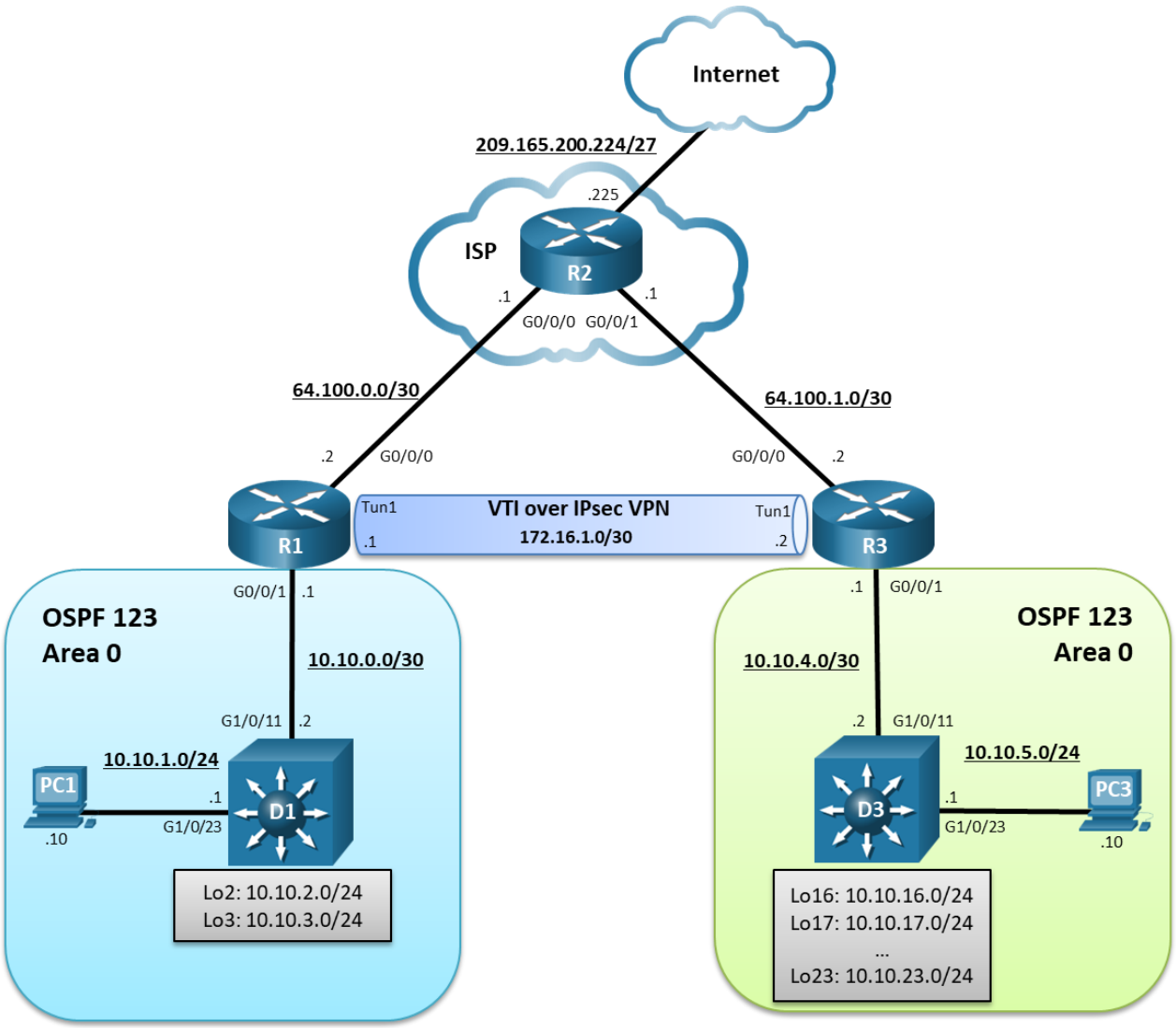
Lab - Implement IPsec VTI Site-to-Site VPNs

Topology



Addressing Table

Device	Interface	IPv4 Address	Default Gateway
R1	G0/0/0	64.100.0.2/30	N/A
	G0/0/1	10.10.0.1/29	
	Tunnel 1	172.16.1.1/30	
R2	G0/0/0	64.100.0.1/30	N/A

Device	Interface	IPv4 Address	Default Gateway
	G0/0/1	64.100.1.1/30	
	Lo0	209.165.200.225	
R3	G0/0/0	64.100.1.2/30	N/A
	G0/0/1	10.10.4.1/30	
	Tunnel 1	172.16.1.2/30	
D1	G1/0/11	10.10.0.2/29	N/A
	G1/0/23	10.10.1.1/24	
	Lo2	10.10.2.1/24	
	Lo3	10.10.3.1/24	
D3	G1/0/11	10.10.0.3/29	N/A
	G1/0/23	10.10.5.1/24	
	Lo16	10.10.16.1/24	
	Lo17	10.10.17.1/24	
	Lo18	10.10.18.1/24	
	Lo19	10.10.19.1/24	
	Lo20	10.10.20.1/24	
	Lo21	10.10.21.1/24	
	Lo22	10.10.22.1/24	
	Lo23	10.10.23.1/24	
PC1	NIC	10.10.1.10/24	10.10.1.1
PC3	NIC	10.10.5.10/24	10.10.5.1

Objectives

Part 1: Build the Network, Configure Basic Device Settings and Static Routing

Part 2: Configure Static IPsec VTI on R1 and R3

Part 3: Verify Static IPsec VTI on R1 and R3

Background / Scenario

IPsec can only send unicast IP traffic. Therefore, it does not support protocols that require multicast or broadcast communication such as routing protocols. Although GRE over IPsec can be configured to provide security and support for routing protocols, there is a newer more efficient method that can be used.

IPsec Virtual Tunnel Interface (VTI) greatly simplifies the VPN configuration process and provides a simpler alternative to using GRE tunnels for encapsulation and crypto maps with IPsec. Like GRE over IPsec, IPsec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic. Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using the IP routing table simplifies the IPsec VPN configuration compared to the more complex

process of using access control lists (ACLs) with the crypto map in native IPsec configurations. VTI over IPsec also encapsulates IPv4 or IPv6 traffic without the need for an additional GRE header. GRE adds a 4-byte header to every packet.

In this lab, you will build and configure a static VTI over IPsec with pre-shared key to enable a site-to-site VPN capable of supporting the OSPF routing protocol.

Note: This lab is an exercise in developing, deploying, and verifying how VNPs operate and does not reflect networking best practices.

Note: The routers used with this CCNP hands-on lab are Cisco 4221routers and the two Layer 3 switches are Catalyst 3650 switches. Other routers and Layer 3 switches and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the routers and switches have been erased and have no startup configurations. If you are unsure contact your instructor.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 Switches (Cisco 3650 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 PCs (Choice of operating system with a terminal emulation program installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

Part 1: Build the Network, Configure Basic Device Settings and Static Routing

In Part 1, you will set up the network topology, configure basic settings, interface addressing, and single-area OSPFv2 on the routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for the routers.

a. Console into each router and switch, enter global configuration mode, and apply the basic settings, and interface addressing. A command list for each device is provided for your reference.

Routing is enabled as follows:

- R2 has a static route to the networks connected to R1 (i.e., 10.10.0.0/22) and two static routes to the networks connected to R3 (i.e., 10.10.4.0/22, 10.10.16.0/21).
- R1 and R3 each have a default static route to R2.
- OSPFv2 routing is enabled between R1 and D1, and R1 is propagating the default route to D1.
- OSPFv2 routing is enabled between R3 and D3, and R3 is propagating the default route to D3.
- A command list for each device is listed below to perform initial configurations.

Router R1

```
hostname R1
no ip domain lookup
```

```
line con 0
 logging sync
 exec-time 0 0
 exit
banner motd # This is R1, Implement IPsec VTI Site-to-Site VPNs #
interface g0/0/0
 description Connection to R2
 ip add 64.100.0.2 255.255.255.252
 no shut
 exit
interface GigabitEthernet0/0/1
 description Connection to D1
 ip address 10.10.0.1 255.255.255.252
 no shut
 exit
router ospf 123
 router-id 1.1.1.1
 auto-cost reference-bandwidth 1000
 network 10.10.0.0 0.0.0.3 area 0
 default-information originate
exit
ip route 0.0.0.0 0.0.0.0 64.100.0.1
```

Router R2

```
hostname R2
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
banner motd # This is R2, Implement IPsec VTI Site-to-Site VPNs #
interface g0/0/0
 description Connection to R1
 ip add 64.100.0.1 255.255.255.252
 no shut
 exit
interface GigabitEthernet0/0/1
 description Connection to R3
 ip address 64.100.1.1 255.255.255.252
 no shut
 exit
int lo0
 description Internet simulated address
 ip add 209.165.200.225 255.255.255.224
 exit
ip route 0.0.0.0 0.0.0.0 Loopback0
```

```
ip route 10.10.0.0 255.255.252.0 64.100.0.2
ip route 10.10.4.0 255.255.252.0 64.100.1.2
ip route 10.10.16.0 255.255.248.0 64.100.1.2
```

Router R3

```
hostname R3
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
banner motd # This is R3, Implement IPsec VTI Site-to-Site VPNs #
interface g0/0/0
 description Connection to R2
 ip add 64.100.1.2 255.255.255.252
 no shut
 exit
interface GigabitEthernet0/0/1
 description Connection to D3
 ip address 10.10.4.1 255.255.255.252
 no shut
 exit
ip route 0.0.0.0 0.0.0.0 64.100.1.1
router ospf 123
 router-id 3.3.3.1
 auto-cost reference-bandwidth 1000
 network 10.10.4.0 0.0.0.3 area 0
 default-information originate
exit
```

Switch D1

```
hostname D1
no ip domain lookup
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
banner motd # This is D1, Implement IPsec VTI Site-to-Site VPNs #
interface G1/0/11
 description Connection to R1
 no switchport
 ip address 10.10.0.2 255.255.255.252
 no shut
 exit
interface G1/0/23
 description Connection to PC1
```

```
 no switchport
 ip address 10.10.1.1 255.255.255.0
 no shut
 exit
int Lo2
 description Loopback to simulate an OSPF network
 ip add 10.10.2.1 255.255.255.0
 ip ospf network point-to-point
exit
int Lo3
 description Loopback to simulate an OSPF network
 ip add 10.10.3.1 255.255.255.0
 ip ospf network point-to-point
exit
ip routing
router ospf 123
 router-id 1.1.1.2
 auto-cost reference-bandwidth 1000
 network 10.10.0.0 0.0.3.255 area 0
 exit
int range G1/0/1 - 10, G1/0/12 - 22, G1/0/24
 shut
 exit
```

Switch D3

```
hostname D3
no ip domain lookup
line con 0
 logging sync
 exec-time 0 0
 exit
banner motd # This is D3, Implement IPsec VTI Site-to-Site VPNs #
interface G1/0/11
 description Connection to R3
 no switchport
 ip address 10.10.4.2 255.255.255.252
 no shut
 exit
interface G1/0/23
 description Connection to PC3
 no switchport
 ip address 10.10.5.1 255.255.255.0
 no shut
 exit
int Lo16
 description Loopback to simulate an OSPF network
```

```
 ip add 10.10.16.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo17
 description Loopback to simulate an OSPF network
 ip add 10.10.17.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo18
 description Loopback to simulate an OSPF network
 ip add 10.10.18.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo19
 description Loopback to simulate an OSPF network
 ip add 10.10.19.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo20
 description Loopback to simulate an OSPF network
 ip add 10.10.20.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo21
 description Loopback to simulate an OSPF network
 ip add 10.10.21.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo22
 description Loopback to simulate an OSPF network
 ip add 10.10.22.1 255.255.255.0
 ip ospf network point-to-point
 exit
int Lo23
 description Loopback to simulate an OSPF network
 ip add 10.10.23.1 255.255.255.0
 ip ospf network point-to-point
 exit
ip routing
router ospf 123
 router-id 3.3.3.2
 auto-cost reference-bandwidth 1000
 network 10.10.4.0 0.0.1.255 area 0
 network 10.10.16.0 0.0.7.255 area 0
 exit
```

```
int range G1/0/1 - 10, G1/0/12 - 22, G1/0/24
 shut
```

b. Save the running configuration to startup-config.

Step 3: Configure PC1 and PC3 with IP addressing.

Configure the two PCs with the IP addresses listed in the Address Table. Also configure their respective default gateways.

Step 4: On PC1, verify end-to-end connectivity.

a. From PC1, **ping** PC3 (10.10.5.10).

```
PC1> ping 10.10.5.10
```

The pings should be successful. If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.

b. From PC1, **ping** the first loopback on D3 (10.10.16.1).

```
PC1> ping 10.10.16.1
```

The pings should be successful. If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.

c. From PC1, **ping** the default gateway loopback on R2 (209.165.200.225).

```
PC1> ping 209.165.200.225
```

The pings should be successful. If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.

Step 5: Verify the routing table of R1.

a. Verify the OSPF routing table of R1.

```
R1# show ip route ospf | begin Gateway
Gateway of last resort is 64.100.0.1 to network 0.0.0.0

      10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
O        10.10.1.0/24 [110/11] via 10.10.0.2, 00:29:03, GigabitEthernet0/0/1
O        10.10.2.0/24 [110/2] via 10.10.0.2, 00:29:03, GigabitEthernet0/0/1
O        10.10.3.0/24 [110/2] via 10.10.0.2, 00:29:03, GigabitEthernet0/0/1
```

The routing table confirms that R1 has knowledge of the networks connected to D1. Notice that R1 has no knowledge of the routes connected to the R3 OSPF domain. The reason why PC1 can still reach PC3 is because R1 has a default static route to R2. R1 forwarded the traffic to R2 because it did not know where the 10.10.5.0 network was. R2 has a static route to this network and therefore forwarded it to R3.

b. Verify the routing table of R3.

```
R3# show ip route ospf | begin Gateway
Gateway of last resort is 64.100.1.1 to network 0.0.0.0

      10.0.0.0/8 is variably subnetted, 11 subnets, 3 masks
O        10.10.5.0/24 [110/11] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O        10.10.16.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O        10.10.17.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O        10.10.18.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O        10.10.19.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
```

```
O        10.10.20.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O        10.10.21.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O        10.10.22.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
O        10.10.23.0/24 [110/2] via 10.10.4.2, 00:00:41, GigabitEthernet0/0/1
```

Like R1, the routing table of R3 only contains its local routes.

Part 2: Configure Static IPsec VTI on R1 and R3

A limitation of IPsec VPNs is that it only forwards unicast traffic across the VPN tunnel. Therefore, routing protocol traffic is not propagated across the VPN tunnel.

GRE over IPsec VPN could be configured to support routing protocol traffic over the IPsec VPN. However, IP VTI is simpler and more efficient than GRE over IPsec.

IPsec VTI can be configured using:

- **Static VTIs (SVTIs)** - SVTI configurations can be used for site-to-site connectivity in which a tunnel provides always-on access between two sites. The advantage of using SVTIs as opposed to crypto map configurations is that users can enable dynamic routing protocols on the tunnel interface without the extra 4 bytes required for GRE headers, therefore reducing the bandwidth for sending encrypted data.
- **Dynamic VTIs (DVTIs)** - DVTIs can provide highly secure and scalable connectivity for remote-access VPNs. The DVTI technology replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels.

The steps to enable IPsec VTI are very similar to GRE over IPsec except:

Step 1. The tunnel interface is configured with the **tunnel mode ipsec {ipv4 | ipv6}** command.

Step 2. The transform set is configured with the mode tunnel command. An ACL is not required.

Like site-to-site VPNs using crypto maps and GRE over IPsec using crypto maps, IPsec VTI also requires the following:

- ISAKMP policy configuration and pre-shared key configured
- Transform set configured
- IPsec profile configured

In this part, you will configure a static IPsec SVTI to provide an always on site-to-site VPN as shown in the topology diagram.

Step 1: On R1 and R3, configure the ISAKMP policy and pre-shared key.

In this lab, we will use the following parameters for the ISAKMP policy 10 on R1 and R3:

- Encryption: **aes 256**
- Hash: sha256
- Authentication method: **pre-share key**
- Diffie-Hellman group: **14**
- Lifetime: **3600** seconds (60 minutes / 1 hour)

a. Configure ISAKMP policy 10 on R1 and R3.

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# hash sha256
```

```
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 14
R1(config-isakmp)# lifetime 3600
R1(config-isakmp)# exit

R3(config)# crypto isakmp policy 10
R3(config-isakmp)# encryption aes 256
R3(config-isakmp)# hash sha256
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# group 14
R3(config-isakmp)# lifetime 3600
R3(config-isakmp)# exit
```

b. Configure the pre-shared key of **cisco123** on R1 and R3.

Note: Production networks should use longer and more complex keys.

```
R1(config)# crypto isakmp key cisco123 address 64.100.1.2

R3(config)# crypto isakmp key cisco123 address 64.100.0.2
```

Step 2: On R1 and R3, configure the transform set and tunnel mode.

Create a new transform set called VTI-VPN using ESP AES 256 for encryption and ESP SHA256 HMAC for authentication and set the mode to **tunnel**.

Note: The transform set would default to tunnel mode automatically but is configured in the example for emphasis.

```
R1(config)# crypto ipsec transform-set VTI-VPN esp-aes 256 esp-sha256-hmac
R1(cfg-crypto-trans)# mode tunnel
R1(cfg-crypto-trans)# exit

R3(config)# crypto ipsec transform-set VTI-VPN esp-aes 256 esp-sha256-hmac
R3(cfg-crypto-trans)# mode tunnel
R3(cfg-crypto-trans)# exit
```

Step 3: On R1 and R3, configure VTI over IPsec using IPsec profiles.

Configure an IPsec profile called **VTI-PROFILE** using the **crypto ipsec profile** *ipsec-profile-name* global configuration command and set the transform set to VTI-VPN.

```
R1(config)# crypto ipsec profile VTI-PROFILE
R1(ipsec-profile)# set transform-set VTI-VPN
R1(ipsec-profile)# exit

R3(config)# crypto ipsec profile VTI-PROFILE
R3(ipsec-profile)# set transform-set VTI-VPN
R3(ipsec-profile)# exit
```

Step 4: On R1, configure the tunnel interface.

a. Next, configure a tunnel interface on R1.

```
R1(config)# interface Tunnel1
R1(config-if)# bandwidth 4000
```

```
R1(config-if)# ip address 172.16.1.1 255.255.255.252
R1(config-if)# ip mtu 1400
R1(config-if)# tunnel source 64.100.0.2
R1(config-if)# tunnel destination 64.100.1.2
R1(config-if)#
*Jan 21 12:31:13.824: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to up
```

b. Tunnel interfaces default to **tunnel mode gre** mode. However, we must now change the tunnel mode from the default GRE setting to the IPsec setting. Configure Tunnel 1 using the **tunnel mode ipsec ipv4** command.

```
R1(config-if)# tunnel mode ipsec ipv4
R1(config-if)#
*Jan 21 12:32:15.047: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to down
```

c. Next, the IPsec profile **VTI-PROFILE** must be applied using the **tunnel protection ipsec profile** *profile-name* command.

```
R1(config-if)# tunnel protection ipsec profile VTI-PROFILE
R1(config-if)#
*Jan 21 12:32:50.103: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R1(config-if)# exit
```

Notice the informational message that the ISAKMP policy will be used.

Step 5: On R3, configure the tunnel interface.

Now we must mirror the configuration of R1 on R3.

a. Next, configure a GRE tunnel interface on R3.

```
R3(config)# interface Tunnel1
R3(config-if)# bandwidth 4000
R3(config-if)# ip address 172.16.1.2 255.255.255.252
R3(config-if)# ip mtu 1400
R3(config-if)# tunnel source 64.100.1.2
*Feb 20 12:53:14.367: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to down
R3(config-if)# tunnel destination 64.100.0.2
R3(config-if)#
*Feb 20 12:53:16.683: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to up
```

Notice the information messages indicating the line going down and then up.

b. Tunnel 1 must be configured using the **tunnel mode ipsec ipv4** command.

```
R3(config-if)# tunnel mode ipsec ipv4
R3(config-if)#
*Feb 20 12:53:45.931: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to down
```

Again, the Tunnel 1 interface goes down.

c. Finally, the IPsec profile **VTI-PROFILE** must be applied using the **tunnel protection ipsec profile** *profile-name* command.

```
R3(config-if)# tunnel protection ipsec profile VTI-PROFILE
R3(config-if)#
*Feb 20 12:54:05.111: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R3(config-if)#
*Feb 20 12:54:05.381: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Tunnel1, changed state to up
R3(config-if)# exit
```

Notice the informational message that the ISAKMP policy will be used and that the Tunnel 1 interface is up.

Step 6: On R1 and R3, advertise the tunnel interface in OSPF.

a. On R1, configure OSPF to advertise the tunnel interfaces.

```
R1(config)# router ospf 123
R1(config-router)# network 172.16.1.0 0.0.0.3 area 0
R1(config-router)# end
```

b. Next on R3, configure OSPF to advertise the tunnel interfaces.

```
R3(config)# router ospf 123
R3(config-router)# network 172.16.1.0 0.0.0.3 area 0
R3(config-router)# exit
R3(config)#
*Feb 20 13:09:48.456: %OSPF-5-ADJCHG: Process 123, Nbr 1.1.1.1 on Tunnel1
from LOADING to FULL, Loading Done
R3(config)# exit
```

Notice the OSPF adjacency message that appears when the network command is entered.

Part 3: Verify Static IPsec VTI on R1 and R3

Now that the IPsec has been configured, we must verify that the tunnel interfaces are correctly enabled, that the crypto session is active, and then generate traffic to confirm it is traversing securely over the IPsec VTI tunnel.

Step 1: On R1 and R3, verify the tunnel interfaces.

a. Use the **show interfaces tunnel 1** command to verify the interface settings.

```
R1# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.1/30
  MTU 9938 bytes, BW 4000 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 64.100.0.2, destination 64.100.1.2
```

```
Tunnel protocol/transport IPSEC/IP
Tunnel TTL 255
Tunnel transport MTU 1438 bytes
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPSec (profile "VTI-PROFILE")
Last input 00:00:07, output 00:00:08, output hang never
Last clearing of "show interface" counters 00:32:55
Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   20 packets input, 2368 bytes, 0 no buffer
   Received 0 broadcasts (0 IP multicasts)
   0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   23 packets output, 2424 bytes, 0 underruns
   0 output errors, 0 collisions, 0 interface resets
   0 unknown protocol drops
   0 output buffer failures, 0 output buffers swapped out
```

Notice the highlighted output identifying various aspects of the tunnel interface.

b. On R3, use the **show interfaces tunnel 1** command to verify the interface settings.

```
R3# show interface tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.2/30
  MTU 9938 bytes, BW 4000 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 64.100.1.2, destination 64.100.0.2
  Tunnel protocol/transport IPSEC/IP
  Tunnel TTL 255
  Tunnel transport MTU 1438 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "VTI-PROFILE")
  Last input 00:00:03, output 00:00:09, output hang never
  Last clearing of "show interface" counters 00:24:32
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

```
     62 packets input, 6324 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runs, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     58 packets output, 6168 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

Again, the highlighted output identifies various aspects of the tunnel interface.

Step 2: On R1 and R3, verify the crypto settings.

a. On R1, use the **show crypto session** command to verify the operation of the VPN tunnel.

```
R1# show crypto session
Crypto session current status

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 64.100.1.2 port 500
  Session ID: 0
  IKEv1 SA: local 64.100.0.2/500 remote 64.100.1.2/500 Active
  Session ID: 0
  IKEv1 SA: local 64.100.0.2/500 remote 64.100.1.2/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 4, origin: crypto map
```

The output confirms that Tunnel 1 is up and active with R3 (64.100.1.2). The port 500 refers to ISAKMP using UDP port 500.

b. On R3, use the **show crypto session** command to verify the operation of the VPN tunnel.

```
R3# show crypto session
Crypto session current status

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 64.100.0.2 port 500
  Session ID: 0
  IKEv1 SA: local 64.100.1.2/500 remote 64.100.0.2/500 Active
  Session ID: 0
  IKEv1 SA: local 64.100.1.2/500 remote 64.100.0.2/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 4, origin: crypto map
```

Step 3: On R1 and R3, verify the routing tables.

a. Verify the R1 routing table for OSPF routes.

```
R1# show ip route ospf | begin Gateway
Gateway of last resort is 64.100.0.1 to network 0.0.0.0

      10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
```

```
O        10.10.1.0/24 [110/11] via 10.10.0.2, 01:28:00, GigabitEthernet0/0/1
O        10.10.2.0/24 [110/2] via 10.10.0.2, 01:28:00, GigabitEthernet0/0/1
O        10.10.3.0/24 [110/2] via 10.10.0.2, 01:28:00, GigabitEthernet0/0/1
O        10.10.4.0/30 [110/251] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.5.0/24 [110/261] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.16.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.17.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.18.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.19.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.20.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.21.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.22.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
O        10.10.23.0/24 [110/252] via 172.16.1.2, 00:20:31, Tunnel1
```

Notice how R1 has learned about the R3 OSPF networks via the tunnel interface.

b. Verify the R3 routing table for OSPF routes.

```
R3# show ip route ospf | begin Gateway
Gateway of last resort is 64.100.1.1 to network 0.0.0.0

      10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
O        10.10.0.0/30 [110/251] via 172.16.1.1, 00:22:10, Tunnel1
O        10.10.1.0/24 [110/261] via 172.16.1.1, 00:22:10, Tunnel1
O        10.10.2.0/24 [110/252] via 172.16.1.1, 00:22:10, Tunnel1
O        10.10.3.0/24 [110/252] via 172.16.1.1, 00:22:10, Tunnel1
O        10.10.5.0/24 [110/11] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.16.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.17.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.18.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.19.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.20.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.21.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.22.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
O        10.10.23.0/24 [110/2] via 10.10.4.2, 01:28:53, GigabitEthernet0/0/1
```

Notice how R3 has learned about the R1 OSPF networks via the tunnel interface.

c. From D1, trace the path taken to the R3 10.10.5.1 interface.

```
D1# trace 10.10.5.1
Type escape sequence to abort.
Tracing the route to 10.10.5.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.0.1 2 msec 2 msec 2 msec
  2 172.16.1.2 3 msec 2 msec 3 msec
  3 10.10.4.2 3 msec *  4 msec
```

Notice how the path taken is through the VPN tunnel interface.

d. On R1, verify the IPsec SA encrypted and decrypted statistics.

```
R1# show crypto ipsec sa | include encrypt|decrypt
     #pkts encaps: 28, #pkts encrypt: 28, #pkts digest: 28
```

```
    #pkts decaps: 26, #pkts decrypt: 26, #pkts verify: 26
```

e. Verify that there is an operational logical point-to-point link between R1 and R3 using the VTI tunnel interface.

```
R1# show ip route 172.16.0.0
Routing entry for 172.16.0.0/16, 2 known subnets
  Attached (2 connections)
  Variably subnetted with 2 masks
C        172.16.1.0/30 is directly connected, Tunnel1
L        172.16.1.1/32 is directly connected, Tunnel1

R3# show ip route 172.16.0.0
Routing entry for 172.16.0.0/16, 2 known subnets
  Attached (2 connections)
  Variably subnetted with 2 masks
C        172.16.1.0/30 is directly connected, Tunnel1
L        172.16.1.2/32 is directly connected, Tunnel1
```

Step 4: Test the IPsec VTI tunnel.

a. From D1, trace the path taken to the R3 10.10.16.1 interface.

```
D1# trace 10.10.16.1
Type escape sequence to abort.
Tracing the route to 10.10.16.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.0.1 0 msec 0 msec 9 msec
  2 172.16.1.2 0 msec 0 msec 0 msec
  3 10.10.4.2 8 msec *  0 msec
```

Notice now that the path taken is through the VPN tunnel interface.

b. On R1, verify the IPsec SA encrypted and decrypted statistics.

```
R1# show crypto ipsec sa | include encrypt|decrypt
    #pkts encaps: 230, #pkts encrypt: 230, #pkts digest: 230
    #pkts decaps: 200, #pkts decrypt: 200, #pkts verify: 200
```

The output verifies that the IPsec VTI is properly encrypting traffic between both sites. The packets encrypted include the trace packets along with OSPF packets.

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.