Network Security VPN's

Assignment 1:

OpenVPN-Client-Connection-and-IP-Verification

This assignment demonstrates setting up an OpenVPN server on a Linux machine, establishing a VPN connection from a client (same machine), and verifying the change in public IP address before and after connecting to the VPN. The goal is to create a self-hosted VPN environment for secure remote access and privacy. Since I already have an .ovpn configuration file, I can directly use it to connect to the VPN and verified my IP address before and after the connection,

1. Introduction

For this, I used an existing OpenVPN configuration file (.ovpn) to connect my Linux machine to a VPN. I verified the public IP address before and after the VPN connection to demonstrate secure tunneling.

2. Tools Used

- OpenVPN client on Linux
- curl for IP address verification

3. Procedure

Step 1: Check public IP before VPN connection

I ran the command:

```
curl ifconfig.me
```

This showed my real public IP address before the VPN connection.

Step 2: Connect to the VPN using .ovpn file

Using the OpenVPN client, I connected to the VPN:

```
sudo openvpn --config /vpn/client.ovpn
```

I provided the required credentials when prompted, and the connection was successfully established.

Step 3: Verify public IP after VPN connection

In a new terminal session, I ran:

```
curl ifconfig.me
```

This showed the VPN server's public IP address, confirming that my traffic was routed through the VPN.

4. Results

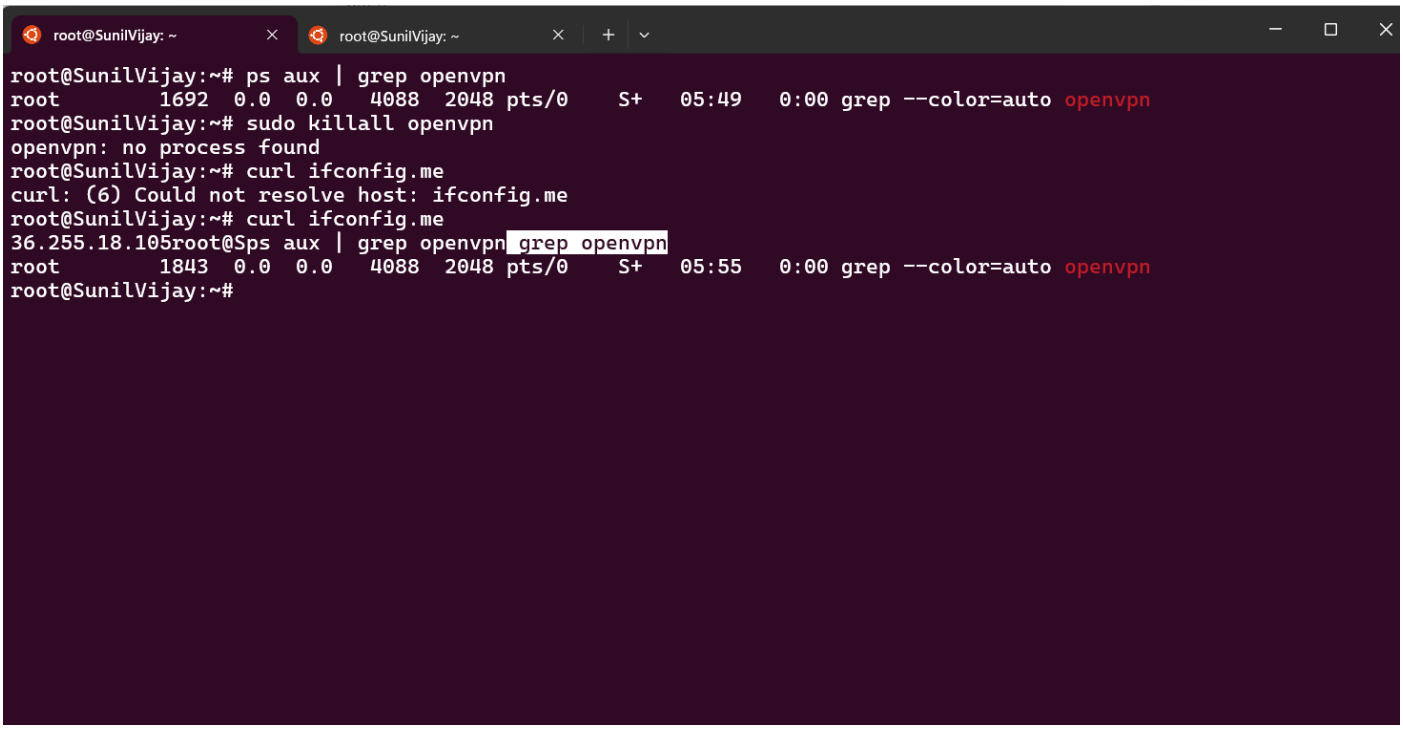
- Original IP (before connection): [36.255.18.105]
- New IP (after connection): [202.21.42.138]

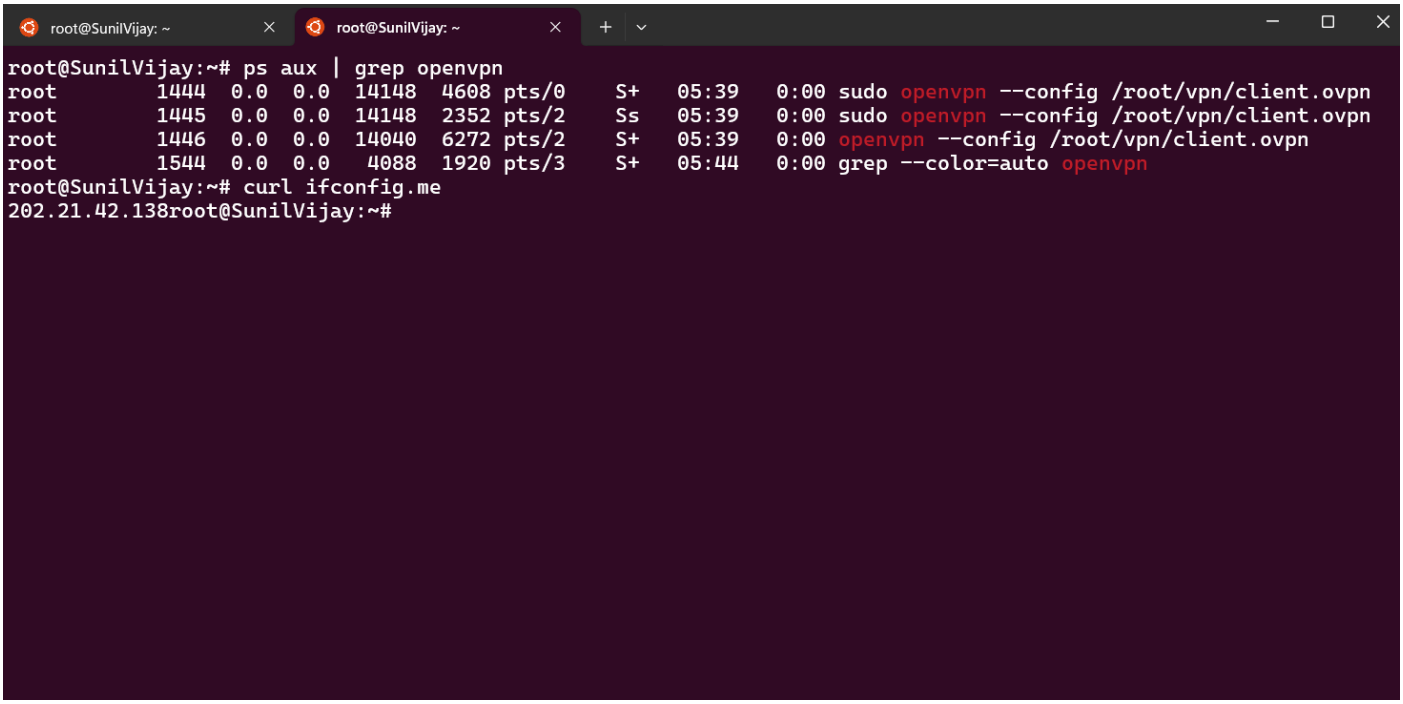
5. Conclusion

Using the provided OpenVPN configuration, I successfully connected to a VPN on my Linux system and verified that my IP address changed, indicating encrypted and private network communication.

6. Attachments

- configuration file
- Terminal outputs for IP before and after connection (Screenshots)





Assignment 2

Nmap-scanning

Nmap is a powerful open-source tool used to scan websites and network hosts by sending packets and analyzing their responses. It helps discover open ports, running services, and potential vulnerabilities, assisting security professionals in network auditing and protection tasks. To properly submit your Nmap scan as a GitHub report or assignment, include the following sections: Introduction, Methodology (scan commands used), Results (with open ports and explanations), and Analysis. Save outputs in plain text or XML and attach these files as evidence in your repository or assignment folder.

Nmap Assignment Report

1. Introduction

This report documents an Nmap scan performed on the TryHackMe website to identify open ports and running services. The objective is to gain insights into the public-facing services for security assessment and documentation purposes.

2. Methodology

Scan Command Used:

```
nmap www.tryhackme.com -oN tryhachmereport_nmap.txt
```

- -oN outputs in normal text format for documentation.

3. Results

Scan Output Summary:

Port	State	Service	Description
80/tcp	open	http	Standard web server port for unencrypted web traffic
443/tcp	open	https	Secure port for encrypted SSL/TLS web traffic
8080/tcp	open	http-proxy	Often used for development, web proxies, or alternate HTTP services

The scan identified three open ports: 80 (HTTP), 443 (HTTPS), and 8080 (HTTP-proxy), all serving web-based content/services.

4. Analysis

- Port 80: Hosts the default web page; traffic is unencrypted.
- Port 443: Provides encrypted, secure web traffic via HTTPS.
- Port 8080: May provide alternate HTTP services or function as a proxy, often used for admin panels or alternate web applications.

tryhackmeoutput_nmap.txt

Command Prompt

```
Microsoft Windows [Version 10.0.26100.6725]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sunil>nmap tryhackme.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-16 12:15 India Standard Time
Nmap scan report for tryhackme.com (104.20.29.66)
Host is up (0.093s latency).
Other addresses for tryhackme.com (not scanned): 172.66.164.239
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
8080/tcp open  http-proxy
8443/tcp open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 53.43 seconds

C:\Users\sunil>
```