Configure Secure Overlay with BGP Route Announcements

Contents

Introduction

Components Used

- BGP Route Announcement

Configuration Example

- Topology Diagram
- Initial Setup
- FlexVPN Server Configuration on the Catalyst 8000v Router
  - 1. Create an IKEv2 Proposal
  - 2. Create an IKEv2 Policy and Associate it with the Proposal.
  - 3. Configure the IKEv2 Authorization Policy
  - 4. Create a IKEv2 Profile
  - 5. Create an IPsec Transform Set
  - 6. Remove Default IPsec Profile
  - 7. Create an IPsec Profile and Associate it with a Transform Set and the IKEv2 Profile.
  - 8. Create a Virtual Template
- NFVIS Secure Overlay Minimum Configuration
  - Review Overlay Status
- BGP Route Announcment Configuration for the FlexVPN Server
- BGP Configuration on NFVIS
  - BGP Review
  - Ensure that the Private Subnets from the FlexVPN Server were Advertised through BGP

Troubleshooting

- NFVIS (FlexVPN Client)
  - NFVIS Log Files
  - Internal Kernel strongswan injected routes
  - Review IPsec0 Interface Status
- Head-End (FlexVPN Server)
  - Review IPsec SAs Build Between Peers
  - Display Active Crypto (encryption) Sessions
  - Reset VPN Connections
  - Perform Debugs for Additional Troubleshooting

Related Articles and Documentation

Introduction

This document describes how to configure secure overlay and eBGP announcements on NFVIS for exclusive vBranch traffic management.

Components Used

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

- ENCS5412 running NFVIS 4.7.1
- Catalyst 8000v running Cisco IOS® XE 17.09.03a

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.

BGP Route Announcement

NFVIS BGP feature works with the secure overlay feature to learn routes from the BGP neighbor over a secure overlay tunnel. These learnt routes or subnets are added into the NFVIS routing table for the secure tunnel, which makes the routes accessible over the tunnel. Since Secure Overlay only permits 1 single private route to be learned from the tunnel; configuring BGP enables overcoming this limitation by establishing adjacency through the encrypted tunnel and injecting exported routes into the NFVIS vpnv4 routing table and vice versa.

Configuration Example

Topology Diagram

The goal for this configuration is to reach the Management IP address of NFVIS from the c8000v. Once the tunnel is established, it is possible to advertise more routes from the private-vrf subnets using eBGP route announcements.

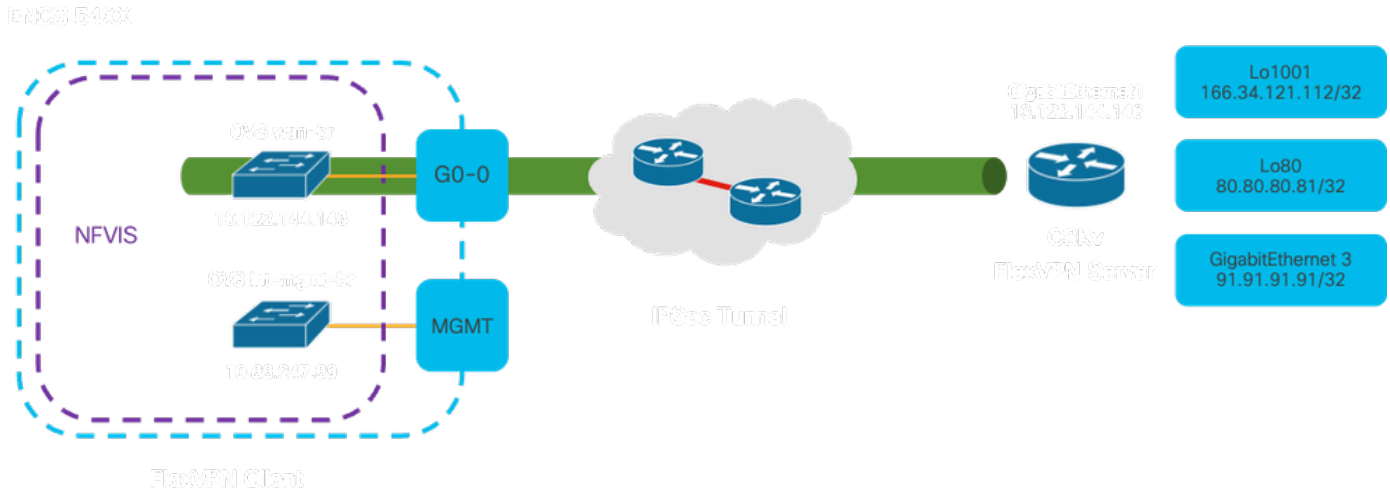


Figure 1. Topology Diagram for the Example prepared on this article

Initial Setup

Configure the relevant IP addressing on the FlexVPN Server (all within global configuration mode)

```
vrf definition private-vrf
 rd 65000:7
 address-family ipv4
 exit-address-family
```

```
vrf definition public-vrf
 address-family ipv4
 exit-address-family

interface GigabitEthernet1
 description Public-Facing Interface
 vrf forwarding public-vrf
 ip address 10.88.247.84 255.255.255.224

interface Loopback1001
 description Tunnel Loopback
 vrf forwarding private-vrf
 ip address 166.34.121.112 255.255.255.255

interface Loopback80
 description Route Announced Loopback
 vrf forwarding private-vrf
 ip address 81.81.81.1 255.255.255.255

interface GigabitEthernet3
 description Route Announced Physical Interface
 vrf forwarding private-vrf
 ip address 91.91.91.1 255.255.255.0
```

For NFVIS, configure the WAN and MGMT interface accordingly

```
system settings mgmt ip address 192.168.1.1 255.255.255.0
system settings wan ip address 10.88.247.89 255.255.255.224
system settings default-gw 10.88.247.65
system settings ip-receive-acl 0.0.0.0/0
 service  [ ssh https netconf scpd ]
 action   accept
 priority 10
!
```

FlexVPN Server Configuration on the Catalyst 8000v Router

1. Create an IKEv2 Proposal

It specifies the security protocols and algorithms that two VPN endpoints must use during the initial phase (Phase 1) of establishing a secure communication channel. The purpose of the IKEv2 proposal is to outline the parameters for authentication, encryption, integrity, and key exchange, thereby ensuring that both endpoints agree on a common set of security measures before exchanging any sensitive data.

```
crypto ikev2 proposal uCPE-proposal
 encryption aes-cbc-256
 integrity sha512
 group 16 14
```

Where:

encryption <algorithm>	The proposal includes the encryption algorithms (like AES or 3DES) that the VPN must use to protect the data. Encryption prevents eavesdroppers from being able to read the traffic that passes through the VPN tunnel.
integrity <hash>	It specifies the algorithms (such as SHA-512) used for ensuring the integrity and authenticity of the messages exchanged during the IKEv2 negotiation. This prevents tampering and replay attacks.

2. Create an IKEv2 Policy and Associate it with the Proposal.

It is a configuration set that dictates the parameters for the initial phase (phase 1) of establishing an IPsec VPN connection. It primarily focuses on how the VPN endpoints authenticate each other and establish a secure communication channel for the VPN setup.

```
crypto ikev2 policy uCPE-policy
 match fvrf public-vrf
 proposal uCPE-proposal
```

3. Configure the IKEv2 Authorization Policy

IKEv2 is a protocol used to set up a secure session between two endpoints on a network, and the authorization policy is a set of rules that determines what resources and services a VPN client is allowed to access once the VPN tunnel is established.

```
crypto ikev2 authorization policy uCPE-author-pol
 pfs
 route set interface Loopback1001
```

Where:

pfs	Perfect Forward Secrecy (PFS) is a feature that enhances the security of a VPN connection by ensuring that each new encryption key is independently secure, even if previous keys are compromised.
route set interface <interface-name>	When a VPN session is successfully established, the routes defined in the IKEv2 authorization policy are automatically added to the device routing table. This ensures that traffic destined for the networks specified in the route set is correctly routed through the VPN tunnel.

4. Create a IKEv2 Profile

An IKEv2 (Internet Key Exchange version 2) policy is a set of rules or parameters used during the IKEv2 phase of establishing an IPsec (Internet Protocol Security) VPN tunnel. IKEv2 is a protocol that facilitates the secure exchange of keys and negotiation of security associations (SAs) between two parties wishing to communicate securely over an untrusted network, such as the internet. The IKEv2 policy defines how this negotiation must take place, specifying various security parameters that both parties must agree upon to establish a secure and encrypted communication channel.

IKEv2 profile MUST have:

- A local and a remote authentication method.
- A match identity or a match certificate or match any statement.

```
crypto ikev2 profile uCPE-profile
 description uCPE profile
 match fvrf public-vrf
 match identity remote any
 authentication remote pre-share key ciscociscocisco123
 authentication local pre-share key ciscociscocisco123
 dpd 60 2 on-demand
 aaa authorization group psk list default uCPE-author-pol local
 virtual-template 1 mode auto
```

Where:

match fvrf public-vrf	Make a profile vrf-aware.
match identity remote any	Measure to recognize a incoming session as valid; in this case, anyone.
authentication remote pre-share key ciscociscocisco123	Specifies that the remote peer must be authenticated using pre-shared keys.
authentication local pre-share key ciscociscocisco123	Specifies that this device (local) must authenticate using pre-shared keys.
dpd 60 2 on-demand	Dead Peer Detection; if no packets were received over a minutec (60 seconds), send 2 dpd packets within this 60 sec interval.
aaa authorization group psk list default uCPE-author-pol local	Route assignment.
virtual-template 1 mode auto	Bind to a virtual-template.

5. Create an IPsec Transform Set

It defines a set of security protocols and algorithms that must be applied to the data traffic passing through the IPsec tunnel. Essentially, the transform set specifies how the data must be encrypted and authenticated, ensuring secure transmission between VPN endpoints. Tunnel mode configures the IPsec tunnel to encapsulate the entire IP packet for secure transport across the network.

```
crypto ipsec transform-set tset_aes_256_sha512 esp-aes 256 esp-sha512-hmac
 mode tunnel
```

Where:

set transform-set <transform-set-name>	Specifies the encryption and integrity algorithms (Example: AES for encryption and SHA for integrity) that must be used to protect the data flowing through the VPN tunnel.
set ikev2-profile <ikev2-profile-name>	Defines the parameters for the negotiation of the security associations (SAs) in phase 1 of the VPN setup, including encryption algorithms, hash algorithms, authentication methods, and the Diffie-Hellman group.
set pfs <group>	An optional setting that, if enabled, ensures that each new encryption key is

	unrelated to any previous key, enhancing security.

6. Remove Default IPsec Profile

Removing the default IPsec profile is a practice adopted for several reasons related to security, customization, and system clarity. The default IPsec profile can not meet the specific security policies or requirements of your network. Removing it ensures that no VPN tunnels inadvertently use suboptimal or insecure settings, reducing the risk of vulnerabilities.

Each network has unique security requirements, including specific encryption and hashing algorithms, key lengths, and authentication methods. Removing the default profile encourages the creation of custom profiles tailored to these specific needs, ensuring the best possible protection and performance.

```
no crypto ipsec profile default
```

7. Create an IPsec Profile and Associate it with a Transform Set and the IKEv2 Profile.

An IPsec (Internet Protocol Security) profile is a configuration entity that encapsulates the settings and policies used to establish and manage IPsec VPN tunnels. It serves as a template that can be applied to multiple VPN connections, standardizing security parameters and simplifying the management of secure communication across a network.

```
crypto ipsec profile uCPE-ips-prof
 set security-association lifetime seconds 28800
 set security-association idle-time 1800
 set transform-set tset_aes_256_sha512
 set pfs group14
 set ikev2-profile uCPE-profile
```

8. Create a Virtual Template

The Virtual-Template interface acts as a dynamic template for virtual access interfaces, providing a scalable and efficient way to manage VPN connections. It allows for the dynamic instantiation of Virtual-Access interfaces. When a new VPN session is initiated, the device creates a Virtual-Access interface based on the configuration specified in the Virtual-Template. This process supports a large number of remote clients and sites by dynamically allocating resources as needed without the need for pre-configured physical interfaces for each connection.

By using Virtual-Templates, FlexVPN deployments can scale efficiently as new connections are established, without the need for manual configuration of each individual session.

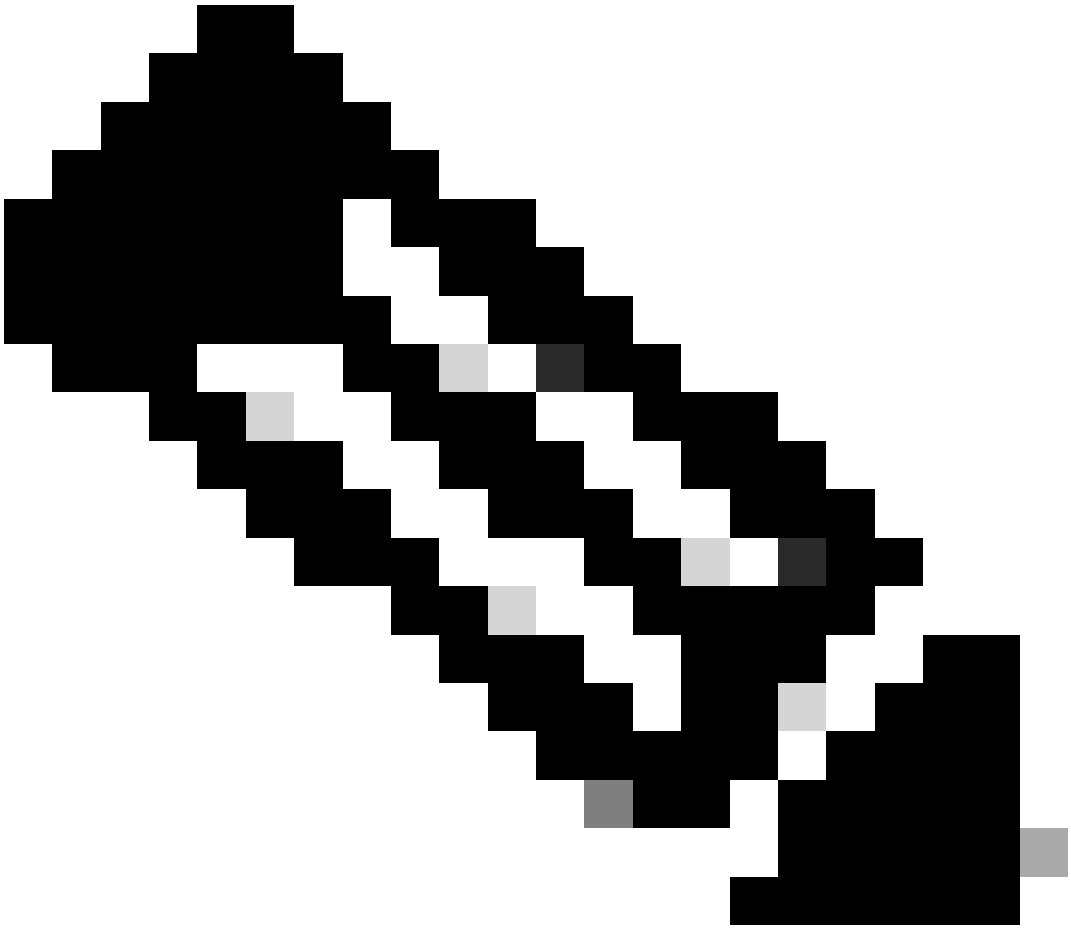
```
interface Virtual-Template1 type tunnel
 vrf forwarding private-vrf
 ip unnumbered Loopback1001
 ip mtu 1400
 ip tcp adjust-mss 1380
 tunnel mode ipsec ipv4
 tunnel vrf public-vrf
```

```
 tunnel protection ipsec profile uCPE-ips-prof
```

NFVIS Secure Overlay Minimum Configuration

Configure the secure-overlay instance

```
secure-overlay myconn local-bridge wan-br local-system-ip-addr 10.122.144.146 local-system-ip-subnet 10
 ike-cipher aes256-sha512-modp4096 esp-cipher aes256-sha512-modp4096
 psk local-psk ciscociscocisco123 remote-psk ciscociscocisco123
 commit
```



Note: When configuring BGP route announcement over an IPSec tunnel, ensure that you configure the secure overlay to use a virtual IP address (not sourced from a physical interface or OVS bridge) for the local tunnel IP address. For the example above, these are the virtual addressing commands changed: local-system-ip-addr 10.122.144.146 local-system-ip-subnet 10.122.144.128/27

Review Overlay Status

```
show secure-overlay
secure-overlay myconn
 state                           up
 active-local-bridge             wan-br
 selected-local-bridge           wan-br
 active-local-system-ip-addr     10.122.144.146
 active-remote-interface-ip-addr 10.88.247.84
 active-remote-system-ip-addr    166.34.121.112
 active-remote-system-ip-subnet  166.34.121.112/32
 active-remote-id                10.88.247.84
```

BGP Route Announcment Configuration for the FlexVPN Server

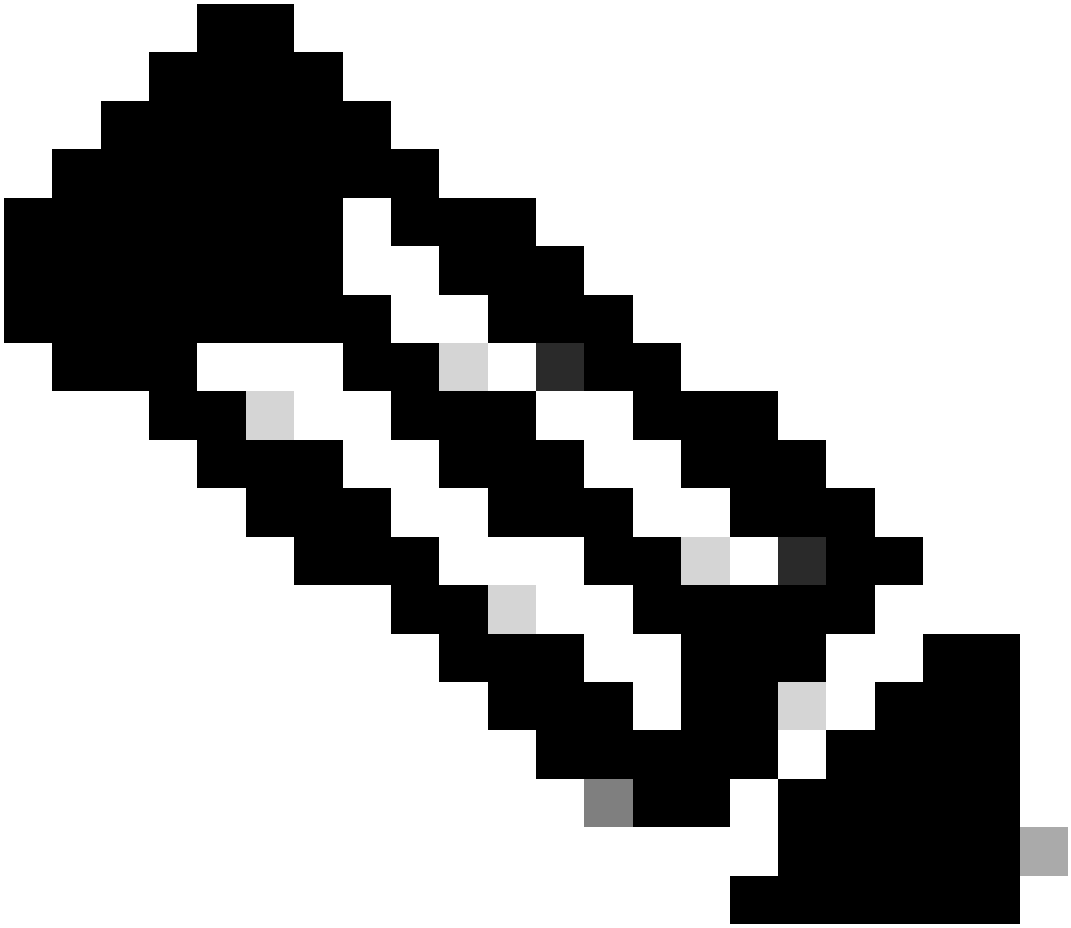
This setup must use eBGP for the the peerings, where the source address (virtual IP address for the local tunnel IP) subnet from NFVIS side must be added to the listen range.

```
router bgp 65000
 bgp router-id 166.34.121.112
 bgp always-compare-med
 bgp log-neighbor-changes
 bgp deterministic-med
 bgp listen range 10.122.144.0/24 peer-group uCPEs
 bgp listen limit 255
 no bgp default ipv4-unicast
 address-family ipv4 vrf private-vrf
  redistribute connected
  redistribute static
  neighbor uCPEs peer-group
  neighbor uCPEs remote-as 200
  neighbor uCPEs ebgp-multihop 10
  neighbor uCPEs timers 610 1835
 exit-address-family
```

Where:

bgp always-compare-med	Configures the router to always compare the MED (Multi-Exit Discriminator) attribute for all routes, regardless of their originating AS.
bgp log-neighbor-changes	Enables logging for events related to changes in BGP neighbor relationships.
bgp deterministic-med	Ensures the comparison of the MED for paths from neighbors in different autonomous systems.
bgp listen range <network>/<mask> peer-group <peer-group-name>	Enables dynamic neighbor discovery within the specified IP range (network/mask) and assigns discovered neighbors to the peer group name. This simplifies configuration by applying common settings to all peers in the group.
bgp listen limit 255	Sets the maximum number of dynamic BGP neighbors that can be accepted within the listen range to 255.
no bgp default ipv4-unicast	Disables the automatic sending of IPv4 unicast routing information to

	BGP neighbors, requiring explicit configuration to enable this.
redistribute connected	Redistributes routes from directly connected networks into BGP (Private subnets from the FlexVPN server that belong to the private-vrf)
redistribute static	Redistributes static routes into BGP.
neighbor uCPEs ebgp-multihop 10	Allows EBGP (External BGP) connections with peers in the peer group to span up to 10 hops, useful for connecting devices not directly adjacent.
neighbor uCPEs timers <keep-alive> <hold-down>	Sets the BGP keepalive and hold-down timers for neighbors in the peer group respectivley (610 seconds and 1835 seconds for the example).



Note: A outbound prefix list can be configured to control neighbor route advertisements in the peer group: neighbor prefix-list out

BGP Configuration on NFVIS

Start the BGP process with eBGP neighborship settings

```
router bgp 200
 router-id 10.122.144.146
 neighbor 166.34.121.112 remote-as 65000
 commit
```

BGP Review

This output reveals the condition of a BGP session as reported by the BIRD Internet Routing Daemon. This routing software is responsible for handling IP routes and making decisions regarding their direction. From the information given, its clear that the BGP session is in an "Established" state, indicating successful completion of the BGP peering process, and the session is currently active. It has successfully imported four routes, and its noted that there is an upper limit of 15 routes that can be imported.

```
nfvis# support show bgp
BIRD 1.6.8 ready.
name     proto    table    state  since       info
bgp_166_34_121_112 BGP      bgp_table_166_34_121_112 up     09:54:14    Established
  Preference:     100
  Input filter:   ACCEPT
  Output filter:  ACCEPT
  Import limit:   15
    Action:       disable
  Routes:         4 imported, 0 exported, 8 preferred
  Route change stats:     received   rejected   filtered    ignored   accepted
    Import updates:              4          0          0          0          4
    Import withdraws:            0          0        ---          0          0
    Export updates:              4          4          0        ---          0
    Export withdraws:            0        ---        ---        ---          0
  BGP state:          Established
    Neighbor address: 166.34.121.112
    Neighbor AS:      65000
    Neighbor ID:      166.34.121.112
    Neighbor caps:    refresh enhanced-refresh AS4
    Session:          external multihop AS4
    Source address:   10.122.144.146
    Route limit:      4/15
    Hold timer:       191/240
    Keepalive timer:  38/80
```

Ensure that the Private Subnets from the FlexVPN Server were Advertised through BGP

When configuring BGP route announcement, the only configurable address-family or transmission combination is ipv4 unicastfor IPSec. To view the BGP status, the configurable address-family or transmission for IPSec is vpnv4 unicast.

```
nfvis# show bgp vpnv4 unicast
Family Transmission Router ID       Local AS Number
vpnv4  unicast      10.122.144.146  200
```

With the **show bgp vpnv4 unicast route** command, you can retrieve information about the VPNv4 unicast

routes known to the BGP process.

```
nfvis# show bgp vpnv4 unicast route
Network            Next-Hop        Metric LocPrf Path
81.81.81.1/32      166.34.121.112  0      100    65000 ?
91.91.91.0/24      166.34.121.112  0      100    65000 ?
10.122.144.128/27  166.34.121.112  0      100    65000 ?
166.34.121.112/32  166.34.121.112  0      100    65000 ?
```

For the head-end VPN server, an overview of the BGP configuration and operational state can be generated to quickly assess the health and configuration of BGP sessions.

```
c8000v# show ip bgp summary
Number of dynamically created neighbors in vrf private-vrf: 1/(100 max)
Total dynamically created neighbors: 1/(255 max), Subnet ranges: 1
```

Additionaly, detailed information about the VPNv4 (VPN over IPv4) routing table entries managed by BGP can be displayed, it must include specific attributes of each VPNv4 route, such as the routes prefix, next-hop IP address, the originating AS number, and various BGP attributes like local preference, MED (Multi-Exit Discriminator), and community values.

```
c8000v# show ip bgp vpnv4 all
BGP table version is 5, local router ID is 166.34.121.112
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 65000:7 (default for vrf private-vrf)
 *>   10.122.144.128/27
                      0.0.0.0                  0         32768 ?
 *>   81.81.81.1/32    0.0.0.0                  0         32768 ?
 *>   91.91.91.0/24    0.0.0.0                  0         32768 ?
 *>   166.34.121.112/32
                      0.0.0.0                  0         32768 ?
```

Troubleshooting

NFVIS (FlexVPN Client)

NFVIS Log Files

You can view all initialization and error logs for the IPsec phases from the NFVIS **charon.log** log file:

```
nfvis# show log charon.log
Feb  5 07:55:36.771 00[JOB] spawning 16 worker threads
Feb  5 07:55:36.786 05[CFG] received stroke: add connection 'myconn'
Feb  5 07:55:36.786 05[CFG] added configuration 'myconn'
Feb  5 07:55:36.787 06[CFG] received stroke: initiate 'myconn'
Feb  5 07:55:36.787 06[IKE] <myconn|1> initiating IKE_SA myconn[1] to 10.88.247.84
Feb  5 07:55:36.899 06[ENC] <myconn|1> generating IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_
Feb  5 07:55:36.899 06[NET] <myconn|1> sending packet: from 10.88.247.89[500] to 10.88.247.84[500] (741
Feb  5 07:55:37.122 09[NET] <myconn|1> received packet: from 10.88.247.84[500] to 10.88.247.89[500] (80
Feb  5 07:55:37.122 09[ENC] <myconn|1> parsed IKE_SA_INIT response 0 [ SA KE No V V V V N(NATD_S_IP) N(
Feb  5 07:55:37.122 09[IKE] <myconn|1> received Cisco Delete Reason vendor ID
Feb  5 07:55:37.122 09[ENC] <myconn|1> received unknown vendor ID: 43:49:53:43:4f:56:50:4e:2d:52:45:56:
Feb  5 07:55:37.122 09[ENC] <myconn|1> received unknown vendor ID: 43:49:53:43:4f:2d:44:59:4e:41:4d:49:
Feb  5 07:55:37.122 09[IKE] <myconn|1> received Cisco FlexVPN Supported vendor ID
Feb  5 07:55:37.122 09[CFG] <myconn|1> selected proposal: IKE:AES_CBC_256/HMAC_SHA2_512_256/PRF_HMAC_SHA
Feb  5 07:55:37.235 09[IKE] <myconn|1> cert payload ANY not supported - ignored
Feb  5 07:55:37.235 09[IKE] <myconn|1> authentication of '10.88.247.89' (myself) with pre-shared key
Feb  5 07:55:37.235 09[IKE] <myconn|1> establishing CHILD_SA myconn{1}
Feb  5 07:55:37.236 09[ENC] <myconn|1> generating IKE_AUTH request 1 [ IDi N(INIT_CONTACT) IDr AUTH SA
Feb  5 07:55:37.236 09[NET] <myconn|1> sending packet: from 10.88.247.89[4500] to 10.88.247.84[4500] (4
Feb  5 07:55:37.322 10[NET] <myconn|1> received packet: from 10.88.247.84[4500] to 10.88.247.89[4500] (
Feb  5 07:55:37.322 10[ENC] <myconn|1> parsed IKE_AUTH response 1 [ V IDr AUTH SA TSi TSr N(SET_WINSIZE
Feb  5 07:55:37.323 10[IKE] <myconn|1> authentication of '10.88.247.84' with pre-shared key successful
Feb  5 07:55:37.323 10[IKE] <myconn|1> IKE_SA myconn[1] established between 10.88.247.89[10.88.247.89].
Feb  5 07:55:37.323 10[IKE] <myconn|1> scheduling rekeying in 86190s
Feb  5 07:55:37.323 10[IKE] <myconn|1> maximum IKE_SA lifetime 86370s
Feb  5 07:55:37.323 10[IKE] <myconn|1> received ESP_TFC_PADDING_NOT_SUPPORTED, not using ESPv3 TFC padd
Feb  5 07:55:37.323 10[CFG] <myconn|1> selected proposal: ESP:AES_CBC_256/HMAC_SHA2_512_256/NO_EXT_SEQ
Feb  5 07:55:37.323 10[IKE] <myconn|1> CHILD_SA myconn{1} established with SPIs cfc15900_i 49f5e23c_o a
Feb  5 07:55:37.342 11[NET] <myconn|1> received packet: from 10.88.247.84[4500] to 10.88.247.89[4500] (
Feb  5 07:55:37.342 11[ENC] <myconn|1> parsed INFORMATIONAL request 0 [ CPS(SUBNET VER U_PFS) ]
Feb  5 07:55:37.342 11[IKE] <myconn|1> Processing informational configuration payload CONFIGURATION
Feb  5 07:55:37.342 11[IKE] <myconn|1> Processing information configuration payload of type CFG_SET
Feb  5 07:55:37.342 11[IKE] <myconn|1> Processing attribute INTERNAL_IP4_SUBNET
Feb  5 07:55:37.342 11[ENC] <myconn|1> generating INFORMATIONAL response 0 [ ]
Feb  5 07:55:37.342 11[NET] <myconn|1> sending packet: from 10.88.247.89[4500] to 10.88.247.84[4500] (9
```

Internal Kernel strongswan injected routes

On Linux, **strongswan** (multiplatform IPsec implementation used by NFVIS) installs routes (including BGP VPNv4 unicast routes) into routing table **220**by default and hence requires the kernel to support policy based routing.

```
nfvis# support show route 220
10.122.144.128/27 dev ipsec0 proto bird scope link
81.81.81.1 dev ipsec0 proto bird scope link
91.91.91.0/24 dev ipsec0 proto bird scope link
166.34.121.112 dev ipsec0 scope link
```

Review IPsec0 Interface Status

You can get further details about the ipsec0 virtual interface with the use of **ifconfig**

```
nfvis# support show ifconfig ipsec0
ipsec0: flags=209<UP,POINTOPOINT,RUNNING,NOARP>  mtu 9196
        inet 10.122.144.146  netmask 255.255.255.255  destination 10.122.144.146
        tunnel   txqueuelen 1000  (IPIP Tunnel)
        RX packets 5105  bytes 388266 (379.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5105  bytes 389269 (380.1 KiB)
        TX errors 1  dropped 0 overruns 0  carrier 1  collisions 0
```

Head-End (FlexVPN Server)

Review IPsec SAs Build Between Peers

From the output bellow, the encrypted tunnel is built between 10.88.247.84 throu the Virtual-Access1 interface and 10.88.247.89 for traffic that goes between networks 0.0.0.0/0 and 10.122.144.128/27; two Encapsulating Security Payload (ESP)SAs built inbound and outbound.

```
c8000v# show crypto ipsec sa

interface: Virtual-Access1
    Crypto map tag: Virtual-Access1-head-0, local addr 10.88.247.84

   protected vrf: private-vrf
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (10.122.144.128/255.255.255.224/0/0)
   current_peer 10.88.247.89 port 4500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 218, #pkts encrypt: 218, #pkts digest: 218
    #pkts decaps: 218, #pkts decrypt: 218, #pkts verify: 218
    #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

     local crypto endpt.: 10.88.247.84, remote crypto endpt.: 10.88.247.89
     plaintext mtu 1422, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1
     current outbound spi: 0xC91BCDE0(3374042592)
     PFS (Y/N): Y, DH group: group16

     inbound esp sas:
      spi: 0xB80E6942(3087952194)
        transform: esp-256-aes esp-sha512-hmac ,
        in use settings ={Tunnel, }
        conn id: 2123, flow_id: CSR:123, sibling_flags FFFFFFFF80000048, crypto map: Virtual-Access1-he
         sa timing: remaining key lifetime (k/sec): (4607969/27078)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0xC91BCDE0(3374042592)
        transform: esp-256-aes esp-sha512-hmac ,
        in use settings ={Tunnel, }
```

```
        conn id: 2124, flow_id: CSR:124, sibling_flags FFFFFFFF80000048, crypto map: Virtual-Access1-he
         sa timing: remaining key lifetime (k/sec): (4607983/27078)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     outbound ah sas:

     outbound pcp sas:
```

Display Active Crypto (encryption) Sessions

The output for the **show crypto session detail** must provide comprehensive details about each active crypto session, including the type of VPN (such as site-to-site or remote access), the encryption and hashing algorithms in use, and the security associations (SAs) for both inbound and outbound traffic. As it also displays statistics about the encrypted and decrypted traffic, such as the number of packets and bytes; this can be useful for monitoring the amount of data being secured by the VPN and for troubleshooting throughput issues.

```
c8000v# show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation
R - IKE Auto Reconnect, U - IKE Dynamic Route Update
S - SIP VPN

Interface: Virtual-Access1
Profile: uCPE-profile
Uptime: 11:39:46
Session status: UP-ACTIVE
Peer: 10.88.247.89 port 4500 fvrf: public-vrf ivrf: private-vrf
      Desc: uCPE profile
      Phase1_id: 10.88.247.89
  Session ID: 1235
  IKEv2 SA: local 10.88.247.84/4500 remote 10.88.247.89/4500 Active
          Capabilities:D connid:2 lifetime:12:20:14
  IPSEC FLOW: permit ip   0.0.0.0/0.0.0.0 10.122.144.128/255.255.255.224
       Active SAs: 2, origin: crypto map
       Inbound:  #pkts dec'ed 296 drop 0 life (KB/Sec) 4607958/7 hours, 20 mins
       Outbound: #pkts enc'ed 296 drop 0 life (KB/Sec) 4607977/7 hours, 20 mins
```

Reset VPN Connections

The **clear crypto**commands are used to manually reset VPN connections, or clearing security associations (SAs) without needing to reboot the entire device.

- **clear crypto ikev2** would clears IKEv2 security associations (IKEv2 SAs).
- **clear crypto session** would clear IKEv1 (isakmp)/IKEv2 and IPSec SAs.

- **clear crypto sa** would clear only the IPSec SAs.

- **clear crypto ipsec sa** would delete the active IPSec security associations.

Perform Debugs for Additional Troubleshooting

IKEv2 debugs can help identifying and troubleshooting errors on the head-end device (c8000v) that can occur during the IKEv2 negotiation process and FlexVPN client connections, such as problems with establishing the VPN session, policy application, or any client-specific errors.

```
c8000v# terminal no monitor
c8000v(config)# logging buffer 1000000
c8000v(config)# logging buffered debugging
c8000v# debug crypto ikev2 error
c8000v# debug crypto ikev2 internal
c8000v# debug crypto ikev2 client flexvpn
```

Related Articles and Documentation

Secure Overlay and Single IP Configuration

BGP Support on NFVIS

Secure Overlay and BGP Commands