Air University, Aerospace and Aviation Campus, Kamra

Department of Computer Science

BS. Cyber Security

NETWORK SECURITY LAB

SEMESTER LAB PROJECT REPORT

BSCYS – IV

Project Title	Secure VPN Tunnel using OpenVPN
Semester	BS(CYS) – IV (Spring 2025)
Number of group members	Zunaira (235051) Javeria Gul (235031) Laiba (235043) Fakhra (235057) Maimona (235056)
Date of Submission	June 12,2025
Submitted To	**MAHNOOR GILLANI** Lecturer (Cyber Security) Department of Cyber Security, AACK
Remarks by Teacher:	

Table of Contents

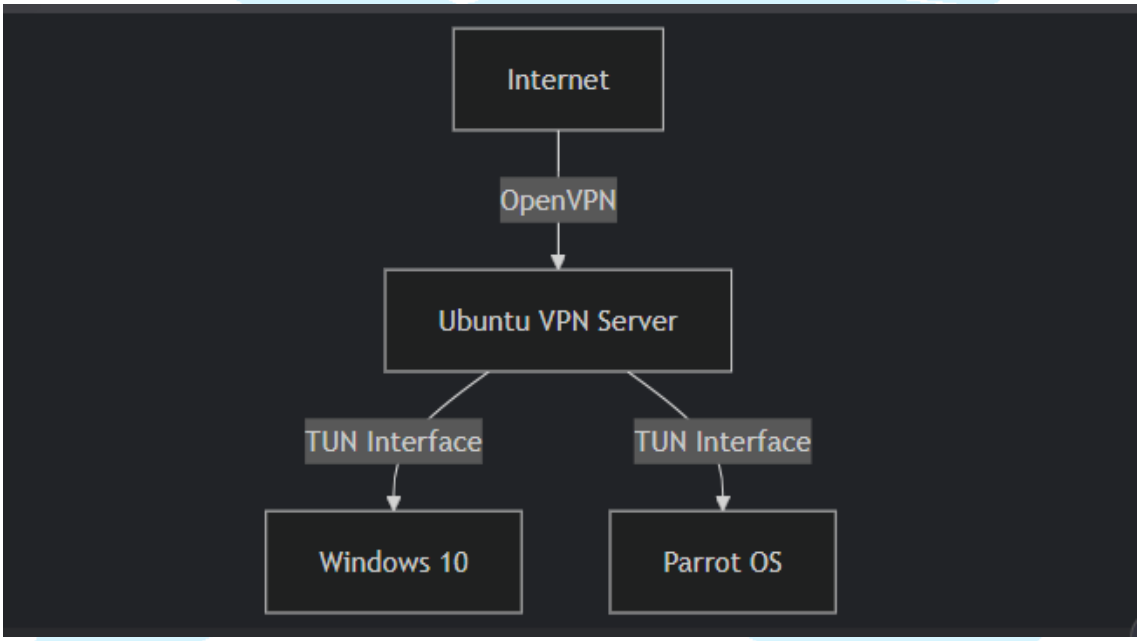
Project Overview

Objectives

This project demonstrates:

- Secure OpenVPN tunnel establishment between multiple OS platforms
- Encrypted file transfer verification
- Insider threat simulation and analysis
- Implementation of protective security measures

Network Architecture:



IP Assignments:

Virtual Machine	VPN Server IP	Lan Interface IP
Ubuntu Server	10.8.0.1	192.168.28.129
Windows Client	10.8.0.3	192.168.28.132
Parrot OS Client	10.8.0.2	192.168.28.133

Scope

This project focuses on implementing and evaluating a secure VPN infrastructure with the following boundaries:

In Scope

1. **VPN Implementation**
    - OpenVPN server setup on Ubuntu 20.04
    - Client configurations for Windows 10 and Parrot OS
    - Certificate-based authentication
2. **Security Testing**
    - Encrypted file transfer validation (SCP/SMB)
    - Simulated insider attacks from compromised client
    - Traffic encryption analysis using Wireshark
3. **Defensive Measures**
    - Firewall configuration (UFW/IPtables)
    - Client-to-client access control
    - Service hardening (SMB restrictions)

Out of Scope

1. Enterprise-scale VPN deployment
2. Physical network infrastructure
3. Advanced Persistent Threat (APT) simulations
4. Mobile device connectivity

System Requirements

Hardware

Component	Specification
Server	Ubuntu 20.04 (2vCPU, 4GB RAM)
Client 1	Windows 10 (x64)
Client 2	Parrot OS (Security Edition)

Software

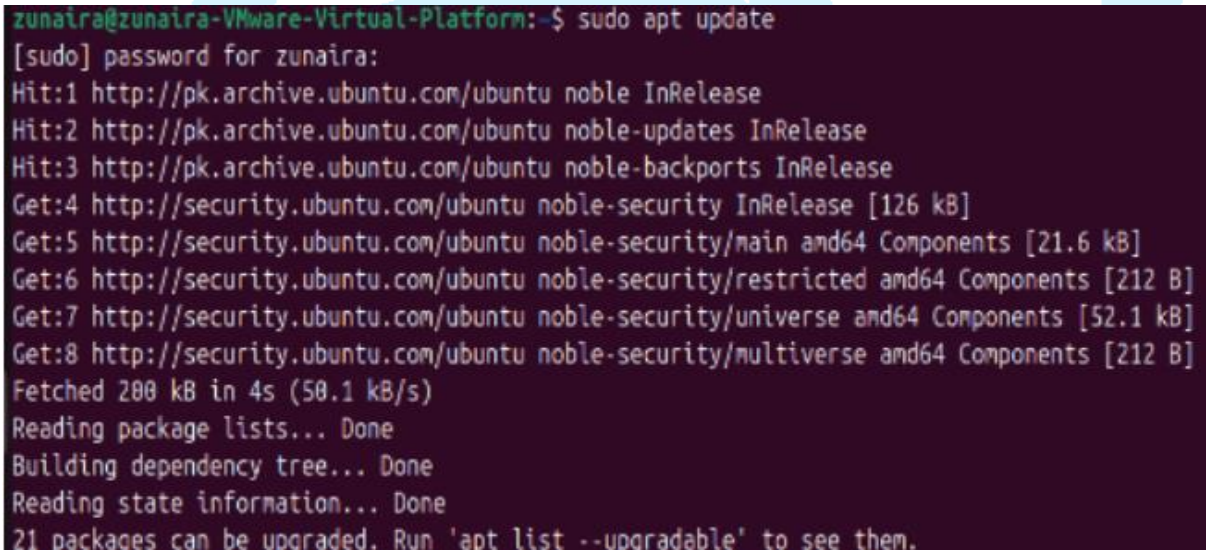
- OpenVPN 2.4.7
- Easy-RSA 3.0.6
- Wireshark 3.4.0
- Nmap 7.80

Implementation:

PART 1: Set Up VPN Tunnel

On Ubuntu (VPN Server)

- Update & install OpenVPN



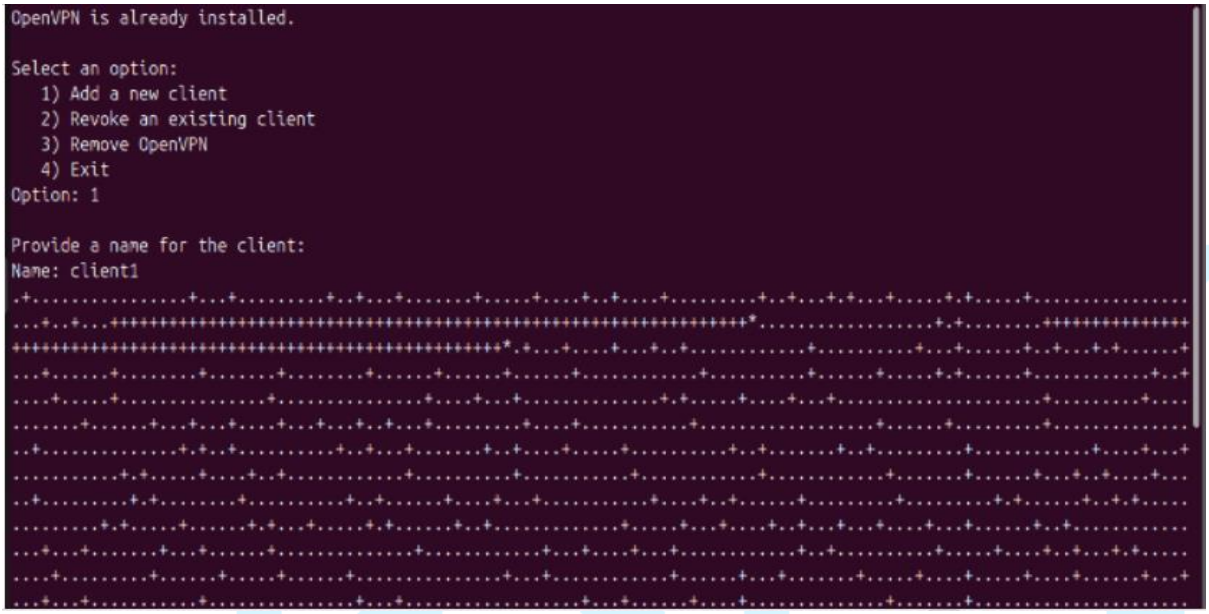
```
zunaira@zunaira-VMware-Virtual-Platform:~$ sudo apt install wget curl openvpn
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.4-1ubuntu4.1).
curl is already the newest version (8.5.0-2ubuntu10.6).
openvpn is already the newest version (2.6.12-0ubuntu0.24.04.3).
0 upgraded, 0 newly installed, 0 to remove and 21 not upgraded.
```

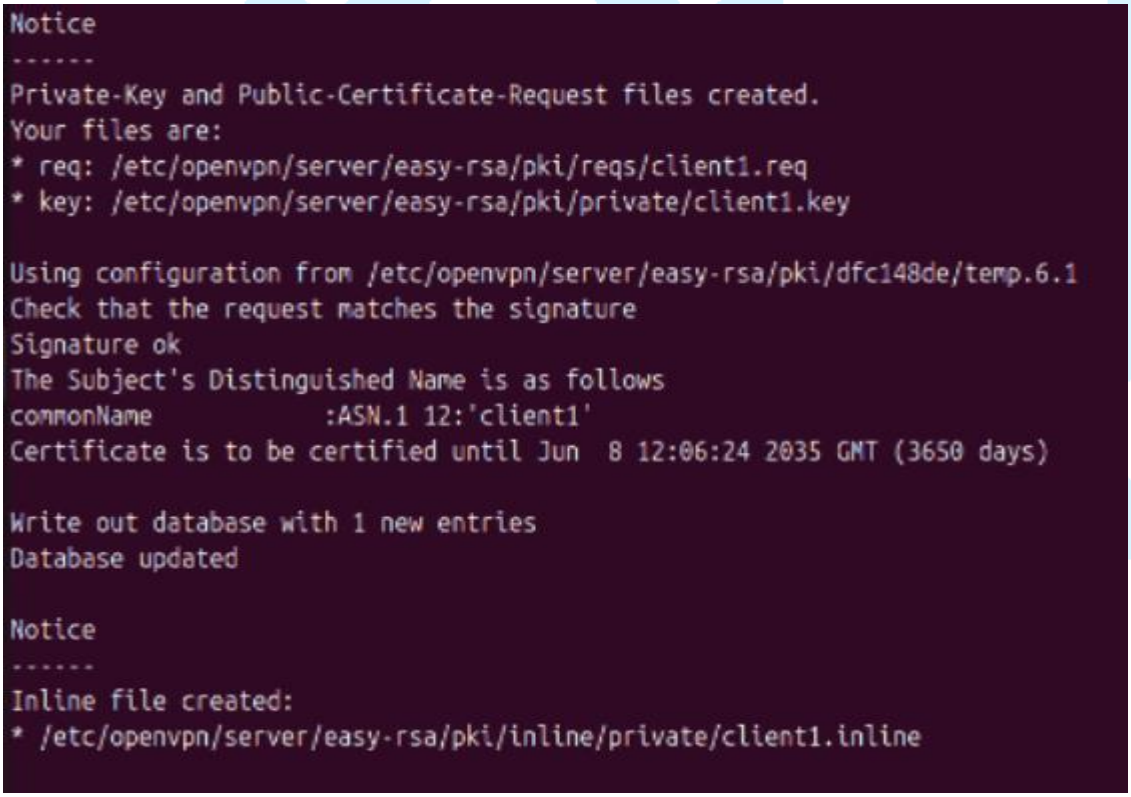
Download & run install script:

```
zunaira@zunaira-VMware-Virtual-Platform:~$ wget https://git.io/vpn -O openvpn-install.sh
--2025-06-10 17:04:41--  https://git.io/vpn
Resolving git.io (git.io)... 140.82.113.21
Connecting to git.io (git.io)|140.82.113.21|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://raw.github.com/Nyr/openvpn-install/master/openvpn-install.sh [following]
--2025-06-10 17:04:43--  https://raw.github.com/Nyr/openvpn-install/master/openvpn-install.sh
Resolving raw.github.com (raw.github.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.github.com (raw.github.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://raw.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh [following]
--2025-06-10 17:04:44--  https://raw.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ..
.
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24807 (24K) [text/plain]
Saving to: 'openvpn-install.sh'

openvpn-install.sh           100%[=============================================>]  24.23K  --.-KB/s    in 0.03s

2025-06-10 17:04:45 (807 KB/s) - 'openvpn-install.sh' saved [24807/24807]
```

Choose options during setup:

- Name: client1 (for Windows)





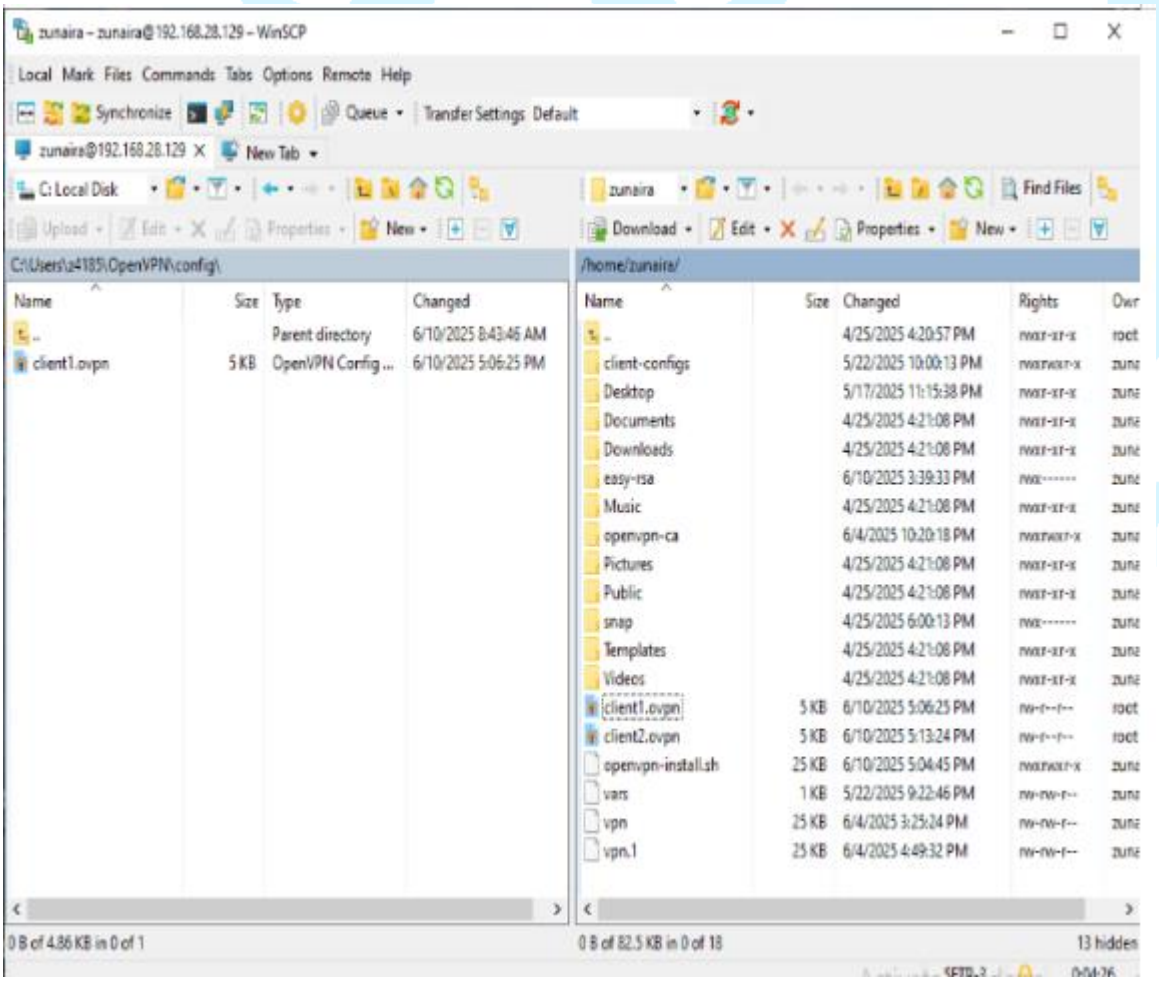
Check for .ovpn file:

```
zunaira@zunaira-VMware-Virtual-Platform:~$ ls ~/*.ovpn
/home/zunaira/client1.ovpn
```

On Windows 10 (Client)

- Install OpenVPN GUI

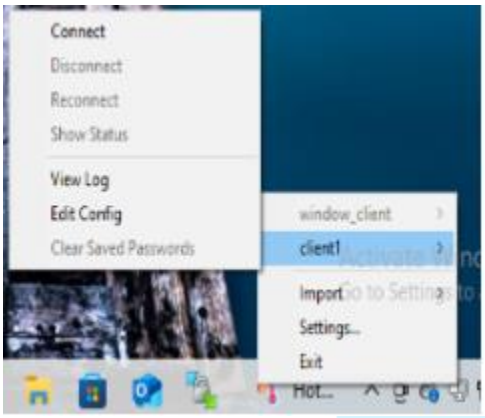
- Transfer .ovpn file (client1.ovpn) using WinSCP



Place file in:

C:\Users\z4185\OpenVPN\config

- Right-click OpenVPN GUI → Run as Admin → Connect



On Parrot OS (Client/Attacker)

- Transfer client2.ovpn via SCP:

```
┌─[✗]─[user@parrot]─[~/Desktop]
└──╼ $scp zunaira@192.168.28.129:/home/zunaira/client2.ovpn ~/
zunaira@192.168.28.129's password:
client2.ovpn                                  100% 4977   888.7KB/s   00:00
```

- Connect to VPN:

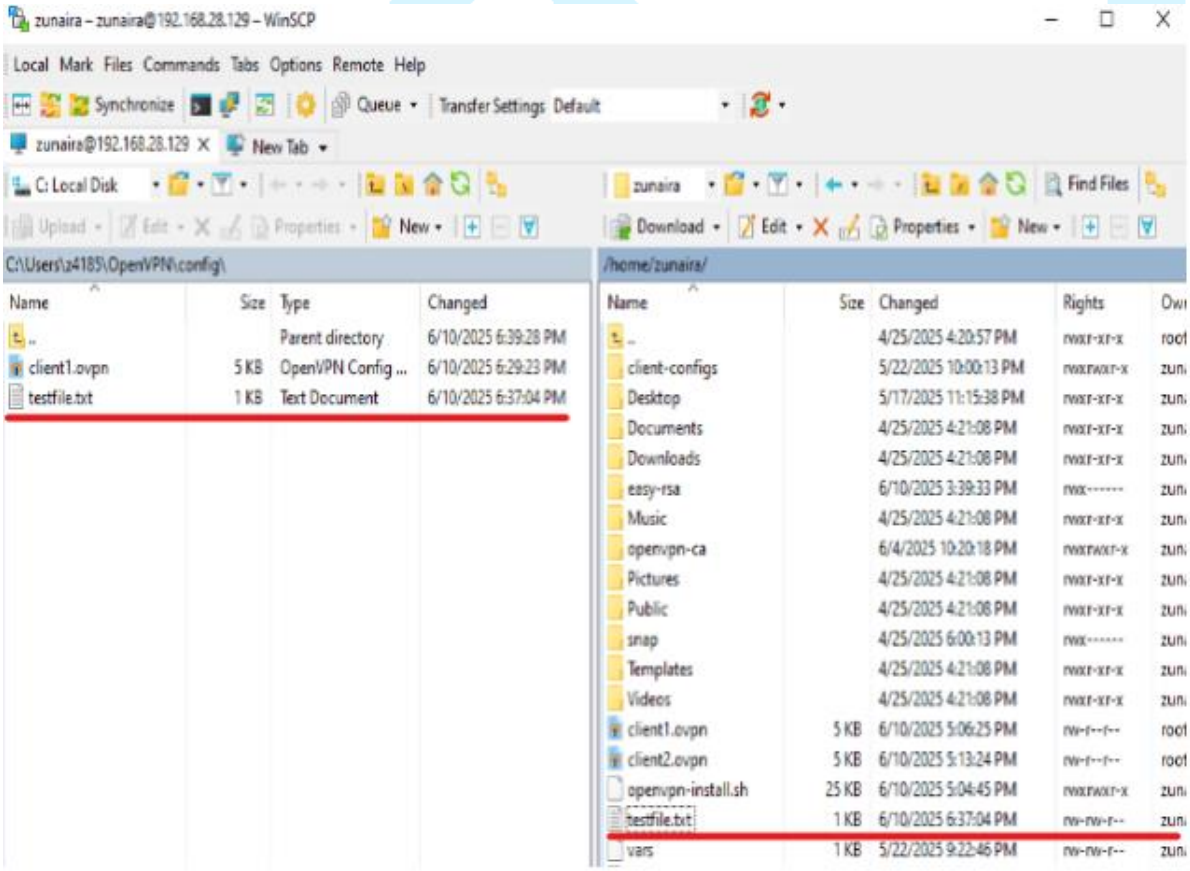
```
┌──(user@parrot)-[~]
└─$ sudo openvpn --config client2.ovpn
2025-06-10 17:29:30 --cipher is not set. Previous OpenVPN version defaulted to BF-CBC as fallback when cipher negotiation failed in this case. If you need this fallback please add '--data-ciph
ers-fallback BF-CBC' to your configuration and/or add BF-CBC to --data-ciphers.
2025-06-10 17:29:30 OpenVPN 2.5.1 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Feb 24 2021
2025-06-10 17:29:30 library versions: OpenSSL 1.1.1k  25 Mar 2021, LZO 2.10
2025-06-10 17:29:30 Outgoing Control Channel Encryption: Cipher 'AES-256-CTR' initialized with 256 bit key
2025-06-10 17:29:30 Outgoing Control Channel Encryption: Using 256 bit message hash 'SHA256' for HMAC authentication
2025-06-10 17:29:30 Incoming Control Channel Encryption: Cipher 'AES-256-CTR' initialized with 256 bit key
2025-06-10 17:29:30 Incoming Control Channel Encryption: Using 256 bit message hash 'SHA256' for HMAC authentication
2025-06-10 17:29:30 TCP/UDP: Preserving recently used remote address: [AF_INET]192.168.28.129:1194
2025-06-10 17:29:30 Socket Buffers: R=[212992->212992] S=[212992->212992]
2025-06-10 17:29:30 UDP link local: (not bound)
2025-06-10 17:29:30 UDP link remote: [AF_INET]192.168.28.129:1194
2025-06-10 17:29:30 TLS: Initial packet from [AF_INET]192.168.28.129:1194, sid=739a52d6 882daa2e
2025-06-10 17:29:30 VERIFY OK: depth=1, CN=Easy-RSA CA
2025-06-10 17:29:30 VERIFY KU OK
2025-06-10 17:29:30 Validating certificate extended key usage
2025-06-10 17:29:30 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2025-06-10 17:29:30 VERIFY EKU OK
2025-06-10 17:29:30 VERIFY OK: depth=0, CN=server
2025-06-10 17:29:30 WARNING: 'link-mtu' is used inconsistently, local='link-mtu 1585', remote='link-mtu 1601'
2025-06-10 17:29:30 WARNING: 'keysize' is used inconsistently, local='keysize 128', remote='keysize 256'
2025-06-10 17:29:30 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, 2048 bit RSA
2025-06-10 17:29:30 [server] Peer Connection Initiated with [AF_INET]192.168.28.129:1194
2025-06-10 17:29:30 PUSH: Received control message: 'PUSH_REPLY,redirect-gateway def1 bypass-dhcp,dhcp-option DNS 8.8.8.8,dhcp-option DNS 8.8.4.4,block-outside-dns,route-gateway 10.8.0.1,topol
ogy subnet,ping 10,ping-restart 120,ifconfig 10.8.0.2 255.255.255.0,peer-id 0,cipher AES-256-GCM'
2025-06-10 17:29:30 Unrecognized option or missing or extra parameter(s) in [PUSH-OPTIONS]:4: block-outside-dns (2.5.1)
2025-06-10 17:29:30 OPTIONS IMPORT: timers and/or timeouts modified
2025-06-10 17:29:30 OPTIONS IMPORT: --ifconfig/up options modified
2025-06-10 17:29:30 OPTIONS IMPORT: route options modified
2025-06-10 17:29:30 OPTIONS IMPORT: route-related options modified
2025-06-10 17:29:30 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
2025-06-10 17:29:30 OPTIONS IMPORT: peer-id set
2025-06-10 17:29:30 OPTIONS IMPORT: adjusting link_mtu to 1624
2025-06-10 17:29:30 OPTIONS IMPORT: data channel crypto options modified
```

```
2025-06-10 17:29:30 OPTIONS IMPORT: peer-id set
2025-06-10 17:29:30 OPTIONS IMPORT: adjusting link_mtu to 1624
2025-06-10 17:29:30 OPTIONS IMPORT: data channel crypto options modified
2025-06-10 17:29:30 Data Channel: using negotiated cipher 'AES-256-GCM'
2025-06-10 17:29:30 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2025-06-10 17:29:30 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2025-06-10 17:29:30 net_route_v4_best_gw query: dst 0.0.0.0
2025-06-10 17:29:30 net_route_v4_best_gw result: via 192.168.28.2 dev eth0
2025-06-10 17:29:30 ROUTE_GATEWAY 192.168.28.2/255.255.255.0 IFACE=eth0 HWADDR=00:0c:29:77:31:f6
2025-06-10 17:29:30 TUN/TAP device tun0 opened
2025-06-10 17:29:30 net_iface_mtu_set: mtu 1500 for tun0
2025-06-10 17:29:30 net_iface_up: set tun0 up
2025-06-10 17:29:30 net_addr_v4_add: 10.8.0.2/24 dev tun0
2025-06-10 17:29:30 net_route_v4_add: 192.168.28.129/32 via 192.168.28.2 dev eth0 table 0 metric -1
2025-06-10 17:29:30 net_route_v4_add: 0.0.0.0/1 via 10.8.0.1 dev [NULL] table 0 metric -1
2025-06-10 17:29:30 net_route_v4_add: 128.0.0.0/1 via 10.8.0.1 dev [NULL] table 0 metric -1
2025-06-10 17:29:30 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2025-06-10 17:29:30 Initialization Sequence Completed
```

PART 2: Transfer Files Between Windows ↔ Ubuntu

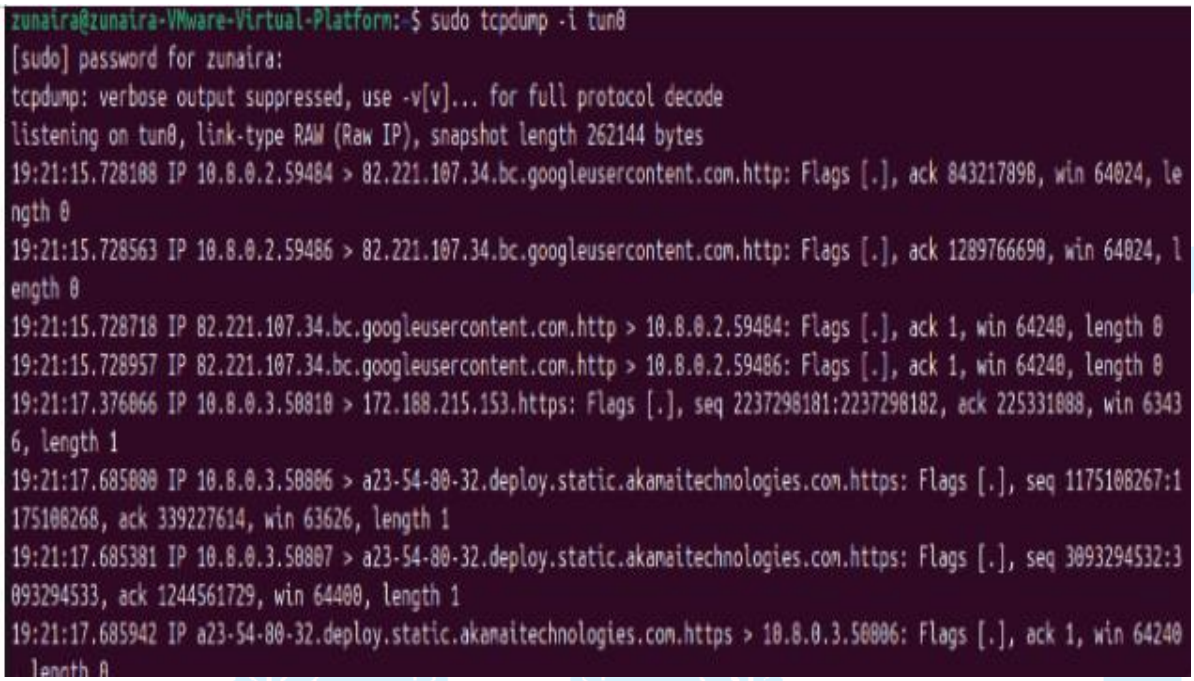
This already works using SCP or FileZilla. Confirm it's through the VPN tunnel by checking IP routes (ipconfig / ifconfig).

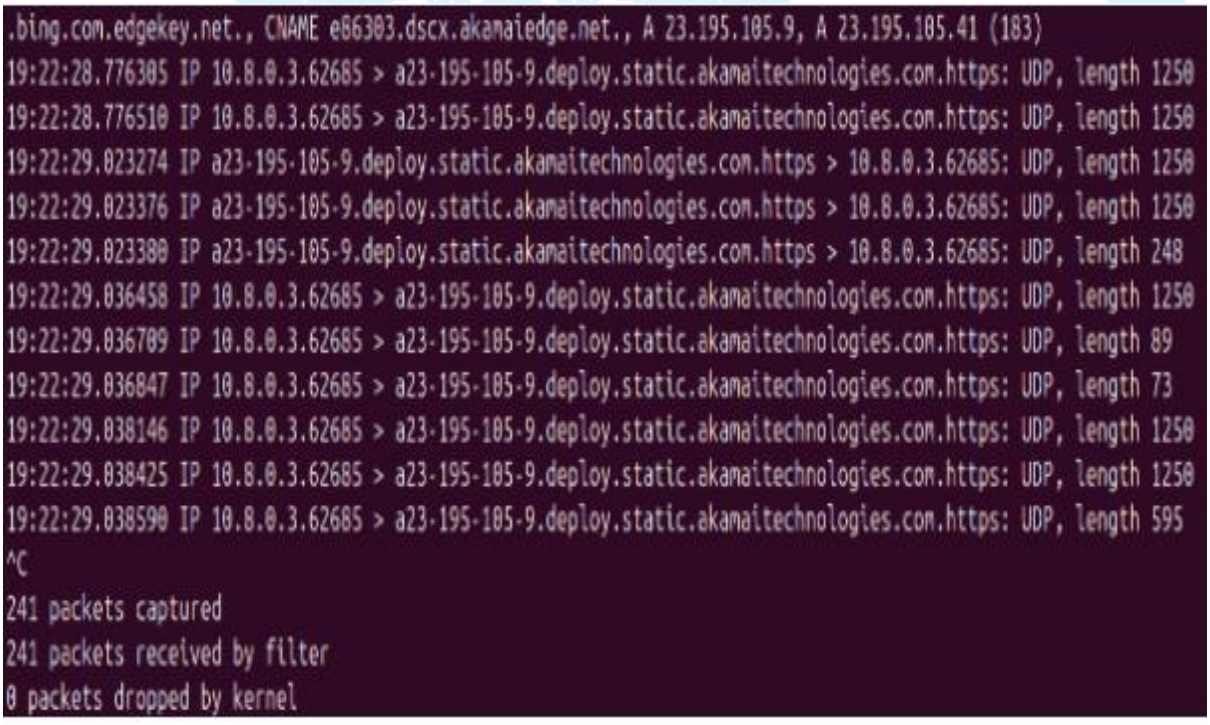
- Use WinSCP:



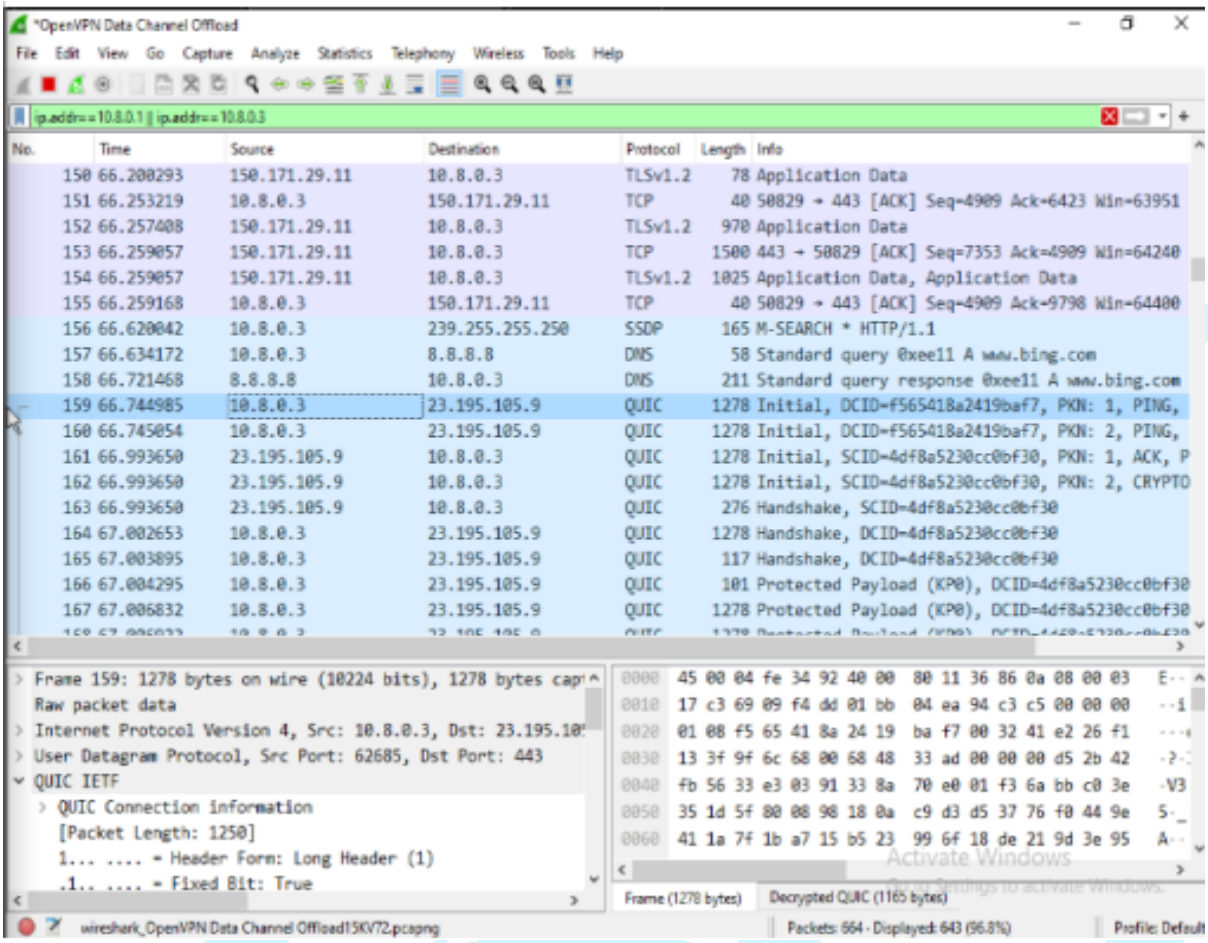
- Verify VPN Route

From Ubuntu:





From Windows, open Wireshark and filter:

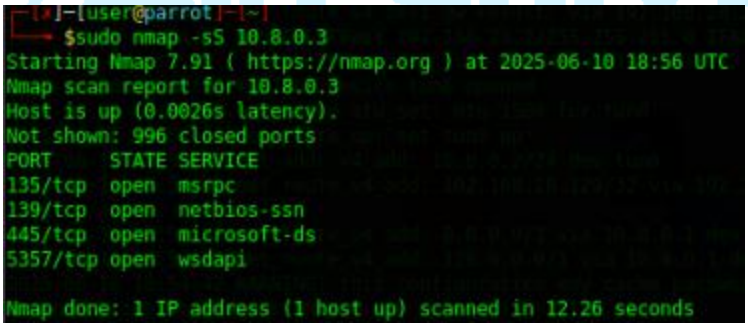


PART 3: Perform Insider Attack from Parrot

Simulate Insider Actions:

- Try scanning Windows from Parrot using nmap

Step 1:Scan Windows Using Nmap



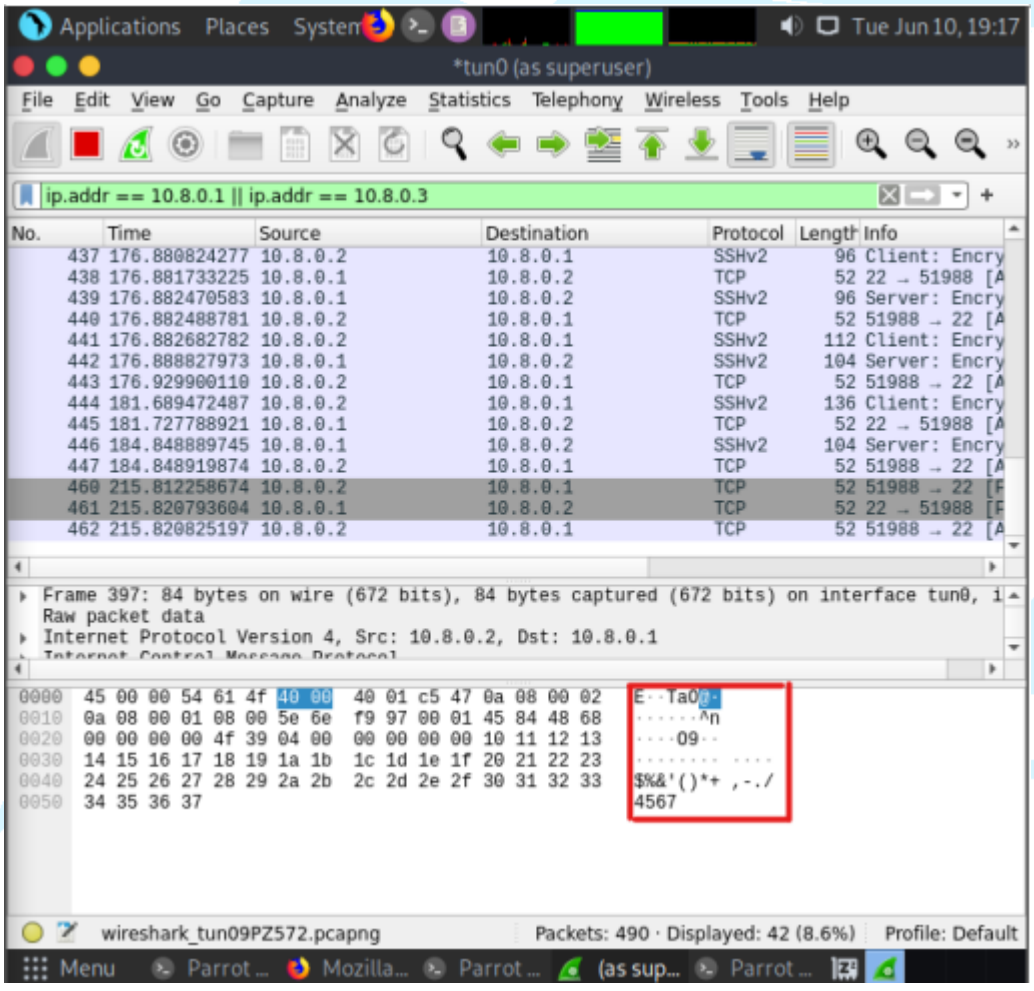
- **Attempt unauthorized file access using:**

Step 2: Try Accessing Shared Files via SMB

Unauthorized SMB Access Attempt from Parrot (VPN Insider) — Access Denied

- **Try sniffing VPN traffic using Wireshark (sudo wireshark)**

Step 3: Passive Sniffing with Wireshark

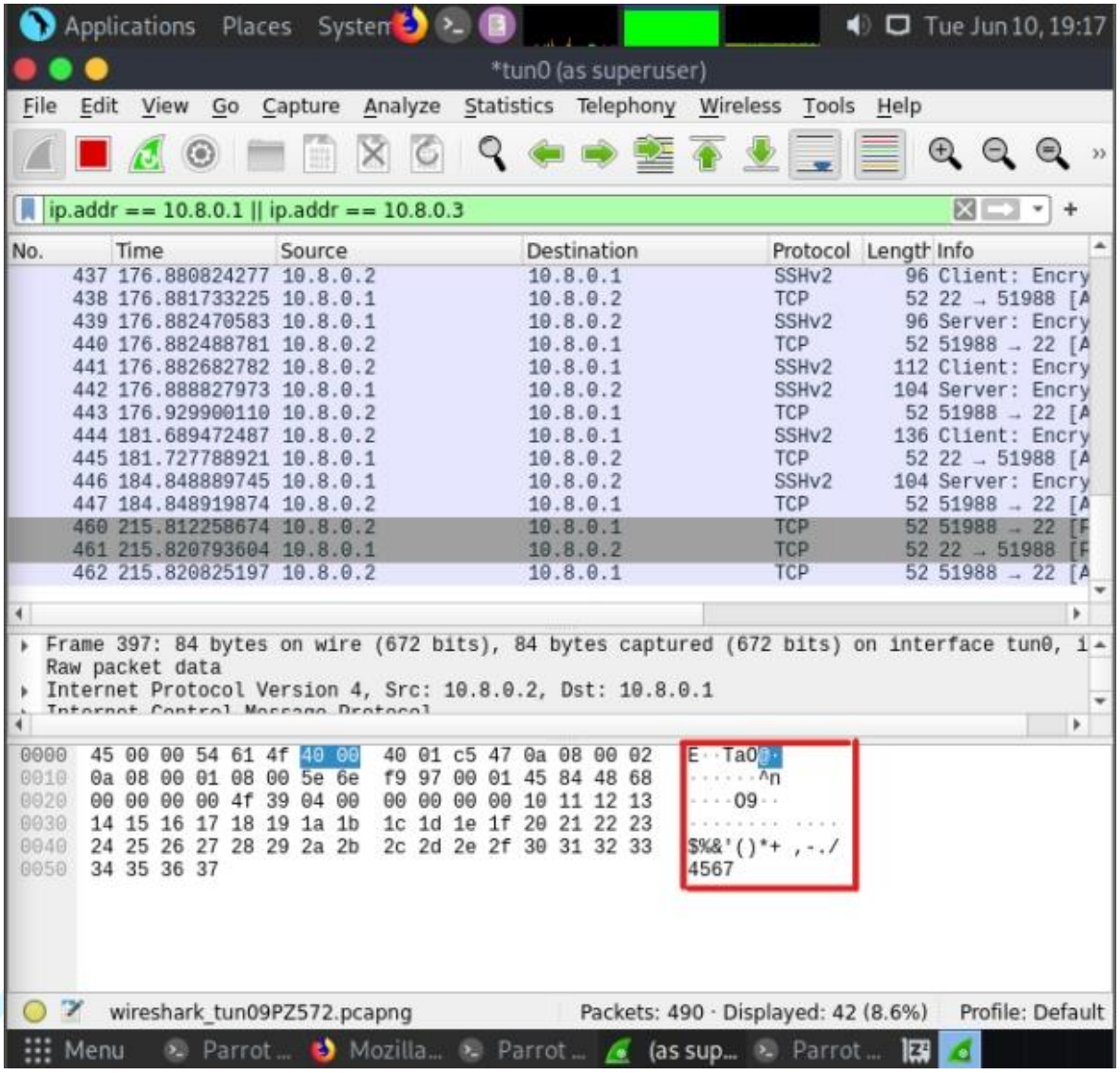


Due to OpenVPN's client-to-server encryption model, peer VPN clients (e.g., Parrot OS) cannot passively sniff traffic between other clients (e.g., Windows ↔ Ubuntu) unless positioned as a gateway or VPN server.

PART 4: Analyze Traffic in Wireshark

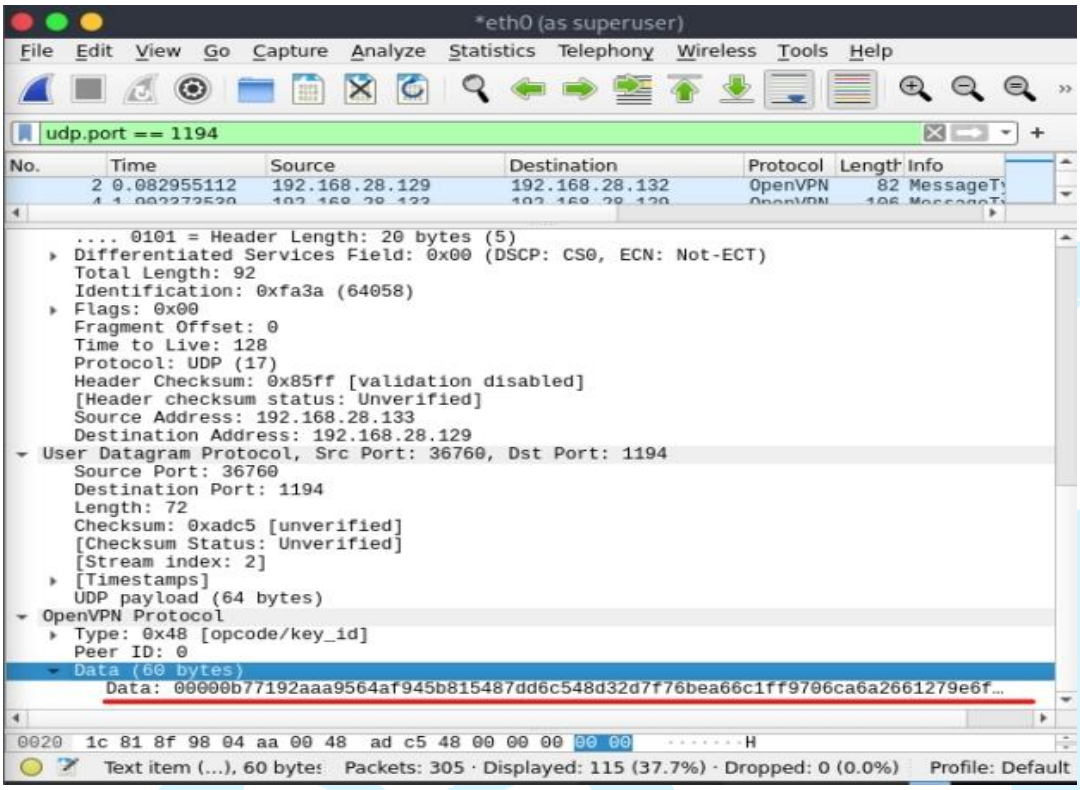
Check for Encryption:

- Start capture on tun0 (VPN interface).



- Look for OpenVPN packets (UDP 1194).

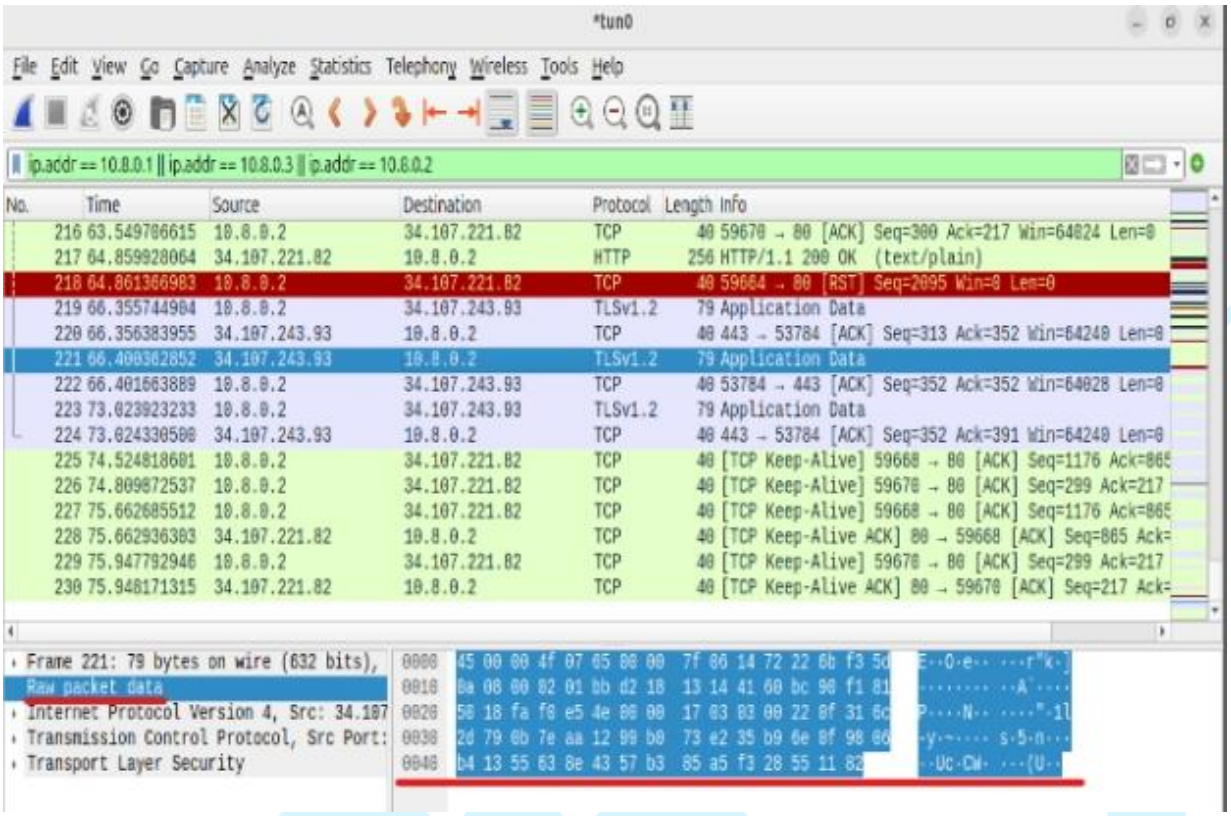
Observe that payload is encrypted – compare with normal LAN traffic.



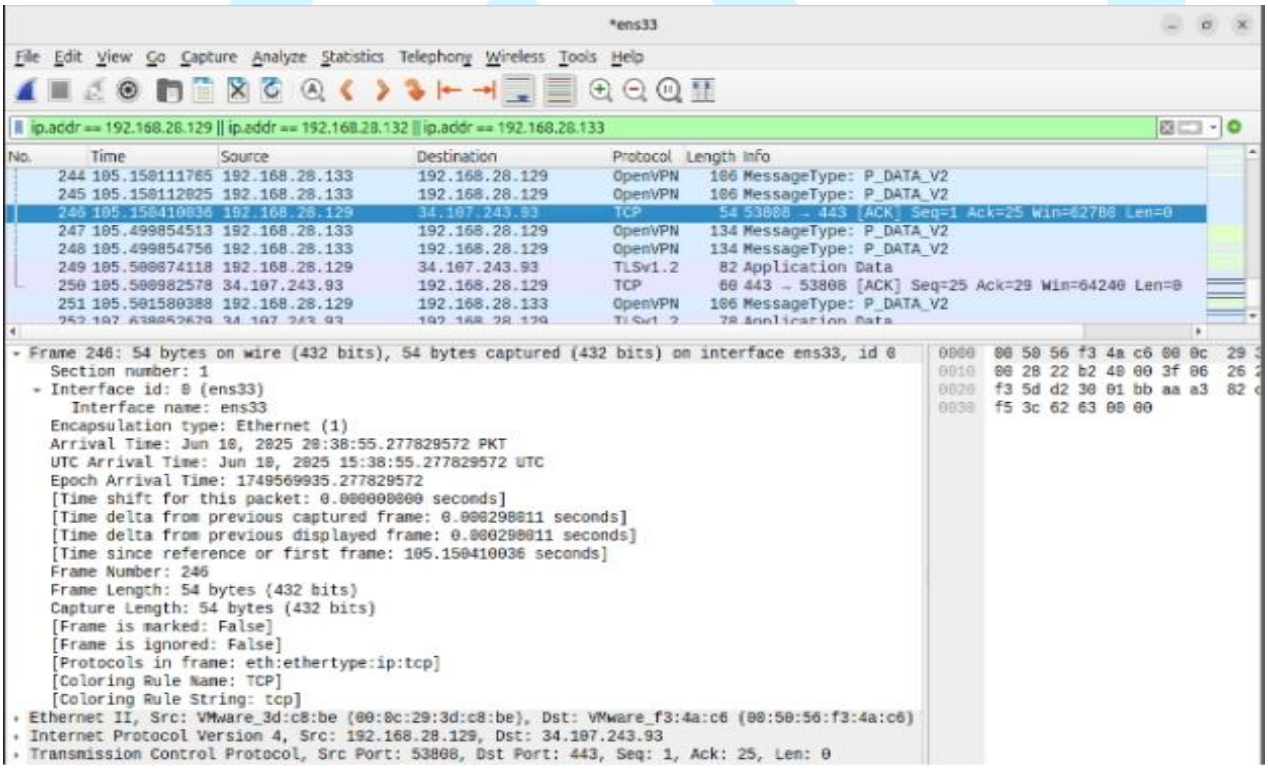
Observe Payload

- Under Data, there is a hex dump (right side: random-looking hex bytes)
- The ASCII panel (right-most column) will show unreadable garbage
- No readable text, file names, usernames, or commands

1. **Capture Decrypted VPN Traffic (Inside the Tunnel)**



2. **Capture Decrypted VPN Traffic (Inside the Tunnel)**



PART 5: Create Security Rules to Prevent Insider Attack

On Ubuntu (VPN Server):

- UFW firewall rules:

```
zunaira@zunaira-VMware-Virtual-Platform:~$ sudo ufw allow 1194/udp
Skipping adding existing rule
Skipping adding existing rule (v6)
```

```
zunaira@zunaira-VMware-Virtual-Platform:~$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
```

```
zunaira@zunaira-VMware-Virtual-Platform:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Block access between VPN clients (client isolation):

- Edit /etc/openvpn/server.conf:

```
zunaira@zunaira-VMware-Virtual-Platform: ~
 GNU nano 7.2                        /etc/openvpn/server.conf *
local 0.0.0.0
proto udp
port 1194
dev tun
server 10.7.8.0 255.255.255.0
topology subnet
ca ca.crt
cert server.crt
key server.key
dh dh.pem
tls-crypt ta.key  # CHANGED from tls-auth
data-ciphers AES-256-GCM:AES-128-GCM  # CHANGED from cipher
data-ciphers-fallback AES-256-CBC
auth SHA256
user nobody
group nogroup
persist-key
persist-tun
keepalive 10 120
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
ifconfig-pool-persist /var/log/openvpn/ipp.txt  # FIXED path
status /var/log/openvpn/openvpn-status.log
verb 3
explicit-exit-notify 1
#client-to-client
```

1. **Add specific rules using iptables:**

   Block Parrot OS VPN IP from Accessing Windows

   - Use iptables:

```
zunaira@zunaira-VMware-Virtual-Platform:~$ sudo iptables -A FORWARD -s 10.8.0.2 -d 10.8.0.3 -j DROP
```

2. **Restrict file sharing services**

```
zunaira@zunaira-VMware-Virtual-Platform:~$ sudo ufw deny from 10.8.0.2 to any port 445
Rule added
zunaira@zunaira-VMware-Virtual-Platform:~$ sudo ufw deny from 10.8.0.2 to any port 139
Rule added
```

```
zunaira@zunaira-VMware-Virtual-Platform:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 1194/udp                   ALLOW IN    Anywhere
[ 2] 22/tcp                     ALLOW IN    Anywhere
[ 3] OpenSSH                    ALLOW IN    Anywhere
[ 4] 445                        DENY IN     10.8.0.2
[ 5] 139                        DENY IN     10.8.0.2
[ 6] 1194/udp (v6)              ALLOW IN    Anywhere (v6)
[ 7] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 8] OpenSSH (v6)               ALLOW IN    Anywhere (v6)
```

Findings & Analysis

Vulnerability	Risk Level	Mitigation	Result
Unencrypted SMB	High	Disabled SMBv1	Access denied
Client-to-client access	Medium	Enabled isolation	Scan blocked
Port scanning	Low	IP filtering	Limited visibility

Conclusion

Key Achievements

- Successfully established encrypted VPN tunnel
- Verified payload encryption via Wireshark
- Implemented effective client isolation
- Blocked unauthorized access attempts

Recommendations

1. Implement two-factor authentication
2. Enable VPN connection logging
3. Regular certificate rotation (90-day policy)