

RFC 430x IPsec Support

The RFC 430x IPsec Support includes features—RFC 430x IPsec Support Phase 1 and RFC430x IPsec Support Phase 2—that implement Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.

- Information About RFC 430x IPsec Support, on page 1
- How to Configure RFC 430x IPsec Support, on page 2
- Configuration Examples for RFC 430x IPsec Support, on page 5
- Additional References for RFC 430x IPsec Support, on page 7
- Feature Information for RFC 430x IPsec Support, on page 7

Information About RFC 430x IPsec Support

RFC 430x IPsec Support Phase 1

The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.

RFC 4301 specifies the base architecture for IPsec-compliant systems. RFC 4301 describes how to provide a set of security services for traffic at the IP layer, in both the IPv4 and IPv6 environments. The RFC 430x IPsec Support Phase 1 feature provides support for the following RFC 4301 implementations on Cisco IOS software.

- **Security association (SA) lifetime**—The lifetime of a security association between IPsec and Internet Key Exchange (IKE) or Internet Key Exchange Version 2 (IKEv2) must not exceed the lifetime of the authentication certificate.
- **OPAQUE selectors**—OPAQUE indicates that the corresponding selector field is not available for verification. When IKEv2 encounters an OPAQUE selector, IKEv2 skips, does not process the OPAQUE selector, and moves to next selector for policy verification.
- **Explicit Congestion Notification (ECN) support**—ECN is propagated when decrypting an IPsec packet thereby ensuring the packet source and destination are aware of congestion that occurs within the network.
- **Fragment processing**—Peers must not send Initial and noninitial fragments in the same tunnel. There must be a separate tunnel mode SA for carrying initial and noninitial fragments and separate tunnel mode SA for noninitial fragments. IPsec peers must support discarding of packets and stateful fragment checking to accommodate bypass traffic.
- **Do not fragment-(DF) bit processing**—DF-bit processing must be set on a per SA basis.

- **Dummy packet generation support**—It should be possible to send dummy packets via IPsec SA to encapsulate the packets when traffic is flowing via IPsec SA tunnel.

RFC 430x IPsec Support Phase 2

The RFC 430x IPsec Support Phase 2 feature provides support for the RFC 4301 implementation of encryption and decryption of Internet Control Message Protocol (ICMP) packets on Cisco IOS software.

ICMP error messages are sent when an ICMP error occurs. For example, when a host is not reachable, the intermediate device sends a message to the originator of the ICMP request that the host is not reachable. When an ICMP error message reaches an IPsec encryption policy, it may not be classified to match an existing SA. So, the packets are classified based on the data inside the ICMP error message. This data contains the source and destination address of the original ICMP message. If an SA is found based on the address in the ICMP error message, the SA is used. If there is no SA, an SA is created if the policy permits. For decryption, the post decrypt check is performed on the data inside the ICMP error message if a valid SA is not found.

The encryption and decryption of ICMP error messages can be verified through the encrypt and decrypt counters displayed in the output of the **show crypto ipsec sa** command.

How to Configure RFC 430x IPsec Support

Configuring RFC 430x IPsec Support Globally

Perform this task to configure the RFC 4301 implementations globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*}
4. **crypto ipsec security-association ecn** {**discard** | **propogate**}
5. **exit**

DETAILED STEPS

Procedure

	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.

	Command or Action	Purpose
Step 3	**crypto ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*} **Example:** `Device(config)# crypto ipsec security-association dummy seconds 5`	Enables the generation and transmission of dummy packets in an IPsec traffic flow.
Step 4	**crypto ipsec security-association ecn** {**discard** | **propogate**} **Example:** `Device(config)# crypto ipsec security-association ecn discard`	Enables the Explicit Congestion Notification (ECN) settings in an IPsec traffic flow.
Step 5	**exit** **Example:** `Device(config-crypto-map)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring RFC 430x IPsec Support Per Crypto Map

Perform this task to configure the RFC 4301 implementations per crypto map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **ipsec-isakmp**
4. **set ipsec security-association dfbit** {**clear** | **copy** | **set**}
5. **set ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*}
6. **set ipsec security-association ecn** {**discard** | **propogate**}
7. **end**
8. **show crypto map ipsec sa**

DETAILED STEPS

Procedure

	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.

	Command or Action	Purpose
Step 3	**crypto map** *map-name seq-num* **ipsec-isakmp** **Example:** `Device(config)# crypto map cmap 1 ipsec-isakmp`	Specifies the crypto map entry to be created or modified and enters crypto map configuration mode.
Step 4	**set ipsec security-association dfbit** {**clear** | **copy** | **set**} **Example:** `Device(config-crypto-map)# set ipsec security-association dfbit set`	Enables do not fragment (DF)-bit processing per security association (SA) for an IPsec traffic flow in a crypto map.
Step 5	**set ipsec security-association dummy** {**pps** *rate* | **seconds** *seconds*} **Example:** `Device(config-crypto-map)# set ipsec security-association dummy seconds 5`	Enables the generation and transmission of dummy packets for an IPsec traffic flow in a crypto map.
Step 6	**set ipsec security-association ecn** {**discard** | **propogate**} **Example:** `Device(config-crypto-map)# set ipsec security-association ecn propogate`	Enables the Explicit Congestion Notification (ECN) settings per SA for an IPsec traffic flow in a crypto map.
Step 7	**end** **Example:** `Device(config-crypto-map)# end`	Exits crypto map configuration mode and returns to privileged EXEC mode.
Step 8	**show crypto map ipsec sa** **Example:** `Device# show crypto map ipsec sa`	Displays the settings used by IPsec SAs.

Example

The following is sample output from the **show crypto map ipsec sa** command:

```
Device# show crypto map ipsec sa

interface: Tunnel0
 Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::32F7:DFF:FE54:7FD1
protected vrf: (none)
local ident (addr/mask/prot/port): (3FFE:2002::32F7:DFF:FE54:7FD1/128/47/0)
remote ident (addr/mask/prot/port): (3FFE:2002::C671:FEFF:FE88:EB82/128/47/0)
current_peer 3FFE:2002::C671:FEFF:FE88:EB82 port 500
 PERMIT, flags={origin_is_acl,}
#pkts encaps: 36, #pkts encrypt: 36, #pkts digest: 36
#pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
#send dummy packets 852600, #recv dummy packets 424905

local crypto endpt.: 3FFE:2002::32F7:DFF:FE54:7FD1,
```

```
remote crypto endpt.: 3FFE:2002::C671:FEFF:FE88:EB82
plaintext mtu 1430, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb GigabitEthernet0/0/1
current outbound spi: 0xE963D1EC(3915633132)
PFS (Y/N): N, DH group: none
Dummy packet: Initializing

inbound esp sas:
spi: 0xF4E01B9A(4108327834)
 transform: esp-3des esp-md5-hmac,
 in use settings ={Tunnel, }
 conn id: 2053, flow_id: ESG:53, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

 sa timing: remaining key lifetime (k/sec): (4608000/2343)
 IV size: 8 bytes
 replay detection support: Y
 Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0xE963D1EC(3915633132)
 transform: esp-3des esp-md5-hmac,
 in use settings ={Tunnel, }
 conn id: 2054, flow_id: ESG:54, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

 sa timing: remaining key lifetime (k/sec): (4608000/2343)
 IV size: 8 bytes
 replay detection support: Y
 Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:
```

Configuration Examples for RFC 430x IPsec Support

Example: Configuring RFC 430x IPsec Support Globally

The following examples shows how to configure RFC 430x IPsec Support globally:

```
Device> enable
Device# configure terminal
Device(config)# crypto ipsec security-association dummy seconds 15
Device(config)# crypto ipsec security-association ecn propogate
Device(config-crypto-map)# exit
```

Example: Configuring RFC 430x IPsec Support Per Crypto Map

The following examples shows how to configure RFC 430x IPsec Support per crypto map:

```
Device> enable
Device# configure terminal
Device(config)# crypto map cmap 1 ipsec-isakmp
```

```
Device(config-crypto-map)# set security-association copy
Device(config-crypto-map)# set security-association dummy seconds 15
Device(config-crypto-map)# set security-association ecn propogate
Device(config-crypto-map)# end
Device# show crypto map ipsec sa

interface: Tunnel0
 Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::32F7:DFF:FE54:7FD1
protected vrf: (none)
local ident (addr/mask/prot/port): (3FFE:2002::32F7:DFF:FE54:7FD1/128/47/0)
remote ident (addr/mask/prot/port): (3FFE:2002::C671:FEFF:FE88:EB82/128/47/0)
current_peer 3FFE:2002::C671:FEFF:FE88:EB82 port 500
 PERMIT, flags={origin_is_acl,}
#pkts encaps: 36, #pkts encrypt: 36, #pkts digest: 36
#pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
#send dummy packets 852600, #recv dummy packets 424905

local crypto endpt.: 3FFE:2002::32F7:DFF:FE54:7FD1,
remote crypto endpt.: 3FFE:2002::C671:FEFF:FE88:EB82
plaintext mtu 1430, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb GigabitEthernet0/0/1
current outbound spi: 0xE963D1EC(3915633132)
PFS (Y/N): N, DH group: none
Dummy packet: Initializing

inbound esp sas:
spi: 0xF4E01B9A(4108327834)
 transform: esp-3des esp-md5-hmac,
 in use settings ={Tunnel, }
 conn id: 2053, flow_id: ESG:53, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

 sa timing: remaining key lifetime (k/sec): (4608000/2343)
 IV size: 8 bytes
 replay detection support: Y
 Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0xE963D1EC(3915633132)
 transform: esp-3des esp-md5-hmac,
 in use settings ={Tunnel, }
 conn id: 2054, flow_id: ESG:54, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

 sa timing: remaining key lifetime (k/sec): (4608000/2343)
 IV size: 8 bytes
 replay detection support: Y
 Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:
```

Additional References for RFC 430x IPsec Support

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IKEv2 configuration	
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFC 4301	*Security Architecture for the Internet Protocol*

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 RFC 430x IPsec Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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.

Table 1: Feature Information for RFC430x IPsec Support

Feature Name	Releases	Feature Information
RFC430x IPsec Support Phase 1		The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301. The following commands were introduced or modified: **crypto ipsec security-association dummy, crypto ipsec security-association ecn, set ipsec security-association dfbit, set ipsec security-association dummy, set ipsec security-association ecn, show crypto map ipsec sa.**
RFC430x IPsec Support Phase 2		The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301. No commands were modified or updated for this feature.