**Question1: Configure Secure Passwords and SSH**

**🔹 Part 1: Configure RTA (Router)**

1. **Configure IP on PCA:**

IP: 172.16.1.10

Subnet Mask: 255.255.255.0

Gateway: 172.16.1.1

1. **Console into RTA from PCA Terminal**
2. **Router Configuration:**

enable

configure terminal

hostname RTA

interface g0/0/0

ip address 172.16.1.1 255.255.255.0

no shutdown

exit

service password-encryption

security passwords min-length 10

enable secret StrongSecret123

no ip domain-lookup

ip domain-name netsec.com

username admin secret AdminPass123

crypto key generate rsa

(Enter) → Choose 1024

login block-for 180 attempts 4 within 120

line vty 0 4

transport input ssh

login local

exec-timeout 6

end

write memory

1. **Test SSH from PCA Command Prompt:**

ssh -l admin 172.16.1.1

username : Admin

password : AdminPass123

**🔹 Part 2: Configure SW1 (Switch)**

1. **Console into SW1 from PCA Terminal**
2. **Switch Configuration:**

enable

configure terminal

hostname SW1

interface vlan 1

ip address 172.16.1.2 255.255.255.0

no shutdown

exit

ip default-gateway 172.16.1.1

interface range f0/2-24, g0/2

shutdown

exit

service password-encryption

enable secret StrongSwitch123

no ip domain-lookup

ip domain-name netsec.com

username switchadmin secret SwitchPass123

crypto key generate rsa

(Enter) → Choose 1024

line vty 0 4

transport input ssh

login local

exec-timeout 6

end

write memory

**Question 2: Configure Server-based Authentication with TACACS+ and RADIUS**

**Part 1: TACACS+ on R2**

**Step 1: Test connectivity**

* Ping between PCs as required (done on PCs, not shown here).

**Step 2: Configure local backup user on R2**

R2(config)# username Admin2 secret admin2pa55

**Step 3: Configure TACACS+ server details**

R2(config)# tacacs-server host 192.168.2.2

R2(config)# tacacs-server key tacacspa55

**Step 4: Enable AAA and configure login authentication**

R2(config)# aaa new-model

R2(config)# aaa authentication login default group tacacs+ local

**Step 5: Apply AAA authentication to console line**

R2(config)# line console 0

R2(config-line)# login authentication default

R2(config-line)# exit

**Step 6: Verify by logging in on console or via vty**

**Part 2: RADIUS on R3**

**Step 1: Configure local backup user on R3**

R3(config)# username Admin3 secret admin3pa55

**Step 2: Configure RADIUS server details**

R3(config)# radius-server host 192.168.3.2

R3(config)# radius-server key radiuspa55

**Step 3: Enable AAA and configure login authentication**

R3(config)# aaa new-model

R3(config)# aaa authentication login default group radius local

**Step 4: Apply AAA authentication to console line**

R3(config)# line console 0

R3(config-line)# login authentication default

R3(config-line)# exit

**Step 5: Verify by logging in on console or via vty**

Exit

Exit

**Question 3: Configure Local AAA for Console and VTY Access**

**🔹 Part 1: Local AAA for Console Access on R1**

**Step 1: Create local user**

R1(config)# username Admin1 secret admin1pa55

**Step 2: Enable AAA and set local login for console**

R1(config)# aaa new-model

R1(config)# aaa authentication login default local

**Step 3: Apply AAA to console line**

R1(config)# line console 0

R1(config-line)# login authentication default

**Step 4: Test from console (CLI will ask for username/password)**

**🔹 Part 2: Local AAA for VTY (SSH) Access on R1**

**Step 1: Configure domain and RSA key**

R1(config)# ip domain-name netsec.com

R1(config)# crypto key generate rsa

*Choose 1024 bits when prompted*

**Step 2: Create named AAA method for SSH**

R1(config)# aaa authentication login SSH-LOGIN local

**Step 3: Configure vty lines for SSH only**

R1(config)# line vty 0 4

R1(config-line)# login authentication SSH-LOGIN

R1(config-line)# transport input ssh

R1(config-line)# exit

**Step 4: Enable SSH on R1**

R1(config)# ip ssh version 2

**✅ Verification from PC-A**

Open **PC-A > Desktop > Command Prompt**, type:

PC> ssh -l Admin1 192.168.1.1

It will ask:

Password: admin1pa55

If successful, you're logged into R1 using **SSH with local AAA**.

**Question 4: Identify Packet Flow**

**✅ Part 1: Verifying Connectivity**

1. PC0 > Desktop > Web Browser

2. URL: www.cisco.pka → Go

3. URL: www.web.pka → Go

**✅ Part 2: Remote LAN Network Topology**

1. Switch to Simulation Mode (Shift + S)

2. Show All/None → Edit Filters → Select:

- IPv4 > DNS

- Misc > HTTP

3. PC0 > Web Browser → URL: www.web.pka → Go

4. Click Capture / Forward until page loads

5. Break link: Real Time Mode (Shift + R)

- Delete tool > remove link: Switch0 ↔ Switch1

- Wait 30 sec or Fast Forward

6. Use Arrow Tool to exit Delete Mode

7. Simulation Mode (Shift + S)

- Tablet0 > Web Browser → www.web.pka

- Click Auto Capture/Play

**✅ Part 3: WAN Network Topology**

**Step 1: PC0 to** [**www.cisco.pka**](http://www.cisco.pka/)

1. PC0 > Web Browser → www.cisco.pka → Go

2. Capture / Forward to observe HTTP flow

3. Real Time Mode (Shift + R)

- Delete link: Router4 ↔ Router2

4. Simulation Mode (Shift + S)

- Tablet0 > Web Browser → www.cisco.pka

**Step 2: PC1 to** [**www.web.pka**](http://www.web.pka/)

1. PC1 > Desktop > Command Prompt

2. Run command:

tracert www.web.pka

3. Match IPs using:

Router CLI > show ip interface brief

4. Simulation Mode (Shift + S)

- PC1 > Web Browser → www.web.pka → Go

5. Click Capture / Forward to load page

**Question 5: Backup Using NAS (Manual)**

1. Go to NAS OS > Backup > Add Backup > Local Backup

2. Choose source and destination

3. Select "Manual" type

4. Name the backup

5. Start from Backup menu (click gray arrow)

**Question 6: ACL Demonstration**

(((**Recreate ACL 11**

**✅ Step 1: Define ACL 11**

R1(config)# access-list 11 deny 192.168.10.0 0.0.0.255

R1(config)# access-list 11 permit any

**✅ Step 2: Apply ACL 11 on the correct interface**

R1(config)# interface s0/0/0

R1(config-if)# ip access-group 11 out

**✅ Step 3: Verify ACL is active**

R1# show access-lists

R1# show run | include interface|access

**✅ Step 4: Test ACL**

PC1 > Command Prompt:

ping 192.168.30.12 # Should fail (PC4)

ping 192.168.31.12 # Should fail (DNS)

ping 192.168.10.11 # Should succeed (PC2)

ping 192.168.11.10 # Should succeed (PC3)

))))

**✅ Part 1: Verify Local Connectivity and Test ACL**

**🔹 Step 1: Ping local PCs**

PC1 > Command Prompt:

ping 192.168.10.11 # Ping PC2

ping 192.168.11.10 # Ping PC3

✅ Both pings should **succeed** (same router, allowed by default).

**🔹 Step 2: Ping remote devices (before removing ACL)**

ping 192.168.30.12 # Ping PC4

ping 192.168.31.12 # Ping DNS Server

❌ These pings **fail** due to ACL.

**✅ Part 2: Remove the ACL and Repeat the Test**

**🔹 Step 1: View ACL configuration**

R1# show access-lists

R1# show run | include interface|access

✅ Shows:

access-list 11 deny 192.168.10.0 0.0.0.255

access-list 11 permit any

interface s0/0/0

ip access-group 11 out

**🔹 Step 2: Remove ACL from interface**

R1(config)# interface s0/0/0

R1(config-if)# no ip access-group 11 out

**🔹 Step 3: Delete the ACL itself**

R1(config)# no access-list 11

**🔹 Step 4: Test connectivity again**

PC1 > Command Prompt:

ping 192.168.30.12 # Ping PC4

ping 192.168.31.12 # Ping DNS Server

✅ Pings should now **succeed**.

**Question 7: Configure Named Standard IPv4 ACLs**

**✅ Part 1: Configure and Apply a Named Standard ACL**

**🔹 Step 1: Configure ACL named File\_Server\_Restrictions on R1**

R1(config)# ip access-list standard File\_Server\_Restrictions

R1(config-std-nacl)# permit host 192.168.20.4

R1(config-std-nacl)# permit host 192.168.100.100

R1(config-std-nacl)# deny any

🔸 PC1 (192.168.20.4) and Web Server (192.168.100.100) are allowed  
🔸 All others are denied

**🔹 Step 2: Apply ACL to interface F0/1 (toward File Server)**

R1(config)# interface fastEthernet 0/1

R1(config-if)# ip access-group File\_Server\_Restrictions out

**🔹 Step 3: Save config**

R1# write

**✅ Part 2: Verify ACL Implementation**

**🔹 Step 1: Check ACL configuration**

R1# show access-lists

Expected Output:

Standard IP access list File\_Server\_Restrictions

10 permit host 192.168.20.4

20 permit host 192.168.100.100

30 deny any

**🔹 Step 2: Confirm ACL is applied**

R1# show ip interface fastEthernet 0/1

**🔹 Step 3: Test from PCs**

PC0: ping 192.168.200.100 # Should fail

PC1: ping 192.168.200.100 # Should succeed

PC2: ping 192.168.200.100 # Should fail

✅ ACL is working correctly — only allowed hosts reach the file server.

**Question 8. Configure Numbered Standard IPv4 ACLs**

**✅ Part 1: Plan the ACL Implementation**

**🔹 Policies:**

* **R2:** Block 192.168.11.0/24 → 192.168.20.254 (WebServer)
* **R3:** Block 192.168.10.0/24 → 192.168.30.0/24 (PC3 network)

**✅ Part 2: Configure and Apply ACLs**

**🔷 On R2:**

**Step 1: Create ACL 1**

R2(config)# access-list 1 deny 192.168.11.0 0.0.0.255

R2(config)# access-list 1 permit any

**Step 2: Apply ACL on interface facing WebServer (G0/0)**

R2(config)# interface gigabitEthernet 0/0

R2(config-if)# ip access-group 1 out

**Step 3: Verify**

R2# show access-lists

R2# show ip interface gigabitEthernet 0/0

**🔷 On R3:**

**Step 1: Create ACL 1**

R3(config)# access-list 1 deny 192.168.10.0 0.0.0.255

R3(config)# access-list 1 permit any

**Step 2: Apply ACL on interface facing PC3 (G0/0)**

R3(config)# interface gigabitEthernet 0/0

R3(config-if)# ip access-group 1 out

**Step 3: Verify**

R3# show access-lists

R3# show ip interface gigabitEthernet 0/0

**✅ Step 3: Test ACLs**

From PCs:

PC1: ping 192.168.11.10 # Should succeed

PC1: ping 192.168.20.254 # Should succeed

PC2: ping 192.168.20.254 # ❌ Should fail (denied by R2)

PC1: ping 192.168.30.10 # ❌ Should fail (denied by R3)

PC2: ping 192.168.30.10 # Should succeed

PC3: ping 192.168.20.254 # Should succeed

✅ **ACLs are now correctly blocking unwanted traffic per policy**.

**Question 9. Configure Extended ACLs - Scenario 1**

**✅ Part 1: Configure, Apply, and Verify an Extended Numbered ACL**

**🔹 Step 1: Configure ACL 100 on R1**

R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ftp

R1(config)# access-list 100 permit icmp 172.22.34.64 0.0.0.31 host 172.22.34.62

➡️ 172.22.34.64/27 = PC1’s subnet  
➡️ 172.22.34.62 = Server IP

**🔹 Step 2: Apply ACL 100 inbound on interface G0/0**

R1(config)# interface gigabitEthernet 0/0

R1(config-if)# ip access-group 100 in

**🔹 Step 3: Test from PC1**

PC1> ping 172.22.34.62 # ✅ should work

PC1> ftp 172.22.34.62 # ✅ username: cisco, password: cisco

PC1> ping 172.22.34.98 # ❌ should fail (PC2)

**✅ Part 2: Configure, Apply, and Verify an Extended Named ACL**

**🔹 Step 1: Configure Named ACL HTTP\_ONLY**

R1(config)# ip access-list extended HTTP\_ONLY

R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www

R1(config-ext-nacl)# permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62

➡️ 172.22.34.96/28 = PC2’s subnet

**🔹 Step 2: Apply named ACL inbound on interface G0/1**

R1(config)# interface gigabitEthernet 0/1

R1(config-if)# ip access-group HTTP\_ONLY in

**🔹 Step 3: Test from PC2**

PC2> ping 172.22.34.62 # ✅ should work

PC2> open browser → http://172.22.34.62 # ✅ should work

PC2> ftp 172.22.34.62 # ❌ should fail

PC2> ping 172.22.34.66 # ❌ should fail (PC1)

✅ **ACLs now meet the policy**:

* PC1: FTP + ICMP only
* PC2: HTTP + ICMP only
* No PC1 ↔ PC2 direct access

**Question 10: Configure Extended ACLs – Scenario 2**

**Part 1: Configure Port Security**

S1> enable

S1# configure terminal

! Step a: Enable port security on f0/1 and f0/2

S1(config)# interface range f0/1 - 2

S1(config-if-range)# switchport mode access

S1(config-if-range)# switchport port-security

! Step b: Allow only one device per port

S1(config-if-range)# switchport port-security maximum 1

! Step c: Enable sticky MAC address learning

S1(config-if-range)# switchport port-security mac-address sticky

! Step d: Set violation mode to restrict

S1(config-if-range)# switchport port-security violation restrict

! Step e: Disable all unused ports

S1(config)# interface range f0/3 - 24, g0/1 - 2

S1(config-if-range)# shutdown

**Part 2: Verify Port Security**

! a: Ping from PC1 to PC2 (done on PC1 terminal)

! b: Verify MAC addresses in running config

S1# show run | begin interface

! c: Show port security summary

S1# show port-security

S1# show port-security address

! g: Show violations on f0/2

S1# show port-security interface f0/2

Done.

(((((((((((((((((**Is VLAN 1 IP Address Needed?**

Yes — but only **if** you want to **remotely manage the switch (S1)**.

S1(config)# interface vlan 1

S1(config-if)# ip address 10.10.10.2 255.255.255.0

S1(config-if)# no shutdown

S1(config)# ip default-gateway 10.10.10.1 ! optional, for remote access across networks

)))))))))))))))))))))))

**Question 11: Layer 2 VLAN Security**

**✅ PART 1: VERIFY CONNECTIVITY**

**Step 1: Ping from C2 to C3 (Both in VLAN 10)**

> ping [C3 IP Address]

✅ **Expected: Successful pings** – Both are in same VLAN 10.

**Step 2: Ping from C2 (VLAN 10) to D1 (VLAN 5)**

> ping [D1 IP Address]

✅ **Expected: Unsuccessful pings** – They are in different VLANs unless routed.

**✅ PART 2: CREATE REDUNDANT LINK BETWEEN SW-1 AND SW-2**

**Step 1: Connect SW-1 to SW-2**

* Use **crossover cable**:
  + SW-1 F0/23 ↔ SW-2 F0/23

**Step 2: Configure trunk on both ends**

**On SW-1:**

enable

conf t

interface fa0/23

description Trunk to SW-2

switchport mode trunk

switchport trunk native vlan 15

switchport nonegotiate

exit

**On SW-2:**

enable

conf t

interface fa0/23

description Trunk to SW-2

switchport mode trunk

switchport trunk native vlan 15

switchport nonegotiateexit

**✅ PART 3: ENABLE VLAN 20 AS MANAGEMENT VLAN**

**Step 1: Create VLAN 20 on SW-A**

enable

conf t

vlan 20

name MANAGEMENT

interface vlan 20

ip address 192.168.20.2 255.255.255.0

no shutdown

exit

**Step 2: Create VLAN 20 and interface VLAN 20 on other switches**

Repeat similar config on: SW-B, SW-1, SW-2, Central

**Example (SW-B):**

enable

conf t

vlan 20

interface vlan 20

ip address 192.168.20.3 255.255.255.0

no shutdown

exit

**Step 3: Connect Management PC to SW-A F0/1**

* Assign static IP to Management PC:  
  IP: 192.168.20.10  
  Subnet Mask: 255.255.255.0  
  Gateway: 192.168.20.1

**Step 4: Assign SW-A F0/1 to VLAN 20**

interface fa0/1

switchport mode access

switchport access vlan 20

exit

**Step 5: Ping all switches from Management PC**

> ping 192.168.20.2 # SW-A

> ping 192.168.20.3 # SW-B

> ping 192.168.20.4 # SW-1

> ping 192.168.20.5 # SW-2

> ping 192.168.20.6 # Central

✅ **Expected: All should reply successfully.**

**✅ PART 4: MANAGEMENT PC ACCESS TO ROUTER R1**

**Step 1: On R1 - Add subinterface for VLAN 20**

enable

conf t

interface g0/0.3

encapsulation dot1q 20

ip address 192.168.20.1 255.255.255.0

no shutdown

exit

**Step 2: Verify ping from Management PC to Router**

> ping 192.168.20.1

✅ **Expected: Ping should succeed.**

**Step 3: Create ACL to restrict access to VLAN 20**

Let’s say Management PC = 192.168.20.10

conf t

access-list 100 permit ip host 192.168.20.10 192.168.20.0 0.0.0.255

access-list 100 deny ip any 192.168.20.0 0.0.0.255

access-list 100 permit ip any any

interface g0/0

ip access-group 100 in

exit

**Step 4: Security Verification**

**a. SSH from Management PC:**

> ssh -l SSHadmin 192.168.20.1

Password: ciscosshpa55

✅ **Expected: Login success**

**b. Ping switches and router from Management PC**

> ping 192.168.20.2

> ping 192.168.20.3

> ping 192.168.20.1

✅ **Expected: All should reply**

**c. Ping from D1 to Management PC**

> ping 192.168.20.10

❌ **Expected: Ping should fail** – ACL blocks it

**✅ FINAL STEP: VERIFY RESULTS**

* Click **Check Results** in Packet Tracer or similar simulator.
* Ensure completion percentage shows **100%**
* If not, re-check IPs, ACL, and VLAN interfaces.

**Question 12: Configure ASA Basic Settings and Firewall Using the CLI**.

**✅ Part 1: Verify Connectivity and Explore the ASA**

**Step 1: From PC-C, ping router interfaces**

> ping 172.16.3.1 # Should succeed (R3 G0/1)

> ping 209.165.200.225 # Should succeed (R1 G0/0)

🟥 Pings to PC-B, ASA, and DMZ will fail at this point (expected).

**Step 2: View ASA version and interfaces**

asa> enable

asa# show version

asa# show interface ip brief

**Step 3: Explore file system and flash**

asa# show file system

asa# show flash:

**✅ Part 2: Configure Basic ASA Settings**

**Step 1: Set hostname and domain**

asa(config)# hostname NETSEC-ASA

NETSEC-ASA(config)# domain-name netsec.com

**Step 2: Set enable password**

NETSEC-ASA(config)# enable password ciscoenpa55

**Step 3: Set time (optional)**

NETSEC-ASA# clock set 11:00:00 June 26 2025

**Step 4: Configure interfaces**

**a. G1/1 (OUTSIDE)**

NETSEC-ASA(config)# interface g1/3

NETSEC-ASA(config-if)# nameif OUTSIDE

NETSEC-ASA(config-if)# ip address 209.165.200.226 255.255.255.248

NETSEC-ASA(config-if)# security-level 0

NETSEC-ASA(config-if)# no shutdown

**b. G1/2 (INSIDE)**

NETSEC-ASA(config)# interface g1/2

NETSEC-ASA(config-if)# nameif INSIDE

NETSEC-ASA(config-if)# ip address 192.168.1.1 255.255.255.0

NETSEC-ASA(config-if)# security-level 100

NETSEC-ASA(config-if)# no shutdown

**Step 5: Verify interfaces**

NETSEC-ASA# show interface ip brief

NETSEC-ASA# show ip address

**Step 6: Test pings**

* From **PC-B**:

> ping 192.168.1.1 # ✅ Should work

> ping 209.165.200.226 # ❌ Should fail (by default ASA blocks it)

**✅ Part 3: Routing, NAT, and Inspection**

**Step 1: Configure static default route**

NETSEC-ASA(config)# route OUTSIDE 0.0.0.0 0.0.0.0 209.165.200.225

NETSEC-ASA# show route

NETSEC-ASA# ping 10.1.1.1 # R1 S0/0/0

**Step 2: Configure PAT using network object**

NETSEC-ASA(config)# object network INSIDE-NET

NETSEC-ASA(config-network-object)# subnet 192.168.1.0 255.255.255.0

NETSEC-ASA(config-network-object)# nat (INSIDE,OUTSIDE) dynamic interface

NETSEC-ASA(config-network-object)# exit

NETSEC-ASA# show run object

NETSEC-ASA# show nat

* From **PC-B**: Try pinging

> ping 209.165.200.225 # Echo request = translated; reply = blocked (expected)

**✅ Part 4: DHCP, AAA, and SSH**

**Step 1: DHCP server on ASA**

NETSEC-ASA(config)# dhcpd address 192.168.1.5-192.168.1.36 INSIDE

NETSEC-ASA(config)# dhcpd dns 209.165.201.2 interface INSIDE

NETSEC-ASA(config)# dhcpd enable INSIDE

* Change PC-B to **DHCP mode**. It should get IP like 192.168.1.x

**Step 2: Local AAA**

NETSEC-ASA(config)# username admin password adminpa55

NETSEC-ASA(config)# aaa authentication ssh console LOCAL

**Step 3: SSH access config**

NETSEC-ASA(config)# crypto key generate rsa modulus 1024

> When prompted: no

NETSEC-ASA(config)# ssh 192.168.1.0 255.255.255.0 INSIDE

NETSEC-ASA(config)# ssh 172.16.3.3 255.255.255.255 OUTSIDE

NETSEC-ASA(config)# ssh timeout 10

* From **PC-C**:

> ssh -l admin 209.165.200.226

* From **PC-B**:

> ssh -l admin 192.168.1.1

**✅ Part 5: DMZ, Static NAT, and ACLs**

**Step 1: Configure G1/3 (DMZ)**

NETSEC-ASA(config)# interface g1/1

NETSEC-ASA(config-if)# ip address 192.168.2.1 255.255.255.0

NETSEC-ASA(config-if)# nameif DMZ

NETSEC-ASA(config-if)# security-level 70

NETSEC-ASA(config-if)# no shutdown

**Step 2: Static NAT for DMZ server**

NETSEC-ASA(config)# object network DMZ-SERVER

NETSEC-ASA(config-network-object)# host 192.168.2.3

NETSEC-ASA(config-network-object)# nat (DMZ,OUTSIDE) static 209.165.200.227

NETSEC-ASA(config-network-object)# exit

**Step 3: ACL to permit HTTP/ICMP from outside to DMZ server**

NETSEC-ASA(config)# access-list OUTSIDE-DMZ permit icmp any host 192.168.2.3

NETSEC-ASA(config)# access-list OUTSIDE-DMZ permit tcp any host 192.168.2.3 eq 80

NETSEC-ASA(config)# access-group OUTSIDE-DMZ in interface OUTSIDE

**Step 4: Test web access to DMZ server**

* From **PC-C browser**:

http://209.165.200.227

✅ **Expected: DMZ server responds**

**Question 13: Configure and Verify a Site-to-Site IPsec**

**✅ PART 1: Configure IPsec on R1**

**🔹 Step 1: Test connectivity (from PC-A to PC-C)**

On PC-A:

> ping 192.168.3.3

✅ If routing is correctly configured (via OSPF), ping should succeed.

**🔹 Step 2: Enable the Security Technology Package**

On **R1**:

R1# show version

If securityk9 is **not enabled**:

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

R1# write memory

R1# reload

After reload:

R1# show version

Ensure securityk9 is active.

**🔹 Step 3: Identify Interesting Traffic**

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

**🔹 Step 4: Configure IKE Phase 1 (ISAKMP Policy)**

R1(config)# crypto isakmp policy 10

R1(config-isakmp)# encryption aes 256

R1(config-isakmp)# authentication pre-share

R1(config-isakmp)# group 5

R1(config-isakmp)# exit

R1(config)# crypto isakmp key vpnpa55 address 10.2.2.2

**🔹 Step 5: Configure IKE Phase 2 (IPsec Policy)**

R1(config)# crypto ipsec transform-set VPN-SET esp-aes 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 6: Apply Crypto Map to Outgoing Interface**

R1(config)# interface s0/1/0

R1(config-if)# crypto map VPN-MAP

**✅ PART 2: Configure IPsec on R3**

**🔹 Step 1: Enable Security Technology Package**

R3# show version

If needed:

R3(config)# license boot module c1900 technology-package securityk9

R3# write memory

R3# reload

**🔹 Step 2: Configure Interesting Traffic ACL**

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

**🔹 Step 3: Configure IKE Phase 1 (ISAKMP)**

R3(config)# crypto isakmp policy 10

R3(config-isakmp)# encryption aes 256

R3(config-isakmp)# authentication pre-share

R3(config-isakmp)# group 5

R3(config-isakmp)# exit

R3(config)# crypto isakmp key vpnpa55 address 10.1.1.2

**🔹 Step 4: Configure IKE Phase 2 (IPsec)**

R3(config)# crypto ipsec transform-set VPN-SET esp-aes 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)# set transform-set VPN-SET

R3(config-crypto-map)# exit

**🔹 Step 5: Apply Crypto Map to Outgoing Interface**

R3(config)# interface s0/0/1

R3(config-if)# crypto map VPN-MAP

**✅ PART 3: Verify the IPsec VPN**

**🔹 Step 1: Check Tunnel Before Traffic**

R1# show crypto ipsec sa

✅ You should see counters with 0 packets initially.

**🔹 Step 2: Generate Interesting Traffic**

On **PC-A**, ping **PC-C**:

> ping 192.168.3.3

**🔹 Step 3: Re-check Tunnel**

R1# show crypto ipsec sa

✅ You should now see non-zero counters for encrypted and decrypted packets.

**🔹 Step 4: Create Uninteresting Traffic**

On **PC-A**, ping **PC-B** (e.g., 192.168.2.3):

> ping 192.168.2.3

**🔹 Step 5: Verify No Change**

R1# show crypto ipsec sa

✅ Packet counts should **not increase** — proves VPN tunnel only triggers for interesting traffic.

**🔹 Step 6: Final Check**

In Packet Tracer:  
✅ Click **Check Results** or **Grade Activity**  
✅ **Target: 100% Completion**

Let me know if you want the **full configs for R1 and R3** ready for copy-paste!