Troubleshoot SDWAN cEdge IPsec Anti Replay Failures

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

SD-WAN Replay Detection Considerations

- Group Key vs. Pairwise Key
- Encoded SPI
- Multiple Sequence Number Space for QoS

Commands to Force the Newly Configured Replay Window to Take Effect

Troubleshoot Replay Drop Failures

- Troubleshoot Data Collection
- Troubleshoot Workflow

Troubleshoot Example for ASR1001-x

- Solution
- Additional Wireshark Capture Tool

Related Information

Introduction

This document describes the IPsec Anti-Replay behavior in SD-WAN IPsec for cEdges routers and how to troubleshoot Anti-Replay issues.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Software-defined Wide Area Network (SD-WAN)
- Internet Protocol Security (IPsec)

Components Used

The information in this document is based on these software and hardware versions:

- C8000V Version17.06.01
- ASR1001-X Version 17.06.03a
- vManage Version 20.7.1

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.

Background Information

IPsec authentication provides built-in anti-replay protection against old or duplicated IPsec packets with the sequence number in the ESP header checked on the receiver. Anti-replay packet drops is one of the most common data-plane issues with IPsec due to packets delivered out of order, or outside of the anti-replay window. A general troubleshoot approach for IPsec anti-replay drops can be found in Troubleshoot IPsec Anti Replay Check Failures, and the general technique applies to SD-WAN as well. However, some implementation differences exist between traditional IPsec and IPsec used in the Cisco SD-WAN solution. This article is intended to explain these differences and the approach on cEdge platforms with Cisco IOS ®XE.

SD-WAN Replay Detection Considerations

Group Key vs. Pairwise Key

Unlike traditional IPsec, where IPsec SAs are negotiated between two peers with the use of the IKE protocol, SD-WAN uses a group key concept. In this model, an SD-WAN edge device periodically generates data plane inbound SA per TLOC (Transport Locator) and sends these SAs to the SD-WAN controller, which in turn propagates the SA to the rest of the edge devices in the SD-WAN fabric.

For a more detailed description of the SD-WAN data plane operations, see SD-WAN Data Plane Security Overview.

Note: Since Cisco IOS ®XE 16.12.1b/SD-WAN 19.2, IPsec pairwise keys are supported; see IPsec Pairwise Keys Overview. With Pairwise keys, IPsec anti-replay protection works exactly like traditional IPsec. This article primarily focuses on replay check with the use of the group key model.

Encoded SPI

In the IPsec ESP header, the SPI (Security Parameter Index) is a 32-bit value that the receiver uses to identify the SA to which an inbound packet is decrypted with. In SD-WAN, this inbound SPI can be identified with **show crypto ipsec sa**:

<#root>

```
cedge-2#show crypto ipsec sa | se inbound
     inbound esp sas:


spi: 0x123(291)

        transform: esp-gcm 256 ,
        in use settings ={Transport UDP-Encaps, esn}
        conn id: 2083, flow_id: CSR:83, sibling_flags FFFFFFFF80000008, crypto map: Tunnel1-vesen-head-
        sa timing: remaining key lifetime 9410 days, 4 hours, 6 mins
        Kilobyte Volume Rekey has been disabled
        IV size: 8 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)
```

Note: Even though the inbound SPI is the same for all tunnels, the receiver has a different SA and the correspondent replay-window object associated with the SA for each peer edge device since the SA is identified by the source, destination IP address, source, destination ports 4-tuple, and the SPI number. So essentially, each peer has its own anti-replay window object.

In the actual packet sent by the peer device, notice the SPI value is different from the previous output. Here is an example from the packet-trace output with the packet copy option enabled:

<#root>

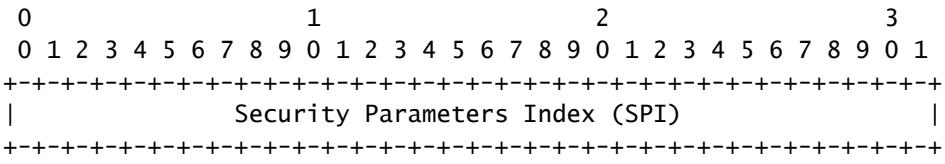
```
Packet Copy In
  45000102 0cc64000 ff111c5e ac127cd0 ac127cd1 3062303a 00eea51b

04000123

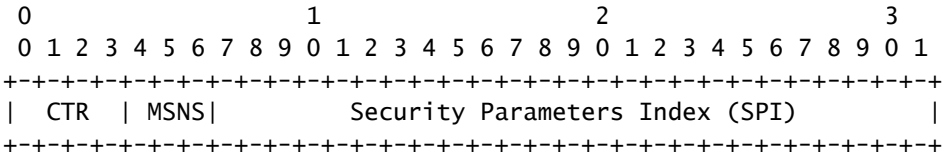
  00000138 78014444 f40d7445 3308bf7a e2c2d4a3 73f05304 546871af 8d4e6b9f
```

The actual SPI in the ESP header is **0x04000123.** The reason for this is the first bits in the SPI for SD-WAN are encoded with additional information, and only the low bits of the SPI field are allocated for the actual SPI.

Traditional IPsec:



SD-WAN:



Where:

- **CTR** (first 4 bits, bits 0-3) - Control Bits are used to indicate the specific type of control packets. For example control bit 0x80000000 is used for BFD.
- **MSNS** (next 3 bits, bits 4-6) - Multiple Sequence Number Space Index: This is used to locate the correct sequence counter in the sequence counter array to check for replay for the given packet. For SD-WAN, the 3-bit of MSNS allows for 8 different traffic classes to be mapped into their own sequence number space. This implies the effective SPI value that can be used for SA selection is the reduced low order 25 bits from the full 32-bit value of the field.

Multiple Sequence Number Space for QoS

It is common to observe IPsec replay failures in an environment where packets are delivered out of order due to QoS, for example, LLQ (Low Latency Queuing), since QoS is always run after IPsec encryption and encapsulation. The Multiple Sequence Number Space solution solves this problem with the use of multiple sequence number spaces mapped to different QoS traffic classes for a given Security Association. The different sequence number space is indexed by the MSNS bits encoded in the ESP packet SPI field as depicted. For a more detailed description, see IPsec Anti Replay Mechanism for QoS.

As noted previously, this Multiple Sequence Number implementation implies the effective SPI value can be used for SA selection is the reduced low order 25 bits. Another practical consideration when the replay window size is configured with this implementation is the configured replay-window size is for the aggregate replay window, so the effective replay window size for each Sequence Number Space is 1/8 of the aggregate.

Configuration example:

```
config-t
 Security
  IPsec
```

```
 replay-window 1024
Commit
```

Note: The effective replay window size for each Sequence Number Space is 1024/8 = 128!

Note: Since the Cisco IOS ®XE. 17.2.1, the aggregate replay window size has been increased to 8192, so each Sequence Number Space can have a maximum replay window of 8192/8 = 1024 packets.

On a cEdge device, the last sequence number received for each sequence number space can be obtained from the **show crypto ipsec sa peer x.x.x.x platform** IPsec dataplane output:

<#root>

```
cedge-2#show crypto ipsec sa peer 172.18.124.208 platform

<snip>

------------------ show platform hardware qfp active feature ipsec datapath crypto-sa 5 ----------------

 Crypto Context Handle: ea54f530
 peer sa handle: 0
 anti-replay enabled
 esn enabled
 Inbound SA
 Total SNS: 8
 Space                 highest ar number
 ----------------------------------------


 0                               39444

   1                                   0

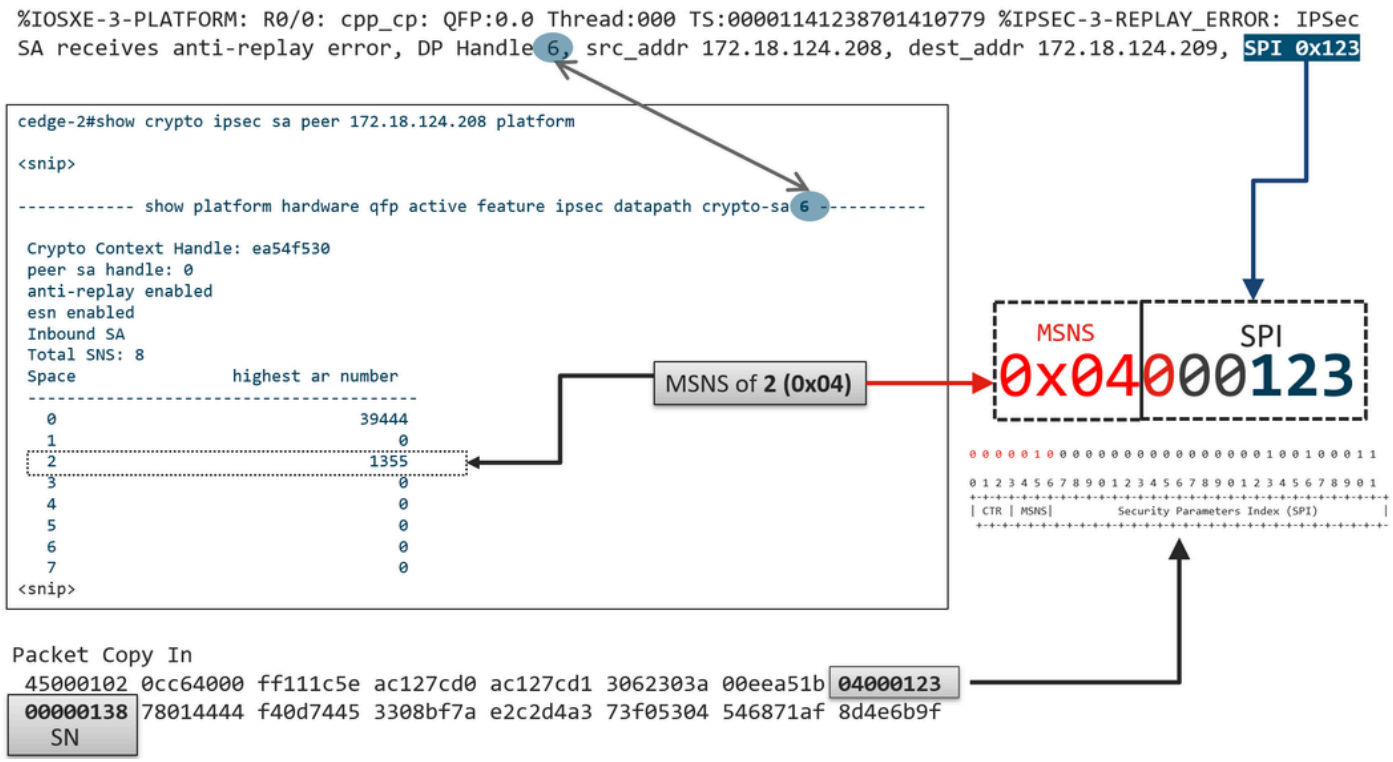

2                               1355

   3                                   0
   4                                   0
   5                                   0
   6                                   0
   7                                   0
<snip>
```

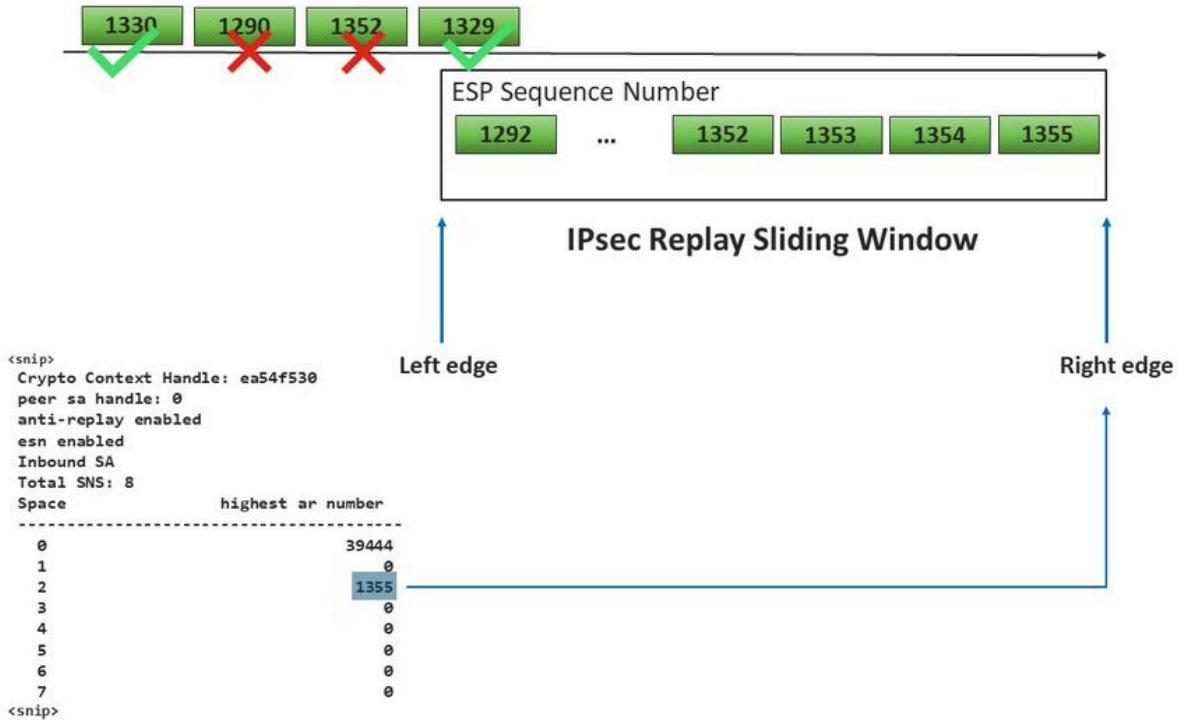
In the example, the highest anti-replay window sequence number (right edge of the anti-replay sliding window) for MSNS of 0 **(0x00)** is **39444**, and that for MSNS of **2 (0x04)** is **1335**, and these counters are used to check if the sequence number is inside of the replay window for packets in the same sequence number space.

Note: There are implementation differences betweem the ASR1k platform and the rest of the Cisco IOS ®XE routing platforms (ISR4k, ISR1k, CSR1kv, Catalyst8000 Series routers). As a result, there are some discrepancies in terms of the show commands and their output for these platforms.

It is possible to corralate the Anti-Replay errors and the show outputs to find the SPI, and the Sequence number index as shown in the image.



With the previous information obtained, the right edge (top window) and the sliding window; as shown in the image.



Commands to Force the Newly Configured Replay Window to Take Effect

Unlike the regular IPsec (non SD-WAN), the rekey command does not take effect for the anti-replay window.

```
request platform software sdwan security ipsec-rekey
```

These commands trigger the configured replay window to take effect:

Warning: Ensure you understand the potential impact of any command, they affect the control connections and data plane.

```
clear sdwan control connection
```

or

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

or

```
interface Tunnelx
 shutdown
 no shutdown
```

Troubleshoot Replay Drop Failures

Troubleshoot Data Collection

For the IPsec anti-replay drops, it is important to understand the conditions and potential triggers of the problem. At a minimum, collect the set of information for to provide the context:

- Device information for both the sender and receiver for the replay packet drops, it includes type of device, cEdge vs. vEdge, software version, and configuration.
- Problem history. How long has the deployment been in place? When did the problem start? Any recent changes to the network or traffic conditions.
- Any pattern to the replay drops, for example, is it sporadic or constant? Time of the problem and/or significant event, does it only happen during high traffic peak production hours, or only during rekey, and so on.

With the previous information collected, proceed with the troubleshoot workflow.

Troubleshoot Workflow

The general troubleshooting approach for IPsec replay issues is just like how it is performed for traditional IPsec. Take into account the per-peer SA sequence space and Multiple Sequence Number Space as explained. Then go through these steps:

Step 1. First identify the peer for the replay drop from the syslog and the drop rate. For drop statistics, always collect multiple timestamped snapshots of the output so that the drop rate can be quatified:

```
*Feb 19 21:28:25.006: %IOSXE-3-PLATFORM: R0/0: cpp_cp: QFP:0.0 Thread:000 TS:00001141238701410779 %IPSE
```

```
cedge-2#show platform hardware qfp active feature ipsec datapath drops
Load for five secs: 1%/0%; one minute: 1%; five minutes: 1%
No time source, *11:25:53.524 EDT Wed Feb 26 2020
----------------------------------------------------------------------
Drop Type  Name                                          Packets
----------------------------------------------------------------------
        4  IN_US_V4_PKT_SA_NOT_FOUND_SPI                        30
       19  IN_CD_SW_IPSEC_ANTI_REPLAY_FAIL                      41
```

Note: It is not uncommon to see occasional replay drops due to packet delivery reordering in the network, however, persistent replay drops impact the service and they can be investigated.

Step 2a. For a relatively low traffic rate, take a packet-trace with the condition set to the peer ipv4 address with the ***copy packet*** option. Examine the sequence numbers for the packet dropped against the current replay window (right edge) and sequence numbers in the adjacent packets. Confirm if they are indeed duplicate(s) or outside of the replay window.

Step 2b. For a high traffic rate with no predictable triggers, configure an EPC capture with circular buffer and EEM to stop the capture when replay errors are detected. Since EEM is currently not supported on vManage as of 19.3, this implies the cEdge would have to be in CLI mode when this troubleshooting task is performed.

Step 3. Collect the **show crypto ipsec sa peer x.x.x.x platform** output on the receiver, ideally at the same time of the packet capture or when the packet-trace is collected. This command includes the real-time dataplane replay window information for both the inbound and outbound SA.

Step 4. If the packet dropped is out of order, then take simultaneous captures from both the sender and receiver to identify if the problem is with the source or with the underlay network delivery layer.

Step 5. If the packets are dropped even though they are neither duplicate nor outside of the replay window, then it is usually indicative of a software problem on the receiver.

Troubleshoot Example for ASR1001-x

Problem description:

HW: ASR1001-X
SW: 17.06.03a

Multiple anti-replay errors receive for the session peer 10.62.33.91, therefore the BFD session constantly flaps and the traffic between these two sites is affected.

```
Jul 26 20:31:20.879: %IOSXE-3-PLATFORM: RO/O: cpp_cp: QFP:0.0 Thread:027 TS:00000093139972173042 %IPSEC
Jul 26 20:32:23.567: %IOSXE-3-PLATFORM: RO/O: cpp_cp: QFP:0.0 Thread:009 TS:00000093202660128696 %IPSEC
Jul 26 20:33:33.939: %IOSXE-3-PLATFORM: RO/O: cpp_cp: QFP:0.0 Thread:051 TS:00000093273031417384 %IPSEC
Jul 26 20:34:34.407: %IOSXE-3-PLATFORM: RO/O: cpp_cp: QFP:0.0 Thread:020 TS:00000093333499638628 %IPSEC
```

Step 1. Check Configured Anti Replay Window is 8192.

<#root>

```
cEdge#show sdwan security-info
 security-info authentication-type deprecated
 security-info rekey 86400

 security-info replay-window 8192


 security-info encryption-supported "AES_GCM_256 (and AES_256_CBC for multicast)"
 security-info fips-mode Disabled
 security-info pairwise-keying Disabled
 security-info pwk-sym-rekey Enabled
 security-info extended-ar-window Disabled
 security-info integrity-type "ip-udp-esp esp"
```

Note: The effective replay window size for each Sequence Number Space must be 8192/8= 1024 in this example.

Step2. Verify the effective replay window size for peer 10.62.33.91 to compare and confirm the configured value.

<#root>

```
show crypto ipsec sa peer 10.62.33.91 platform
<snip>
------------------ show platform hardware qfp active feature ipsec sa

22

 ------------------
<snip>
------------------ show platform software ipsec fp active encryption-processor 0 context c441ff4c -----
<snip>

       window size: 64                  <-- Effective Window Size


  window base(ESN): 0
   Multi-SNS window_top
```

```
-----------------------------------
  index: 0, win_top: 0x00000000010dc0
  index: 1, win_top: 0000000000000000
  index: 2, win_top: 0x00000000b65f00
  index: 3, win_top: 0000000000000000
  index: 4, win_top: 0000000000000000
  index: 5, win_top: 0000000000000000
  index: 6, win_top: 0000000000000000
  index: 7, win_top: 0000000000000000
traffic hard limit: 12876354284605669376
        byte count: 0
        packet count: 11378618
```

The **Window Size: 64** displayed in the output does not match what the configured replay window of **8192 (8192//8=1024)**, which means even though it was configured, the command did not take effect.

Note: The effective replay window is only displayed on the ASR platforms. To ensure the actual size for the anti-replay window is the same as the configured size, apply one of the commands in the previous section to force the configured window size to take effect.

Step 3. Configure and enable packet trace and monitor capture (optional) simultaneously for inbound traffic from session source: 10.62.33.91, destination: 10.62.63.251

```
cEdge#debug platform packet-trace packet 2048 circular fia-trace data-size 2048
cEdge#debug platform packet-trace copy packet both size 2048 L3
cEdge#debug platform condition ipv4 10.62.33.91/32 in
cEdge#debug platform condition start
```

Step 4. Collect packet trace summary:

```
cEdge#show platform packet summay
<snip>
811     Te0/0/0.972         Te0/0/1.301    FWD
812     Te0/0/0.972         Te0/0/1.301    FWD
813     Te0/0/0.972         Te0/0/1.301    FWD
814     Te0/0/0.972         Te0/0/1.301    FWD
815     Te0/0/0.972         Te0/0/1.301    FWD
816     Te0/0/0.972         Te0/0/0.972    DROP 56 (IpsecInput)
817     Te0/0/0.972         Te0/0/0.972    DROP 56 (IpsecInput)
818     Te0/0/0.972         Te0/0/0.972    DROP 56 (IpsecInput)
819     Te0/0/0.972         Te0/0/0.972    DROP 56 (IpsecInput)
<snip>
837     Te0/0/0.972         Te0/0/1.301    FWD
838     Te0/0/0.972         Te0/0/1.301    FWD
```

Step 5. Expand some dropped (IpsecInput) packets captured.

(IpsecInput) Packet drops:

<#root>

```
cEdge#sh platform pack pack 816
Packet: 816 CBUG ID: 973582
Summary
  Input : TenGigabitEthernet0/0/0.972
  Output : TenGigabitEthernet0/0/0.972
  State : DROP 56

(IpsecInput)



  Timestamp
    Start : 97495234494754 ns (07/26/2022 21:43:56.25110 UTC)
    Stop : 97495234610186 ns (07/26/2022 21:43:56.25225 UTC)
Path Trace
  Feature: IPV4(Input)
  Input : TenGigabitEthernet0/0/0.972
  Output : <unknown>
  Source : 10.62.33.91
  Destination : 10.62.63.251
  Protocol : 17 (UDP)
    SrcPort : 12367
    DstPort : 12347
<snip>

Packet Copy In
45000072 ab314000 fd115c77 0a3e215b 0a3e3ffb 304f303b 005e0000

04000106




00b6dfed


 00000000 d0a60d5b 6161b06e 453d0e3d 5ab694ce 5311bbb6 640ecd68
7ceb2726 80e39efd 70e5549e 57b24820 fb963be5 76d01ff8 273559b0 32382ab4
c601d886 da1b3b94 7a2826e2 ead8f308 c464

817 DROP:
-------------------------------
Packet: 817
<snip>
Packet Copy In
45000072 ab314000 fd115c77 0a3e215b 0a3e3ffb 304f303b 005e0000

04000106




00b6dfec


 00000000 cc72d5dd ef73fe25 2440bed6 31378b78 3c506ee5 98e3dba4
bc9e6aa0 50ea98f6 7dee25c8 c1579ce0 1212290c 650f5947 57b9bc04 97c7996c
```

```
d4dbf3e6 25b33684 a7129b67 141a5e73 8736
```

SD-WAN uses UDP encapsulated ESP:

- The UDP header is 304f303b 00770000,
- The next is SPI (**04000106**)
- Therefore **00b6e00d** is the secuence number (SN).
- The MSNS index is **2** (**x04**00106) due to 32-bit SPI (**0 0 0 0 0 1 0** 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1.)

Step 6. Verify the MSNS index

<#root>

```
show crypto ipsec sa peer 10.62.33.91 platform
<snip>
------------------- show platform hardware qfp active feature ipsec sa

22


 -------------------
<snip>
------------------- show platform software ipsec fp active encryption-processor 0 context c441ff4c -----
<snip>

       window size: 64


  window base(ESN): 0
   Multi-SNS window_top
-----------------------------------
  index: 0, win_top: 0x00000000010dc0
  index: 1, win_top: 0000000000000000

  index: 2, win_top: 0x00000000b65f00


  index: 3, win_top: 0000000000000000
  index: 4, win_top: 0000000000000000
  index: 5, win_top: 0000000000000000
  index: 6, win_top: 0000000000000000
  index: 7, win_top: 0000000000000000
traffic hard limit: 12876354284605669376
        byte count: 0
        packet count: 11378618
```

The highest anti-replay window sequence number (right edge of the anti-replay sliding window) for MSNS of 2 (0x04) is **0b65f00.**

Step 7. Expand some forwarded (FWD) captured packets.

Forwarded packets:

<#root>

```
Packet: 838
<snip>
Packet Copy In
 4564008e ab044000 fd115c24 0a3e215b 0a3e3ffb 304f303b 007a0000

04000106



 00b6e015


 00000000 088bbd6a f4e4b35f b131143f ef1f91eb 659149f7 dbe6b025
 be7fbfd0 5fad1c71 014321f1 3e0d38f2 cc8d0e5f 1494e4fa 097c7723 dfc7ceef
 4a14f444 abcc1777 0bb9337f cd70c1da 01fc5262 848b657c 3a834680 b07b7092
 81f07310 4eacd656 ed36894a e468
```

Packet: 837

<#root>

```
Packet: 837
<snip>
Packet Copy In
4564008e ab044000 fd115c24 0a3e215b 0a3e3ffb 304f303b 007a0000 04000106

00b6e014


 00000000 76b2a256 8e835507 13d14430 ae16d62c c152cdfd 2657c20c
01d7ce1d b3dfa451 a2cbf6e9 32f267f9 e10e9dec 395a0f9e 38589adb aad8dfb8
a3b72c8d a96f2dce 2a1557ab 67959b6e 94bbbb0a cfc4fc9e 391888da af0e492c
80bebb0e 9d7365a4 153117a6 4089
```

Step 8. Collect and obtain the sequence number information from multiple packets forwarded (FWD) before, after and the drops.

```
FWD:
839 PKT: 00b6e003 FWD
838 PKT: 00b6e001 FWD
837 PKT: 00b6e000 FWD
815 PKT: 00b6e044 FWD
814 PKT: 00b6dfe8 FWD
813 PKT: 00b6e00d FWD

DROP:
```

```
816 PKT: 00b6dfed DROP
817 PKT: 00b6dfec DROP
818 PKT: 00b6dfeb DROP
819 PKT: 00b6dfe9 DROP
820 PKT: 00b6dfea DROP
```

Step 9. Convert the sequence number from hex to decimal and reorder them to simplify calculation:

<#root>

```
REORDERED:
813 PKT: 00b6e00d FWD --- Decimal: 11984909
814 PKT: 00b6dfe8 FWD --- Decimal: 11984872
```

815 PKT: 00b6e044 FWD --- Decimal: 11984964

******* Highest Value**

```
816 PKT: 00b6dfed DROP--- Decimal: 11984877
817 PKT: 00b6dfec DROP--- Decimal: 11984876
818 PKT: 00b6dfeb DROP--- Decimal: 11984875
819 PKT: 00b6dfe9 DROP--- Decimal: 11984873
820 PKT: 00b6dfea DROP--- Decimal: 11984874
<snip>
837 PKT: 00b6e014 FWD --- Decimal: 11984916
838 PKT: 00b6e015 FWD --- Decimal: 11984917
839 PKT: 00b6e016 FWD --- Decimal: 11984918
```

Note: If the sequence number is greater than the highest sequence number in the window, the packet has its integrity checked. If the packet passes the integrity verification check, the sliding window is then moved to the right.

Step 10. Calculate the delta between the highest sequence number and the received sequence number for each packet:

<#root>

```
Difference:
```

815 PKT: Decimal:

11984964

******* Highest Value**

```
--------------------------------------
    815(Highest)  -  X PKT  = Diff
--------------------------------------
816 PKT:
```

11984964

-

```
11984877 = 87 DROP
817 PKT:
```

11984964

-

```
11984876 = 88 DROP
818 PKT:
```

11984964

-

```
11984875 = 89 DROP
819 PKT:
```

11984964

-

```
11984873 = 91 DROP
820 PKT:
```

11984964

-

```
11984874 = 90 DROP
<snip>
837 PKT:
```

11984964

-

11984916 = 48

FWD

838 PKT:

11984964

-

11984917 = 47

FWD

839 PKT:

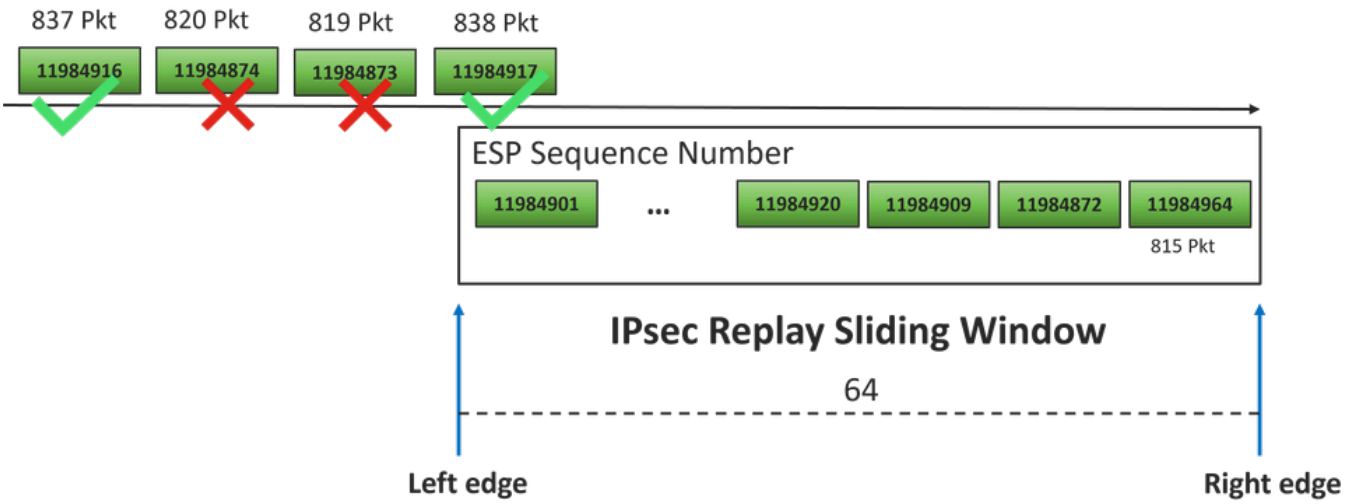
11984964

-

11984918 = 45

FWD

For this example, it is possible to visualize the sliding window with the window size 64 and right edge 11984964 as shown in the image:



The received sequence number for drop packets is way ahead of the right edge of the replay window for that sequence space.

Solution

Since the window size is still in the previous value 64, as seen in step 2, one of the commands in the section **Commands to force the newly configured replay window to take effect** must be applied for the increased 1024 window size to take effect.

Additional Wireshark Capture Tool

Another useful tool to help correlate the ESP SPI and Sequence number is Wireshark.

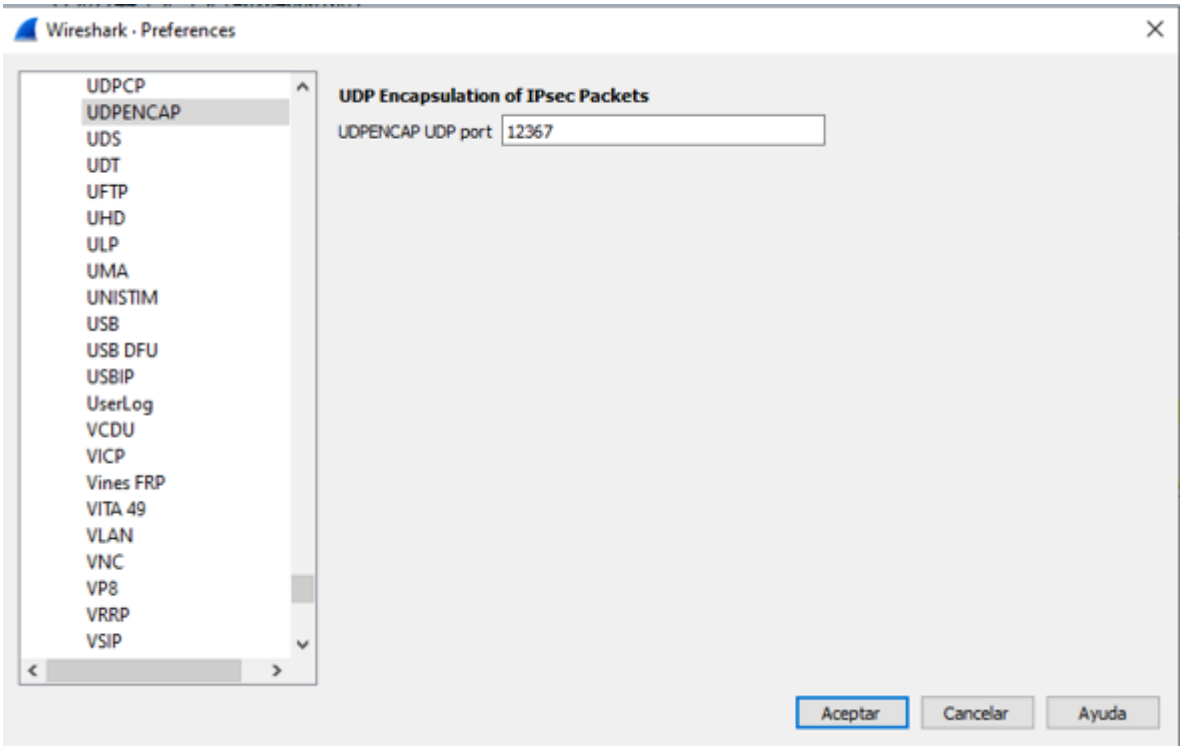
Note: It is important to collect the packet capture when the issue occurs and if it is possible at the same time the fia trace is collected as described previously

---

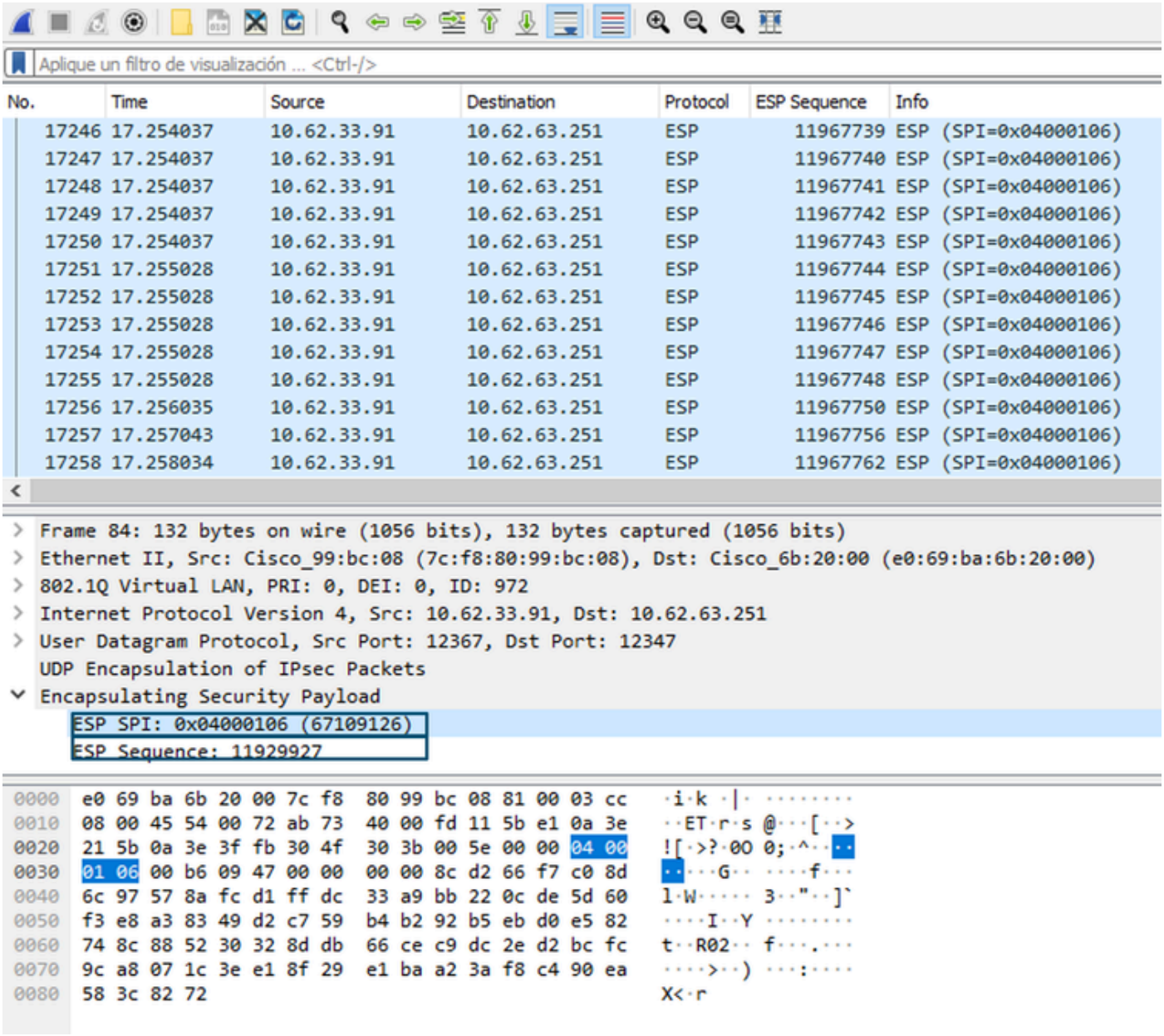
Configure the packet capture for inbound direction and export it to the pcap file.

```
monitor caputure CAP match ipv4 host 10.62.33.91 host 10.62.63.251 buffer size 20 inter TenGigabitEther
monitor caputure CAP star
monitor caputure CAP stop
monitor caputure CAP export bootflash:Anti-replay.pca
```

When pcap capture is opened in Wireshark, to see the ESP SPI and sequence number, expand one packet by right clicking and select **protocol preferences**. Search for **UDPENCAP** and change the default port to SD-WAN port (source port) as shown in the image:



After UDPENCAP is in place with the correct port, the ESP information now displays, as shown in the image:



Related Information

- IPsec Anti-Replay Check Failures TechZone Article
- IPsec Anti-Replay Window Expanding and Disabling
- Cisco Technical Support & Downloads