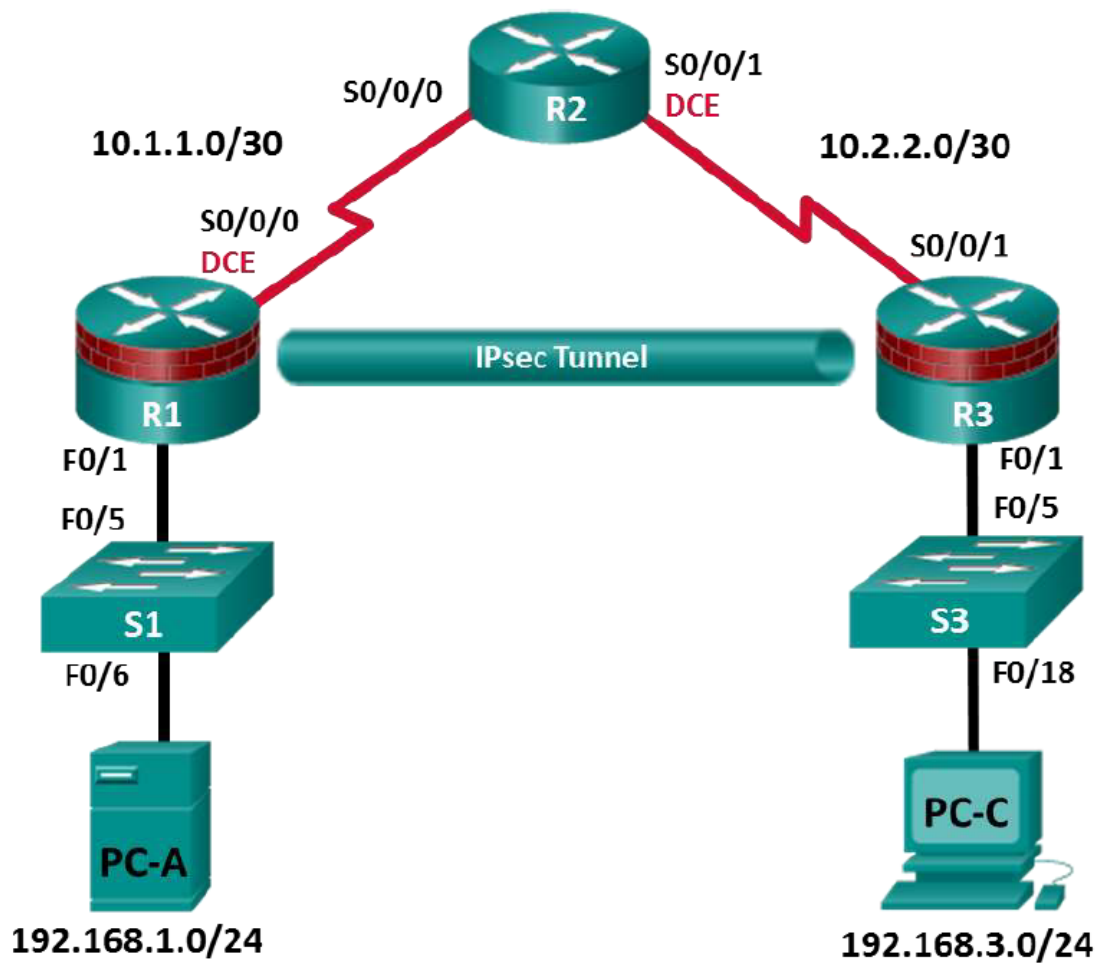


## Site-to-Site IPsec VPN

### Topology



### IP Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	F0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
R3	F0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1	10.2.2.1	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/6
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

## Introduction

VPNs can provide a secure method of transmitting data over a public network, such as the Internet. VPN connections can help reduce the costs associated with leased lines. Site-to-Site VPNs typically provide a secure (IPsec or other) tunnel between a branch office and a central office. Another common implementation of VPN technology is remote access to a corporate office from a telecommuter location, such as a small office or home office.

In this lab, you will build and configure a multi-router network, use Cisco IOS to configure a site-to-site IPsec VPN, and then test the VPN. The IPsec VPN tunnel is from R1 to R3 via R2. R2 acts as a pass-through and has no knowledge of the VPN. IPsec provides secure transmission of sensitive information over unprotected networks, such as the Internet. IPsec acts at the network layer and protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers.

**Note:** Before beginning, ensure that the routers and the switches have been erased and have no startup configurations.

```
Router# erase startup-config
Router# reload
```

## Required Resources

- 3 routers
- 2 switches (not required)
- 2 PCs
- Serial and Ethernet cables, as shown in the topology
- Console cables to configure Cisco networking devices

## Part 1: Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings, such as the interface IP addresses, dynamic routing, device access, and passwords.

**Note:** All tasks should be performed on R1, R2, and R3. The procedure for R1 is shown here as an example.

### 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 each router.

- Configure hostnames, as shown in the topology.
- Configure the interface IP addresses, as shown in the IP Addressing Table.
- Configure a clock rate of 64000 for the serial router interfaces with a DCE serial cable attached.

```
R1(config)# interface S0/0/0
R1(config-if)# clock rate 64000
```

### Step 3: Disable DNS lookup.

Disable DNS lookup to prevent the router from attempting to translate incorrectly entered commands.

```
R1(config)# no ip domain-lookup
```

### Step 4: Configure the OSPF routing protocol on R1, R2, and R3.

- On R1, use the following commands:

```
R1(config)# router ospf 101
R1(config-router)# network 192.168.1.0 0.0.0.255 area 0
```

```
R1(config-router)# network 10.1.1.0 0.0.0.3 area 0
```

- b. On R2, use the following commands:

```
R2(config)# router ospf 101
```

```
R2(config-router)# network 10.1.1.0 0.0.0.3 area 0
```

```
R2(config-router)# network 10.2.2.0 0.0.0.3 area 0
```

- c. On R3, use the following commands:

```
R3(config)# router ospf 101
```

```
R3(config-router)# network 192.168.3.0 0.0.0.255 area 0
```

```
R3(config-router)# network 10.2.2.0 0.0.0.3 area 0
```

### Step 5: Configure PC host IP settings.

- Configure a static IP address, subnet mask, and default gateway for PC-A, as shown in the IP Addressing Table.
- Configure a static IP address, subnet mask, and default gateway for PC-C, as shown in the IP Addressing Table.

### Step 6: Verify basic network connectivity.

- Ping from R1 to the R3 Fa0/1 interface at IP address 192.168.3.1.  
If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.
- Ping from PC-A on the R1 LAN to PC-C on the R3 LAN.  
If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.

### Step 7: Configure and encrypt passwords.

Configure the same settings for R1 and R3. R1 is shown here as an example.

- Configure a minimum password length.

Use the **security passwords** command to set a minimum password length of 10 characters.

```
R1(config)# security passwords min-length 10
```

- Configure the enable secret password on both routers with a password of **seclab12345**. Use the type 9 (SCRYPT) hashing algorithm.

```
R1(config)# enable algorithm-type scrypt secret seclab12345
```

- Create a local **admin01** account using **admin01pass** for the password. Use the type 9 (SCRYPT) hashing algorithm.

```
R1(config)# username admin01 algorithm-type scrypt secret admin01pass
```

### Step 8: Configure the console line.

Configure the console to use the local database for login. For additional security, configure the line to log out after five minutes of inactivity. Issue the **logging synchronous** command to prevent console messages from interrupting command entry.

```
R1(config)# line console 0
```

```
R1(config-line)# login local
```

```
R1(config-line)# exec-timeout 5 0
```

```
R1(config-line)# logging synchronous
```

### Step 9: Configure SSH Server.

- Configure a domain name **securitylab.com**.

```
R1(config)# ip domain-name securitylab.com
```

- b. Configure the RSA keys with **1024** for the number of modulus bits.

```
R1(config)# crypto key generate rsa general-keys modulus 1024
```

- c. Issue the command to force the use of SSH version 2.

```
R1(config)# ip ssh version 2
```

- d. Configure the vty lines on R1 and R3 to use the local database for login. Remote access to the routers should only be allowed using SSH. Configure the vty lines to logout after five minutes of inactivity.

```
R1(config)# line vty 0 4
```

```
R1(config-line)# login local
```

```
R1(config-line)# exec-timeout 5 0
```

```
R1(config-line)# transport input ssh
```

### Step 10: Save the basic running configuration for all three routers.

Save the running configuration to the startup configuration from the privileged EXEC mode prompt on R1, R2, and R3.

```
R1# copy running-config startup-config
```

## Part 2: Configure a Site-to-Site VPN with Cisco IOS

In Part 2 of this lab, you will configure an IPsec VPN tunnel between R1 and R3 that passes through R2. You will configure R1 and R3 using the Cisco IOS CLI. You will then review and test the resulting configuration.

### Task 1: Configure IPsec VPN Settings on R1 and R3.

#### Step 1: Verify connectivity from the R1 LAN to the R3 LAN.

In this task, you will verify that PC-A on the R1 LAN can ping PC-C on the R3 LAN with no tunnel in place.

Ping the PC-C IP address of **192.168.3.3** from PC-A.

```
PC-A:\> ping 192.168.3.3
```

If the pings are unsuccessful, troubleshoot the basic device configurations before continuing.

#### Step 2: Enable IKE policies on R1 and R3.

IPsec is an open framework that allows for the exchange of security protocols as new technologies, and encryption algorithms as they are developed.

There are two central configuration elements in the implementation of an IPsec VPN:

- Implement Internet Key Exchange (IKE) parameters
- Implement IPsec parameters

- a. Verify that IKE is supported and enabled.

IKE Phase 1 defines the key exchange method used to pass and validate IKE policies between peers. In IKE Phase 2, the peers exchange and match IPsec policies for the authentication and encryption of data traffic.

IKE must be enabled for IPsec to function. IKE is enabled, by default, on IOS images with cryptographic feature sets. If it is disabled, you can enable it with the **crypto isakmp enable** command. Use this command to verify that the router IOS supports IKE and that it is enabled.

```
R1(config)# crypto isakmp enable
```

```
R3(config)# crypto isakmp enable
```

- b. Establish an ISAKMP policy and view the available options.

To allow IKE Phase 1 negotiation, you must create an ISAKMP policy and configure a peer association involving that ISAKMP policy. An ISAKMP policy defines the authentication and encryption algorithms and the hash function used to send control traffic between the two VPN endpoints. When an ISAKMP security association has been accepted by the IKE peers, IKE Phase 1 has been completed. IKE Phase 2 parameters will be configured later.

Issue the **crypto isakmp policy number** global configuration mode command on R1 for policy 10.

```
R1(config)# crypto isakmp policy 10
```

- c. View the various IKE parameters available using Cisco IOS help by typing a question mark (?).

```
R1(config-isakmp)# ?
```

ISAKMP commands:

authentication	Set authentication method for protection suite
default	Set a command to its defaults
encryption	Set encryption algorithm for protection suite
exit	Exit from ISAKMP protection suite configuration mode
group	Set the Diffie-Hellman group
hash	Set hash algorithm for protection suite
lifetime	Set lifetime for ISAKMP security association
no	Negate a command or set its defaults

### Step 3: Configure the IKE Phase 1 ISAKMP policy on R1 and R3.

Your choice of an encryption algorithm determines how confidential the control channel between the endpoints is. The hash algorithm controls data integrity, ensuring that the data received from a peer has not been tampered with in transit. The authentication type ensures that the packet was sent and signed by the remote peer. The Diffie-Hellman group is used to create a secret key shared by the peers that has not been sent across the network.

- a. Configure an ISAKMP policy with a priority of **10**. Use **pre-shared key** as the authentication type, **aes 256** for the encryption algorithm, **sha** as the hash algorithm, and the Diffie-Hellman group **14** key exchange. Give the policy a lifetime of **3600** seconds (one hour).

```
R1(config)# crypto isakmp policy 10  
R1(config-isakmp)# hash sha  
R1(config-isakmp)# authentication pre-share  
R1(config-isakmp)# group 14  
R1(config-isakmp)# lifetime 3600  
R1(config-isakmp)# encryption aes 256  
R1(config-isakmp)# end
```

- b. Configure the same policy on R3.

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

- c. Verify the IKE policy with the **show crypto isakmp policy** command.

```
R1# show crypto isakmp policy  
Global IKE policy  
Protection suite of priority 10  
    encryption algorithm:  AES - Advanced Encryption Standard (256 bit keys).  
    hash algorithm:        Secure Hash Standard
```

```
authentication method: Pre-Shared Key
Diffie-Hellman group: #14 (2048 bit)
lifetime: 3600 seconds, no volume limit
```

#### Step 4: Configure pre-shared keys.

Because pre-shared keys are used as the authentication method in the IKE policy, a key must be configured on each router that points to the other VPN endpoint. These keys must match for authentication to be successful. The global configuration mode **crypto isakmp key <key-string> address <ip-address>** command is used to enter a pre-shared key. Use the IP address of the remote peer, which is the remote interface that the peer would use to route traffic to the local router.

Which IP addresses should you use to configure the IKE peers, given the topology diagram and IP addressing table?

---

---

---

The IP addresses you should use to configure the IKE peers are R1 S0/0/0 IP address 10.1.1.1 and R3 S0/0/1 IP address 10.2.2.1. These are the addresses that are used to send normal traffic between R1 and R3.

- Each IP address that is used to configure the IKE peers is also referred to as the IP address of the remote VPN endpoint. Configure the pre-shared key of **seclab123** on router R1. Production networks should use a complex key. This command points to the remote peer R3 S0/0/1 IP address.

```
R1(config)# crypto isakmp key seclab123 address 10.2.2.1
```

- Configure the pre-shared key **seclab123** on router R3. The command for R3 points to the R1 S0/0/0 IP address.

```
R3(config)# crypto isakmp key seclab123 address 10.1.1.1
```

#### Step 5: Configure the IPsec transform set and lifetime.

- The IPsec transform set is another crypto configuration parameter that routers negotiate to form a security association. To create an IPsec transform set, use the **crypto ipsec transform-set <tag>** command. Use ? to see which parameters are available.

```
R1(config)# crypto ipsec transform-set 50 ?
  ah-md5-hmac    AH-HMAC-MD5 transform
  ah-sha-hmac    AH-HMAC-SHA transform
  comp-lzs       IP Compression using the LZS compression algorithm
  esp-3des       ESP transform using 3DES(EDE) cipher (168 bits)
  esp-aes        ESP transform using AES cipher
  esp-des        ESP transform using DES cipher (56 bits)
  esp-md5-hmac   ESP transform using HMAC-MD5 auth
  esp-null       ESP transform w/o cipher
  esp-seal       ESP transform using SEAL cipher (160 bits)
  esp-sha-hmac   ESP transform using HMAC-SHA auth
```

- On R1 and R3, create a transform set with tag 50 and use an ESP transform with an AES 256 cipher with ESP and the SHA hash function. The transform sets must match.

```
R1(config)# crypto ipsec transform-set 50 esp-aes 256 esp-sha-hmac
R1(cfg-crypto-trans)# exit
```

```
R3(config)# crypto ipsec transform-set 50 esp-aes 256 esp-sha-hmac
R3(cfg-crypto-trans)# exit
```

The IPsec transform set specifies the cryptographic algorithms and functions (transforms) that a router employs on the actual data packets sent through the IPsec tunnel. These algorithms include the encryption, encapsulation, authentication, and data integrity services that IPsec can apply.

- c. You can also change the IPsec security association lifetime from the default of 3600 seconds. On R1 and R3, set the IPsec security association lifetime to 30 minutes, or 1800 seconds.

```
R1(config)# crypto ipsec security-association lifetime seconds 1800
```

```
R3(config)# crypto ipsec security-association lifetime seconds 1800
```

### Step 6: Define interesting traffic.

To make use of the IPsec encryption with the VPN, it is necessary to define extended access lists to tell the router which traffic to encrypt. A packet that is permitted by an access list used for defining IPsec traffic is encrypted if the IPsec session is configured correctly. A packet that is denied by one of these access lists is not dropped it is sent unencrypted. Also, like any other access list, there is an implicit deny at the end, which means the default action is to not encrypt traffic. If there is no IPsec security association correctly configured, no traffic is encrypted and traffic is forwarded unencrypted.

In this scenario, from the perspective of R1, the traffic you want to encrypt is traffic going from R1's Ethernet LAN to R3's Ethernet LAN or vice versa from the perspective of R3. These access lists are used outbound on the VPN endpoint interfaces and must mirror each other.

- a. Configure the IPsec VPN interesting traffic ACL on R1.

```
R1(config)# access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
```

- b. Configure the IPsec VPN interesting traffic ACL on R3.

```
R3(config)# access-list 101 permit ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255
```

IPsec does evaluate whether access lists are mirrored. IPsec does not form a security association if the peers do not have mirrored access lists to select interesting traffic.

### Step 7: Create and apply a crypto map.

A crypto map associates traffic that matches an access list to a peer and various IKE and IPsec settings. After the crypto map is created, it can be applied to one or more interfaces. The interfaces that it is applied to should be the ones facing the IPsec peer.

To create a crypto map, use **crypto map <name> <sequence-num> <type>** command in global configuration mode to enter crypto map configuration mode for that sequence number. Multiple crypto map statements can belong to the same crypto map and are evaluated in ascending numerical order. Enter crypto map configuration mode on R1. Use a type of ipsec-isakmp, which means IKE is used to establish IPsec security associations.

- a. Create the crypto map on R1, name it **CMAP**, and use **10** as the sequence number. A message displays after the command is issued.

```
R1(config)# crypto map CMAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
```

- b. Use the **match address <access-list>** command to specify which access list defines which traffic to encrypt.

```
R1(config-crypto-map)# match address 101
```

- c. To view the list of possible **set** commands that you can do with a crypto map, use the help function.

```
R1(config-crypto-map)# set ?
identity          Identity restriction.
ip                Interface Internet Protocol config commands
isakmp-profile    Specify isakmp Profile
```

nat	Set NAT translation
peer	Allowed Encryption/Decryption peer.
pfs	Specify pfs settings
reverse-route	Reverse Route Injection.
security-association	Security association parameters
transform-set	Specify list of transform sets in priority order

- d. Setting a peer IP or hostname is required. Set it to R3's remote VPN endpoint interface using the following command.

```
R1(config-crypto-map)# set peer 10.2.2.1
```

- e. Use the **set transform-set <tag>** command to hard code the transform set to be used with this peer. Set the perfect forwarding secrecy type using the **set pfs <type>** command, and modify the default IPsec security association life time with the **set security-association lifetime seconds <seconds>** command.

```
R1(config-crypto-map)# set pfs group14
R1(config-crypto-map)# set transform-set 50
R1(config-crypto-map)# set security-association lifetime seconds 900
R1(config-crypto-map)# exit
```

- f. Create a mirrored matching crypto map on R3.

```
R3(config)# crypto map CMAP 10 ipsec-isakmp
R3(config-crypto-map)# match address 101
R3(config-crypto-map)# set peer 10.1.1.1
R3(config-crypto-map)# set pfs group14
R3(config-crypto-map)# set transform-set 50
R3(config-crypto-map)# set security-association lifetime seconds 900
R3(config-crypto-map)# exit
```

- g. Apply the crypto map to interfaces.

**Note:** The SAs are not established until the crypto map has been activated by interesting traffic. The router generates a notification that crypto is now on.

Apply the crypto maps to the appropriate interfaces on R1 and R3.

```
R1(config)# interface S0/0/0
R1(config-if)# crypto map CMAP
*Jan 28 04:09:09.150: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R1(config)# end
```

```
R3(config)# interface S0/0/1
R3(config-if)# crypto map CMAP
*Jan 28 04:10:54.138: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R3(config)# end
```

## Task 2: Verify the Site-to-Site IPsec VPN Configuration.

### Step 1: Verify the IPsec configuration on R1 and R3.

- a. Previously, you used the **show crypto isakmp policy** command to display the configured ISAKMP policies on the router. The **show crypto ipsec transform-set** command displays the configured IPsec policies in the form of the transform sets.

```
R1# show crypto ipsec transform-set
Transform set 50: { esp-256-aes esp-sha-hmac  }
    will negotiate = { Tunnel,  },
```



```
Transform set #${default_transform_set_1}: { esp-aes esp-sha-hmac }  
will negotiate = { Transport, },
```

```
Transform set #${default_transform_set_0}: { esp-3des esp-sha-hmac }  
will negotiate = { Transport, },
```

R3# **show crypto ipsec transform-set**

```
Transform set 50: { esp-256-aes esp-sha-hmac }  
will negotiate = { Tunnel, },
```

```
Transform set #${default_transform_set_1}: { esp-aes esp-sha-hmac }  
will negotiate = { Transport, },
```

```
Transform set #${default_transform_set_0}: { esp-3des esp-sha-hmac }  
will negotiate = { Transport, },
```

- b. Use the **show crypto map** command to display the crypto maps that will be applied to the router.

R1# **show crypto map**

```
Crypto Map "CMAP" 10 ipsec-isakmp  
  Peer = 10.2.2.1  
  Extended IP access list 101  
    access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255  
  Current peer: 10.2.2.1  
  Security association lifetime: 4608000 kilobytes/900 seconds  
  Responder-Only (Y/N): N  
  PFS (Y/N): Y  
  DH group: group14  
  Transform sets={  
    50: { esp-256-aes esp-sha-hmac } ,  
  }  
  Interfaces using crypto map CMAP:  
    Serial0/0/0
```

R3# **show crypto map**

```
Crypto Map "CMAP" 10 ipsec-isakmp  
  Peer = 10.1.1.1  
  Extended IP access list 101  
    access-list 101 permit ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255  
  Current peer: 10.1.1.1  
  Security association lifetime: 4608000 kilobytes/900 seconds  
  Responder-Only (Y/N): N  
  PFS (Y/N): Y  
  DH group: group14  
  Transform sets={  
    50: { esp-256-aes esp-sha-hmac } ,  
  }  
  Interfaces using crypto map CMAP:  
    Serial0/0/1
```

### Task 3: Verify the IPsec VPN Operation.

#### Step 1: Display ISAKMP security associations.

The **show crypto isakmp sa** command reveals that no IKE SAs exist yet. When interesting traffic is sent, this command output will change.

```
R1# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst                src                state                conn-id status

IPv6 Crypto ISAKMP SA
```

#### Step 2: Display IPsec security associations.

The **show crypto ipsec sa** command shows the unused SA between R1 and R3.

**Note:** The number of packets sent across is zero, and there is a lack of any security associations listed toward the bottom of the output. The output for R1 is shown here.

```
R1# show crypto ipsec sa

interface: Serial0/0/0
  Crypto map tag: CMAP, local addr 10.1.1.1

  protected vrf: (none)
  local  ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
  remote ident (addr/mask/prot/port): (192.168.3.0/255.255.255.0/0/0)
  current_peer 10.2.2.1 port 500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #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.: 10.1.1.1, remote crypto endpt.: 10.2.2.1
  path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
  current outbound spi: 0x0(0)
  PFS (Y/N): N, DH group: none

  inbound esp sas:

  inbound ah sas:

  inbound pcp sas:

  outbound esp sas:

  outbound ah sas:

  outbound pcp sas:
```

IPsec has not begun to negotiate an SA over which it will encrypt traffic because no interesting traffic has been identified.

**Step 3: Generate some uninteresting test traffic and observe the results.**

- Ping from R1 to the R3 S0/0/1 interface IP address **10.2.2.1**. These pings should be successful.
- Issue the **show crypto isakmp sa** command.
- Ping from R1 to the R3 F0/1 interface IP address **192.168.3.1**. These pings should be successful.
- Issue the **show crypto isakmp sa** command again. Was an SA created for these pings? Explain.

---

SA was not created. The source address of both pings was the R1 S0/0/0 address of 10.1.1.1. In the first case, the destination address was 10.2.2.1. In the second case, the destination address was 192.168.3.1. This is not "interesting" traffic. The ACL 101 that is associated with the crypto map for R1 defines interesting traffic as IP packets from the 192.168.1.0/24 network to the 192.168.3.0/24 network.

- Issue the **debug ip ospf hello** command. You should see OSPF hello packets passing between R1 and R3.

```
R1# debug ip ospf hello
OSPF hello events debugging is on
R1#
*Apr  7 18:04:46.467: OSPF: Send hello to 224.0.0.5 area 0 on GigabitEthernet0/1
from 192.168.1.1
*Apr  7 18:04:50.055: OSPF: Send hello to 224.0.0.5 area 0 on Serial0/0/0 from
10.1.1.1
*Apr  7 18:04:52.463: OSPF: Rcv hello from 10.2.2.2 area 0 from Serial0/0/0
10.1.1.2
*Apr  7 18:04:52.463: OSPF: End of hello processing
*Apr  7 18:04:55.675: OSPF: Send hello to 224.0.0.5 area 0 on GigabitEthernet0/1
from 192.168.1.1
*Apr  7 18:04:59.387: OSPF: Send hello to 224.0.0.5 area 0 on Serial0/0/0 from
10.1.1.1
*Apr  7 18:05:02.431: OSPF: Rcv hello from 10.2.2.2 area 0 from Serial0/0/0
10.1.1.2
*Apr  7 18:05:02.431: OSPF: End of hello processing
```

- Turn off debugging with the **no debug ip ospf hello** or **undebug all** command.
- Re-issue the **show crypto isakmp sa** command. Was an SA created between R1 and R3?

---

This is router-to-router routing protocol traffic. The source and destination of these packets is not interesting, does not initiate the SA, and is not encrypted.

**Step 4: Generate some interesting test traffic and observe the results.**

- Use an extended ping from R1 to the R3 F0/1 interface IP address **192.168.3.1**. Extended ping allows you to control the source address of the packets. Respond as shown in the following example. Press **Enter** to accept the defaults, except where a specific response is indicated.

```
R1# ping 192.168.3.1 source 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:

Packet sent with a source address of 192.168.1.1
..!!!
Success rate is 100 percent (3/5), round-trip min/avg/max = 92/92/92 ms
```

- b. Re-issue the **show crypto isakmp sa** command.

```
R1# show crypto isakmp sa
```

```
IPv4 Crypto ISAKMP SA
```

dst	src	state	conn-id	status
10.2.2.1	10.1.1.1	QM_IDLE	1001	ACTIVE

```
IPv6 Crypto ISAKMP SA
```

The source was 192.168.1.1 and the destination was 192.168.3.1. This is interesting traffic based on the ACL 101 definition. An SA is established and packets travel through the tunnel as encrypted traffic.

What are the endpoints of the IPsec VPN tunnel?

---

---

---

- c. Ping from PC-A to PC-C. If the pings were successful, issue the **show crypto ipsec sa** command. How many packets have been transformed between R1 and R3?

---

---

---

**Done**