

Understand SD-WAN and Traditional Tunnels SPI Recover Differences

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Problem](#)

[Solution](#)

[Recovery for Traditional IPSec Tunnels](#)

[Recovery for SD-WAN Tunnels - Scenario 1](#)

[Recovery for SD-WAN Tunnels - Scenario 2](#)

Introduction

This document describes how to recover SD-WAN and Third Party Tunnels from %RECVD_PKT_INV_SPI error.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Catalyst Software-Defined Wide Area Network (SD-WAN)
- Internet Protocol Security (IPSec).
- Bidirectional Forwarding Detection (BFD).

Components Used

The information in this document is based on:

- Cisco IOS® XE Catalyst SD-WAN Edges.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Problem

The concept of a Security Association (SA) is fundamental to IPSec. An SA is a relationship between two endpoints that describes how the endpoints use security services to communicate securely.

A Security Parameter Index (SPI) is 32-bit number that is chosen to uniquely identify a particular SA for

any connected device using IPsec.

One of the most common IPsec issues is that SAs can become out of sync due to an invalid SPI value, that consequently causes an IPSEC Tunnel down status as the packets are dropped by the peer and syslog messages are received in the Router.

Third Party Tunnels:

```
Jan  8 15:00:23.723 EDT: : %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for
```

For SD-WAN Tunnels:

```
Jan 10 12:18:43.404 EDT: : %CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for
```

These logs are accompanied by drops in the Quantum Flow Processor (QFP) that belongs to the Forwarding Processor (FP).

```
<#root>
```

```
Router#
```

```
show platform hardware qfp active feature ipsec datapath drops
```

```
-----  
Drop Type Name                                     Packets  
-----  
1 IN_V4_PKT_HIT_INVALID_SA                          1  
4 IN_US_V4_PKT_SA_NOT_FOUND_SPI                    9393888 <-- sub code error  
  
19 IN_OCT_ANTI_REPLAY_FAIL                          342
```

Solution

Recovery for Traditional IPsec Tunnels

In order to recover traditional IPsec Tunnels it is necessary to manually force the renegotiation of the current SAs values relation; this is performed by clear the IPsec SAs with the EXEC mode command:

```
<#root>
```

```
Router#
```

```
clear crypto sa peer 10.20.20.1
```


Recovery for SD-WAN Tunnels - Scenario 1

The **clear crypto sa peer EXEC** command works only for traditional IPsec Tunnels due to the existence of Internet Key Exchange (IKE), which automatically negotiates the association and generates a new SPI value. However, it is not possible to use that command on an SD-WAN Tunnel. The reason for this is because in SD-WAN tunnels, IKE is not used.

Because of it, an homologous command for SD-WAN Tunnels is used:

```
<#root>
Router#
request platform software sdwan security ipsec-rekey
```

The **request platform software sdwan security ipsec-rekey** command generates a new key immediately, then the tunnel comes up. In the opposite way, the command does not affect a traditional IPsec Tunnel if it exists.

 **Note:** The **request platform software sdwan security ipsec-rekey** this command takes effect in all the existing SD-WAN Tunnels opposite to the **clear crypto sa peer** that takes effect only in the SA specified.

Recovery for SD-WAN Tunnels - Scenario 2

If mistakenly the **clear crypto sa peer** command is used to deleted one of the SD-WAN tunnels SAs, the deletion happens successfully; however, a new SPI value is not generated again, because in an SD-WAN Tunnel, OMP is the one that triggers that action not IKE. Once in this status, even whether the command **request platforms software sdwan security ipsec-rekey** is issued after the **clear crypto sa peer**, the Tunnel does not come up. The encapsulations and decapsulations of the SA remain in zero, consequently the BFD session remains in a down state.

```
Router#clear crypto sa peer 10.20.20.1
Router#show crypto ipsec sa peer 10.20.20.1
interface: Tunnel10001
Crypto map tag: Tunnel10001-vesen-head-0, local addr 10.10.10.1

protected vrf: (none)
local ident (addr/mask/prot/port): (10.10.10.1/255.255.255.255/0/12346)
remote ident (addr/mask/prot/port): (10.20.20.1/255.255.255.255/0/12366)
current_peer 10.20.20.1 port 500
PERMIT, flags={origin_is_acl,}
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
#pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
#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
```

The only recovery option after the deletion of the SA is with ANY OF THESE three EXEC commands:

```
<#root>
```

```
Router#
```

```
clear sdwan omp all
```

The **clear sdwan omp all** command flaps all BFD sessions present in the device.

```
<#root>
```

```
Router#
```

```
request platforms software sdwan port_hop <color>
```

The **clear sdwan control connections** command causes the TLOC to use the next available port number on the local color specified, which causes a flap of not only all BFD sessions of that color, but the control connections of that color as well.

```
<#root>
```

```
Router#
```

```
clear sdwan control connections
```

The last command also assists in the recovery, however the impact of it is on all control connections and BFD sessions present in the device.