



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	Fakhra (235057)
Date of Submission	June 12,2025
Submitted To	MAHNOOR GILLANI Lecturer (Cyber Security) Department of Cyber Security, AACK
Remarks by Teacher:	

AIR UNIVERSITY

Table of Contents

Project Overview	3
Objectives.....	3
Network Architecture:	3
IP Assignments:	Error! Bookmark not defined.
Scope	3
In Scope	3
Out of Scope.....	4
System Requirements	4
Hardware.....	4
Software	4
Implementation:	5
PART 1: Set Up VPN Tunnel	5
PART 2: Transfer Files Between Windows ↔ Ubuntu	9
PART 3: Perform Insider Attack from Parrot.....	12
PART 4: Analyze Traffic in Wireshark.....	14
PART 5: Create Security Rules to Prevent Insider Attack.....	17
Findings & Analysis	18
Conclusion.....	18
Key Achievements.....	18
Recommendations	18

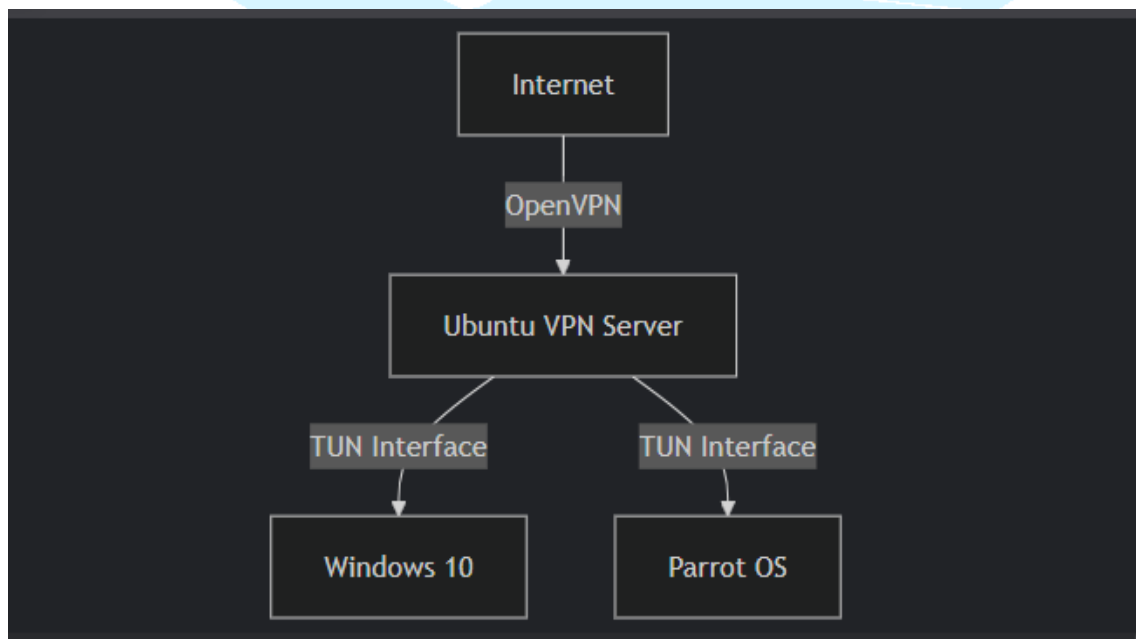
Project Overview

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:



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

```
Hit:1 http://pk.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://pk.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://pk.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:7 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.1 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Fetched 200 kB in 4s (50.1 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
21 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
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.
```

AIR UNIVERSITY

Download & run install script:

```
--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.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh [following]
--2025-06-10 17:04:43-- https://raw.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.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)

[illegible]

```
Notice
-----
Private-Key and Public-Certificate-Request files created.
Your files are:
* req: /etc/openvpn/server/easy-rsa/pki/reqs/client1.req
* key: /etc/openvpn/server/easy-rsa/pki/private/client1.key

Using configuration from /etc/openvpn/server/easy-rsa/pki/dfc148de/temp.6.1
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName      :ASN.1 12:'client1'
Certificate is to be certified until Jun  8 12:06:24 2035 GMT (3650 days)

Write out database with 1 new entries
Database updated

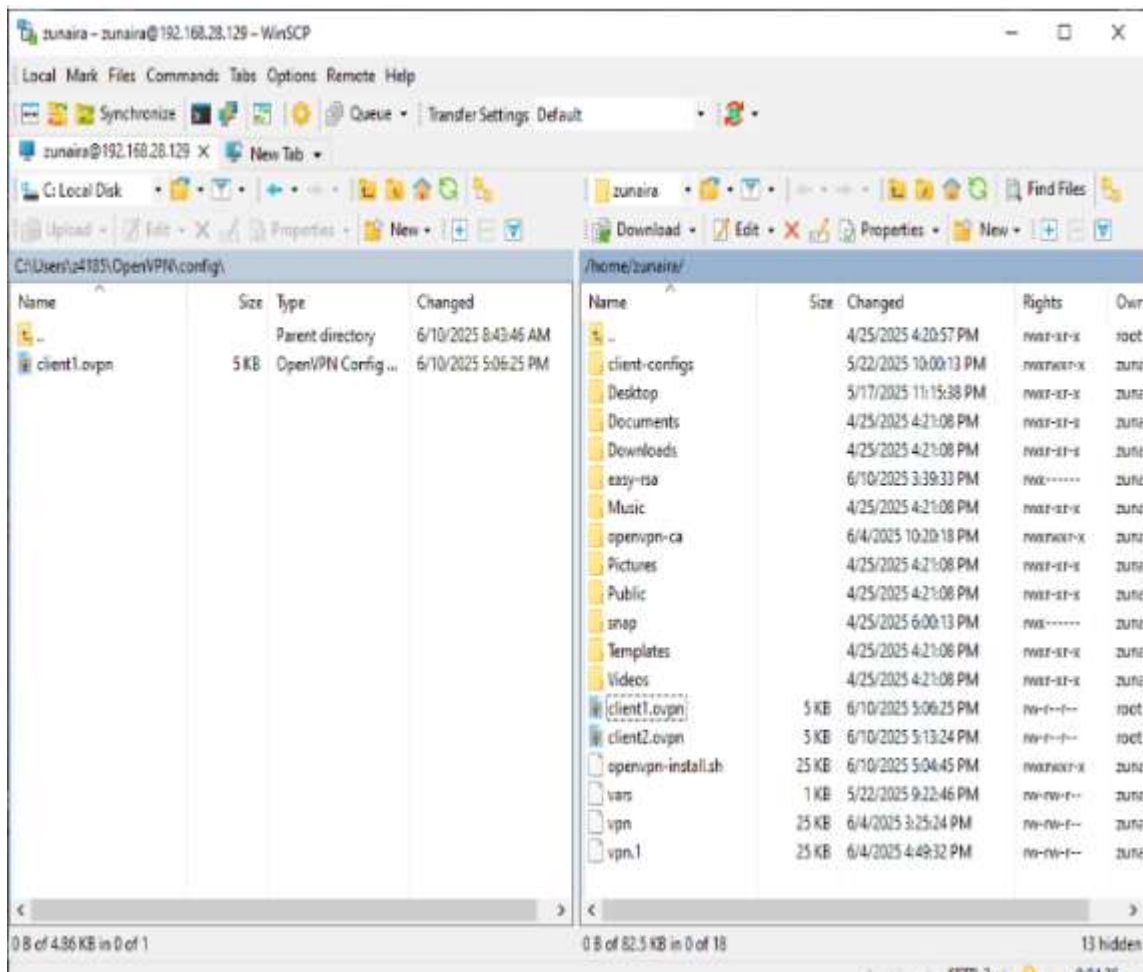
Notice
-----
Inline file created:
* /etc/openvpn/server/easy-rsa/pki/inline/private/client1.inline
```

On Windows 10 (Client)

- Install OpenVPN GUI



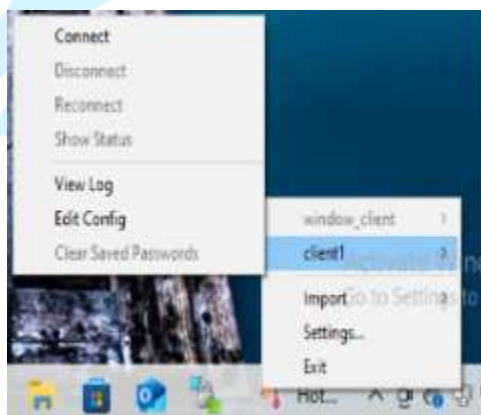
- Transfer .ovpn file (client1.ovpn) using WinSCP



Place file in:

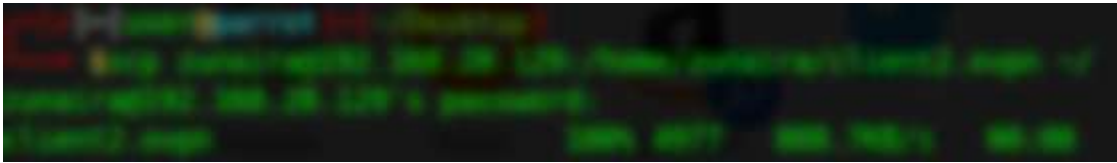
C:\Users\z4185\OpenVPN\config

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



On Parrot OS (Client/Attacker)

- Transfer client2.ovpn via SCP:



- 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-ciphers-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] [HMAC/PKTRNG] [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=729a52d6 80bba2e
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 EXU 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 1500', 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,topology 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 (2.3.1)
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-van32 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

```

```

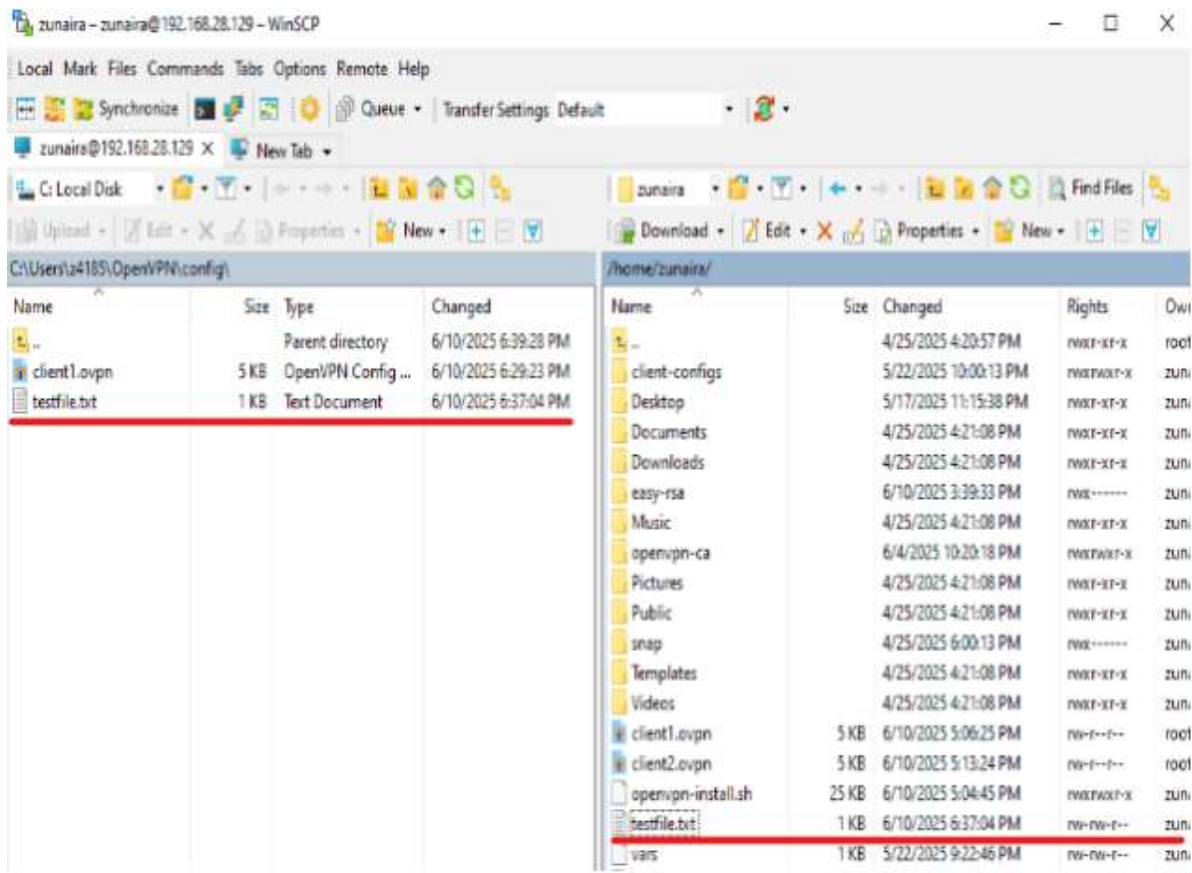
2025-06-10 17:29:30 OPTIONS IMPORT: peer-id set
2025-06-10 17:29:30 OPTIONS IMPORT: adjusting link mtu to 1624
2025-06-10 17:29:30 OPTIONS IMPORT: data channel crypto options modified
2025-06-10 17:29:30 Data Channel: using negotiated cipher 'AES-256-GCM'
2025-06-10 17:29:30 Outgoing Data Channel: Cipher '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:

AIR UNIVERSITY

```

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
19:21:15.728108 IP 10.8.0.2.59484 > 82.221.107.34.bc.googleusercontent.com.http: Flags [.], ack 843217898, win 64024, length 0
19:21:15.728563 IP 10.8.0.2.59486 > 82.221.107.34.bc.googleusercontent.com.http: Flags [.], ack 1289766698, win 64024, length 0
19:21:15.728718 IP 82.221.107.34.bc.googleusercontent.com.http > 10.8.0.2.59484: Flags [.], ack 1, win 64240, length 0
19:21:15.728957 IP 82.221.107.34.bc.googleusercontent.com.http > 10.8.0.2.59486: Flags [.], ack 1, win 64240, length 0
19:21:17.376066 IP 10.8.0.3.50810 > 172.188.215.153.https: Flags [.], seq 2237298181:2237298182, ack 225331088, win 63436, length 1
19:21:17.685080 IP 10.8.0.3.50806 > a23-54-80-32.deploy.static.akamaitechnologies.com.https: Flags [.], seq 1175108267:1175108268, ack 339227614, win 63626, length 1
19:21:17.685381 IP 10.8.0.3.50807 > a23-54-80-32.deploy.static.akamaitechnologies.com.https: Flags [.], seq 3093294532:3093294533, ack 1244561729, win 64400, length 1
19:21:17.685942 IP a23-54-80-32.deploy.static.akamaitechnologies.com.https > 10.8.0.3.50806: Flags [.], ack 1, win 64240, length 0

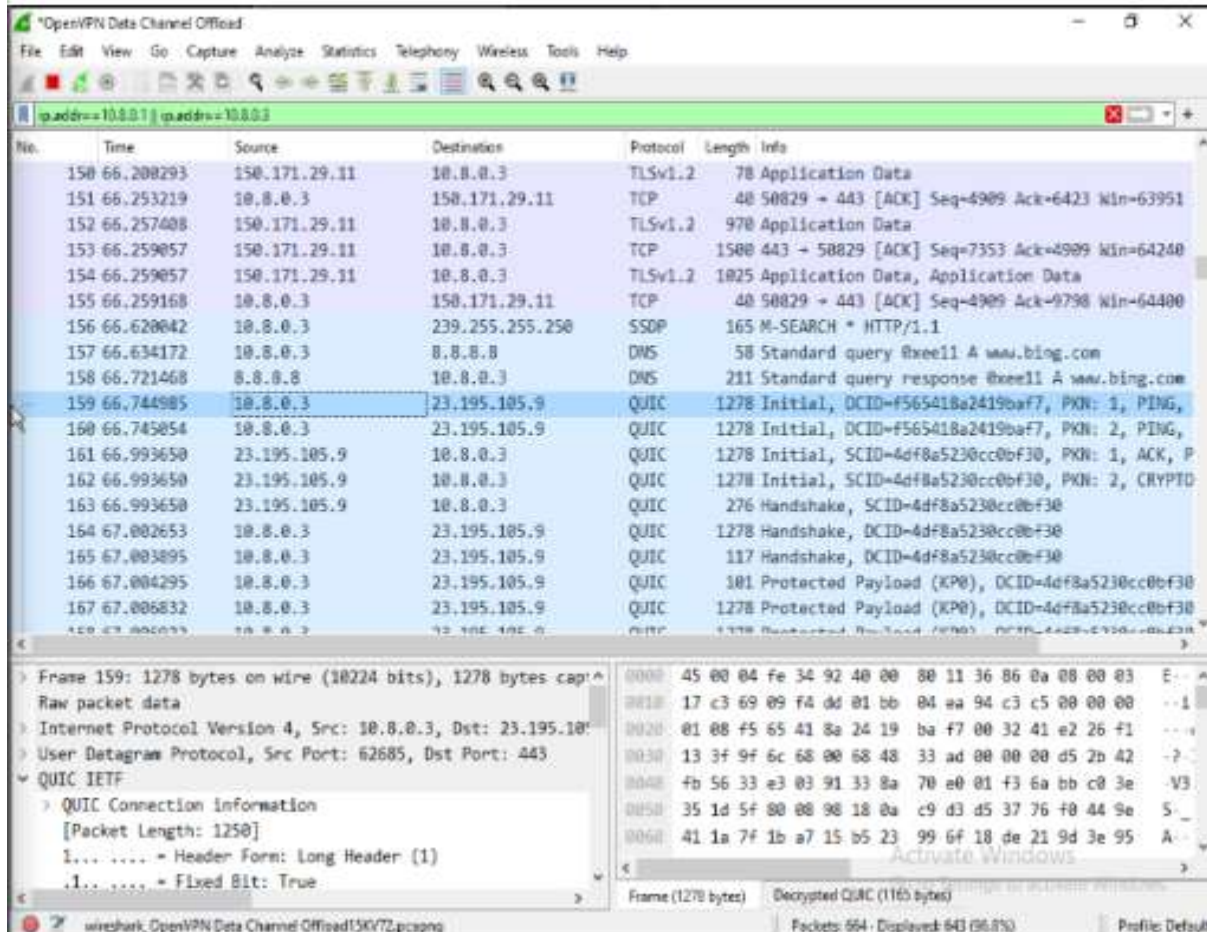
```

```

.bing.com.edgekey.net., CNAME e86303.dscx.akamaiedge.net., A 23.195.105.9, A 23.195.105.41 (183)
19:22:28.776305 IP 10.8.0.3.62685 > a23-195-105-9.deploy.static.akamaitechnologies.com.https: UDP, length 1250
19:22:28.776510 IP 10.8.0.3.62685 > a23-195-105-9.deploy.static.akamaitechnologies.com.https: UDP, length 1250
19:22:29.023274 IP a23-195-105-9.deploy.static.akamaitechnologies.com.https > 10.8.0.3.62685: UDP, length 1250
19:22:29.023376 IP a23-195-105-9.deploy.static.akamaitechnologies.com.https > 10.8.0.3.62685: UDP, length 1250
19:22:29.023380 IP a23-195-105-9.deploy.static.akamaitechnologies.com.https > 10.8.0.3.62685: UDP, length 248
19:22:29.036450 IP 10.8.0.3.62685 > a23-195-105-9.deploy.static.akamaitechnologies.com.https: UDP, length 1250
19:22:29.036709 IP 10.8.0.3.62685 > a23-195-105-9.deploy.static.akamaitechnologies.com.https: UDP, length 89
19:22:29.036847 IP 10.8.0.3.62685 > a23-195-105-9.deploy.static.akamaitechnologies.com.https: UDP, length 73
19:22:29.038146 IP 10.8.0.3.62685 > a23-195-105-9.deploy.static.akamaitechnologies.com.https: UDP, length 1250
19:22:29.038425 IP 10.8.0.3.62685 > a23-195-105-9.deploy.static.akamaitechnologies.com.https: UDP, length 1250
19:22:29.038590 IP 10.8.0.3.62685 > a23-195-105-9.deploy.static.akamaitechnologies.com.https: UDP, length 595
^C
241 packets captured
241 packets received by filter
0 packets dropped by kernel

```


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

```

[~]-[user@parrot:~]-
$ sudo nmap -sS 10.8.0.3
Starting Nmap 7.91 ( https://nmap.org ) at 2025-06-10 18:56 UTC
Nmap scan report for 10.8.0.3
Host is up (0.0026s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5357/tcp   open  wsdapi
Nmap done: 1 IP address (1 host up) scanned in 12.26 seconds
  
```

- Attempt unauthorized file access using:

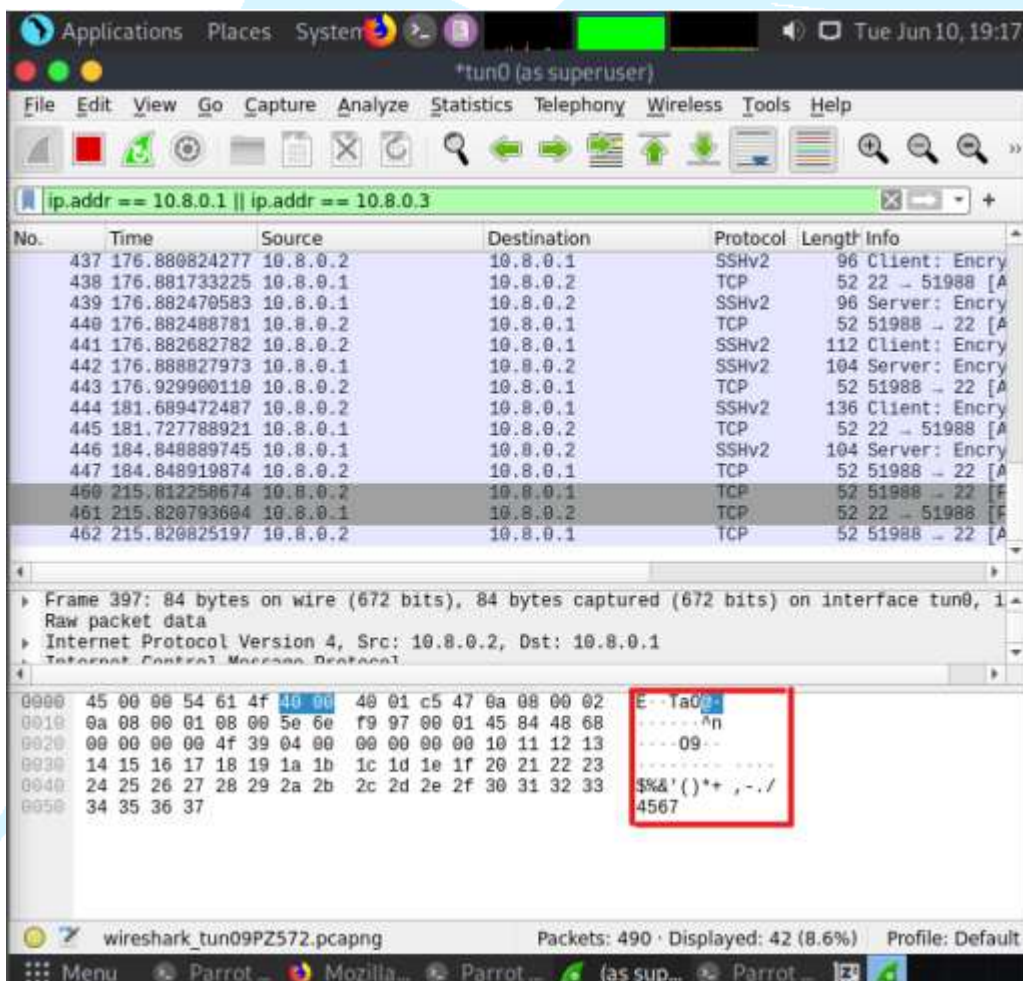
Step 2: Try Accessing Shared Files via SMB

```
(user@parrot)-[~]
$ smbclient //10.8.0.3/c$ -u desktop-cl4mkam\z4185
session setup failed: NT STATUS_ACCESS_DENIED
```

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).

The screenshot shows the Wireshark network traffic analysis interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for packet capture and analysis. The packet list pane shows a table of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The packet details pane shows the raw packet data for a selected packet, with a red box highlighting the ASCII representation of the data.

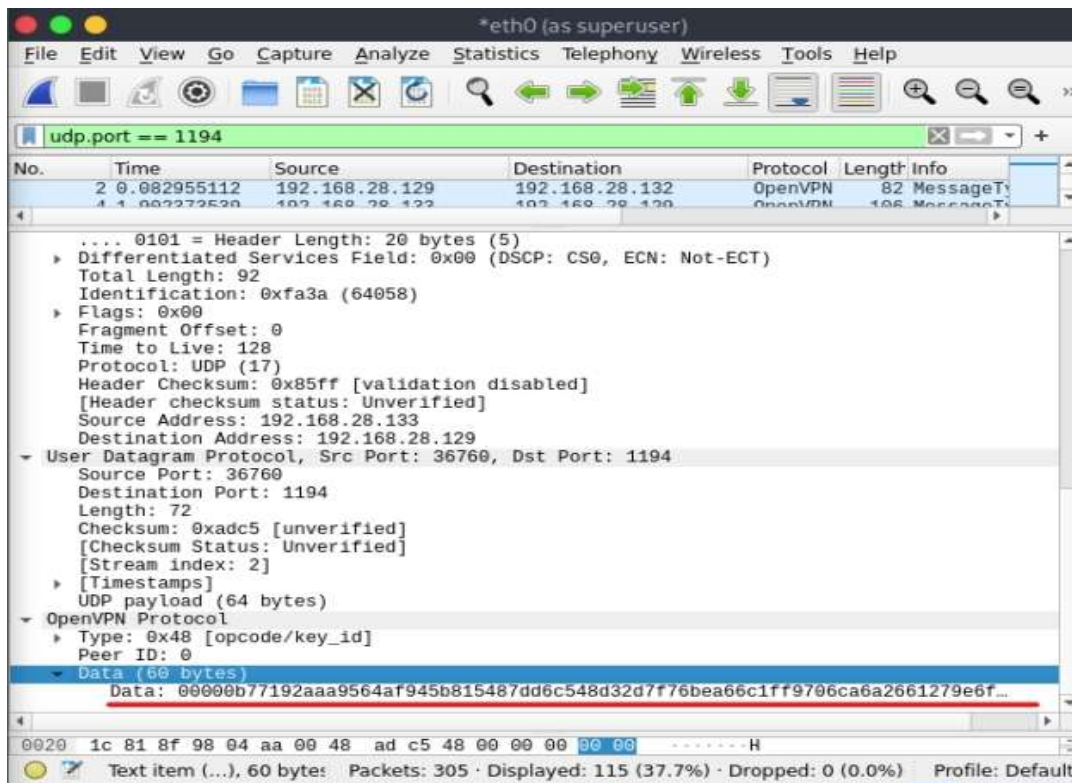
No.	Time	Source	Destination	Protocol	Length	Info
437	176.880824277	10.8.0.2	10.8.0.1	SSHv2	96	Client: Encry
438	176.881733225	10.8.0.1	10.8.0.2	TCP	52	22 → 51988 [A
439	176.882470583	10.8.0.1	10.8.0.2	SSHv2	96	Server: Encry
440	176.882488781	10.8.0.2	10.8.0.1	TCP	52	51988 → 22 [A
441	176.882682782	10.8.0.2	10.8.0.1	SSHv2	112	Client: Encry
442	176.888827973	10.8.0.1	10.8.0.2	SSHv2	104	Server: Encry
443	176.929900110	10.8.0.2	10.8.0.1	TCP	52	51988 → 22 [A
444	181.689472487	10.8.0.2	10.8.0.1	SSHv2	136	Client: Encry
445	181.727788921	10.8.0.1	10.8.0.2	TCP	52	22 → 51988 [A
446	184.848889745	10.8.0.1	10.8.0.2	SSHv2	104	Server: Encry
447	184.848919874	10.8.0.2	10.8.0.1	TCP	52	51988 → 22 [A
460	215.812258674	10.8.0.2	10.8.0.1	TCP	52	51988 → 22 [F
461	215.820793604	10.8.0.1	10.8.0.2	TCP	52	22 → 51988 [F
462	215.820825197	10.8.0.2	10.8.0.1	TCP	52	51988 → 22 [A

Frame 397: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface tun0, 1
Raw packet data
Internet Protocol Version 4, Src: 10.8.0.2, Dst: 10.8.0.1
Internet Control Message Protocol

0000 45 00 00 54 61 4f 40 00 40 01 c5 47 0a 08 00 02 E..Ta0.
0010 0a 08 00 01 08 00 5e 6e f9 97 00 01 45 84 48 68^n
0020 00 00 00 00 4f 39 04 00 00 00 00 00 10 11 12 1309..
0030 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
0040 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 \$%&'()*+,-./
0050 34 35 36 37 4567

- 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)

The screenshot shows a Wireshark capture on interface `tun0`. The filter is `ip.addr == 10.8.0.1 || ip.addr == 10.8.0.3 || ip.addr == 10.8.0.2`. The packet list shows various protocols including TCP, HTTP, and TLSv1.2. Packet 218 is highlighted in red, showing a TCP RST from 10.8.0.2 to 34.107.221.82. The packet details pane for packet 221 shows the raw packet data, which is a TCP segment with sequence number 59678 and acknowledgment number 217.

No.	Time	Source	Destination	Protocol	Length	Info
216	63.549786615	10.8.0.2	34.107.221.82	TCP	40	59678 → 80 [ACK] Seq=300 Ack=217 Win=64024 Len=0
217	64.859828064	34.107.221.82	10.8.0.2	HTTP	256	HTTP/1.1 200 OK (text/plain)
218	64.861386983	10.8.0.2	34.107.221.82	TCP	40	59684 → 80 [RST] Seq=2895 Win=0 Len=0
219	66.355744984	10.8.0.2	34.107.243.93	TLSv1.2	79	Application Data
220	66.356383955	34.107.243.93	10.8.0.2	TCP	40	443 → 53784 [ACK] Seq=313 Ack=352 Win=64240 Len=0
221	66.408362852	34.107.243.93	10.8.0.2	TLSv1.2	79	Application Data
222	66.401663889	10.8.0.2	34.107.243.93	TCP	40	53784 → 443 [ACK] Seq=352 Ack=352 Win=64028 Len=0
223	73.823923233	10.8.0.2	34.107.243.93	TLSv1.2	79	Application Data
224	73.824330500	34.107.243.93	10.8.0.2	TCP	40	443 → 53784 [ACK] Seq=352 Ack=391 Win=64240 Len=0
225	74.524818681	10.8.0.2	34.107.221.82	TCP	40	[TCP Keep-Alive] 59688 → 80 [ACK] Seq=1176 Ack=865
226	74.809872537	10.8.0.2	34.107.221.82	TCP	40	[TCP Keep-Alive] 59678 → 80 [ACK] Seq=299 Ack=217
227	75.662685512	10.8.0.2	34.107.221.82	TCP	40	[TCP Keep-Alive] 59688 → 80 [ACK] Seq=1176 Ack=865
228	75.662936303	34.107.221.82	10.8.0.2	TCP	40	[TCP Keep-Alive ACK] 80 → 59688 [ACK] Seq=865 Ack=
229	75.947792946	10.8.0.2	34.107.221.82	TCP	40	[TCP Keep-Alive] 59678 → 80 [ACK] Seq=299 Ack=217
230	75.948171315	34.107.221.82	10.8.0.2	TCP	40	[TCP Keep-Alive ACK] 80 → 59678 [ACK] Seq=217 Ack=

Frame 221: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface tun0, id 0
Raw packet data
Internet Protocol Version 4, Src: 34.107.243.93, Dst: 10.8.0.2
Transmission Control Protocol, Src Port: 443, Dst Port: 53784
Transport Layer Security

2. Capture Decrypted VPN Traffic (Inside the Tunnel)

The screenshot shows a Wireshark capture on interface `ens33`. The filter is `ip.addr == 192.168.28.129 || ip.addr == 192.168.28.132 || ip.addr == 192.168.28.133`. The packet list shows various protocols including OpenVPN and TLSv1.2. Packet 246 is highlighted in blue, showing a TCP segment with sequence number 53808 and acknowledgment number 443. The packet details pane for packet 246 shows the raw packet data, which is a TCP segment with sequence number 53808 and acknowledgment number 443.

No.	Time	Source	Destination	Protocol	Length	Info
244	105.158111785	192.168.28.133	192.168.28.129	OpenVPN	186	MessageType: P_DATA_V2
245	105.158112825	192.168.28.133	192.168.28.129	OpenVPN	186	MessageType: P_DATA_V2
246	105.158410036	192.168.28.129	34.107.243.93	TCP	54	53808 → 443 [ACK] Seq=1 Ack=25 Win=62768 Len=0
247	105.499854513	192.168.28.133	192.168.28.129	OpenVPN	134	MessageType: P_DATA_V2
248	105.499854756	192.168.28.133	192.168.28.129	OpenVPN	134	MessageType: P_DATA_V2
249	105.500874118	192.168.28.129	34.107.243.93	TLSv1.2	82	Application Data
250	105.500882578	34.107.243.93	192.168.28.129	TCP	60	443 → 53808 [ACK] Seq=25 Ack=28 Win=64240 Len=0
251	105.501580388	192.168.28.129	192.168.28.133	OpenVPN	186	MessageType: P_DATA_V2
252	105.501580388	192.168.28.129	192.168.28.133	TLSv1.2	79	Application Data

Frame 246: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface ens33, id 0
Section number: 1
Interface id: 0 (ens33)
Interface name: ens33
Encapsulation type: Ethernet (1)
Arrival Time: Jun 18, 2025 28:38:55.277829572 PKT
UTC Arrival Time: Jun 18, 2025 15:38:55.277829572 UTC
Epoch Arrival Time: 1749569935.277829572
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000298011 seconds]
[Time delta from previous displayed frame: 0.000298011 seconds]
[Time since reference or first frame: 105.158410036 seconds]
Frame Number: 246
Frame Length: 54 bytes (432 bits)
Capture Length: 54 bytes (432 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp]
[Coloring Rule Name: TCP]
[Coloring Rule String: tcp]
Ethernet II, Src: VMware_3d:c8:be (00:0c:29:3d:c8:be), Dst: VMware_f3:4a:c6 (00:50:56:f3:4a:c6)
Internet Protocol Version 4, Src: 192.168.28.129, Dst: 34.107.243.93
Transmission Control Protocol, Src Port: 53808, Dst Port: 443, Seq: 1, Ack: 25, Len: 0

PART 5: Create Security Rules to Prevent Insider Attack

On Ubuntu (VPN Server):

- UFW firewall rules:

```
Skipping adding existing rule
Skipping adding existing rule (v6)
```

```
Skipping adding existing rule
Skipping adding existing rule (v6)
```

```
Firewall is active and enabled on system startup
```

Block access between VPN clients (client isolation):

- Edit /etc/openvpn/server.conf:

```
zunaïra@zunaïra-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
```

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)