Troubleshoot IPsec Tunnels and Common Control-Plane Issues with Packet Captures

Contents

Introduction

This document describes how packet captures, other tools, help with control-plane issues when site-to-site VPN on Cisco IOS® XE routers is negotiated.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Basic knowledge of Cisco IOS® CLI configuration.
- Fundamental knowledge of IKEv2 and IPsec.

Components Used

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

- CSR1000V - Cisco IOS XE Software running version 16.12.0.

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

Packet captures are a powerful tool to help you verify whether packets are being sent/received between VPN peer devices. They also confirm if the behavior seen with IPsec debugs aligns to the output collected on the captures since the debugs are a logical interpretation, and the capture represents the physical interaction between the peers. Because of that, you could confirm or discard connectivity issues.

Useful Tools

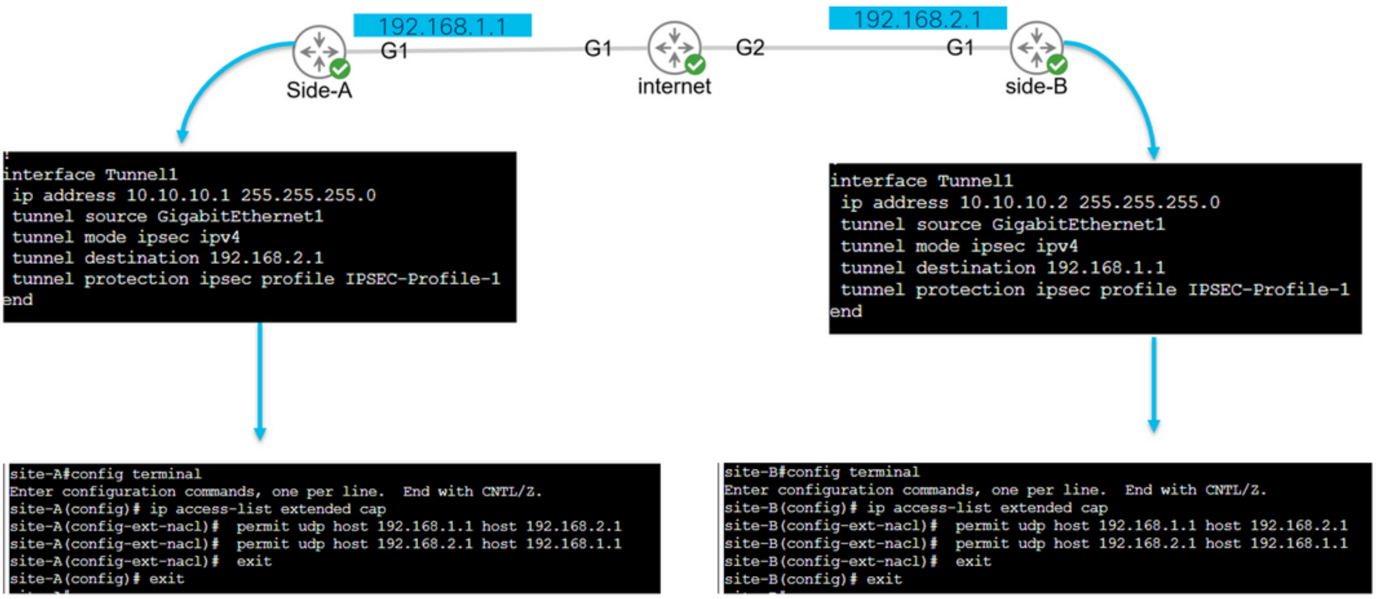
There are useful tools that help you configure the captures, extract the output, and analyze it further. Some of them are:

- Wireshark: This is a well-known and used open-source packet analyzer.
- Monitor captures: Cisco IOS XE feature on routers that help you collect captures and provide you a light output of what the traffic flow looks like, protocol collected, and its timestamps.

How to Configure Captures on IOS XE Router

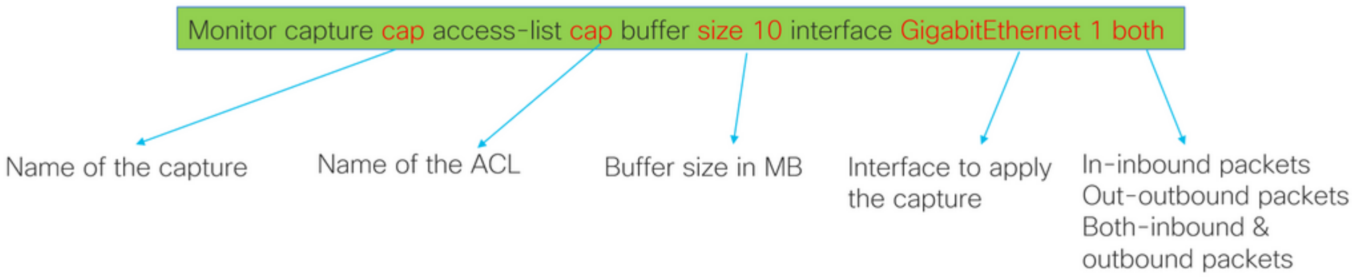
A capture uses an extended access-list (ACL) that defines the type of traffic to be collected, and the source, and destination addresses of the VPN peers or segments of the interesting traffic. A tunnel negotiation uses the UDP port 500 and port 4500 if NAT-T is enabled along the path. Once the negotiation completes and the tunnel is established, the interesting traffic uses IP protocol 50 (ESP) or UDP 4500 if NAT-T is enabled.

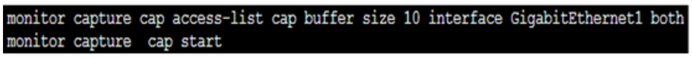
In order to troubleshoot control-plane related issues, VPN peers IP addresses must be used to capture how the tunnel is negotiated.



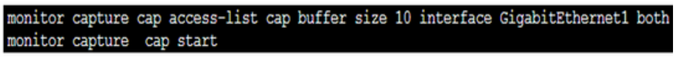
```
config terminal
ip access-list extended <ACL name>
permit udp host <local address> host <peer address>
permit udp host <peer address> host <source address>
exit
exit
```

The configured ACL is used to narrow the captured traffic, and it is placed on the interface used to negotiate the tunnel.

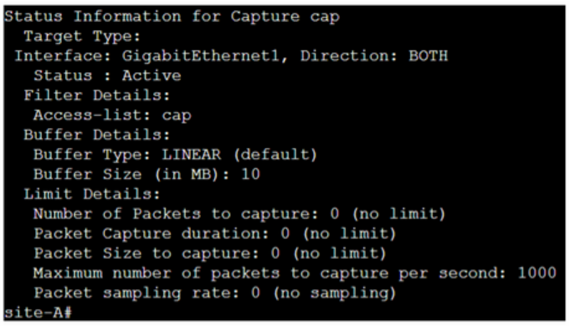




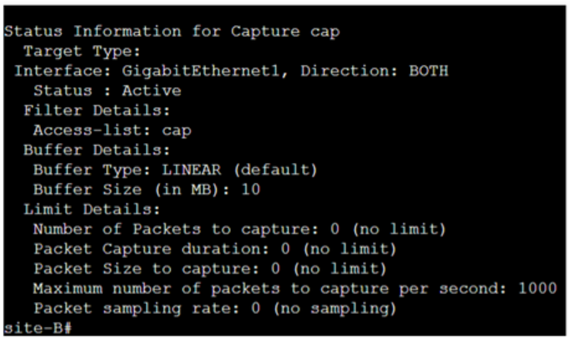
```
monitor capture cap access-list cap buffer size 10 interface GigabitEthernet1 both
monitor capture  cap start
```



```
monitor capture cap access-list cap buffer size 10 interface GigabitEthernet1 both
monitor capture  cap start
```



```
Status Information for Capture cap
  Target Type:
 Interface: GigabitEthernet1, Direction: BOTH
   Status : Active
  Filter Details:
   Access-list: cap
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 10
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
site-A#
```



```
Status Information for Capture cap
  Target Type:
 Interface: GigabitEthernet1, Direction: BOTH
   Status : Active
  Filter Details:
   Access-list: cap
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 10
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
site-B#
```

```
monitor capture <capture name> access-list <ACL name> buffer size <custom buffer size in MB> interface
```

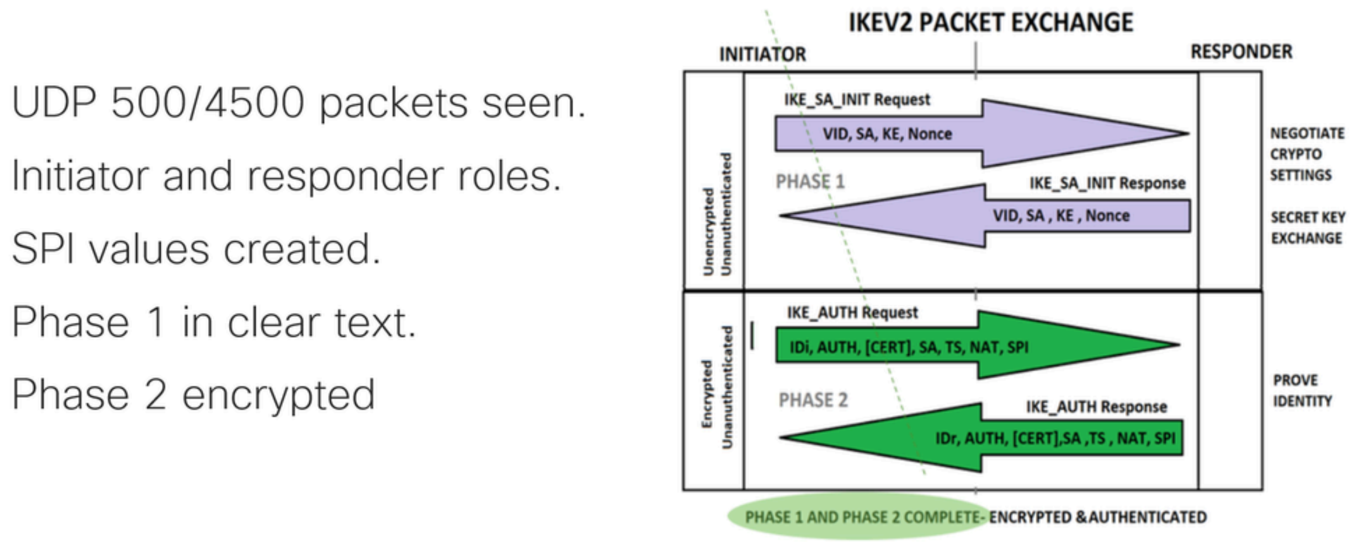
Once the capture is configured, it can be manipulated to stop it, clear it, or extract the traffic collected with the next commands:

- **Check the general capture info:** show monitor capture
- **Start/stop the capture:** monitor capture cap start/stop
- **Verify the capture is collecting packets**: show monitor capture cap buffer

- **See a brief output of the traffic**: show monitor capture cap buffer brief
- **Clear the capture:** monitor capture cap clear
- **Extract the capture output:**
    - monitor cap cap buff dump
    - monitor capture cap export bootflash:capture.pcap

Analyze the Tunnel Establishment with Packet Captures

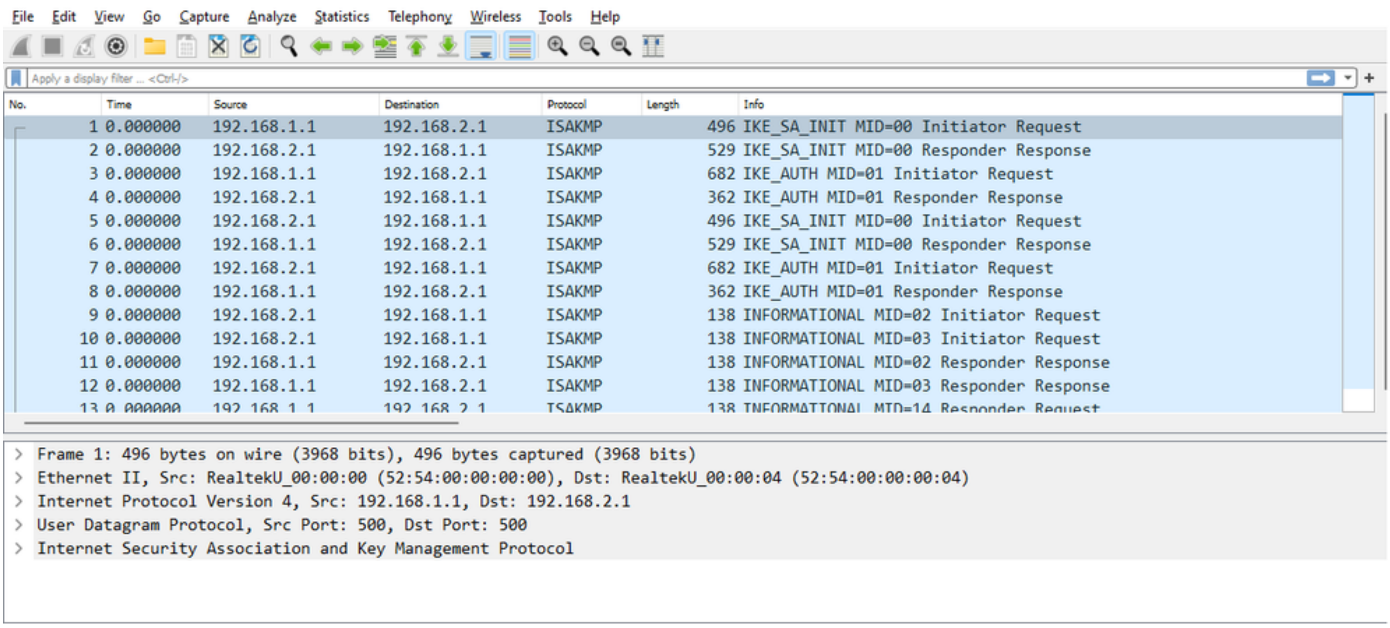
As mentioned earlier, to negotiate the IPSec tunnel, packets are sent over UDP with port 500 and port 4500 if NAT-T is enabled. With captures, more information can be seen from those packets such as the phase that is being negotiated (phase 1 or phase 2), the role of each device (initiator or responder), or the SPI values that were just created.



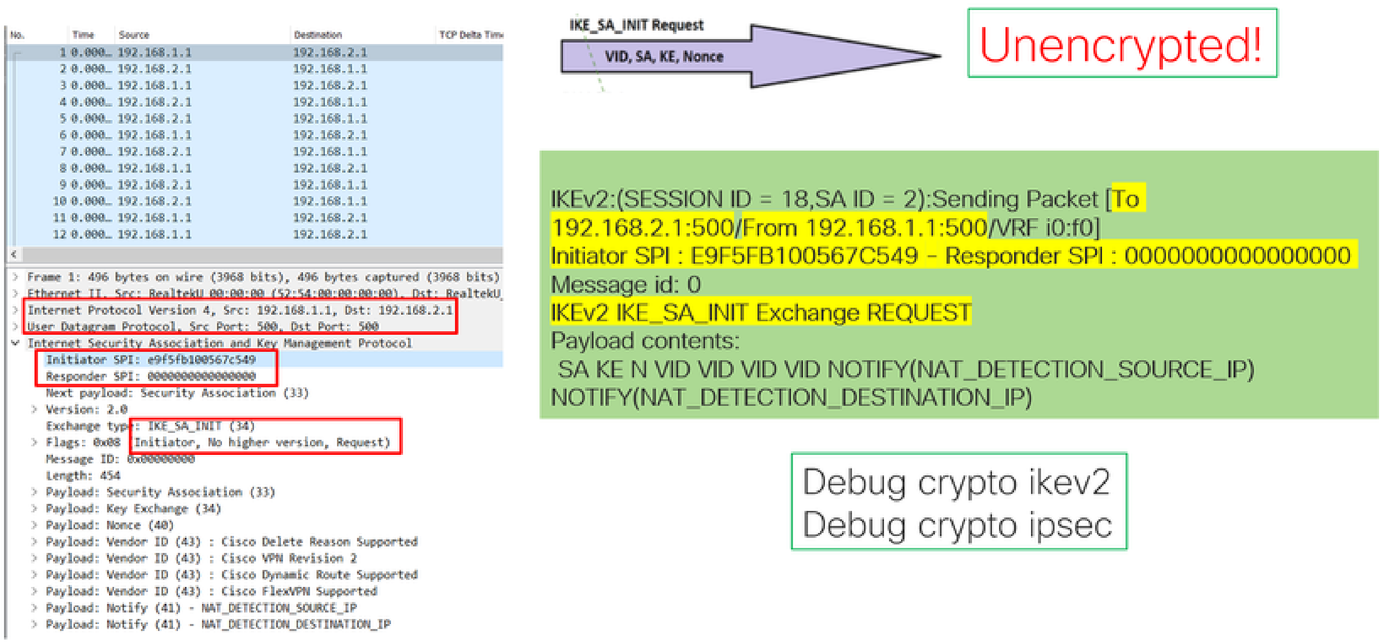
Showing the brief output of the capture from the router, the interaction between the peers is seen, sending UDP packets.

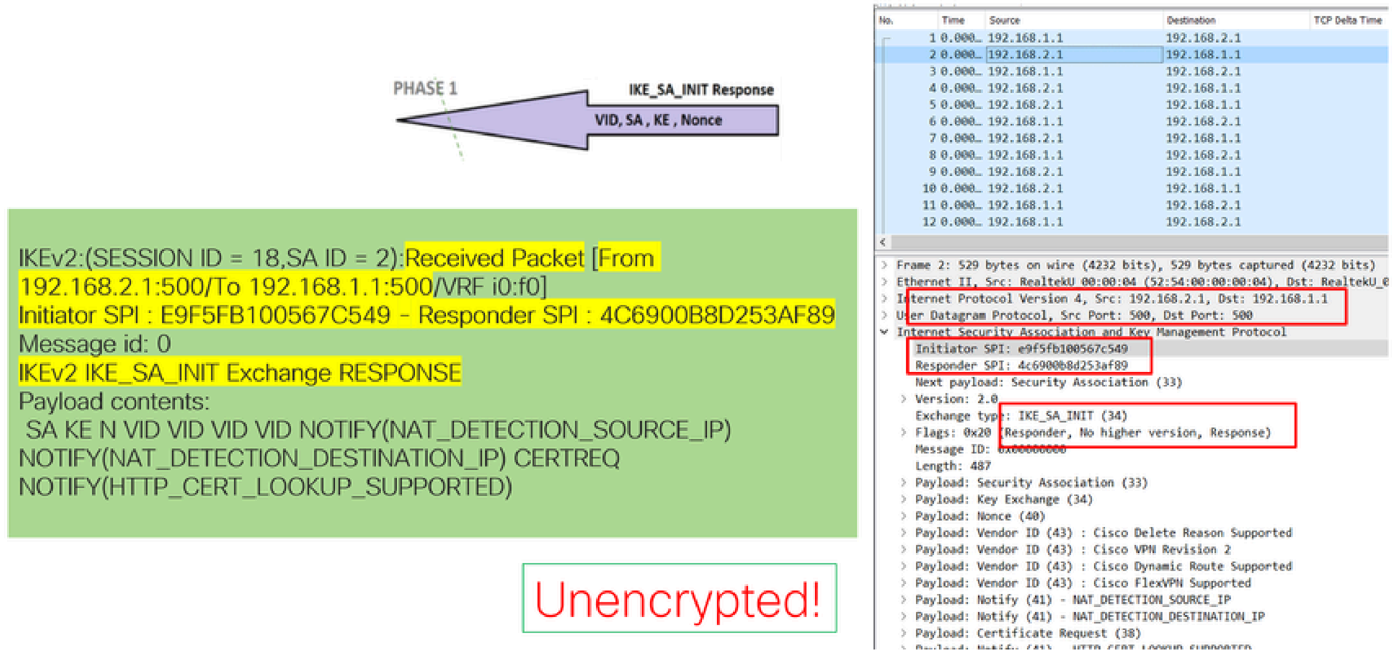
```
site-A#show monitor cap cap buffer brief
 -----------------------------------------------------------------------------
 #   size   timestamp     source             destination      dscp    protocol
 -----------------------------------------------------------------------------
   0   496    0.000000   192.168.1.1      ->  192.168.2.1      48  CS6   UDP
   1   529    0.011992   192.168.2.1      ->  192.168.1.1      48  CS6   UDP
   2   682    0.026991   192.168.1.1      ->  192.168.2.1      48  CS6   UDP
   3   362    0.035993   192.168.2.1      ->  192.168.1.1      48  CS6   UDP
   4   496    0.579016   192.168.2.1      ->  192.168.1.1      48  CS6   UDP
   5   529    0.593023   192.168.1.1      ->  192.168.2.1      48  CS6   UDP
   6   682    0.610020   192.168.2.1      ->  192.168.1.1      48  CS6   UDP
   7   362    0.616017   192.168.1.1      ->  192.168.2.1      48  CS6   UDP
   8   138    0.638019   192.168.2.1      ->  192.168.1.1      48  CS6   UDP
   9   138    0.638019   192.168.2.1      ->  192.168.1.1      48  CS6   UDP
  10   138    0.641009   192.168.1.1      ->  192.168.2.1      48  CS6   UDP
  11   138    0.655016   192.168.1.1      ->  192.168.2.1      48  CS6   UDP
```

After extracting the dump and exporting the pcap file from the router, more information from the packets is visible using wireshark.

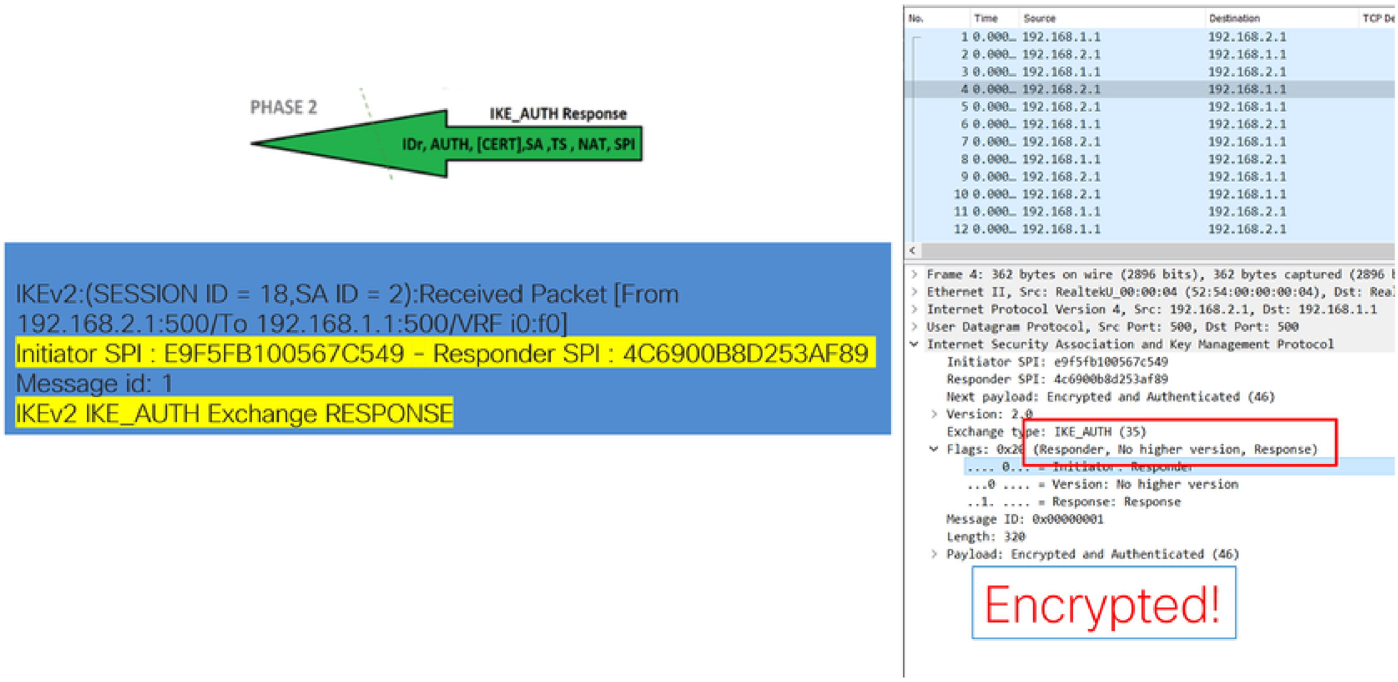


On the Internet Protocol section of the first IKE_SA_INIT Exchange packet sent, the source and destination addresses of the UDP packet are located. On the User Datagram Protocol section, the ports used and the Internet Security Association and Key Management Protocol section the version of the protocol, the type of message being exchanged, and the role of the device, as well as SPI created are seen. When collecting IKEv2 debugs, the same information is presented within the debug logs.

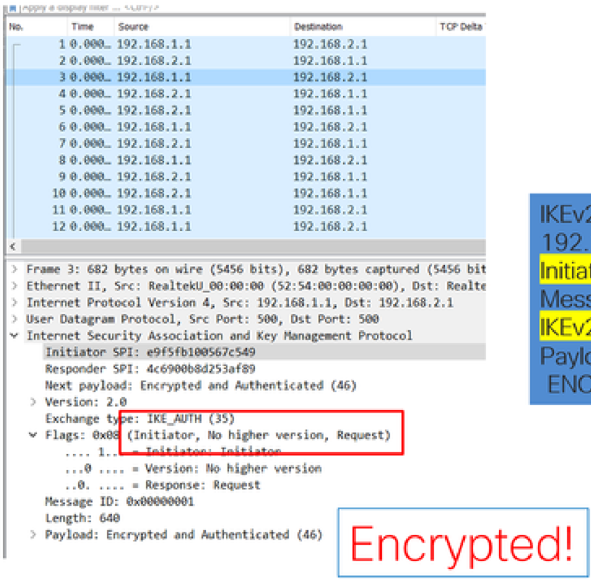




When the IKE_AUTH Exchange negotiation takes place, the payload is encrypted but, some information about the negotiation is visible, such as the SPI previously created, and the type of transaction being made.

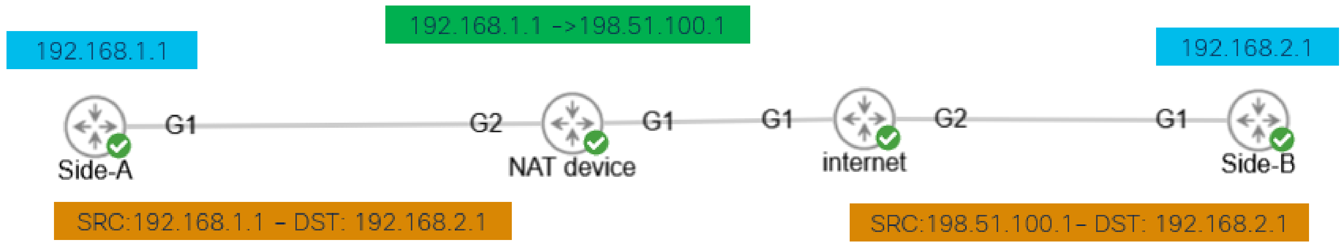


Once the last IKE_AUTH Exchange packet is received, the tunnel negotiation is completed.



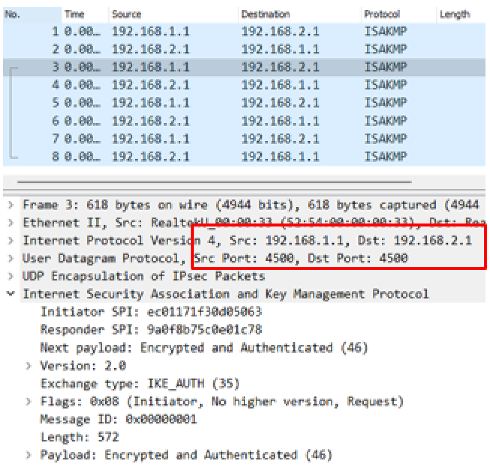
IKEv2:(SESSION ID = 18,SA ID = 2):Sending Packet [To 192.168.2.1:500/From 192.168.1.1:500/VRF i0:f0]
Initiator SPI : E9F5FB100567C549 - Responder SPI : 4C6900B8D253AF89
Message id: 1
IKEv2 IKE_AUTH Exchange REQUEST
Payload contents:
ENCR

Transaction When NAT is in Between



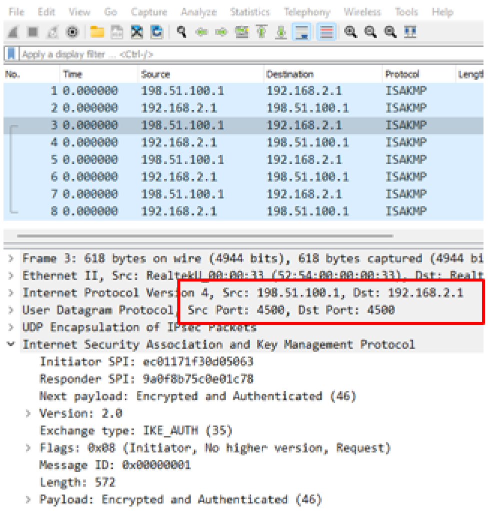
Nat-transversal is another feature that can be seen when the tunnel negotiation takes place. If an intermediate device is natting one or both addresses used for the tunnel, the devices change the UDP port from 500 to 4500 when phase 2 (IKE_AUTH Exchange) is negotiated.

Capture taken on Side-A:



IKEv2:(SESSION ID = 10,SA ID = 1):Received Packet [From 192.168.1.1:4500/To 192.168.2.1:4500/VRF i0:f0]
Initiator SPI : EC01171F30D05063 - Responder SPI : 9A0F8B75C0E01C78
Message id: 1
IKEv2 IKE_AUTH Exchange REQUEST
....
IKEv2:(SESSION ID = 10,SA ID = 1):Stopping timer to wait for auth message
IKEv2:(SESSION ID = 10,SA ID = 1):Checking NAT discovery
IKEv2:(SESSION ID = 10,SA ID = 1):NAT INSIDE found
IKEv2:(SESSION ID = 10,SA ID = 1):NAT detected float to init port 4500, resp port 4500

Capture taken on Side-B:



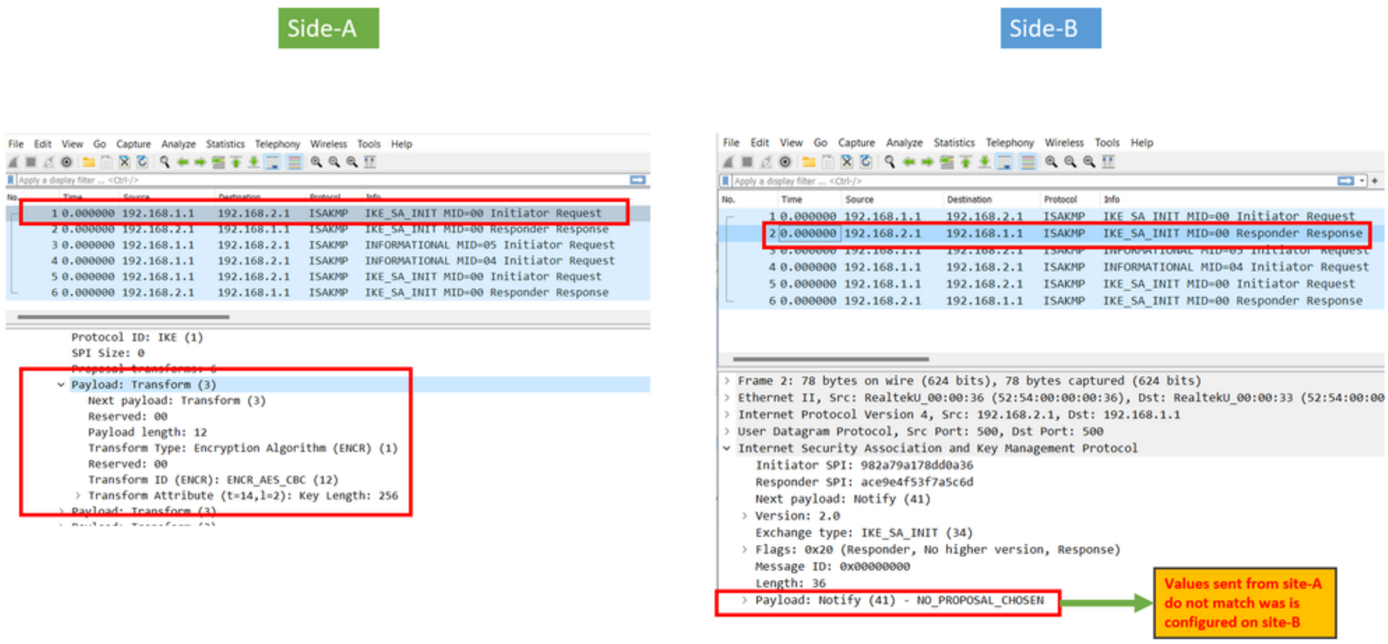
IKEv2:(SESSION ID = 11,SA ID = 1):Sending Packet [To
192.168.2.1:4500/From 198.51.100.1:4500/VRF i0:f0]
Initiator SPI : EC01171F30D05063 - Responder SPI : 9A0F8B75C0E01C78
Message id: 1
IKEv2 IKE_AUTH Exchange REQUEST
Payload contents:

Common Control-Plane Issues

There could be local or external factors that affect the tunnel negotiation and can be identified with captures as well. The next scenarios are the most common.

Configuration Mismatch

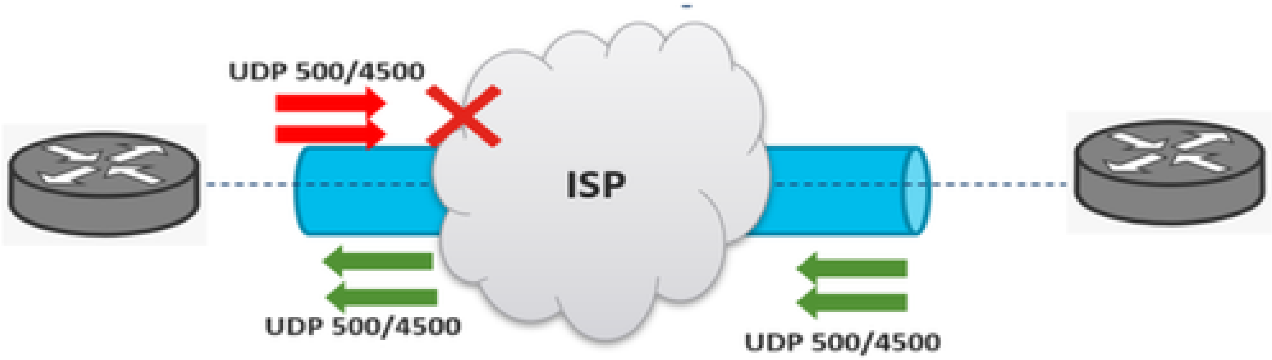
This scenario can resolved by looking at each device phase 1 and phase 2 configuration. However, there could be scenarios in which there is no access to the remote end. Captures help out by identifying which device sends a NO_PROPOSAL_CHOSEN within the packets either on phase 1 or 2. That response indicates something can be wrong with the configuration and which phase needs to be adjusted.

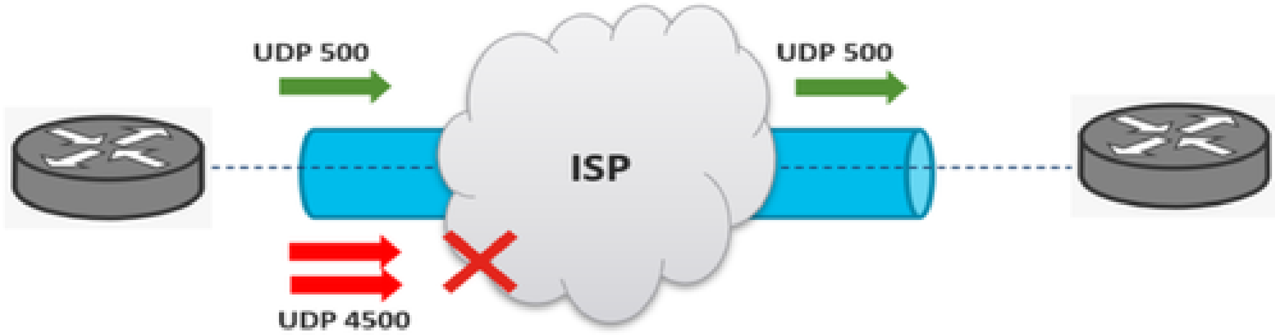


Retransmissions

An IPSec tunnel negotiation can fail due to the negotiation packets being dropped along the path between the end devices. The packets dropped can be phase 1 or phase 2 packets. When this is the case, the device that expects a response packet retransmits the last packet, and if there is no response after 5 attempts, the tunnel is concluded and restarted from the beginning.

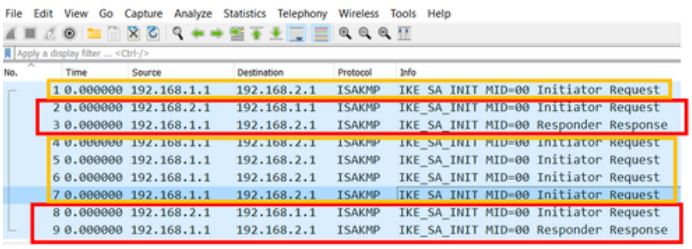
Captures on each side of the tunnel help by identifying what could possibly block the traffic and in which direction it is affected.



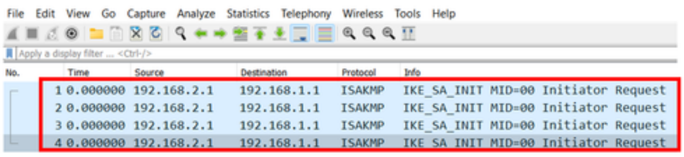


Side-A

Side-B



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.1	192.168.2.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
2	0.000000	192.168.2.1	192.168.1.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
3	0.000000	192.168.1.1	192.168.2.1	ISAKMP	IKE_SA_INIT MID=00 Responder Response
4	0.000000	192.168.1.1	192.168.2.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
5	0.000000	192.168.1.1	192.168.2.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
6	0.000000	192.168.1.1	192.168.2.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
7	0.000000	192.168.1.1	192.168.2.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
8	0.000000	192.168.2.1	192.168.1.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
9	0.000000	192.168.1.1	192.168.2.1	ISAKMP	IKE_SA_INIT MID=00 Responder Response



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.1	192.168.1.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
2	0.000000	192.168.2.1	192.168.1.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
3	0.000000	192.168.2.1	192.168.1.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request
4	0.000000	192.168.2.1	192.168.1.1	ISAKMP	IKE_SA_INIT MID=00 Initiator Request

A device or service in between is blocking UDP packets that come from side-A