Cyber Security Lab
Assignment 9

Virtual Private Network

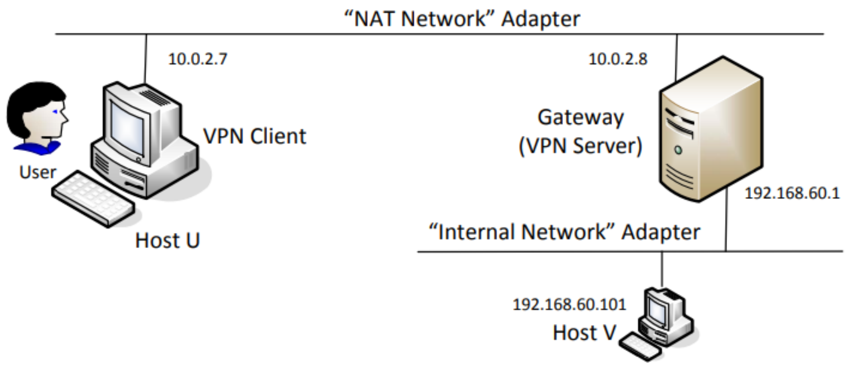
CYS24014 - Shree Varshaa R M

May 10, 2025

Introduction

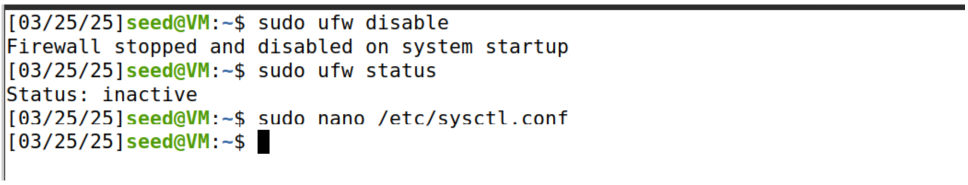
This document outlines the implementation of a simple TLS/SSL-based VPN (miniVPN) as part of the SEED Labs VPN assignment. The VPN establishes secure communication between a client (Host U) and a private network via a gateway server, allowing Host U to securely access Host V in the private network.

Virtual Machine Setup

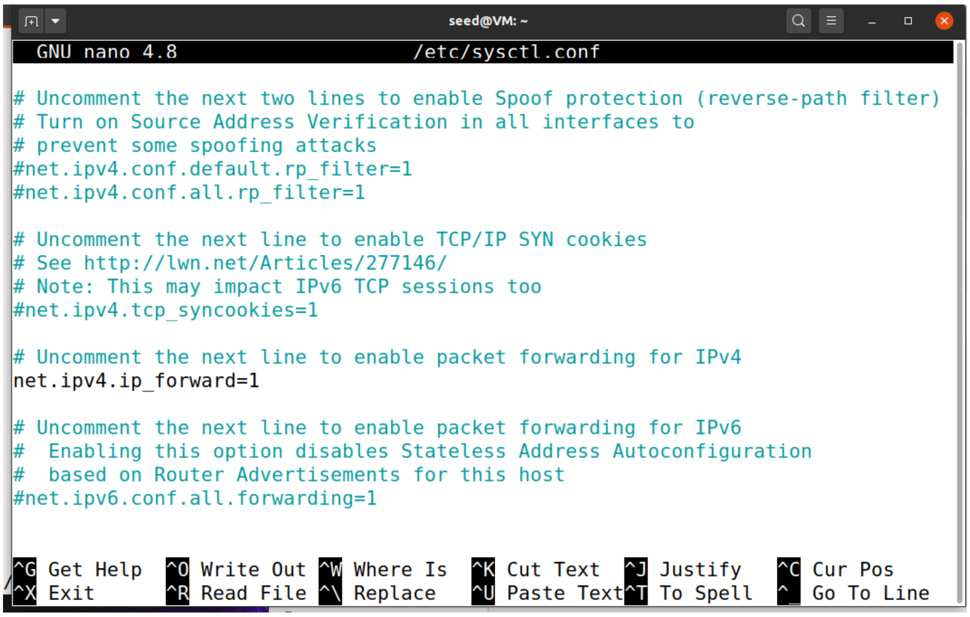


Downloading Required files and Initial Configuration

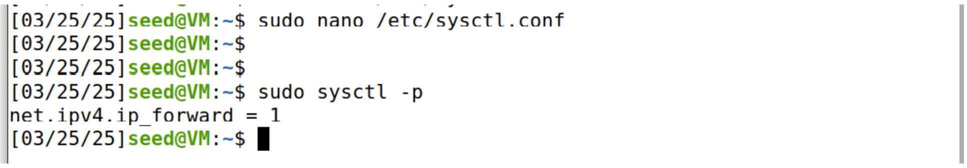
- Disabling uncomplicated firewall (UFW) temporarily
- Enabling ipv4 forwarding by setting it to 1



```
[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:~$ 
```



```
  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:~$
```

- Now, clone three virtual machines for this experiment

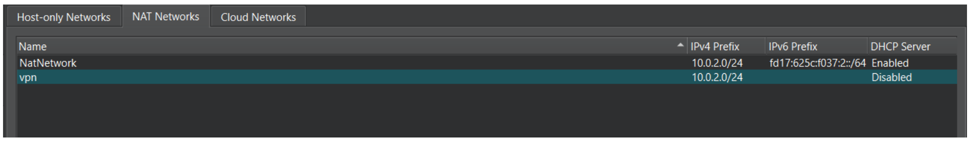
Lab Environmental Setup

Network Setup:

- Client and Server connected via ”NAT Network” (simulating Internet)
- Server and Host V connected via ”Internal Network”

Configuring Network Interfaces:

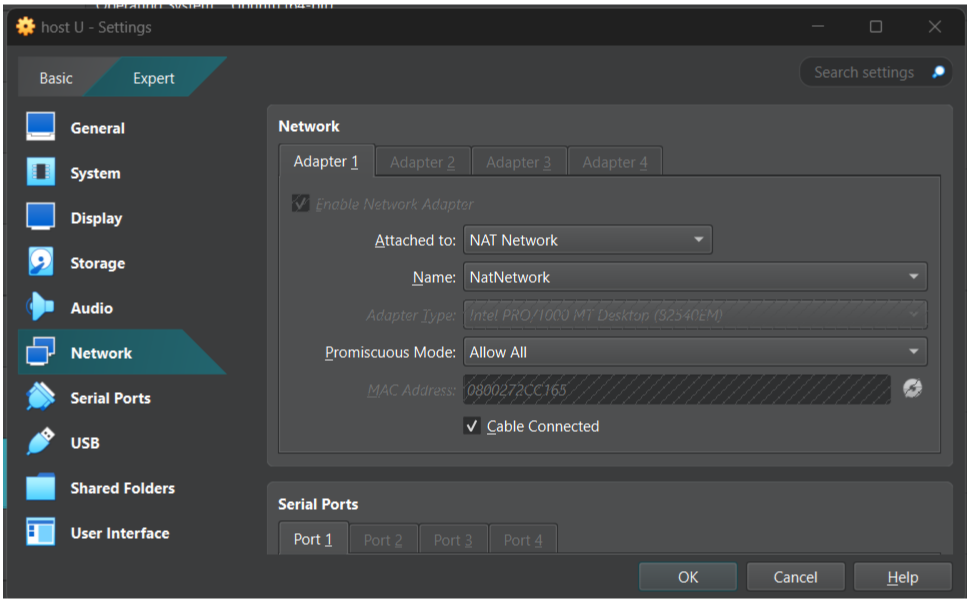
- Adjusting Network setting for each Virtual Machine



Name	IPv4 Prefix	IPv6 Prefix	DHCP Server
NatNetwork	10.0.2.0/24	fd17:625c:f037:2::/64	Enabled
vpn	10.0.2.0/24		Disabled

Configuring three virtual machines:

- **VPN Client (Host U):** Connects to the VPN server
- **Adapter 1:** NAT Network



host U - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: NAT Network

Name: NatNetwork

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

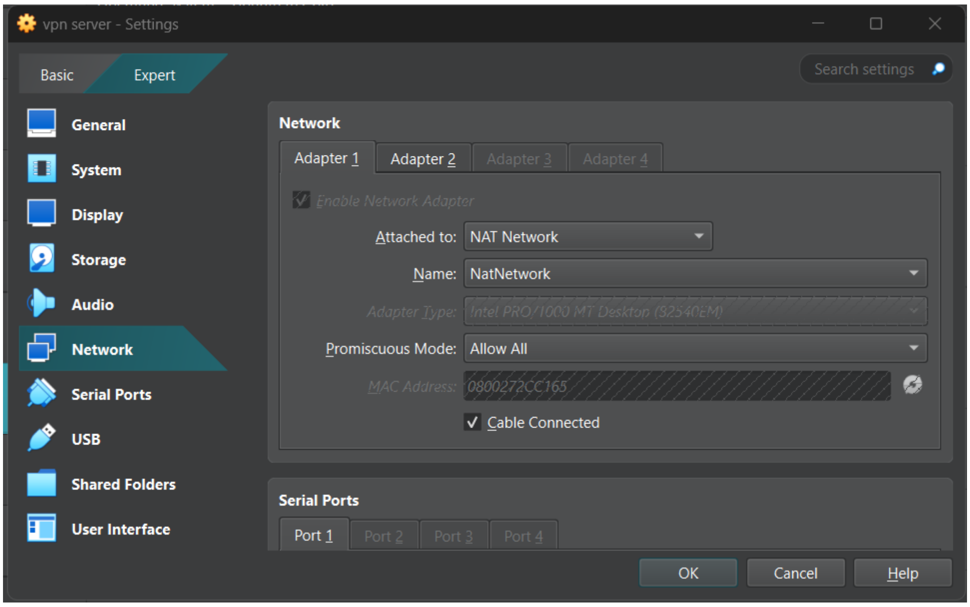
MAC Address: 0800272CC165

☑ Cable Connected

Serial Ports

Port 1 | Port 2 | Port 3 | Port 4

- **VPN Server:** Acts as gateway to the private network
- **Adapter 1:** NAT Network (Simulating an external internet connection)



vpn server - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: NAT Network

Name: NatNetwork

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

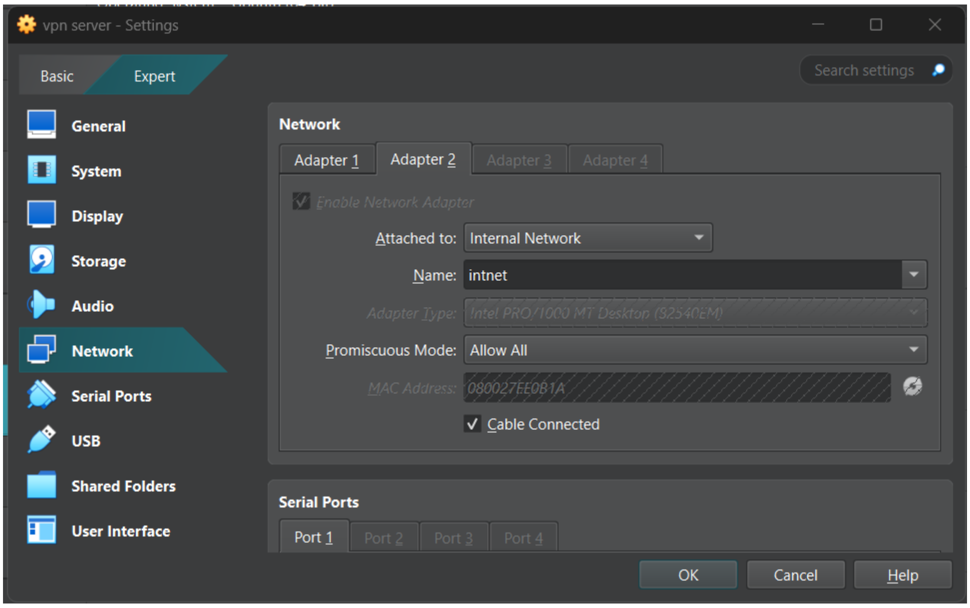
MAC Address: 0800272CC165

☑ Cable Connected

Serial Ports

Port 1 | Port 2 | Port 3 | Port 4

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



Network

☑ Enable Network Adapter

Attached to: Internal Network

Name: intnet

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

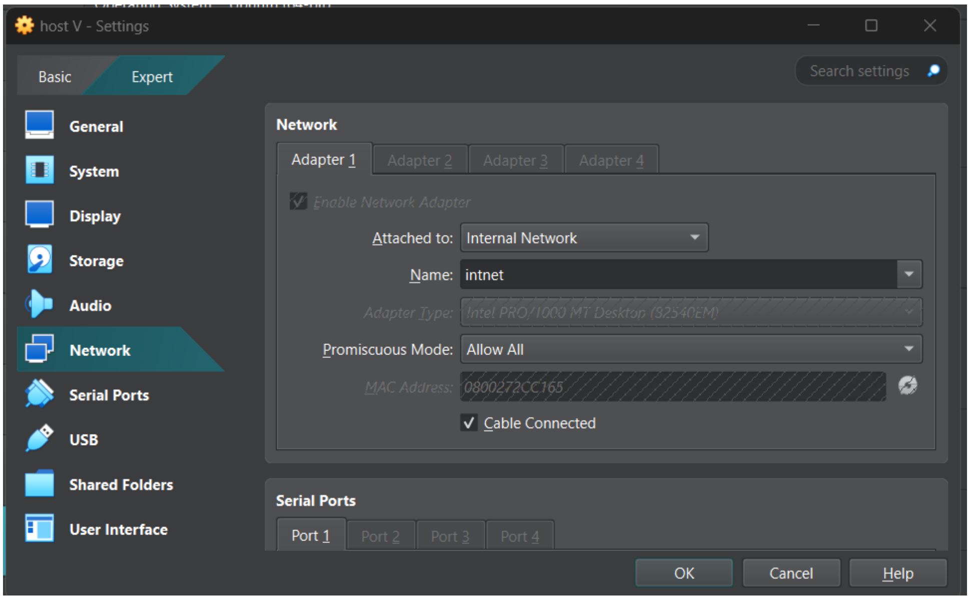
Promiscuous Mode: Allow All

MAC Address: 080027EE0B1A

☑ Cable Connected

Serial Ports

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



Network

☑ Enable Network Adapter

Attached to: Internal Network

Name: intnet

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

MAC Address: 0800272CC165

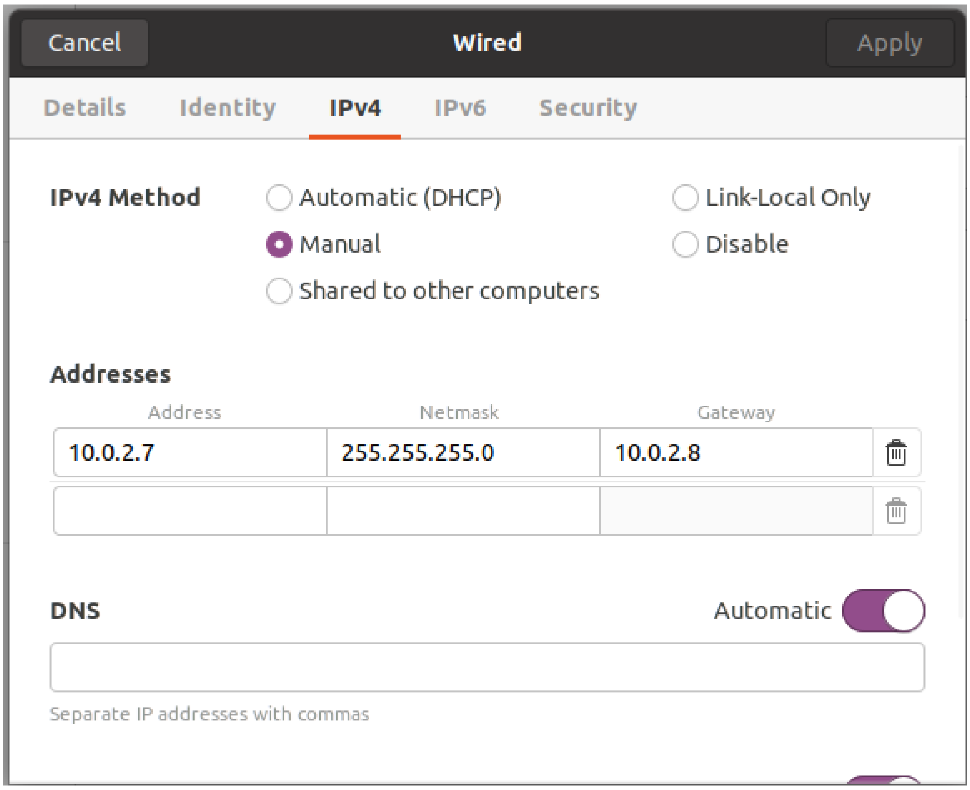
☑ Cable Connected

Serial Ports

Network Configuration:

VPN Client (Host U):

- IP address: 10.0.2.7
- Gateway: 10.0.2.8
- Network Adapter: NAT Network



IPv4 Method

○ Automatic (DHCP) ○ Link-Local Only

● Manual ○ Disable

○ Shared to other computers

Addresses

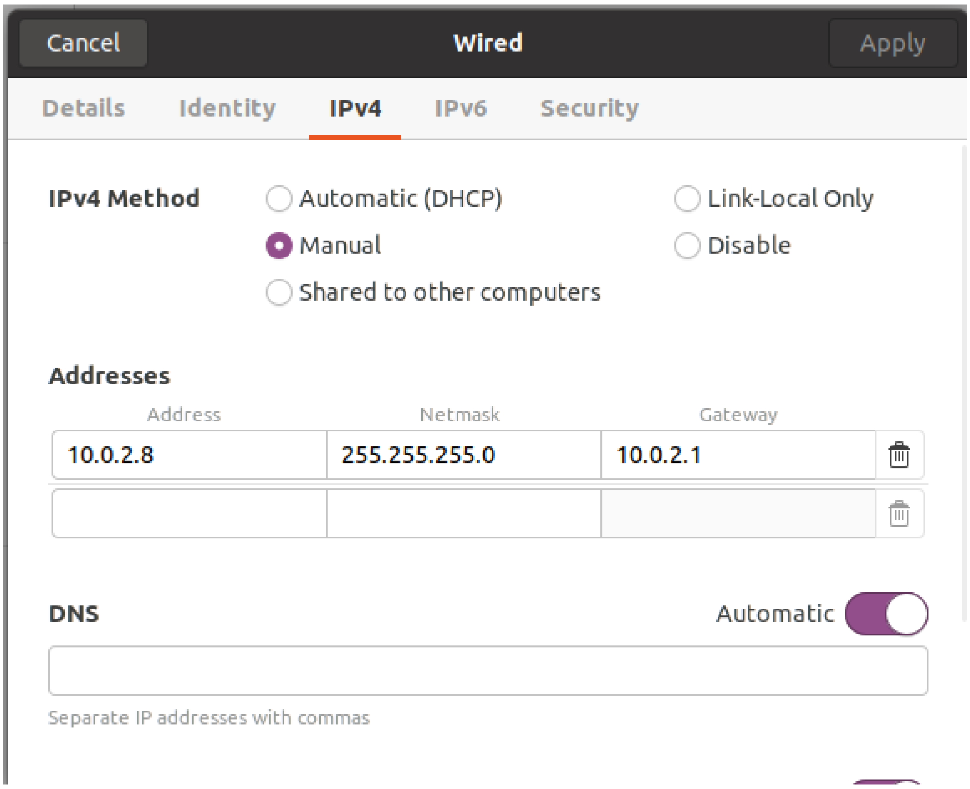
Address	Netmask	Gateway
10.0.2.7	255.255.255.0	10.0.2.8

DNS Automatic ●

Separate IP addresses with commas

VPN Server (Gateway):

- External Network Interface:
  - IP address: 10.0.2.8
  - Gateway: 10.0.2.1
  - Adapter 1: NAT Network



IPv4 Method

○ Automatic (DHCP)
● Manual
○ Shared to other computers
○ Link-Local Only
○ Disable

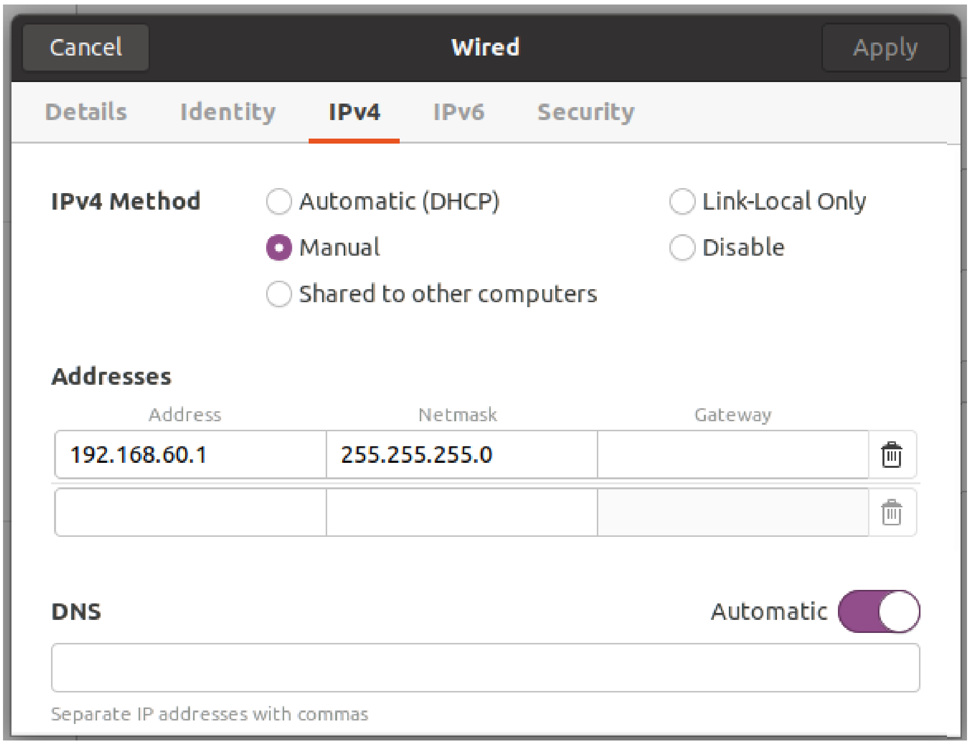
Addresses

Address	Netmask	Gateway
10.0.2.8	255.255.255.0	10.0.2.1

DNS Automatic ●

Separate IP addresses with commas

- Internal Network Interface (Private):
  - IP address: 10.0.2.8
  - Gateway: 10.0.2.1
  - Adapter 2: Internal Network



IPv4 Method

○ Automatic (DHCP)
● Manual
○ Shared to other computers
○ Link-Local Only
○ Disable

Addresses

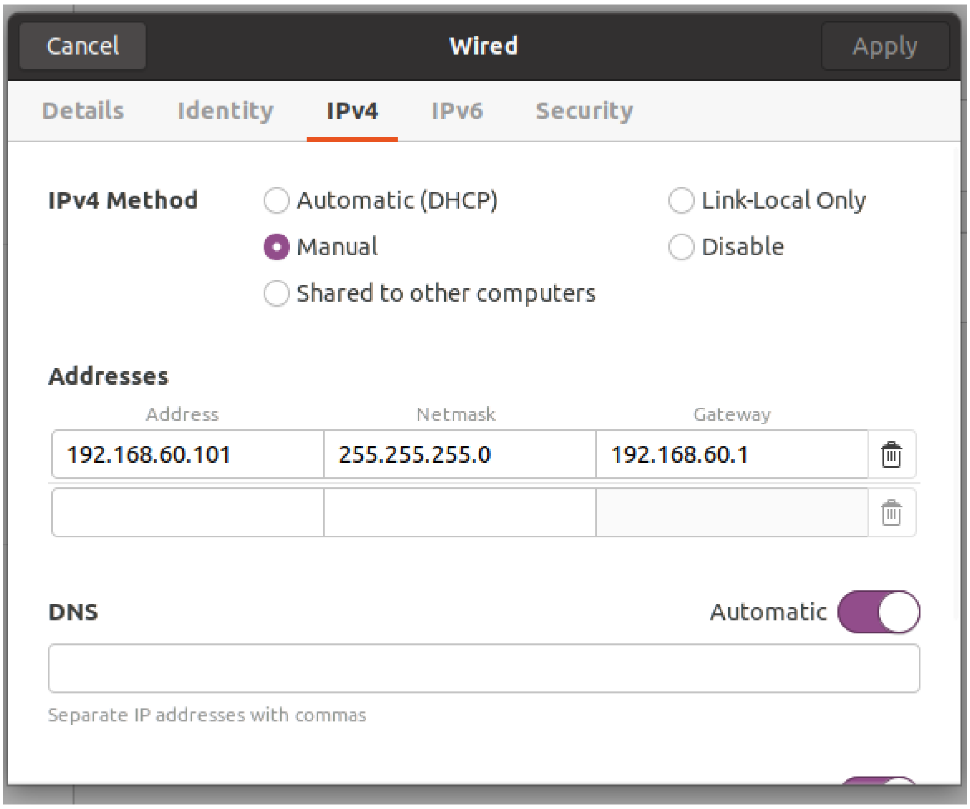
Address	Netmask	Gateway
192.168.60.1	255.255.255.0	

DNS Automatic ●

Separate IP addresses with commas

Host V:

- IP address: 192.168.60.101
- Gateway: 192.168.60.1
- Network Adapter: NAT Network



IPv4 Method

○ Automatic (DHCP) ○ Link-Local Only

● Manual ○ Disable

○ Shared to other computers

Addresses

Address	Netmask	Gateway
192.168.60.101	255.255.255.0	192.168.60.1

DNS Automatic ●

Separate IP addresses with commas

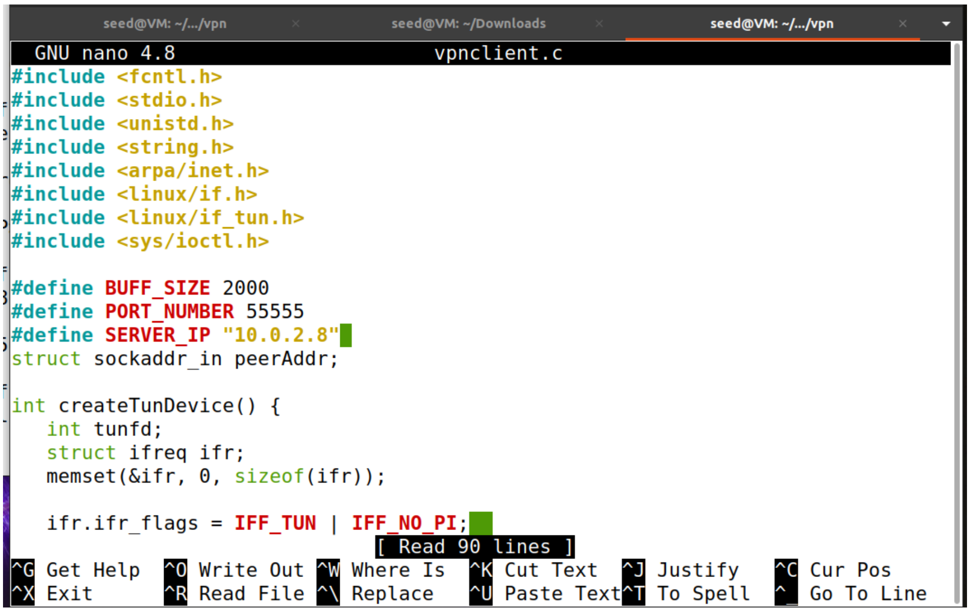
Network Topology:

- The VPN tunnel carries encrypted traffic between client and server
- The server acts as gateway to the protected private network
- Host V is only accessible through the VPN tunnel
- The NAT network simulates the public Internet

Steps to implement VPN

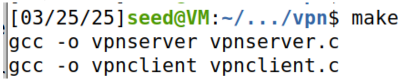
- Install and configure three virtual machines
- Assign network adapters to each virtual machine based on the configuration
- Ensuring that VPN server adapters have been connected correctly
- Manually configure Host V as no DHCP is available in the network

Compile the VPN scripts



```
  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
```

- Then, use the make command to compile both vpnserver.c and vpnclient.c.

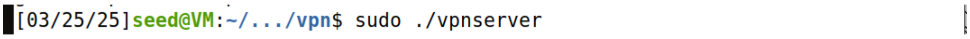


```
[03/25/25]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
```

- Edit the vpnclient.c file to include the server's IP address (NAT network).

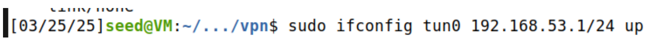
Running the VPN Server

- Start the VPN Server on the VPN Server Machine
- sudo ./vpnserver



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

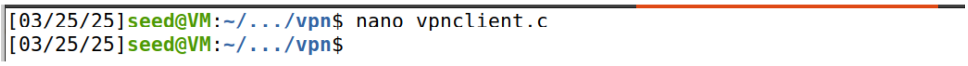
- Configure the TUN Interface on the VPN Server
- sudo ifconfig tun0 192.168.53.1/24 up



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

Running the VPN Client:

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

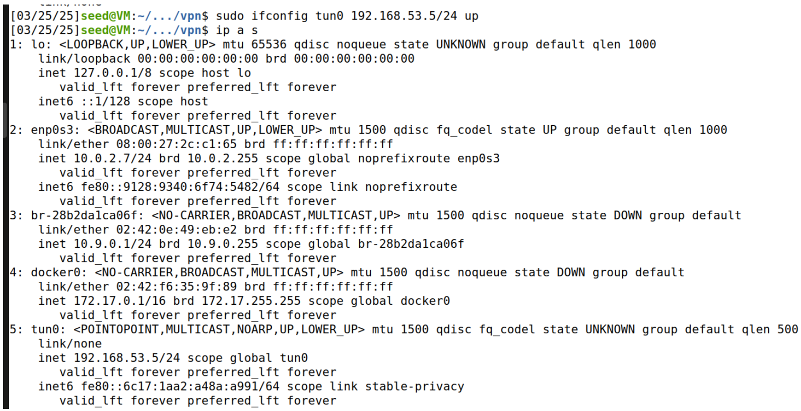


```
[03/25/25]seed@VM:~/.../vpn$ nano vpnclient.c
[03/25/25]seed@VM:~/.../vpn$
```

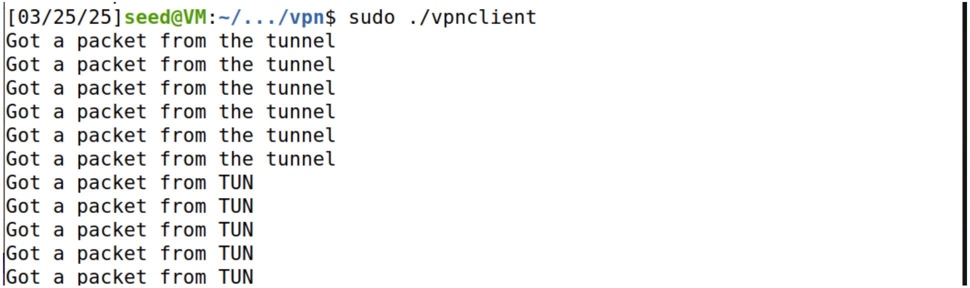
- Set up the TUN interface:
- Once this is complete, the VPN Client should be connected to the VPN Server.

Configuring Routing:

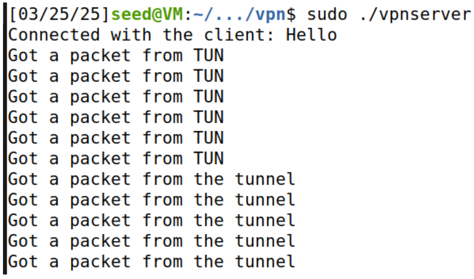
- To ensure proper communication, routing rules must be configured:



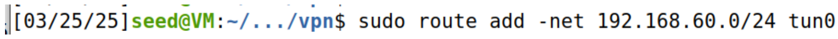
```
[03/25/25]seed@VM:~/.../vpn$ sudo ifconfig tun0 192.168.53.5/24 up
[03/25/25]seed@VM:~/.../vpn$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft 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
```

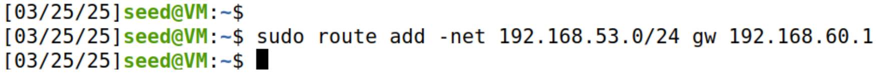


```
[03/25/25]seed@VM:~/.../vpn$ sudo ./vpnserver
Connected with the client: Hello
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
Got a packet from the tunnel
Got a packet from the tunnel
Got a packet from the tunnel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



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

- 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:~$ █
```

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

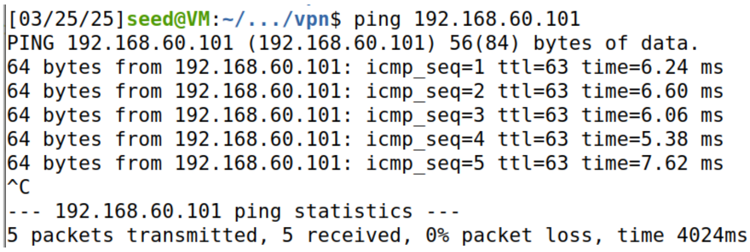
- On the VPN Client, add a route for the private network (192.168.60.0/24) via the VPN tunnel:

Testing the VPN Connection:

- Ping Test: Verify if Host U can communicate with Host V

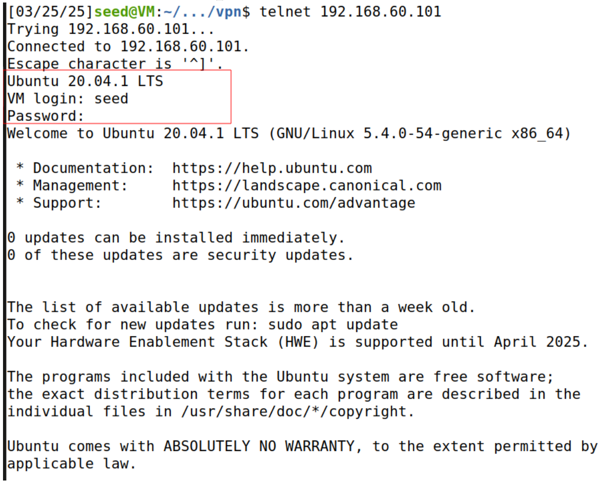
Tunnel Disruption Test

- Establish a telnet connection from Host U to Host V:

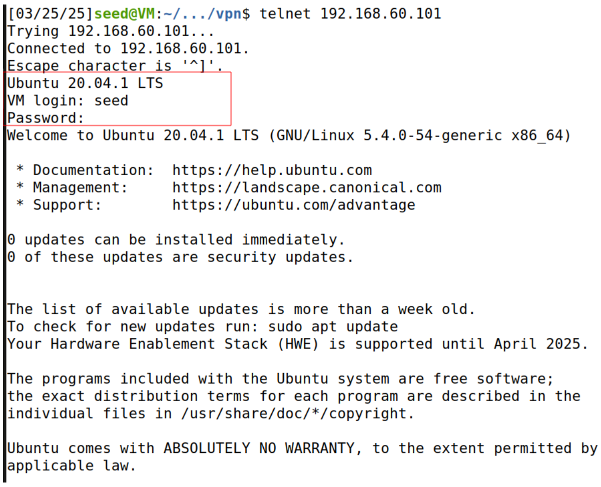


```
[03/25/25]seed@VM:~/.../vpn$ ping 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=63 time=6.24 ms
64 bytes from 192.168.60.101: icmp_seq=2 ttl=63 time=6.60 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=63 time=6.06 ms
64 bytes 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
```

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

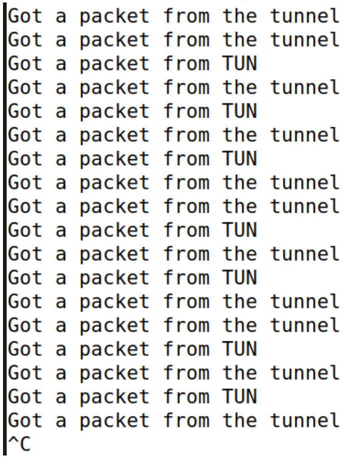


```
[03/25/25]seed@VM:~/.../vpn$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
VM login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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.
```



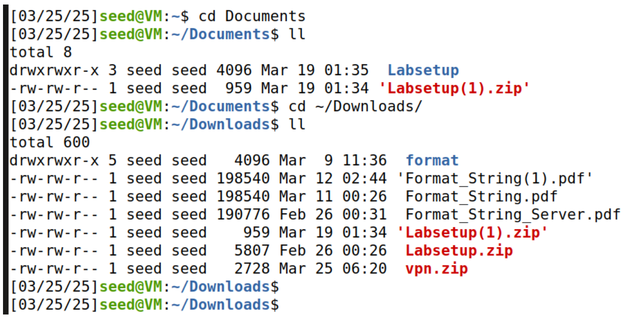
```
[03/25/25]seed@VM:~/.../vpn$ telnet 192.168.60.101
Trying 192.168.60.101...
Connected to 192.168.60.101.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
VM login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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:

We successfully implemented a functional TLS/SSL VPN with:

- Secure encrypted tunnel using TUN/TAP
- Mutual authentication (server certificates and client passwords)
- Support for multiple concurrent clients
- Proper routing configuration for private network access

The implementation demonstrates core VPN concepts including tunneling, encryption, authentication, and routing while providing practical security for network communications.