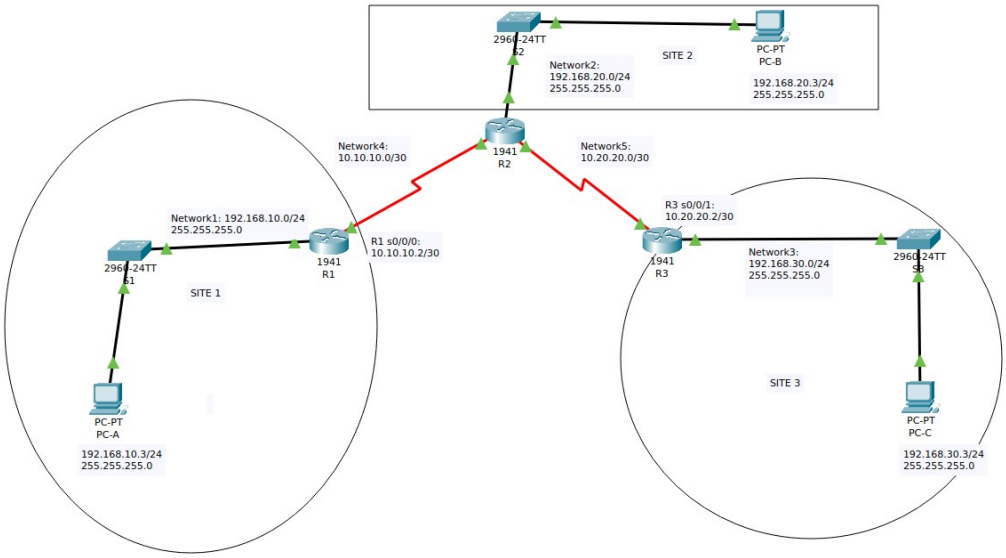
Configure and Verify Ipsec – VPN from site – to – site



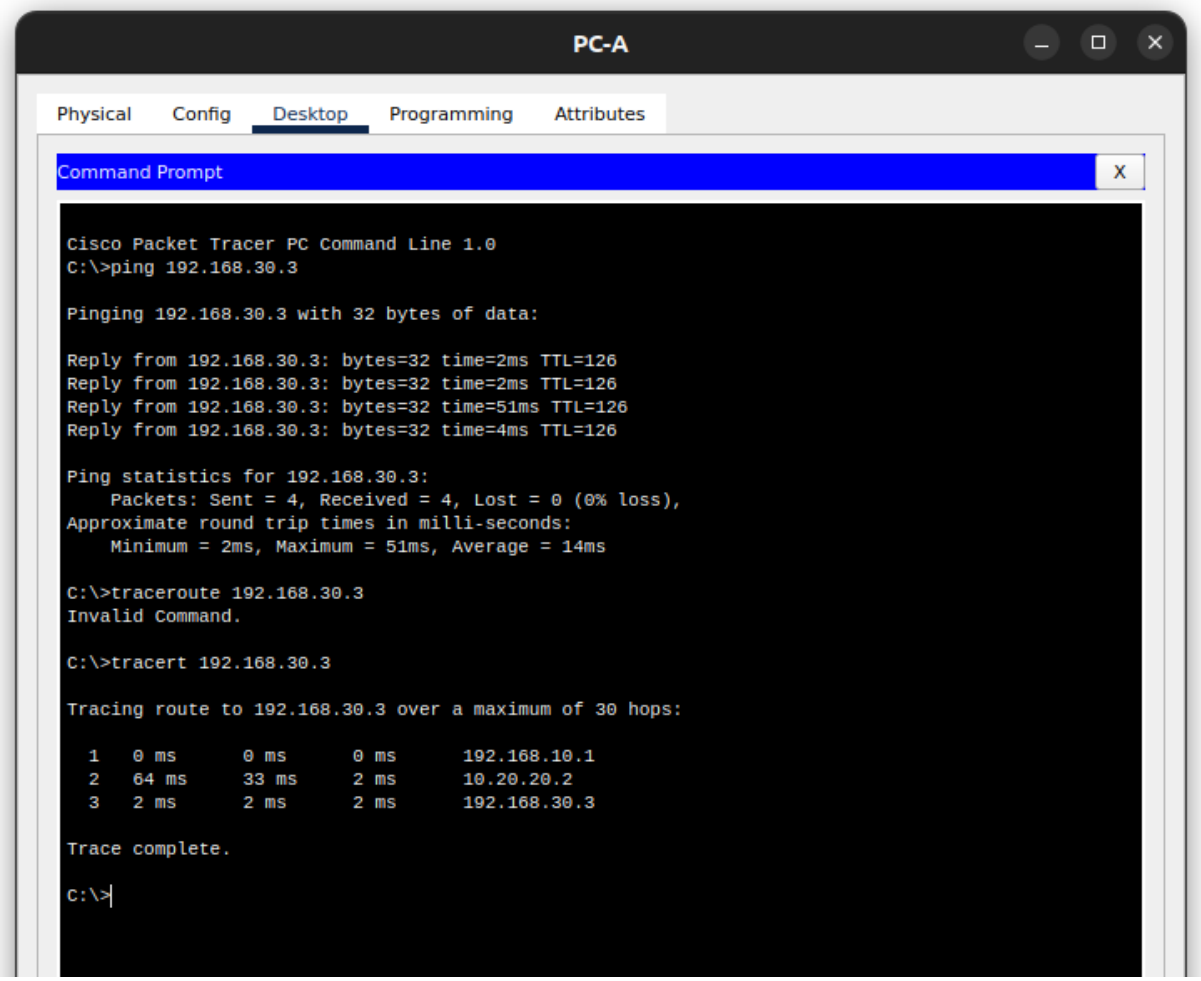
Addressing Table:

Device	Interface	IP Address	Submet Mask	Default Gateway
R1	G0/0	192.168.10.1	255.255.255.0	n/a
	S0/0/0/	10.10.10.2	255.255.255.252	n/a
R2	G0/0	192.168.20.1	255.255.255.0	n/a
	S0/0/0	10.10.10.1	255.255.255.252	n/a
	S0/0/1	10.20.20.1	255.255.255.252	n/a
R3	G0/0	192.168.30.1	255.255.255.0	n/a
	S0/0/1	10.20.20.2	255.255.255.252	n/a
PC-A	NIC	192.168.10.3	255.255.255.0	192.168.10.1
PC-B	NIC	192.168.20.3	255.255.255.0	192.168.20.1
PC-C	NIC	192.168.30.3	255.255.255.0	192.168.30.1

Note: The routers are being configured with **Open Shortest Path Finder 10 (area 0).**

Objectives:
- Verify Connectivity throughout the network
- Configure IPSec tunnel R1 to support a site to site with R3

Step 1: Config IPsec Parameters On R1
1. Test Connectivity (ping from PC-A to PC-C)
2. Perfom a Traceroute from PC-A to PC-C. For example. tracert 192.168.30.3



Step 2: Enable Security Technology Package
1. On R1, issue the **show version** command to view the Security Technology Package license information.
2. If the Security Package has not been enabled, use the following command to enable the package.

*R1(config)# **license boot module c1900 technology-package securityk9***

3. Accept the end-user license agreement.
4. Save the running-config and **reload** the router to enable the security license.
5. Verify that the Security Technology Package has been enabled by using the **show version** command.

```
License Info:

License UDI:

-------------------------------------------------
Device#   PID                  SN
-------------------------------------------------
*0        CISCO1941/K9         FTX1524627S-


Technology Package License Information for Module:'c1900'

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

Configuration register is 0x2102
```

Step 3: Identify Traffic on R1

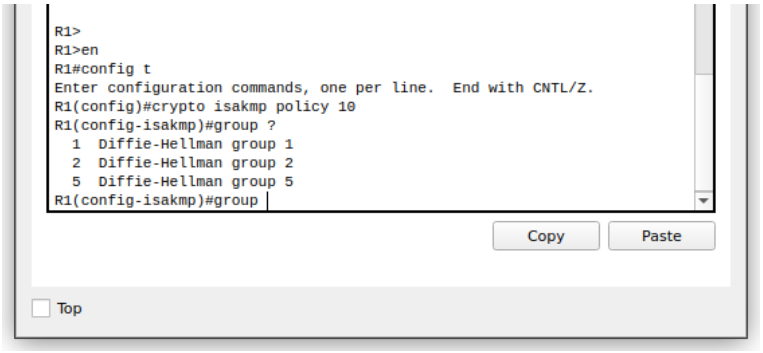
1. Configure ACL 100 to identify the traffic from LAN on R1 to the LAN on R3 as interesting. This interesting traffic will trigger the IPsec VPN to be implemented when there is traffice b/w the R1 to R3 LANs. All other traffic sourced from the LANs will not be encrypted. Due to the implicit **deny all.**

R1(config)# ***access-list 100 permit ip 192.168.10.0 0.0.0.255 192.168.30.0 0.0.0.255***

Step 4: Configure The IKE Phase 1 ISAKMP Policy on R1.

1. Configure the **Crypto ISAKMP policy 10** properties on R1 along with the shared crypto key **vpn.** Refer to the ISAKMP Phase 1 table for the specific parameters to configure. We configured only the encryption, key exchange method, and DH method must be configured.

Note: As shown in figure below, the highest DH group currently supported by Packet Tracer is **group 5.**



R1(config)# crypto isakmp policy 10
R1(config-isakmp)# encryption aes 128
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 5
R1(config-isakmp)# exit
R1(config)# crypto isakmp key vpn address 10.20.20.2

Step 5: Configure IKE Phase 2 IPSec policy on R1

1. Create the transform-set VPN-P2 to use **esp-aes** and esp-sha-hmac.
R1(config)# ***crypto ipsec transform-set VPN-P2 esp-aes esp-sha-hmac***
2. 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 map VPN-MAP 10 ipsec-isakmp***
*R1(config-crypto-map)# **description VPN connection to R3***
*R1(config-crypto-map)# **set peer 10.20.20.2***
*R1(config-crypto-map)# **set transform-set VPN-P2***
*R1(config-crypto-map)# **match address 100***
*R1(config-crypto-map)# **exit***

- bind the **VPN-MAP** crypto map to the outgoing Serial 0/0/0 interface.
*R1(config)# **interface s0/0/0***
*R1(config-if)# **crypto map VPN-MAP***

Step 6: Configure IPsec Parameters on R3
1. Enable the Security Technology package on R3, issue the **show version** command to verify that the Security Technology Package License.
2. Follow **step 2** to enable the license.
3. *R3(config)# **acess-list 100 permit ip 192.168.30.0 0.0.0.255 192.168.10.0 0.0.0.255***
4. Configure IKE Phase 1 ISAKMP properties on R3.
    a. *R3(config)# crypto isakmp policy 10*
    b. R3(config-isakmp)# encryption aes 128
    c. R3(config-isakmp)# authentication pre-share
    d. R3(config-isakmp)# group 5
    e. R3(config-isakmp)# exit
    f. R3(config)# crypto isakmp key vpn address 10.10.10.2

5. Configure IKE Phase 2 IPSec Policy on R3
- 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.
    a. R3(config)# crypto map VPN-MAP 10 ipsec-isakmp
    b. R3(config-crypto-map)# description VPN connection to R1
    c. R3(config-crypto-map)# set peer 10.10.10.2
    d. R3(config-crypto-map)# set transform-set VPN-P2
    e. R3(config-crypto-map)# match address 100
    f. R3(config-crypto-map)# exit

Step 7: Configure the crypto map on the outgoing
- Bind the VPN-MAP crypto map to the outgoing Serial 0/0/1 interface. Note: This is not graded.
    a. R3(config)# interface s0/0/1
    b. R3(config-if)# crypto map VPN-MAP

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>en
R1#show crypto ipsec sa

interface: Serial0/0/0
    Crypto map tag: VPN-MAP, local addr 10.10.10.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.168.10.0/255.255.255.0/0/0)
   remote  ident (addr/mask/prot/port):
(192.168.30.0/255.255.255.0/0/0)
   current_peer 10.20.20.2 port 500
    PERMIT, flags={origin_is_acl,}
   #pkts encaps: 16, #pkts encrypt: 16, #pkts digest: 0
   #pkts decaps: 14, #pkts decrypt: 14, #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.: 10.10.10.2, remote crypto endpt.:10.20.20.2
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
     current outbound spi: 0x0(0)

     inbound esp sas:

 --More--
```

Step 2: Create interesting traffic.

Ping PC-C from PC-A.

```
C:\>ping 192.168.10.3

Pinging 192.168.10.3 with 32 bytes of data:

Reply from 192.168.10.3: bytes=32 time=42ms TTL=126
Reply from 192.168.10.3: bytes=32 time=2ms TTL=126
Reply from 192.168.10.3: bytes=32 time=2ms TTL=126
Reply from 192.168.10.3: bytes=32 time=49ms TTL=126

Ping statistics for 192.168.10.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 49ms, Average = 23ms

C:\>
```

Top

Step 3: Verify the tunnel after interesting traffic.

On R1, re-issue the show crypto ipsec sa command. Notice that the number of packets is more than 0, which indicates that the IPsec VPN tunnel is working.

Step 5: Verify the tunnel.

On R1, re-issue the show crypto ipsec sa command. Notice that the number of packets has not changed, which verifies that uninteresting traffic is not encrypted.