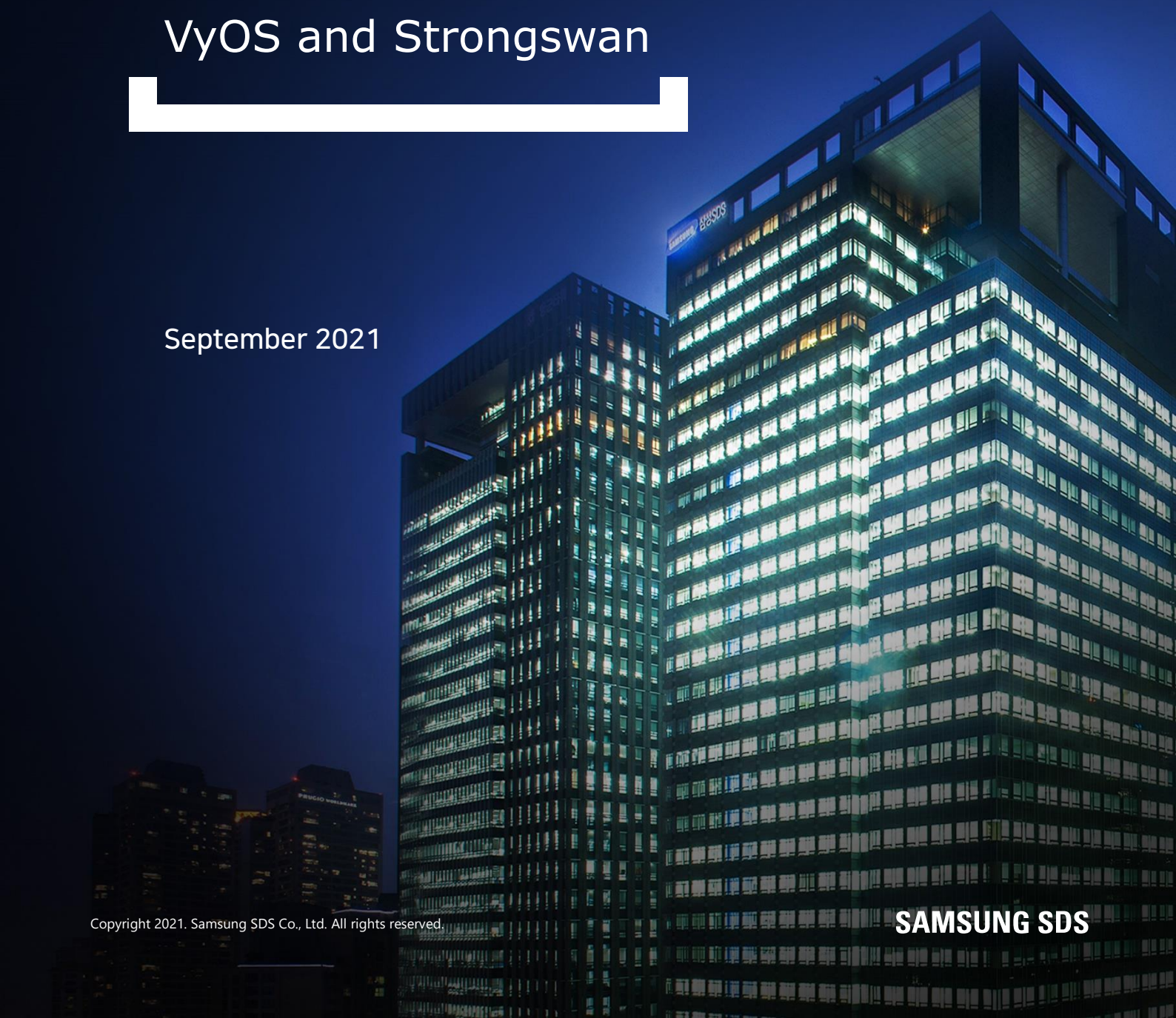
Technical Guide

Interworking with open source IPSec VPNs

VyOS and Strongswan



September 2021

SAMSUNG SDS

Contents

1. IPSec VPN overview

The purposes of this document are to promote understanding on the basic concept of Internet Protocol Security (IPSec) Virtual Private Network (VPN) and to look at the interworking of SDS Cloud's IPSec VPN with major open source VPNs.

In general, interoprability is crucial to IPSec VPN and it is necessary to verify the compatibility of interconnected solutions, in order to prevent issues such as not being able to connect normally due to differences in simple setting values or only one-way traffic flows.

This document will explain the main IPSec parameters in a network topology situation where the other VPN device is located behind NAT (Network Address Translation) for site-to-site IPSec VPN connection. Then, SDS Cloud VPN service setting and leading open souce VPNs of Strongswan and VyOS configuration and verificaion measures will also be provided.

1.1 Concept

IPSec is a set of network protocols required to authenticate one another and encrypt data packets to provide encrypted communications between two peers to be connected. Peers can be host-to-host, host-to-network, or network-to-network and a VPN between remote locations can be configured using IPSec.

IPSec uses Authentication Header (AH) and Encapsulation Security Payload (ESP) protocols to provide authentication, integrity, and confidentiality, It provides authentication and key exchange through the Internet Security Association and Key Management Protocol (ISAKMP) framework.

1. **Authentication Header (AH)**: As an IP Extension Header, source authentication and data integrity are guaranteed but encryption is not provided.
2. **Encapsulation Security Payload (ESP)**: A method of encapsulating the existing IP packet, providing confidentiality in addition to source authentication and data integrity.
3. **Security Association (SA)**: For encrypted communication, an encryption algorithm for securing data confidentiality, a hash algorithm for integrity, lifetime, key exchange method, etc. are determined upon agreement.

1.2 Modes of operation

There are two modes of operations: the transport mode protects only the payload excluding the IP Header, whereas the tunneling mode protects the entire IP packet.

The transport mode is mainly used in host-to-host cases, while network-to-network or network-to-host cases often use the tunneling mode. The tunneling mode is particularly useful in general enterprise network environments as it supports NAT traversal.

	Transport Mode	Tunneling Mode
AH	Original IP Header + AH + Original IP Payload	New IP Header + AH + Original IP Packet
ESP	Original IP Header + ESP Header + Original Payload + ESP Trailer & Authentication	New IP Header + ESP Header + Original IP Packet + ESP Trailer + ESP Authentication

1.3 Comparison of IKEv1 and IKEv2

IKE (Internet Key Exchange) is a protocol used for key exchange, including IKEv1 and IKEv2. Between the two, IKEv2 provides more authentication methods than IKEv1, considers mobile environments, defines NAT traversal within the specification, and offers more efficient message exchanges.

The features of each protocol are as follows:

	IKEv1	IKEv2
Feature	Exchange of 6 messages in Phase 1 (Main mode), and exchange of 3 messages in Phase 2 (Quick mode) Multiple subnets per TS (Traffic Selector) are not supported when configuring a policy based VPN	Total 4 message exchanges: IKE_SA_INIT Req/Res, IKE_AUTH Req/Res Support various authentication methods with EAP Mobile device support such as MOBIKE NAT-Traversal supported

1.4 Comparison of route-based VPN and policy-based VPN

A route-based VPN creates an IPSec tunnel interface (usually using a Virtual Tunnel Interface, or VTI), and transmits the traffic to be sent to a peer VPN to the VTI through the tunnel.

On the other hand, a policy-based VPN used the method that creates and implements the policy for remote/local subnets that need to be communicated through an IPSec tunnel using the TS (Traffic Selector).

The features of each protocol are as follows:

	Route-based VPN	Policy-based VPN
Scalability	The number of VPN tunnels is limited by the tunnel interface	The number of VPN tunnels is limited by the number of policies

Dynamic Routing	Dynamic routing support over tunnel interfaces	not supported
Topology	Hub & Spoke, P2P, P2MP supported	Only P2P supported Hub & Spoke not supported
SA status	Always keep SA if tunnel interface is up	If there is no corresponding traffic, the matching SA is released
Vendor Agnostic	Only certain VPN solutions are supported.	Various VPN solutions are supported.
Flexibility	Routing setup required when adding a new network	Applying new policies when adding new networks
Use cases	Source or destination NAT occurs when traversing VPN Duplicate subnets can exist in two LANs Hub & Spoke network topology configurable Primary & Backup VPN configurable Dynamic routing beyond VPN is configurable Multiple subnets can be accessed remotely	When accessing only one subnet to a remote location

2. IPSec VPN interworking

2.1 Network topology

The following is a network configuration that requires communications between individual virtual servers in two VPCs in SDS Cloud and servers in an on-premises data center. In the case of the on-premises data center, it is assumed that the VPN and individual servers exist within NAT and use private IP addresses.

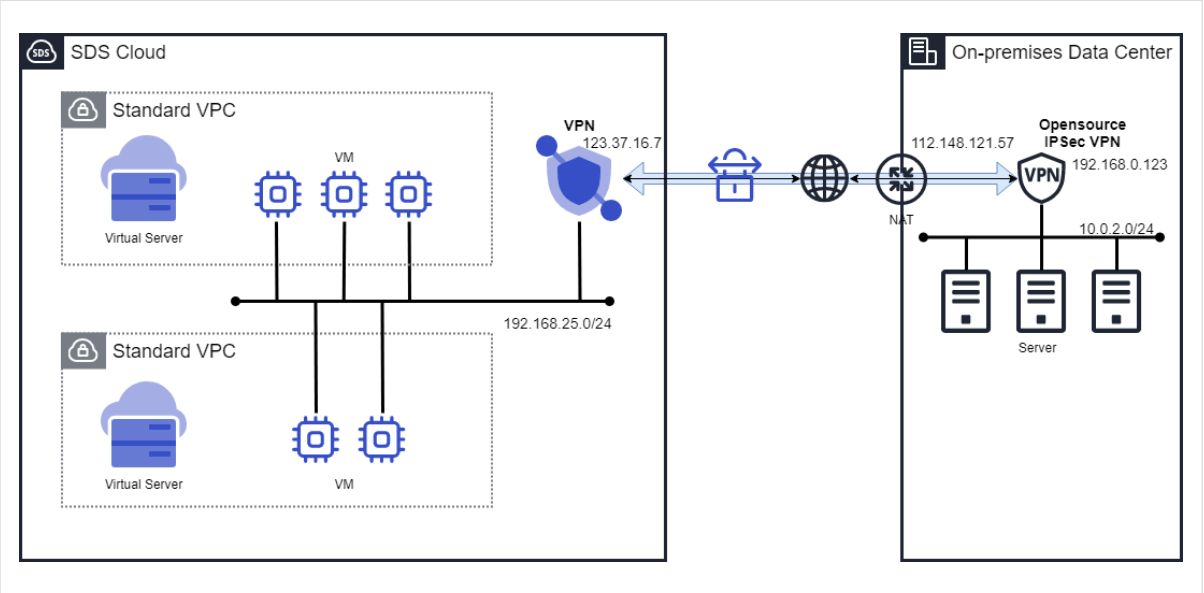


Figure 1. Network topology of IPSec VPN interworking

2.2 Main parameters

As the main parameters to define the IPSec VPN connection, the setting values based on VyOS are the following.

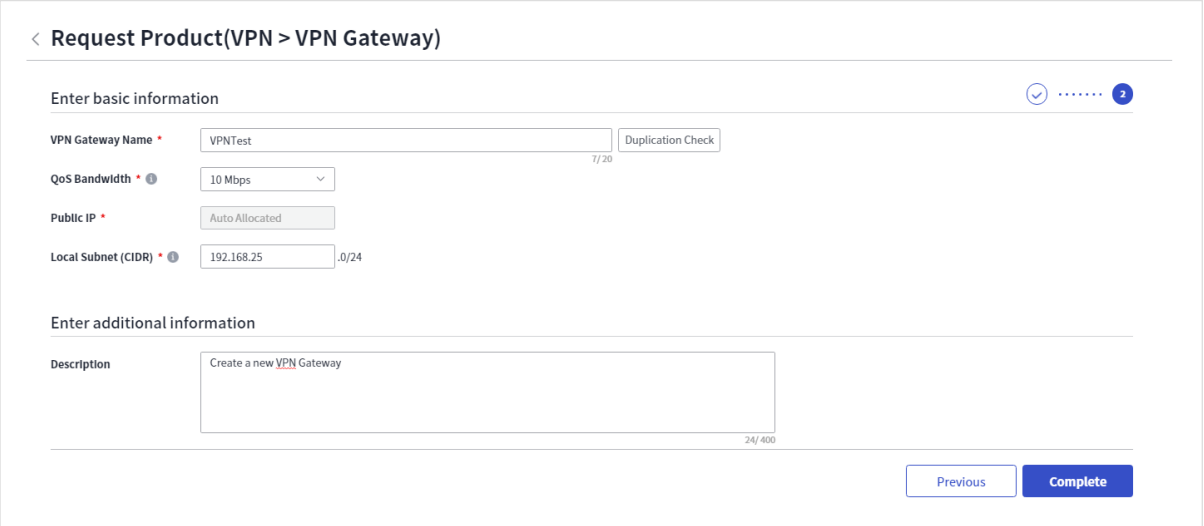
Parameters	Description
IPSec SA Mode	ESP & Tunnel mode
IPSec SA Lifetime	3600 seconds(10 Hours)
IPSec SA Proposal	Encryption AES256, Integrity(Hash) SHA256 Encryption and Hash Algorithms
IPSec PFS	Diffie-Hellman(DH) Group 2(modp_1024) Perfect Forward Secrecy, using dynamic keys to encrypt data
IKE SA DPD	Restart, interval=15 seconds, timeout=60 seconds Dead Peer Detection, checking the status of the IKE Peer, triggers a connection renegotiation attempt when timeout
IKE SA ikev2-reauth	yes Re-authentication for peer in rekeying process when using IKEv2

IKE SA Lifetime	86400 seconds(24 Hours)
IKE SA Proposal	DH Group 2, Encryption AES256, Integrity(Hash) SHA256 Proposal for IKE SAs, Cryptography and Hash Algorithms
Site-to-Site Authentication	id=112.158.121.57, mode=pre-shared-secret ID (usually public IP address of interface used for VPN connection) and method (Pre Shared Key) to be used for Site-to-Site VPN Peer authentication
Site-to-Site Connection Type	Connection-type=Initiate Initial connection attempt after booting and configuration
Site-to-Site ikev2-reauth	Ikev2-reauth=Inherit It inherits and uses the default behavior of IKE group
Site-to-Site local-address	local-address=192.168.0.123 VPN Interface IP address, private IP address if inside NAT
Site-to-Site vti	bind=vti1, esp-group Virtual Tunnel Interface (VTI) to be used for Tunnel as an option to use for route based VPN setup. Specifies the ESP group to be used for traffic encryption
Site-to-Site tunnel #	local prefix=172.20.10.0/24 remote prefix=192.168.25.0/24 This option is used for policy-based VPN settings and specifies the target traffic selector (Local subnet, Remote subnet) to be encrypted with IPSec

3. SDS Cloud VPN service configuration

3.1 Service application

After accessing the SDS Cloud console, select the project, and then select VPN from the Networking product line. After clicking product application in the product list, enter the VPN gateway name, public IP address (automatic input), local subnet (CIDR), and description and complete the process.

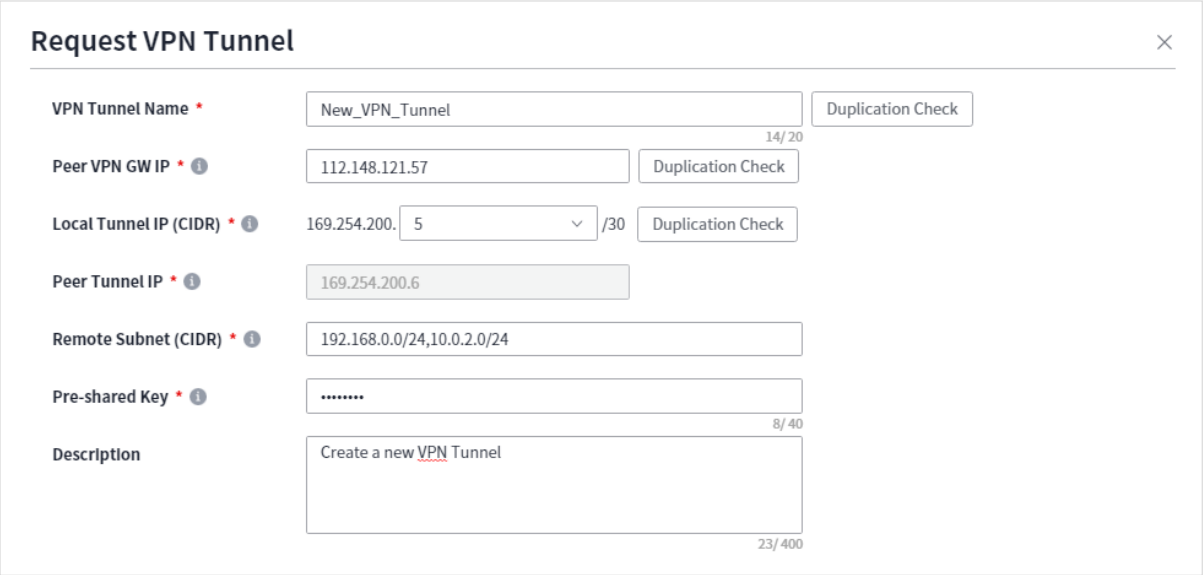


3.2 Creating VPN tunnels

Check the (active) status in VPN Gateway details, and select Create VPN tunnel from the VPN tunnel tab menu.

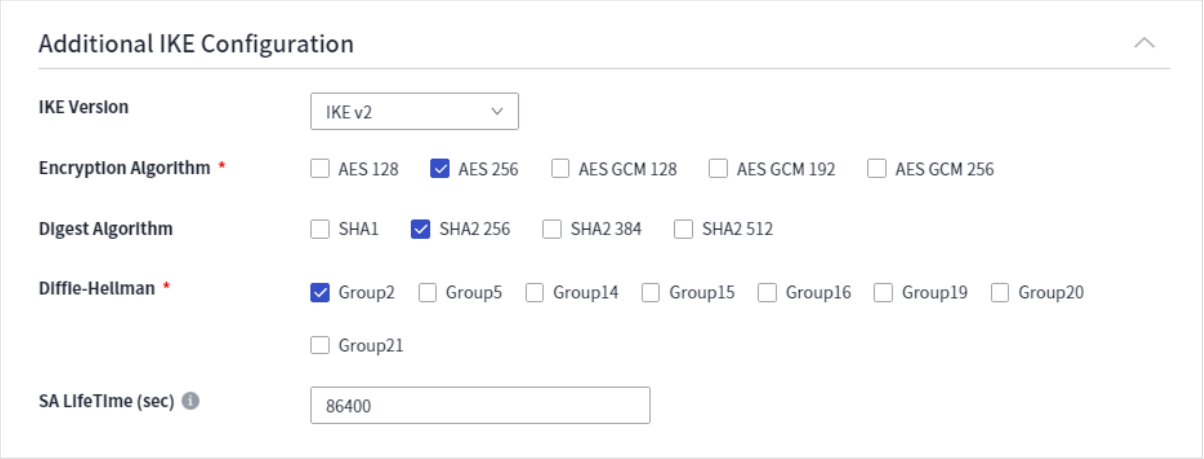
- Basic configurations

Enter the Tunnel name, Peer VPN Gateway IP address (For VPNs in NAT, the public IP address after the private IP address of the Peer VPN Gateway interface going through NAT), Local Tunnel IP (Virtual Tunnel Interface, IP address for VTI setting: 169.254.200.x/30), Peer Tunnel IP (automatically determined when local tunnel IP is selected), Remote Subnet (subnet bandwidth to be connected via Peer VPN Gateway), and Pre-shared Key (password to be used for authentication).



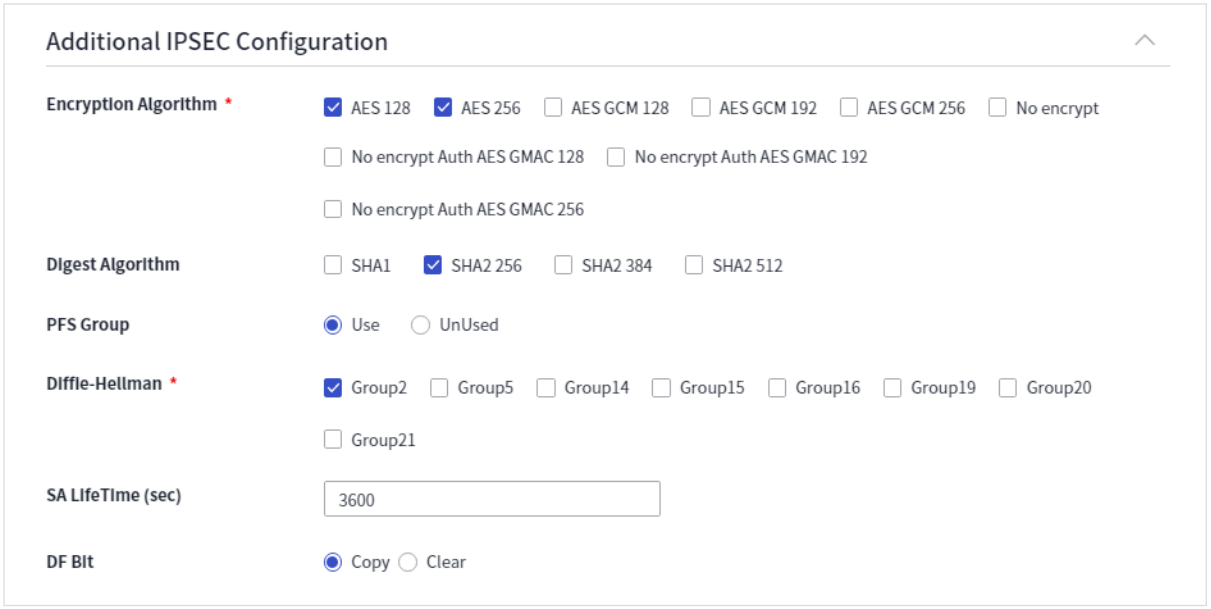
- Additional IKE configurations

Set the option (Proposal) to be used for IKE SA (Security Association). Various algorithms offered by the corresponding VPN may all be selected.
Select Key Exchange Protocol (IKEv1, IKEv2, IKE Flex= is dynamically determined as IKEv1 or v2 based on Peer VPN request), Proposal options for IKE SA Encryption Algorithm (AES256), Digest Algorithm (SHA2 256), and Diffie-Hellman Group (2, modp_1024), and enter the SA Lifetime value.



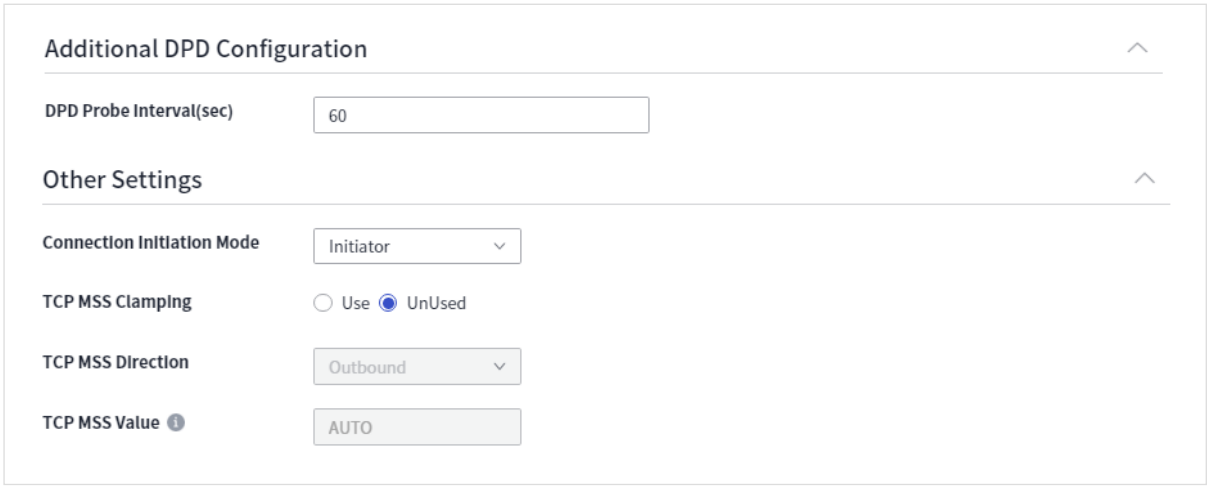
- Additional IPSec configurations

Set the options to be used for IPSec Security Association (SA). Various algorithms offered by the corresponding VPN may all be selected.
Select Proposal Options to use for IPSec SA Encryption Algorithm (AES256), whether or not to use Digest Algorithm (SHA2 256), PFS (Perfect Forward Secrecy), Diffie-Hellman Group (2, modp_1024), SA Lifetime (3600 seconds), and whether or not to copy DF (Don't Fragment).



- Other configurations

Enter Dead Peer Detection (DPD) Probe Interval and select Connection Mode (Initiator). TCP MSS (Maximum Segment Size) Clamping function will adjust the MSS value to prevent IP packet fragmentation with headers (IP, UDP, and ESP) added by IPSec traffic



encryption.

4. Open source IPSec VPN configuration

4.1 VyOS

VyOS is an open source network operating system (NOS) based on the Debian Linux distribution edition, which was a project forked from the open source Vyatta. It is often used as a router, firewall, or IPSec VPN.

The environment settings and examples below have been verified based on VyOS version 1.4 (Sagitta).

- Interface settings

A VPN gateway must have at least one internal and external interface each, and the external interface connected to the SDS Cloud VPN can be configured inside NAT. The VTI interface is a logical interface used for tunnels in a route-based VPN setup.

You can check each interface with the command below.

```
vyos@vyos:~$ show configuration
interfaces {
    ethernet eth0 {
        address 192.168.0.123/24
        hw-id 08:00:27:e9:29:03
        description OUTSIDE
        duplex auto
        speed auto
    }
    ethernet eth1 {
        address 10.0.2.1/24
        hw-id 08:00:27:08:97:23
        description INSIDE
    }
    loopback lo {
    }
    vti vti1 {
        address 169.254.200.6/30
    }
}
```

- Configuring settings

VyOS can be accessed through SSH service setting, and when connected, you can enter the edit mode with the ‘configure’ command.

```
vyos@vyos:~$ configure
[edit]

SSH Service configuration
set service ssh

IPSec ESP configurations
set vpn ipsec esp-group SDS-Cloud-VPN lifetime 386000
set vpn ipsec esp-group SDS-Cloud-VPN mode tunnel
set vpn ipsec esp-group SDS-Cloud-VPN pfs dh-group2
set vpn ipsec esp-group SDS-Cloud-VPN proposal 1 encryption aes256
set vpn ipsec esp-group SDS-Cloud-VPN proposal 1 hash sha256

IPSec IKE configurations
set vpn ipsec ike-group SDS-Cloud-VPN ikev2-reauth yes
set vpn ipsec ike-group SDS-Cloud-VPN key-exchange ikev2
set vpn ipsec ike-group SDS-Cloud-VPN lifetime 36000
set vpn ipsec ike-group SDS-Cloud-VPN proposal 1 encryption aes256
set vpn ipsec ike-group SDS-Cloud-VPN proposal 1 hash sha256
set vpn ipsec ike-group SDS-Cloud-VPN proposal dh-group 2

IPSec site-to-site tunnel configurations
set vpn ipsec ipsec-interfaces interface eth0
set vpn ipsec site-to-site peer 123.37.16.7 authentication id 112.148.121.57
set vpn ipsec site-to-site peer 123.37.16.7 authentication mode pre-shared-secret
set vpn ipsec site-to-site peer 123.37.16.7 authentication pre-shared-secret 'password'
set vpn ipsec site-to-site peer 123.37.16.7 connection-type initiate
set vpn ipsec site-to-site peer 123.37.16.7 ike-group SDS-Cloud-VPN
set vpn ipsec site-to-site peer 123.37.16.7 ikev2-reauth inherit
set vpn ipsec site-to-site peer 123.37.16.7 local-address 192.168.0.123
# CASE 1. Route based VPN configurations
set vpn ipsec site-to-site peer 123.37.16.7 vti bind vti1
set vpn ipsec site-to-site peer 123.37.16.7 vti esp-group SDS-Cloud-VPN
set protocols static route 192.168.25.0/24 next-hop 169.254.200.5
# CASE 2. Policy based VPN configurations
set vpn ipsec site-to-site peer 123.37.16.7 tunnel 0 esp-group SDS-Cloud-VPN
set vpn ipsec site-to-site peer 123.37.16.7 tunnel 0 local prefix 192.168.0.0/24, 10.0.2.0/24
set vpn ipsec site-to-site peer 123.37.16.7 tunnel 0 remote prefix 192.168.25.0/24

Other configurations: Adjustment of TCP MSS and MTU
set firewall options interface vti0 adjust-mss 1394
set interfaces vti vti0 mtu 1436
```

- Saving settings

Save the environment set with the command below and reflect it in the startup configuration.

```
vyos@vyos:~# commit
[edit]

vyos@vyos:~# save
Saving configuration to '/config/config.boot'...
Done
```

4.2 strongSwan

Strongswan is an open source IPSec VPN that works on a variety of platforms, including Linux, Android, macOS, and Windows. The environment settings and examples below have been verified based on strongswan 5.9.1 (Charon).

- Setting to enable packet forwarding

```
# Settings for network environment
[root@vpn ~]# vi /etc/sysctl.conf

net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0

# Apply new settings
[root@vpn ~]# sysctl –p
```

- VPN configurations

You can set the environment at /etc/strongswan/ipsec.conf, ipsec.secrets.

```
[root@vpn ~]# cat /etc/strongswan/ipsec.conf

# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

config setup
        strictcrlpolicy=yes
        uniqueids = no
       charondebug="cfg 2, ike 2, knl 2"

# Add connections here.
conn SDS-Cloud-VPN
        left=192.168.0.123
        leftid="112.148.121.57"
        right=123.37.16.7
        rightsubnet=192.168.25.0/24
        leftsubnet=10.0.2.0/24
        ike=aes256-sha256-modp1024!
        keyexchange=ikev2
        reauth=yes
        ikelifetime=86400s
        dpddelay=15s
        dpdtimeout=60s
        dpdaction=restart
        closeaction=none
        esp=aes256-sha256-modp1024!
        keylife=3600s
        rekeymargin=540s
        type=tunnel
        compress=no
        authby=secret
        auto=start
        keyingtries=%forever
```

```
[root@vpn ~]# cat /etc/ipsec.secrets

# ipsec.secrets - strongSwan IPsec secrets file
192.168.0.123 123.37.16.7 112.148.121.57 : PSK "password"
```

5. Testing and validating IPSec VPN connections

5.1 VyOS

- Check if IKE SA and IPSec SA are connected normally.

```
vyos@vyos:~$ show vpn ike sa
Peer ID / IP                            Local ID / IP
------------                            -------------
123.37.16.7 123.37.16.7                  192.168.0.123 112.158.121.57

 State   IKEVer   Encrypt        Hash            D-H Group        NAT-T   A-Time   L-Time
 -----   ------   -------        ----            ---------        -----   ------   ------
 up       IKEv2    AES_CBC_256   HMAC_SHA2_256_128 MODP_1024        yes     266      0

vyos@vyos:~$ show vpn ipsec sa
Connection                State    Uptime     Bytes In/Out     Packets In/Out     Remote address
Remote ID      Proposal
---------------------   -------   --------   --------------   ----------------   ----------------   -----------   -
--------------------------------------
peer_123-37-16-7_vti    up          4m34s        1K/2K                14/36                  123.37.16.7
N/A             AES_CBC_256/HMAC_SHA2_256_128/MODP_1024
```

- Confirmation on details of IPSec connection status

```
vyos@vyos:~$ show vpn ipsec state
src 192.168.0.123 dst 123.37.16.7
        proto esp spi 0xad716f3e reqid 1 mode tunnel
        replay-window 0 flag af-unspec
        auth-trunc                                                                  hmac(sha256)
0x5ba80c61dee59a161a22d4311d41b63479b223d9dded2f06f4709a59e2de4c26 128
        enc                                                                         cbc(aes)
0x32ba1855204aa95c2ef297dada14cec4ca59387e3c48422481b4813a66683dc2
        encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
        anti-replay context: seq 0x0, oseq 0x24, bitmap 0x00000000
        if_id 0x1
src 123.37.16.7 dst 192.168.0.123
        proto esp spi 0xcc38892a reqid 1 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc                                                                  hmac(sha256)
0x55bc3fee9add1aa0c3c1afc106455325227848a6449bbddbbdcb583252105a05 128
        enc                                                                         cbc(aes)
0x635912634140900a7a73dcbe07896a469b5cd224c2ad5b69b425eebae3d01304
        encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
        anti-replay context: seq 0xe, oseq 0x0, bitmap 0x00003fff
        if_id 0x1
```

- IPSec VPN debugging

```
vyos@vyos:~$ show vpn debug
Status of IKE charon daemon (strongSwan 5.9.1, Linux 5.10.57-amd64-vyos, x86_64):
  uptime: 5 minutes, since Aug 12 10:44:37 2021
  malloc: sbrk 2011136, mmap 0, used 1058224, free 952912
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 4
  loaded plugins: charon test-vectors ldap pkcs11 tpm aesni aes rc2 sha2 sha1 md5 mgf1 rdrand random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl gcrypt af-alg fips-prf gmp curve25519 agent chapoly xcbc cmac hmac ctr ccm gcm drbg curl attr kernel-netlink resolve socket-default connmark stroke vici updown eap-identity eap-aka eap-md5 eap-gtc eap-mschapv2 eap-radius eap-tls eap-ttls eap-tnc xauth-generic xauth-eap xauth-pam tnc-tnccs dhcp lookip error-notify certexpire led addrblock counters
Listening IP addresses:
  172.20.10.5
Connections:
peer_123-37-16-7:   192.168.0.123...123.37.16.7   IKEv2, dpddelay=15s
peer_123-37-16-7:    local:   [112.158.121.57] uses pre-shared key authentication
peer_123-37-16-7:    remote: uses pre-shared key authentication
peer_123-37-16-7_vti:    child:   0.0.0.0/0 ::/0 === 0.0.0.0/0 ::/0 TUNNEL, dpdaction=restart
Security Associations (1 up, 0 connecting):
peer_123-37-16-7[1]:   ESTABLISHED   5   minutes   ago, 192.168.0.123[112.158.121.57]...123.37.16.7[123.37.16.7]
peer_123-37-16-7[1]: IKEv2 SPIs: 81ff24e12c24e9ef_i* aa77470fcbc244cc_r, rekeying in 3 hours
peer_123-37-16-7[1]:   IKE   proposal: AES_CBC_256/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_1024
peer_123-37-16-7_vti{1}:   INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: ce5885e7_i 6fc7c1f6_o
peer_123-37-16-7_vti{1}:    AES_CBC_256/HMAC_SHA2_256_128, 0 bytes_i (0 pkts, 144s ago), 0 bytes_o (0 pkts, 145s ago), rekeying in 49 minutes
peer_123-37-16-7_vti{1}:    0.0.0.0/0 === 0.0.0.0/0
peer_123-37-16-7_vti{2}:   INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: cc38892a_i ad716f3e_o
peer_123-37-16-7_vti{2}:   AES_CBC_256/HMAC_SHA2_256_128/MODP_1024, 1176 bytes_i (14 pkts, 144s ago), 3024 bytes_o (36 pkts, 145s ago), rekeying in 50 minutes
peer_123-37-16-7_vti{2}:    0.0.0.0/0 === 0.0.0.0/0
```

- Check on packet dump

```
vyos@vyos:~$ sudo su
root@vyos:/home/vyos# tcpdump -i vti1
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on vti1, link-type RAW (Raw IP), snapshot length 262144 bytes
10:50:43.716524 IP 192.168.25.3 > 10.0.2.1: ICMP echo request, id 9970, seq 11, length 64
10:50:43.716543 IP 10.0.2.1 > 192.168.25.3: ICMP echo reply, id 9970, seq 11, length 64
10:50:44.720615 IP 192.168.25.3 > 10.0.2.1: ICMP echo request, id 9970, seq 12, length 64
10:50:44.720633 IP 10.0.2.1 > 192.168.25.3: ICMP echo reply, id 9970, seq 12, length 64
```

5.2 strongSwan

- IPSec connection and debugging

```
[root@vpn ~]# strongswan start

[root@vpn ~]# strongswan statusall
Status of IKE charon daemon (strongSwan 5.9.1, Linux 4.18.0-147.8.1.el8_1.x86_64, x86_64):
  uptime: 2 minutes, since Aug 26 02:23:57 2021
  malloc: sbrk 1998848, mmap 0, used 1015712, free 983136
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 4
  loaded plugins: charon test-vectors ldap pkcs11 tpm aesni aes rc2 sha2 sha1 md5 mgf1 rdrand random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl gcrypt af-alg fips-prf gmp curve25519 agent chapoly xcbc cmac hmac ctr ccm gcm curl attr kernel-netlink resolve socket-default connmark stroke vici updown eap-identity eap-aka eap-md5 eap-gtc eap-mschapv2 eap-radius eap-tls eap-ttls eap-tnc xauth-generic xauth-eap xauth-pam tnc-tnccs dhcp lookip error-notify certexpire led addrblock counters
Listening IP addresses:
  192.168.0.123
Connections:
SDS-Cloud-VPN:  192.168.0.123...123.37.16.7  IKEv2, dpddelay=15s
SDS-Cloud-VPN:   local:  [112.158.121.57] uses pre-shared key authentication
SDS-Cloud-VPN:   remote: [123.37.16.10] uses pre-shared key authentication
SDS-Cloud-VPN:   child:  10.0.2.0/24 === 192.168.25.0/24 TUNNEL, dpdaction=restart
Security Associations (1 up, 0 connecting):
SDS-Cloud-VPN[1]: ESTABLISHED 2 minutes ago, 192.168.0.123[112.158.121.57]...123.37.16.7[123.37.16.7]
SDS-Cloud-VPN[1]: IKEv2 SPIs: 771dfb03de6d3ac4_i* 853390be58e29d9d_r, pre-shared key reauthentication in 23 hours
SDS-Cloud-VPN[1]: IKE proposal: AES_CBC_256/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_1024
SDS-Cloud-VPN{1}:  INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: c9ce42eb_i 7a716a14_o
SDS-Cloud-VPN{1}:  AES_CBC_256/HMAC_SHA2_256_128, 0 bytes_i (0 pkts, 5s ago), 0 bytes_o (0 pkts, 5s ago), rekeying in 39 minutes
SDS-Cloud-VPN{1}:   10.0.2.0/24 === 192.168.25.0/24
```

- Check on IPSec connection status

```
[root@vpn ~]# ip -s xfrm state
src 192.168.0.123 dst 123.37.16.7
        proto esp spi 0x7a716a14(2054253076) reqid 1(0x00000001) mode tunnel
        replay-window 0 seq 0x00000000 flag af-unspec (0x00100000)
        auth-trunc                                                    hmac(sha256)
0x8fce17d5f1f35fba72c26bb4b333bdb3954aa889e8c8050b0f2c6701a4e9c8b7 (256 bits) 128
        enc                                                           cbc(aes)
0x95db5c60b4d21f1fb43e4d006e6104d51cd3962083ea0d96565ed0de7e533c84 (256 bits)
        encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
        anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
        lifetime config:
          limit: soft (INF)(bytes), hard (INF)(bytes)
          limit: soft (INF)(packets), hard (INF)(packets)
          expire add: soft 2633(sec), hard 3600(sec)
          expire use: soft 0(sec), hard 0(sec)
        lifetime current:
          0(bytes), 0(packets)
          add 2021-08-26 02:24:01 use -
        stats:
          replay-window 0 replay 0 failed 0
src 123.37.16.7 dst 192.168.0.123
        proto esp spi 0xc9ce42eb(3385737963) reqid 1(0x00000001) mode tunnel
        replay-window 32 seq 0x00000000 flag af-unspec (0x00100000)
        auth-trunc                                                    hmac(sha256)
0x8e28679a4348bbda7787eaad61d7246704f33233f986a860cda6fdc488bafaf7 (256 bits) 128
        enc                                                           cbc(aes)
0x12b9d972b3d06cc25993f7640d97e1dee18b9723e336963806e340e5c5b21d34 (256 bits)
        encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
        anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
        lifetime config:
          limit: soft (INF)(bytes), hard (INF)(bytes)
          limit: soft (INF)(packets), hard (INF)(packets)
          expire add: soft 2536(sec), hard 3600(sec)
          expire use: soft 0(sec), hard 0(sec)
        lifetime current:
          0(bytes), 0(packets)
          add 2021-08-26 02:24:01 use -
        stats:
          replay-window 0 replay 0 failed 0
```

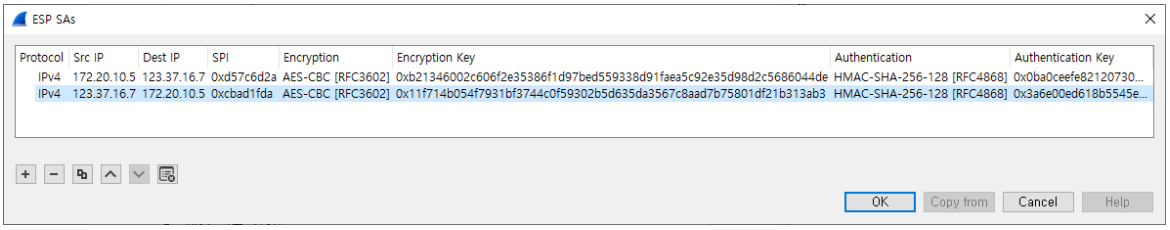
5.3 Traffic decryption

- Decoding of ESP packets using Wireshark

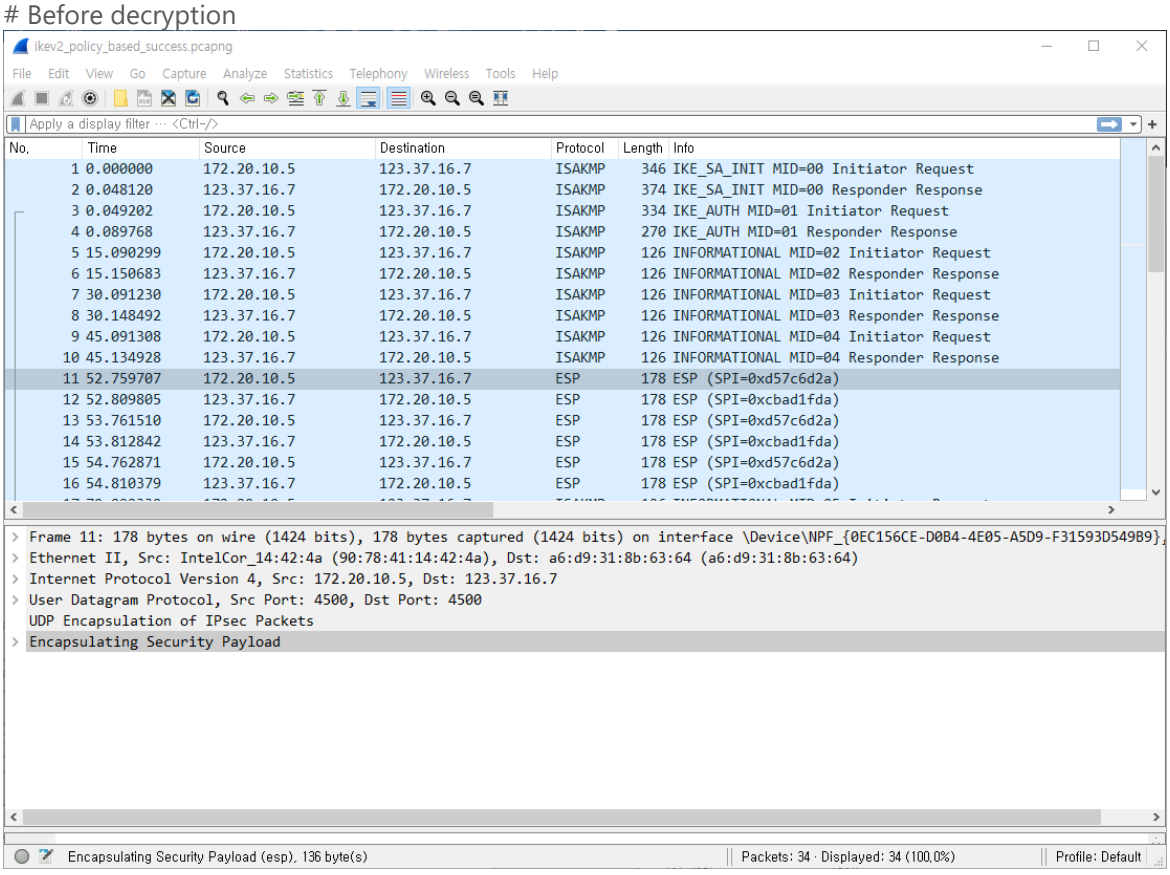
From the Wireshark menu, select Edit > Preferences > Protocols > ESP, check the decryption option (Attepts to detect/decode encrypted ESP payloads), and edit 'Edit SAs' option. Enter and save information related to IPSec ESP SA states (Source IP,

Destination IP, SPI, Encryption Algorithm, Encryption Key, Hash Algorithm, and Hash Key) to check the decrypted ESP packet.

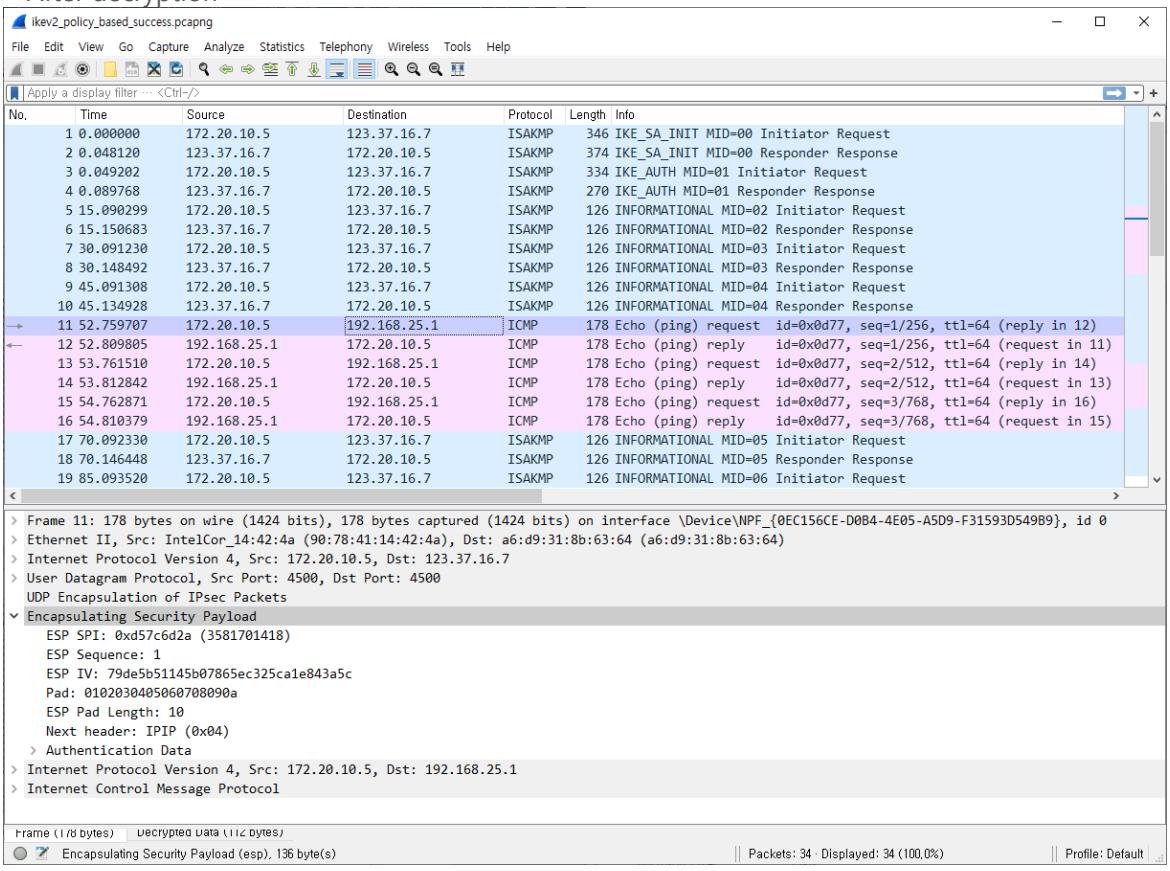
```
vyos@vyos:~$ show vpn ipsec state
src 172.20.10.5 dst 123.37.16.7
        proto esp spi 0xd57c6d2a reqid 2 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc                                                    hmac(sha256)
0x0ba0ceefe82120730d89cab2e43d6bfc198ad9f37782789184af9669190c408b 128
        enc                                                           cbc(aes)
0xb21346002c606f2e35386f1d97bed559338d91faea5c92e35d98d2c5686044de
        encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
src 123.37.16.7 dst 172.20.10.5
        proto esp spi 0xcbad1fda reqid 2 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc                                                    hmac(sha256)
0x3a6e00ed618b5545e117236d71c8ed5fec19e5452b4e8978f6fbd3c2457322d1 128
        enc                                                           cbc(aes)
0x11f714b054f7931bf3744c0f59302b5d635da3567c8aad7b75801df21b313ab3
        encap type espinudp sport 4500 dport 4500 addr 0.0.0.0
```



Before decryption



After decryption



6. Compatibility and considerations

- Compatibility with VyOS version 1.4 or earlier

There is a compatibility issue with route-based VPNs (VTI connection) in the IKEv2 settings, so the connection cannot be established normally.
You could set it up as a policy-based VPN if you must use IKEv2, or use IKEv1 instead.

- Policy-based VPN does not support IKEv1 Multi-Subnet

For the policy-based VPNs in IKEv1 setting, local and remote subnets can only be configured with one subnet.
If you want to set up a policy to connect multiple subnets, you need to change the key exchange protocol to IKEv2.