

# Configuring Security Parameters

This article describes how to change security parameters for the control plane and the data plane.

## 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 vSmart controller to be a server, firewalls protect TCP servers better than UDP servers.

You configure the control plane tunnel protocol on a vSmart controller:

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

With this change, all control plane tunnels between the vSmart controller and vEdge routers and between the controller and vManage NMSs use TLS. Control plane tunnels to vBond orchestrators always use DTLS, because these connections must be handled by UDP.

In a domain with multiple vSmart controllers, when you configure TLS on one of the vSmart 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 vSmart controllers, if you have not configured TLS on them, they use TLS on the control plane tunnel only to that one vSmart controller, and they use DTLS tunnels to all the other vSmart controllers and to all their connected vEdge routers. To have all vSmart controllers use TLS, configure it on all of them.

By default, the vSmart 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 vSmart controller. For example:

```
vSmart-2# show control connections
```

						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	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 on vManage NMS

If you configure the vManage NMS 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 vManage NMS. To display information about these processes and about the number of ports that are being forwarded, use the **show control summary** command. The output shows that four vdaemon processes are running:

```
vManage# show control summary
```

INSTANCE	VBOND COUNTS	VMANAGE COUNTS	VSMART COUNTS	VEDGE 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	vIptela 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.viptela.com						
site-id	5000						
domain-id	0						
protocol	dtls						
tls-port	23456						
...							
...							
...							
number-active-wan-interfaces	1						

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

This output shows that the listening TCP port is 23456. If you are running vManage NMS 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 vManage NMS, up to a maximum of 8.

---

## Configure Data Plane Security Parameters

In the data plane, IPsec is enabled by default on all vEdge routers, and by default IPsec tunnel connections use the AH-SHA1 HMAC for authentication on the IPsec tunnels. On vEdge routers, you can change the type of authentication, and you can modify the IPsec rekeying timer and the size of the IPsec anti-replay window.

---

### Configure Allowed Authentication Types

By default, IPsec tunnel connections use AH-SHA1 HMAC and ESP HMAC-SHA1 for authentication, choosing whichever authentication method is stronger. To modify the negotiated authentication types or to disable authentication, use the following command:

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

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:

- **ah-sha1-hmac** enables AH-SHA1 HMAC and ESP HMAC-SHA1.
- **ah-no-id** enables a modified version of AH-SHA1 HMAC and ESP HMAC-SHA1. This option accommodates some non-Viptela 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 Viptela AH software ignore the ID field in the IP header so that the Viptela software can work in conjunction with these devices.
- **sha1-hmac** enables ESP HMAC-SHA1.
- **none** maps to no authentication. You can choose this option in situations where data plane authentication and integrity are not a concern.

For information about which data packet fields are affected by these authentication types, see the "Data Plane Integrity" section in [Data Plane Security Overview](#).

vEdge routers 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 vEdge router advertises AH-HMAC-SHA1, ESP HMAC-SHA1, and none, and a second vEdge router advertises ESP HMAC-SHA1 and none, the two routers negotiate to use ESP HMAC-SHA1 on the IPsec tunnel connection between them. If no common authentication



types are configured on the two vEdge peers, no IPsec tunnel is established between them.

The encryption algorithm on IPsec tunnel connections is either AES-256-GCM or AES-256-CBC. For unicast traffic, if the remote side supports AES-256-GCM, that encryption algorithm is used. Otherwise, AES-256-CBC is used. For multicast traffic, the encryption algorithm is AES-256-CBC. You cannot modify the choice made by the software.

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

---

## Change the Rekeying Timer

Before vEdge routers can exchange data traffic, they set up a secure authenticated communications channel between them. The vEdge devices use the DTLS or TLS control plane connection between them as the channel, and they use the AES-256 cipher to perform encryption. Each vEdge 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:

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

The configuration looks like this:

```
security
ipsec
  rekey seconds
!
```

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

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

```
vEdge# show ipsec local-sa
```

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

If this key is compromised, use the **request security ipsec-rekey** command to generate a new key immediately. This command increments the existing key, so in our example the SPI changes to 257:

```
vEdge# request security ipsec-rekey
vEdge# show ipsec local-sa
```

TLOC ADDRESS	TLOC COLOR	SPI	SOURCE IP	SOURCE PORT	KEY HASH
172.16.255.15	lte	257	10.1.15.15	12346	*****b93a

After the new key is generated, the vEdge router sends it immediately to all its DTLS or TLS peers, and they begin using it as soon as they receive it. Note that the old compromised SPI (256) will continue to be used for a short period of time, until it times out.

To stop using the compromised 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 key to 258. Note, however, that some packets will be dropped for a short period of time, until all the remote vEdge routers learn the new key.

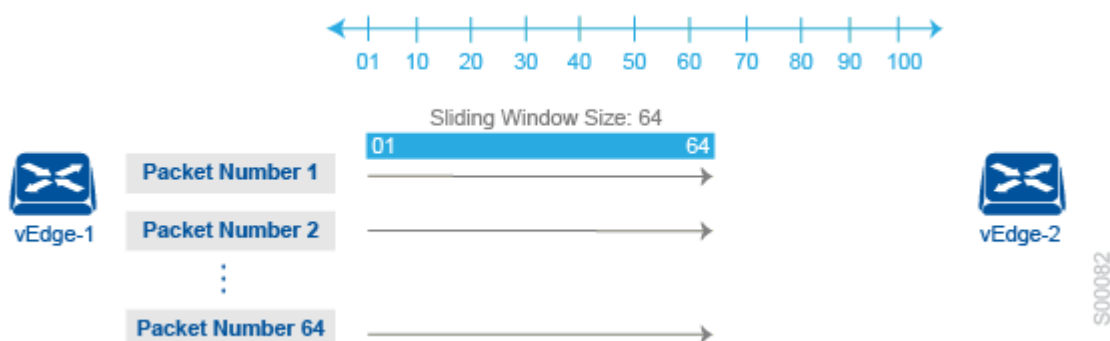
```
vEdge# request security ipsec-rekey
vEdge# request security ipsec-rekey
vEdge# ipsec local-sa
```



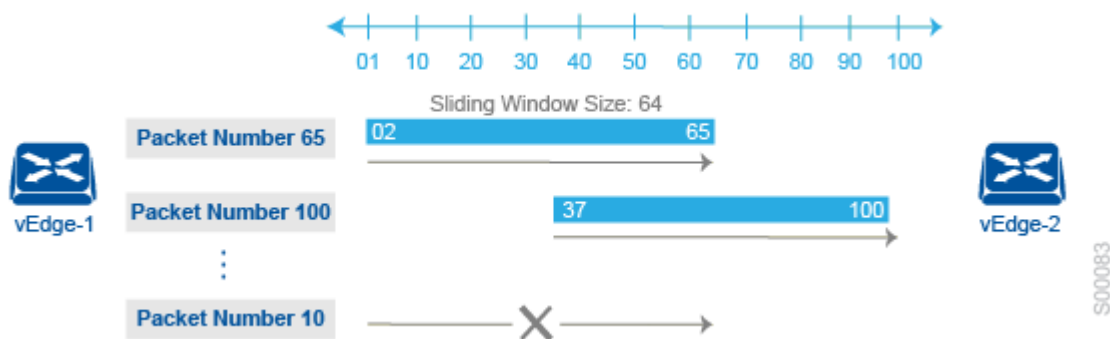
TLOC ADDRESS	TLOC COLOR	SPI	SOURCE IP	SOURCE PORT	KEY HASH
172.16.255.15	lte	258	10.1.15.15	12346	*****b93a

## 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 8192 that is a power of 2 (that is, 64, 128, 256, 512, 1024, 2048, 4096, or 8192). To modify the anti-replay window size, use the **replay-window** command, specifying the size of the window:

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

The configuration looks like this:

```
security
```



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

If QoS is configured on a vEdge 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, increase the size of the anti-replay window.

---

## Additional Information

[show control connections](#)

[show security-info](#)

[Security Overview](#)

