Course: Cloud and Network Security – C2 – 2025.

Student: Luke Mbogo

Student No: cs-cns09-25076

Sunday, June 23rd, 2025.

Week 4 Assignment 2:

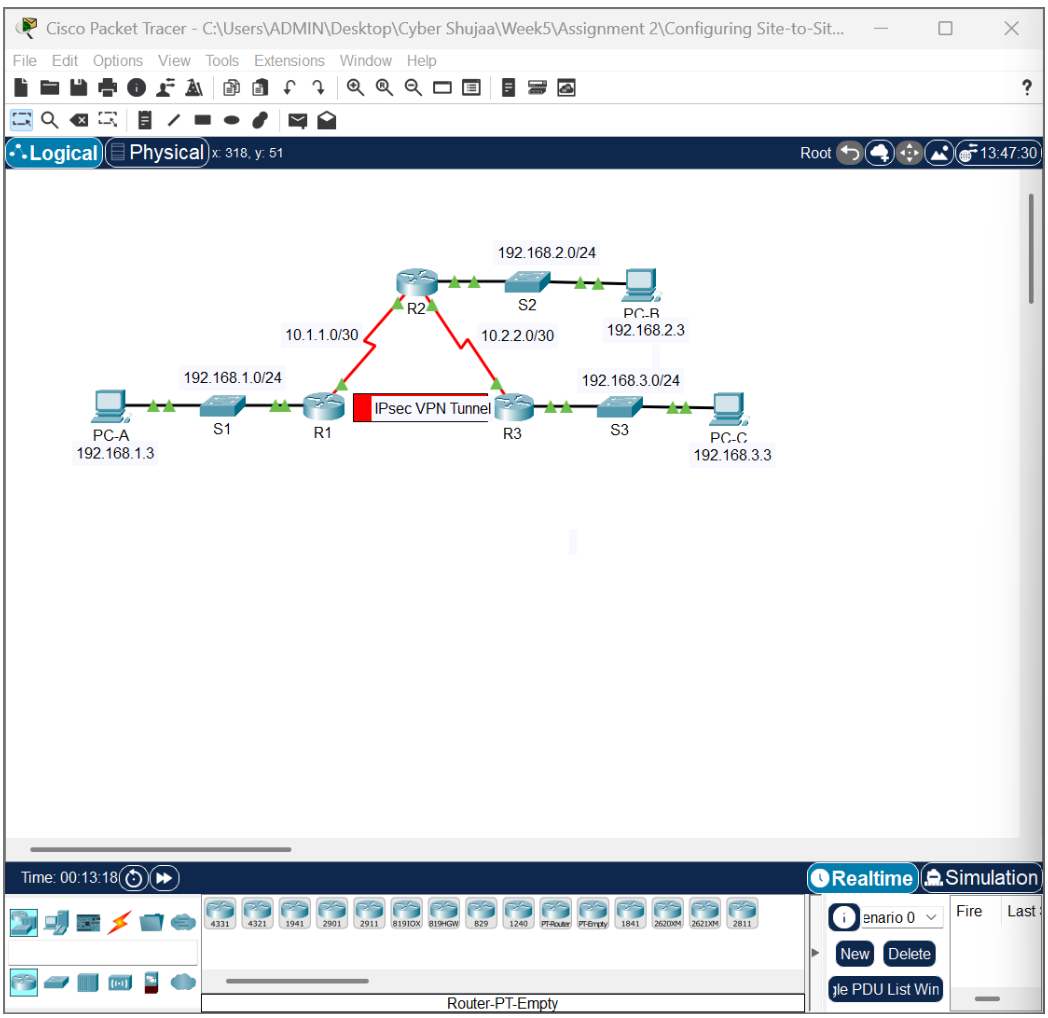
Configuring Site-to-Site VPNs (Packet Tracer).

Contents

Introduction

In this assignment, I configured and verified a **site-to-site IPsec VPN tunnel** between two routers (R1 and R3) using Cisco Packet Tracer. The main objective was to secure communication between two remote LANs (192.168.1.0/24 and 192.168.3.0/24) by encrypting data using IPsec VPN protocols. The configuration process involved enabling the security license, defining ISAKMP and IPsec parameters, identifying interesting traffic using ACLs, and applying crypto maps to WAN interfaces. Once the tunnel was established, I tested and verified its functionality by generating both interesting and uninteresting traffic. The results confirmed that the VPN tunnel only encrypts traffic that meets the defined criteria. This exercise helped me understand how IPsec VPNs function in real network environments and the importance of encryption in protecting data over untrusted networks.

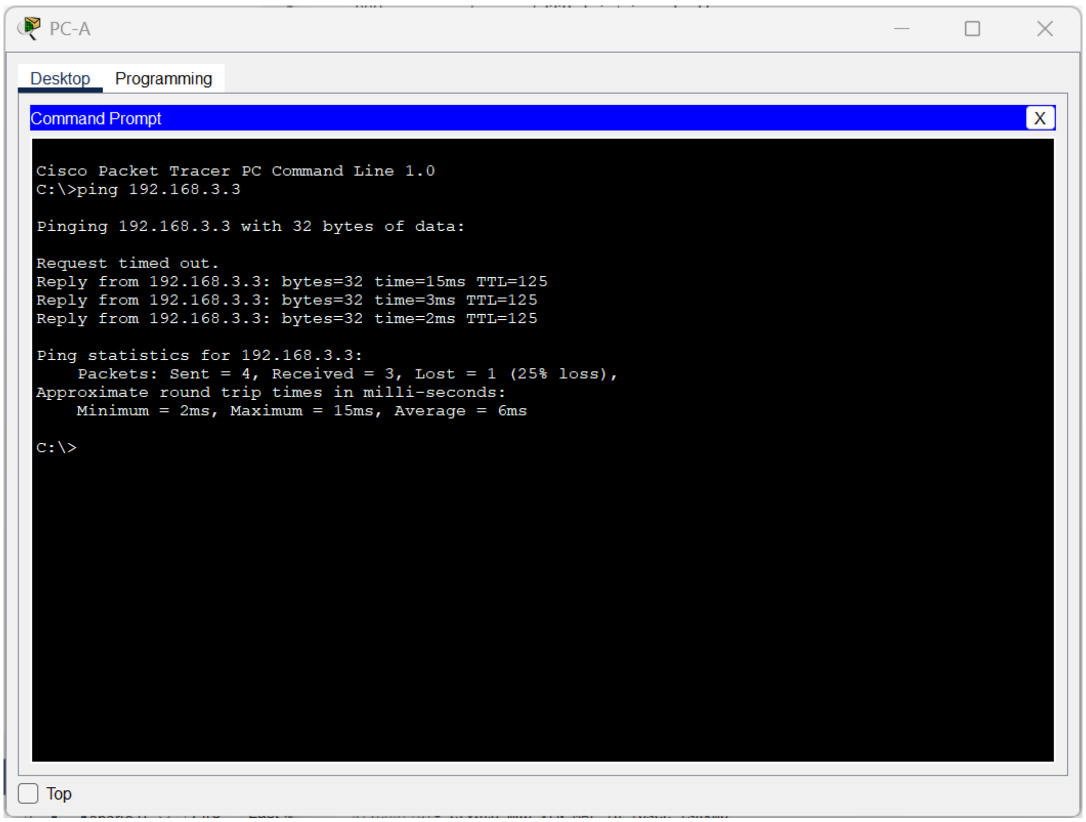
Network Topology



Part 1: Configure IPsec Parameters on R1

Step 1: Test Connectivity

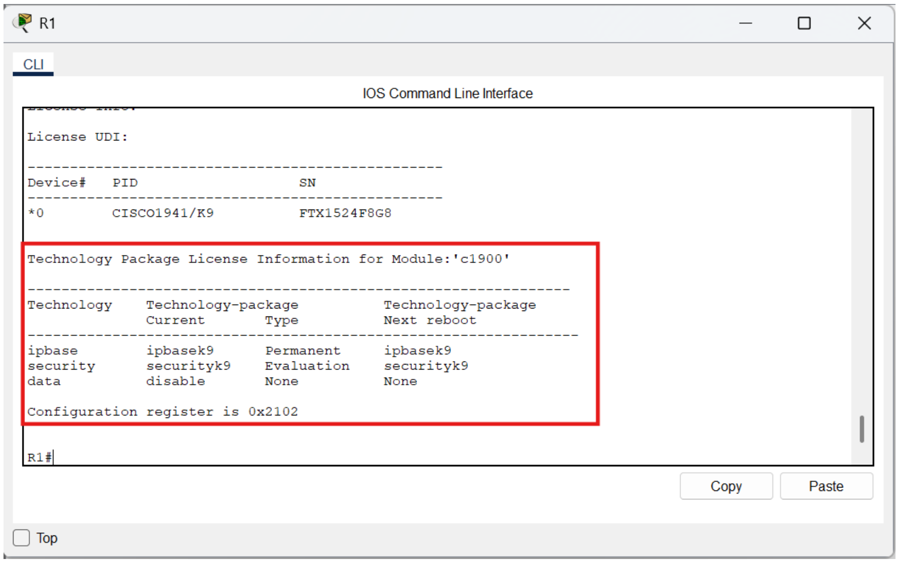
To confirm that the network was functioning before implementing the VPN, I initiated a ping from PC-A to PC-C. The ping was successful, verifying that there was basic end-to-end connectivity between the two LANs. This connectivity served as a baseline to later test whether the IPsec VPN would successfully encrypt traffic between the sites.



Step 2: Enable the Security Technology package.

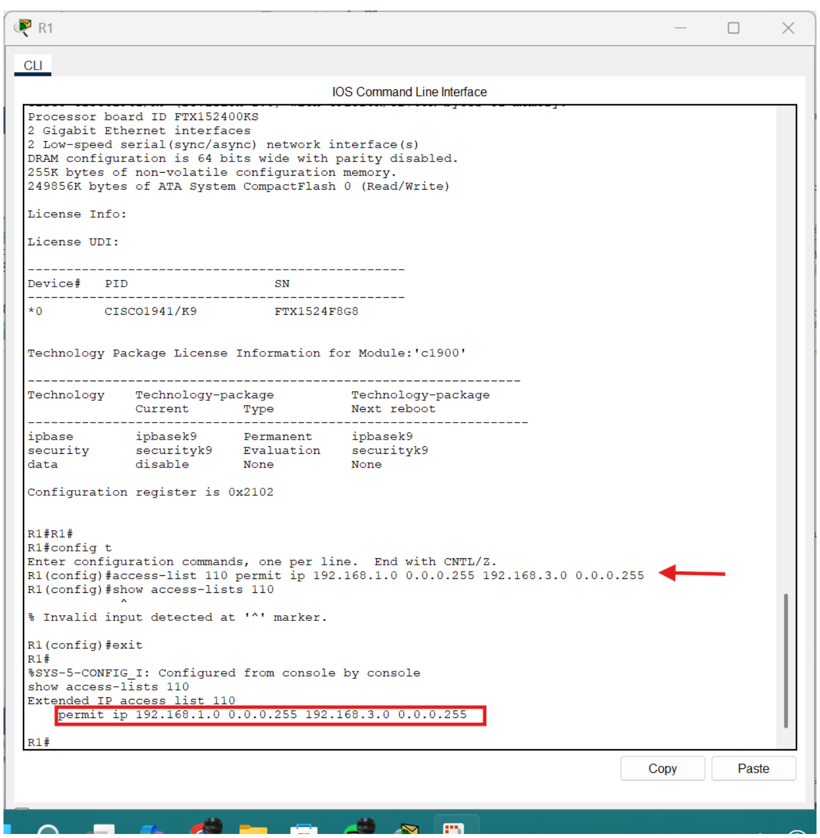
I enabled the **Security Technology Package** on R1 to unlock cryptographic features required for VPN configuration. This was done using the command: [license boot module c1900 technology-package securityk9]

After accepting the end-user license agreement, I saved the running configuration and reloaded the router. Finally, I used the show version command to confirm that the **securityk9 license** was active and set to load on the next reboot. This step ensured the router could support ISAKMP, IPsec, and other security-related features.



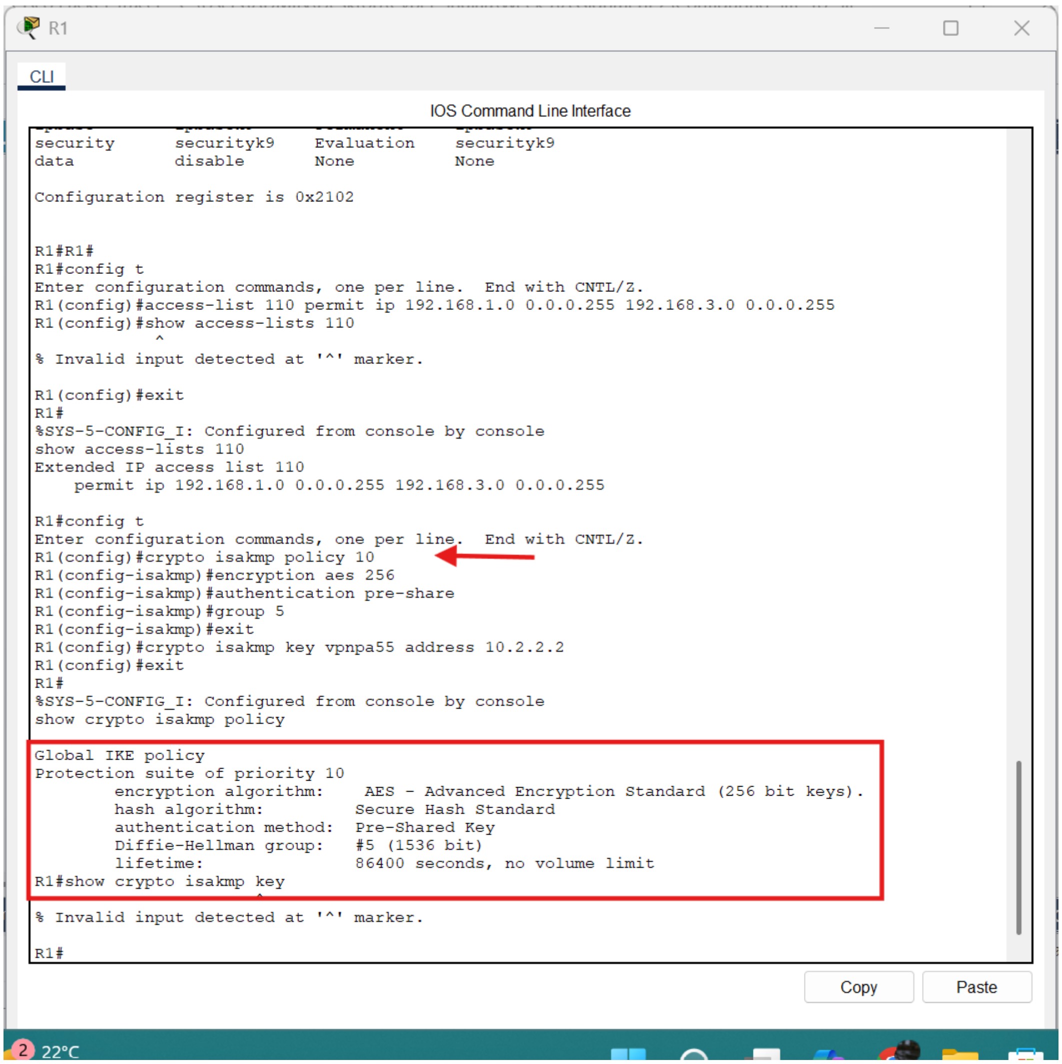
Step 3: Identify interesting traffic on R1.

To define the traffic that should be encrypted by the IPsec VPN, I configured **Access Control List (ACL) 110** on R1. This ACL specifically permitted IP traffic from the **192.168.1.0/24** LAN (R1 side) to the **192.168.3.0/24** LAN (R3 side). This traffic was marked as "interesting," meaning it would trigger the creation of the VPN tunnel. All other traffic was implicitly denied and therefore excluded from encryption, as no explicit deny statement was necessary.



Step 4: Configure the IKE Phase 1 ISAKMP policy on R1.

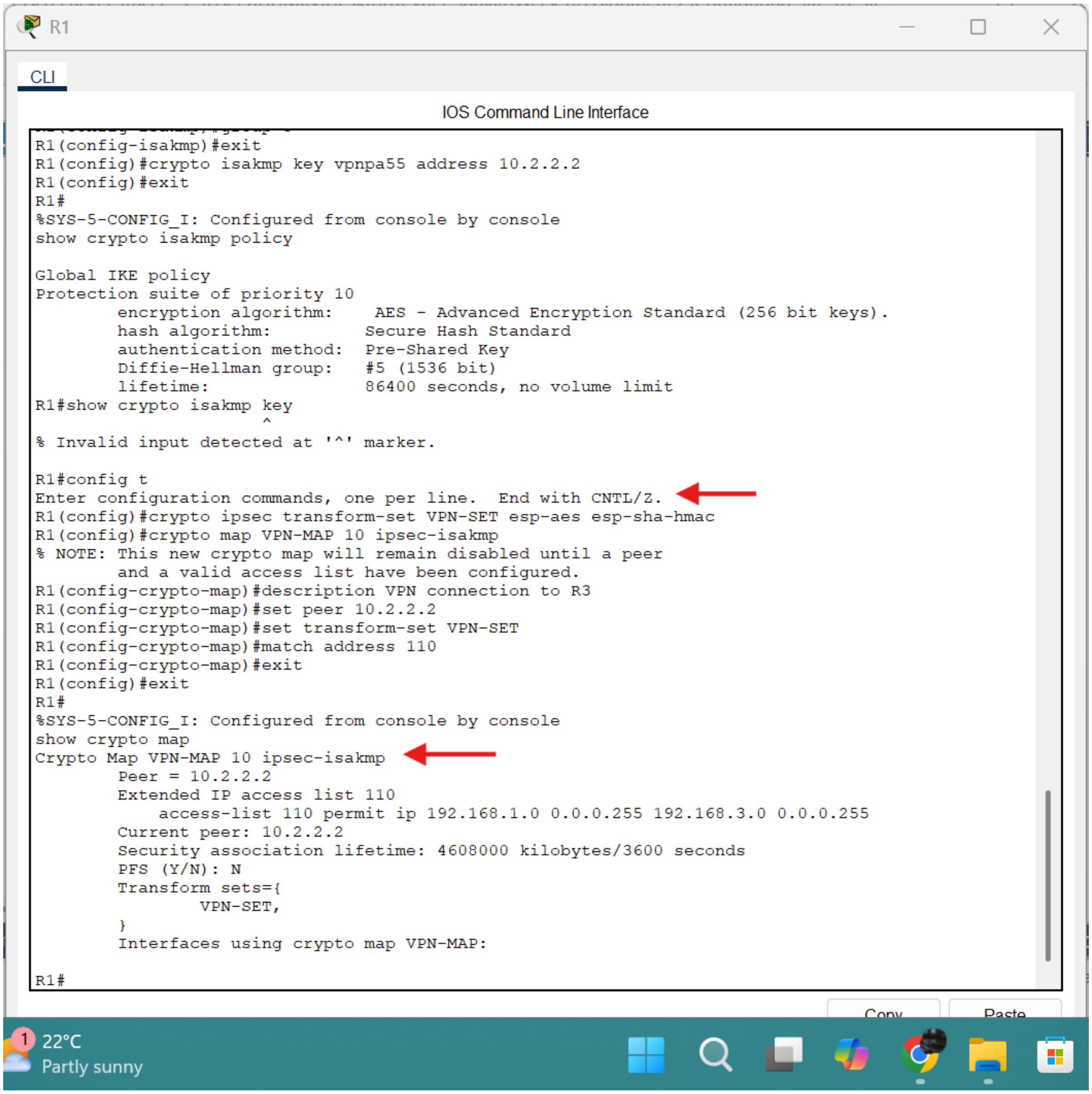
To begin establishing a secure VPN tunnel, I configured **ISAKMP Phase 1** on R1 using **policy 10**. The encryption method was set to **AES 256**, the authentication method to **pre-shared key**, and the **Diffie-Hellman group** to **group 5** (the highest supported by Packet Tracer). I then defined the shared key vpnpa55 and associated it with the IP address of R3's WAN interface (10.2.2.2). This phase ensures secure negotiation of VPN parameters before any actual data is exchanged.




```
security      securityk9   Evaluation    securityk9
data          disable      None          None

Configuration register is 0x2102


R1#R1#
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 110 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
R1(config)#show access-lists 110
            ^
% Invalid input detected at '^' marker.

R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
show access-lists 110
Extended IP access list 110
    permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255

R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
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
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
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:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
R1#show crypto isakmp key
                        ^
% Invalid input detected at '^' marker.

R1#
```

Step 5: Configure the IKE Phase 2 IPsec Policy on R1

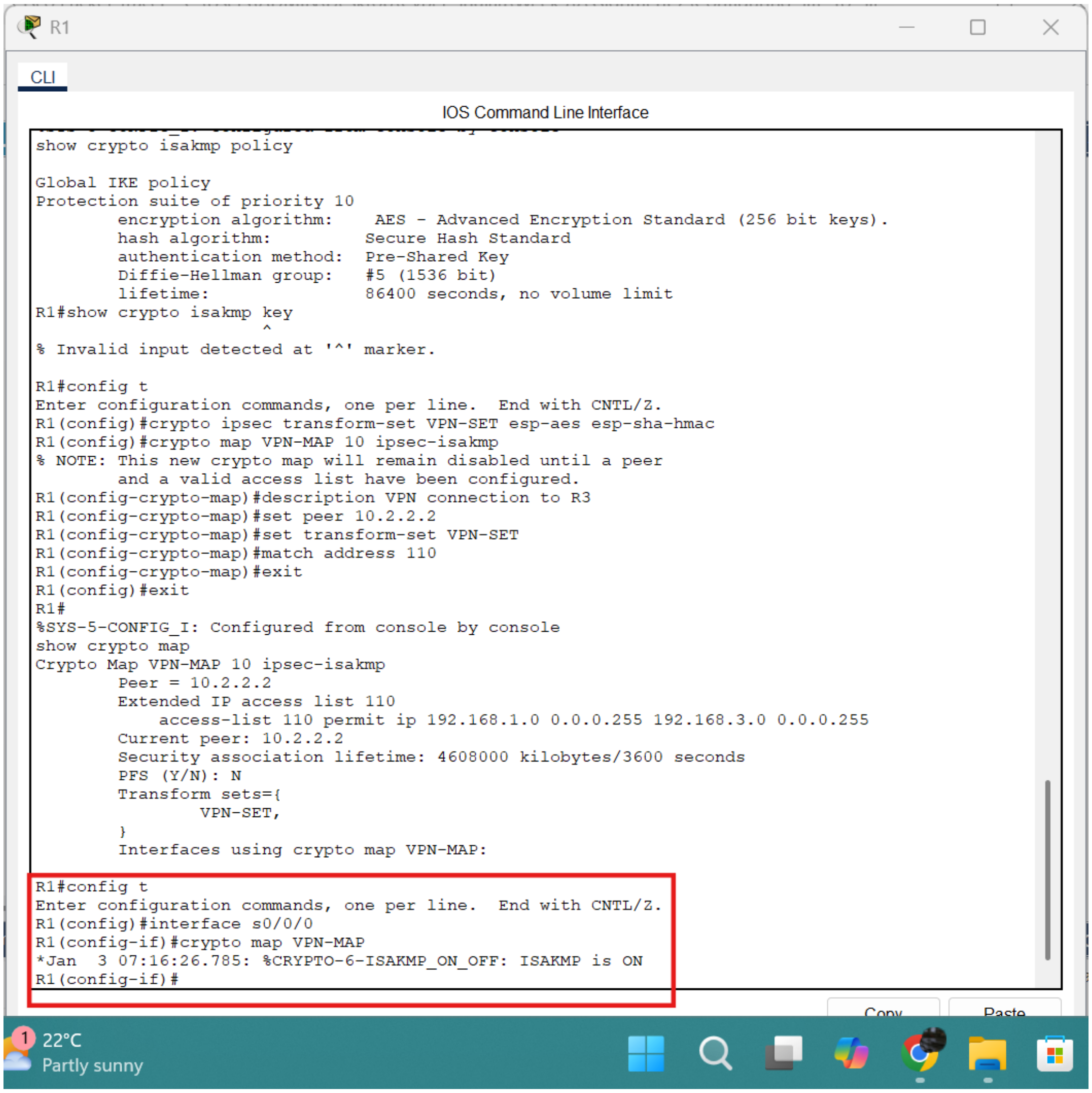
In this step, I configured **IPsec Phase 2** settings to define how the actual data will be encrypted. I created a **transform set named VPN-SET** using esp-aes for encryption and esp-sha-hmac for integrity. Next, I created a **crypto map named VPN-MAP** with sequence number 10 and defined it as an ipsec-isakmp map. Within the map, I set the peer IP to R3's WAN interface (10.2.2.2), specified the transform set to be used, and linked the map to **ACL 110** to apply the policy only to interesting traffic. This configuration ensures that only specific traffic is encrypted according to the IPsec policy.




```
R1(config-isakmp)#exit
R1(config)#crypto isakmp key vpnpa55 address 10.2.2.2
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
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:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
R1#show crypto isakmp key
                            ^
% Invalid input detected at '^' marker.

R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac
R1(config)#crypto map VPN-MAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
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
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
show crypto map
Crypto Map VPN-MAP 10 ipsec-isakmp
        Peer = 10.2.2.2
        Extended IP access list 110
            access-list 110 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
        Current peer: 10.2.2.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                VPN-SET,
        }
        Interfaces using crypto map VPN-MAP:

R1#
```

Step 6: Configure the crypto map on the outgoing interface.

To activate the IPsec configuration, I applied the VPN-MAP crypto map to R1's **outgoing Serial 0/0/0 interface**. This step is essential, as it enables the router to process and encrypt any traffic matching the defined ACL when it leaves through this interface. Once applied, the router began using ISAKMP to establish secure tunnels with the peer, and the IPsec policy became operational for matching traffic.


R1
CLI
IOS Command Line Interface
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: #5 (1536 bit)
lifetime: 86400 seconds, no volume limit
R1#show crypto isakmp key
^
% Invalid input detected at '^' marker.

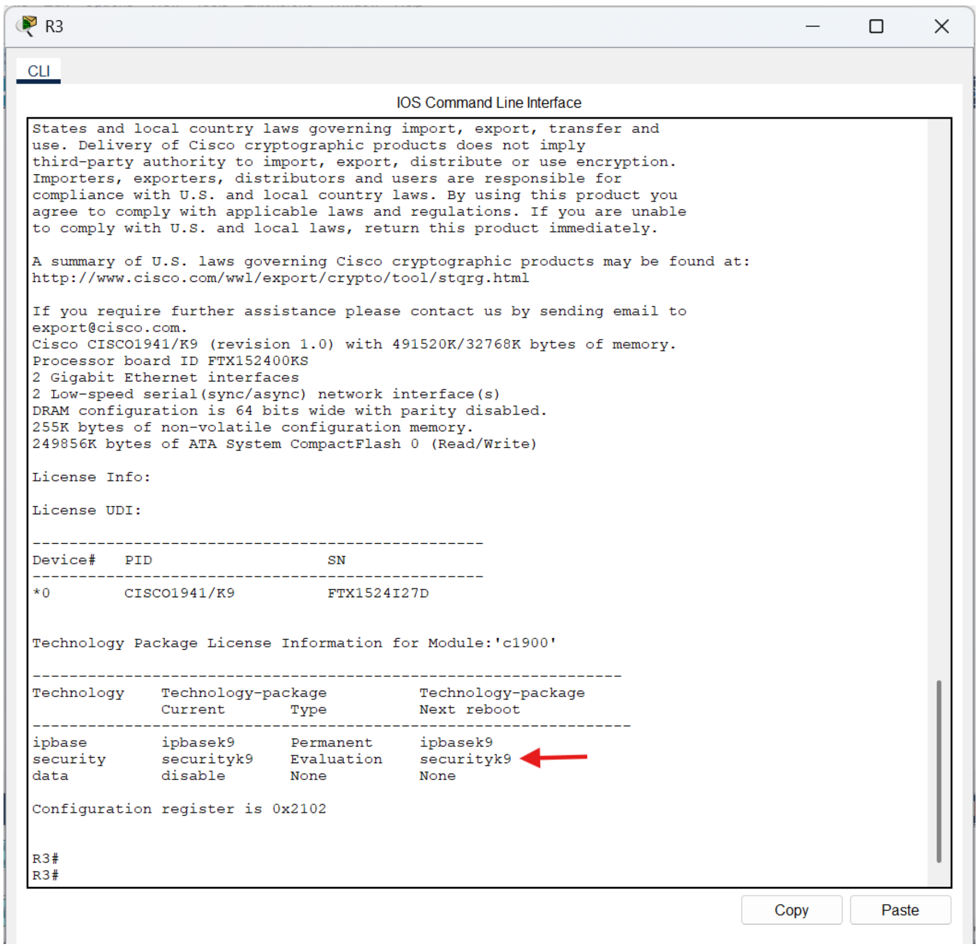
R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac
R1(config)#crypto map VPN-MAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
R1(config-crypto-map)#description 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
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console
show crypto map
Crypto Map VPN-MAP 10 ipsec-isakmp
Peer = 10.2.2.2
Extended IP access list 110
access-list 110 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
Current peer: 10.2.2.2
Security association lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform sets={
VPN-SET,
}
Interfaces using crypto map VPN-MAP:

R1#config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface s0/0/0
R1(config-if)#crypto map VPN-MAP
*Jan 3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R1(config-if)#
22°C
Partly sunny

Part 2: Configure IPsec Parameters on R3

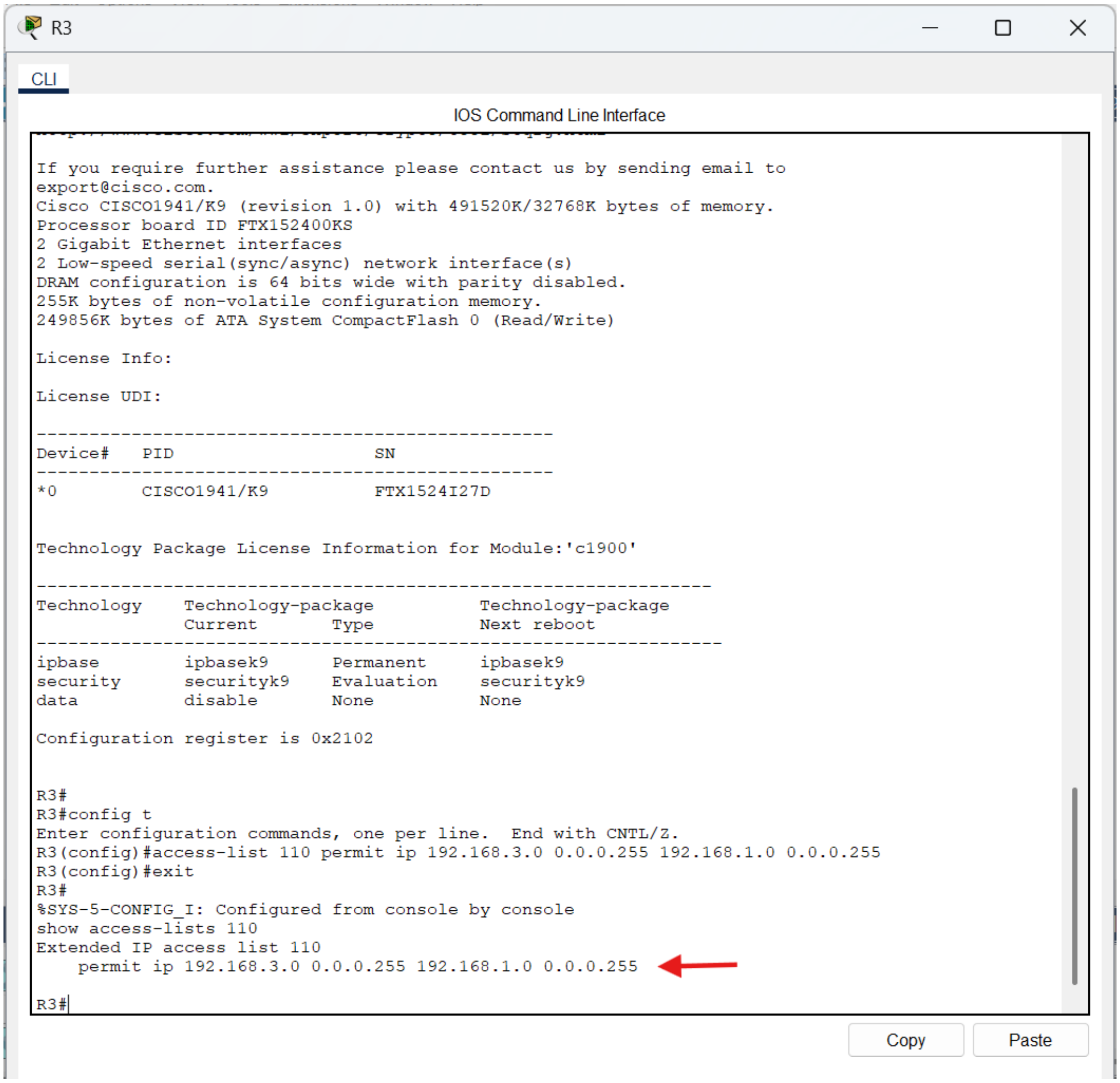
Step 1: Enable the Security Technology package.

On R3, I used the show version command to confirm that the **securityk9 technology package** was active. It was shown as an **evaluation license**, which is sufficient for enabling IPsec VPN features in Packet Tracer. Since the license was already enabled, there was **no need to reload the router**. This step ensured that R3 had the necessary cryptographic capabilities to participate in the VPN tunnel.



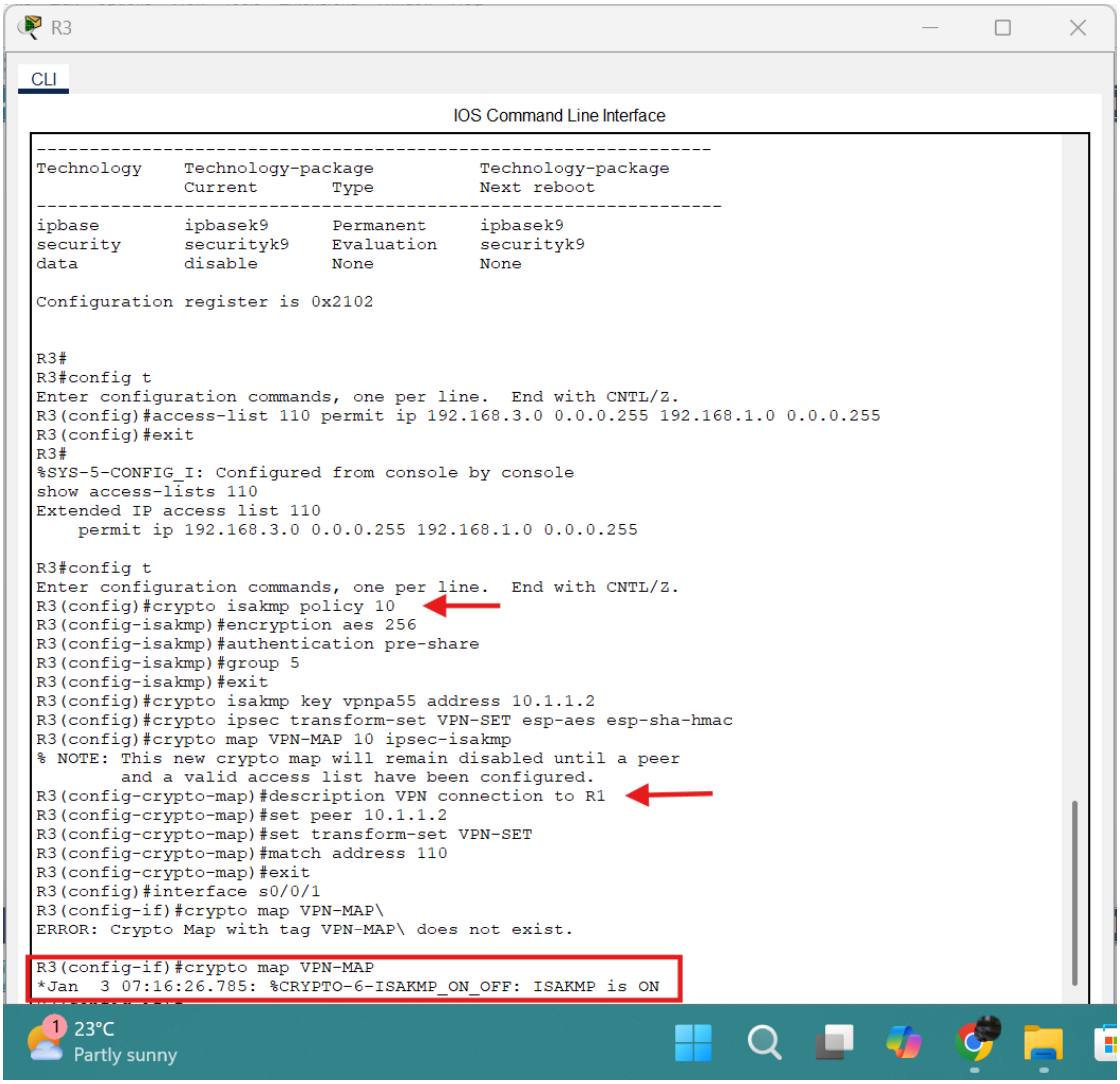
Step 2: Configure router R3 to support a site-to-site VPN with R1.

To mirror the configuration on R1, I created **ACL 110** on R3 to define the **interesting traffic**. This ACL permits IP traffic originating from R3's LAN (**192.168.3.0/24**) destined for R1's LAN (**192.168.1.0/24**). This traffic will trigger the IPsec VPN tunnel when communication occurs between the two networks. Just like on R1, the implicit "deny all" ensures that all other traffic is excluded from encryption.



Step 3: Configure the IKE Phase 1 ISAKMP properties on R3.

To establish a secure VPN tunnel with R1, I configured **ISAKMP policy 10** on R3. The settings matched those on R1 — using **AES 256 encryption**, **pre-shared key authentication**, and **Diffie-Hellman group 5**. I then set the **shared key vpnpa55** and associated it with R1's WAN IP address (**10.1.1.2**). This step completed the IKE Phase 1 setup, allowing secure negotiation of tunnel parameters between R1 and R3.

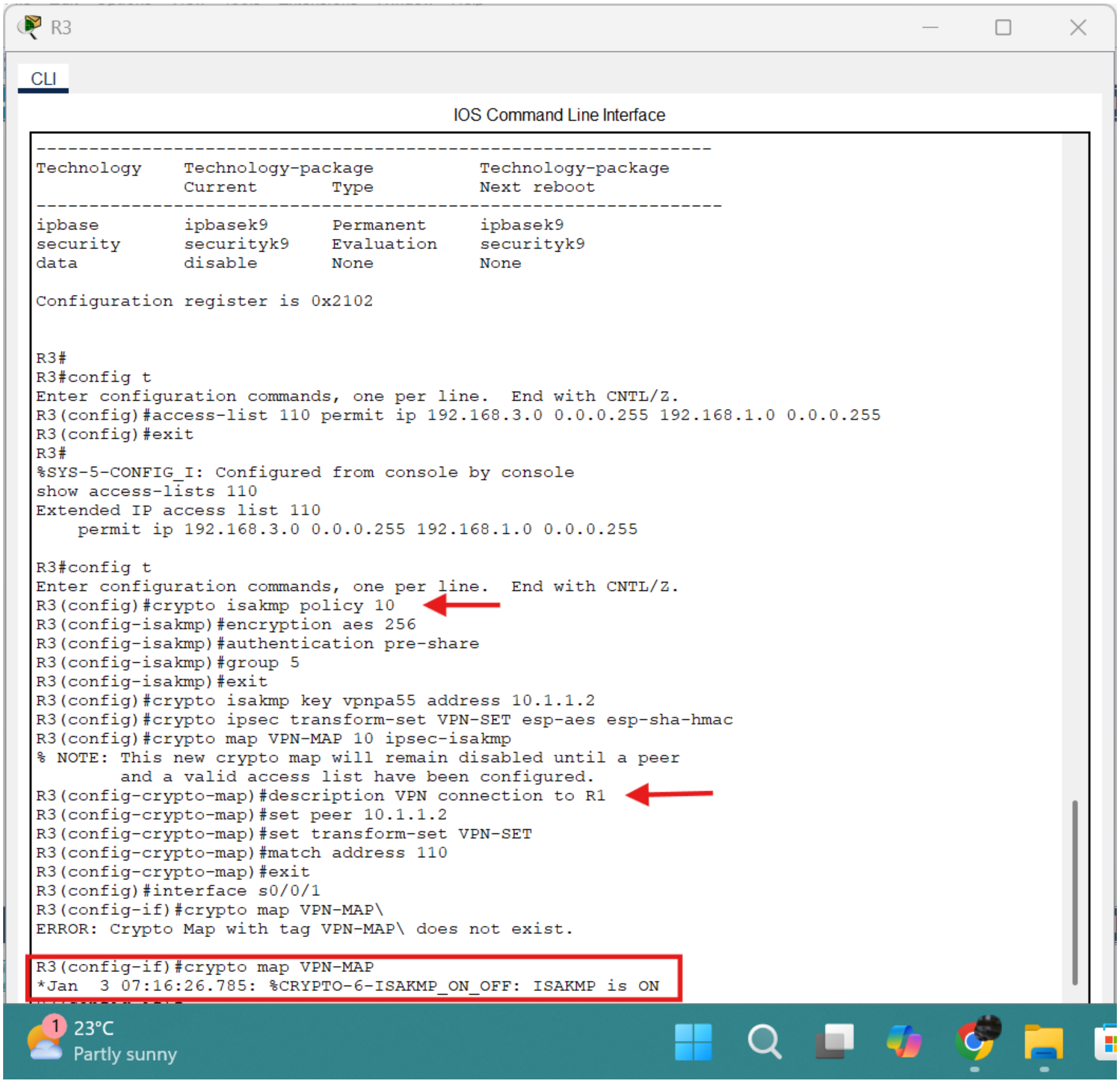


Step 4 & 5: Configure the IKE Phase 2 IPsec Policy and Apply the Crypto Map on R3

To complete the VPN setup on R3, I first configured the **IPsec transform-set** named VPN-SET to use esp-aes for encryption and esp-sha-hmac for integrity. Then, I created a **crypto map named VPN-MAP** using sequence number 10 and identified it as an ipsec-isakmp map. Within the crypto map, I:

- Added a description for clarity,
- Set the **peer IP to R1's WAN address (10.1.1.2)**,
- Applied the previously created transform set,
- And matched **ACL 110** to define interesting traffic.

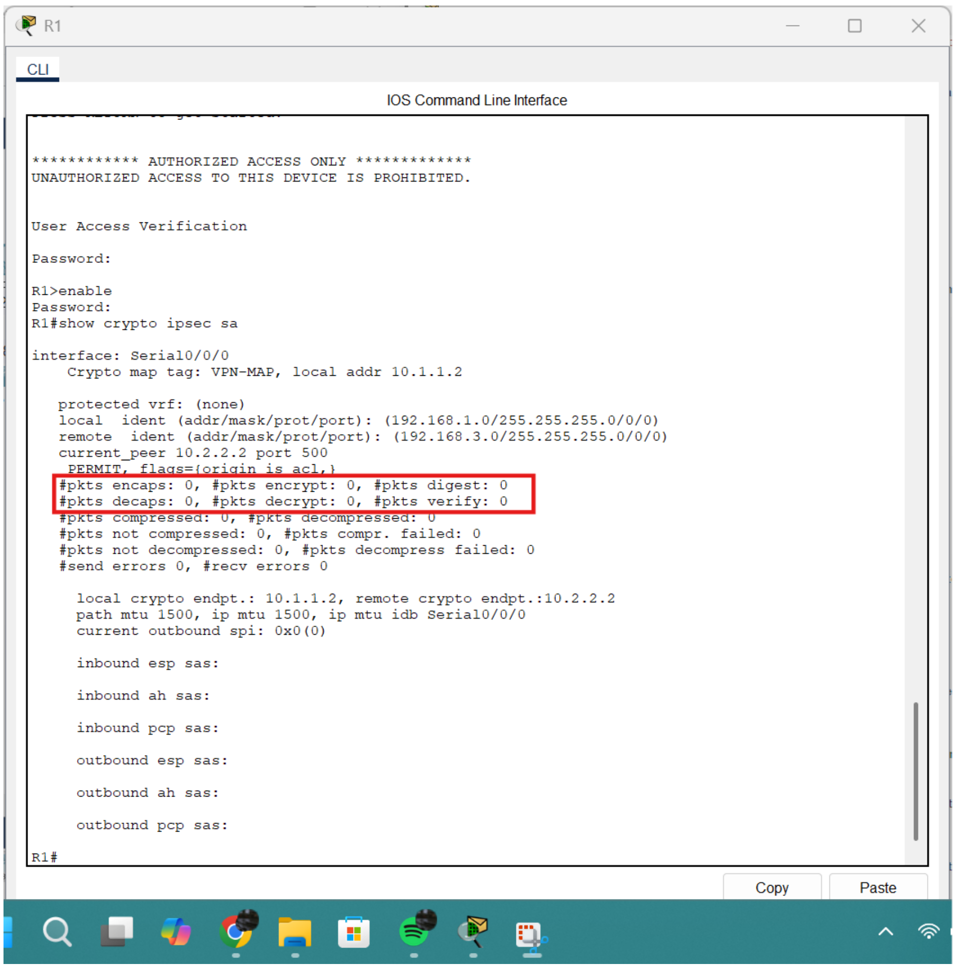
Finally, I **bound the crypto map to the outgoing Serial 0/0/1 interface**, which enabled the router to process and encrypt traffic through the VPN tunnel whenever matching traffic is detected.



Part 3: Verify the IPsec VPN

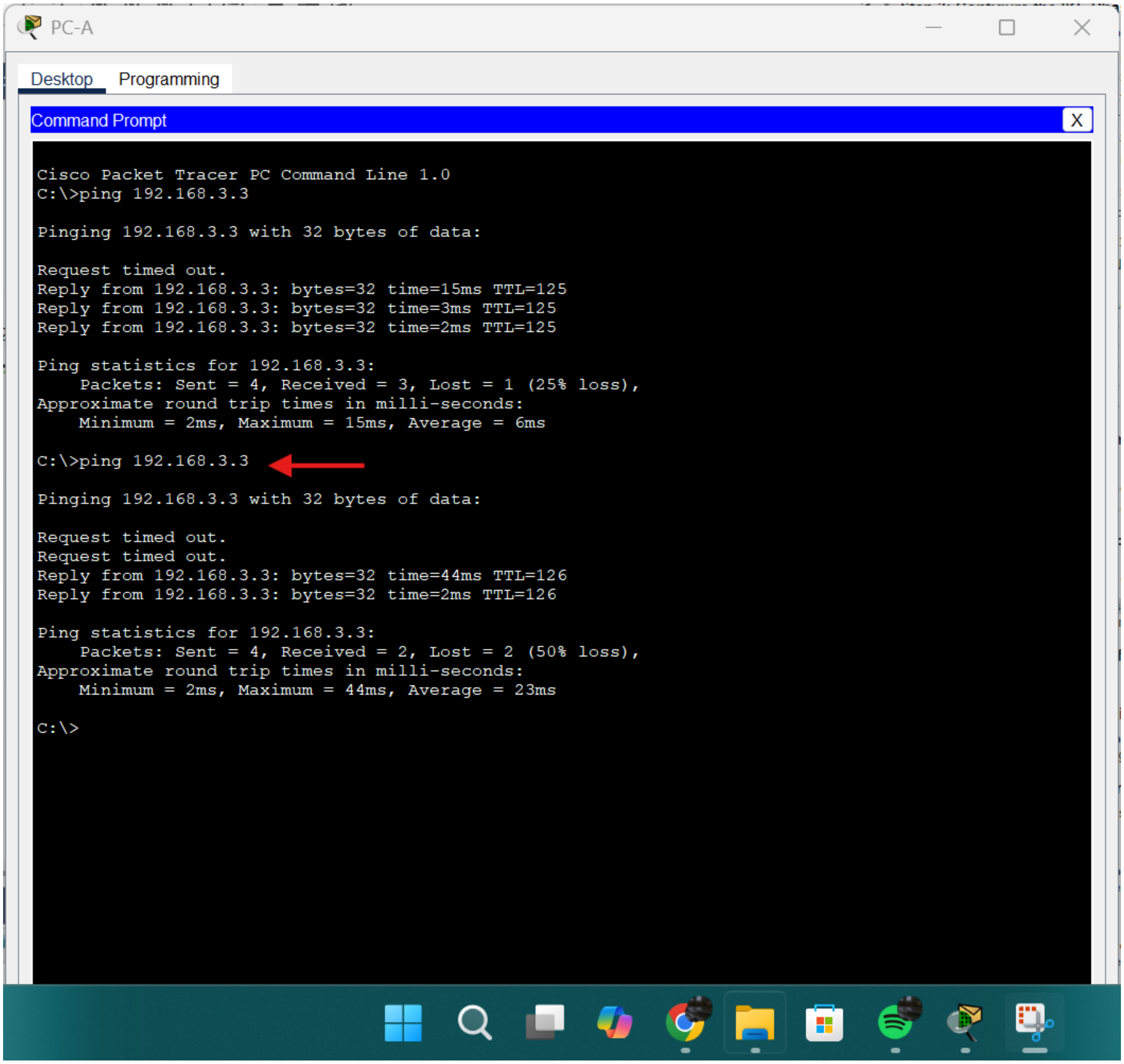
Step 1: Verify the tunnel prior to interesting traffic.

Before generating any traffic between the LANs, I ran the command show crypto ipsec sa on R1. At this stage, all key counters — including encapsulated, encrypted, decapsulated, and decrypted packets — were set to 0. This confirmed that no VPN traffic had passed through the tunnel yet, which is the expected behavior before any interesting traffic is detected.



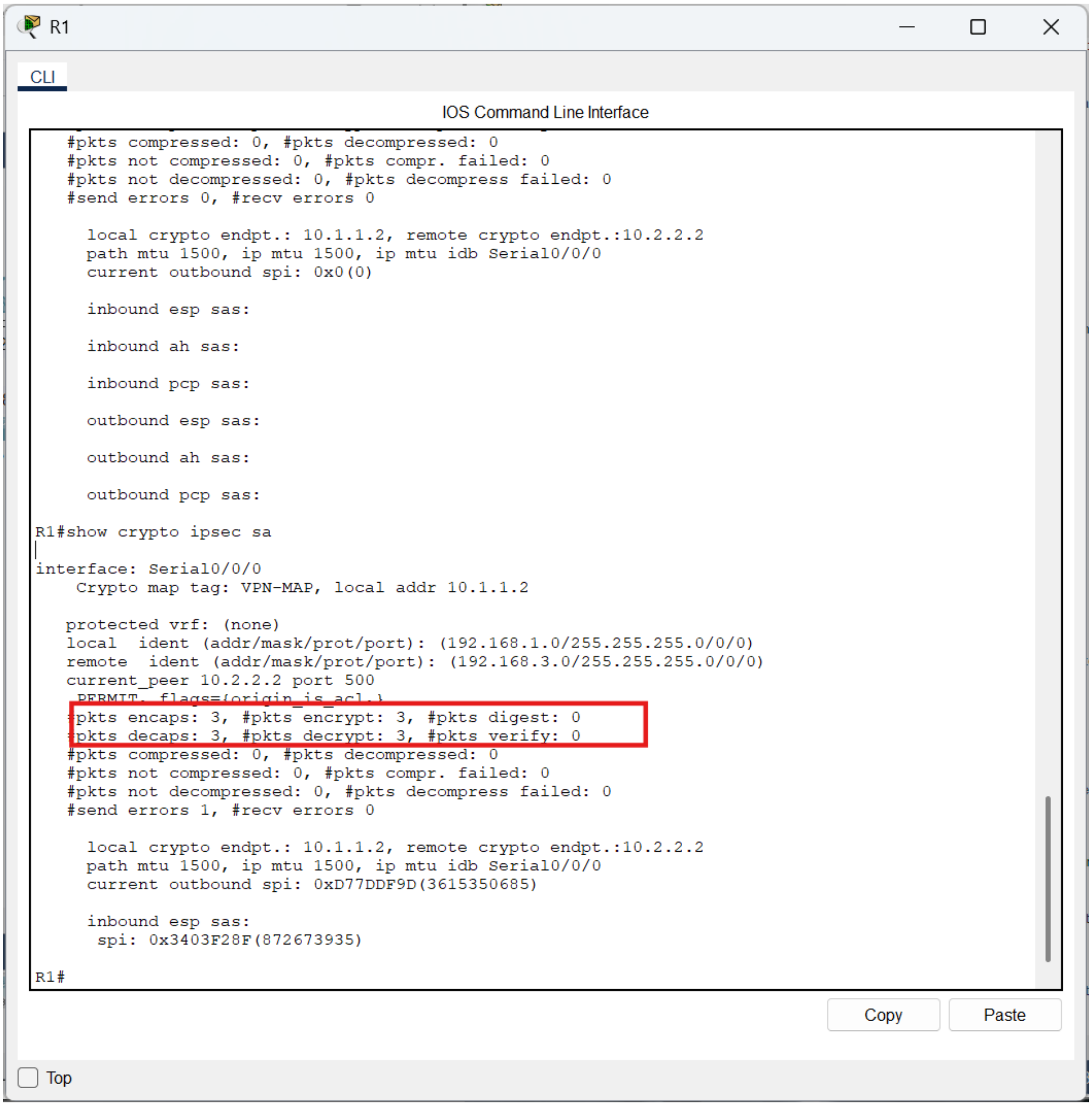
Step 2: Create interesting traffic.

To trigger the IPsec VPN tunnel, I initiated a **ping from PC-A to PC-C**, which generates traffic between the **192.168.1.0/24** and **192.168.3.0/24** networks. Since this traffic matches the criteria defined in ACL 110 on both routers, it is considered **interesting traffic** and activates the IPsec tunnel. The ping was successful, indicating that the tunnel was initiated properly.



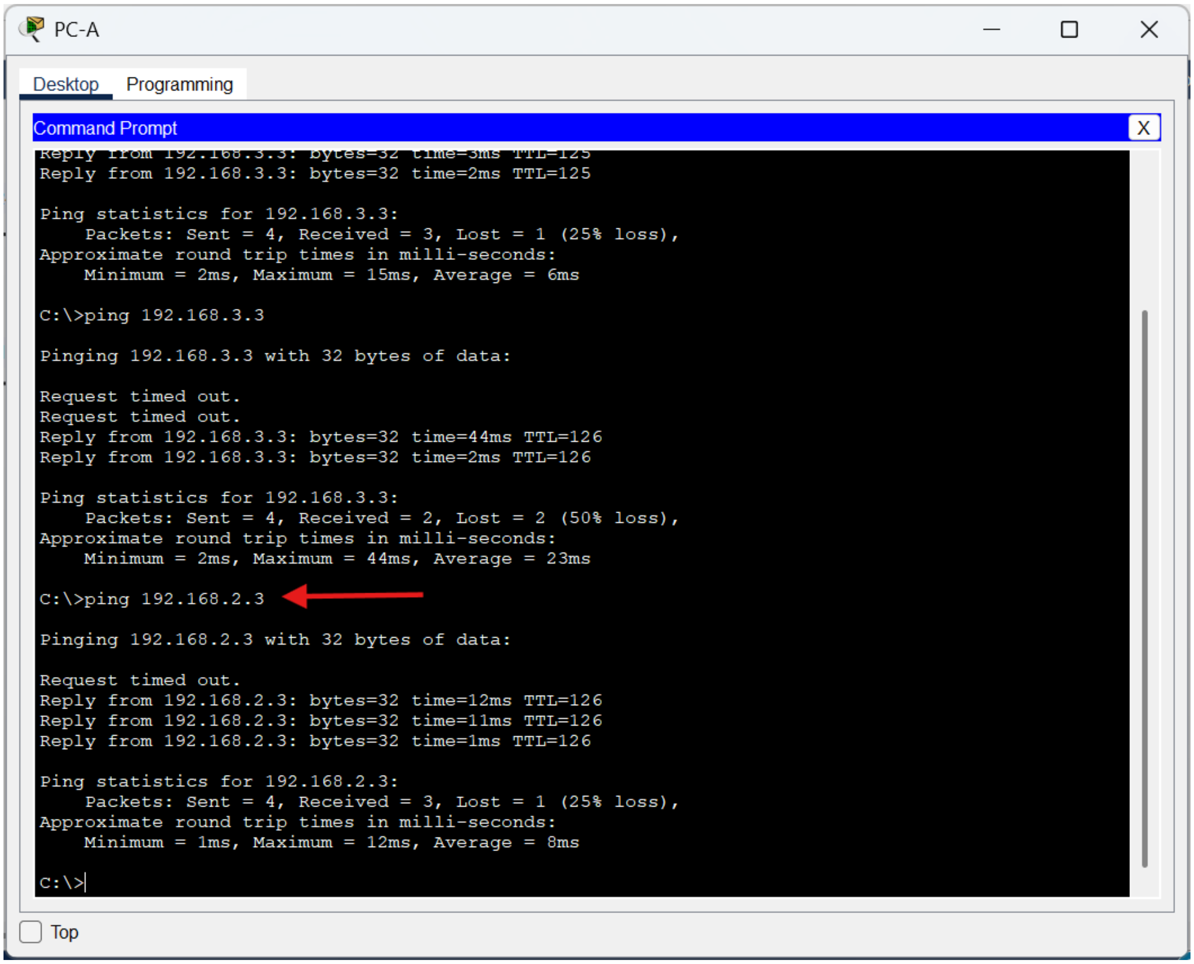
Step 3: Verify the tunnel after interesting traffic.

After generating interesting traffic with a ping from PC-A to PC-C, I reissued the show crypto ipsec sa command on R1. This time, the output showed that the number of **encapsulated**, **encrypted**, **decapsulated**, and **decrypted packets** was greater than zero. This confirmed that the **IPsec VPN tunnel was successfully established** and actively encrypting and decrypting traffic between the two LANs.



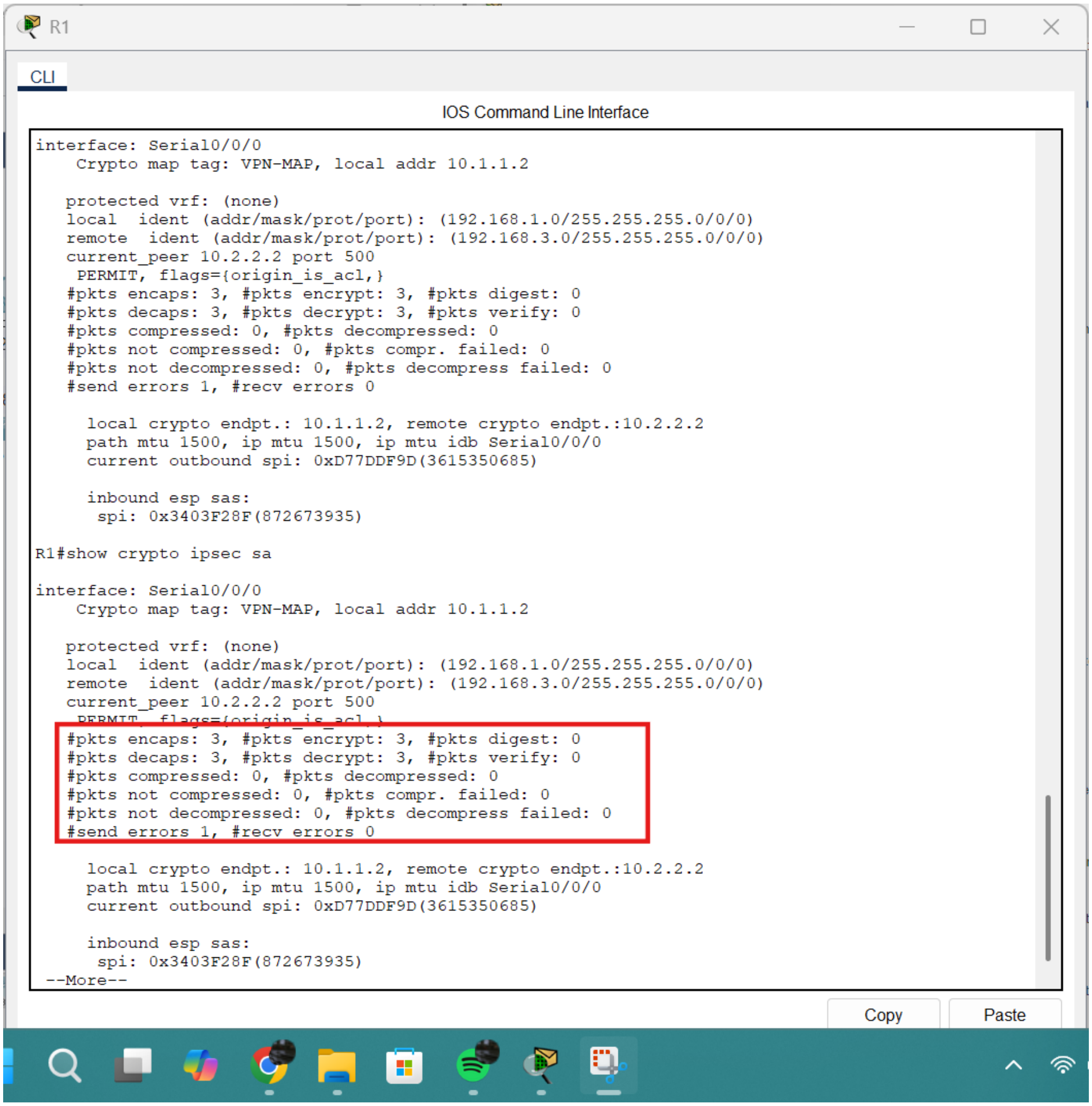
Step 4: Create uninteresting traffic.

To test that only defined traffic is encrypted, I initiated a **ping from PC-A to PC-B**. This traffic does **not match the ACL 110** criteria and is therefore considered **uninteresting**. As expected, the ping succeeded, but it **did not trigger the VPN tunnel**, since this communication was outside the scope of the encrypted IPsec policy.



Step 5: Verify the Tunnel

After generating uninteresting traffic, I ran the show crypto ipsec sa command again on R1. The packet counters for **encrypted** and **decrypted traffic** remained **unchanged**, confirming that the uninteresting traffic — such as the ping from PC-A to PC-B — was **not encrypted** and did **not pass through the IPsec tunnel**. This behavior validated that the ACLs correctly limited encryption to only specified traffic.



Summary.

In this exercise, I configured and verified a **site-to-site IPsec VPN** between **R1** and **R3** using CLI commands in **Cisco Packet Tracer**. The secure tunnel connected the LANs **192.168.1.0/24** (behind **PC-A**) and **192.168.3.0/24** (behind **PC-C**) through **R2**, which acted as a transit router.

The setup involved the following key devices:

- **R1**, **R2**, and **R3** routers (Cisco 1941)
- **PC-A**, **PC-B**, and **PC-C** as endpoint hosts

- **Switches S1, S2, and S3** for LAN connectivity

IKE Phase 1 was configured using **AES 256**, **pre-shared authentication**, and **DH group 5**. **IKE Phase 2** used **esp-aes** and **esp-sha-hmac** in the transform set VPN-SET. ACLs defined **interesting traffic**, and crypto maps were bound to the serial interfaces to enable secure tunneling.

Ping tests and the **show crypto ipsec sa** command confirmed that the tunnel was successfully triggered by interesting traffic and that encryption occurred as expected. The exercise was completed with full success and verified results.

Conclusion.

This exercise helped me understand how to configure and verify a **site-to-site IPsec VPN** using the Cisco IOS CLI. I gained hands-on experience with setting up **ISAKMP (IKE Phase 1)** and **IPsec (IKE Phase 2)** parameters, creating **crypto maps**, and applying them to interfaces. I also learned how to define **interesting traffic using ACLs** and verify tunnel activity using relevant commands like show crypto ipsec sa.

By simulating traffic between remote LANs and observing encryption behavior, I saw how **IPsec ensures secure communication over untrusted networks**. Overall, the practical approach in Packet Tracer improved my understanding of VPN configurations and their real-world applications.