including key, type, window size, lifetime, and challenge. These defaults are inherited when you enable authentication for each neighbor or interface. However, you can also configure these parameters individually on a per-neighbor or per-interface basis in which case the inherited global defaults are ignored.

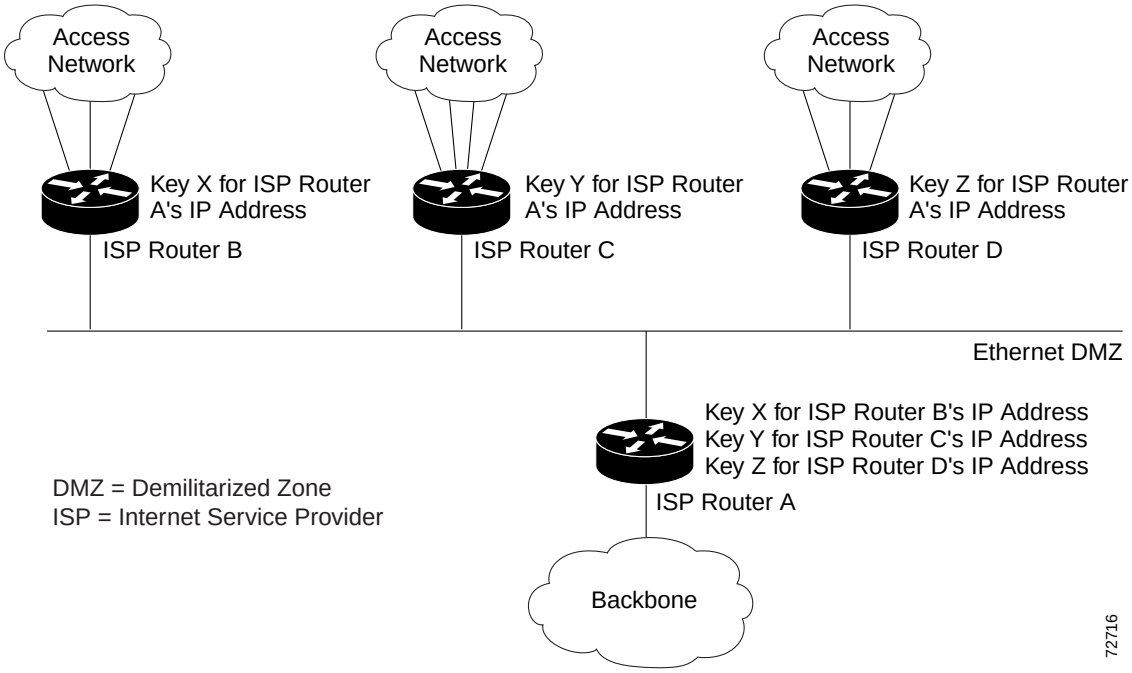
Using global authentication and parameter inheritance can simplify configuration because you can enable or disable authentication without having to change each per-neighbor or per-interface attribute. You can activate authentication for all neighbors by using two commands, one to define a global default key and one to enable authentication globally. However, using the same key for all neighbors does not provide the best network security.

Note RSVP uses the following rules when choosing which authentication parameter to use when that parameter is configured at multiple levels (per-interface, per-neighbor, or global). RSVP goes from the most specific to the least specific; that is, per-neighbor, per-interface, and then global. The rules are slightly different when searching the configuration for the right key to authenticate an RSVP message—per-neighbor, per-ACL, per-interface, and then global.

Per-Neighbor Keys

In Figure 2, to enable authentication between Internet service provider (ISP) Routers A and B, A and C, and A and D, the ISPs must share a common key. However, sharing a common key also enables authentication between ISP Routers B and C, C and D, and B and D. You may not want authentication among all the ISPs because they might be different companies with unique security domains Figure 2.

Figure 2 RSVP Message Authentication in an Ethernet Configuration



On ISP Router A, you create a different key for ISP Routers B, C, and D and assign them to their respective IP addresses using RSVP commands. On the other routers, create a key to communicate with ISP Router A's IP address.

Key Chains

For each RSVP neighbor, you can configure a list of keys with specific IDs that are unique and have different lifetimes so that keys can be changed at predetermined intervals automatically without any disruption of service. Automatic key rotation enhances network security by minimizing the problems that could result if an untrusted source obtained, deduced, or guessed the current key.

Note If you use overlapping time windows for your key lifetimes, RSVP asks the Cisco IOS software key manager component for the next live key starting at time T. The key manager walks the keys in the chain until it finds the first one with start time S and end time E such that S <= T <= E. Therefore, the key with the smallest value (E–T) may not be used next.

Benefits of RSVP Message Authentication

Improved Security

The RSVP Message Authentication feature greatly reduces the chance of an RSVP-based spoofing attack and provides a secure method to control QoS access to a network.

Multiple Environments

The RSVP Message Authentication feature can be used in traffic engineering (TE) and non-TE environments as well as with the subnetwork bandwidth manager (SBM).

Multiple Platforms and Interfaces

The RSVP Message Authentication feature can be used on any supported RSVP platform or interface.

How to Configure RSVP Message Authentication

The following configuration parameters instruct RSVP on how to generate and verify integrity objects in various RSVP messages.

Note There are two configuration procedures: full and minimal. There are also two types of authentication procedures: interface and neighbor.

Per-Interface Authentication—Full Configuration

Perform the following procedures for a full configuration for per-interface authentication:

- Enabling RSVP on an Interface, page 6 (required)
- Configuring an RSVP Authentication Type, page 7 (optional)
- Configuring an RSVP Authentication Key, page 8 (required)
- Enabling RSVP Key Encryption, page 10 (optional)
- Enabling RSVP Authentication Challenge, page 10 (optional)
- Configuring RSVP Authentication Lifetime, page 12 (optional)
- Configuring RSVP Authentication Window Size, page 13 (optional)
- Activating RSVP Authentication, page 15 (required)
- Verifying RSVP Message Authentication, page 16 (optional)

Per-Interface Authentication—Minimal Configuration

Perform the following tasks for a minimal configuration for per-interface authentication:

- Enabling RSVP on an Interface, page 6 (required)
- Configuring an RSVP Authentication Key, page 8 (required)
- Activating RSVP Authentication, page 15 (required)

Per-Neighbor Authentication—Full Configuration

Perform the following procedures for a full configuration for per-neighbor authentication:

- Configuring an RSVP Authentication Type, page 7 (optional)
- Enabling RSVP Authentication Challenge, page 10 (optional)
- Enabling RSVP Key Encryption, page 10 (optional)
- Configuring RSVP Authentication Lifetime, page 12 (optional)
- Configuring RSVP Authentication Window Size, page 13 (optional)
- Activating RSVP Authentication, page 15 (required)
- Verifying RSVP Message Authentication, page 16 (optional)
- Configuring a Key Chain, page 17 (required)
- Binding a Key Chain to an RSVP Neighbor, page 18 (required)

Per-Neighbor Authentication—Minimal Configuration

Perform the following tasks for a minimal configuration for per-neighbor authentication:

- Activating RSVP Authentication, page 15 (required)
- Configuring a Key Chain, page 17 (required)
- Binding a Key Chain to an RSVP Neighbor, page 18 (required)

Enabling RSVP on an Interface

Perform this task to enable RSVP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-kbps* [*single-flow-kbps*]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured.
Step 4	**ip rsvp bandwidth** [*interface-kbps* [*single-flow-kbps*]] **Example:** `Router(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10,000,000. **Note** Repeat this command for each interface that you want to enable.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring an RSVP Authentication Type

Perform this task to configure an RSVP authentication type.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication type** {**md5** | **sha-1**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`interface` *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step if you are configuring an authentication type for a neighbor or setting a global default.

	Command or Action	Purpose

Step 4	**ip rsvp authentication type** {**md5** | **sha-1**}
	
	Example:
	For interface authentication:
	`Router(config-if)# ip rsvp authentication type sha-1`
	For neighbor authentication:
	`Router(config)# ip rsvp authentication neighbor address 10.1.1.1 type sha-1`
	or
	`Router(config)# ip rsvp authentication neighbor access-list 1 type sha-1`
	For a global default:
	`Router(config)# ip rsvp authentication type sha-1`
		Specifies the algorithm used to generate cryptographic signatures in RSVP messages on an interface or globally.
		• The algorithms are **md5**, the default, and **sha-1**, which is newer and more secure than **md5**.
		Note Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.

Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring an RSVP Authentication Key

Perform this task to configure an RSVP authentication key.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication key** *passphrase*
5. **exit**
6. **ip rsvp authentication key-chain** *chain*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode. **Note** If you want to configure a key, proceed to Step 3; if you want to configure a key chain, proceed to Step 6.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step and go to Step 6 if you want to configure only a key chain.
Step 4	**ip rsvp authentication key** *passphrase* **Example:** `Router(config-if)# ip rsvp authentication key 11223344`	Specifies the data string (key) for the authentication algorithm. • The key consists of 8 to 40 characters. It can include spaces and multiple words. It can also be encrypted or appear in clear text when displayed. **Note** Omit this step if you want to configure a key chain.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.

Step 6	**ip rsvp authentication key-chain** *chain*
	
	Example:
	For neighbor authentication:
	`Router(config)# ip rsvp authentication neighbor address 10.1.1.1 key-chain xzy`
	or
	`Router(config)# ip rsvp authentication neighbor access-list 1 key-chain xzy`
	For a global default:
	`Router(config)# ip rsvp authentication key-chain xzy`
		Specifies the data string (key chain) for the authentication algorithm.
		• The key chain must have at least one key, but can have up to 2,147,483647 keys.
		Note You cannot use the **ip rsvp authentication key** and the **ip rsvp authentication key-chain** commands on the same router interface. The commands supersede each other; however, no error message is generated.
		Note Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.

Step 7	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling RSVP Key Encryption

Perform this task to enable RSVP key encryption when the key is stored in the router configuration. (This prevents anyone from seeing the clear text key in the configuration file.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key config-key 1** *string*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**key config-key 1** *string* **Example:** `Router(config)# key config-key 1 11223344`	Enables key encryption in the configuration file. **Note** The *string* argument can contain up to eight alphanumeric characters.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling RSVP Authentication Challenge

Perform this task to enable RSVP authentication challenge.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication challenge**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step if you are configuring an authentication challenge for a neighbor or setting a global default.

Step 4	**ip rsvp authentication challenge**
	
	Example:
	For interface authentication:
	`Router(config-if)# ip rsvp authentication challenge`
	For neighbor authentication:
	`Router(config)# ip rsvp authentication neighbor address 10.1.1.1 challenge`
	or
	`Router(config)# ip rsvp authentication neighbor access-list 1 challenge`
	For a global default:
	`Router(config)# ip rsvp authentication challenge`
		Makes RSVP perform a challenge-response handshake on an interface or globally when RSVP learns about any new challenge-capable neighbors on a network.
		Note Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.

Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring RSVP Authentication Lifetime

Perform this task to configure the lifetimes of security associations between RSVP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication lifetime** *hh*:*mm*:*ss*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0` **Note** Omit this step if you are configuring an authentication lifetime for a neighbor or setting a global default.	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured.

	Command or Action	Purpose

Step 4	**ip rsvp authentication lifetime** *hh:mm:ss*
	
	Example:
	For interface authentication:
	`Router(config-if)# ip rsvp authentication lifetime 00:05:00`
	For neighbor authentication:
	`Router(config)# ip rsvp authentication neighbor address 10.1.1.1 lifetime 00:05:00`
	or
	`Router(config)# ip rsvp authentication neighbor access-list 1 lifetime 00:05:00`
	For a global default:
	`Router(config)# ip rsvp authentication 00:05:00`
		Controls how long RSVP maintains security associations with RSVP neighbors on an interface or globally.
		• The default security association for **hh:mm:ss** is 30 minutes; the range is 1 second to 24 hours.
		Note Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.

Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring RSVP Authentication Window Size

Perform this task to configure the RSVP authentication window size.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication window-size** *n*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step if you are configuring a window size for a neighbor or setting a global default.

Step 4	**ip rsvp authentication window-size** *n*
	
	Example:
	For interface authentication:
	`Router(config-if)# ip rsvp authentication window-size 2`
	For neighbor authentication:
	`Router(config)# ip rsvp authentication neighbor address 10.1.1.1 window-size 2`
	or
	`Router(config)# ip rsvp authentication neighbor access-list 1 window-size`
	For a global default:
	`Router(config)# ip rsvp authentication window-size 2`
		Specifies the maximum number of authenticated messages that can be received out of order on an interface or globally.
		• The default value is one message; the range is 1 to 64 messages.
		Note Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.

Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Activating RSVP Authentication

Perform this task to activate RSVP authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured. **Note** Omit this step if you are configuring authentication for a neighbor or setting a global default.

	Command or Action	Purpose

Step 4	**ip rsvp authentication**
	
	Example:
	For interface authentication:
	`Router(config-if)# ip rsvp authentication`
	
	For neighbor authentication:
	`Router(config)# ip rsvp authentication neighbor address 10.1.1.1`
	or
	`Router(config)# ip rsvp authentication neighbor access-list 1`
	
	For a global default:
	`Router(config)# ip rsvp authentication`
		Activates RSVP cryptographic authentication on an interface or globally.
		
		Note Omit the **neighbor address** *address* or the **neighbor access-list** *acl-name* or *acl-number* to set the global default.

Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Verifying RSVP Message Authentication

Perform this task to verify that the RSVP Message Authentication feature is functioning.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp interface** [**detail**] [*interface-type interface-number*]
3. **show ip rsvp authentication** [**detail**] [**from** {*ip-address* | *hostname*}] [**to** {*ip-address* | *hostname*}]
4. **show ip rsvp counters** [**authentication** | **interface** *interface-unit* | **neighbor** | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp interface** [**detail**] [*interface-type interface-number*] **Example:** `Router# show ip rsvp interface detail`	Displays information about interfaces on which RSVP is enabled, including the current allocation budget and maximum available bandwidth. • The optional **detail** keyword displays the bandwidth, signaling, and authentication parameters.

	Command or Action	Purpose
Step 3	**show ip rsvp authentication** [**detail**] [**from** {*ip-address* | *hostname*}] [**to** {*ip-address* | *hostname*}] **Example:** `Router# show ip rsvp authentication detail`	Displays the security associations that RSVP has established with other RSVP neighbors. • The optional **detail** keyword displays state information that includes IP addresses, interfaces enabled, and configured cryptographic authentication parameters about security associations that RSVP has established with neighbors.

Step 4	**show ip rsvp counters** [**authentication** | **interface** *interface-unit* | **neighbor** | **summary**]
	
	Example:
	`Router# show ip rsvp counters summary`
	
	`Router# show ip rsvp counters authentication`
		Displays all RSVP counters.
		Note The errors counter increments whenever an authentication error occurs, but can also increment for errors not related to authentication.
		• The optional **authentication** keyword shows a list of RSVP authentication counters.
		• The optional **interface** *interface-unit* keyword argument combination shows the number of RSVP messages sent and received by the specific interface.
		• The optional **neighbor** keyword shows the number of RSVP messages sent and received by the specific neighbor.
		• The optional **summary** keyword shows the cumulative number of RSVP messages sent and received by the router. It does not print per-interface counters.

Configuring a Key Chain

Perform this task to configure a key chain for neighbor authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. {**key** [*key-ID*] | **key-string** [*text*] | **accept-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}] | **send-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Router(config)# key chain neighbor_V`	Enters key-chain mode.

Step 4	{**key** [*key-ID*] | **key-string** [*text*] | **accept-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}] | **send-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}]
	
	Example:
	`Router(config-keychain)# key 1`
	
	`Router(config-keychain)# key-string ABcXyz`
		Selects the parameters for the key chain. (These are submodes.)
		Note For details on these parameters, see the *Cisco IOS IP Command Reference, Volume 2 of 4, Routing Protocols, Release 12.3T.*
		Note **accept-lifetime** is ignored when a key chain is assigned to RSVP.

Step 5	**end** **Example:** `Router(config-keychain)# end`	Returns to privileged EXEC mode.

Binding a Key Chain to an RSVP Neighbor

Perform this task to bind a key chain to an RSVP neighbor for neighbor authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp authentication neighbor address** *address* **key-chain** *key-chain-name*

   or

   ip rsvp authentication neighbor access-list *acl-name* or *acl-number* **key-chain** *key-chain-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp authentication neighbor address** *address* **key-chain** *key-chain-name* or **ip rsvp authentication neighbor access-list** *acl-name* or *acl-number* **key-chain** *key-chain-name* **Example:** `Router(config)# ip rsvp authentication neighbor access-list 1 key-chain neighbor_V`	Binds a key chain to an IP address or to an ACL and enters key-chain mode. **Note** If you are using an ACL, you must create it before you bind it to a key chain. See the **ip rsvp authentication** command in the Command Reference section for examples.
Step 4	**end** **Example:** `Router(config-keychain)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

After you enable RSVP authentication, RSVP logs system error events whenever an authentication check fails. These events are logged instead of just being displayed when debugging is enabled because they may indicate potential security attacks. The events are generated when:

- RSVP receives a message that does not contain the correct cryptographic signature. This could be due to misconfiguration of the authentication key or algorithm on one or more RSVP neighbors, but it may also indicate an (unsuccessful) attack.
- RSVP receives a message with the correct cryptographic signature, but with a duplicate authentication sequence number. This may indicate an (unsuccessful) message replay attack.
- RSVP receives a message with the correct cryptographic signature, but with an authentication sequence number that is outside the receive window. This could be due to a reordered burst of valid RSVP messages, but it may also indicate an (unsuccessful) message replay attack.
- Failed challenges result from timeouts or bad challenge responses.

To troubleshoot the RSVP Message Authentication feature, use the following commands in privileged EXEC mode.

Command	Purpose
`Router# debug ip rsvp authentication`	Displays output related to RSVP authentication.
`Router# debug ip rsvp dump signalling`	Displays brief information about signaling (Path and Resv) messages.
`Router# debug ip rsvp errors`	Displays error events including authentication errors.

Configuration Examples for RSVP Message Authentication

This section provides the following configuration examples:

- RSVP Message Authentication Per-Interface: Example, page 20
- RSVP Message Authentication Per-Neighbor: Example, page 22

RSVP Message Authentication Per-Interface: Example

In the following example, the cryptographic authentication parameters, including type, key, challenge, lifetime, and window size are configured; and authentication is activated:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface e0/0
Router(config-if)# ip rsvp bandwidth 7500 7500
Router(config-if)# ip rsvp authentication type sha-1
Router(config-if)# ip rsvp authentication key 11223344
Router(config-if)# ip rsvp authentication challenge
Router(config-if)# ip rsvp authentication lifetime 00:30:05
Router(config-if)# ip rsvp authentication window-size 2
Router(config-if)# ip rsvp authentication
```

In the following output from the **show ip rsvp interface detail** command, notice the cryptographic authentication parameters that you configured for the Ethernet0/0 interface:

```
Router# show ip rsvp interface detail

Et0/0:
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 7500K bits/sec
     Max. allowed (per flow): 7500K bits/sec
     Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Neighbors:
     Using IP encap: 0.  Using UDP encap: 0
   Signalling:
     Refresh reduction: disabled
   Authentication: enabled
     Key:         11223344
     Type:        sha-1
     Window size: 2
     Challenge:   enabled
```

In the preceding example, the authentication key appears in clear text. If you enter the **key-config-key 1** *string* command, the key appears encrypted, as in the following example:

```
Router# show ip rsvp interface detail

 Et0/0:
   Bandwidth:
     Curr allocated: 0 bits/sec
Max. allowed (total): 7500K bits/sec
     Max. allowed (per flow): 7500K bits/sec
     Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Neighbors:
     Using IP encap: 0.  Using UDP encap: 0
   Signalling:
     Refresh reduction: disabled
   Authentication: enabled
     Key:         <encrypted>
     Type:        sha-1
     Window size: 2
     Challenge:   enabled
```

In the following output, notice that the authentication key changes from encrypted to clear text after the **no key config-key 1** command is issued:

```
Router# show running-config interface e0/0

Building configuration...

Current configuration :247 bytes
!
interface Ethernet0/0
 ip address 192.168.101.2 255.255.255.0
 no ip directed-broadcast
 ip pim dense-mode
 no ip mroute-cache
 no cdp enable
 ip rsvp bandwidth 7500 7500
 ip rsvp authentication key 7>70>9:7<872>?74
 ip rsvp authentication
end
```

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# no key config-key 1
Router(config)# end


Router# show running-config
*Jan 30  08:02:09.559:%SYS-5-CONFIG_I:Configured from console by console
int e0/0
Building configuration...

Current configuration :239 bytes
!
interface Ethernet0/0
 ip address 192.168.101.2 255.255.255.0
 no ip directed-broadcast
 ip pim dense-mode
 no ip mroute-cache
 no cdp enable
 ip rsvp bandwidth 7500 7500
 ip rsvp authentication key 11223344
 ip rsvp authentication
end
```

RSVP Message Authentication Per-Neighbor: Example

In the following example, a key chain with two keys for each neighbor is defined, then an access list and a key chain are created for neighbors V, Y, and Z and authentication is explicitly enabled for each neighbor and globally. However, only the neighbors specified will have their messages accepted; messages from other sources will be rejected. This enhances network security.

For security reasons, you should change keys on a regular basis. When the first key expires, the second key automatically takes over. At that point, you should change the first key's key-string to a new value and then set the send lifetimes to take over after the second key expires. The router will log an event when a key expires to remind you to update it.

The lifetimes of the first and second keys for each neighbor overlap. This allows for any clock synchronization problems that might cause the neighbors not to switch keys at the right time. You can avoid these overlaps by configuring the neighbors to use Network Time Protocol (NTP) to synchronize their clocks to a time server.

For an MPLS/TE configuration, physical addresses and router IDs are given.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# key chain neighbor_V
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string R72*UiAXy
Router(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# key 2
Router(config-keychain-key)# key-string Pl349&DaQ
Router(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# key chain neighbor_Y
Router(config-keychain)# key 3
Router(config-keychain-key)# key-string *ZXFwR!03
Router(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# key 4
```

```
Router(config-keychain-key)# key-string UnGR8f&lOmY
Router(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# key chain neighbor_Z
Router(config-keychain)# key 5
Router(config-keychain-key)# key-string P+T=77&/M
Router(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# key 6
Router(config-keychain-key)# key-string payattention2me
Router(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Router(config-keychain-key)# exit
Router(config-keychain)# exit
Router(config)# end
```

Note You can use the **key-config-key 1** *string* command to encrypt key chains for an interface, a neighbor, or globally.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip access-list standard neighbor_V
Router(config-std-nacl)# permit 10.0.0.1 <------- physical address
Router(config-std-nacl)# permit 10.0.0.2 <------- physical address
Router(config-std-nacl)# permit 10.0.0.3 <------- router ID
Router(config-std-nacl)# exit
Router(config)# ip access-list standard neighbor_Y
Router(config-std-nacl)# permit 10.0.0.4 <------- physical address
Router(config-std-nacl)# permit 10.0.0.5 <------- physical address
Router(config-std-nacl)# permit 10.0.0.6 <------- router ID
Router(config-std-nacl)# exit
Router(config)# ip access-list standard neighbor_Z
Router(config-std-nacl)# permit 10.0.0.7 <------- physical address
Router(config-std-nacl)# permit 10.0.0.8 <------- physical address
Router(config-std-nacl)# permit 10.0.0.9 <------- router ID
Router(config-std-nacl)# exit
Router(config)# ip rsvp authentication neighbor access-list neighbor_V key-chain
neighbor_V
Router(config)# ip rsvp authentication neighbor access-list neighbor_Y key-chain
neighbor_Y
Router(config)# ip rsvp authentication neighbor access-list neighbor_Z key-chain
neighbor_Z
Router(config)# ip rsvp authentication
Router(config)# end
```

Additional References

The following sections provide references related to the RSVP Message Authentication feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	• *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.4T • *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2SR • *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.2SX
QoS features including signaling, classification, and congestion management	*Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.4
Inter-AS features including local policy support and per-neighbor keys authentication	*MPLS Traffic Engineering—Inter-AS-TE* feature module
Error messages	*Cisco IOS Software System Error Messages*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1321	*The MD5 Message Digest Algorithm*
RFC 2104	*HMAC: Keyed-Hashing for Messaging Authentication*
RFC 2205	*Resource Reservation Protocol*
RFC 2209	*RSVP—Version 1 Message Processing Rules*

RFCs	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2747	*RSVP Cryptographic Authentication*
RFC 3097	*RSVP Crytographic Authentication—Updated Message Type Value*
RFC 3174	*US Secure Hash Algorithm 1 (SHA1)*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

This feature uses no new or modified commands.

- **clear ip rsvp authentication**
- **debug ip rsvp authentication**
- **ip rsvp authentication**
- **ip rsvp authentication challenge**
- **ip rsvp authentication key**
- **ip rsvp authentication key-chain**
- **ip rsvp authentication lifetime**
- **ip rsvp authentication neighbor**
- **ip rsvp authentication type**
- **ip rsvp authentication window-size**
- **show ip rsvp authentication**
- **show ip rsvp counters**
- **show ip rsvp interface**

clear ip rsvp authentication

To eliminate Resource Reservation Protocol (RSVP) security associations before their lifetimes expire, use the **clear ip rsvp authentication** command in privileged EXEC mode.

clear ip rsvp authentication [*ip-address* | *hostname*]

Syntax Description

ip-address	(Optional) Frees security associations with a specific neighbor.
hostname	(Optional) Frees security associations with a specific host.

Note The difference between the *ip-address* and *hostname* arguments is the difference of specifying the neighbor by its IP address or by its name.

Command Default

The default behavior is to clear all security associations.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **clear ip rsvp authentication** command for the following reasons:

- To eliminate security associations before their lifetimes expire
- To free up memory
- To resolve a problem with a security association being in some indeterminate state
- To force reauthentication of neighbors

You can delete all RSVP security associations if you do not enter an IP address or a hostname, or just the ones with a specific RSVP neighbor or host.

If you delete a security association, it is re-created as needed when the trusted RSVP neighbors start sending more RSVP messages.

Examples

The following command shows how to clear all security associations before they expire:

```
Router# clear ip rsvp authentication
```

Related Commands

Command	Description
ip rsvp authentication lifetime	Controls how long RSVP maintains security associations with other trusted RSVP neighbors.
show ip rsvp authentication	Displays the security associations that RSVP has established with other RSVP neighbors.

debug ip rsvp authentication

To display debugging output related to Resource Reservation Protocol (RSVP) authentication, use the **debug ip rsvp authentication** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ip rsvp authentication

no debug ip rsvp authentication

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

After you enable RSVP authentication, RSVP logs system error events whenever an authentication check fails. These events are logged instead of just being displayed when debugging is enabled because they may indicate potential security attacks. The events are generated when:

- RSVP receives a message that does not contain the correct cryptographic signature. This could be due to misconfiguration of the authentication key or algorithm on one or more RSVP neighbors, but it may also indicate an (unsuccessful) attack.
- RSVP receives a message with the correct cryptographic signature, but with a duplicate authentication sequence number. This may indicate an (unsuccessful) message replay attack.
- RSVP receives a message with the correct cryptographic signature, but with an authentication sequence number that is outside the receive window. This could be due to a reordered burst of valid RSVP messages, but it may also indicate an (unsuccessful) message replay attack.
- Failed challenges result from timeouts or bad challenge responses.

Examples

The following example shows output from the **debug ip rsvp authentication** command in which the authentication type (digest) and the sequence number have been validated:

```
Router# debug ip rsvp authentication

RSVP authentication debugging is on
```

```
Router# show debugging

*Jan 30 08:10:46.335:RSVP_AUTH:Resv integrity digest from 192.168.101.2 valid
*Jan 30 08:10:46.335:RSVP_AUTH:Resv integrity sequence number 13971113505298841601 from
192.168.101.2 valid
*Jan 30 08:10:46.335:RSVP_AUTH:Resv from 192.168.101.2 passed all authentication checks
```

Note Cisco routers using RSVP authentication on Cisco IOS software ideally should have clocks that can be accurately restored to the correct time when the routers boot. This capability is available on certain Cisco routers that have clocks with battery backup. For those platforms that do not have battery backup, consider configuring the router to keep its clock synchronized with a Network Time Protocol (NTP) time server. Otherwise, if two adjacent routers have been operating with RSVP authentication enabled and one of them reboots such that its clock goes backward in time, it is possible (but unlikely) the router that did not reboot will log RSVP authentication sequence number errors.

Related Commands

Command	Description
ip rsvp authentication	Activates RSVP cryptographic authentication.
show debugging	Displays active debug output.

ip rsvp authentication

To activate Resource Reservation Protocol (RSVP) cryptographic authentication, use the **ip rsvp authentication** command in interface configuration mode. To deactivate authentication, use the **no** form of this command.

ip rsvp authentication

no ip rsvp authentication

Syntax Description

This command has no arguments or keywords.

Command Default

RSVP cryptographic authentication is deactivated.

Command Modes

Interface configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **ip rsvp authentication** command to deactivate and then reactivate RSVP authentication without reentering the other RSVP authentication configuration commands. You should not enable authentication unless you have previously configured a key. If you issue this command before the **ip rsvp authentication key** command, you get a warning message indicating that RSVP discards all messages until you specify a key. The **no ip rsvp authentication** command disables RSVP cryptographic authentication. However, the command does not automatically remove any other authentication parameters that you have configured. You must issue a specific **no ip rsvp authentication** command; for example, **no ip rsvp authentication key**, **no ip rsvp authentication type**, or **no ip rsvp authentication window-size**, if you want to remove them from the configuration.

The **ip rsvp authentication** command is similar to the **ip rsvp neighbor** command. However, the **ip rsvp authentication** command provides better authentication and performs system logging.

Examples

The following command activates authentication on an interface:

```
Router(config-if)# ip rsvp authentication
```

The following command deactivates authentication on an interface:

```
Router(config-if)# no ip rsvp authentication
```

Related Commands

Command	Description
ip rsvp authentication key	Specifies the key (string) for the RSVP authentication algorithm.
ip rsvp authentication type	Specifies the algorithm used to generate cryptographic signatures in RSVP messages.
ip rsvp authentication window-size	Specifies the maximum number of RSVP authenticated messages that can be received out of order.
ip rsvp neighbor	Enables neighbors to request a reservation.

ip rsvp authentication challenge

To make Resource Reservation Protocol (RSVP) perform a challenge-response handshake with any new RSVP neighbors on a network, use the **ip rsvp authentication challenge** command in interface configuration mode. To disable the challenge-response handshake, use the **no** form of this command.

ip rsvp authentication challenge

no ip rsvp authentication challenge

Syntax Description

This command has no arguments or keywords.

Command Default

The challenge-response handshake initiated by this command is disabled.

Command Modes

Interface configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

The **ip rsvp authentication challenge** command requires RSVP to perform a challenge-response handshake with any new RSVP neighbors that are discovered on a network. Such a handshake allows the router to thwart RSVP message replay attacks while booting, especially if there is a long period of inactivity from trusted RSVP neighbors following the reboot. If messages from trusted RSVP neighbors arrive very quickly after the router reboots, then challenges may not be required because the router will have reestablished its security associations with the trusted nodes before the untrusted nodes can attempt replay attacks.

If you enable RSVP authentication globally on an interface over which a Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) label switched path (LSP) travels and the router on which authentication is enabled experiences a stateful switchover (SSO), the following occurs:

- If challenges are disabled (you did not specify the **ip rsvp authentication challenge** command), the LSP recovers properly.
- If challenges are enabled (you specified the **ip rsvp authentication challenge** command), more RSVP signaling messages are required and the LSP takes longer to recover or the forwarding state may time out and the LSP does not recover. If a timeout occurs, data packet forwarding is interrupted while the headend router signals a new LSP.

If you enable RSVP authentication challenges, you should consider enabling RSVP refresh reduction by using the **ip rsvp signalling refresh reduction** command. While a challenge handshake is in progress, the receiving router that is initiating the handshake discards all RSVP messages from the node that is being challenged until the handshake-initiating router receives a valid challenge response.

Note If a neighbor does not reply to the first challenge message after 1 second, the Cisco IOS software sends another challenge message and waits 2 seconds. If no response is received to the second challenge, the Cisco IOS software sends another and waits 4 seconds. If no response to the third challenge is received, the Cisco IOS software sends a fourth challenge and waits 8 seconds. If there is no response to the fourth challenge, the Cisco IOS software stops the current challenge to that neighbor, logs a system error message, and does not create a security association for that neighbor. This kind of exponential backoff is used to recover from challenges dropped by the network or busy neighbors.

Activating refresh reduction enables the challenged node to resend dropped messages more quickly once the handshake has completed. This causes RSVP to reestablish reservation state faster when the router reboots.

Enable authentication challenges wherever possible to reduce the router's vulnerability to replay attacks.

Examples

The following command shows how to enable RSVP to perform a challenge-response handshake:

```
Router(config-if)# ip rsvp authentication challenge
```

Related Commands

Command	Description
ip rsvp signalling refresh reduction	Enables RSVP refresh reduction.

ip rsvp authentication key

To specify the key (string) for the Resource Reservation Protocol (RSVP) authentication algorithm, use the **ip rsvp authentication key** command in interface configuration mode. To disable the key, use the **no** form of this command.

ip rsvp authentication key *pass-phrase*

no ip rsvp authentication key

Syntax Description

pass-phrase	Phrase that ranges from 8 to 40 characters. See "Usage Guidelines" for additional information.

Command Default

No key is specified.

Command Modes

Interface configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **ip rsvp authentication key** command to select the key for the authentication algorithm. This key is a passphrase of 8 to 40 characters. It can include spaces; quotes are not required if spaces are used. The key can consist of more than one word. We recommend that you make the passphrase as long as possible. This key must be the same for all RSVP neighbors on this interface. As with all passwords, you should choose them carefully so that attackers cannot easily guess them.

Here are some guidelines:

- Use a mixture of upper- and lowercase letters, digits, and punctuation.
- If using just a single word, do not use a word contained in any dictionary of any language, spelling lists, or other lists of words.
- Use something easily remembered so you do not have to write it down.
- Do not let it appear in clear text in any file or script or on a piece of paper attached to a terminal.

By default, RSVP authentication keys are stored in clear text in the router configuration file, but they can optionally be stored as encrypted text in the configuration file. To enable key encryption, use the global configuration **key config-key 1** *string* command. After you enter this command, the passphrase parameter of each **ip rsvp authentication key** command is encrypted with the Data Encryption Standard (DES) algorithm when you save the configuration file. If you later issue a **no key config-key 1** *string* command, the RSVP authentication key is stored in clear text again when you save the configuration.

The *string* argument is not stored in the configuration file; it is stored only in the router's private NVRAM and will not appear in the output of a **show running-config** or **show config** command. Therefore, if you copy the configuration file to another router, any encrypted RSVP keys in that file will not be successfully decrypted by RSVP when the router boots and RSVP authentication will not operate correctly. To recover from this, follow these steps on the new router:

1. For each RSVP interface with an authentication key, issue a **no ip rsvp authentication key** command to clear the old key.
2. For that same set of RSVP interfaces, issue an **ip rsvp authentication key** command to reconfigure the correct clear text keys.
3. Issue a global **key config-key 1** *string* command to reencrypt the RSVP keys for the new router.
4. Save the configuration.

Examples

The following command sets the passphrase to 11223344 in clear text:

```
Router(config-if)# ip rsvp authentication key 11223344
```

To encrypt the authentication key, issue the **key config-key 1** *string* command as follows:

```
Router# configure terminal
Router(config)# key config-key 1 11223344
Router(config)# end
```

Related Commands

Command	Description
key config-key	Defines a private DEF key for the router.

ip rsvp authentication key-chain

To specify a list of keys for the Resource Reservation Protocol (RSVP) neighbors, use the **ip rsvp authentication key-chain** command in global configuration mode. To disable the key chain, use the **no** form of this command. To set the key chain to its default, use the **default** form of this command.

ip rsvp authentication key-chain *string*

no ip rsvp authentication key-chain

default ip rsvp authentication key-chain

Syntax Description

string	Name of key chain; must have at least one key, but can have up to 2,147,483647 keys.

Command Default

No key chain is specified.

Command Modes

Global configuration

Command History

Release	Modification
12.0(29)S	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **ip rsvp authentication key-chain** command to select the key chain.

Note You cannot use the **ip rsvp authentication key** and the **ip rsvp authentication key-chain** commands on the same router interface. The commands supersede each other; however, no error message is generated.

Examples

The following commands set the key chain to RSVPkey for neighbor authentication:

```
Router(config)# ip rsvp authentication neighbor address 10.1.1.1 key-chain RSVPkey
```

or

```
Router(config)# ip rsvp authentication neighbor access-list 1 key-chain RSVPkey
```

The following command sets the global default key chain to RSVPkey:

```
Router(config)# ip rsvp authentication key-chain RSVPkey
```

Related Commands

Command	Description
ip rsvp authentication key	Specifies the interface key (string) for the RSVP authentication algorithm.
show key chain	Displays authentication key information.

ip rsvp authentication lifetime

To control how long Resource Reservation Protocol (RSVP) maintains security associations with other trusted RSVP neighbors, use the **ip rsvp authentication lifetime** command in interface configuration mode. To disable the lifetime setting, use the **no** form of this command.

ip rsvp authentication lifetime *hh*:*mm*:*ss*

no ip rsvp authentication lifetime *hh*:*mm*:*ss*

Syntax Description

hh:*mm*:*ss*	Hours: minutes: seconds that RSVP maintains security associations with other trusted RSVP neighbors. The range is 1 second to 24 hours. The default is 30 minutes.

Command Default

If you do not specify a security association lifetime setting, 30 minutes is used.

Command Modes

Interface configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **ip rsvp authentication lifetime** command to indicate when to end security associations with RSVP trusted neighbors. If an association's lifetime expires, but at least one valid, RSVP authenticated message was received in that time period, RSVP resets the security association's lifetime to this configured value. When a neighbor stops sending RSVP signaling messages (that is, the last reservation has been torn down), the memory used for the security association is freed as well as when the association's lifetime period ends. The association can be re-created if that RSVP neighbor resumes its signaling. Setting the lifetime to shorter periods allows memory to be recovered faster when the router is handling a lot of short-lived reservations. Setting the lifetime to longer periods reduces the workload on the router when establishing new authenticated reservations.

Use the **clear ip rsvp authentication** command to free security associations before their lifetimes expire.

Examples

The following command sets the lifetime period for 30 minutes and 5 seconds:

```
Router(config-if)# ip rsvp authentication lifetime 00:30:05
```

Related Commands

Command	Description
clear ip rsvp authentication	Eliminates RSVP security associations before their lifetimes expire.

ip rsvp authentication neighbor

To activate Resource Reservation Protocol (RSVP) cryptographic authentication for a neighbor, use the **ip rsvp authentication neighbor** command in global configuration mode. To deactivate authentication for a neighbor, use the **no** form of this command. To set this command to the global default, use the **default** form of this command.

ip rsvp authentication neighbor [{**access-list** *acl-name-or-number*} | {**address** *address*}] [**challenge**] [**key-chain** *name*] [**type** {**md5 | sha-1**}] [**window-size** *number-of-messages*]

no ip rsvp authentication neighbor

default ip rsvp authentication neighbor

Syntax Description

access-list *acl-name-or-number*	A standard numbered or named IP access list that describes the set of neighbor IP addresses that share this key.
address *address*	A single IP address for a specific neighbor; usually one of the neighbor's physical or logical (loopback) interfaces.
challenge	(Optional) Requires RSVP to perform a challenge-response handshake with an RSVP neighbor for which RSVP does not have an existing security association in memory.
key-chain *name*	(Optional) The name of a key chain that contains the set of keys to be used to communicate with the neighbor.
type	(Optional) The algorithm to generate cryptographic signatures in RSVP messages.
md5	(Optional) RSA Message Digest 5 algorithm.
sha-1	(Optional) National Institute of Standards and Technologies (NIST) Secure Hash Algorithm-1; it is newer and more secure than **md5**.
window-size *number-of-messages*	(Optional) The maximum number of authenticated messages that can be received out of order. The range is 1 to 64, with a default of 1.

Command Default

Neighbor cryptographic authentication is disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.0(29)S	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

If you omit the optional keywords, the **ip rsvp authentication neighbor** command enables RSVP cryptographic authentication for a neighbor. Using the optional keywords inherits the global defaults.

In order to enable per-neighbor authentication, you must issue the **ip rsvp authentication neighbor** command (or the **no ip rsvp authentication neighbor** command to disable authentication). If you issue the **ip rsvp authentication** command without **neighbor**, then this command enables authentication for all neighbors and interfaces, regardless of whether there are any per-neighbor or per-interface keys defined. If you issue the **ip rsvp authentication neighbor** command, then authentication is enabled only for that neighbor.

Access Control Lists

A single ACL can describe all the physical and logical interfaces that one neighbor can use to receive RSVP messages from a router; this can be useful when multiple routes exist between two neighbors. One ACL could also specify a number of different neighbors who, along with your router, will share the same key(s); however, this is generally not considered to be good network security practice.

If numbered, the ACL must be in the 1 to 99 range or the 1300 to 1999 range, giving a total of 798 numbered ACLs that can be used to configure neighbor keys (assuming some of them are not being used for other purposes). There is no enforced limit on the number of standard named IP ACLs. The IP addresses used in the ACL should contain at least the neighbor's physical interface addresses; router ID addresses can be added if necessary, especially when using Multi-Protocol Label Switching (MPLS) Traffic Engineering (TE).

The existing **ip access-list standard** command must be used for creating named or numbered standard IP ACLs for RSVP neighbors because standard ACLs deal with just source or destination addresses while extended ACLs deal with five tuples and are more complex to configure. The RSVP CLI returns an error message if any type of ACL other than standard is specified; for example,

```
Router(config)# ip rsvp authentication neighbor access-list 10 key-chain wednesday

% Invalid access list name.
RSVP error: unable to find/create ACL
```

Named standard IP ACLs are also recommended because you can include the neighbor router's hostname as part of the ACL name, thereby making it easy to identify the per-neighbor ACLs in your router configuration.

The RSVP CLI displays an error message if a valid named or numbered ACL is specified, but a nonexistent or invalid key chain has not been associated with it, since the lack of a key chain could cause RSVP messages to or from that neighbor to be dropped; for example,

```
Router(config)# ip rsvp authentication neighbor access-list myneighbor key-chain xyz

RSVP error: Invalid argument(s)
```

Key Chains

In the key-chain parameter, the keys are used in order of ascending expiration deadlines. The only restriction on the name is that it cannot contain spaces. The key-chain parameter is optional; that is, you could omit it if you were trying to change other optional authentication parameters for the RSVP neighbor. However, when searching for a key, RSVP ignores any **ip rsvp authentication neighbor access-list** command that does not include a key-chain parameter that refers to a valid key chain with at least one unexpired key.

Error and Warning Conditions

The RSVP CLI returns an error if any of the key IDs in the chain are duplicates of key IDs in any other chains already assigned to RSVP; for example,

```
Router(config)# ip rsvp authentication neighbor access-list myneighbor key-chain abc

RSVP error: key chains abc and xyz contain duplicate key ID 1
RSVP error: Invalid argument(s)
```

The RSVP CLI returns an error if the specified key chain does not exist or does not contain at least one unexpired key.

If a key chain is properly defined and RSVP later tries to send a message to that neighbor, but cannot find a valid, unexpired per-neighbor or per-interface key, RSVP generates the RSVP_AUTH_NO_KEYS_LEFT system message indicating that a key could not be obtained for that neighbor.

If the key chain contains keys with finite expiration times, RSVP generates the RSVP_AUTH_ONE_KEY_EXPIRED message to indicate when each key has expired.

If RSVP receives a message from a neighbor with the wrong digest type, it generates the RSVP_MSG_AUTH_TYPE_MISMATCH system message indicating that there is a digest type mismatch with that neighbor.

If RSVP receives a message that is a duplicate of a message already in the window or is outside the window, RSVP logs the BAD_RSVP_MSG_RCVD_AUTH_DUP or the BAD_RSVP_MSG_RCVD_AUTH_WIN error message indicating that the message sequence number is invalid.

If a challenge of a neighbor fails or times out, RSVP generates the BAD_RSVP_MSG_RCVD_AUTH_COOKIE system message or the RSVP_MSG_AUTH_CHALLENGE_TIMEOUT message, indicating that the specified neighbor failed to respond successfully to a challenge.

Examples

In the following example, an access list and a key chain are created for neighbors V, Y, and Z and authentication is enabled globally using inheritance for all other authentication parameters:

```
Router# configure terminal
Router(config)# ip access-list standard neighbor_V
Router(config-std-nacl)# permit 10.0.0.2
Router(config-std-nacl)# permit 10.1.16.1
Router(config-std-nacl)# exit
Router(config)# ip access-list standard neighbor_Y
Router(config-std-nacl)# permit 10.0.1.2
Router(config-std-nacl)# permit 10.16.0.1
Router(config-std-nacl)# exit
Router(config)# ip access-list standard neighbor_Z
Router(config-std-nacl)# permit 10.16.0.2
Router(config-std-nacl)# permit 10.1.0.2
Router(config-std-nacl)# permit 10.0.1.2
Router(config-std-nacl)# exit
Router(config)# ip rsvp authentication neighbor access-list neighbor_V key-chain
neighbor_V
Router(config)# ip rsvp authentication neighbor access-list neighbor_Y key-chain
neighbor_Y
Router(config)# ip rsvp authentication neighbor access-list neighbor_Z key-chain
neighbor_Z
Router(config)# ip rsvp authentication
Router(config)# end
```

In the following example, an access list and a key chain are created for neighbors V, Y, and Z and authentication is explicitly enabled for each neighbor:

```
Router(config)# ip rsvp authentication neighbor access-list neighbor_V key-chain
neighbor_V
Router(config)# ip rsvp authentication neighbor access-list neighbor_V
Router(config)# ip rsvp authentication neighbor access-list neighbor_Y key-chain
neighbor_Y
Router(config)# ip rsvp authentication neighbor access-list neighbor_Y
Router(config)# ip rsvp authentication neighbor access-list neighbor_Z key-chain
neighbor_Z
Router(config)# ip rsvp authentication neighbor access-list neighbor_Z
Router(config)# end
```

Related Commands

Command	Description
ip rsvp authentication	Activates RSVP cryptographic authentication.

ip rsvp authentication type

To specify the type of algorithm used to generate cryptographic signatures in Resource Reservation Protocol (RSVP) messages, use the **ip rsvp authentication type** command in interface configuration or global configuration mode. To specify that no type of algorithm is used, use the **no** form of this command. To remove the type from your configuration, use the **default** form of this command.

Note Before you use the **no ip rsvp authentication type** command, see the "Usage Guidelines" section for more information.

Syntax for T Releases

ip rsvp authentication type {**md5** | **sha-1**}

no ip rsvp authentication type

default ip rsvp authentication type

Syntax for 12.0S and 12.2S Releases

ip rsvp authentication type {**md5 | sha-1**}

default ip rsvp authentication type

Syntax Description

md5	RSA Message Digest 5 algorithm.
sha-1	National Institute of Standards and Technologies (NIST) Secure Hash Algorithm-1; it is newer and more secure than MD5.

Command Default

If no algorithm is specifed, **md5** is used.

Command Modes

Interface configuration
Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.0(29)S	This command was introduced in global configuration mode for all neighbors. A **default** form of the command was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **ip rsvp authentication type** command to specify the algorithm to generate cryptographic signatures in RSVP messages. If you do not specify an algorithm, **md5** is used.

If you use the **ip rsvp authentication type** command rather than the **ip rsvp authentication neighbor type** command, the global default for type changes.

The **no ip rsvp authentication type** command is not supported in Cisco IOS Releases 12.0S and 12.2S because every security association must have a digest type, and you cannot disable it. Use the **default ip rsvp authentication type** command to remove the authentication type from a configuration and force the type to its default.

Although the **no ip rsvp authentication type** command is supported in Cisco IOS T releases, the **default ip rsvp rsvp authentication type** command is recommended to remove the authentication type from a configuration and force the type to its default.

Examples

T Releases Example

The following command sets the type to **sha-1** for interface authentication:

```
Router(config-if)# ip rsvp authentication type sha-1
```

12.0S and 12.2S Releases Examples

The following commands set the type to **sha-1** for neighbor authentication:

```
Router(config)# ip rsvp authentication neighbor address 10.1.1.1 type sha-1
```

or

```
Router(config)# ip rsvp authentication neighbor access-list 1 type sha-1
```

The following command sets the global default type to sha-1 for authentication:

```
Router(config)# ip rsvp authentication type sha-1
```

Default Command Example

The following command removes the type from your configuration and forces the type to its default:

```
Router(config)# default ip rsvp authentication type
```

Related Commands

Command	Description
ip rsvp authentication key	Specifies the key (string) for the RSVP authentication algorithm.
ip rsvp authentication neighbor type	Sets the type for a specific neighbor.

ip rsvp authentication window-size

To specify the maximum number of Resource Reservation Protocol (RSVP) authenticated messages that can be received out of order, use the **ip rsvp authentication window-size** command in interface configuration mode. To disable the window size (or to use the default value of 1), use the **no** form of this command.

ip rsvp authentication window-size [*number-of-messages*]

no ip rsvp authentication window-size

Syntax Description

number-of-messages	(Optional) Maximum number of authenticated messages that can be received out of order. The range is 1 to 64; the default value is 1.

Command Default

If no window size is specified, a value of 1 is used.

Command Modes

Interface configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **ip rsvp authentication window-size** command to specify the maximum number of RSVP authenticated messages that can be received out of order. All RSVP authenticated messages include a sequence number that is used to prevent replays of RSVP messages.

With a default window size of one message, RSVP rejects any duplicate authenticated messages because they are assumed to be replay attacks. However, sometimes bursts of RSVP messages become reordered between RSVP neighbors. If this occurs on a regular basis, and you can verify that the node sending the burst of messages is trusted, you can use the **ip rsvp authentication window-size** command option to allow for the burst size such that RSVP will not discard such reordered bursts. RSVP will still check for duplicate messages within these bursts.

Examples

The following command sets the window size to 2:

```
Router(config-if)# ip rsvp authentication window-size 2
```

Related Commands

Command	Description
ip rsvp authentication	Activates RSVP cryptographic authentication.

show ip rsvp authentication

To display the security associations that Resource Reservation Protocol (RSVP) has established with other RSVP neighbors, use the **show ip rsvp authentication** command in user EXEC or privileged EXEC mode.

show ip rsvp authentication [**detail**] [**from** {*ip-address* | *hostname*}] [**to** {*ip-address* | *hostname*}]

Syntax Description

detail	(Optional) Displays additional information about RSVP security associations.
from	(Optional) Specifies the starting point of the security associations.
to	(Optional) Specifies the ending point of the security associations.
ip-address	(Optional) Information about a neighbor with a specified IP address.
hostname	(Optional) Information about a particular host.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.0(29)S	The optional **from** and **to** keywords were added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **show ip rsvp authentication** command to display the security associations that RSVP has established with other RSVP neighbors. You can display all security associations or specify an IP address or hostname of a particular RSVP neighbor, which restricts the size of the display.

The difference between the *ip-address* and *hostname* arguments is whether you specify the neighbor by its IP address or by its name.

Examples

The following is sample output from the **show ip rsvp authentication** command:

```
Router# show ip rsvp authentication

Codes: S - static, D - dynamic, N - neighbor, I -interface, C - chain
From            To              I/F      Mode    Key-Source Key-ID       Code
192.168.102.1   192.168.104.3   Et2/2    Send    RSVPKey    1            DNC
192.168.104.1   192.168.104.3   Et2/2    Send    RSVPKey    1            DNC
192.168.104.1   192.168.104.3   AT1/0.1  Send    RSVPKey    1            DNC
192.168.106.1   192.168.104.3   AT1/0.1  Send    RSVPKey    1            DNC
192.168.106.1   192.168.106.2   AT1/0.1  Send    RSVPKey    1            DNC
192.168.106.2   192.168.104.1   AT1/0.1  Receive RSVPKey    1            DNC
192.168.106.2   192.168.106.1   AT1/0.1  Receive RSVPKey    1            DNC
```

Table 1 describes the significant fields shown in the display.

Table 1 show ip rsvp authentication Field Descriptions

Field	Description
Codes	Keys can be either static (manually configured) or dynamic (created from a per-ACL key or obtained from a key management server such as Kerberos). Cisco IOS software does not currently support dynamic keys from key management servers. If the field contains the string per-neighbor, it means the security association is using a per-neighbor key; if the field contains the string per-interface, it means the security association is using a per-interface key. If the field contains the string chain, it means the key for the security association comes from the key chain specified in the Key Source.
From	Starting point of the security association.
To	Ending point of the security association.
I/F	Name and number of the interface over which the security association is being maintained.
Mode	Separate associations maintained for sending and receiving RSVP messages for a specific RSVP neighbor. Possible values are **Send** or **Receive**.
Key-Source	Indicates where the key was configured.
Key-ID	A string which, along with the IP address, uniquely identifies a security association. The key ID is automatically generated in Cisco IOS software by using the per-interface **ip rsvp authentication key** command, but it is configured in Cisco IOS software when using key chains for per-neighbor or per-interface RSVP keys. The key ID may be configurable on other RSVP platforms. A key ID is provided in every RSVP authenticated message initiated by a sender and is stored by every RSVP receiver. **Note** **Key Expired** in this field means that all possible keys used for this neighbor have expired.
Code	Indicates the type of key ID used.

The following is sample output from the **show ip rsvp authentication detail** command:

```
Router# show ip rsvp authentication detail

From:                   192.168.102.1
To:                     192.168.104.3
Neighbor:               192.168.102.2
Interface:              Ethernet2/2
Mode:                   Send
Key ID:                 1
Key ACL:                R2 (populated)
Key Source:             RSVPKey (enabled)
Key Type:               Dynamic per-neighbor chain
Handle:                 01000411
Hash Type:              MD5
Lifetime:               00:30:00
Expires:                00:17:08
Challenge:              Supported
Window size:            1
Last seq # sent:        14167519095569779135

From:                   192.168.104.1
To:                     192.168.104.3
```

```
Neighbor:               192.168.102.2
Interface:              Ethernet2/2
Mode:                   Send
Key ID:                 1
Key ACL:                R2 (populated)
Key Source:             RSVPKey (enabled)
Key Type:               Dynamic per-neighbor chain
Handle:                 0400040F
Hash Type:              MD5
Lifetime:               00:30:00
Expires:                00:22:06
Challenge:              Supported
Window size:            1
Last seq # sent:        14167520384059965440

From:                   192.168.104.1
To:                     192.168.104.3
Neighbor:               192.168.106.2
Interface:              ATM1/0.1
Mode:                   Send
Key ID:                 1
Key ACL:                R3 (populated)
Key Source:             RSVPKey (enabled)
Key Type:               Dynamic per-neighbor chain
Handle:                 02000404
Hash Type:              MD5
Lifetime:               00:30:00
Expires:                00:16:37
Challenge:              Supported
Window size:            1
Last seq # sent:        14167518979605659648

From:                   192.168.106.1
To:                     192.168.104.3
Neighbor:               192.168.106.2
Interface:              ATM1/0.1
Mode:                   Send
Key ID:                 1
Key ACL:                R3 (populated)
Key Source:             RSVPKey (enabled)
Key Type:               Dynamic per-neighbor chain
Handle:                 01000408
Hash Type:              MD5
Lifetime:               00:30:00
Expires:                00:11:37
Challenge:              Supported
Window size:            1
Last seq # sent:        14167517691115473376

From:                   192.168.106.1
To:                     192.168.106.2
Neighbor:               192.168.106.2
Interface:              ATM1/0.1
Mode:                   Send
Key ID:                 1
Key ACL:                R3 (populated)
Key Source:             RSVPKey (enabled)
Key Type:               Dynamic per-neighbor chain
Handle:                 8D00040E
Hash Type:              MD5
Lifetime:               00:30:00
Expires:                00:29:29
Challenge:              Supported
Window size:            1
```

```
Last seq # sent:        14167808344437293057

From:                   192.168.106.2
To:                     192.168.104.1
Neighbor:               192.168.106.2
Interface:              ATM1/0.1
Mode:                   Receive
Key ID:                 1
Key ACL:                R3 (populated)
Key Source:             RSVPKey (enabled)
Key Type:               Dynamic per-neighbor chain
Handle:                 CD00040A
Hash Type:              MD5
Lifetime:               00:30:00
Expires:                00:29:33
Challenge:              Not configured
Window size:            1
Last seq # rcvd:        14167808280012783626

From:                   192.168.106.2
To:                     192.168.106.1
Neighbor:               192.168.106.2
Interface:              ATM1/0.1
Mode:                   Receive
Key ID:                 1
Key ACL:                R3 (populated)
Key Source:             RSVPKey (enabled)
Key Type:               Dynamic per-neighbor chain
Handle:                 C0000412
Hash Type:              MD5
Lifetime:               00:30:00
Expires:                00:29:33
Challenge:              Not configured
Window size:            1
Last seq # rcvd:        14167808280012783619
```

Table 2 describes the significant fields shown in the display.

Table 2 show ip rsvp authentication detail Field Descriptions

Field	Description
From	Starting point of the security association.
To	Ending point of the security association.
Neighbor	IP address of the RSVP neighbor with which the security association is being maintained.
Interface	Name and number of the interface over which the security association is being maintained.
Mode	Separate associations maintained for sending and receiving RSVP messages for a specific RSVP neighbor. Possible values are **Send** or **Receive**.
Key ID	A string which, along with the IP address, uniquely identifies a security association. The key ID is automatically generated in Cisco IOS software by using the per-interface **ip rsvp authentication key** command, but it is configured in Cisco IOS software when using key chains for per-neighbor or per-interface RSVP keys. The key ID may be configurable on other RSVP platforms. A key ID is provided in every RSVP authenticated message initiated by a sender and is stored by every RSVP receiver. **Note** **Key Expired** in this field means that all possible keys used for this neighbor have expired.

Table 2 show ip rsvp authentication detail Field Descriptions (continued)

Field	Description
Key ACL	For key types that say dynamic and chain, this field indicates which ACL matched that neighbor, and therefore, which key chain to use. Possible values include: • **populated** = ACL has entries in it. • **removed** = ACL has been removed from the configuration.
Key Source	Indicates where the key was configured and whether it is enabled or disabled. For key chains, this indicates the name of the key chain; the Key ID field indicates which key in the chain is currently being used. For per-interface keys, this field contains the name of the interface that was configured with the key.
Key Type	Static (manually configured) or dynamic (created from a per-ACL key or obtained from a key management server such as Kerberos). **Note** Cisco IOS software does not currently support dynamic keys from key management servers.
Handle	Internal database ID assigned to the security association by RSVP for bookkeeping purposes.
Hash Type	Type of secure hash algorithm being used with that neighbor.
Lifetime	Maximum amount of time (in hours, minutes, and seconds) that can elapse before a security association is expired. **Note** This is not how long a key is valid; to obtain duration times for keys, use the **show key chain** command.
Expires	Amount of time remaining (in days, hours, minutes, and seconds) before the security association expires. **Note** This is not when the current key expires; to obtain expiration times for keys, use the **show key chain** command.
Challenge	For receive-type security associations, possible values are **Not Configured**, **Completed**, **In Progress**, and **Failed**. For send-type security associations, the value is **Supported**. Cisco IOS software can always respond to challenges; however, there may be non-Cisco neighbors that do not implement challenges.
Window size	Indicates the size of the window for receive-type security associations and the maximum number of authenticated RSVP messages that can be received out-of-order before a replay attack is to be suspected.
Last seq # sent	Displayed only for send-type security associations. It indicates the sequence number used to send the last authenticated message to the RSVP neighbor. Use this information to troubleshoot certain types of authentication problems.
Last valid seq # rcvd	Displayed only for receive-type security associations. It indicates the authentication sequence number of the last valid RSVP message received from the neighbor. By default, it shows only one sequence number. However, if you use the **ip rsvp authentication window-size** command to increase the authentication window size to *n*, then the last *n* valid received sequence numbers are displayed. Use this information to troubleshoot certain types of authentication problems.

Related Commands

Command	Description
clear ip rsvp authentication	Eliminates RSVP security associations before their lifetimes expire.

show ip rsvp counters

To display the number of Resource Reservation Protocol (RSVP) messages that were sent and received on each interface, use the **show ip rsvp counters** command in user EXEC or privileged EXEC mode.

show ip rsvp counters [**authentication**] [**interface** *type number* | **summary** | **neighbor**]

Syntax Description

authentication	(Optional) Displays a list of RSVP authentication counters.
interface *type number*	(Optional) Displays the number of RSVP messages sent and received for the specified interface name.
summary	(Optional) Displays the cumulative number of RSVP messages sent and received by the router over all interfaces.
neighbor	(Optional) Displays the number of RSVP messages sent and received by the specified neighbor.

Command Default

If you enter the **show ip rsvp counters** command without an optional keyword, the command displays the number of RSVP messages that were sent and received for each interface on which RSVP is configured.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
12.0(14)ST	This command was introduced.
12.2(13)T	The **neighbor** keyword was added, and the command was integrated into Cisco IOS Release 12.2(13)T.
12.2(15)T	The command output was modified to show the errors counter incrementing whenever an RSVP message is received on an interface with RSVP authentication enabled, but the authentication checks failed on that message.
12.2(11)S	This command was integrated into Cisco IOS Release 12.2(11)S.
12.0(22)S	This command was integrated into Cisco IOS Release 12.0(22)S.
12.0(29)S	The **authentication** keyword was added, and the command output was modified to include hello and message queues information.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Examples

The following example shows the values for the number of RSVP messages of each type that were sent and received by the router over all interfaces, including the hello and message queues information:

```
Router# show ip rsvp counters summary

All Interfaces          Recv      Xmit                        Recv      Xmit
    Path                 110        15    Resv                   50        28
    PathError              0         0    ResvError               0         0
    PathTear               0         0    ResvTear                0         0
    ResvConf               0         0    RTearConf               0         0
    Ack                    0         0    Srefresh                0         0
    Hello               5555      5554    IntegrityChalle         0         0
    IntegrityRespon        0         0    DSBM_WILLING            0         0
    I_AM_DSBM              0         0
    Unknown                0         0    Errors                  0         0

Recv Msg Queues                Current       Max
    RSVP                             0         2
    Hello (per-I/F)                  0         1
    Awaiting Authentication          0         0
```

Table 3 describes the significant fields shown in the display.

Table 3 show ip rsvp counters summary Field Descriptions

Field	Description
All Interfaces	Types of messages displayed for all interfaces. **Note** Hello is a summary of graceful restart, reroute (hello state timer), and Fast Reroute messages.
Recv	Number of messages received on the specified interface or on all interfaces.
Xmit	Number of messages transmitted from the specified interface or from all interfaces.
Recv Msg Queues	Queues for received messages for RSVP, hello per interface, and awaiting authentication. • Current—Number of messages queued. • Max—Maximum number of messages ever queued.

Related Commands

Command	Description
clear ip rsvp counters	Clears (sets to zero) all IP RSVP counters that are being maintained.

show ip rsvp interface

To display Resource Reservation Protocol (RSVP)-related information, use the **show ip rsvp interface** command in user EXEC or privileged EXEC mode.

show ip rsvp interface [*type number*] [**detail**]

Syntax Description

type	(Optional) Type of the interface.
number	(Optional) Number of the interface.
detail	(Optional) Displays additional information about interfaces.

Command Modes

User EXEC
Privileged EXEC

Command History

Release	Modification
11.2	This command was introduced.
12.2(2)T	The optional **detail** keyword was added.
12.2(4)T	This command was implemented on the Cisco 7500 series and the ATM-permanent virtual circuit (PVC) interface.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(13)T	The following modifications were made to this command: • Rate-limiting and refresh-reduction information were added to the output display. • This command was modified to display RSVP global settings when no keywords or arguments are entered.
12.2(15)T	The following modifications were made to this command: • The command output was modified to display the effects of compression on admission control and the RSVP bandwidth limit counter. • Cryptographic authentication parameters were added to the display.
12.2(18)SXF2	This command was integrated into Cisco IOS Release 12.2(18)SFX2.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use the **show ip rsvp interface** command to display information about interfaces on which RSVP is enabled, including the current allocation budget and maximum available bandwidth. Enter the optional **detail** keyword for additional information, including bandwidth and signaling parameters and blockade state.

Use the **show ip rsvp interface detail** command to display information about the RSVP parameters associated with an interface. These parameters include the following:

- Total RSVP bandwidth
- RSVP bandwidth allocated to existing flows
- Maximum RSVP bandwidth that can be allocated to a single flow
- The type of admission control supported (header compression methods)
- The compression methods supported by RSVP compression prediction

Examples

The following command shows information for each interface on which RSVP is enabled:

```
Router# show ip rsvp interface

interface    allocated  i/f max  flow max sub max
PO0/0        0          200M     200M     0
PO1/0        0          50M      50M      0
PO1/1        0          50M      50M      0
PO1/2        0          50M      50M      0
PO1/3        0          50M      50M      0
Lo0          0          200M     200M     0
```

Table 1 describes the fields shown in the display.

Table 4 show ip rsvp interface Field Descriptions

Field	Description
interface	Interface name.
allocated	Current allocation budget.
i/f max	Maximum allocatable bandwidth.
flow max	Largest single flow allocatable on this interface.
sub max	Largest sub-pool value allowed on this interface.

Detailed RSVP Information Example

The following command shows detailed RSVP information for each interface on which RSVP is enabled:

```
Router# show ip rsvp interface detail

PO0/0:
   Bandwidth:
     Curr allocated:0 bits/sec
     Max. allowed (total):200M bits/sec
     Max. allowed (per flow):200M bits/sec
     Max. allowed for LSP tunnels using sub-pools:0 bits/sec
     Set aside by policy (total):0 bits/sec
   Signalling:
     DSCP value used in RSVP msgs:0x3F
     Number of refresh intervals to enforce blockade state:4
```

```
     Number of missed refresh messages:4
     Refresh interval:30

 PO1/0:
   Bandwidth:
     Curr allocated:0 bits/sec
     Max. allowed (total):50M bits/sec
     Max. allowed (per flow):50M bits/sec
     Max. allowed for LSP tunnels using sub-pools:0 bits/sec
     Set aside by policy (total):0 bits/sec
Signalling:
     DSCP value used in RSVP msgs:0x3F
     Number of refresh intervals to enforce blockade state:4
     Number of missed refresh messages:4
     Refresh interval:30

PO1/1:
   Bandwidth:
     Curr allocated:0 bits/sec
     Max. allowed (total):50M bits/sec
     Max. allowed (per flow):50M bits/sec
     Max. allowed for LSP tunnels using sub-pools:0 bits/sec
     Set aside by policy (total):0 bits/sec
   Signalling:
     DSCP value used in RSVP msgs:0x3F
     Number of refresh intervals to enforce blockade state:4
     Number of missed refresh messages:4
     Refresh interval:30

 PO1/2:
   Bandwidth:
     Curr allocated:0 bits/sec
     Max. allowed (total):50M bits/sec
     Max. allowed (per flow):50M bits/secMax. allowed for LSP tunnels using sub-pools:0
bits/sec
     Set aside by policy (total):0 bits/sec
   Signalling:
     DSCP value used in RSVP msgs:0x3F
     Number of refresh intervals to enforce blockade state:4
     Number of missed refresh messages:4
     Refresh interval:30

 PO1/3:
   Bandwidth:
     Curr allocated:0 bits/sec
     Max. allowed (total):50M bits/sec
     Max. allowed (per flow):50M bits/sec
     Max. allowed for LSP tunnels using sub-pools:0 bits/sec
     Set aside by policy (total):0 bits/sec
   Signalling:
     DSCP value used in RSVP msgs:0x3F
     Number of refresh intervals to enforce blockade state:4
     Number of missed refresh messages:4
     Refresh interval:30

 Lo0:
   Bandwidth:
     Curr allocated:0 bits/sec
     Max. allowed (total):200M bits/sec
     Max. allowed (per flow):200M bits/sec
     Max. allowed for LSP tunnels using sub-pools:0 bits/sec
     Set aside by policy (total):0 bits/sec
   Signalling:
     DSCP value used in RSVP msgs:0x3F
```

```
    Number of refresh intervals to enforce blockade state:4
    Number of missed refresh messages:4
    Refresh interval:30
```

Table 2 describes the significant fields shown in the detailed display for interface PO0/0. The fields for the other interfaces are similar.

Table 5 show ip rsvp interface detail Field Descriptions—Detailed RSVP Information Example

Field	Description
PO0/0	Interface name.
Bandwidth	The RSVP bandwidth parameters in effect are the following: • Curr allocated = amount of bandwidth currently allocated in bits per second. • Max. allowed (total) = maximum amount of bandwidth allowed in bits per second. • Max. allowed (per flow) = maximum amount of bandwidth allowed per flow in bits per second. • Max. allowed for LSP tunnels using sub-pools = maximum amount of bandwidth allowed for label switched path (LSP) tunnels in bits per second. • Set aside by policy (total) = the amount of bandwidth set aside by the local policy in bits per second.
Signalling	The RSVP signalling parameters in effect are the following: • DSCP value used in RSVP msgs = differentiated services code point (DSCP) used in RSVP messages. • Number of refresh intervals to enforce blockade state = how long in milliseconds before the blockade takes effect. • Number of missed refresh messages = how many refresh messages until the router state expires. • Refresh interval = how long in milliseconds until a refresh message is sent.

RSVP Compression Method Prediction Example

The following example from the **show ip rsvp interface detail** command shows the RSVP compression method prediction configuration for each interface on which RSVP is configured:

```
Router# show ip rsvp interface detail

 Et2/1:
   Bandwidth:
     Curr allocated:0 bits/sec
     Max. allowed (total):1158K bits/sec
     Max. allowed (per flow):128K bits/sec
     Max. allowed for LSP tunnels using sub-pools:0 bits/sec
     Set aside by policy (total):0 bits/sec
   Admission Control:
     Header Compression methods supported:
       rtp (36 bytes-saved), udp (20 bytes-saved)
   Neighbors:
     Using IP encap:0.  Using UDP encap:0
```

```
  Signalling:
    Refresh reduction:disabled
  Authentication:disabled

Se3/0:
  Bandwidth:
    Curr allocated:0 bits/sec
    Max. allowed (total):1158K bits/sec
    Max. allowed (per flow):128K bits/sec
    Max. allowed for LSP tunnels using sub-pools:0 bits/sec
    Set aside by policy (total):0 bits/sec
  Admission Control:
    Header Compression methods supported:
      rtp (36 bytes-saved), udp (20 bytes-saved)
  Neighbors:
    Using IP encap:1.  Using UDP encap:0
  Signalling:
    Refresh reduction:disabled
  Authentication:disabled
```

Table 3 describes the significant fields shown in the display for Ethernet interface 2/1. The fields for serial interface 3/0 are similar.

Table 6 show ip rsvp interface detail Field Descriptions—RSVP Compression Method Prediction Example

Field	Description
Et2/1: Se3/0	Interface name.
Bandwidth	The RSVP bandwidth parameters in effect are the following: • Curr allocated = amount of bandwidth currently allocated in bits per second. • Max. allowed (total) = maximum amount of bandwidth allowed in bits per second. • Max. allowed (per flow) = maximum amount of bandwidth allowed per flow in bits per second. • Max. allowed for LSP tunnels using sub-pools = maximum amount of bandwidth allowed for LSP tunnels in bits per second. • Set aside by policy (total) = the amount of bandwidth set aside by the local policy in bits per second.
Admission Control	The type of admission control in effect are the following: • Header Compression methods supported—Real-Time Transport Protocol (RTP) or User Data Protocol (UDP) compression schemes and the number of bytes saved per packet.
Neighbors	The number of neighbors using IP and UDP encapsulation.
Signalling	The type of signaling in effect; Refresh reduction is either enabled (active) or disabled (inactive).
Authentication	Authentication is either enabled (active) or disabled (inactive).

Cryptographic Authentication Example

The following example of the **show ip rsvp interface detail** command displays detailed information, including the cryptographic authentication parameters, for all RSVP-configured interfaces on the router:

```
Router# show ip rsvp interface detail

 Et0/0:
   Bandwidth:
    Curr allocated: 0 bits/sec
    Max. allowed (total): 7500K bits/sec
    Max. allowed (per flow): 7500K bits/sec
    Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
    Set aside by policy (total):0 bits/sec
   Neighbors:
    Using IP encap: 0.  Using UDP encap: 0
   Signalling:
    Refresh reduction: disabled
   Authentication: enabled
    Key:           11223344
    Type:          sha-1
    Window size:   2
    Challenge:     enabled
```

Table 4 describes the significant fields shown in the display.

Table 7 show ip rsvp interface detail Field Descriptions—Cryptograhic Authentication Example

Field	Description
Et0/0	Interface name.
Bandwidth	The RSVP bandwidth parameters in effect are the following: • Curr allocated = amount of bandwidth currently allocated in bits per second. • Max. allowed (total) = maximum amount of bandwidth allowed in bits per second. • Max. allowed (per flow) = maximum amount of bandwidth allowed per flow in bits per second. • Max. allowed for LSP tunnels using sub-pools = maximum amount of bandwidth allowed for LSP tunnels in bits per second. • Set aside by policy (total) = the amount of bandwidth set aside by the local policy in bits per second.
Neighbors	The number of neighbors using IP and UDP encapsulation.
Signalling	The type of signaling in effect; Refresh reduction is either enabled (active) or disabled (inactive).

Table 7 show ip rsvp interface detail Field Descriptions—Cryptograhic Authentication Example (continued)

Field	Description
Authentication	Authentication is either enabled (active) or disabled (inactive). The parameters are the following: • Key = The key (string) for the RSVP authentication algorithm displayed in clear text (for example, 11223344) or encrypted <encrypted>. • Type = The algorithm to generate cryptographic signatures in RSVP messages; possible values are md5 and sha-1. • Window size = Maximum number of RSVP authenticated messages that can be received out of order. • Challenge = The challenge-response handshake performed with any new RSVP neighbors that are discovered on a network; possible values are **enabled** (active) or **disabled** (inactive).

Related Commands

Command	Description
show ip rsvp installed	Displays RSVP-related installed filters and corresponding bandwidth information.
show ip rsvp neighbor	Displays current RSVP neighbors.

Glossary

bandwidth—The difference between the highest and lowest frequencies available for network signals. The term also is used to describe the rated throughput capacity of a given network medium or protocol.

DMZ—demilitarized zone. The neutral zone between public and corporate networks.

flow—A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

key—A data string that is combined with source data according to an algorithm to produce output that is unreadable until decrypted.

QoS—quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

router—A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

RSVP—Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

security association—A block of memory used to hold all the information RSVP needs to authenticate RSVP signaling messages from a specific RSVP neighbor.

spoofing—The act of a packet illegally claiming to be from an address from which it was not actually sent. Spoofing is designed to foil network security mechanisms, such as filters and access lists.

TE—traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

trusted neighbor—A router with authorized access to information.