OSPFv2 Cryptographic Authentication

To prevent unauthorized or invalid routing updates in your network, Open Shortest Path First version 2 (OSPFv2) protocol packets must be authenticated.

There are two methods of authentication that are defined for OSPFv2: plain text authentication and cryptographic authentication. This module describes how to configure cryptographic authentication using the Hashed Message Authentication Code - Secure Hash Algorithm (HMAC-SHA). OSPFv2 specification (RFC 2328) allows only the Message-Digest 5 (MD5) algorithm for cryptographic authentication. However, RFC 5709 (OSPFv2 HMAC-SHA Cryptographic Authentication) allows OSPFv2 to use HMAC-SHA algorithms for cryptographic authentication.

- Finding Feature Information, on page 1
- Prerequisites for OSPFv2 Cryptographic Authentication, on page 1
- Information About OSPFv2 Cryptographic Authentication, on page 2
- How to Configure OSPFv2 Cryptographic Authentication, on page 2
- Configuration Examples for OSPFv2 Cryptographic Authentication, on page 5
- Additional References for OSPFv2 Cryptographic Authentication, on page 7
- Feature Information for OSPFv2 Cryptographic Authentication, on page 8

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

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

Prerequisites for OSPFv2 Cryptographic Authentication

Ensure that Open Shortest Path First version 2 (OSPFv2) is configured on your network.

Information About OSPFv2 Cryptographic Authentication

Configuring OSPFv2 Cryptographic Authentication

The OSPFv2 Cryptographic Authentication feature allows you to configure a key chain on the OSPF interface to authenticate OSPFv2 packets by using HMAC-SHA algorithms. You can use an existing key chain that is being used by another protocol, or you can create a key chain specifically for OSPFv2.

A key chain is a list of keys. Each key consists of a key string, which is also called the password or passcode. A key-string is essential for a key to be operational. Each key is identified by a unique key ID. To authenticate the OSPFv2 packets, it is essential that the cryptographic authentication algorithm be configured with a key. OSPFv2 supports keys with key IDs ranging from 1 to 255. The combination of the cryptographic authentication algorithm and the key is known as a Security Association (SA).

The authentication key on a key chain is valid for a specific time period called lifetime. An SA has the following configurable lifetimes:

- Accept lifetime
- Send lifetime

While adding a new key, the Send lifetime is set to a time in the future so that the same key can be configured on all devices in the network before the new key becomes operational. Old keys are removed only after the new key is operational on all devices in the network. When packets are received, the key ID is used to fetch the data for that key. The packet is verified using the cryptographic authentication algorithm and the configured key ID. If the key ID is not found, the packet is dropped.

Note When key chain has more than one key, OSPF selects the key that has the maximum life time. Key having an infinite lifetime is preferred. If keys have the same lifetime, then key with the higher key ID is preferred.

Use the **ip ospf authentication key-chain** command to configure key chains for OSPFv2 cryptographic authentication.

Note If OSPFv2 is configured to use a key chain, all MD5 keys that were previously configured using the **ip ospf message-digest-key** command are ignored.

How to Configure OSPFv2 Cryptographic Authentication

Defining a Key Chain

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **key chain** *name*
4. **key** *key-id*
5. **key-string** *name*
6. **cryptographic-algorithm** *name*
7. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name* **Example:** `Device(config)# key chain sample1`	Specifies the key chain name and enters key-chain configuration mode.
Step 4	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Specifies the key identifier and enters key-chain key configuration mode. The range is from 1 to 255.
Step 5	**key-string** *name* **Example:** `Device(config-keychain-key)# key-string string1`	Specifies the key string.
Step 6	**cryptographic-algorithm** *name* **Example:** `Device(config-keychain-key)# cryptographic-algorithm hmac-sha-256`	Configures the key with the specified cryptographic algorithm.
Step 7	**send-lifetime** *start-time* {**infinite**	*end-time*	**duration** *seconds*} **Example:** `Device(config-keychain-key)# send-lifetime local 10:00:00 5 July 2013 infinite`	Sets the time period during which an authentication key on a key chain is valid to be sent during key exchange with another device.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-keychain-key)# end`	Exits key-chain key configuration mode and returns to privileged EXEC mode.

Defining Authentication on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip ospf authentication key-chain** *name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet0/0/0`	Specifies an interface type and number and enters interface configuration mode.
Step 4	**ip ospf authentication key-chain** *name* **Example:** `Device(config-if)# ip ospf authentication key-chain ospf1`	Specifies the key chain for an interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for OSPFv2 Cryptographic Authentication

Example: Defining a Key Chain

The following example shows how to configure a key chain:

```
Device> enable
Device# configure terminal
Device(config)# key chain sample1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string ThisIsASampleKey12345
Device(config-keychain-key)# cryptographic-algorithm hmac-sha-256
Device(config-keychain-key)# send-lifetime local 10:00:00 5 July 2013 infinite
Device(config-keychain-key)# end
```

Example: Verifying a Key Chain

The following sample output from the **show key chain** command displays the key chain information:

```
Device# show key chain Key-chain sample1

     key 1 -- text "ThisIsASampleKey12345"
         accept lifetime (always valid) - (always valid) [valid now]
         send lifetime (10:00:00 PDT Jul 5 2013) - (infinite)
```

The table below describes the significant fields in the output:

Table 1: show ip ospf interface Field Descriptions

Field	Description
key	Status of the configured key.
accept lifetime	The time interval within which the device accepts the key during key exchange with another device.
send lifetime	The time interval within which the device sends the key during a key exchange with another device.

Example: Defining Authentication on an Interface

The following example shows how to define authentication on Gigabit Ethernet interface 0/0/0:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet0/0/0
Device (config-if)# ip ospf authentication key-chain sample1
Device (config-if)# end
```

Example: Verifying Authentication on an Interface

The following sample output of the **show ip ospf interface** command displays the cryptographic key information:

```
Device# show ip ospf interface GigabitEthernet0/0/0

GigabitEthernet0/0/0 is up, line protocol is up
  Internet Address 192.168.8.2/24, Area 1, Attached via Interface Enable
  Process ID 1, Router ID 10.1.1.8, Network Type BROADCAST, Cost: 10
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           10        no          no            Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.1.8, Interface address 192.168.8.2
  Backup Designated router (ID) 10.1.1.9, Interface address 192.168.8.9
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Can be protected by per-prefix Loop-Free FastReroute
  Can be used for per-prefix Loop-Free FastReroute repair paths
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.1.9  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
  Cryptographic authentication enabled
    Sending SA: Key 25, Algorithm HMAC-SHA-256 - key chain sample1
```

The table below describes the significant fields in the output:

Table 2: show ip ospf interface Field Descriptions

Field	Description
GigabitEthernet	Status of the physical link and operational status of the protocol.
Internet Address	Interface IP address, subnet mask, and area address.
Area	OSPF area.
Process ID	OSPF process ID.
Cost	Administrative cost assigned to the interface.
Topology-MTID	MTR topology Multitopology Identifier (MTID) is a number assigned so that the protocol can identify the topology associated with information that it sends to its peers.
Transmit Delay	Transmit delay (in seconds), interface state, and router priority.
State	Operational state of the interface.

Field	Description
Designated Router	Designated router ID and respective interface IP address.
Backup Designated router	Backup designated router ID and respective interface IP address.
Timer intervals configured	Configuration of timer intervals.
Neighbor Count	Count of network neighbors and list of adjacent neighbors.
Cryptographic authentication	Status of cryptographic authentication.
Sending SA	Status of the sending SA (Security Association). Key, cryptographic algorithm, and key chain used.

Additional References for OSPFv2 Cryptographic Authentication

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
OSPF commands	Cisco IOS IP Routing: OSPF Command Reference

Standards and RFCs

Standard	Title
RFC 2328	OSPF Version 2, April 1998
RFC 5709	OSPFv2 HMAC-SHA Cryptographic Authentication, October 2009

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for OSPFv2 Cryptographic Authentication

Table 3: Feature Information for OSPFv2 Cryptographic Authentication

Feature Name	Releases	Feature Information
OSPFv2 Cryptographic Authentication	15.4(1)T	The OSPFv2 Cryptographic Authentication feature prevents unauthorized or invalid routing updates in your network by authenticating Open Shortest Path First version 2 (OSPFv2) protocol packets using HMAC-SHA algorithms. The following command was modified: **ip ospf authentication**.