Session Initiation Protocol Triggered VPN

Session Initiation Protocol Triggered VPN (SIP-Triggered VPN or VPN-SIP) is a service offered by service providers where a VPN is set up using Session Initiation Protocol (SIP) for on-demand media or application sharing between peers. The VPN-SIP feature defines the process in which two SIP user agents resolve each other's IP addresses, exchange the fingerprints of their self-signed certificates, third-party certificates, or pre-shared key securely, and agree to establish an IPsec-based VPN.

Service providers offer the VPN-SIP service to their customers that have SIP-based services such as bank ATMs or branches. This VPN-SIP service replaces an ISDN connection for backup network functionality. If the primary broadband service link goes down, these bank ATMs or branches connect to their central headend or data centres through the VPN-SIP service.

The SIP server of the service provider, which coordinates the VPN-SIP service, is also used for billing of the service based on the time the service is used.

- Feature Information for VPN-SIP, on page 2
- Information about VPN-SIP, on page 2
- Prerequisites for VPN-SIP, on page 6
- Restrictions for VPN-SIP, on page 6
- How to Configure VPN-SIP, on page 7
- Configuration Examples for VPN-SIP, on page 12
- Configuring DHCP in VPN-SIP, on page 13
- Troubleshooting for VPN-SIP, on page 24
- Additional References for VPN-SIP , on page 32

Feature Information for VPN-SIP

Table 1: Feature Information for VPN-SIP

Feature Name	Releases	Feature Information
Session Initiation Protocol Triggered VPN		VPN-SIP is a service offered by service providers where a VPN is setup for on-demand media or application sharing between peers, using Session Initiation Protocol (SIP). The following commands were introduced: **nat force-encap, show vpn-sip session, show vpn-sip sip, show vpn-sip registration-status, vpn-sip local-number, vpn-sip logging, vpn-sip tunnel source**.

Information about VPN-SIP

Components for VPN-SIP Solution

VPN-SIP uses IPSec Static Virtual Tunnel Interface (SVTI). IPSec SVTI stays in active (UP) state even when there is no IPSec security association (SA) established between the tunnel interface and the SVTI peer.

The following are three components for the VPN-SIP Solution:

- SIP
- VPN-SIP
- Crypto (IP Security (IPsec), Internet Key Exchange (IKE), Tunnel Protection (TP), Public Key Infrastructure (PKI) modules within crypto)

Sesssion Initiation Protocol

SIP is used as a name resolution mechanism to initiate an IKE session. VPN-SIP uses SIP service to establish a VPN connection to a home or a small business router that does not have a fixed IP address. This connection is achieved using self-signed certificates or pre-shared keys. SIP negotiates the use of IKE for media sessions in the Session Description Protocol (SDP) offer-and-answer model.

SIP is statically configured. One tunnel interface must be configured for each remote SIP number.

SIP also provides billing capabilities for service providers to charge customers based on the SIP number, for using the VPN-SIP service. Billing based on SIP numbers happens in the service provider network and is independent of the end devices like Cisco VPN-SIP routers.

VPN-SIP Solution

VPN-SIP is the central block that coordinates between SIP and Crypto modules, and provides an abstraction between them.

When traffic destined to a remote network behind a SIP number is routed to the tunnel interface, the IPSec control plane gets a trigger from packet switching path as there is no IPSEC SA configured to that peer. IPsec control plane passes the trigger to VPN-SIP as the tunnel is configured for VPN-SIP.

Note Static routes for remote networks for that SIP number must be configured to point to that tunnel interface.

When the VPN-SIP service is triggered, SIP sets up the call with a SIP phone number pair. SIP also passes incoming call details to the VPN-SIP and negotiates IKE media sessions using local address and fingerprint information of the local self-signed certificate or pre-shared key. SIP also passes remote address and fingerprint information to VPN-SIP.

The VPN-SIP service listens to tunnel status updates and invokes SIP to tear down the SIP session. The VPN-SIP service also provides a means to display current and active sessions.

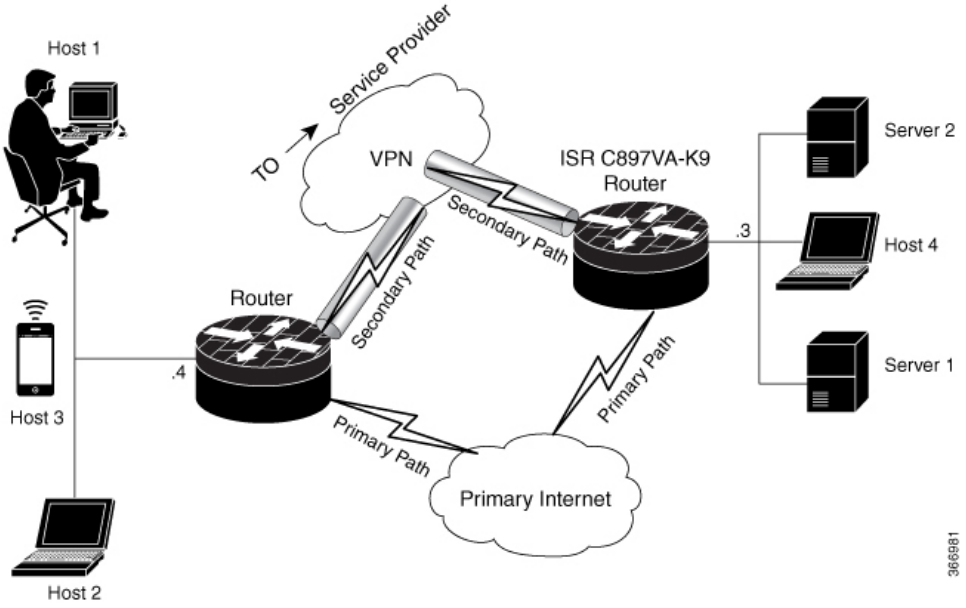
Feature at a glance

The following steps summarize how the VPN-SIP feature works:

- IP SLA monitors the primary link using route tracking. When the primary link fails IP SLA detects this failure.
- Once the primary path fails, IP SLA switches the default route to the higher metric route that is configured on the router.
- When relevant traffic tries to flow using the secondary link, SIP sends an invite message to the SIP server to obtain the VPN peer information.
- The router receives the VPN peer information (IP address, local and remote SIP numbers, IKE port, and finger print) and it establishes VPN-SIP tunnel.
- When the primary path comes back up, IP SLA detects the primary path and the route falls back to the original path. When the idle timer expires, IPSec is torn down and a SIP call is disconnected.

Following is the topology for the VPN-SIP solution:

Figure 1: VPN-SIP Topology



SIP Call Flow

The SIP call flow is divided into initiation at the local peer and call receipt at the remote peer.

At SIP Call Intitiation

When packets are routed to an SVTI interface in data plane, the SIP call must be placed to the peer SIP number to resolve its address, so that VPN tunnel can be brought up.

- When local auth-type is PSK, IKEv2 finds the matching key for a peer SIP number. The IKEv2 keyring must be configured with id_key_id type (string) as SIP number for each SIP peer. IKEv2 computes the fingerprint of the looked-up key and passes it to VPN-SIP.
- When local auth-type is a self-signed certificate or an third-party certificate, IKEv2 computes the fingerprint of the local certificate configured under the IKEv2 profile and passes it to the VPN-SIP

The VPN-SIP module interacts with SIP to setup SIP call to the peer. When the call is successful, VPN-SIP sets the tunnel destination of SVTI to the resolved IP address, requesting SVTI to initiate the VPN tunnel.

Note When a wildcard key is required, use the authentication local pre-share key command and the authentication remote pre-share key command in IKEv2 profile.

When SIP call is received at the remote peer

When a SIP call is received from a peer, following interactions occur between various crypto modules:

- The Tunnel Protection helps VPN-SIP module to set tunnel destination address.

- IKEv2 returns local auth-type (PSK or PKI) and local fingerprint to the VPN-SIP module. When local auth-type is PSK, IKEv2 finds a matching key for a corresponding SIP number.

Note IKEv2 only knows peer by its SIP number.

During the SIP call negotiation between peers, each peer must select a unique local IKEv2 port number to be exchanged over the SDP. To support different port numbers for each session, the VPN-SIP module programmatically configures IP Port Address Translation (PAT) to translate between IKEv2 port (4500) and the port number exchanged over SDP. For the translation to work IP NAT must be configured on secondary link and the loopback interface configured as the VPN-SIP tunnel source. The lifetime of the translation is limited to the lifetime of the VPN-SIP session.

SDP Offer and Answer

Following is the sample for SDP offer and answer that is negotiated in the SIP call as defined in RFC 6193:

```
offer SDP
      ...
      m=application 50001 udp ike-esp-udpencap
      c=IN IP4 10.6.6.49
      a=ike-setup:active
      a=fingerprint:SHA-1 \
      b=AS:512
      4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
      ...

   answer SDP
      ...
      m=application 50002 udp ike-esp-udpencap
      c=IN IP4 10.6.6.50
      a=ike-setup:passive
      a=fingerprint:SHA-1 \
      b=AS:512
      D2:9F:6F:1E:CD:D3:09:E8:70:65:1A:51:7C:9D:30:4F:21:E4:4A:8E
```

As part of the SDP negotiation, both peers negotiate the maximum bandwidth rate for the VPN-SIP session using the b=AS :number SDP attribute. if the peers mention different bandwidth numbers in their SDP, both of them should honor the minimum value as the maximum bandwidth. If b=AS :number SDP attribute is missing in the offer or answer, the SIP call is not successfully set up.

The negotiated maximum bandwidth is applied on the SVTI tunnel interface through the programmatically configured QoS policy in the output direction. The programmatically configured QoS policy is not applied and session fails, if there is a pre-existing statically configured policy.

Once SIP call is complete and address of the peer is resolved, VPN-SIP sets tunnel destination of SVTI and sends a request to initiate tunnel.

IKEv2 Negotiation

Following is the process for IKEv2 Security Session (SA) negotiation:

- Before starting the session, IKEv2 checks with VPN-SIP if the session is a VPN-SIP session.
- If it's a VPN-SIP session and local auth-type is PSK, IKEv2 looks up the PSK key pair using SIP number of the peer instead of IP address of the peer.

- For validating self-signed certificate, IKEv2 checks if the certificate is self-signed and validates the certificate.
    - In addition to existing AUTH payload validation as part of IKEv2 protocol, IKEv2 calculates hash of the received certificate or looked-up PSK and compares with the fingerprint from SIP negotiation that IKEv2 queries from VPN-SIP module. Only if the fingerprint matches, IKEv2 considers authentication of peer is valid. If not, IKEv2 declares that peer has failed to authenticate and fails the VPN session.

VPN-SIP solution depends on IPSEC idle timer to detect that traffic is no longer routed over the backup VPN. The idle-time configuration under the IPSec Profile is mandatory for session to be disconnected when there is no traffic. 120 seconds is the recommended time.

VPN-SIP and SIP coordinate to tear down SIP call.

When IPsec idle time expires the VPN-SIP module informs the IKEv2 to bring down the IPsec tunnel. VPN-SIP requests the SIP module to disconnect the SIP call, without waiting for confirmation from the IKEv2.

When SIP call disconnect is received from the peer, VPN-SIP module informs the IKEv2 to bring down the IPsec tunnel, and acknowledges to SIP to tear down the SIP call.

Prerequisites for VPN-SIP

- Security K9 license must be enabled on the router.
- The routers must have a minimum memory of 1 GB.
- For the SIP register request of the SIP User Agent to succeed, the SIP registrar must be available to the VPN-SIP routers.
- The DHCP server must support option 120 and 125 to obtain the SIP server address, which is needed for registration and establishing the SIP session.
- Proper routing configurations must be completed to ensure backup WAN path is used when primary path is down.
- Maximum Transmission Unit (MTU) of the tunnel interface must be less than the MTU of the secondary WAN interface.
- When self-signed or third-party certificates are used for IKEv2 authentication, configure IKEv2 fragmentation on the VPN-SIP router to avoid fragmentation at the IP layer.
- NAT SIP ALG must be disabled.
- Caller ID notification service must be configured in the network.

Restrictions for VPN-SIP

- VPN-SIP and CUBE/SIP gateway cannot be configured on the same device. When CUBE license is active on the device, only CUBE will be functional.
- Only IPv4 is supported for transport and media (IPv4 transport for SIP registration, SIP signaling, and IPv4 packets encrypted over IPv4 transport).

- SIP signalling with peer devices behind NAT is not supported (ICE and STUN are not supported.
- SIP negotiation is supported only in global VRF.
- Remote-access VPN features like private address assignment, configuration mode exchange (CP payloads), routes exchange, are not supported.
- Routing protocols over the VPN-SIP session are not supported.
- Only Rivest-Shamir-Addleman (RSA) server self-signed certificates are supported.
- Pre-shared key lookup functionality using authentication, authorization, and accounting (AAA) is not supported.
- The IPSec idle timer is configured per IPSec profile using the ipsec-profile command. The idle time is the same for all VPN-SIP sessions that use a specific IPSec profile.
- Track objects that are used for IPSLA monitoring, have a maximum limit of 1000 objects in Cisco IOS software. When one track object is used to track one peer router, maximum number of VPN-SIP sessions that one IOS device can have is limited by the maximum number of track objects.
- Only one local SIP number is supported on Cisco IOS software.
- If there is a pre-existing statically configured policy, the programmatically configured QoS policy is not applied and session fails. Remove any statically configured QoS policy on the SVTI interface.
- Cisco does not support the interoperability with VPN-SIP implementation of other vendors.
- For the class policies included in the `policy-map` attached to the VPN-SIP tunnel, only Priority Queueing and Class-Based Weighted Fair Queueing (CBWFQ) are supported.
- For CBWFQ configurations, only the `bandwidth percent` *`percent`* command is supported. The `bandwidth` *`bandwidth`* command is not supported as the bandwidth of the VPN-SIP session varies depending on the negotiation with the peer router.
- VPN-SIP configuration is not supported on IPv6.
- VPN-SIP configuration is supported only in autonomous mode.
- Complex SIP call scenarios such as refer, fork etc. are not supported in VPN-SIP configuration.

How to Configure VPN-SIP

Configuring VPN-SIP

The following steps describe the process of configuring VPN-SIP:

1. Configure the tunnel authentication using third party certificates, self-signed certificates, or pre-shared keys.
   a. Tunnel Authentication using Certificates

      Configure a trustpoint to obtain a certificate from a certification authority (CA) server that is located in the customer's network. This is required for tunnel authentication. Use the following configuration:

```
peer1(config)# crypto pki trustpoint CA
 enrollment url http://10.45.18.132/
 serial-number none
 subject-name CN=peer2
 revocation-check crl
 rsakeypair peer2

peer2(config)# crypto pki authenticate CA
Certificate has the following attributes:
       Fingerprint MD5: F38A9B4C 2D80490C F8E7581B BABE7CBD
      Fingerprint SHA1: 4907CC36 B1957258 5DFE23B2 649E7DDA 99BDB7C3
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.

peer2(config)#crypto pki enroll CA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: CN=peer2
% The subject name in the certificate will include: peer2
% Include an IP address in the subject name? [no]:
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose CA' command will show the fingerprint.
Certificate map for Trustpoint
crypto pki certificate map data 1
issuer-name co cn = orange
```

b. Tunnel authentication using self-signed certificate

Configure a PKI trust point to generate a self-signed certificate on the device, when authenticating using a self-signed certificate. Use the following configuration:

```
peer4(config)#crypto pki trustpoint Self
    enrollment selfsigned
    revocation-check none
    rsakeypair myRSA
    exit
crypto pki enroll Self

Do you want to continue generating a new Self Signed Certificate? [yes/no]: yes
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes

Router Self Signed Certificate successfully created
```

c. Configure tunnel authentication using a pre-shared key

```
crypto ikev2 keyring keys
peer peer1
identity key-id 1234
pre-shared-key key123
```

2. **a.** Configure IKEv2 Profile for Certificate

```
crypto ikev2 profile IPROF
match certificate data
identity local key-id 5678
authentication remote rsa-sig
```

```
authentication local rsa-sig
keyring local keys
pki trustpoint self
nat force-encap
```

b. Configure an IKEv2 Profile for pre-shared keys

```
crypto ikev2 profile IPROF
match identity remote any
identity local key-id 5678
authentication remote pre-share
authentication local pre-share
keyring local keys
nat force-encap
```

Note To complete the IKEv2 SA configuration, the **nat force-encap** command must be configured on both peers. Since, UDP encapsulation is negotiated in SDP, IKEv2 must start and continue on port 4500.

3. Configure an IPsec profile

```
crypto ipsec profile IPROF
set security-association idle-time 2000
```

4. Configure a LAN side interface

```
interface Vlan101
         ip address 10.3.3.3 255.255.255.0
         no shutdown
!
     interface GigabitEthernet2
         switchport access vlan 101
         no ip address
```

5. Configure a loopback interface

The loopback interface is used as the source interface for the secondary VPN tunnel.

```
interface loopback 1
    ip address 10.11.1.1 255.0.0.0
    ip nat inside
```

6. Configure a secondary interface.

Note Make sure the secondary interface is configured to receive the IP address, SIP server address, and vendor specific information via DHCP.

```
interface GigabitEthernet8
     ip dhcp client request sip-server-address
     ip dhcp client request vendor-identifying-specific
     ip address dhcp
    ip nat outside
```

7. Configure the tunnel interface

```
interface Tunnel1
     ip address 10.3.2.1 255.255.255.255
     load-interval 30
     tunnel source Loopback1
```

```
tunnel mode ipsec ipv4
tunnel destination dynamic
tunnel protection ipsec profile IPROF ikev2-profile IPROF
vpn-sip local-number 5678 remote-number 1234 bandwidth 1000
```

Use the **vpn-sip local-number** *local-number* **remote-number** *remote-number* **bandwidth** *bw-number* command to configure the sVTI interface for VPN-SIP. Bandwidth is the maximum data transmission rate that must be negotiated with this peer and the negotiated value is set on the tunnel interface. Allowed values are 64, 512, and 1000 kbps.

Once an SVTI is configured for VPN-SIP, changes cannot be made to tunnel mode, tunnel destination, tunnel source, and tunnel protection. To change the mode, source, destination, or tunnel protection you must remove the VPN-SIP configuration from the SVTI interface.

8. Add static routes to destination networks

Add a secondary route with a higher metric.

```
ip route 192.168.10.0 255.255.255.0 Tunnel0 track 1
ip route 192.168.10.0 255.255.255.0 Tunnel1 254
```

9. Configure IP SLA

```
ip sla 1
         icmp-echo 10.11.11.1
         threshold 500
         timeout 500
         frequency 2
        ip sla schedule 1 life forever start-time now
```

10. Configure route tracking

```
track 1 ip sla 1 reachability
```

11. Enable VPN-SIP

```
vpn-sip enable
vpn-sip local-number 5678 address ipv4 GigabitEthernet8
vpn-sip tunnel source Loopback1
vpn-sip logging
```

To configure VPN-SIP, you must configure local SIP number and local address. The **vpn-sip local-number** *SIP-number* **address ipv4** *WAN-interface-name* command configures the local SIP number that is used for SIP call and the associated IPv4 address.

Note Only IPv4 addresses can be configured. Crypto module does not support dual stack.

- Backup WAN interface address may change based on DHCP assignment.

When the primary WAN interface is functional, the destination of the VPN-SIP tunnel is set to the backup WAN interface, so that the tunnel interface is active. Destination is set to IP address of the peer that is learnt from SDP of SIP negotiation when traffic is routed to the tunnel interface. When primary WAN interface fails and the back routes are activated, packets are routed to the sVTI through backup.

Note We recommend that you use an unused non-routable address as the address of the loopback interface and do not configure this loopback interface for any other purpose. Once a loopback interface is configured, VPN-SIP listens to any updates to the interface and blocks them. The **vpn-sip logging** command enables the system logging of VPN-SIP module for events, such as session up, down, or failure.

Verifying VPN-SIP on a Local Router

Verifying Registration Status

```
Peer1# show vpn-sip registration-status
 SIP registration of local number 0388881001 : registered 10.6.6.50
```

Verifying SIP Registrar

```
Peer1#show vpn-sip sip registrar

Line          destination      expires(sec)  contact   transport     call-id
=============================================================================================================
0388881001    example.com         2359        10.6.6.50  UDP
3176F988-9EAA11E7-8002AFA0-8EF41435
```

Verifying VPN-SIP Status

```
Peer1#show vpn-sip session detail
VPN-SIP session current status

Interface: Tunnel1
   Session status: SESSION_UP (I)
   Uptime        : 00:00:42
   Remote number : 0388881001 =====> This is the Remote Router's SIP number
   Local number  : 0388882001 =====> Local router's SIP number
   Remote address:port: 10.6.6.49:50002
   Local address:port : 10.6.6.50:50001
   Crypto conn handle: 0x8000017D
   SIP Handle        : 0x800000C7
   SIP callID        : 1554
   Configured/Negotiated bandwidth: 64/64 kbps
```

Verifying Crypto Session

```
Peer1# 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-sip

Interface: Tunnel1
Profile: IPROF
Uptime: 00:03:53
Session status: UP-ACTIVE
Peer: 10.6.6.49 port 4500 fvrf: (none) ivrf: (none)
      Phase1_id: 10.6.6.49
      Desc: (none)
```

```
 Session ID: 43
 IKEv2 SA: local 10.11.1.1/4500 remote 10.6.6.49/50002 Active
         Capabilities:S connid:1 lifetime:23:56:07 ====> Capabilities:S indicates this is
a  SIP VPN_SIP Session
 IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
       Active SAs: 2, origin: crypto map
       Inbound:  #pkts dec'ed 6 drop 0 life (KB/Sec) 4222536/3366
       Outbound: #pkts enc'ed 4 drop 0 life (KB/Sec) 4222537/3366
```

Verifying IP NAT Translations

```
Peer1#sh ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 2.2.2.2:4500       10.6.6.50:50001    10.6.6.49:50002    10.6.6.49:50002
```

Verifying DHCP SIP Configuration

```
Peer9#show vpn-sip sip dhcp
SIP DHCP Info

SIP-DHCP interface:  GigabitEthernet8

SIP server address:
Domain name:         dns:example.com
```

Configuration Examples for VPN-SIP

Using self-signed certificates for authentication

The following is sample configuration to configure VPN-SIP using self-signed certificates for authentication. There is no distinction between initiator and responder role in VPN-SIP. The configuration on a peer node will be identical with local SIP numbers changed.

```
// Self-signed certificate
crypto pki trustpoint selfCert
  rsakeypair myRSA
  enrollment selfsigned
  revocation-check none
!
crypto ikev2 profile vpn-sip-profile
 match identity remote any
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint selfCert // Use same self-signed trustpoint for sign and verify
 nat force-encap
!
crypto ipsec profile vpn-sip-ipsec
 set security-association idle-time 120
!
vpn-sip enable
vpn-sip local-number 0388883001 address ipv4 GigabitEthernet1
vpn-sip tunnel source Loopback11
vpn-sip logging
!
// one tunnel per peer - configuration is for peer with a SIP-number of 0388884001
int tunnel0
 ip unnumbered loopback 0
 tunnel source loopback11
 tunnel mode ipsec ipv4
```

```
 tunnel destination dynamic
 tunnel protection ipsec profile vpn-sip-ipsec ikev2-profile vpn-sip-profile
 vpn-sip local-number 0388883001 remote-number 0388884001 bandwidth 1000
!
// ip unnumbered of tunnel interfaces
int loopback 0
  ip address 10.21.1.1 255.255.255.255
!
int loopback11
ip address 10.9.9.9 255.255.255.255
ip nat inside
!
// one tunnel per peer - this is for peer with SIP-number 0388885001
int tunnel1
 ip unnumbered loopback 0
 tunnel source loopback11
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile vpn-sip-ipsec ikev2-profile iprof
 vpn-sip sip-local 0388883001 sip-remote 0388885001 bandwidth 1000
!
interface GigabitEthernet8
 ip dhcp client request sip-server-address
 ip dhcp client request vendor-identifying-specific
 ip address dhcp
ip nat outside


// backup routes configured with higher AD so that these routes will be activated only when
 primary path goes down. AD need to be chosen to be greater than that of primary route.
ip route 10.0.0.0 255.0.0.0 tunnel 0 250
ip route 10.1.0.0 255.0.0.0 tunnel 0 250
ip route 10.2.0.0 255.0.0.0 tunnel 0 250
ip route 10.3.0.0 255.0.0.0 tunnel 0 250
```

Configuring DHCP in VPN-SIP

Configure DHCP for VPN-SIP

From Cisco IOS XE Release 17.11.1a, you can install a VPN-SIP-enabled router behind a home gateway (HGW). In this installation, the HGW assigns an extension number to the tunnel interface through the Dynamic Host Configuration Protocol (DHCP) instead of a fixed telephone number. This allows you to aggregate data and voice on your network, which can be useful in scenarios where you need to share the same physical subscriber line for both analog and digital data.

In addition, to comply with the HGW network specifications, DHCP for VPN-SIP requires the MAC address of the WAN-side interface to the HGW network through the vendor-class-data DHCP option. With this configuration, the device communicates the MAC address of its own WAN interface to the home gateway network through the vendor-class-data option of the DHCP requests.

Supported PIDs and Firmware

The following table specifies the HGW PIDs and the firmware versions that are tested. Cisco does not provide support for the HGW installed at a customer's location or the operation of an HGW. We recommend that you verify your environment before using this feature.

HGW PID	Firmware Version
RT-400NE	8.06
RT-400MI	09.00.0015
RT-400KI	08.00.0040
RT-500MI	08.00.0004
RT-500KI	08.00.0020
RX-600MI	01.00.0001
RX-600KI	01.00.0001
OG410Xi	2.32
OG410Xa	2.32

Configure DHCP for VPN-SIP

When you configure a DHCP local number, the device defers SIP registration until it receives a DHCP response. The device expects the DHCP server to provide an extension number. This extension number is then used to register with the SIP server. On successful registration, the device initiates a session with the SIP server and receives an extension number, an external number, and other available numbers through a 200 OK response.

Note The external number is the number with which the router is identified globally. This external number is also required to establish a data connection.

With the DHCP enhancement, there are two channels for data connection—SIP signalling channel and IPsec data connection. If the data packets require tunnel protection, a SIP call is initiated.

Perform the following procedures to configure DHCP for VPN-SIP.

Enable the DHCP Client

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type number**
4. **ip dhcp client request sip-server-address**
5. **ip dhcp client request vendor-identifying-specific**
6. **ip address dhcp**
7. **ip dhcp client vendor-class mac-address**
8. **ip nat outside**

9. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface type number** **Example:** `Router(config)# interface gigabitethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip dhcp client request sip-server-address** **Example:** `Router(config-if)# ip dhcp client request sip-server-address`	Configures the DHCP client to request a SIP server address from a DHCP server.
Step 5	**ip dhcp client request vendor-identifying-specific** **Example:** `Router(config-if)# ip dhcp client request vendor-identifying-specific`	Configures the DHCP client to request vendor-specific information from a DHCP server.
Step 6	**ip address dhcp** **Example:** `Router(config-if)# ip address dhcp`	Acquires an IP address on the interface from the DHCP.
Step 7	**ip dhcp client vendor-class mac-address** **Example:** `Router(config-if)# ip dhcp client vendor-class mac-address`	Complies with the HGW's DHCP specification.
Step 8	**ip nat outside** **Example:** `Router(config-if)# ip nat outside`	Connects the interface to the outside network.
Step 9	**exit** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enable DHCP Client Sample Configuration

The following is a sample code for enabling a DHCP client.

```
interface GigabitEthernet 0/0/0
 ip dhcp client request sip-server-address
 ip dhcp client request vendor-identifying-specific
 ip address dhcp
 ip dhcp client vendor-class mac-address
 ip nat outside
```

Configure Tunnel Authentication

You can configure tunnel authentication by using third-party certificates, self-signed certificates, or by using preshared keys (PSKs). To configure tunnel authentication, perform one of the following tasks.

Configure Tunnel Authentication Using Certificates

Configure a trustpoint to obtain a certificate from a certification authority (CA) server that is located in the customer's network. This is required for tunnel authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** name
4. **enrollment url** `url`
5. **serial-number**
6. **subject-name** `[subject-name]`
7. **revocation-check crl**
8. **rsakeypair**
9. **crypto pki authenticate CA**
10. **crypto pki enroll CA** `name`
11. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto pki trustpoint** name **Example:** `Router(config)# crypto pki trustpoint CA`	Specifies the trustpoint and a given name, and enters the ca-trustpoint configuration mode.
Step 4	**enrollment url** `url` **Example:** `Router(ca-trustpoint)# enrollment url http://10.45.18.132/`	Specifies the URL of the CA to which your router should send certificate requests.
Step 5	**serial-number** **Example:** `Router(ca-trustpoint)# serial-number`	Specifies the serial number of the router in the certificate request, unless you use the **none** keyword. Use the **none** keyword if you don't want to include a serial number in the certificate request.
Step 6	**subject-name** `[subject-name]` **Example:** `Router(ca-trustpoint)# subject-name CN=peer2`	Specifies the requested subject name that is used in the certificate request. If you don’t specify the subject name, the fully qualified domain name (FQDN), which is the default subject name, is used.
Step 7	**revocation-check crl** **Example:** `Router(ca-trustpoint)# revocation-check crl`	Checks the validity of the certificate through the Certificate Revocation Lists (CRL) mechanism.
Step 8	**rsakeypair** **Example:** `Router (ca-trustoint)# rsakeypair peer2`	Provides a key pair for the trustpoint.
Step 9	**crypto pki authenticate CA** **Example:** `Router(config)# crypto pki authenticate CA`	Authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA.
Step 10	**crypto pki enroll CA** `name` **Example:** `Router(config)# crypto pki enroll CA`	Generates the certificate request and displays the request for copying and pasting into the certificate server. You are prompted for enrollment information such as whether to include the router FQDN and IP address in the certificate request. You are also given a choice about displaying the certificate request on the console terminal.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.

Example: Configuring Tunnel Authentication Using Certificates

This is the sample code for configuring tunnel authentication using certificates.

```
peer1(config)# crypto pki trustpoint CA
 enrollment url http://10.45.18.132/
 serial-number none
 subject-name CN=peer2
 revocation-check crl
 rsakeypair peer2

peer2(config)# crypto pki authenticate CA
Certificate has the following attributes:
Fingerprint MD5: F38A9B4C 2D80490C F8E7581B BABE7CBD
Fingerprint SHA1: 4907CC36 B1957258 5DFE23B2 649E7DDA 99BDB7C3
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.

peer2(config)#crypto pki enroll CA
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA
Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration. Please make a
note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: CN=peer2
% The subject name in the certificate will include: peer2
% Include an IP address in the subject name? [no]:
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose CA' command will show the fingerprint.
Certificate map for Trustpoint
crypto pki certificate map data 1
issuer-name co cn = orange
```

Configure Tunnel Authentication Using Self-Signed Certificates

To configure tunnel authentication using a self-signed certificate, run the **crypto pki trustpoint self** command. This command enables you to configure a PKI trust point to generate a self-signed certificate on the device.

```
Router(config)# crypto pki trustpoint self
enrollment self signed
revocation-check none
rsakeypair myRSA
exit

crypto pki enroll self
Do you want to continue generating a new Self Signed Certificate? [yes/no]: yes
% Include the router serial number in the subject name? [yes/no]: yes
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes
Router Self Signed Certificate successfully created
```

Configure Tunnel Authentication Using PreShared Keys

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 keyring keyring-name**
4. **peer name**
5. **address {ipv4-address [mask] | ipv6-address prefix}**
6. **identity {address { ipv4-address | ipv6-address} | fqdn name | email email-id | key-id key-id}**

7. **pre-shared-key {local| remote} {0| 6| line}**
8. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 keyring keyring-name** **Example:** `Router(config)# crypto ikev2 keyring kyr1`	Defines an IKEv2 keyring, and enters IKEv2 keyring configuration mode.
Step 4	**peer name** **Example:** `Router(config-ikev2-keyring)# peer peer1`	Defines the peer or peer group, and enters IKEv2 keyring peer configuration mode.
Step 5	**address {ipv4-address [mask] | ipv6-address prefix}** **Example:** `Router(config-ikev2-keyring-peer)# address 10.0.0.1 255.255.255.0`	Specifies an IP address or a range for the peer. This IP address is the IKE endpoint address and is independent of the identity address.
Step 6	**identity {address { ipv4-address | ipv6-address} | fqdn name | email email-id | key-id key-id}** **Example:** `Router(config-ikev2-keyring-peer)# identity key-id 1234`	Identifies the IKEv2 peer through the following identities: • E-mail • FQDN • IPv4 address • Key ID The identity is available for key lookup on the IKEv2 responder only.
Step 7	**pre-shared-key {local| remote} {0| 6| line}** **Example:** `Router(config-ikev2-keyring-peer)# pre-shared-key key123`	Specifies the PSK for the peer. Enter the **local** or the **remote** keyword to specify an asymmetric PSK. By default, the PSK is symmetric.
Step 8	**exit** **Example:**	Exits keyring peer configuration mode mode, and returns to configuration mode.

	Command or Action	Purpose
	Router(config-ikev2-keyring-peer)# end	

Example: Configure Tunnel Authentication Using PreShared Keys

This is a sample code for configuring tunnel authentication using preshared keys

```
crypto ikev2 keyring keys
 peer p1
 identity key-id 0388881001
 pre-shared-key cisco
 !
 peer p2
 identity key-id 0388882002
 pre-shared-key cisco
!
crypto ikev2 keyring HUB-KEY
peer SPOKES
address 0.0.0.0 0.0.0.0
pre-shared-key cisco
```

Configure the IKEv2 Profile for a Certificate

To configure the certificate for your IKEv2 profile, run the **crypto ikev2 profile IPROF** command. The following is a sample code for configuring the IKEv2 profile for a certificate.

```
Router(config)# crypto ikev2 profile IPROF-psk
 match identity remote any
 identity local key-id dhcp
 authentication remote pre-share
 authentication local pre-share
 keyring local keys
 nat force-encap
```

Configure an IPSec Profile

To configure an IPSec profile, run the **crypto ipsec profile IPROF** command. The following is a sample code for configuring an IPSec profile.

```
Router(config)# crypto ipsec profile IPROF
 set security-association idle-time 300
```

Enable VPN-SIP

To enable the VPN-SIP feature, run the **vpn-sip enable** command. The following is a sample code to enable VPN-SIP.

```
Router(config)# vpn-sip enable
vpn-sip local-number dhcp address ipv4 GigabitEthernet0/0/0
vpn-sip tunnel source Loopback1
```

Configure a LAN Side Interface

To configure a LAN side interface, run the **interface VLAN <interface>** command. The following is a sample code to configure a LAN side interface.

```
Router(config)# interface GigabitEthernet2
ip address 192.0.2.3 255.255.255.0
no shutdown
```

Configure a Loopback Interface

To configure a loopback interface, run the **interface loopback** <number> command. The following is a code sample to configure a loopback interface.

```
Router(config)# interface Loopback1
 ip address 10.255.255.3 255.255.255.0
 ip nat inside
```

Configure a Tunnel Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel source** {*ip-address* | *interface-type number*}
5. **tunnel destination**
6. **tunnel protection IPsec profile** *name*
7. **vpn-sip local-number dhcp remote-number bandwidth**
8. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel1`	Configures a tunnel interface and enters the interface configuration mode. The *number* argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces that you can create.
Step 4	**tunnel source** {*ip-address* | *interface-type number*} **Example:**	Sets the source IP address or the source interface type number for a tunnel interface. Since the **tunnel protection IPsec profile** command is also used in this procedure, the

	Command or Action	Purpose
	`Router(config-if)# ip address 12.12.12.12 255.255.255.255` `tunnel source Loopback1`	tunnel source must specify an interface and not an IP address.
Step 5	**tunnel destination** **Example:** `Router(config-if)# tunnel destination destination dynamic`	Specifies the destination of the tunnel.
Step 6	**tunnel protection IPsec profile** *name* **Example:** `Router(config-if)# tunnel protection ipsec profile IPROF ikev2-profile IPROF-psk`	Associates a tunnel interface with an IPsec profile. The *name* argument specifies the name of the IPsec profile. This value must match the name specified in the **crypto IPsec profile** <*name*>command.
Step 7	**vpn-sip local-number dhcp remote-number bandwidth** **Example:** `Router(config-if)# vpn-sip local-number dhcp remote-number 0388881001 bandwidth 1000`	Configures the interface for VPN-SIP. Bandwidth is the maximum data transmission rate that must be negotiated with this peer; the negotiated value is set on the tunnel interface. Choose one of these values—64, 128, 256, 512, or 1000 kbps. **Note** After you configure an interface for VPN-SIP, you cannot make any changes to the tunnel mode, tunnel destination, tunnel source, and tunnel protection. To change the mode, source, destination, or tunnel protection, you must remove the VPN-SIP configuration from the interface.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.

Example: Configure a Tunnel Interface

This is a sample code to configure a tunnel interface.

```
Router(config)# interface Tunnel1
 ip address 10.12.12.12 255.255.255.255
 tunnel source Loopback1
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile IPROF ikev2-profile IPROF-psk
 vpn-sip local-number dhcp remote-number 0388881001 bandwidth 1000
!
interface Tunnel10
 ip address 10.20.20.21 255.255.255.255
 tunnel source Loopback1
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile IPROF ikev2-profile IPROF-psk
 vpn-sip local-number dhcp remote-number 0388882002 bandwidth 100
```

Verify the DHCP Configuration in VPN-SIP

The following show command outputs indicate how to verify whether the DHCP in VPN-SIP is successfully configured on the Cisco IOS XE router behind the HGW.

```
Router_behind_HGW# show vpn-sip sip dhcp
SIP DHCP Info
SIP-DHCP interface:  GigabitEthernet 0/0/0
SIP server address:  ipv4:192.168.1.1
Domain name:         dns:ntt-east.ne.jp
```

```
Router_behind_HGW# show vpn-sip registration-status
  SIP registration of local number dhcp : registered 192.168.1.200
  Local dynamic number via dhcp[3], via SIP[0398765432]
```

```
Router_behind_HGW# show vpn-sip sip registrar
Line          destination      expires(sec)  contact
transport     call-id
============================================================
3             ntt-east.ne.jp   2439          192.168.1.20
UDP           FFFFFFFFCCE6C415-5D8611ED-FFFFFFFF810AE9D4-FFFFFFFFD
```

```
Router_behind_HGW# show vpn-sip session detail
VPN-SIP session current status
Interface: Tunnel0
    Session status: SESSION_UP (I)
    Uptime        : 00:00:37
    Remote number : 0387654321
    Local number  : dhcp
    Remote address:port: aaa.bbb.ccc.ddd:27129
    Local address:port : 192.168.1.200:50026
    Crypto conn handle: 0x4000003D
    SIP Handle        : 0x4000001B
    SIP callID        : 301
    Configured/Negotiated bandwidth: 256/256 kbps
    Applied service policy:
```

```
Router_behind_HGW# show crypto session
Crypto session current status
Interface: Tunnel0
Profile: IPROF
Session status: UP-ACTIVE
Peer: aaa.bbb.ccc.ddd port 27129
   Session ID: 26
   IKEv2 SA: local 10.255.255.1/4500 remote aaa.bbb.ccc.ddd/27129 Active
   IPSEC FLOW: permit ip   0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
   Active SAs: 2, origin: crypto map
```

```
Router_behind_HGW# show crypto ikev2 sa
  IPv4 Crypto IKEv2  SA

Tunnel-id Local                 Remote                fvrf/ivrf
   Status
1         10.255.255.1/4500     aaa.bbb.ccc.ddd/27129  none/none
    READY
       Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH
       Grp:19, Auth sign: PSK, Auth verify: PSK
       Life/Active Time: 86400/86 sec
       CE id: 1022, Session-id: 22
       Local spi: 59E8EED28441BC32
       Remote spi: B5487716A19873BE
       IPv6 Crypto IKEv2  SA
```

```
Router_behind_HGW# show crypto ipsec sa

interface: Tunnel0
```

```
Crypto map tag: Tunnel0-head-0, local addr 10.255.255.1
protected vrf: (none)
local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
current_peer aaa.bbb.ccc.ddd port 27129
PERMIT, flags={origin_is_acl,}
     #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
     #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
     #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.: 10.255.255.1, remote crypto endpt.:
aaa.bbb.ccc.ddd
plaintext mtu 1422, path mtu 1500, ip mtu 1500, ip mtu idb
GigabitEthernet0/0/0
current outbound spi: 0xE0F51D37(3774160183)
PFS (Y/N): N, DH group: none

inbound esp sas:
      spi: 0x493D896(76798102)
      transform: esp-aes esp-sha-hmac ,
      in use settings ={Tunnel UDP-Encaps, }
      conn id: 2044, flow_id: ESG:44, sibling_flags FFFFFFFF80004048,
crypto map: Tunnel0-head-0, initiator : True
       sa timing: remaining key lifetime (k/sec): (4607999/3509)
       IV size: 16 bytes
       replay detection support: Y
       Status: ACTIVE(ACTIVE)

      inbound ah sas:

      inbound pcp sas:

      outbound esp sas:
       spi: 0xE0F51D37(3774160183)
         transform: esp-aes esp-sha-hmac ,
         in use settings ={Tunnel UDP-Encaps, }
         conn id: 2043, flow_id: ESG:43, sibling_flags FFFFFFFF80004048,
crypto map: Tunnel0-head-0, initiator : True
         sa timing: remaining key lifetime (k/sec): (4607999/3509)
         IV size: 16 bytes
         replay detection support: Y
         Status: ACTIVE(ACTIVE)
      outbound ah sas:
      outbound pcp sas:


Router_behind_HGW# show ip nat translations
Pro  Inside global         Inside local          Outside local
Outside global
udp  192.168.1.200:50269   10.255.255.1:4500     aaa.bbb.ccc.ddd:23060
aaa.bbb.ccc.ddd:23060
Total number of translations: 1
```

Troubleshooting for VPN-SIP

Viewing Tunnel Interface in Show Output

Symptom

Show VPN-SIP session doesn't show any information about the tunnel interface. In the following example, information about the tunnel interface, tunnel1 is not shown:

```
Peer5-F#show vpn-sip session
VPN-SIP session current status

Interface: Tunnel2
   Session status: READY_TO_CONNECT
   Remote number : 0334563333
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 192.30.18.22:0

Interface: Tunnel3
   Session status: READY_TO_CONNECT
   Remote number : 0323452222
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 192.30.18.22:0

Interface: Tunnel4
   Session status: READY_TO_CONNECT
   Remote number : 0612349999
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 192.30.18.22:0

Interface: Tunnel6
   Session status: READY_TO_CONNECT
   Remote number : 0634567777
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
```

Possible Cause

VPN-SIP is not configured on the tunnel interface

```
Peer5-F#sh run int tun1
Building configuration...

Current configuration : 201 bytes
!
interface Tunnel1
 ip address 10.5.5.5 255.0.0.0
 tunnel source Loopback11
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile test-prof ikev2-profile test
end
```

Recommended Action

Configure VPN-SIP on the tunnel interface.

```
:

Peer5-F#show running interface tunnel 1
Building configuration...

Current configuration : 278 bytes
!
interface Tunnel1
 ip address 10.5.5.5 255.255.255.255
 tunnel source Loopback11
```

```
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile test-prof ikev2-profile test
 vpn-sip local-number 0623458888 remote-number 0312341111 bandwidth 1000
end
```

Following is the running output for the above scenario:

```
Peer5-F#show vpn-sip session detail
VPN-SIP session current status

Interface: Tunnel1
   Session status: READY_TO_CONNECT
   Remote number : 0312341111
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0

   Crypto conn handle: 0x8000002C
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 1000/0 kbps

Interface: Tunnel2
   Session status: READY_TO_CONNECT
   Remote number : 0334563333
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
   Crypto conn handle: 0x80000012
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 512/0 kbps

Interface: Tunnel3
   Session status: READY_TO_CONNECT
   Remote number : 0323452222
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
   Crypto conn handle: 0x80000031
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 512/0 kbps

Interface: Tunnel4
   Session status: READY_TO_CONNECT
   Remote number : 0612349999
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
   Crypto conn handle: 0x8000002F
   SIP Handle        : 0x0
   SIP callID        : --
   Configured/Negotiated bandwidth: 1000/0 kbps

Interface: Tunnel6
   Session status: READY_TO_CONNECT
   Remote number : 0634567777
   Local number  : 0623458888
   Remote address:port: 0.0.0.0:0
   Local address:port : 172.30.18.22:0
   Crypto conn handle: 0x80000026
   SIP Handle        : 0x0
```

```
   SIP callID        : --
   Configured/Negotiated bandwidth: 1000/0 kbps
```

Troubleshooting SIP Registration Status

Symptom

SIP registration status is Not Registered

```
Peer5#show vpn-sip sip registrar
Line          destination      expires(sec)  contact
transport     call-id
============================================================

Peer5-F#show vpn-sip registration-status

 SIP registration of local number 0623458888 : not registered
```

Possible Cause

IP address is not configured on the WAN interface.

```
Peer5#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES unset  down                  down
GigabitEthernet0/1         unassigned      YES unset  up                    up
GigabitEthernet0/2         unassigned      YES unset  down                  down
GigabitEthernet0/3         unassigned      YES unset  down                  down
GigabitEthernet0/4         unassigned      YES unset  up                    up
GigabitEthernet0/5         10.5.5.5        YES manual up                    up
Vlan1                      10.45.1.5        YES NVRAM  up                     up
NVI0                       10.1.1.1         YES unset  up                     up
Loopback1                  10.1.1.1         YES NVRAM  up                     up
Loopback5                  10.5.5.5        YES NVRAM  administratively down down
Loopback11                 10.11.11.11     YES NVRAM  up                    up
Tunnel1                    10.5.5.5         YES NVRAM  up                     down
Tunnel2                    10.2.2.2         YES NVRAM  up                     down
Tunnel3                    10.3.3.3         YES NVRAM  up                     down
Tunnel4                    10.4.4.4         YES NVRAM  up                     down
Tunnel6                    10.8.8.8         YES NVRAM  up                     down

Peer5-F#show run interface gigabitEthernet 0/4
Building configuration...

Current configuration : 213 bytes
!
interface GigabitEthernet0/4
 ip dhcp client request sip-server-address
 ip dhcp client request vendor-identifying-specific
 no ip address           ====> no IP address
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
end
```

Recommended Action

Use the **ip address dhcp** command to configure the interface IP address.

```
Peer5-F#show running-config interface gigabitEthernet 0/4
Building configuration...

Current configuration : 215 bytes
```

```
!
interface GigabitEthernet0/4
 ip dhcp client request sip-server-address
 ip dhcp client request vendor-identifying-specific
 ip address dhcp         ====> configure IP address DHCP
 ip nat outside
 ip virtual-reassembly in
 duplex auto
 speed auto
end


Peer5-F#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES unset  down                  down
GigabitEthernet0/1         unassigned      YES unset  up                    up
GigabitEthernet0/2         unassigned      YES unset  down                  down
GigabitEthernet0/3         unassigned      YES unset  down                  down
GigabitEthernet0/4         172.30.18.22    YES DHCP   up                    up
GigabitEthernet0/5         10.5.5.5        YES manual up                    up
Vlan1                      10.45.1.5        YES NVRAM  up                     up
NVI0                       10.1.1.1         YES unset  up                     up
Loopback1                  10.1.1.1         YES NVRAM  up                     up
Loopback5                  10.5.5.5        YES NVRAM  administratively down down
Loopback11                 10.11.11.11     YES NVRAM  up                    up
Tunnel1                    10.6.5.5         YES NVRAM  up                     down
Tunnel2                    10.2.2.2         YES NVRAM  up                     down
Tunnel3                    10.3.3.3         YES NVRAM  up                     down
Tunnel4                    10.4.4.4         YES NVRAM  up                     down
Tunnel6                    10.8.8.8         YES NVRAM  up                     down

Peer5-F#show vpn-sip sip registrar
Line          destination      expires(sec)  contact
transport     call-id
============================================================
0623458888    example.com   2863          172.30.18.22
UDP           1E83ECF0-AF0611E7-802B8FCF-594EB9E7@122.50.18.22

Peer5-F#show vpn-sip registration-status

 SIP registration of local number 0623458888 : registered 172.30.18.22
```

Session stuck in Negotiating IKE state

Symptom

VPN-SIP session stuck in Negotiating IKE state.

```
Peer5#show vpn-sip session remote-number 0612349999 detail
VPN-SIP session current status

Interface: Tunnel4
   Session status: NEGOTIATING_IKE (R)
   Uptime        : 00:00:58
   Remote number : 0612349999
   Local number  : 0623458888
   Remote address:port: 172.30.168.3:24825
   Local address:port : 172.30.18.22:50012
   Crypto conn handle: 0x8000002E
   SIP Handle        : 0x8000000C
   SIP callID        : 16
   Configured/Negotiated bandwidth: 1000/1000 kbps
```

Possible Cause

Bad configuration related to IKEv2.

In the following example the Key ID that is configured in the keyring does not match the SIP numberof the remote peer.

```
Peer5-F#show running-config interface tunnel 4
Building configuration...

Current configuration : 276 bytes
!
interface Tunnel4
 ip address 10.4.4.4 255.0.0.0
 tunnel source Loopback11
 tunnel mode ipsec ipv4
 tunnel destination dynamic
 tunnel protection ipsec profile test-prof ikev2-profile test
 VPN-SIP local-number 0623458888 remote-number 0612349999 bandwidth 1000  ====> Remote
number mentioned here doesn’t match the remote number in the keyring
end

IKEv2 Keyring configs:
!
crypto ikev2 keyring keys
 peer peer1
  identity key-id 0312341111
  pre-shared-key psk1
 !
 peer abc
  identity key-id 0345674444
  pre-shared-key psk1
 !
 peer peer2
  identity key-id 0334563333
  pre-shared-key psk10337101690
 !
 peer peer6
  identity key-id 0634567777
  pre-shared-key cisco123
 !
 peer peer3
  identity key-id 0323452222
  pre-shared-key cisco123
 !
 peer peer4
  identity key-id 0645676666
  pre-shared-key psk1
 !
 peer NONID
  identity fqdn example.com
  pre-shared-key psk1
 !
!
!
crypto ikev2 profile test
 match identity remote any
 identity local key-id 0623458888
 authentication remote pre-share
 authentication local pre-share
 keyring local keys
 dpd 10 6 periodic
 nat force-encap
```

Recommended Action

Correct the keyring configurations.

```
rypto ikev2 keyring keys
 peer peer1
  identity key-id 0312341111
  pre-shared-key psk1
 !
 peer abc
  identity key-id 0345674444
  pre-shared-key psk1
 !
 peer peer2
  identity key-id 0334563333
  pre-shared-key psk1
 !
 peer peer6
  identity key-id 0634567777
  pre-shared-key psk1
 !
 peer peer3
  identity key-id 0323452222
  pre-shared-key psk1
 !
 peer peer4
  identity key-id 0612349999
  pre-shared-key psk1
 !
 peer NONID
  identity fqdn example.com
  pre-shared-key psk1
 !
!
!
crypto ikev2 profile test
 match identity remote any
 identity local key-id 0623458888
 authentication remote pre-share
 authentication local pre-share
 keyring local keys
 dpd 10 6 periodic
 nat force-encap
!

Peer5-F#show vpn-sip session remote-number  0612349999 detail
VPN-SIP session current status

Interface: Tunnel4
   Session status: SESSION_UP (R)
   Uptime        : 00:02:04
   Remote number : 0612349999
   Local number  : 0623458888
   Remote address:port: 172.30.168.3:24845
   Local address:port : 172.30.18.22:50020
   Crypto conn handle: 0x8000004E
   SIP Handle        : 0x80000014
   SIP callID        : 24
   Configured/Negotiated bandwidth: 1000/1000 kbps
```

Troubleshooting Session Initiation

Symptom

Session does not initiate and gets stuck in Negotiating IKE state

Possible Cause

Fagmentation of IKE packets when a large PKI certificate is included in the IKE authentication message.

Recommended Action

Configure IKEv2 fragmentation on the routers.

Debug Commands

The folllwing debug commands are available to debug VPN-SIP configuration:

Table 2: debug commands

Command Name	Description
debug vpn-sip event	Prints debug messages for SVTI registration with VPN-SIP, SIP registration, call setup, and so on.
debug vpn-sip errors	Prints error messages only when an error occurs during initialization, registration, call setup, and so on.
debug vpn-sip sip all	Enables all SIP debugging traces.
debug vpn-sip sip calls	Enables SIP SPI calls debugging trace.
debug vpn-sip sip dhcp	Enables SIP-DHCP debugging trace
debug vpn-sip sip error	Enables SIP error debugging trace
debug vpn-sip sip events	Enables SIP events debugging trace.
debug vpn-sip sip feature	Enables feature level debugging.
debug vpn-sip sip function	Enables SIP function debugging trace.
debug vpn-sip sip info	Enables SIP information debugging trace.
debug vpn-sip sip level	Enables information level debugging.
debug vpn-sip sip media	Enables SIP media debugging trace.
debug vpn-sip sip messages	Enables SIP SPI messages debugging trace
debug vpn-sip sip non-call	Enables Non-Call-Context trace (OPTIONS, SUBSCRIBE, and so on)
debug vpn-sip sip preauth	Enable SIP preauth debugging trace.
debug vpn-sip sip states	Enable SIP SPI states debugging trace.
debug vpn-sip sip translate	Enables SIP translation debugging trace.
debug vpn-sip sip transport	Enables SIP transport debugging traces.
debug vpn-sip sip verbose	Enables verbose mode.

Additional References for VPN-SIP

Standards and RFCs

Standard/RFC	Title
RFC 6193 (with Restrictions)	Media Description for the Internet Key Exchange Protocol (IKE) in the Session Description Protocol (SDP)