OSPFv3 Authentication Trailer

Feature History for OSPFv3 Authentication Trailer

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	OSPFv3 Authentication Trailer: The OSPFv3 Authentication Trailer is a security feature that provides an alternative mechanism, to IPsec, to authenticate OSPFv3 protocol packets.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

OSPFv3 Authentication Trailer

The OSPFv3 Authentication Trailer is a security feature that provides an alternative mechanism, to IPsec, to authenticate Open Shortest Path First version 3 (OSPFv3) protocol packets. Prior to the OSPFv3 authentication trailer, OSPFv3 IPsec was the only mechanism for authenticating protocol packets. The OSPFv3 authentication trailer is also platform independent.

Key aspects of OSPFv3 Authentication Trailer

The OSPFv3 authentication trailer feature, also referred to as non-IPsec cryptographic authentication feature, includes the following key aspects:

- a special data block, the authentication trailer, is attached to the end of the OSPFv3 packet,

- the length of the authentication trailer is included in the IPv6 payload length, but not the in the length of the OSPFv3 packet,
- for OSPFv3 hello packets and database description packets, in the OSPFv3 Options field, the
  - the Link-Local Signaling (LLS) block is established by setting the L-bit. If present, the LLS data block is included in the cryptographic authentication computation along with the OSPFv3 packet.
  - the authentication trailer bit is set. The authentication trailer bit indicates that the authentication trailer is present.
- The authentication trailer bit setting from the OSPFv3 hello and database description packets is preserved in the OSPFv3 neighbor data structure. For other OSPFv3 packet types (that do not include the OSPFv3 options field), the authentication trailer is determined by using the neighbor data structure.

OSPFv3 authentication trailer components

To authenticate outgoing and incoming OSPFv3 packets, OSPFv3 authentication trailer uses

- cryptographic keys: OSPFv3 authentication trailer uses Cisco IOS key chains to create and manage the keys.
- message digest: A fixed-size cryptographic hash value generated from the contents of a message or data packet. It acts as a unique fingerprint of the data, ensuring data integrity and authenticity.
- Authentication algorithm: A cryptographic method used to generate and verify the message digest attached to OSPFv3 packets.
- Security Association ID: An identifier that links the authentication algorithm and the secret key needed to generate and verify the message digest for authenticating OSPFv3 packets.

How OSPFv3 packet authentication works with OSPFv3 Authentication Trailer

Workflow

With OSPFv3 authentication trailer

1. the Security Association ID links the authentication algorithm with the secret key to calculate the message digest.
2. This digest is attached to the OSPFv3 packet to verify that the packet has not been altered in transit and that it comes from a trusted source.
3. When a device receives the packet, it uses the same Security Association ID (to link the same algorithm with the same secret key) to recalculate the message digest and compares it to the received digest.
4. If they match, the packet is considered authentic and unmodified; otherwise, it is discarded to maintain secure communication.

Key selection for outgoing and incoming packet

When sending OSPFv3 packets (outgoing),

- the device selects the key from the key chain based on these rules:
  - Choose the key that will expire last (i.e., the one with the latest stop time).
  - If two keys share the same expiration time, select the key with the highest key ID.
- If the authentication is configured but the last valid key has expired, packets are still sent using the key, and a syslog message is also generated.
- If no valid key is available, the packet is sent without the authentication trailer.

When a packet is received (incoming), its key ID is used to find the corresponding key in the key chain.

- If the key ID is not found or the security association is invalid, the packet is dropped.
- If the corresponding key is found, the packet is verified using the configured algorithm and key for that key ID.

Key chain rollover

Key chains support rollover using key lifetimes. You can add a new key to a key chain with a future send start time. This setting allows the new key to be configured on all the devices before the keys are used.

Sequence number

The OSPFv3 authentication trailer feature also provides packet replay protection through sequence number.

OSPFv3 hello packets have higher priority than other OSPFv3 packets, so they can get reordered on the outgoing interface. This reordering can create problems with sequence number verification on neighboring devices. To prevent sequence mismatch, OSPFv3 verifies the sequence number separately for each packet type.

See RFC 7166 for more details on the authentication procedure.

Deployment of authentication trailer on the network

During the initial rollout of the OSPFv3 authentication trailer feature, the deployment mode allows adjacency to be maintained between devices that have the authentication trailer configured and those that do not yet have it configured. In deployment mode, devices process packets differently to achieve this compatability.

- Outgoing packets: The OSPF checksum is calculated even if the authentication trailer is configured. This ensures compatibility with devices not yet using the authentication trailer.
- Incoming packets: Packets that either lack the authentication trailer or have an incorrect authentication hash are dropped to maintain security.
- The command **show ospfv3 neighbor detail** in deployment mode displays the authentication status of the last received packet, which helps verify whether the authentication trailer feature is functioning correctly before switching to normal mode.

Once verified, the mode can be changed to normal using the **authentication mode normal** command, where the authentication trailer is fully enforced for all packets.

Configure the OSPFv3 Authentication Trailer

To configure OSPFv3 authentication trailer, perform this procedure:

Before you begin

An authentication key is required for configuring OSPFv3 authentication trailer.

Procedure

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 ethernet 1/0/1
```

Configures an interface.

Step 4 **ospfv3** [*process-id*] [**ipv4** | **ipv6**] **authentication** {**key-chain** *chain-name* | **null**}

Example:

```
Device(config-if)# ospfv3 1 ipv6 authentication key-chain ospf-1
```

Specifies the authentication type for an OSPFv3 instance.

- *process-id*: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

  Process ID can be a positive integer from 1 to 65535.

- **keychain** : This option enables keychain authentication. Keychain authentication is a more robust and flexible method for managing authentication keys. It allows you to define multiple keys within a named keychain, along with their lifetimes, enabling automatic time-based key rotation. This enhances security by regularly changing the cryptographic keys used for authentication.

- *chain-name*: Refers to a pre-configured keychain that contains the actual authentication keys.

- **null**: this option configures "null authentication," meaning that no authentication is applied to OSPFv3 packets. While it effectively disables authentication, it is a configurable state. This might be used in specific scenarios where authentication is not required on a particular interface or area, even if other parts of the OSPFv3 domain use authentication.

Step 5 **router ospfv3** [*process-id*]

Example:

```
Device(config-if)# router ospfv3 1
```

Enters OSPFv3 router configuration mode.

process-id: The process ID is an internally used, identification parameter that is locally assigned. Each OSPF has a unique process ID.

Process ID can be a positive integer from 1 to 65535.

Step 6 **address-family ipv6 unicast**

Example:

```
Device(config-router)# address-family ipv6 unicast
```

Configures the IPv6 address family in the OSPFv3 process and enters IPv6 address family configuration mode.

unicast: Specifies the configuration is for IPv6 unicast routing.

Step 7 **area** *area-id* **authentication** {**key-chain** *chain-name* | **null**}

Example:

```
Device(config-router-af)# area 1 authentication key-chain ospf-chain-1
```

Configures the authentication trailer on all interfaces in the OSPFv3 area.

- *area-id*: Identifier of the area about which routes are to be summarized. It can be specified as either a decimal value or as an IPv6 prefix.
- **keychain** : This option enables keychain authentication. Keychain authentication is a more robust and flexible method for managing authentication keys. It allows you to define multiple keys within a named keychain, along with their lifetimes, enabling automatic time-based key rotation. This enhances security by regularly changing the cryptographic keys used for authentication.
- *chain-name*: Refers to a pre-configured keychain that contains the actual authentication keys.
- **null**: this option configures "null authentication," meaning that no authentication is applied to OSPFv3 packets. While it effectively disables authentication, it is a configurable state. This might be used in specific scenarios where authentication is not required on a particular interface or area, even if other parts of the OSPFv3 domain use authentication.

Step 8 **area** *area-id* **virtual-link** *router-id* **authentication key-chain** *chain-name*

Example:

```
Device(config-router-af)# area 1 virtual-link 1.1.1.1 authentication key-chain ospf-chain-1
```

Configures the authentication for virtual links.

- *area-id* **virtual-link** *router-id*: Specifies the OSPF area ID and the router ID of the router at the other end of the virtual link.
- **keychain** : This option enables keychain authentication. Keychain authentication is a more robust and flexible method for managing authentication keys. It allows you to define multiple keys within a named keychain, along with their lifetimes, enabling automatic time-based key rotation. This enhances security by regularly changing the cryptographic keys used for authentication.

- *chain-name*: Refers to a pre-configured keychain that contains the actual authentication keys.

Step 9 **area** *area-id* **sham-link** *source-address destination-address* **authentication key-chain** *chain-name*

Example:

```
Device(config-router-af)# area 1 sham-link 1.1.1.1 1.1.1.0 authentication key-chain
ospf-chain-1
```

Configures the authentication for sham-links.

- *area-id* **sham-link**: This specifies the OSPF area ID to which the sham link belongs. The sham link must be configured within an existing OSPF area, and typically, this is the same area that the connected Customer Edge (CE) routers belong to.
- *source-address*: This is the IP address of the local PE router's endpoint for the sham link. This is usually a loopback interface address that is part of the VPN routing and forwarding (VRF) instance and is advertised via BGP, not OSPF.
- *destination-address*: This is the IP address of the remote PE router's endpoint for the sham link. This address must also be a loopback interface address on the remote PE, part of the same VRF, and advertised via BGP.
- **keychain** : This option enables keychain authentication. Keychain authentication is a more robust and flexible method for managing authentication keys. It allows you to define multiple keys within a named keychain, along with their lifetimes, enabling automatic time-based key rotation. This enhances security by regularly changing the cryptographic keys used for authentication.
- *chain-name*: Refers to a pre-configured keychain that contains the actual authentication keys.

Step 10 **authentication mode** {**deployment** | **normal**}

Example:

```
Device(config-router-af)# authentication mode deployment
```

(Optional) Specifies the type of authentication used for the OSPFv3 instance.

- **deployment**: This mode is typically used during the initial setup, testing, or troubleshooting of an authentication system.
- **normal**: This is the standard operational mode once the authentication system has been thoroughly tested and validated.

Step 11 **end**

Example:

```
Device(config-router-af)# end
```

Exits IPv6 address family configuration mode and returns to privileged EXEC mode.

Step 12 **show ospfv3 interface**

Example:

```
Device# show ospfv3
```

(Optional) Displays OSPFv3-related interface information.

Step 13 **show ospfv3 neighbor** [*detail*]

Example:

```
Device# show ospfv3 neighbor detail
```

(Optional) Displays OSPFv3 neighbor information on a per-interface basis.

detail: This is an optional keyword that, when added, provides more extensive and detailed information about each OSPFv3 neighbor.

Step 14 **debug ospfv3**

Example:

```
Device# debug ospfv3
```

(Optional) Displays debugging information for OSPFv3.

Configuration examples for the OSPFv3 Authentication Trailer

The following sections provide examples on how to configure the OSPFv3 authentication trailer and how to verify the OSPFv3 authentication trailer configuration.

Example: Configure the OSPFv3 Authentication Trailer

The following example shows how to define authentication trailer on GigabitEthernet interface 1/0/1:

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# ospfv3 1 ipv6 authentication key-chain ospf-1
Device(config-if)# router ospfv3 1
Device(config-router)# address-family ipv6 unicast
Device(config-router-af)# area 1 authentication key-chain ospf-1
Device(config-router-af)# area 1 virtual-link 1.1.1.1 authentication key-chain ospf-1
Device(config-router-af)# area 1 sham-link 1.1.1.1 authentication key-chain ospf-1
Device(config-router-af)# authentication mode deployment
Device(config-router-af)# end
Device(config)# key chain ospf-1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string ospf
Device(config-keychain-key)# cryptographic-algorithm hmac-sha-256
```

Example: Verify OSPFv3 Authentication Trailer

The following example shows the output of the **show ospfv3** command.

```
Device# show ospfv3
 OSPFv3 1 address-family ipv6
 Router ID 1.1.1.1
…
RFC1583 compatibility enabled
 Authentication configured with deployment key lifetime
 Active Key-chains:
  Key chain ospf-1: Send key 1, Algorithm HMAC-SHA-256, Number of interfaces 1
    Area BACKBONE(0)
```