# Internet Protocol Security



# Information about Internet Protocol Security

Internet Protocol Security (IPsec) is a framework of open standards for ensuring secure private communications over the Internet. Based on standards developed by the Internet Engineering Task Force (IETF), IPsec ensures confidentiality, integrity, and authenticity of data communications across a public network. IPsec provides a necessary component of a standards-based, flexible solution for deploying a network-wide security policy.

Cisco Catalyst 9800 Series Wireless Controller supports IPsec configuration. The support for IPSec secures syslog traffic.

This section provides information about how to configure IPsec between Cisco Catalyst 9800 Series Wireless Controller and syslog (peer IP).

IPsec provides the following network security services:

- Data confidentiality: The IPsec sender can encrypt packets before transmitting them across a network.
- Data integrity: The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.
- Data origin authentication: The IPsec receiver can authenticate the source of the sent IPsec packets. This service is dependent upon the data integrity service.
- Anti-replay: The IPsec receiver can detect and reject replayed packets.

IPsec provides secure *tunnels* between two peers, such as two devices. The administrator defines which packets are considered sensitive and should be sent through these secure tunnels and specifies the parameters that should be used to protect these sensitive packets by specifying the characteristics of these tunnels. When the IPsec peer recognizes a sensitive packet, the peer sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

More accurately, these *tunnels* are sets of security associations (SAs) that are established between two IPsec peers. The SAs define the protocols and algorithms to be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol.

With IPsec, administrators can define the traffic that needs to be protected between two IPsec peers by configuring access lists and applying these access lists to interfaces using crypto map sets. Therefore, traffic may be selected on the basis of the source and destination address, and optionally the Layer 4 protocol and port. (The access lists used for IPsec are only used to determine the traffic that needs to be protected by IPsec, not the traffic that should be blocked or permitted through the interface. Separate access lists define blocking and permitting at the interface.)

A crypto map set can contain multiple entries, each with a different access list. The crypto map entries are searched in a sequence--the device attempts to match the packet to the access list specified in that entry.

When a packet matches a permit entry in a particular access list, and the corresponding crypto map entry is tagged as cisco, connections are established, if necessary. If the crypto map entry is tagged as ipsec-isakmp, IPsec is triggered. If there is no SA that the IPsec can use to protect this traffic to the peer, IPsec uses IKE to negotiate with the remote peer to set up the necessary IPsec SAs on behalf of the data flow. The negotiation uses information specified in the crypto map entry as well as the data flow information from the specific access list entry.

Once established, the set of SAs (outbound to the peer) is then applied to the triggering packet and to subsequent applicable packets as those packets exit the device. *Applicable* packets are packets that match the same access list criteria that the original packet matched. For example, all applicable packets could be encrypted before being forwarded to the remote peer. The corresponding inbound SAs are used when processing the incoming traffic from that peer.

Access lists associated with IPsec crypto map entries also represent the traffic that the device needs protected by IPsec. Inbound traffic is processed against crypto map entries--if an unprotected packet matches a permit entry in a particular access list associated with an IPsec crypto map entry, that packet is dropped because it was not sent as an IPsec-protected packet.

Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms, and other settings that can be applied to IPsec-protected traffic. During the IPsec SA negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

# Internet Key Exchange Version 1 Transform Sets

An Internet Key Exchange version 1 (IKEv1) transform set represents a certain combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

Privileged administrators can specify multiple transform sets and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry is used in the IPsec SA negotiation to protect the data flows specified by that crypto map entry's access list.

During IPsec security association negotiations with IKE, peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and applied to the protected traffic as part of both peers' IPsec SAs.

**Note** If a transform set definition is changed during operation that the change is not applied to existing security associations, but is used in subsequent negotiations to establish new SAs. If you want the new settings to take effect sooner, you can clear all or part of the SA database by using the **clear crypto sa** command.

The following snippet helps to configure IPsec IKEv1 to use AES-CBC-128 for payload encryption. AES-CBC-256 can be selected with *encryption aes 256*:

```
device # conf t
device (config)#crypto isakmp policy 1
device (config-isakmp)# hash sha
device (config-isakmp)# encryption aes
```

# Configure IPSec Using Internet Key Exchange Version 1

Follow the procedure given below to configure IPsec IKEv1 to use AES-CBC-128 for payload encryption:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **crypto isakmp policy** *priority*<br>**Example:**<br>`Device(config)# crypto isakmp policy 1` | Defines an Internet Key Exchange (IKE) policy and assigns a priority to the policy.<br>• *priority*: Uniquely identifies the IKE policy and assigns a priority to the policy. Valid values: 1 to 10,000; 1 is the highest priority. |
| **Step 3** | **hash sha**<br>**Example:**<br>`Device(config-isakmp)# hash sha` | Specifies the hash algorithm. |
| **Step 4** | **encryption aes**<br>**Example:**<br>`Device(config-isakmp)# encryption aes` | Configures IPsec IKEv1 to use AES-CBC-128 for payload encryption. AES-CBC-256 can be selected with 'encryption aes 256'. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** The authorized administrator must ensure that the keysize for this setting is greater than or equal to the keysize selected for ESP in section IPsec Transforms and Lifetimes. If AES 128 is selected here, then the highest keysize that can be selected on the device for ESP is AES 128 (either CBC or GCM).<br>Both confidentiality and integrity are configured with the **hash sha** and **encryption aes** commands respectively. As a result, confidentiality-only mode is disabled. |
| **Step 5** | **authentication pre-share**<br>**Example:**<br>`Device(config-isakmp)# authentication pre-share` | Configures IPsec to use the specified preshared keys as the authentication method. Preshared keys require that you separately configure these preshared keys. |
| **Step 6** | **exit**<br>**Example:**<br>`Device(config-isakmp)# exit` | Exits config-isakmp configuration mode. |
| **Step 7** | **crypto isakmp key** *keystring* **address** *peer-address*<br>**Example:**<br>`Device(config)# crypto isakmp key cisco123!cisco123!CISC address 192.0.2.1` | Configures a preshared authentication key.<br>**Note** To ensure a secure configuration, we recommend that you enter the pre-shared keys with at least 22 characters in length and can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")").<br>The device supports pre-shared keys up to 127 characters in length. While longer keys increase the difficulty of brute-force attacks, longer keys increase processing time. |
| **Step 8** | **group 14**<br>**Example:**<br>`Device(config-isakmp)# group 14` | Specifies the Diffie-Hellman (DH) group identifier as 2048-bit DH group 14 and selects DH Group 14 (2048-bit MODP) for IKE. However, 19 (256-bit Random ECP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), 15 (3072 bit MODP), |

| | Command or Action | Purpose |
|---|---|---|
| | | and 16 (4096-bit MODP) are also allowed and supported. |
| **Step 9** | **lifetime** *seconds*<br>**Example:**<br>`Device(config-isakmp)# lifetime 86400` | Specifies the lifetime of the IKE SA. The default time value for Phase 1 SAs is 24 hours (86400 seconds), but this setting can be changed using the command above with different values.<br>• *seconds*: Time, in seconds, before each SA expires. Valid values: 60 to 86,400; default value: 86,400.<br>**Note** The shorter the lifetime (up to a point), the more secure your IKE negotiations will be. However, with longer lifetimes, future IPsec SAs can be set up more quickly. |
| **Step 10** | **crypto isakmp aggressive-mode disable**<br>**Example:**<br>`Device(config-isakmp)# crypto isakmp aggressive-mode disable` | Ensures all IKEv1 Phase 1 exchanges will be handled in the default main mode. |
| **Step 11** | **exit**<br>**Example:**<br>`Device(config-isakmp)# exit` | Exits config-isakmp configuration mode. |

# Internet Key Exchange Version 2 Transform Sets

An Internet Key Exchange Version 2 (IKEv2) proposal is a set of transforms used in the negotiation of IKEv2 SA as part of the IKE_SA_INIT exchange. An IKEv2 proposal is regarded as complete only when it has at least an encryption algorithm, an integrity algorithm, and a Diffie-Hellman (DH) group configured. If no proposal is configured and attached to an IKEv2 policy, then the default proposal is used in the negotiation. The following snippet helps in configuring the IPsec with IKEv2 functionality for the device:

```
device # conf t
device(config)#crypto ikev2 proposal sample
device(config-ikev2-proposal)# integrity sha1
device (config-ikev2-proposal)# encryption aes-cbc-128
device(config-ikev2-proposal)# group 14
device(config-ikev2-proposal)# exit
device(config)# crypto ikev2 keyring keyring-1
device (config-ikev2-keyring)# peer peer1
device (config-ikev2-keyring-peer)# address 192.0.2.4 255.255.255.0
device (config-ikev2-keyring-peer)# pre-shared-key cisco123!cisco123!CISC
device (config-ikev2-keyring-peer)# exit
device(config)#crypto ikev2 keyring keyring-1
```

```
device (config-ikev2-keyring)# peer peer1
device (config-ikev2-keyring-peer)# address 192.0.2.4 255.255.255.0
device (config-ikev2-keyring-peer)# pre-shared-key cisco123!cisco123!CISC
device (config-ikev2-keyring-peer)# exit
device(config)#crypto logging ikev2
```

# Configure IPSec Using Internet Key Exchange Version 2

Follow the procedure given below to configure the IPsec with IKEv2:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **crypto ikev2 proposal** *name*<br>**Example:**<br>`Device(config)# crypto ikev2 proposal name` | Defines an IKEv2 proposal name. |
| **Step 3** | **integrity sha1**<br>**Example:**<br>`Device(config-ikev2-proposal)# integrity sha1` | Defines an IKEv2 proposal name. |
| **Step 4** | **encryption aes-cbc-128**<br>**Example:**<br>`Device(config-ikev2-proposal)# encryption aes-cbc-128` | Configures IPsec IKEv2 to use AES-CBC-128 for payload encryption. AES-CBC-256 can be selected with encryption aes-cbc-256. AES-GCM-128 and AES-GCM-256 can also be selected similarly.<br>**Note** The authorized administrator must ensure that the keysize for this setting is greater than or equal to the keysize selected for ESP in section IPsec Transforms and Lifetimes. If AES 128 is selected here, then the highest keysize that can be selected on the device for ESP is AES 128 (either CBC or GCM).<br>Both confidentiality and integrity are configured with the **hash sha** and **encryption aes** commands respectively. As a result, confidentiality-only mode is disabled. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **group 14**<br>**Example:**<br>`Device(config-ikev2-proposal)# group 14` | Selects DH Group 14 (2048-bit MODP) for IKE. However, 19 (256-bit Random ECP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), 15 (3072 bit MODP), and 16 (4096-bit MODP) are also allowed and supported. |
| **Step 6** | **exit**<br>**Example:**<br>`Device(config-ikev2-proposal)# exit` | Exists IKEv2 proposal configuration mode. |
| **Step 7** | **crypto ikev2 keyring** *keyring-name*<br>**Example:**<br>`Device(config)# crypto ikev2 keyring keyring-1` | Defines an IKEv2 keyring. |
| **Step 8** | **peer** *peer-name*<br>**Example:**<br>`Device(config-ikev2-keyring)# peer peer1` | Defines the peer or peer group. |
| **Step 9** | **address {ipv4-address [mask] \| ipv6-address prefix}**<br>**Example:**<br>`Device(config-ikev2-keyring)# address 192.0.2.4 255.255.255.0` | Specifies an IPv4 or IPv6 address or range for the peer.<br>**Note** This IP address is the IKE endpoint address and is independent of the identity address. |
| **Step 10** | **pre-shared-key** *local*<br>**Example:**<br>`Device(config-ikev2-keyring)# pre-shared-key cisco123!cisco123!CISC` | Specifies the preshared key for the peer. You can enter the local or remote keyword to specify an asymmetric preshared key. By default, the preshared key is symmetric. |

| | Command or Action | Purpose |
|---|---|---|
| | | **Note** To ensure a secure configuration, we recommend that you enter the pre-shared keys with at least 22 characters in length and can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")").<br>The device supports pre-shared keys up to 127 characters in length. While longer keys increase the difficulty of brute-force attacks, longer keys increase processing time.<br>HEX keys generated off system can also be input for IKEv2 using the following instead of the pre-shared-key command above: *pre-shared-key hex [hex key]*. For example: pre-shared-key hex 0x6A6B6C. This configures IPsec to use pre-shared keys. |
| **Step 11** | **exit**<br>**Example:**<br>`Device(config-ikev2-keyring)# exit` | Exits IKEv2 keyring peer configuration mode. |
| **Step 12** | **crypto logging ikev2**<br>**Example:**<br>`Device(config)# crypto logging ikev2` | Enables IKEv2 syslog messages.<br>**Note** The configuration above is not a complete IKE v2 configuration, and that additional settings will be needed. |

# IPsec Transforms and Lifetimes

Regardless of the IKE version selected, the device must be configured with the proper transform for IPsec ESP encryption and integrity as well as IPsec lifetimes.

`device (config)#` **`crypto ipsec transform-set example esp-aes 128 esp-sha-hmac`**

Note that this configures IPsec ESP to use HMAC-SHA-1 and AES-CBC-128. To change this to the other allowed algorithms the following options can replace **esp-aes 128** in the command above:

| Encryption Algorithm | Command |
|---|---|
| AES-CBC-256 | **esp-aes 256** |
| AES-GCM-128 | **esp-gcm 128** |

| Encryption Algorithm | Command |
|---|---|
| AES-GCM-256 | **esp-gcm 256** |

**Note** The size of the key selected here must be less than or equal to the key size selected for the IKE encryption setting. If AES-CBC-128 was selected there for use with IKE encryption, then only AES-CBC-128 or AES-GCM-128 may be selected here.

```
device(config-crypto)# mode tunnel
```

This configures tunnel mode for IPsec. Tunnel is the default, but by explicitly specifying tunnel mode, the device will request tunnel mode and will accept only tunnel mode.

```
device(config-crypto)# mode transport
```

This configures transport mode for IPsec.

```
device(config)# crypto ipsec security-association lifetime seconds 28800
```

The default time value for Phase 2 SAs is 1 hour. There is no configuration required for this setting since the default is acceptable. However to change the setting to 8 hours as claimed in the Security Target the crypto ipsec security-association lifetime command can be used as specified above.

```
device(config)# crypto ipsec security-association lifetime kilobytes 100000
```

This configures a lifetime of 100 MB of traffic for Phase 2 SAs. The default amount for this setting is 2560KB, which is the minimum configurable value for this command. The maximum configurable value for this command is 4GB.

# Use of X.509 With Internet Key Exchange Version

Cisco Catalyst 9800 Series Wireless Controller supports RSA and ECDSA based certificates.

Once X.509v3 keys are installed on the device, they can be set for use with IKEv1 with the commands:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **crypto isakmp** *policy-name*<br>**Example:**<br>`Device(config)#crypto isakmp policy 1` | Defines an Internet Key Exchange (IKE) policy and assigns a priority to the policy. |
| **Step 3** | **authentication [remote \| local] rsa-sig**<br>**Example:** | Uses RSA based certificates for IKEv1 authentication. |

| | Command or Action | Purpose |
|---|---|---|
| | `Device(config-isakmp)#authentication rsa-sig` | |
| **Step 4** | **authentication [remote \| local] ecdsa-sig**<br>**Example:**<br>`Device(config-isakmp)#authentication ecdsa-sig` | Uses ecdsa based certificates for IKEv1 authentication. |

# For IKEv2 Commands

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **crypto ikev2 profile sample**<br>**Example:**<br>`Device(config)# crypto ikev2 profile sample` | Defines an Internet Key Exchange (IKE) policy and assigns a profile. |
| **Step 3** | **authentication [remote \| local] rsa-sig**<br>**Example:**<br>`Device(config-ikev2-profile)# authentication rsa-sig` | Uses RSA based certificates for IKEv1 authentication. |
| **Step 4** | **authentication [remote \| local] ecdsa-sig**<br>**Example:**<br>`Device(config-ikev2-profile)# authentication ecdsa-sig` | Uses ecdsa based certificates for IKEv1 authentication.<br>Authentication fails if an invalid certificate is loaded. |

# IPsec Session Interruption and Recovery

If an IPsec session with a peer is unexpectedly interrupted, the connection will be broken. In this scenario, no administrative interaction is required. The IPsec session will be reestablished (a new SA set up) once the peer is back online.

# Example: Configure IPSec Using ISAKMP

The following sample outputs display the IPSec **isakmp** configuration:

```
crypto isakmp policy 1
 encr aes 256
```

```
 hash sha256
 authentication pre-share
 group 14
 lifetime 28800

crypto isakmp key 0 Cisco!123 address 192.0.2.4
crypto isakmp peer address 192.0.2.4

crypto ipsec transform-set aes-gcm-256 esp-gcm 256
 mode tunnel

crypto map IPSEC_ewlc_to_syslog 1 ipsec-isakmp
 set peer 192.0.2.4
 set transform-set aes-gcm-256
 match address acl_ewlc_to_syslog

interface Vlan15
 crypto map IPSEC_ewlc_to_syslog
end
```

# Verifying IPSec Traffic

The following example shows how to verify the IPSec traffic configuration in isakmp configuration:

```
Device# show crypto map
Crypto Map IPv4 "IPSEC_ewlc_to_syslog" 1 ipsec-isakmp
        Peer = 192.0.2.4
        Extended IP access list acl_ewlc_to_syslog
            access-list acl_ewlc_to_syslog permit ip host 192.0.2.2 host 192.0.2.4
        Current peer: 192.0.2.4
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): N
        Mixed-mode : Disabled
        Transform sets={
                aes-gcm-256:  { esp-gcm 256  } ,
        }
        Interfaces using crypto map IPSEC_ewlc_to_syslog:
                Vlan15

Device# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
192.0.2.5       192.0.2.4       QM_IDLE           1011 ACTIVE

IPv6 Crypto ISAKMP SA

Device# show crypto ipsec sa

interface: Vlan15
    Crypto map tag: IPSEC_ewlc_to_syslog, local addr 192.0.2.5

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.0.2.5/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (192.0.2.4/255.255.255.255/0/0)
   current_peer 192.0.2.4 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 1626, #pkts encrypt: 1626, #pkts digest: 1626
    #pkts decaps: 1625, #pkts decrypt: 1625, #pkts verify: 1625
    #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
```

```
     local crypto endpt.: 192.0.2.5, remote crypto endpt.: 192.0.2.4
     plaintext mtu 1446, path mtu 1500, ip mtu 1500, ip mtu idb Vlan15
     current outbound spi: 0x17FF2F4C(402599756)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0x4B77AD78(1266134392)
        transform: esp-gcm 256 ,
        in use settings ={Tunnel, }
        conn id: 2041, flow_id: HW:41, sibling_flags FFFFFFFF80004048, crypto map:
IPSEC_ewlc_to_syslog
        sa timing: remaining key lifetime (k/sec): (4607904/1933)
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x17FF2F4C(402599756)
        transform: esp-gcm 256 ,
        in use settings ={Tunnel, }
        conn id: 2042, flow_id: HW:42, sibling_flags FFFFFFFF80004048, crypto map:
IPSEC_ewlc_to_syslog
        sa timing: remaining key lifetime (k/sec): (4607904/1933)
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     outbound ah sas:
     outbound pcp sas:

Device# show ip access-lists acl_ewlc_to_syslog
Extended IP access list acl_ewlc_to_syslog
    10 permit ip host 192.0.2.5 host 192.0.2.4 (17 matches)
```

# Example: Configure IPSec Using Internet Key Exchange Version 2

The following sample outputs display the IPSec **IKEv2** configuration:

```
topology : [192.0.2.6]DUT — (infra) — PEER[192.0.2.9]

ikev2 config in 192.0.2.6 (peer is 192.0.2.9)
hostname for 192.0.2.9: Edison-M1
hostname for 192.0.2.6: prsna-nyquist-192.0.2.6

ip access-list extended ikev2acl
 permit ip host 192.0.2.6 host 192.0.2.9

crypto ikev2 proposal PH1PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy PH1POLICY
 proposal PH1PROPOSAL
```

```
crypto ikev2 keyring PH1KEY
  peer Edison-M1
  address 192.0.2.9
  pre-shared-key Cisco!123Cisco!123Cisco!123


crypto ikev2 profile PH1PROFILE
 match identity remote address 192.0.2.9 255.255.255.255
 authentication remote pre-share
 authentication local pre-share
 keyring local PH1KEY

crypto ipsec transform-set aes256-sha1 esp-aes 256 esp-sha-hmac
 mode tunnel

crypto map ikev2-cryptomap 1 ipsec-isakmp
 set peer 192.0.2.9
 set transform-set aes256-sha1
 set ikev2-profile PH1PROFILE
 match address ikev2acl

interface Vlan15
 ip address 192.0.2.6 255.255.255.0
 crypto map ikev2-cryptomap
```

# Verifying IPSec With Internet Key Exchange Version 2 Traffic

The following example shows how to verify the IPSec traffic configuration in IKEv2 configuration:

```
Device# show ip access-lists
Extended IP access list ikev2acl
    10 permit ip host 192.0.2.6 host 192.0.2.9 (80 matches)

prsna-nyquist-192.0.2.6#show crypto map
Crypto Map IPv4 "ikev2-cryptomap" 1 ipsec-isakmp
        Peer = 192.0.2.9
        IKEv2 Profile: PH1PROFILE
        Extended IP access list ikev2acl
            access-list ikev2acl permit ip host 192.0.2.6 host 192.0.2.9
        Current peer: 192.0.2.9
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): N
        Mixed-mode : Disabled
        Transform sets={
                aes256-sha1:  { esp-256-aes esp-sha-hmac  } ,
        }
        Interfaces using crypto map ikev2-cryptomap:
                Vlan15
Device# show crypto ikev2 sa detailed
 IPv4 Crypto IKEv2  SA

Tunnel-id Local                 Remote                fvrf/ivrf            Status
1         192.0.2.6/500         192.0.2.9/500          none/none            READY
      Encr: AES-CBC, keysize: 256, PRF: SHA256, Hash: SHA256, DH Grp:14, Auth sign: PSK,
Auth verify: PSK
      Life/Active Time: 86400/1002 sec
      CE id: 1089, Session-id: 2
      Status Description: Negotiation done
      Local spi: 271D20169FE91074       Remote spi: 13895472E3B910AF
      Local id: 192.0.2.6
      Remote id: 192.0.2.9
```

```
      Local req msg id:  2              Remote req msg id:  0
      Local next msg id: 2              Remote next msg id: 0
      Local req queued:  2              Remote req queued:  0
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      Fragmentation not  configured.
      Dynamic Route Update: disabled
      Extended Authentication not configured.
      NAT-T is not detected
      Cisco Trust Security SGT is disabled
      Initiator of SA : Yes
Device# show crypto ipsec sa detail

interface: Vlan15
    Crypto map tag: ikev2-cryptomap, local addr 192.0.2.6

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.0.2.6/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (192.0.2.9/255.255.255.255/0/0)
   current_peer 192.0.2.9 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 80, #pkts encrypt:80, #pkts digest: 80
    #pkts decaps: 80, #pkts decrypt: 80, #pkts verify: 80
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #pkts no sa (send) 0, #pkts invalid sa (rcv) 0
    #pkts encaps failed (send) 0, #pkts decaps failed (rcv) 0
    #pkts invalid prot (recv) 0, #pkts verify failed: 0
    #pkts invalid identity (recv) 0, #pkts invalid len (rcv) 0
    #pkts replay rollover (send): 0, #pkts replay rollover (rcv) 0
    ##pkts replay failed (rcv): 0
    #pkts tagged (send): 0, #pkts untagged (rcv): 0
    #pkts not tagged (send): 0, #pkts not untagged (rcv): 0
    #pkts internal err (send): 0, #pkts internal err (recv) 0

     local crypto endpt.: 192.0.2.6, remote crypto endpt.: 192.0.2.9
     plaintext mtu 1438, path mtu 1500, ip mtu 1500, ip mtu idb Vlan15
     current outbound spi: 0xB546157A(3041269114)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0x350925BC(889791932)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 838, flow_id: 838, sibling_flags FFFFFFFF80000040, crypto map:
ikev2-cryptomap
        sa timing: remaining key lifetime (k/sec): (4287660676/2560)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xB546157A(3041269114)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 837, flow_id: 837, sibling_flags FFFFFFFF80000040, crypto map:
ikev2-cryptomap
        sa timing: remaining key lifetime (k/sec): (4287660672/2560)
        IV size: 16 bytes
```

```
   replay detection support: Y
   Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:
```