# Configure Security Parameters

**Note** To achieve simplification and consistency, the Cisco SD-WAN solution has been rebranded as Cisco Catalyst SD-WAN. In addition, from Cisco IOS XE SD-WAN Release 17.12.1a and Cisco Catalyst SD-WAN Release 20.12.1, the following component changes are applicable: **Cisco vManage** to **Cisco Catalyst SD-WAN Manager**, **Cisco vAnalytics** to **Cisco Catalyst SD-WAN Analytics**, **Cisco vBond** to **Cisco Catalyst SD-WAN Validator**, and **Cisco vSmart** to **Cisco Catalyst SD-WAN Controller**. See the latest Release Notes for a comprehensive list of all the component brand name changes. While we transition to the new names, some inconsistencies might be present in the documentation set because of a phased approach to the user interface updates of the software product.

This section describes how to change security parameters for the control plane and the data plane in the Cisco Catalyst SD-WAN overlay network.



# Configure Control Plane Security Parameters

By default, the control plane uses DTLS as the protocol that provides privacy on all its tunnels. DTLS runs over UDP.

You can change the control plane security protocol to TLS, which runs over TCP. The primary reason to use TLS is that, if you consider the Cisco SD-WAN Controller to be a server, firewalls protect TCP servers better than UDP servers.

You configure the control plane tunnel protocol on a Cisco SD-WAN Controller:

```
vSmart(config)# security control protocol tls
```

With this change, all control plane tunnels between the Cisco SD-WAN Controller and the routers and between the Cisco SD-WAN Controller and Cisco SD-WAN Manager use TLS. Control plane tunnels to Cisco Catalyst SD-WAN Validator always use DTLS, because these connections must be handled by UDP.

In a domain with multiple Cisco SD-WAN Controllers, when you configure TLS on one of the Cisco SD-WAN Controllers, all control plane tunnels from that controller to the other controllers use TLS. Said another way, TLS always takes precedence over DTLS. However, from the perspective of the other Cisco SD-WAN

Controllers, if you have not configured TLS on them, they use TLS on the control plane tunnel only to that one Cisco SD-WAN Controller, and they use DTLS tunnels to all the other Cisco SD-WAN Controllers and to all their connected routers. To have all Cisco SD-WAN Controllers use TLS, configure it on all of them.

By default, the Cisco SD-WAN Controller listens on port 23456 for TLS requests. To change this:

```
vSmart(config)# security control tls-port number
```

The port can be a number from 1025 through 65535.

To display control plane security information, use the **show control connections** command on the Cisco SD-WAN Controller. For example:

```
vSmart-2# show control connections

                                                                    PEER                    PEER
PEER    PEER     PEER            SITE   DOMAIN  PEER             PRIVATE  PEER           PUBLIC
TYPE    PROTOCOL SYSTEM IP       ID     ID      PRIVATE IP       PORT     PUBLIC IP      PORT
REMOTE COLOR  STATE  UPTIME
------------------------------------------------------------------------------------------------------
vedge   dtls     172.16.255.11   100    1       10.0.5.11        12346    10.0.5.11      12346
lte           up    0:07:48:58
vedge   dtls     172.16.255.21   100    1       10.0.5.21        12346    10.0.5.21      12346
lte           up    0:07:48:51
vedge   dtls     172.16.255.14   400    1       10.1.14.14       12360    10.1.14.14     12360
lte           up    0:07:49:02
vedge   dtls     172.16.255.15   500    1       10.1.15.15       12346    10.1.15.15     12346
default       up    0:07:47:18
vedge   dtls     172.16.255.16   600    1       10.1.16.16       12346    10.1.16.16     12346
default       up    0:07:41:52
vsmart  tls      172.16.255.19   100    1       10.0.5.19        12345    10.0.5.19      12345
default       up    0:00:01:44
vbond   dtls     -               0      0       10.1.14.14       12346    10.1.14.14     12346
default       up    0:07:49:08

vSmart-2# control connections

                                                                    PEER                    PEER
PEER     PEER     PEER            SITE   DOMAIN  PEER             PRIVATE  PEER           PUBLIC
TYPE     PROTOCOL SYSTEM IP       ID     ID      PRIVATE IP       PORT     PUBLIC IP      PORT
 REMOTE COLOR  STATE   UPTIME
------------------------------------------------------------------------------------------------------
vedge    tls      172.16.255.11   100    1       10.0.5.11        12345    10.0.5.11      12345
 lte           up     0:00:01:18
vedge    tls      172.16.255.21   100    1       10.0.5.21        12345    10.0.5.21      12345
 lte           up     0:00:01:18
vedge    tls      172.16.255.14   400    1       10.1.14.14       12345    10.1.14.14     12345
 lte           up     0:00:01:18
vedge    tls      172.16.255.15   500    1       10.1.15.15       12345    10.1.15.15     12345
 default       up     0:00:01:18
vedge    tls      172.16.255.16   600    1       10.1.16.16       12345    10.1.16.16     12345
 default       up     0:00:01:18
vsmart   tls      172.16.255.20   200    1       10.0.12.20       23456    10.0.12.20     23456
 default       up     0:00:01:32
vbond    dtls     -               0      0       10.1.14.14       12346    10.1.14.14     12346
 default       up     0:00:01:33
```

# Configure DTLS in Cisco SD-WAN Manager

If you configure the Cisco SD-WAN Manager to use TLS as the control plane security protocol, you must enable port forwarding on your NAT. If you are using DTLS as the control plane security protocol, you do not need to do anything.

The number of ports forwarded depends on the number of vdaemon processes running on the Cisco SD-WAN Manager. To display information about these processes and about and the number of ports that are being forwarded, use the **show control summary** command shows that four vdaemon processes are running:

```
vManage# show control summary
          VBOND     VMANAGE    VSMART    VEDGE
INSTANCE  COUNTS    COUNTS     COUNTS    COUNTS
-----------------------------------------------
0           2          0          2        7
1           2          0          0        5
2           2          0          0        5
3           2          0          0        4
```

To see the listening ports, use the **show control local-properties** command:

```
vManage# show control local-properties

organization-name            Cisco SD-WAN Inc Test
certificate-status           Installed
root-ca-chain-status         Installed

certificate-validity         Valid
certificate-not-valid-before May 20 00:00:00 2015 GMT
certificate-not-valid-after  May 20 23:59:59 2016 GMT

dns-name                     vbond.cisco.com
site-id                      5000
domain-id                    0
protocol                     dtls
tls-port                     23456
...
...
...
number-active-wan-interfaces 1

                PUBLIC       PUBLIC PRIVATE       PRIVATE
  ADMIN  OPERATION LAST
INDEX INTERFACE IP           PORT   IP            PORT    VSMARTS  VMANAGES COLOR  CARRIER
  STATE  STATE     CONNECTION
-------------------------------------------------------------------------------------------------------
0     eth0      72.28.108.37 12361  172.16.98.150 12361   2        0        silver default
  up     up       0:00:00:08
```

This output shows that the listening TCP port is 23456. If you are running Cisco SD-WAN Manager behind a NAT, you should open the following ports on the NAT device:

- 23456 (base - instance 0 port)
- 23456 + 100 (base + 100)
- 23456 + 200 (base + 200)
- 23456 + 300 (base + 300)

Note that the number of instances is the same as the number of cores you have assigned for the Cisco SD-WAN Manager, up to a maximum of 8.

# Configure Security Parameters Using the Security Feature Template

Use the Security feature template for all Cisco vEdge devices. On the edge routers and on Cisco SD-WAN Validator, use this template to configure IPsec for data plane security. On Cisco SD-WAN Manager and Cisco SD-WAN Controller, use the Security feature template to configure DTLS or TLS for control plane security.

### Configure Security Parameters

1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.
2. Click **Feature Templates** and then click **Add Template**.

**Note** In Cisco vManage Release 20.7.1 and earlier releases, **Feature Templates** is called **Feature**.

3. From the Devices list in the left pane, choose a device.

   The templates applicable to the selected device appear in the right pane.
4. Click **Security** to open the template.
5. In the **Template Name** field, enter a name for the template. The name can be up to 128 characters and can contain only alphanumeric characters.
6. In the **Template Description** field, enter a description of the template. The description can be up to 2048 characters and can contain only alphanumeric characters.

When you first open a feature template, for each parameter that has a default value, the scope is set to Default (indicated by a check mark), and the default setting or value is shown. To change the default or to enter a value, click the scope drop-down menu to the left of the parameter field and choose one of the following:

***Table 1:***

| Parameter Scope | Scope Description |
|---|---|
| Device Specific (indicated by a host icon) | Use a device-specific value for the parameter. For device-specific parameters, you cannot enter a value in the feature template. You enter the value when you attach a Viptela device to a device template .<br><br>When you click Device Specific, the Enter Key box opens. This box displays a key, which is a unique string that identifies the parameter in a CSV file that you create. This file is an Excel spreadsheet that contains one column for each key. The header row contains the key names (one key per column), and each row after that corresponds to a device and defines the values of the keys for that device. You upload the CSV file when you attach a Viptela device to a device template. For more information, see Create a Template Variables Spreadsheet .<br><br>To change the default key, type a new string and move the cursor out of the Enter Key box.<br><br>Examples of device-specific parameters are system IP address, hostname, GPS location, and site ID. |

| Parameter Scope | Scope Description |
|---|---|
| Global (indicated by a globe icon) | Enter a value for the parameter, and apply that value to all devices.<br>Examples of parameters that you might apply globally to a group of devices are DNS server, syslog server, and interface MTUs. |

### Configure Control Plane Security

**Note** The Configure Control Plane Security section is applicable to Cisco SD-WAN Manager and Cisco SD-WAN Controller only.

To configure the control plane connection protocol on a Cisco SD-WAN Manager instance or a Cisco SD-WAN Controller, choose the **Basic Configuration** area and configure the following parameters:

***Table 2:***

| Parameter Name | Description |
|---|---|
| Protocol | Choose the protocol to use on control plane connections to a Cisco SD-WAN Controller:<br>• DTLS (Datagram Transport Layer Security). This is the default.<br>• TLS (Transport Layer Security) |
| Control TLS Port | If you selected TLS, configure the port number to use:*Range:* 1025 through 65535*Default:* 23456 |

Click **Save**

### Configure Data Plane Security

To configure data plane security on a Cisco SD-WAN Validator or a Cisco vEdge router, choose the **Basic Configuration** and **Authentication Type** tabs, and configure the following parameters:

***Table 3:***

| Parameter Name | Description |
|---|---|
| Rekey Time | Specify how often a Cisco vEdge router changes the AES key used on its secure DTLS connection to the Cisco SD-WAN Controller. If OMP graceful restart is enabled, the rekeying time must be at least twice the value of the OMP graceful restart timer.*Range:* 10 through 1209600 seconds (14 days)*Default:* 86400 seconds (24 hours) |
| Replay Window | Specify the size of the sliding replay window.<br>*Values:* 64, 128, 256, 512, 1024, 2048, 4096, 8192 packets*Default:* 512 packets |
| IPsec pairwise-keying | This is turned off by default. Click **On** to turn it on. |

| Parameter Name | Description |
|---|---|
| Authentication Type | Select the authentication types from the **Authentication List**, and click the arrow pointing right to move the authentication types to the **Selected List** column.<br><br>Authentication types supported from Cisco SD-WAN Release 20.6.1:<br><br>• **esp**: Enables Encapsulating Security Payload (ESP) encryption and integrity checking on the ESP header.<br><br>• **ip-udp-esp:** Enables ESP encryption. In addition to the integrity checks on the ESP header and payload, the checks also include the outer IP and UDP headers.<br><br>• **ip-udp-esp-no-id**: Ignores the ID field in the IP header so that Cisco Catalyst SD-WAN can work in conjunction with the non-Cisco devices.<br><br>• **none**: Turns integrity checking off on IPSec packets. We don't recommend using this option.<br><br>Authentication types supported in Cisco SD-WAN Release 20.5.1 and earlier:<br><br>• **ah-no-id**: Enable an enhanced version of AH-SHA1 HMAC and ESP HMAC-SHA1 that ignores the ID field in the packet's outer IP header.<br><br>• **ah-sha1-hmac**: Enable AH-SHA1 HMAC and ESP HMAC-SHA1.<br><br>• **none**: Select no authentication.<br><br>• **sha1-hmac**: Enable ESP HMAC-SHA1.<br><br>**Note** For an edge device running on Cisco SD-WAN Release 20.5.1 or earlier, you may have configured authentication types using a **Security** template. When you upgrade the device to Cisco SD-WAN Release 20.6.1 or later, update the selected authentication types in the **Security** template to the authentication types supported from Cisco SD-WAN Release 20.6.1. To update the authentication types, do the following:<br><br>1. From the Cisco SD-WAN Manager menu, choose **Configuration** > **Templates**.<br><br>2. Click **Feature Templates**.<br><br>3. Find the **Security** template to update and click … and click **Edit**.<br><br>4. Click **Update**. Do not modify any configuration.<br><br>Cisco SD-WAN Manager updates the **Security** template to display the supported authentication types. |

Click **Save**.

# Configure Data Plane Security Parameters

In the data plane, IPsec is enabled by default on all routers, and by default IPsec tunnel connections use an enhanced version of the Encapsulating Security Payload (ESP) protocol for authentication on IPsec tunnels.

On the routers, you can change the type of authentication, the IPsec rekeying timer, and the size of the IPsec anti-replay window.

# Configure Allowed Authentication Types

### Authentication Types in Cisco SD-WAN Release 20.6.1 and Later

From Cisco SD-WAN Release 20.6.1, the following integrity types are supported:

- **esp:** This option enables Encapsulating Security Payload (ESP) encryption and integrity checking on the ESP header.
- **ip-udp-esp:** This option enables ESP encryption. In addition to the integrity checks on the ESP header and the payload, the checks also include the outer IP and UDP headers.
- **ip-udp-esp-no-id:** This option is is similar to ip-udp-esp, however, the ID field of the outer IP header is ignored. Configure this option in the list of integrity types to have the Cisco Catalyst SD-WAN software ignore the ID field in the IP header so that the Cisco Catalyst SD-WAN can work in conjunction with non-Cisco devices.
- **none:** This option turns integrity checking off on IPSec packets. We don't recommend using this option.

By default, IPsec tunnel connections use an enhanced version of the Encapsulating Security Payload (ESP) protocol for authentication. To modify the negotiated interity types or to disable integrity check, use the following command:

**integrity-type** { **none** | **ip-udp-esp** | **ip-udp-esp-no-id** | **esp** }

### Authentication Types Before Cisco SD-WAN Release 20.6.1

By default, IPsec tunnel connections use an enhanced version of the Encapsulating Security Payload (ESP) protocol for authentication. To modify the negotiated authentication types or to disable authentication, use the following command:

```
Device(config)# security ipsec authentication-type (ah-sha1-hmac | ah-no-id | sha1-hmac |
| none)
```

By default, IPsec tunnel connections use AES-GCM-256, which provides both encryption and authentication.

Configure each authentication type with a separate **security ipsec authentication-type** command. The command options map to the following authentication types, which are listed in order from most strong to least strong:

**Note** The `sha1` in the configuration options is used for historical reasons. The authentication options indicate over how much of the packet integrity checking is done. They do not specify the algorithm that checks the integrity. Except for the encryption of multicast traffic, the authentication algorithms supported by Cisco Catalyst SD-WAN do not use SHA1. However in Cisco SD-WAN Release 20.1.x and onwards, both unicast and multicast do not use SHA1.

- **ah-sha1-hmac** enables encryption and encapsulation using ESP. However, in addition to the integrity checks on the ESP header and payload, the checks also include the outer IP and UDP headers. Hence,

this option supports an integrity check of the packet similar to the Authentication Header (AH) protocol. All integrity and encryption is performed using AES-256-GCM.

- **ah-no-id** enables a mode that is similar to **ah-sha1-hmac**, however the ID field of the outer IP header is ignored. This option accommodates some non-Cisco Catalyst SD-WAN devices, including the Apple AirPort Express NAT, that have a bug that causes the ID field in the IP header, a non-mutable field, to be modified. Configure the **ah-no-id** option in the list of authentication types to have the Cisco Catalyst SD-WAN AH software ignore the ID field in the IP header so that the Cisco Catalyst SD-WAN software can work in conjunction with these devices.
- **sha1-hmac** enables ESP encryption and integrity checking.
- **none** maps to no authentication. This option should only be used if it is required for temporary debugging. You can also choose this option in situations where data plane authentication and integrity are not a concern. Cisco does not recommend using this option for production networks.

For information about which data packet fields are affected by these authentication types, see Data Plane Integrity.

Cisco IOS XE Catalyst SD-WAN devices and Cisco vEdge devices advertise their configured authentication types in their TLOC properties. The two routers on either side of an IPsec tunnel connection negotiate the authentication to use on the connection between them, using the strongest authentication type that is configured on both of the routers. For example, if one router advertises the `ah-sha1-hmac` and `ah-no-id` types, and a second router advertises the `ah-no-id` type, the two routers negotiate to use `ah-no-id` on the IPsec tunnel connection between them. If no common authentication types are configured on the two peers, no IPsec tunnel is established between them.

The encryption algorithm on IPsec tunnel connections depends on the type of traffic:

- For unicast traffic, the encryption algorithm is AES-256-GCM.
- For multicast traffic:
  - Cisco SD-WAN Release 20.1.x and later– the encryption algorithm is AES-256-GCM
  - Previous releases– the encryption algorithm is AES-256-CBC with SHA1-HMAC.

When the IPsec authentication type is changed, the AES key for the data path is changed.

# Change the Rekeying Timer

Before Cisco IOS XE Catalyst SD-WAN devices and Cisco vEdge devices can exchange data traffic, they set up a secure authenticated communications channel between them. The routers use IPSec tunnels between them as the channel, and the AES-256 cipher to perform encryption. Each router generates a new AES key for its data path periodically.

By default, a key is valid for 86400 seconds (24 hours), and the timer range is 10 seconds through 1209600 seconds (14 days). To change the rekey timer value:

```
Device(config)# security ipsec
rekey seconds
```

The configuration looks like this:

```
security
    ipsec
```

```
    rekey seconds
  !
```

If you want to generate new IPsec keys immediately, you can do so without modifying the configuration of the router. To do this, issue the **request security ipsec-rekey** command on the compromised router.

For example, the following output shows that the local SA has a Security Parameter Index (SPI) of 256:

```
Device# show ipsec local-sa

                                          SOURCE           SOURCE
TLOC ADDRESS     TLOC COLOR       SPI     IP               PORT    KEY HASH
-------------------------------------------------------------------------------
172.16.255.15    lte              256     10.1.15.15       12346   *****b93a
```

A unique key is associated with each SPI. If this key is compromised, use the **request security ipsec-rekey** command to generate a new key immediately. This command increments the SPI. In our example, the SPI changes to 257 and the key associated with it is now used:

```
Device# request security ipsec-rekey
Device# show ipsec local-sa
                                          SOURCE           SOURCE
TLOC ADDRESS     TLOC COLOR       SPI     IP               PORT    KEY HASH
-------------------------------------------------------------------------------
172.16.255.15    lte              257     10.1.15.15       12346   *****b93a
```

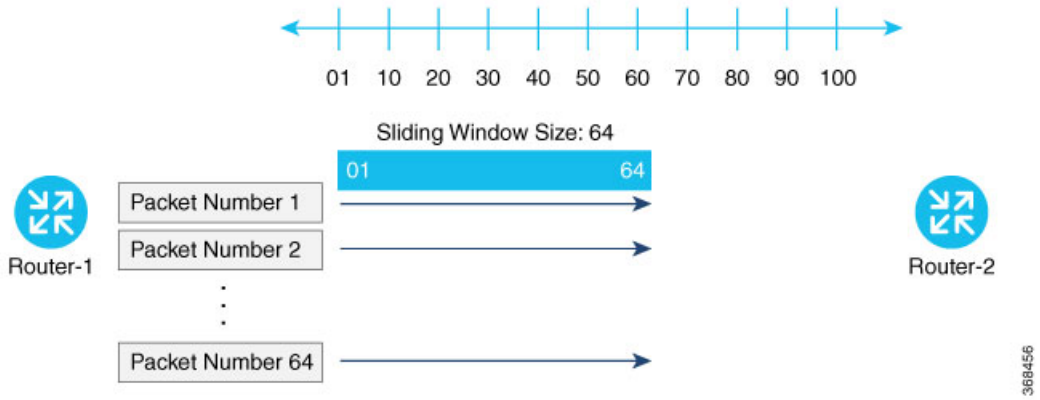
After the new key is generated, the router sends it immediately to the Cisco SD-WAN Controllers using DTLS or TLS. The Cisco SD-WAN Controllers send the key to the peer routers. The routers begin using it as soon as they receive it. Note that the key associated with the old SPI (256) will continue to be used for a short period of time, until it times out.

To stop using the old key immediately, issue the **request security ipsec-rekey** command twice, in quick succession. This sequence of commands removes both SPI 256 and 257 and sets the SPI to 258. The router then uses the associated key of SPI 258. Note, however, that some packets will be dropped for a short period of time, until all the remote routers learn the new key.

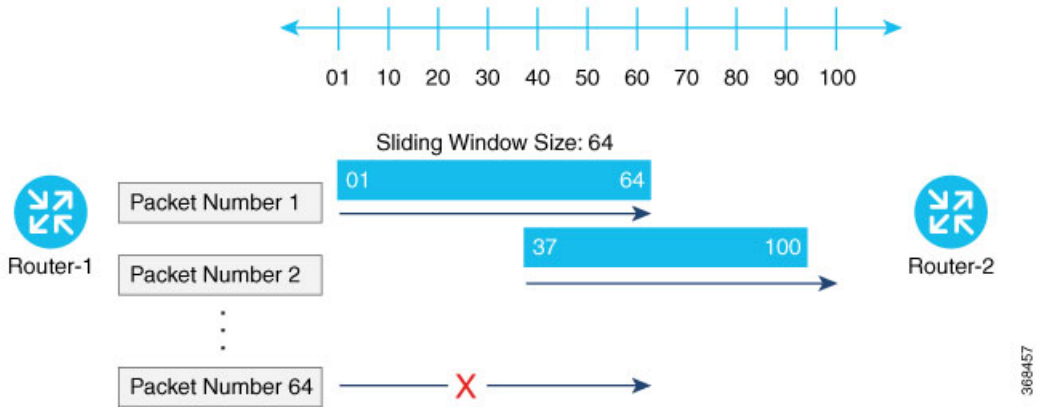
```
Device# request security ipsec-rekey
Device# request security ipsec-rekey
Device# ipsec local-sa
                                          SOURCE           SOURCE
TLOC ADDRESS     TLOC COLOR       SPI     IP               PORT    KEY HASH
-------------------------------------------------------------------------------
172.16.255.15    lte              258     10.1.15.15       12346   *****b93a
```

# Change the Size of the Anti-Replay Window

IPsec authentication provides anti-replay protection by assigning a unique sequence number to each packet in a data stream. This sequence numbering protects against an attacker duplicating data packets. With anti-replay protection, the sender assigns monotonically increasing sequence numbers, and the destination checks these sequence numbers to detect duplicates. Because packets often do not arrive in order, the destination maintains a sliding window of sequence numbers that it will accept.




Packets with sequence numbers that fall to the left of the sliding window range are considered old or duplicates, and the destination drops them. The destination tracks the highest sequence number it has received, and adjusts the sliding window when it receives a packet with a higher value.




By default, the sliding window is set to 512 packets. It can be set to any value between 64 and 4096 that is a power of 2 (that is, 64, 128, 256, 512, 1024, 2048, or 4096). To modify the anti-replay window size, use the **replay-window** command, specifying the size of the window:

```
Device(config)# security ipsec replay-window
number
```

The configuration looks like this:

```
security
   ipsec
     replay-window number
   !
!
```

To help with QoS, separate replay windows are maintained for each of the first eight traffic channels. The configured replay window size is divided by eight for each channel.

If QoS is configured on a router, that router might experience a larger than expected number of packet drops as a result of the IPsec anti-replay mechanism, and many of the packets that are dropped are legitimate ones. This occurs because QoS reorders packets, giving higher-priority packets preferential treatment and delaying lower-priority packets. To minimize or prevent this situation, you can do the following:

- Increase the size of the anti-replay window.
- Engineer traffic onto the first eight traffic channels to ensure that traffic within a channel is not reordered.

# Configure IKE-Enabled IPsec Tunnels

To securely transfer traffic from the overlay network to a service network, you can configure IPsec tunnels that run the Internet Key Exchange (IKE) protocol. IKE-enabled IPsec tunnels provide authentication and encryption to ensure secure packet transport.

You create an IKE-enabled IPsec tunnel by configuring an IPsec interface. IPsec interfaces are logical interfaces, and you configure them just like any other physical interface. You configure IKE protocol parameters on the IPsec interface, and you can configure other interface properties.

**Note** Cisco recommends using IKE Version 2.

**Note** From Cisco SD-WAN 19.2.x release onwards, the pre-shared key needs to be at least 16 bytes in length. The IPsec tunnel establishment fails if the key size is less than 16 characters when the router is upgraded to version 19.2.

The Cisco Catalyst SD-WAN software supports IKE Version 2 as defined in RFC 7296.

One use for IPsec tunnels is to allow vEdge Cloud router VM instances running on Amazon AWS to connect to the Amazon virtual private cloud (VPC). You must configure IKE Version 1 on these routers.

**Note** Cisco vEdge devices support only route-based VPNs in an IPSec configuration because these devices cannot define traffic selectors in the encryption domain.

## Configure an IPsec Tunnel

To configure an IPsec tunnel interface for secure transport traffic from a service network, you create a logical IPsec interface:

```
vEdge(config)# vpn vpn-id interface ipsecnumber
vEdge(config-interface-ipsec)# ip address ipv4-prefix/length
vEdge(config-interface-ipsec)# tunnel-source ip-address | tunnel-source-interface
interface-name)
vEdge(config-interface-ipsec)# tunnel-destination ipv4-address
vEdge(config-interface-ipsec)# no shutdown
```

You can create the IPsec tunnel in the transport VPN (VPN 0) and in any service VPN (VPN 1 through 65530, except for 512).

The IPsec interface has a name in the format **ipsec***number*, where *number* can be from 1 through 255.

Each IPsec interface must have an IPv4 address. This address must be a /30 prefix. All traffic in the VPN that is within this IPv4 prefix is directed to a physical interface in VPN 0 to be sent securely over an IPsec tunnel.

To configure the source of the IPsec tunnel on the local device, you can specify either the IP address of the physical interface (in the **tunnel-source** command) or the name of the physical interface (in the **tunnel-source-interface** command). Ensure that the physical interface is configured in VPN 0.

To configure the destination of the IPsec tunnel, specify the IP address of the remote device in the **tunnel-destination** command.

The combination of a source address (or source interface name) and a destination address defines a single IPsec tunnel. Only one IPsec tunnel can exist that uses a specific source address (or interface name) and destination address pair.

# Configure an IPsec Static Route

To direct traffic from the service VPN to an IPsec tunnel in the transport VPN (VPN 0), you configure an IPsec-specific static route in a service VPN (a VPN other than VPN 0 or VPN 512) :

```
vEdge(config)# vpn vpn-id
vEdge(config-vpn)# ip ipsec-route prefix/length vpn 0 interface
ipsecnumber [ipsecnumber2]
```

The VPN ID is that of any service VPN (VPN 1 through 65530, except for 512).

*prefix*/*length* is the IP address or prefix, in decimal four-part-dotted notation, and prefix length of the IPsec-specific static route.

The interface is the IPsec tunnel interface in VPN 0. You can configure one or two IPsec tunnel interfaces. If you configure two, the first is the primary IPsec tunnel, and the second is the backup. With two interfaces, all packets are sent only to the primary tunnel. If that tunnel fails, all packets are then sent to the secondary tunnel. If the primary tunnel comes back up, all traffic is moved back to the primary IPsec tunnel.

# Enable IKE Version 1

When you create an IPsec tunnel on a vEdge router, IKE Version 1 is enabled by default on the tunnel interface. The following properties are also enabled by default for IKEv1:

- Authentication and encryption—AES-256 advanced encryption standard CBC encryption with the HMAC-SHA1 keyed-hash message authentication code algorithm for integrity
- Diffie-Hellman group number—16
- Rekeying time interval—4 hours
- SA establishment mode—Main

By default, IKEv1 uses IKE main mode to establish IKE SAs. In this mode, six negotiation packets are exchanged to establish the SA. To exchange only three negotiation packets, enable aggressive mode:

**Note** IKE aggressive mode with pre-shared keys should be avoided wherever possible. Otherwise a strong pre-shared key should be chosen.

```
vEdge(config)# vpn vpn-id interface ipsec number ike
vEdge(config-ike)# mode aggressive
```

By default, IKEv1 uses Diffie-Hellman group 16 in the IKE key exchange. This group uses the 4096-bit more modular exponential (MODP) group during IKE key exchange. You can change the group number to 2 (for 1024-bit MODP), 14 (2048-bit MODP), or 15 (3072-bit MODP):

```
vEdge(config)# vpn vpn-id interface ipsec number ike
vEdge(config-ike)# group number
```

By default, IKE key exchange uses AES-256 advanced encryption standard CBC encryption with the HMAC-SHA1 keyed-hash message authentication code algorithm for integrity. You can change the authentication:

```
vEdge(config)# vpn vpn-id interface ipsec number ike
vEdge(config-ike)# cipher-suite suite
```

The authentication *suite* can be one of the following:

- **aes128-cbc-sha1**—AES-128 advanced encryption standard CBC encryption with the HMAC-SHA1 keyed-hash message authentication code algorithm for integrity
- **aes128-cbc-sha2**—AES-128 advanced encryption standard CBC encryption with the HMAC-SHA256 keyed-hash message authentication code algorithm for integrity
- **aes256-cbc-sha1**—AES-256 advanced encryption standard CBC encryption with the HMAC-SHA1 keyed-hash message authentication code algorithm for integrity; this is the default.
- **aes256-cbc-sha2**—AES-256 advanced encryption standard CBC encryption with the HMAC-SHA256 keyed-hash message authentication code algorithm for integrity

By default, IKE keys are refreshed every 1 hours (3600 seconds). You can change the rekeying interval to a value from 30 seconds through 14 days (1209600 seconds). It is recommended that the rekeying interval be at least 1 hour.

```
vEdge(config)# vpn vpn-id interface ipsec number ike
vEdge(config-ike)# rekey seconds
```

To force the generation of new keys for an IKE session, issue the **request ipsec ike-rekey** command.

```
vEdge(config)# vpn vpn-id interfaceipsec number ike
```

For IKE, you can also configure preshared key (PSK) authentication:

```
vEdge(config)# vpn vpn-id interface ipsec number ike
vEdge(config-ike)# authentication-type pre-shared-key pre-shared-secret password
```

*password* is the password to use with the preshared key. It can be an ASCII or a hexadecimal string from 1 through 127 characters long.

If the remote IKE peer requires a local or remote ID, you can configure this identifier:

```
vEdge(config)# vpn vpn-id interface ipsec number ike authentication-type
vEdge(config-authentication-type)# local-id id
vEdge(config-authentication-type)# remote-id id
```

The identifier can be an IP address or any text string from 1 through 63 characters long. By default, the local ID is the tunnel's source IP address and the remote ID is the tunnel's destination IP address.

# Enable IKE Version 2

When you configure an IPsec tunnel to use IKE Version 2, the following properties are also enabled by default for IKEv2:

- Authentication and encryption—AES-256 advanced encryption standard CBC encryption with the HMAC-SHA1 keyed-hash message authentication code algorithm for integrity
- Diffie-Hellman group number—16
- Rekeying time interval—4 hours

By default, IKEv2 uses Diffie-Hellman group 16 in the IKE key exchange. This group uses the 4096-bit more modular exponential (MODP) group during IKE key exchange. You can change the group number to 2 (for 1024-bit MODP), 14 (2048-bit MODP), or 15 (3072-bit MODP):

```
vEdge(config)# vpn vpn-id interface ipsecnumber ike
vEdge(config-ike)# group number
```

By default, IKE key exchange uses AES-256 advanced encryption standard CBC encryption with the HMAC-SHA1 keyed-hash message authentication code algorithm for integrity. You can change the authentication:

```
vEdge(config)# vpn vpn-id interface ipsecnumber ike
vEdge(config-ike)# cipher-suite suite
```

The authentication *suite* can be one of the following:

- **aes128-cbc-sha1**—AES-128 advanced encryption standard CBC encryption with the HMAC-SHA1 keyed-hash message authentication code algorithm for integrity
- **aes128-cbc-sha2**—AES-128 advanced encryption standard CBC encryption with the HMAC-SHA256 keyed-hash message authentication code algorithm for integrity
- **aes256-cbc-sha1**—AES-256 advanced encryption standard CBC encryption with the HMAC-SHA1 keyed-hash message authentication code algorithm for integrity; this is the default.
- **aes256-cbc-sha2**—AES-256 advanced encryption standard CBC encryption with the HMAC-SHA256 keyed-hash message authentication code algorithm for integrity

By default, IKE keys are refreshed every 4 hours (14,400 seconds). You can change the rekeying interval to a value from 30 seconds through 14 days (1209600 seconds):

```
vEdge(config)# vpn vpn-id interface ipsecnumber ike
vEdge(config-ike)#  rekey seconds
```

To force the generation of new keys for an IKE session, issue the **request ipsec ike-rekey** command.

For IKE, you can also configure preshared key (PSK) authentication:

```
vEdge(config)# vpn vpn-id interface ipsecnumber ike
vEdge(config-ike)# authentication-type pre-shared-key pre-shared-secret password
```

*password* is the password to use with the preshared key. It can be an ASCII or a hexadecimal string, or it can be an AES-encrypted key.

If the remote IKE peer requires a local or remote ID, you can configure this identifier:

```
vEdge(config)# vpn vpn-id interface ipsecnumber ike authentication-type
vEdge(config-authentication-type)# local-id id
vEdge(config-authentication-type)# remote-id id
```

The identifier can be an IP address or any text string from 1 through 64 characters long. By default, the local ID is the tunnel's source IP address and the remote ID is the tunnel's destination IP address.

# Configure IPsec Tunnel Parameters

*Table 4: Feature History*

| Feature Name | Release Information | Description |
|---|---|---|
| Additional Cryptographic Algorithmic Support for IPSec Tunnels | Cisco SD-WAN Release 20.1.1 | This feature adds support for `HMAC_SHA256`, `HMAC_SHA384`, and `HMAC_SHA512` algorithms for enhanced security. |

By default, the following parameters are used on the IPsec tunnel that carries IKE traffic:

- Authentication and encryption—AES-256 algorithm in GCM (Galois/counter mode)
- Rekeying interval—4 hours
- Replay window—32 packets

You can change the encryption on the IPsec tunnel to the AES-256 cipher in CBC (cipher block chaining mode, with HMAC using either SHA-1 or SHA-2 keyed-hash message authentication or to null with HMAC using either SHA-1 or SHA-2 keyed-hash message authentication, to not encrypt the IPsec tunnel used for IKE key exchange traffic:

```
vEdge(config-interface-ipsecnumber)# ipsec
vEdge(config-ipsec)# cipher-suite (aes256-gcm | aes256-cbc-sha1 | aes256-cbc-sha256 |
aes256-cbc-sha384 | aes256-cbc-sha512 | aes256-null-sha1 | aes256-null-sha256 |
aes256-null-sha384 | aes256-null-sha512)
```

By default, IKE keys are refreshed every 4 hours (14,400 seconds). You can change the rekeying interval to a value from 30 seconds through 14 days (1209600 seconds):

```
vEdge(config-interface-ipsecnumber)# ipsec
vEdge(config-ipsec)#  rekey seconds
```

To force the generation of new keys for an IPsec tunnel, issue the **request ipsec ipsec-rekey** command.

By default, perfect forward secrecy (PFS) is enabled on IPsec tunnels, to ensure that past sessions are not affected if future keys are compromised. PFS forces a new Diffie-Hellman key exchange, by default using the 4096-bit Diffie-Hellman prime module group. You can change the PFS setting:

```
vEdge(config-interface-ipsecnumber)# ipsec
vEdge(config-ipsec)# perfect-forward-secrecy pfs-setting
```

*pfs-setting* can be one of the following:

- **group-2**—Use the 1024-bit Diffie-Hellman prime modulus group.
- **group-14**—Use the 2048-bit Diffie-Hellman prime modulus group.
- **group-15**—Use the 3072-bit Diffie-Hellman prime modulus group.
- **group-16**—Use the 4096-bit Diffie-Hellman prime modulus group. This is the default.
- **none**—Disable PFS.

By default, the IPsec replay window on the IPsec tunnel is 512 bytes. You can set the replay window size to 64, 128, 256, 512, 1024, 2048, or 4096 packets:

```
vEdge(config-interface-ipsecnumber)# ipsec
vEdge(config-ipsec)# replay-window number
```

# Modify IKE Dead-Peer Detection

IKE uses a dead-peer detection mechanism to determine whether the connection to an IKE peer is functional and reachable. To implement this mechanism, IKE sends a Hello packet to its peer, and the peer sends an acknowledgment in response. By default, IKE sends Hello packets every 10 seconds, and after three unacknowledged packets, IKE declares the neighbor to be dead and tears down the tunnel to the peer. Thereafter, IKE periodically sends a Hello packet to the peer, and re-establishes the tunnel when the peer comes back online.

You can change the liveness detection interval to a value from 0 through 65535, and you can change the number of retries to a value from 0 through 255.

**Note** For transport VPNs, the liveness detection interval is converted to seconds by using the following formula:

$$\text{Interval for retransmission attempt number N} = interval * 1.8^{N-1}$$

For example, if the interval is set to 10 and retries to 5, the detection interval increases as follows:

- Attempt 1: $10 * 1.8^{1-1} = 10$ seconds
- Attempt 2: $10 * 1.8^{2-1} = 18$ seconds
- Attempt 3: $10 * 1.8^{3-1} = 32.4$ seconds
- Attempt 4: $10 * 1.8^{4-1} = 58.32$ seconds
- Attempt 5: $10 * 1.8^{5-1} = 104.976$ seconds

```
vEdge(config-interface-ipsecnumber)# dead-peer-detection interval retries number
```

# Configure Other Interface Properties

For IPsec tunnel interfaces, you can configure only the following additional interface properties:

```
vEdge(config-interface-ipsec)# mtu bytes
vEdge(config-interface-ipsec)# tcp-mss-adjust bytes
```

# Disable Weak SSH Encryption Algorithms on Cisco SD-WAN Manager

*Table 5: Feature History Table*

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Disable Weak SSH Encryption Algorithms on Cisco SD-WAN Manager | Cisco vManage Release 20.9.1 | This feature allows you to disable weaker SSH algorithms on Cisco SD-WAN Manager that may not comply with certain data security standards. |

## Information About Disabling Weak SSH Encryption Algorithms on Cisco SD-WAN Manager

Cisco SD-WAN Manager provides an SSH client for communication with components in the network, including controllers and edge devices. The SSH client provides an encrypted connection for secure data transfer, based on a variety of encryption algorithms. Many organizations require stronger encryption than that provided by SHA-1, AES-128, and AES-192.

From Cisco vManage Release 20.9.1, you can disable the following weaker encryption algorithms so that an SSH client does not use these algorithms:

- SHA-1
- AES-128
- AES-192

Before disabling these encryption algorithms, ensure that Cisco vEdge devices, if any, in the network, are using a software release later than Cisco SD-WAN Release 18.4.6.

### Benefits of Disabling Weak SSH Encryption Algorithms on Cisco SD-WAN Manager

Disabling weaker SSH encryption algorithms improves the security of SSH communication, and ensures that organizations using Cisco Catalyst SD-WAN are compliant with strict security regulations.

## Disable Weak SSH Encryption Algorithms on Cisco SD-WAN Manager Using CLI

1. From the Cisco SD-WAN Manager menu, choose **Tools** > **SSH Terminal**.
2. Choose the Cisco SD-WAN Manager device on which you wish to disable weaker SSH algorithms.
3. Enter the username and password to log in to the device.
4. Enter SSH server mode.

```
vmanage# config terminal
```

```
vmanage(config)# system
vmanage(config-system)# ssh-server
```

5. Do one of the following to disable an SSH encryption algorithm:
   - Disable SHA-1:
     a. `vmanage(config-ssh-server)# no kex-algo sha1`
     b. `vmanage(config-ssh-server)# commit`

        The following warning message is displayed:

        ```
        The following warnings were generated:
        'system ssh-server kex-algo sha1': WARNING: Please ensure all your edges run code
         version > 18.4.6 which negotiates better than SHA1 with vManage. Otherwise those
         edges may become offline.
        Proceed? [yes,no] yes
        ```

     c. Ensure that any Cisco vEdge devices in the network are running Cisco SD-WAN Release 18.4.6 or later and enter **yes**.

   - Disable AES-128 and AES-192:
     a. `vmanage(config-ssh-server)# no cipher aes-128-192`
     b. `vmanage(config-ssh-server)# commit`

        The following warning message is displayed:

        ```
        The following warnings were generated:
        'system ssh-server cipher aes-128-192': WARNING: Please ensure all your edges
        run code version > 18.4.6 which negotiates better than AES-128-192 with vManage.
         Otherwise those edges may become offline.
        Proceed? [yes,no] yes
        ```

     c. Ensure that any Cisco vEdge devices in the network are running Cisco SD-WAN Release 18.4.6 or later and enter **yes**.

# Verify that Weak SSH Encryption Algorithms Are Disabled on Cisco SD-WAN Manager Using the CLI

1. From the Cisco SD-WAN Manager menu, choose **Tools** > **SSH Terminal**.
2. Select the Cisco SD-WAN Manager device you wish to verify.
3. Enter the username and password to log in to the device.
4. Run the following command:

   ```
   show running-config system ssh-server
   ```

5. Confirm that the output shows one or more of the commands that disable weaker encryption algorithms:
   - no cipher aes-128-192
   - no kex-algo sha1