CYBER SECURITY LABORATORY

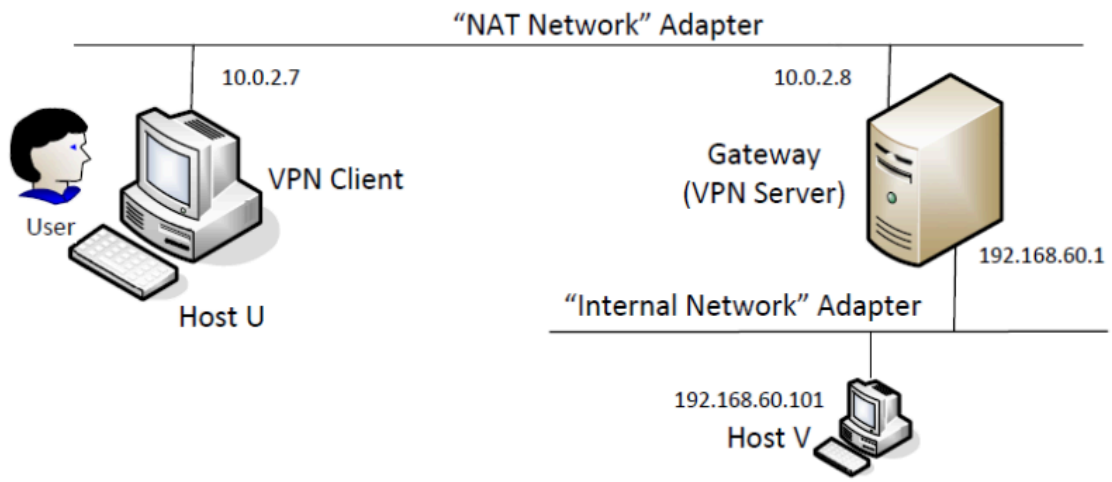
Virtual Private Network (VPN)

Objective:

The aim of this lab is to establish a basic VPN using TLS/SSL. This exercise helps in comprehending fundamental networking and security concepts such as:

- Virtual Private Networks (VPNs)
- TUN/TAP Interfaces and IP Tunneling
- Network Routing
- Public-Key Cryptography and X.509 Certificates
- TLS/SSL Implementation
- Authentication Mechanisms

Virtual Machine Setup:



Downloading Required Files and Initial Configuration

1. Obtain the necessary VPN setup files from the Seed Labs website.

2. Temporarily disable the Uncomplicated Firewall (UFW) using:

```
[03/25/25]seed@VM:~$ sudo ufw disable
Firewall stopped and disabled on system startup
[03/25/25]seed@VM:~$ sudo ufw status
Status: inactive
[03/25/25]seed@VM:~$ sudo nano /etc/sysctl.conf
[03/25/25]seed@VM:~$
```

Enable IPv4 forwarding by setting it to 1.

```
seed@VM: ~
  GNU nano 4.8                    /etc/sysctl.conf

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1


^G Get Help   ^O Write Out ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File ^\ Replace    ^U Paste Text^T To Spell   ^_ Go To Line
```

```
[03/25/25]seed@VM:~$ sudo nano /etc/sysctl.conf
[03/25/25]seed@VM:~$
[03/25/25]seed@VM:~$
[03/25/25]seed@VM:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
[03/25/25]seed@VM:~$
```

Clone the virtual machines for the experiment.

Setting Up Virtual Machines in VirtualBox

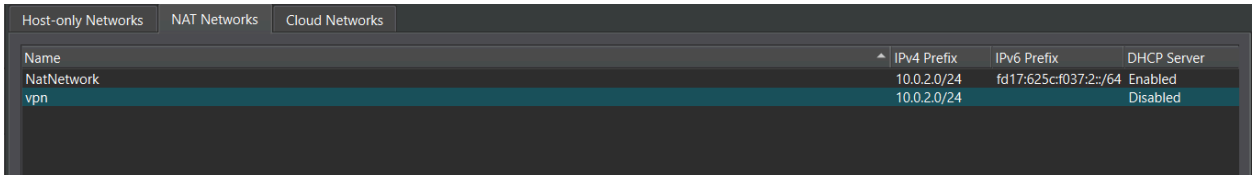
Step 1: Create Three Virtual Machines

Install VirtualBox and configure three separate virtual machines:

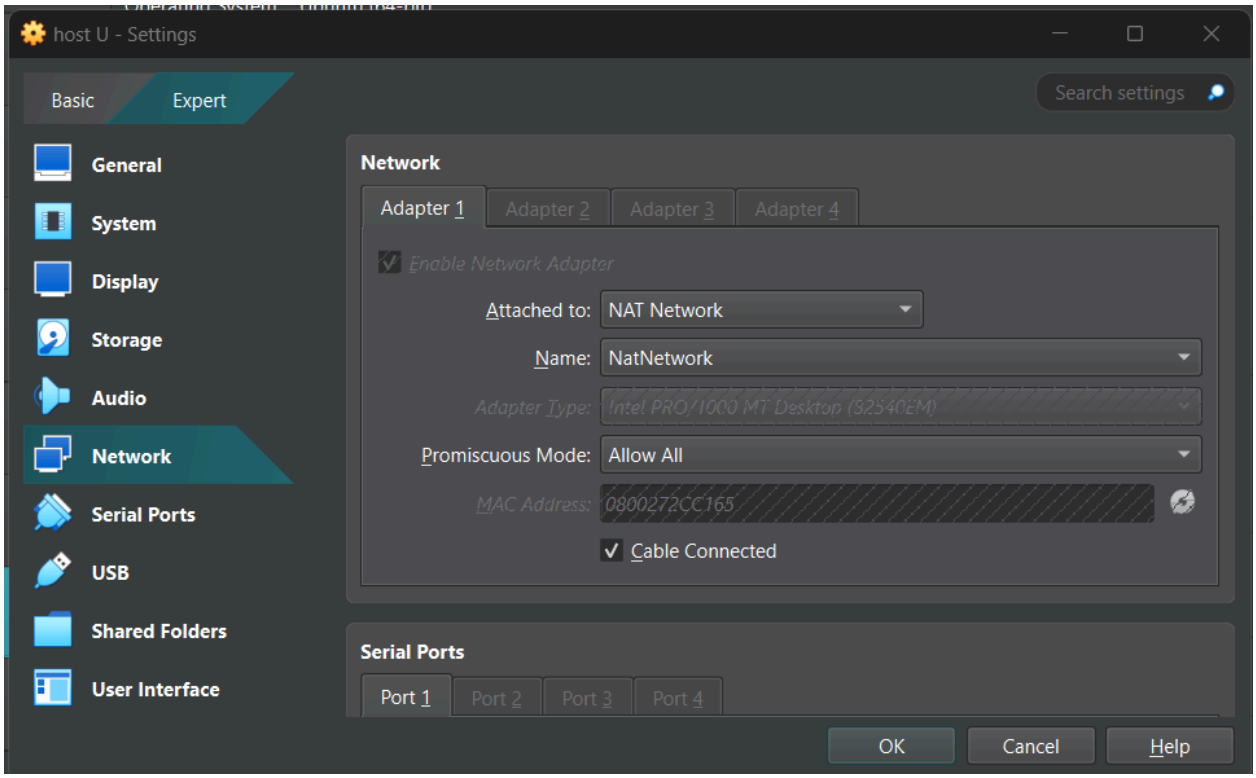
- VPN Client (Host U)
- VPN Server
- Host V

Step 2: Configuring Network Interfaces

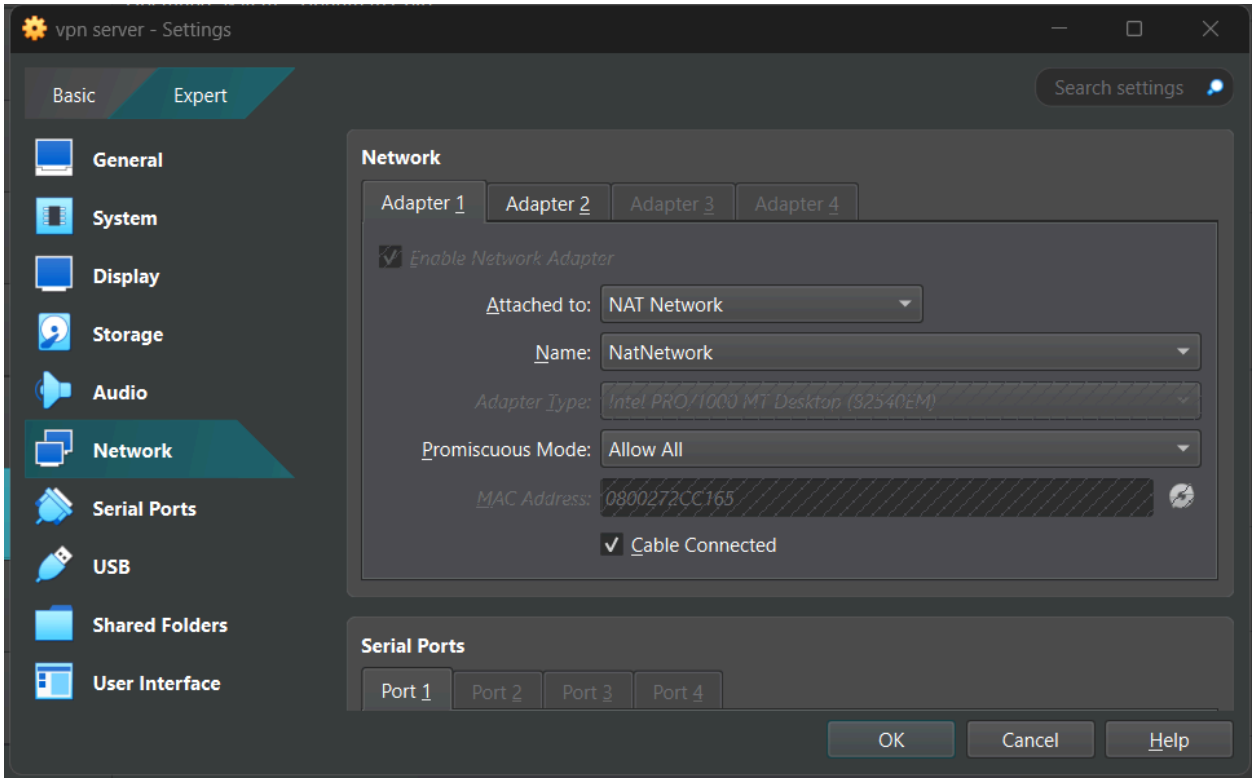
Adjust the network settings of each virtual machine as follows:



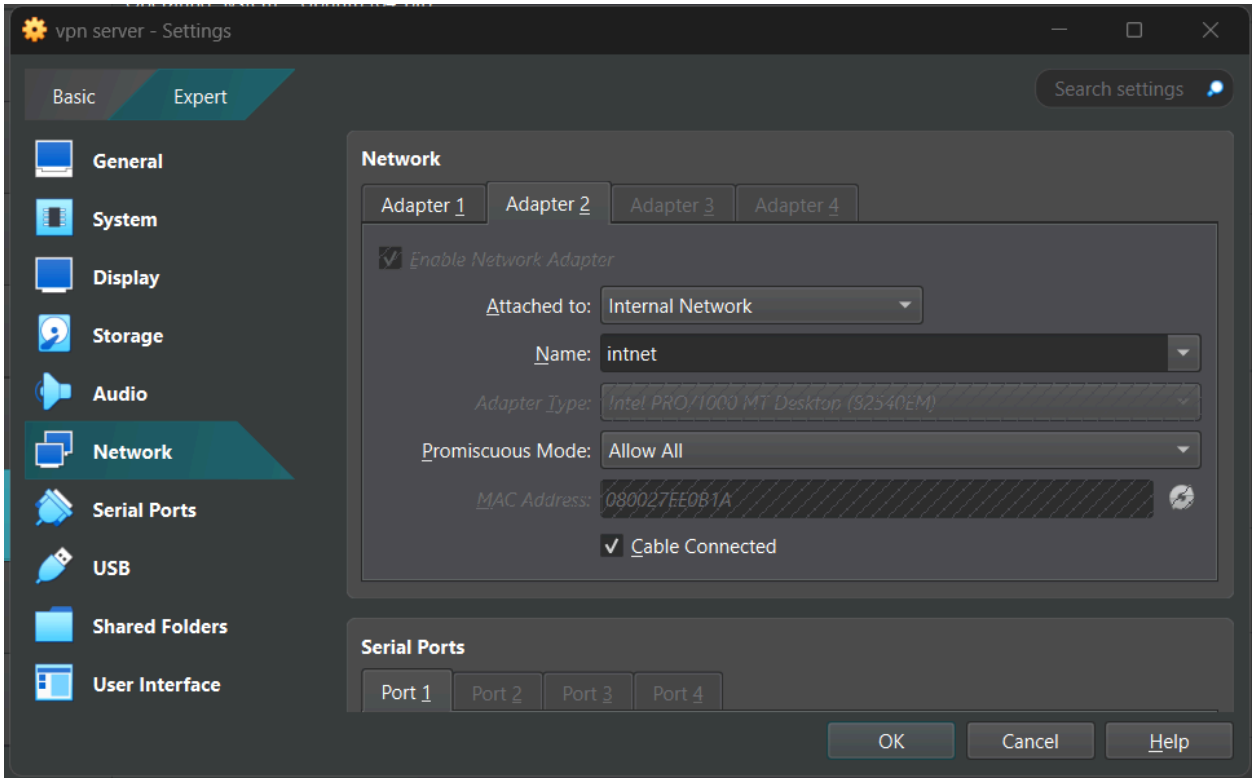
- VPN Client (Host U)
    - Adapter 1: NAT Network (For VPN connection to the server)



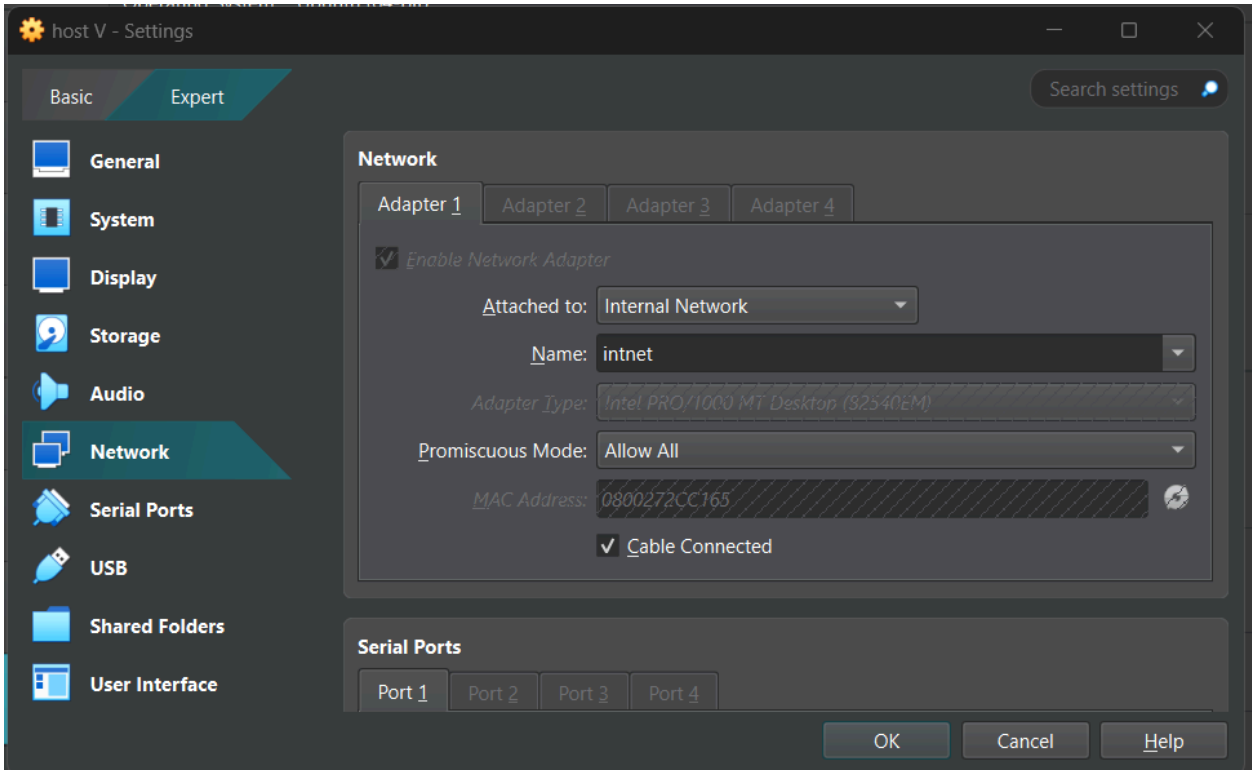
- VPN Server
    - Adapter 1: NAT Network (Simulating an external internet connection)



- Adapter 2: Internal Network (Private communication with Host V)



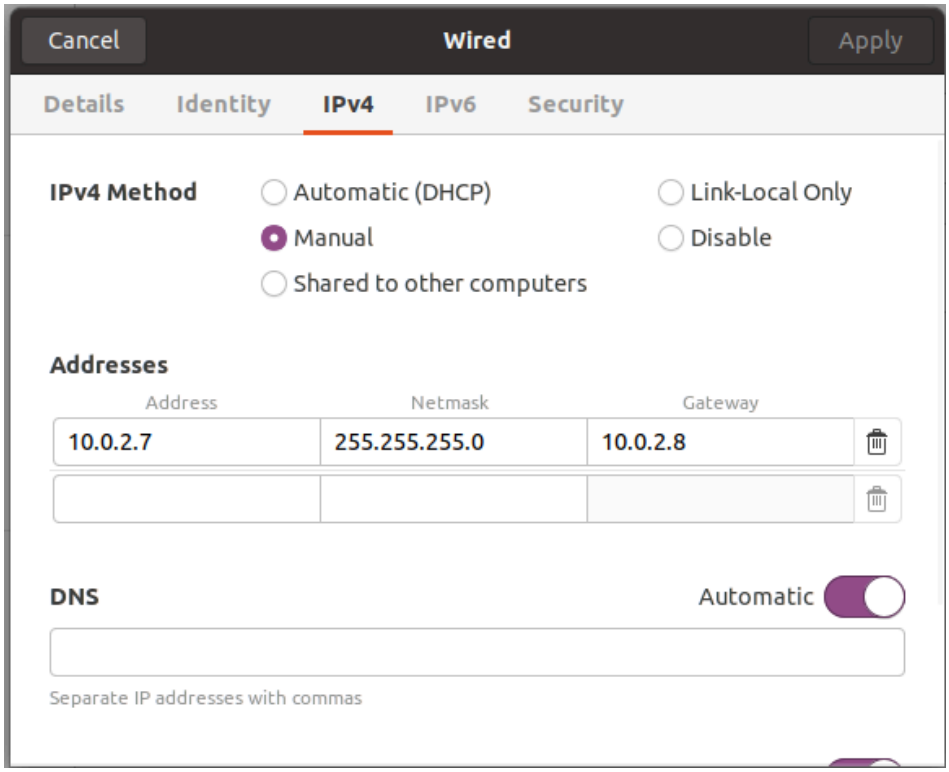
- Host V
    - Adapter 1: Internal Network (Accessible only via the VPN Server)



Network Configuration

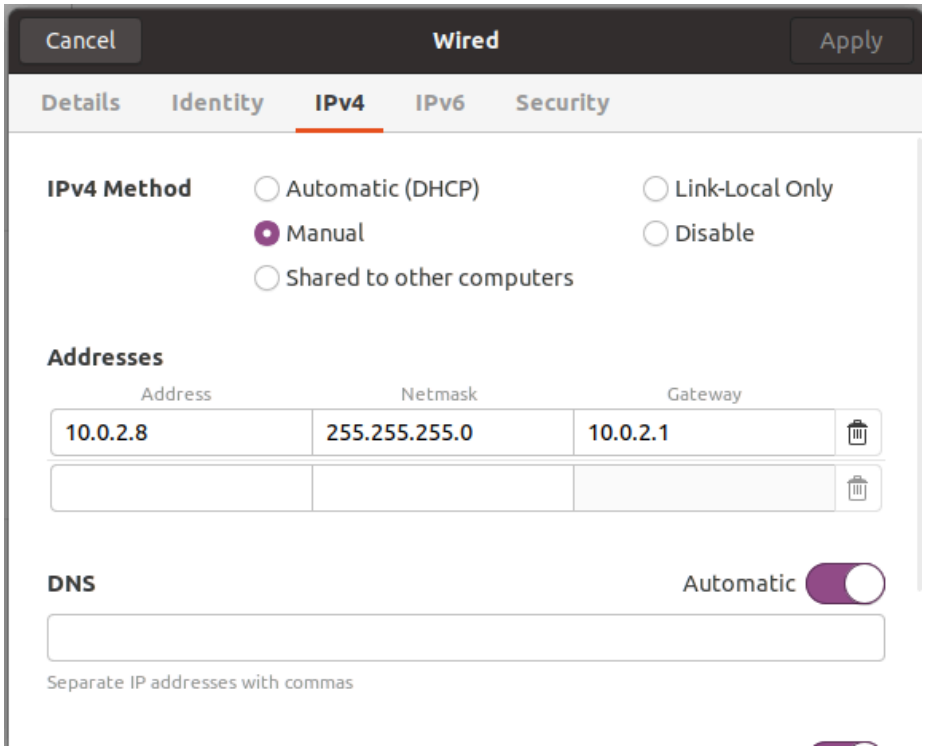
Each machine is assigned a specific IP address and network gateway:

1. VPN Client (Host U)
    - IP Address: 10.0.2.7
    - Gateway: 10.0.2.8
    - Network Adapter: NAT Network

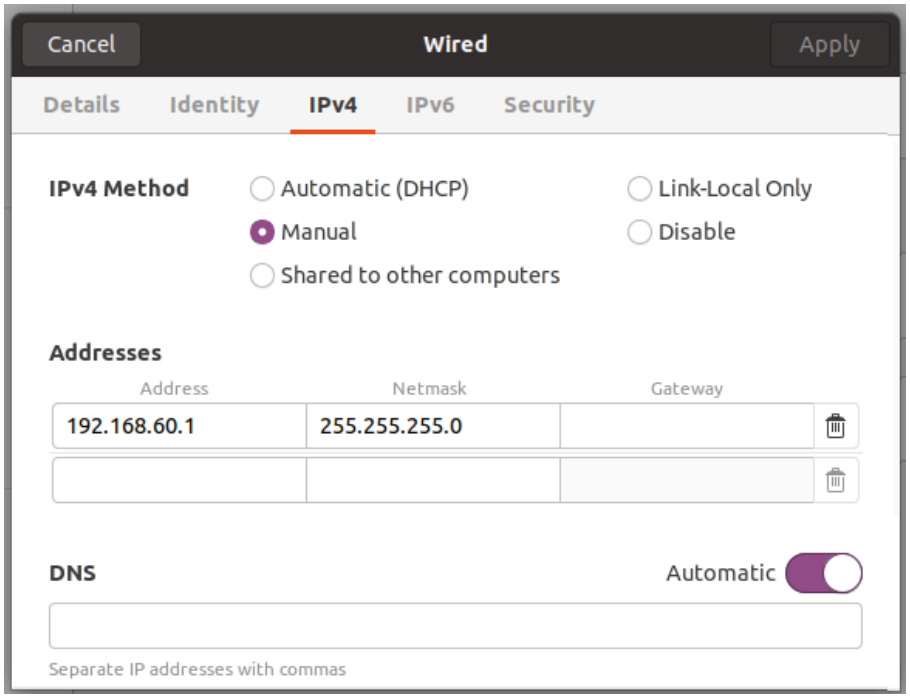


2. VPN Server (Gateway Machine)

- External Network (Internet-facing) Interface:
    - IP Address: 10.0.2.8
    - Gateway: 10.0.2.1



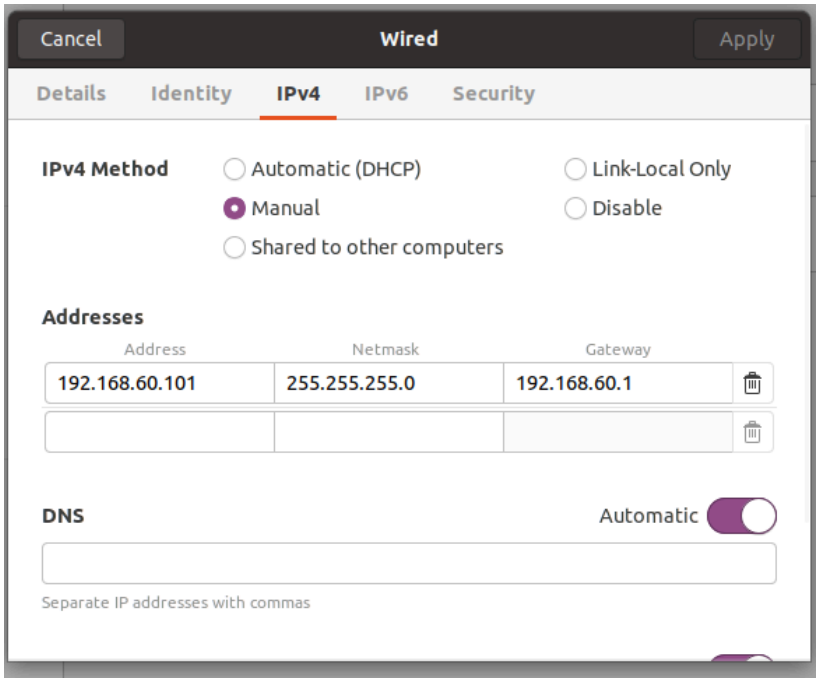
- Internal Network (Private) Interface:
    - IP Address: 192.168.60.1
    - Gateway: None



- Network Adapters:
    - Adapter 1: NAT Network (Connected to Host U)
    - Adapter 2: Internal Network (Connected to Host V)

3. Host V

- IP Address: 192.168.60.101
- Gateway: 192.168.60.1
- Network Adapter: Internal Network



Network Topology:

- The VPN Client (Host U) establishes a connection to the VPN Server over a NAT Network (simulating an internet connection).
- The VPN Server serves as an intermediary, linking Host U with Host V through an Internal Network.
- Host V is only reachable through the VPN Server, making it inaccessible from outside networks.
- Since the Internal Network does not provide DHCP, the IP address of Host V must be assigned manually.

VPN Implementation Steps:

- Successfully install and configure three virtual machines in VirtualBox.

- Assign appropriate network adapters to each VM based on the above configurations.

- Ensure the VPN Server has both NAT and Internal Network adapters set up correctly.

- Manually configure the IP address of Host V, as there is no DHCP service available in the Internal Network.

Compiling the VPN Scripts

- Open vpnclient.c and update the code to include the VPN Server's IP address (from the NAT network).

```
seed@VM: ~/.../vpn
seed@VM: ~/.../vpn        seed@VM: ~/Downloads
[03/25/25]seed@VM:~/.../vpn$ ls
Makefile  README  vpnclient.c  vpnserver.c
[03/25/25]seed@VM:~/.../vpn$ nano vpnclient.c
```

```
seed@VM: ~/.../vpn
seed@VM: ~/.../vpn        seed@VM: ~/Downloads        seed@VM: ~/.../vpn
  GNU nano 4.8                    vpnclient.c
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#include <linux/if.h>
#include <linux/if_tun.h>
#include <sys/ioctl.h>

#define BUFF_SIZE 2000
#define PORT_NUMBER 55555
#define SERVER_IP "10.0.2.8"
struct sockaddr_in peerAddr;

int createTunDevice() {
   int tunfd;
   struct ifreq ifr;
   memset(&ifr, 0, sizeof(ifr));

   ifr.ifr_flags = IFF_TUN | IFF_NO_PI;
                          [ Read 90 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

- Compile both the client and server VPN programs using:

```
seed@VM: ~/.../vpn    seed@VM: ~/.../vpn    seed@VM: ~/.../vpn
[03/25/25]seed@VM:~/.../vpn$ ll
total 16
-rw-r--r-- 1 seed seed   99 Mar 16  2018 Makefile
-rw-r--r-- 1 seed seed  668 Mar 16  2018 README
-rw-r--r-- 1 seed seed 2041 Mar 16  2018 vpnclient.c
-rw-r--r-- 1 seed seed 2225 Mar 16  2018 vpnserver.c
[03/25/25]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
```

Starting the VPN Server:

1. Launch the VPN Server program on the VPN Server machine

```
[03/25/25]seed@VM:~/.../vpn$ sudo ./vpnserver
```

2. Configure the TUN interface on the VPN Server

```
[03/25/25]seed@VM:~/.../vpn$ sudo ifconfig tun0 192.168.53.1/24 up
```

```
[03/25/25]seed@VM:~/.../vpn$ ifconfig
br-28b2da1ca06f: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 10.9.0.1  netmask 255.255.255.0  broadcast 10.9.0.255
        ether 02:42:57:dd:1c:44  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:54:47:4c:bb  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.8  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::b64:be08:7418:13d7  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::9128:9340:6f74:5482  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:2c:c1:65  txqueuelen 1000  (Ethernet)
        RX packets 246  bytes 41269 (41.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 312  bytes 41886 (41.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.60.1  netmask 255.255.255.0  broadcast 192.168.60.255
        inet6 fe80::1c79:d162:f016:828b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ee:0b:1a  txqueuelen 1000  (Ethernet)
        RX packets 95  bytes 20643 (20.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 168  bytes 25439 (25.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1292  bytes 122130 (122.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1292  bytes 122130 (122.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 192.168.53.1  netmask 255.255.255.0  destination 192.168.53.1
        inet6 fe80::552f:6250:509e:33c4  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Running the VPN Client:

1. On the VPN Client (Host U), start the client program

```
[03/25/25]seed@VM:~/.../vpn$ nano vpnclient.c
[03/25/25]seed@VM:~/.../vpn$
```

```
seed@VM: ~/.../vpn        seed@VM: ~/Downloads        seed@VM: ~/.../vpn
  GNU nano 4.8                    vpnclient.c
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <arpa/inet.h>
#include <linux/if.h>
#include <linux/if_tun.h>
#include <sys/ioctl.h>

#define BUFF_SIZE 2000
#define PORT_NUMBER 55555
#define SERVER_IP "10.0.2.8"
struct sockaddr_in peerAddr;

int createTunDevice() {
   int tunfd;
   struct ifreq ifr;
   memset(&ifr, 0, sizeof(ifr));

   ifr.ifr_flags = IFF_TUN | IFF_NO_PI;
                              [ Read 90 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text^T To Spell  ^_ Go To Line
```

```
[03/25/25]seed@VM:~/.../vpn$ sudo ./vpnclient
```

2. Set up the TUN interface:

```
[03/25/25]seed@VM:~/.../vpn$ sudo ifconfig tun0 192.168.53.5/24 up
```

```
[03/25/25]seed@VM:~/.../vpn$ sudo ifconfig tun0 192.168.53.5/24 up
[03/25/25]seed@VM:~/.../vpn$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:2c:c1:65 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.7/24 brd 10.0.2.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::9128:9340:6f74:5482/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: br-28b2da1ca06f: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:0e:49:eb:e2 brd ff:ff:ff:ff:ff:ff
    inet 10.9.0.1/24 brd 10.9.0.255 scope global br-28b2da1ca06f
       valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:f6:35:9f:89 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.5/24 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::6c17:1aa2:a48a:a991/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

```
[03/25/25]seed@VM:~/.../vpn$ sudo ./vpnclient
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
```

```
[03/25/25]seed@VM:~/.../vpn$ sudo ./vpnserver
Connected with the client: Hello
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
```

Once this is complete, the VPN Client should be connected to the VPN Server.

Configuring Routing:

To ensure proper communication, routing rules must be configured:

1. On the VPN Server, enable packet forwarding between the VPN tunnel and the internal network.

2. On the VPN Client, add a route for the private network (192.168.60.0/24) via the VPN tunnel:

```
[03/25/25]seed@VM:~/.../vpn$ sudo route add -net 192.168.60.0/24 tun0
```

3. On Host V, configure routing to direct packets back to Host U via the VPN Server:

```
[03/25/25]seed@VM:~$
[03/25/25]seed@VM:~$ sudo route add -net 192.168.53.0/24 gw 192.168.60.1
[03/25/25]seed@VM:~$
```

Testing the VPN Connection:

1. Ping Test: Verify if Host U can communicate with Host V

```
[03/25/25]seed@VM:~/.../vpn$ ping 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=63 time=6.24 ms
64 bytes from 192.168.60.101: icmp_seq=2 ttl=63 time=6.60 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=63 time=6.06 ms
64 bytes from 192.168.60.101: icmp_seq=4 ttl=63 time=5.38 ms
64 bytes from 192.168.60.101: icmp_seq=5 ttl=63 time=7.62 ms
^C
--- 192.168.60.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4024ms
```

2. Telnet Test: Attempt to establish a telnet connection from Host U to Host V:

```
[03/25/25]seed@VM:~/.../vpn$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
VM login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

If both tests are successful, the VPN tunnel is functioning correctly.

Tunnel Disruption Test

1. Establish a telnet connection from Host U to Host V:

```
[03/25/25]seed@VM:~/.../vpn$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
VM login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Your Hardware Enablement Stack (HWE) is supported until April 2025.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

Stop the VPN Client process on **Host U**, which will terminate the VPN tunnel:

```
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
^C
```

The telnet session should freeze, confirming that the VPN tunnel is essential for communication between Host U and Host V.

```
[03/25/25]seed@VM:~$ cd Documents
[03/25/25]seed@VM:~/Documents$ ll
total 8
drwxrwxr-x 3 seed seed 4096 Mar 19 01:35  Labsetup
-rw-rw-r-- 1 seed seed  959 Mar 19 01:34 'Labsetup(1).zip'
[03/25/25]seed@VM:~/Documents$ cd ~/Downloads/
[03/25/25]seed@VM:~/Downloads$ ll
total 600
drwxrwxr-x 5 seed seed   4096 Mar  9 11:36  format
-rw-rw-r-- 1 seed seed 198540 Mar 12 02:44 'Format_String(1).pdf'
-rw-rw-r-- 1 seed seed 198540 Mar 11 00:26  Format_String.pdf
-rw-rw-r-- 1 seed seed 190776 Feb 26 00:31  Format_String_Server.pdf
-rw-rw-r-- 1 seed seed    959 Mar 19 01:34 'Labsetup(1).zip'
-rw-rw-r-- 1 seed seed   5807 Feb 26 00:26  Labsetup.zip
-rw-rw-r-- 1 seed seed   2728 Mar 25 06:20  vpn.zip
[03/25/25]seed@VM:~/Downloads$
[03/25/25]seed@VM:~/Downloads$
```

The telnet connection becomes unresponsive

Conclusion

The implementation of the VPN was successful, and all expected functionalities operated correctly. The VPN tunnel securely established communication between Host U and Host V, with successful ping and telnet tests validating connectivity. The routing configurations ensured proper packet flow, and Wireshark analysis confirmed encrypted traffic within the VPN. During the tunnel-disruption test, the telnet session froze upon stopping the VPN Client, demonstrating the VPN’s necessity for maintaining the connection.
