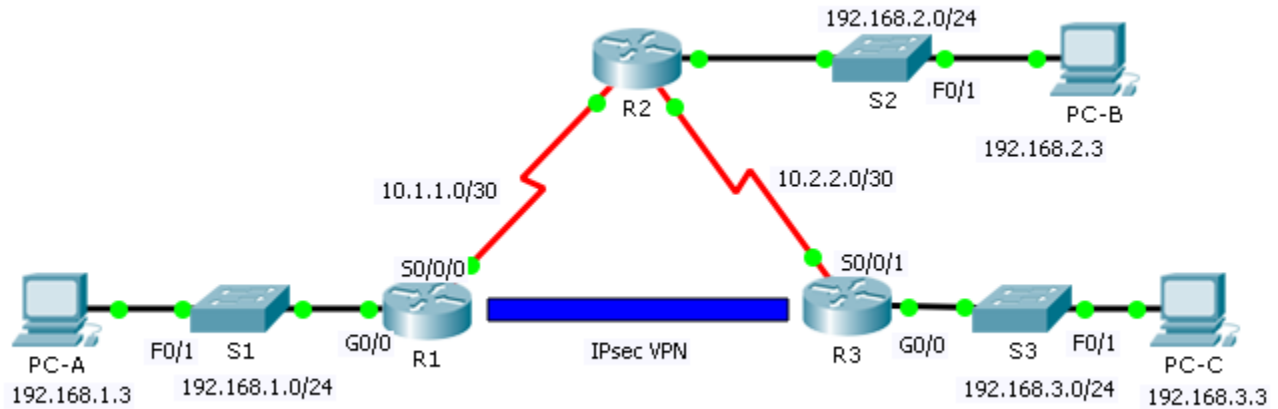


Packet Tracer – Configuring VPNs (Optional)

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	192.168.1.1	255.255.255.0	N/A
	S0/0/0	10.1.1.2	255.255.255.252	N/A
R2	G0/0	192.168.2.1	255.255.255.0	N/A
	S0/0/0	10.1.1.1	255.255.255.252	N/A
	S0/0/1	10.2.2.1	255.255.255.252	N/A
R3	G0/0	192.168.3.1	255.255.255.0	N/A
	S0/0/1	10.2.2.2	255.255.255.252	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.2.3	255.255.255.0	192.168.2.1
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1

ISAKMP Phase 1 Policy Parameters

Parameters		R1	R3
Key distribution method	Manual or ISAKMP	ISAKMP	ISAKMP
Encryption algorithm	DES , 3DES, or AES	AES	AES
Hash algorithm	MD5 or SHA-1	SHA-1	SHA-1
Authentication method	Pre-shared keys or RSA	pre-share	pre-share
Key exchange	DH Group 1 , 2, or 5	DH 2	DH 2
IKE SA Lifetime	86400 seconds or less	86400	86400
ISAKMP Key		cisco	cisco

Bolded parameters are defaults. Other parameters need to be explicitly configured.

IPsec Phase 2 Policy Parameters

Parameters	R1	R3
Transform Set	VPN-SET	VPN-SET
Peer Hostname	R3	R1
Peer IP Address	10.2.2.2	10.1.1.2
Network to be encrypted	192.168.1.0/24	192.168.3.0/24
Crypto Map name	VPN-MAP	VPN-MAP
SA Establishment	ipsec-isakmp	ipsec-isakmp

Objectives

Part 1: Enable Security Features

Part 2: Configure IPsec Parameters on R1

Part 3: Configure IPsec Parameters on R3

Part 4: Verify the IPsec VPN

Scenario

In this activity, you will configure two routers to support a site-to-site IPsec VPN for traffic flowing from their respective LANs. The IPsec VPN traffic will pass through another router that 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, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers.

Part 1: Enable Security Features

Step 1: Activate securityk9 module.

The Security Technology Package license must be enabled to complete this activity.

Note: Both the user EXEC and privileged EXEC pass word is **cisco**.

- a. Issue the **show version** command in the user EXEC or privileged EXEC mode to verify that the Security Technology Package license is activated.

```
-----  
Technology      Technology-package      Technology-package  
                Current      Type      Next reboot  
-----  
ipbase          ipbasek9      Permanent  ipbasek9  
security        None          None       None  
uc              None          None       None  
data            None          None       None
```

Configuration register is 0x2102

- b. If not, activate the **securityk9** module for the next boot of the router, accept the license, save the configuration, and reboot.

```
R1(config)# license boot module c2900 technology-package securityk9  
R1(config)# end  
R1# copy running-config startup-config  
R1# reload
```

- c. After the reloading is completed, issue the **show version** again to verify the Security Technology Package license activation.

Technology Package License Information for Module:'c2900'

```
-----  
Technology      Technology-package      Technology-package  
                Current      Type      Next reboot  
-----  
ipbase          ipbasek9      Permanent  ipbasek9  
security        securityk9    Evaluation  securityk9  
uc              None          None       None  
data            None          None       None
```

- d. Repeat Steps 1a to 1c with R3.

Part 2: Configure IPsec Parameters on R1

Step 1: Test connectivity.

Ping from **PC-A** to **PC-C**.

Step 2: Identify interesting traffic on R1.

Configure ACL 110 to identify the traffic from the LAN on **R1** to the LAN on **R3** as interesting. This interesting traffic will trigger the IPsec VPN to be implemented whenever there is traffic between **R1** to **R3** LANs. All other traffic sourced from the LANs will not be encrypted. Remember that due to the implicit deny any, there is no need to add the statement to the list.

```
R1(config)# access-list 110 permit ip 192.168.1.0 0.0.0.255 192.168.3.0  
0.0.0.255
```

Step 3: Configure the ISAKMP Phase 1 properties on R1.

Configure the crypto ISAKMP policy **10** properties on **R1** along with the shared crypto key **cisco**. Refer to the ISAKMP Phase 1 table for the specific parameters to configure. Default values do not have to be configured therefore only the encryption, key exchange method, and DH method must be configured.

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# encryption aes
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 2
R1(config-isakmp)# exit
R1(config)# crypto isakmp key cisco address 10.2.2.2
```

Step 4: Configure the ISAKMP Phase 2 properties on R1.

Create the transform-set **VPN-SET** to use **esp-3des** and **esp-sha-hmac**. Then create the crypto map **VPN-MAP** that binds all of the Phase 2 parameters together. Use sequence number **10** and identify it as an **ipsec-isakmp** map.

```
R1(config)# crypto ipsec transform-set VPN-SET esp-3des esp-sha-hmac
R1(config)# crypto map VPN-MAP 10 ipsec-isakmp
R1(config-crypto-map)# description VPN connection to R3
R1(config-crypto-map)# set peer 10.2.2.2
R1(config-crypto-map)# set transform-set VPN-SET
R1(config-crypto-map)# match address 110
R1(config-crypto-map)# exit
```

Step 5: Configure the crypto map on the outgoing interface.

Finally, bind the **VPN-MAP** crypto map to the outgoing Serial 0/0/0 interface. **Note:** This is not graded.

```
R1(config)# interface S0/0/0
R1(config-if)# crypto map VPN-MAP
```

Part 3: Configure IPsec Parameters on R3

Step 1: Configure router R3 to support a site-to-site VPN with R1.

Now configure reciprocating parameters on **R3**. Configure ACL **110** identifying the traffic from the LAN on **R3** to the LAN on **R1** as interesting.

```
R3(config)# access-list 110 permit ip 192.168.3.0 0.0.0.255 192.168.1.0
0.0.0.255
```

Step 2: Configure the ISAKMP Phase 1 properties on R3.

Configure the crypto ISAKMP policy **10** properties on **R3** along with the shared crypto key **cisco**.

```
R3(config)# crypto isakmp policy 10
R3(config-isakmp)# encryption aes
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# group 2
R3(config-isakmp)# exit
R3(config)# crypto isakmp key cisco address 10.1.1.2
```

Step 3: Configure the ISAKMP Phase 2 properties on R1.

Like you did on R1, create the transform-set **VPN-SET** to use **esp-3des** and **esp-sha-hmac**. Then create the crypto map **VPN-MAP** that binds all of the Phase 2 parameters together. Use sequence number **10** and identify it as an **ipsec-isakmp** map.

```
R3(config)# crypto ipsec transform-set VPN-SET esp-3des esp-sha-hmac
R3(config)# crypto map VPN-MAP 10 ipsec-isakmp
R3(config-crypto-map)# description VPN connection to R1
R3(config-crypto-map)# set peer 10.1.1.2
R3(config-crypto-map)# set transform-set VPN-SET
R3(config-crypto-map)# match address 110
R3(config-crypto-map)# exit
```

Step 4: Configure the crypto map on the outgoing interface.

Finally, bind the **VPN-MAP** crypto map to the outgoing Serial 0/0/1 interface. **Note:** This is not graded.

```
R3(config)# interface S0/0/1
R3(config-if)# crypto map VPN-MAP
```

Part 4: Verify the IPsec VPN

Step 1: Verify the tunnel prior to interesting traffic.

Issue the **show crypto ipsec sa** command on R1. Notice that the number of packets encapsulated, encrypted, decapsulated and decrypted are all set to 0.

```
R1# show crypto ipsec sa

interface: Serial0/0/0
  Crypto map tag: VPN-MAP, local addr 10.1.1.2

  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 10.2.2.2 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.: 10.1.1.2, remote crypto endpt.:10.2.2.2
  path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
  current outbound spi: 0x0(0)

<output omitted>
```

Step 2: Create interesting traffic.

Ping **PC-C** from **PC-A**.

Step 3: Verify the tunnel after interesting traffic.

On **R1**, re-issue the **show crypto ipsec sa** command. Now notice that the number of packets is more than 0 indicating that the IPsec VPN tunnel is working.

```
R1# show crypto ipsec sa

interface: Serial0/0/0
  Crypto map tag: VPN-MAP, local addr 10.1.1.2

  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 10.2.2.2 port 500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 3, #pkts encrypt: 3, #pkts digest: 0
    #pkts decaps: 3, #pkts decrypt: 3, #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 1, #recv errors 0

  local crypto endpt.: 10.1.1.2, remote crypto endpt.:10.2.2.2
  path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
  current outbound spi: 0x0A496941(172583233)

<output omitted>
```

Step 4: Create uninteresting traffic.

Ping **PC-B** from **PC-A**.

Step 5: Verify the tunnel.

On **R1**, re-issue the **show crypto ipsec sa** command. Finally, notice that the number of packets has not changed verifying that uninteresting traffic is not encrypted.