Host1, NAT

Host2 NAT, internal

Host3, Internal

Task 1:

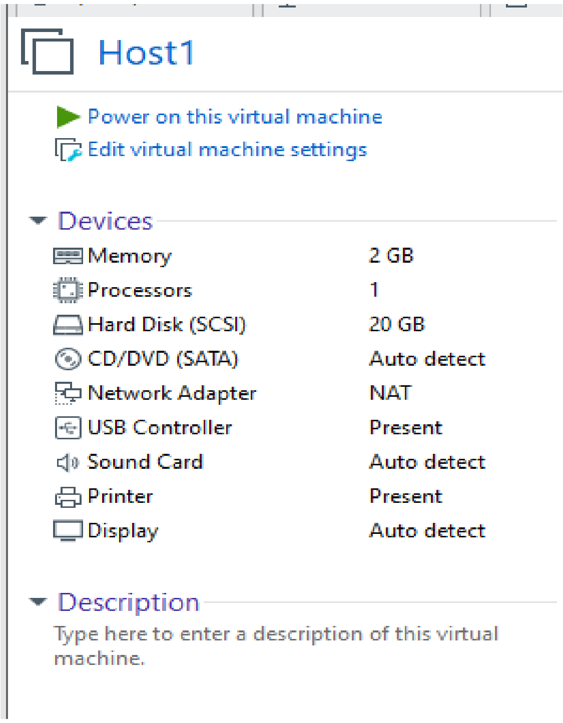
Had to step-up the network configuration for each VM talk to each other in a specific way. Implementing the lab, The VPN client machine will talk to the VPN server using its own virtual interface than route to a private network and find the internal machine.

Created 3 VM's,

Host 1 = VPN Client using NAT network.

Host 2 = VPN Server with 2 Network Adapter, one will NAT with Host 1 and the second one will connect to the internal network.

Host 3 = Internal Network will only talk to the VPN Server.



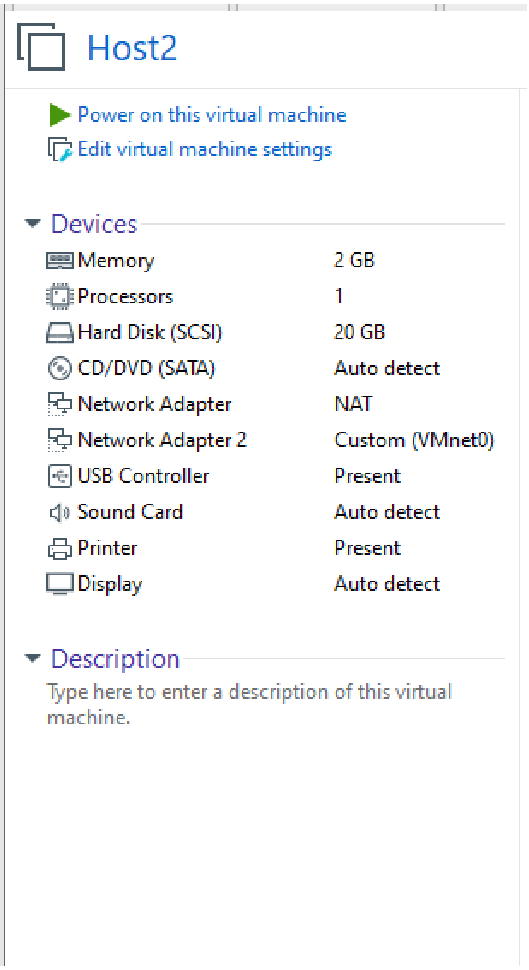
Host1

Devices

Memory	2 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (SATA)	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Description

Type here to enter a description of this virtual machine.



Host2

Power on this virtual machine

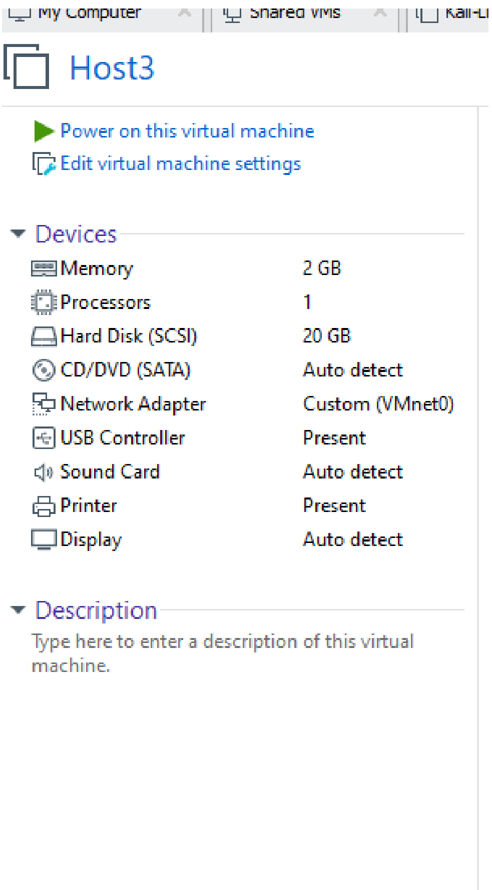
Edit virtual machine settings

Devices

Memory	2 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (SATA)	Auto detect
Network Adapter	NAT
Network Adapter 2	Custom (VMnet0)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Description

Type here to enter a description of this virtual machine.



Host3

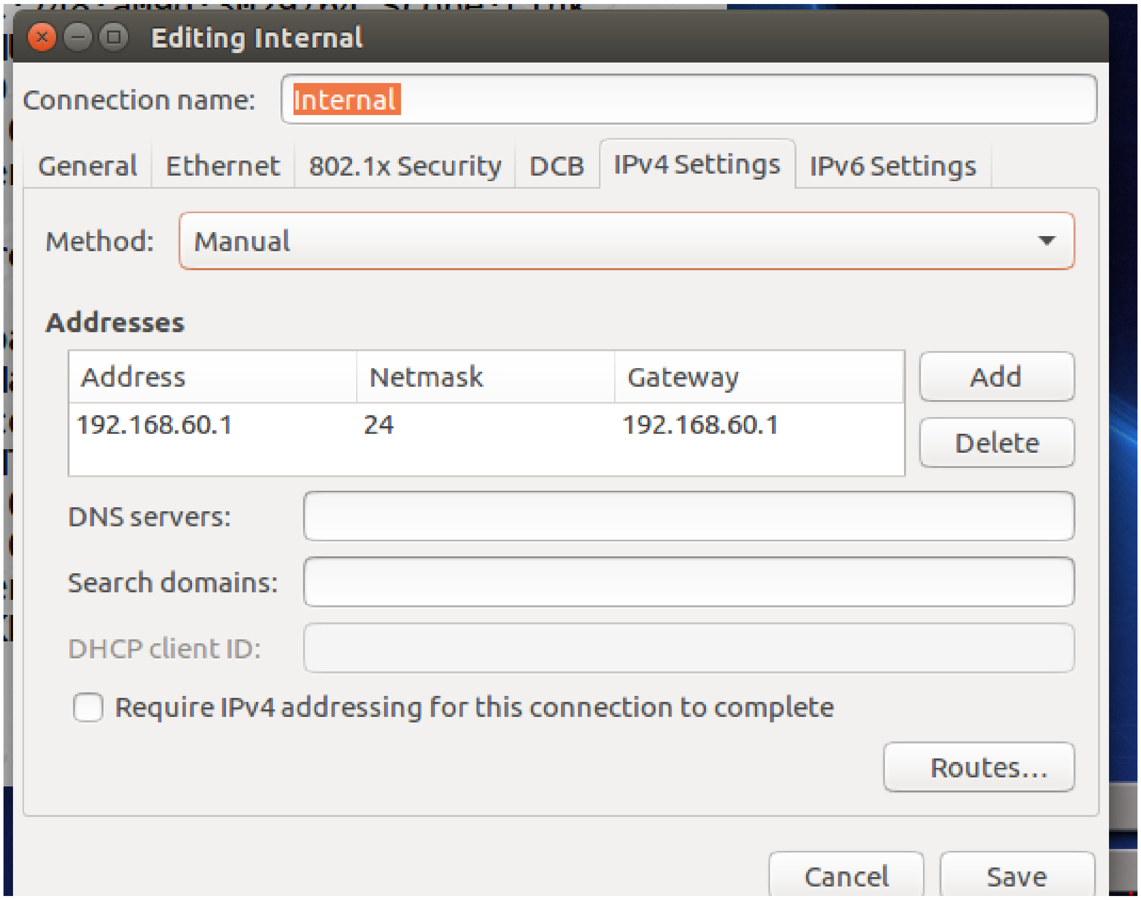
Devices

Memory	2 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (SATA)	Auto detect
Network Adapter	Custom (VMnet0)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Description

Type here to enter a description of this virtual machine.

Setting up an internal network with host 2 and host 3 through network configuration on each machine. Each machine will be connected to an internal network of 192.168.60.x. The VPN server will act as a gateway to the internal machine.



Connection name: Internal

Method: Manual

Addresses

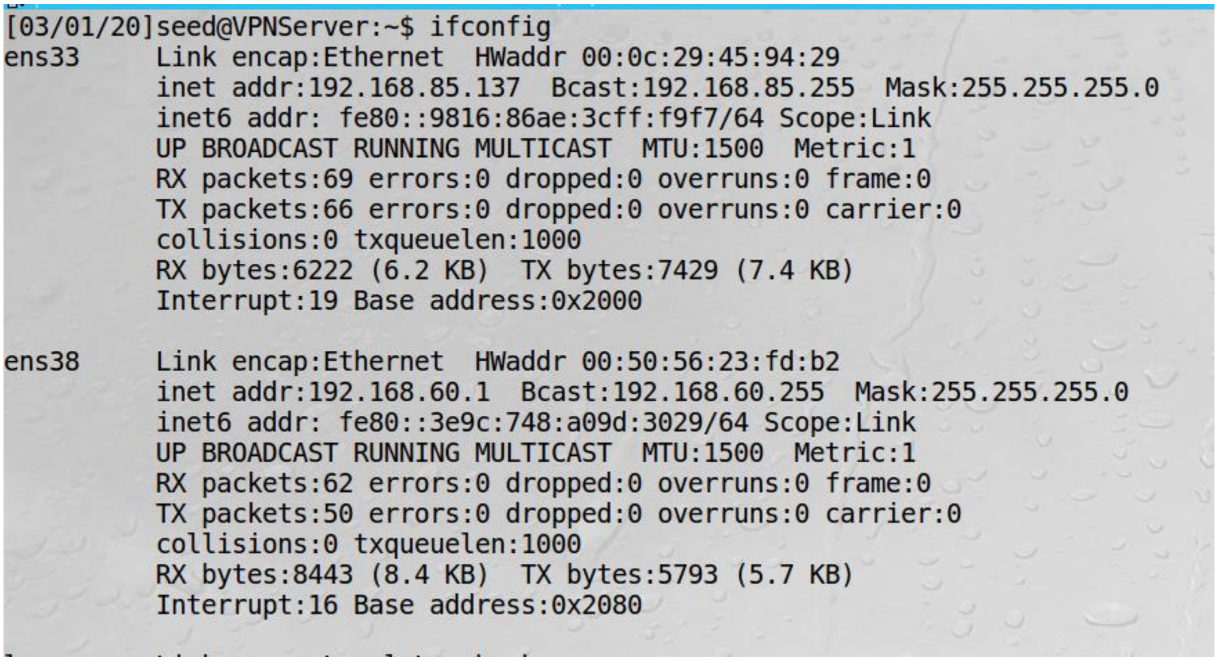
Address	Netmask	Gateway
192.168.60.1	24	192.168.60.1

DNS servers:

Search domains:

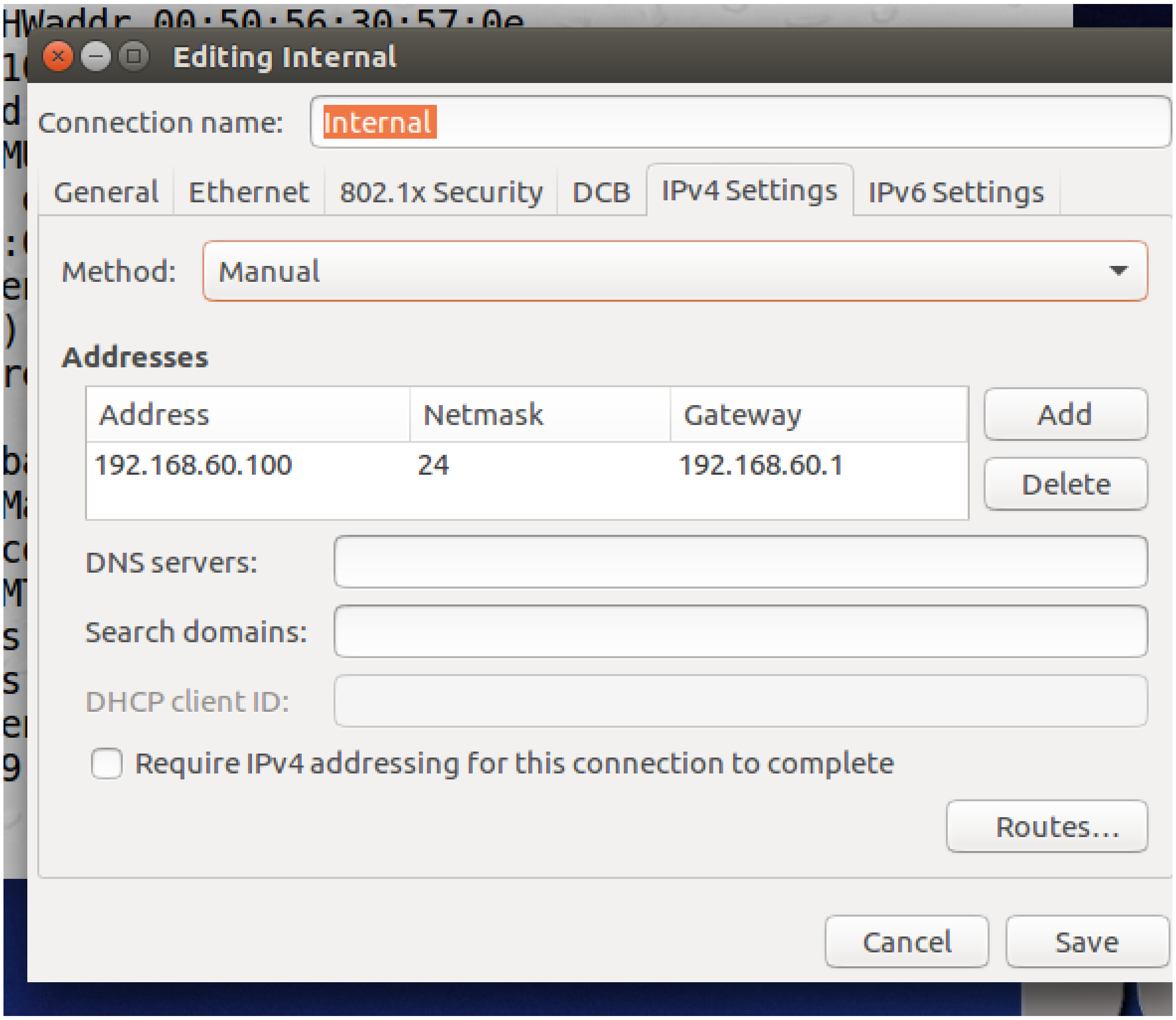
DHCP client ID:

□ Require IPv4 addressing for this connection to complete



```
[03/01/20]seed@VPNServer:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:45:94:29
          inet addr:192.168.85.137  Bcast:192.168.85.255  Mask:255.255.255.0
          inet6 addr: fe80::9816:86ae:3cff:f9f7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:69 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6222 (6.2 KB)  TX bytes:7429 (7.4 KB)
          Interrupt:19 Base address:0x2000

ens38     Link encap:Ethernet  HWaddr 00:50:56:23:fd:b2
          inet addr:192.168.60.1  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::3e9c:748:a09d:3029/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:62 errors:0 dropped:0 overruns:0 frame:0
          TX packets:50 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8443 (8.4 KB)  TX bytes:5793 (5.7 KB)
          Interrupt:16 Base address:0x2080
```

Ens38 is the second adapter that will communicate with the internal network



```
HWaddr 00:50:56:30:57:0e
```

Editing Internal

Connection name: Internal

Method: Manual

Addresses

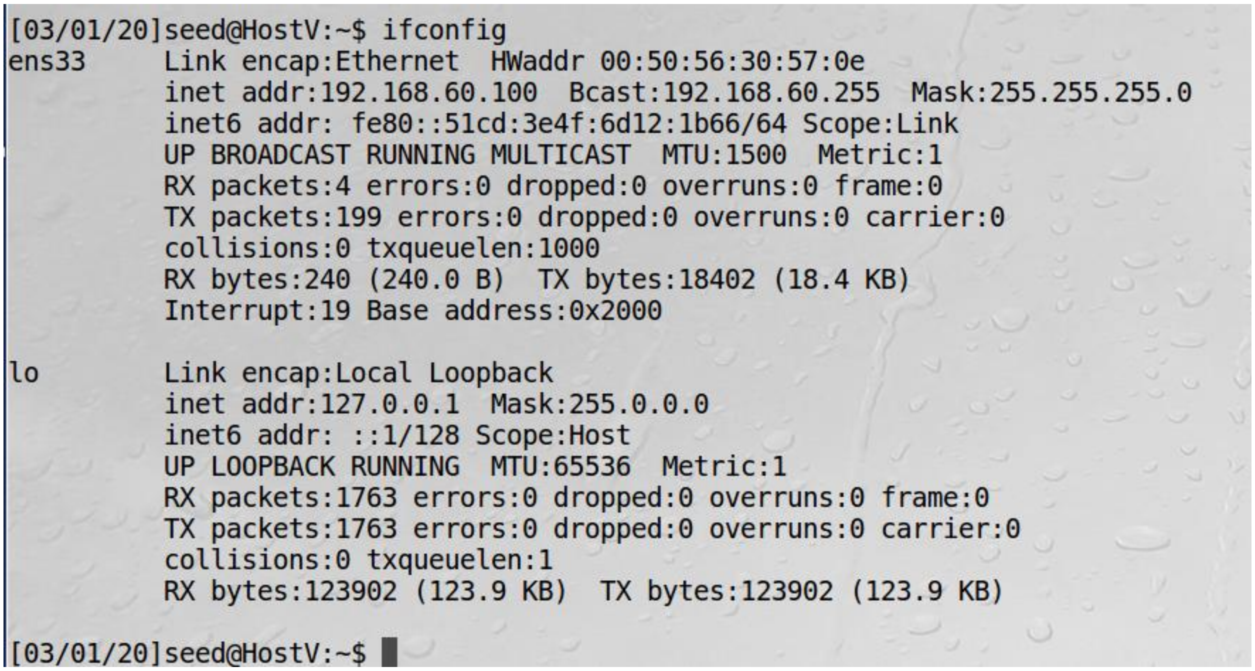
Address	Netmask	Gateway
192.168.60.100	24	192.168.60.1

DNS servers:

Search domains:

DHCP client ID:

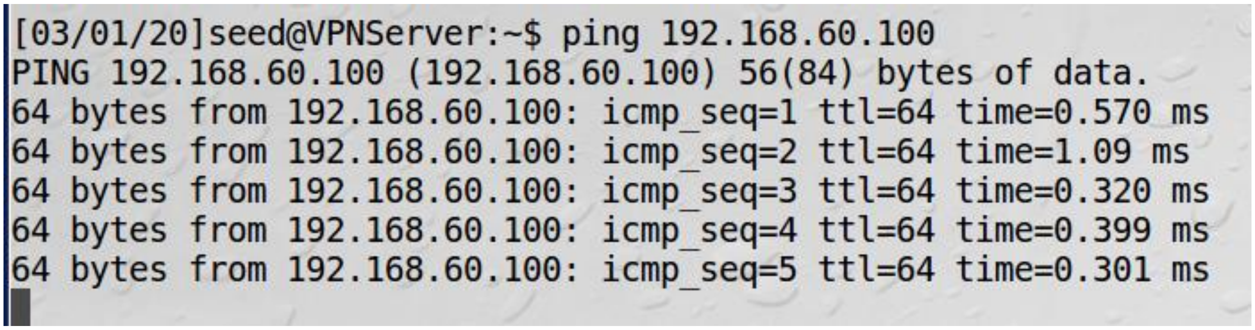
□ Require IPv4 addressing for this connection to complete



```
[03/01/20]seed@HostV:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:50:56:30:57:0e
          inet addr:192.168.60.100  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::51cd:3e4f:6d12:1b66/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:199 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:240 (240.0 B)  TX bytes:18402 (18.4 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1763 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1763 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:123902 (123.9 KB)  TX bytes:123902 (123.9 KB)

[03/01/20]seed@HostV:~$ 
```

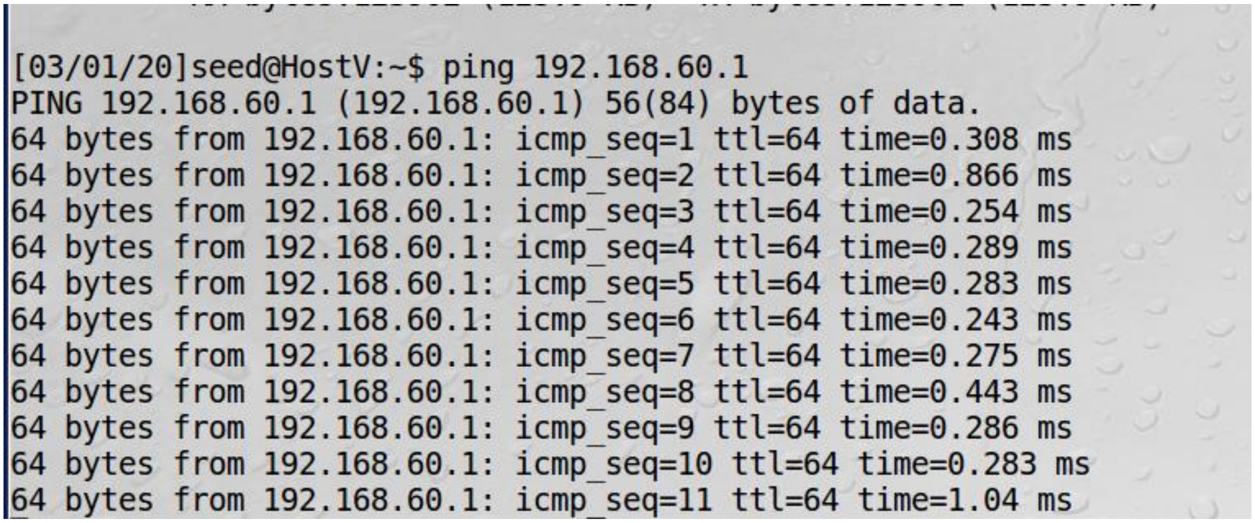
The Internal machine, Host V is on the 192.168.60.x and will talk to 192.168.60.1 on the VPN server.

Check if VPN Server talked to the Client.



```
[03/01/20]seed@VPNServer:~$ ping 192.168.60.100
PING 192.168.60.100 (192.168.60.100) 56(84) bytes of data.
64 bytes from 192.168.60.100: icmp_seq=1 ttl=64 time=0.570 ms
64 bytes from 192.168.60.100: icmp_seq=2 ttl=64 time=1.09 ms
64 bytes from 192.168.60.100: icmp_seq=3 ttl=64 time=0.320 ms
64 bytes from 192.168.60.100: icmp_seq=4 ttl=64 time=0.399 ms
64 bytes from 192.168.60.100: icmp_seq=5 ttl=64 time=0.301 ms
```

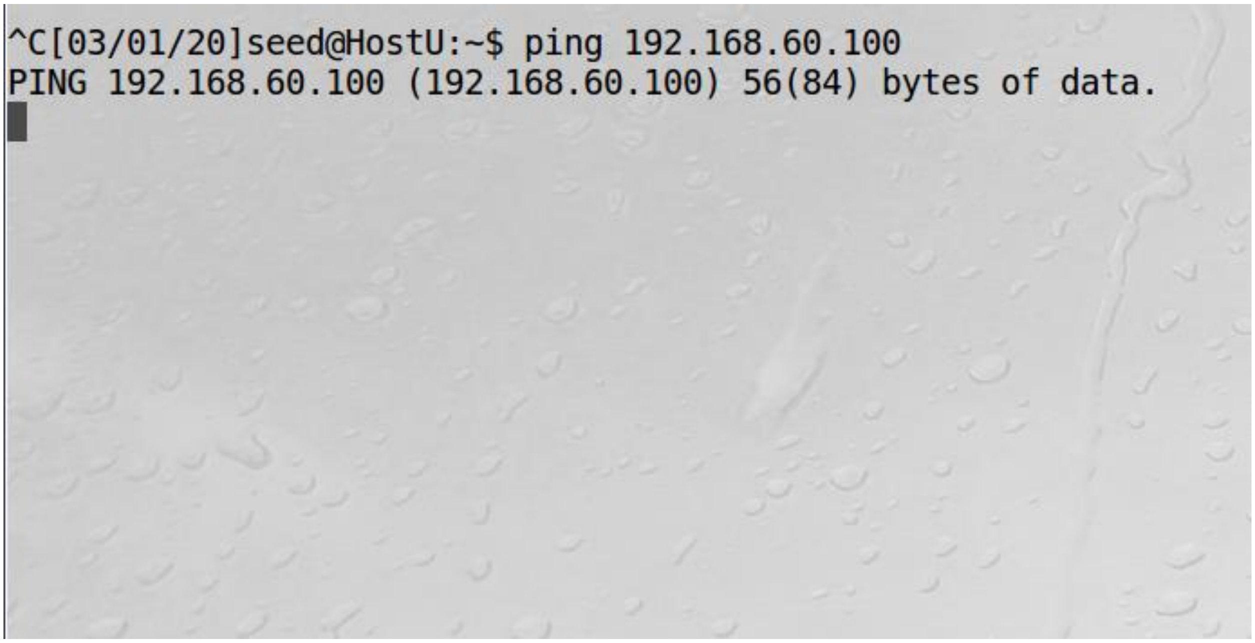
Checking if internal machine can reply back from the VPN Server.



```
[03/01/20]seed@HostV:~$ ping 192.168.60.1
PING 192.168.60.1 (192.168.60.1) 56(84) bytes of data.
64 bytes from 192.168.60.1: icmp_seq=1 ttl=64 time=0.308 ms
64 bytes from 192.168.60.1: icmp_seq=2 ttl=64 time=0.866 ms
64 bytes from 192.168.60.1: icmp_seq=3 ttl=64 time=0.254 ms
64 bytes from 192.168.60.1: icmp_seq=4 ttl=64 time=0.289 ms
64 bytes from 192.168.60.1: icmp_seq=5 ttl=64 time=0.283 ms
64 bytes from 192.168.60.1: icmp_seq=6 ttl=64 time=0.243 ms
64 bytes from 192.168.60.1: icmp_seq=7 ttl=64 time=0.275 ms
64 bytes from 192.168.60.1: icmp_seq=8 ttl=64 time=0.443 ms
64 bytes from 192.168.60.1: icmp_seq=9 ttl=64 time=0.286 ms
64 bytes from 192.168.60.1: icmp_seq=10 ttl=64 time=0.283 ms
64 bytes from 192.168.60.1: icmp_seq=11 ttl=64 time=1.04 ms
```

Looks good

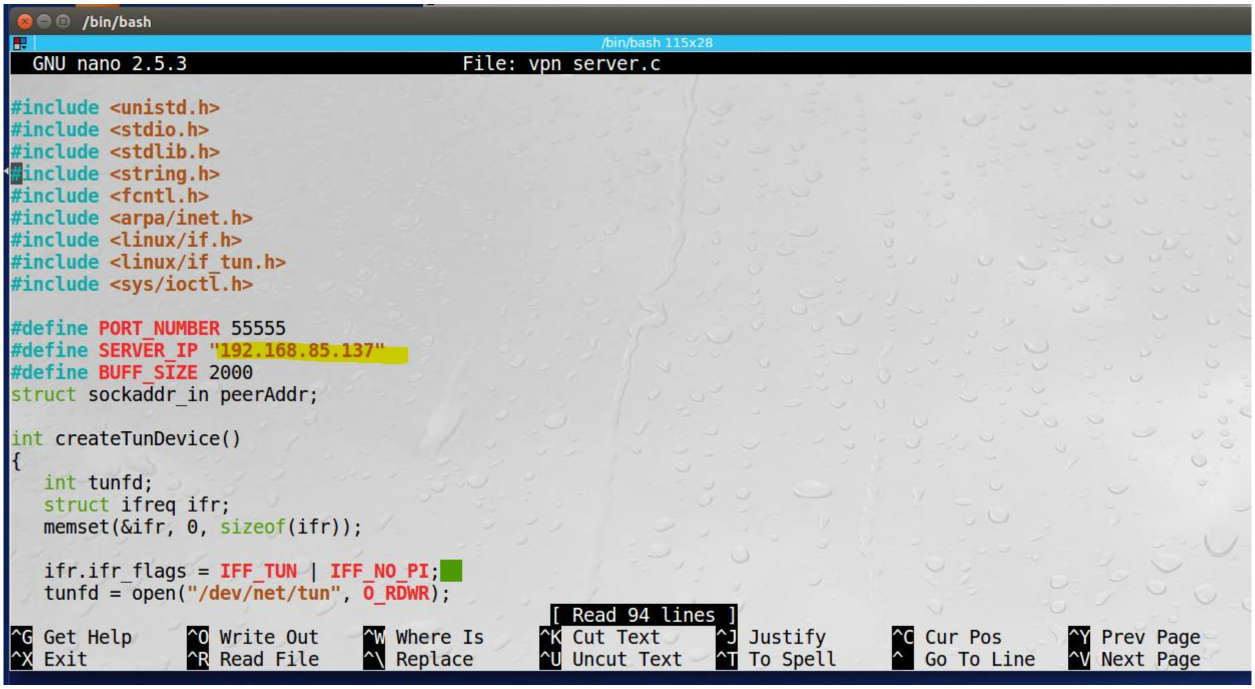
Client should not able to ping to a separate private network.



```
^C[03/01/20]seed@HostU:~$ ping 192.168.60.100
PING 192.168.60.100 (192.168.60.100) 56(84) bytes of data.
```

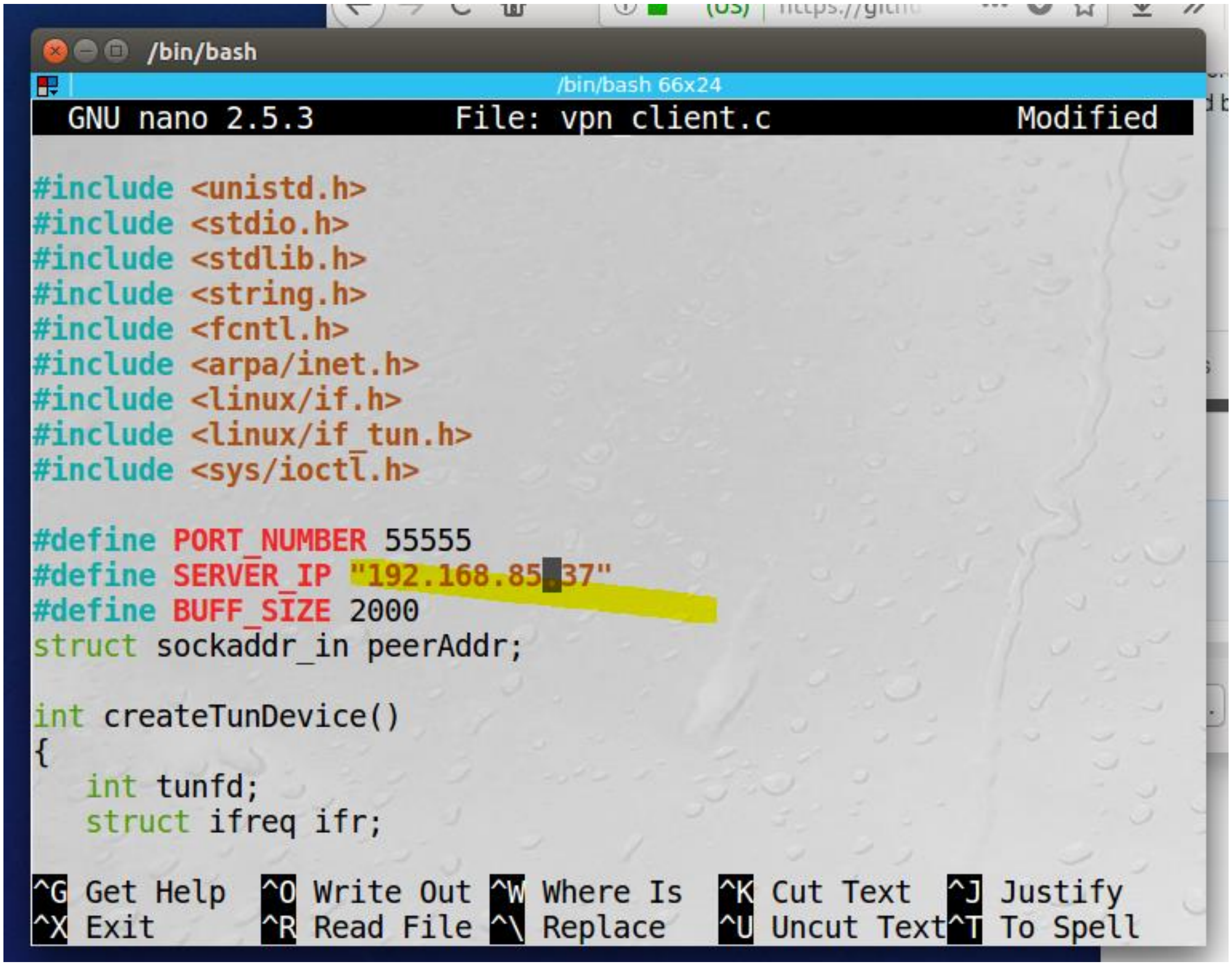
Task 2:

Editing VPN Server script and changing the right value of the IP server which is 192.168.85.137



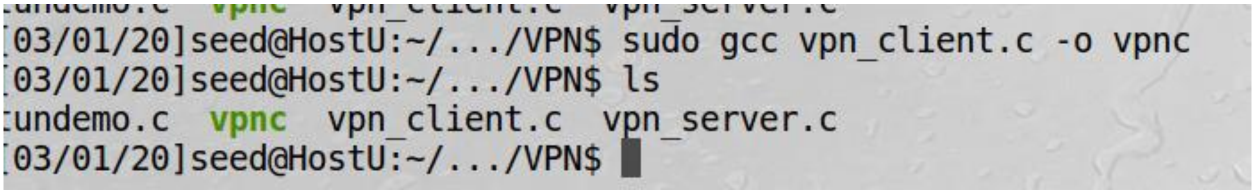
```
  GNU nano 2.5.3                              File: vpn_server.c

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <linux/if.h>
#include <linux/if_tun.h>
#include <sys/ioctl.h>

#define PORT_NUMBER 55555
#define SERVER_IP "192.168.85.137"
#define BUFF_SIZE 2000
struct sockaddr_in peerAddr;

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

   ifr.ifr_flags = IFF_TUN | IFF_NO_PI;
   tunfd = open("/dev/net/tun", O_RDWR);
                                                  [ Read 94 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      ^Y Prev Page
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line   ^V Next Page
```

Same goes with the client machine.

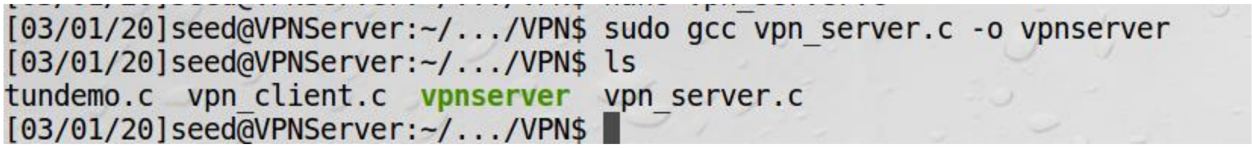


```
  GNU nano 2.5.3           File: vpn_client.c                 Modified

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <linux/if.h>
#include <linux/if_tun.h>
#include <sys/ioctl.h>

#define PORT_NUMBER 55555
#define SERVER_IP "192.168.85.37"
#define BUFF_SIZE 2000
struct sockaddr_in peerAddr;

int createTunDevice()
{
   int tunfd;
   struct ifreq ifr;

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell
```

Had to use the gcc the compile the C language code file.



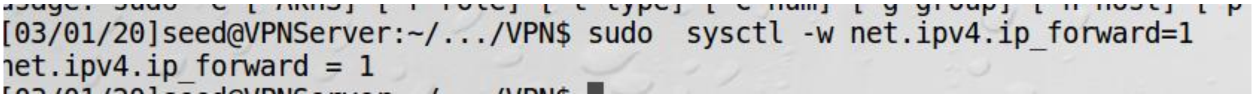
```
[03/01/20]seed@HostU:~/.../VPN$ sudo gcc vpn_client.c -o vpnc
[03/01/20]seed@HostU:~/.../VPN$ ls
tundemo.c  vpnc  vpn_client.c  vpn_server.c
[03/01/20]seed@HostU:~/.../VPN$ 
```



```
[03/01/20]seed@VPNServer:~/.../VPN$ sudo gcc vpn_server.c -o vpnserver
[03/01/20]seed@VPNServer:~/.../VPN$ ls
tundemo.c  vpn_client.c  vpnserver  vpn_server.c
[03/01/20]seed@VPNServer:~/.../VPN$ 
```

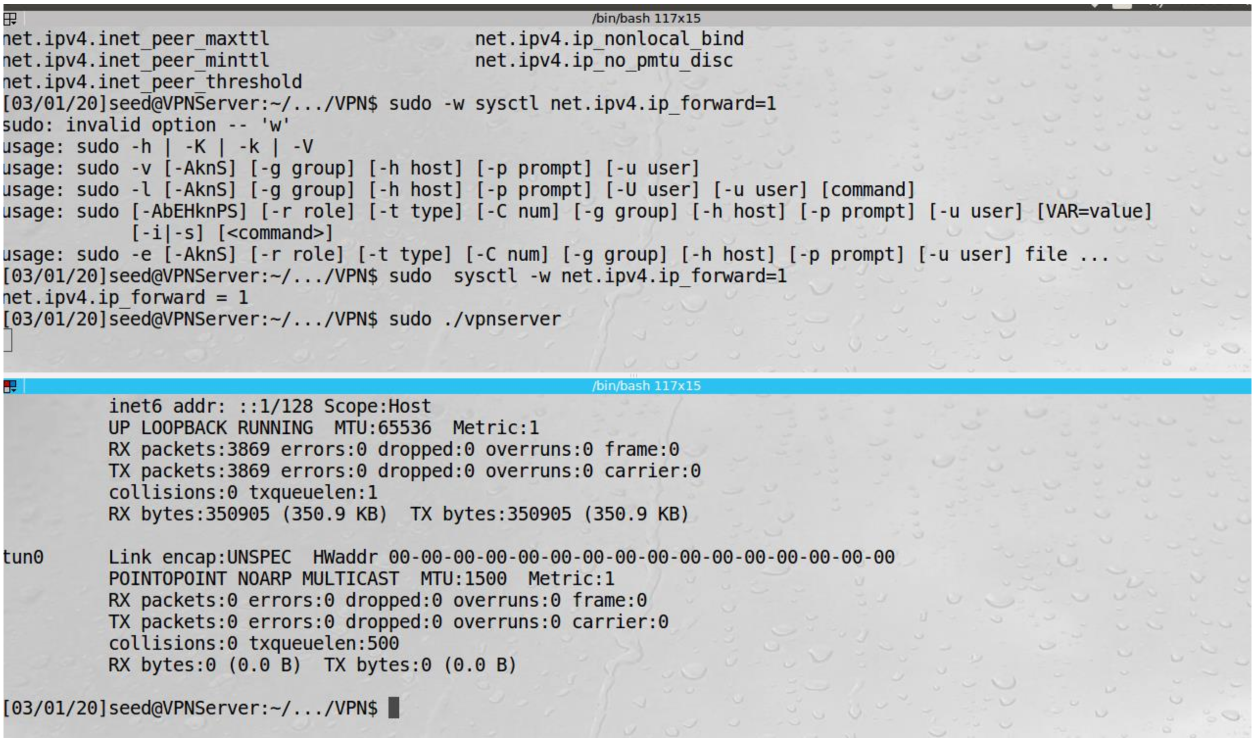
Enable forwarding so packets to the server forward to the application of port 5555

Used the sysctl command to change the ip forward to 1 to enable forwarding.



```
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
[03/01/20]seed@VPNServer:~/.../VPN$ sudo  sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[03/01/20]seed@VPNServer:~/.../VPN$
```

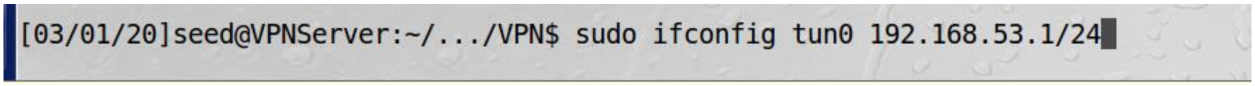
When I ran the VPN Server, it created a virtual interace call tun0 with no ip. Now I had to specify an ip so my Tun0 network can talk to another Tun0 network.



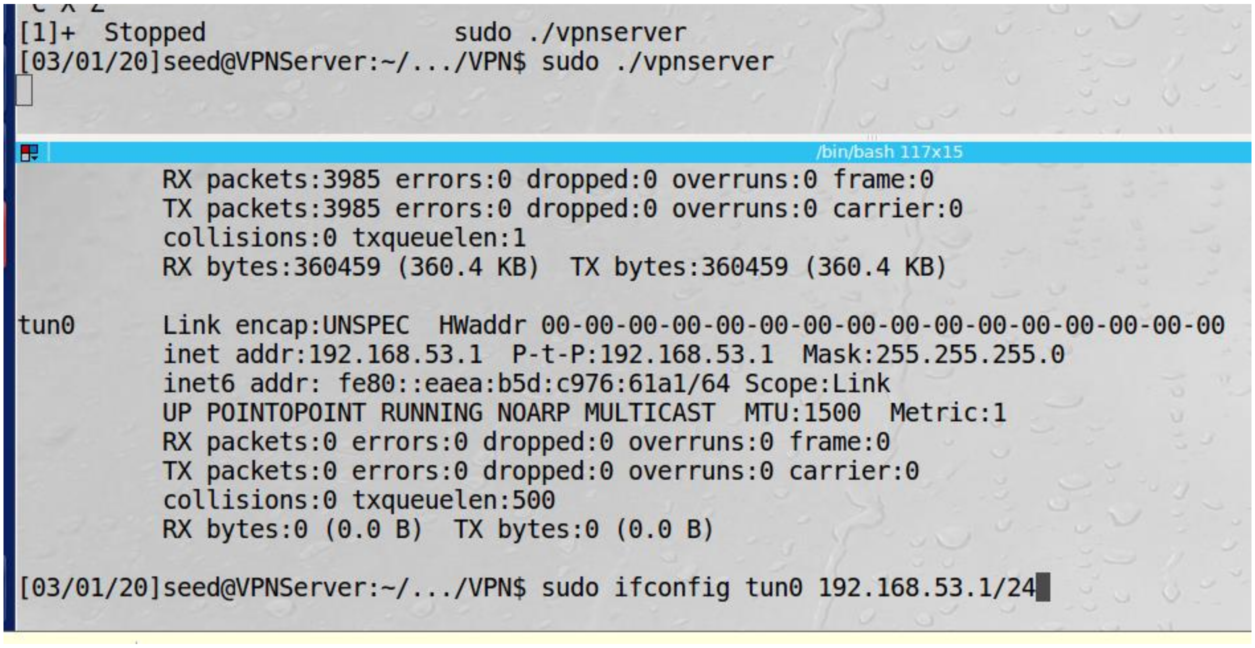
```
net.ipv4.inet_peer_maxttl                  net.ipv4.ip_nonlocal_bind
net.ipv4.inet_peer_minttl                  net.ipv4.ip_no_pmtu_disc
net.ipv4.inet_peer_threshold
[03/01/20]seed@VPNServer:~/.../VPN$ sudo -w sysctl net.ipv4.ip_forward=1
sudo: invalid option -- 'w'
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-u user] [VAR=value]
            [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-u user] file ...
[03/01/20]seed@VPNServer:~/.../VPN$ sudo  sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[03/01/20]seed@VPNServer:~/.../VPN$ sudo ./vpnserver
```

```
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:3869 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3869 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:350905 (350.9 KB)  TX bytes:350905 (350.9 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          POINTOPOINT NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

[03/01/20]seed@VPNServer:~/.../VPN$
```

Used ifconfig to create a network for tun0



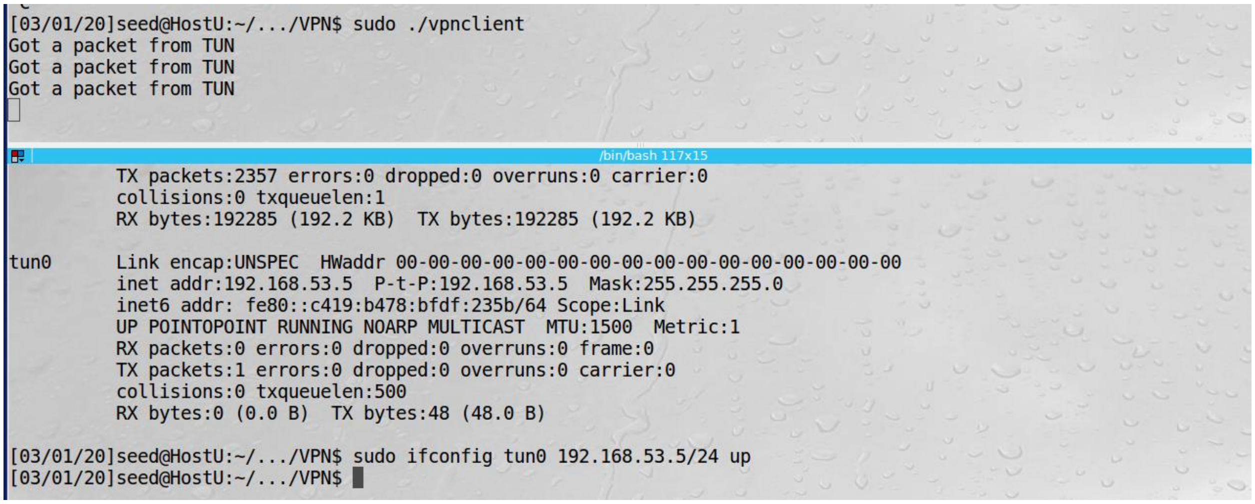
```
[03/01/20]seed@VPNServer:~/.../VPN$ sudo ifconfig tun0 192.168.53.1/24
```



```
[1]+  Stopped                 sudo ./vpnserver
[03/01/20]seed@VPNServer:~/.../VPN$ sudo ./vpnserver
```

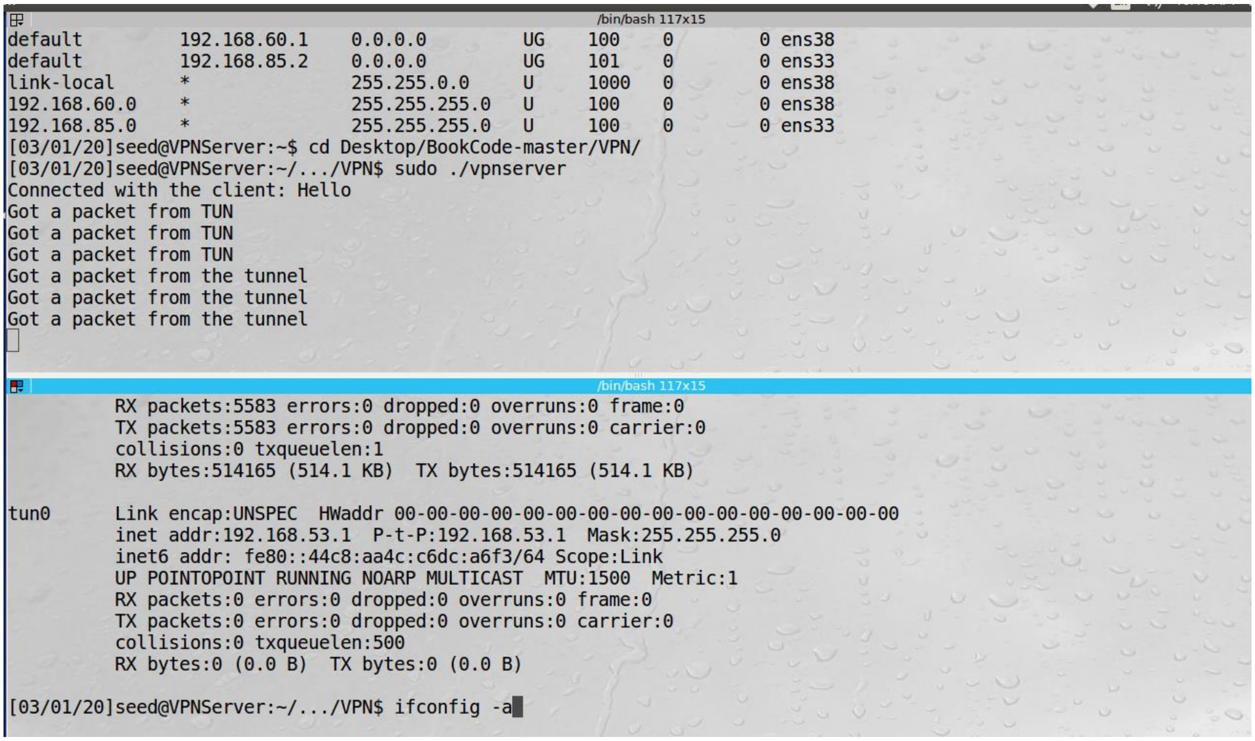
```
          RX packets:3985 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3985 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:360459 (360.4 KB)  TX bytes:360459 (360.4 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.53.1  P-t-P:192.168.53.1  Mask:255.255.255.0
          inet6 addr: fe80::eaea:b5d:c976:61a1/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

[03/01/20]seed@VPNServer:~/.../VPN$ sudo ifconfig tun0 192.168.53.1/24
```

Ran VPNclient script and it also created tun0 network which I also gave an IP that will only talk to the tun0 within the VPN server.



```
[03/01/20]seed@HostU:~/.../VPN$ sudo ./vpnclient
Got a packet from TUN
Got a packet from TUN
Got a packet from TUN
```

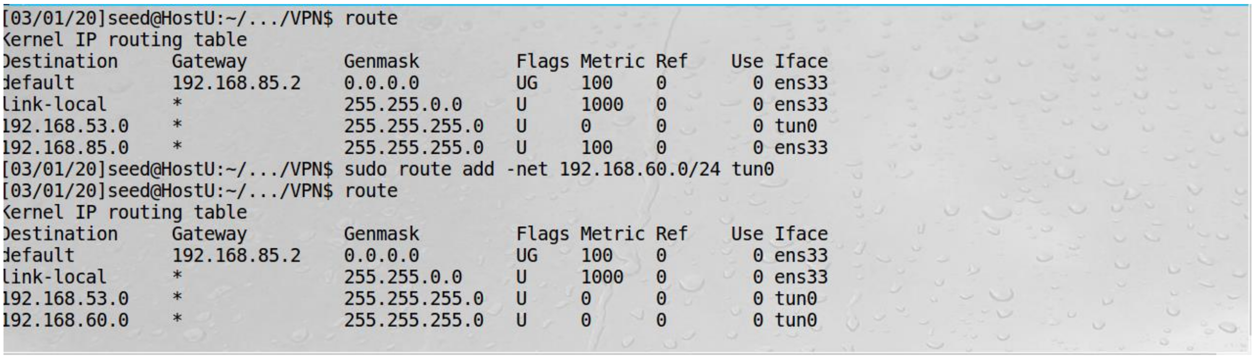
```
          TX packets:2357 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:192285 (192.2 KB)  TX bytes:192285 (192.2 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.53.5  P-t-P:192.168.53.5  Mask:255.255.255.0
          inet6 addr: fe80::c419:b478:bfdf:235b/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:48 (48.0 B)

[03/01/20]seed@HostU:~/.../VPN$ sudo ifconfig tun0 192.168.53.5/24 up
[03/01/20]seed@HostU:~/.../VPN$ 
```



```
default         192.168.60.1    0.0.0.0         UG    100    0        0 ens38
default         192.168.85.2    0.0.0.0         UG    101    0        0 ens33
link-local      *               255.255.0.0     U     1000   0        0 ens38
192.168.60.0    *               255.255.255.0   U     100    0        0 ens38
192.168.85.0    *               255.255.255.0   U     100    0        0 ens33
[03/01/20]seed@VPNServer:~$ cd Desktop/BookCode-master/VPN/
[03/01/20]seed@VPNServer:~/.../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 the tunnel
Got a packet from the tunnel
Got a packet from the tunnel
```

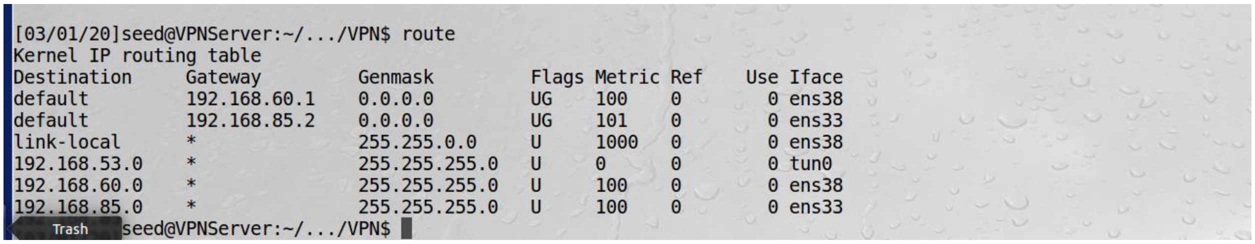
```
          RX packets:5583 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5583 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:514165 (514.1 KB)  TX bytes:514165 (514.1 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.53.1  P-t-P:192.168.53.1  Mask:255.255.255.0
          inet6 addr: fe80::44c8:aa4c:c6dc:a6f3/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

[03/01/20]seed@VPNServer:~/.../VPN$ ifconfig -a
```

Added a route within the VPN client to know if there is a network of a 192.168.60.x network pass it through tun0



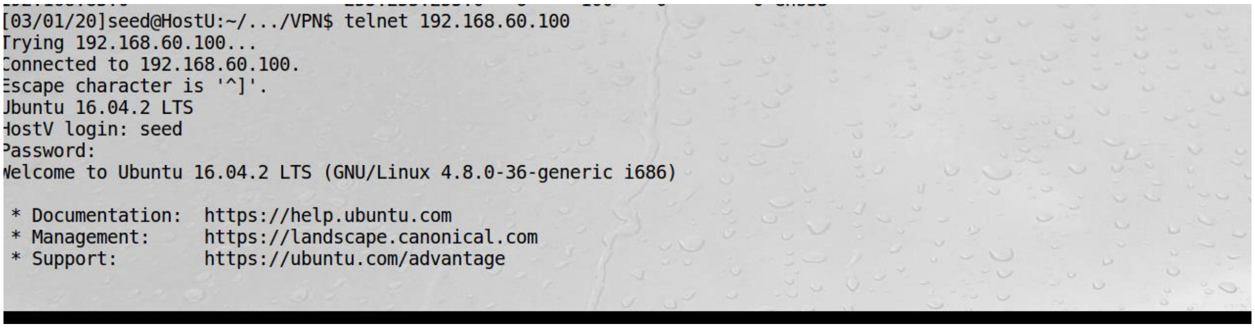
```
[03/01/20]seed@HostU:~/.../VPN$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.85.2    0.0.0.0         UG    100    0        0 ens33
link-local      *               255.255.0.0     U     1000   0        0 ens33
192.168.53.0    *               255.255.255.0   U     0      0        0 tun0
192.168.85.0    *               255.255.255.0   U     100    0        0 ens33
[03/01/20]seed@HostU:~/.../VPN$ sudo route add -net 192.168.60.0/24 tun0
[03/01/20]seed@HostU:~/.../VPN$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.85.2    0.0.0.0         UG    100    0        0 ens33
link-local      *               255.255.0.0     U     1000   0        0 ens33
192.168.53.0    *               255.255.255.0   U     0      0        0 tun0
192.168.60.0    *               255.255.255.0   U     0      0        0 tun0
```

Route looks good so far because I know on my service side if they receive a packet coming from 192.168.60.x then it push through the prive machine , Host V.

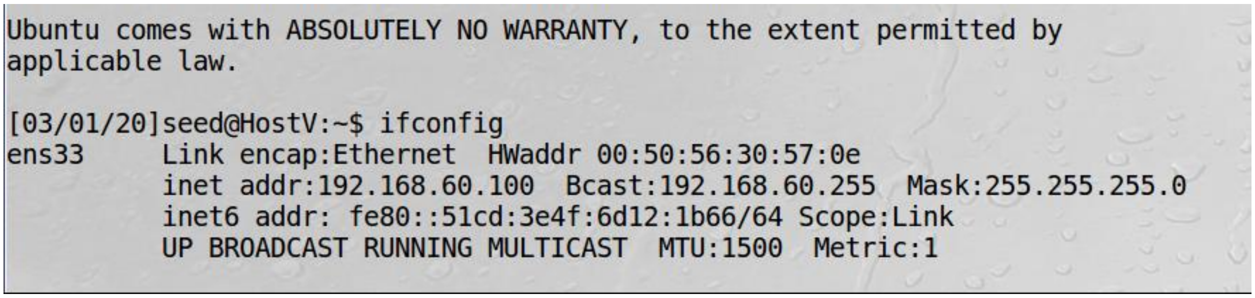


```
[03/01/20]seed@VPNServer:~/.../VPN$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.60.1    0.0.0.0         UG    100    0        0 ens38
default         192.168.85.2    0.0.0.0         UG    101    0        0 ens33
link-local      *               255.255.0.0     U     1000   0        0 ens38
192.168.53.0    *               255.255.255.0   U     0      0        0 tun0
192.168.60.0    *               255.255.255.0   U     100    0        0 ens38
192.168.85.0    *               255.255.255.0   U     100    0        0 ens33
          seed@VPNServer:~/.../VPN$
```

I ran a telnet command in my VPN client and was able to talk to the internal network that only talk to the vpn server because the traffic pass through the tun0 network and routed to the private gateway.

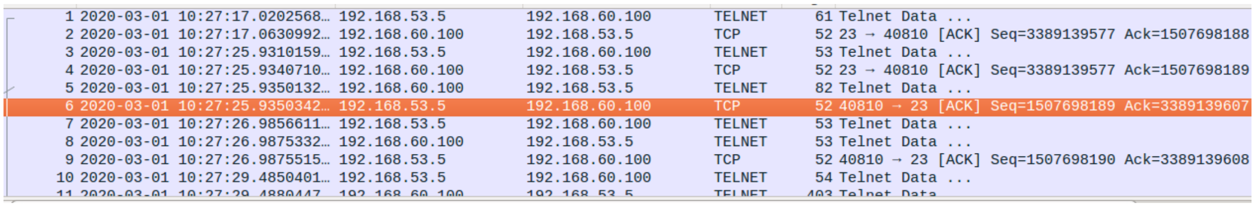


```
[03/01/20]seed@HostU:~/.../VPN$ telnet 192.168.60.100
Trying 192.168.60.100...
Connected to 192.168.60.100.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
HostV login: seed
Password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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



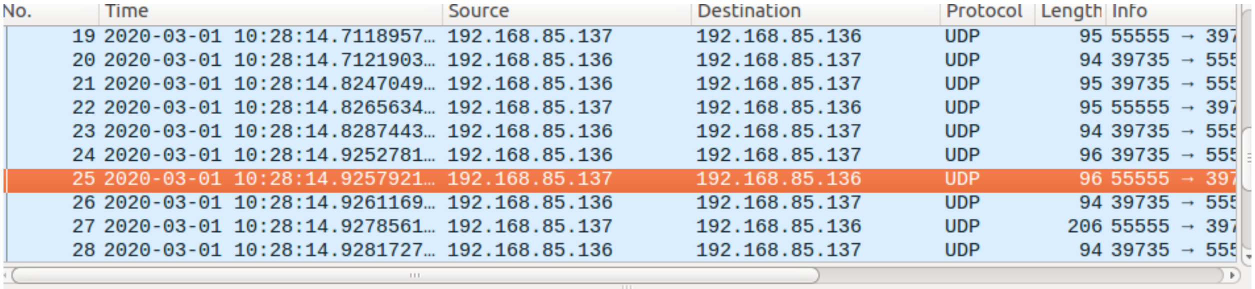
```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

[03/01/20]seed@HostV:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:50:56:30:57:0e
          inet addr:192.168.60.100  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::51cd:3e4f:6d12:1b66/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Here you can see the Tun0 interface talked through the internal network.



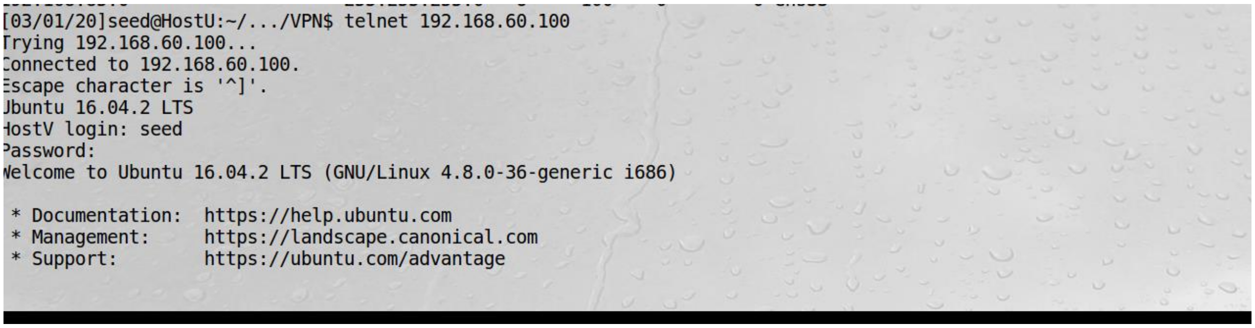
```
 1 2020-03-01 10:27:17.0202568… 192.168.53.5    192.168.60.100  TELNET  61 Telnet Data ...
 2 2020-03-01 10:27:17.0630992… 192.168.60.100  192.168.53.5    TCP     52 23 → 40810 [ACK] Seq=3389139577 Ack=1507698188
 3 2020-03-01 10:27:25.9310159… 192.168.53.5    192.168.60.100  TELNET  53 Telnet Data ...
 4 2020-03-01 10:27:25.9340710… 192.168.60.100  192.168.53.5    TCP     52 23 → 40810 [ACK] Seq=3389139577 Ack=1507698189
 5 2020-03-01 10:27:25.9350132… 192.168.60.100  192.168.53.5    TELNET  82 Telnet Data ...
 6 2020-03-01 10:27:25.9350342… 192.168.53.5    192.168.60.100  TCP     52 40810 → 23 [ACK] Seq=1507698189 Ack=3389139607
 7 2020-03-01 10:27:26.9856611… 192.168.53.5    192.168.60.100  TELNET  53 Telnet Data ...
 8 2020-03-01 10:27:26.9875332… 192.168.60.100  192.168.53.5    TELNET  53 Telnet Data ...
 9 2020-03-01 10:27:26.9875515… 192.168.53.5    192.168.60.100  TCP     52 40810 → 23 [ACK] Seq=1507698190 Ack=3389139608
10 2020-03-01 10:27:29.4850401… 192.168.53.5    192.168.60.100  TELNET  54 Telnet Data ...
11 2020-03-01 10:27:29.4880447  192.168.60.100  192.168.53.5    TELNET 403 Telnet Data
```

What the server saw that 192.168.85.137 send a packet through 102.168.85.136 because we are seeing the traffic within different interface.

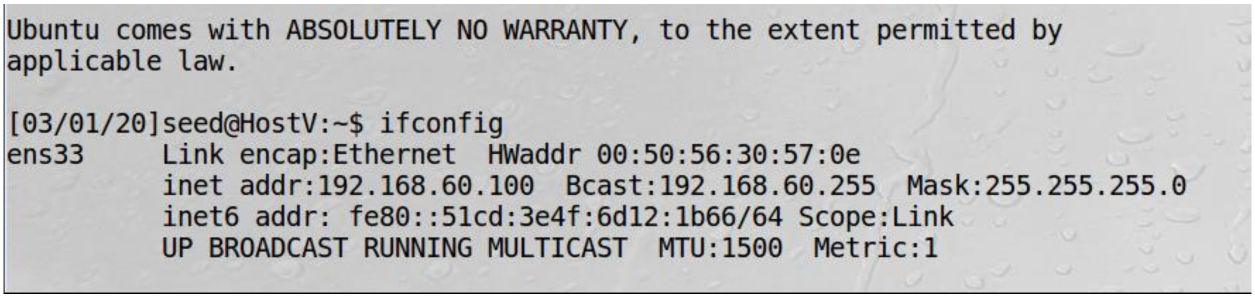


```
No.  Time                           Source          Destination     Protocol  Length Info
  19 2020-03-01 10:28:14.7118957…  192.168.85.137  192.168.85.136  UDP           95 55555 → 397
  20 2020-03-01 10:28:14.7121903…  192.168.85.136  192.168.85.137  UDP           94 39735 → 555
  21 2020-03-01 10:28:14.8247049…  192.168.85.136  192.168.85.137  UDP           95 39735 → 555
  22 2020-03-01 10:28:14.8265634…  192.168.85.137  192.168.85.136  UDP           95 55555 → 397
  23 2020-03-01 10:28:14.8287443…  192.168.85.136  192.168.85.137  UDP           94 39735 → 555
  24 2020-03-01 10:28:14.9252781…  192.168.85.136  192.168.85.137  UDP           96 39735 → 555
  25 2020-03-01 10:28:14.9257921…  192.168.85.137  192.168.85.136  UDP           96 55555 → 397
  26 2020-03-01 10:28:14.9261169…  192.168.85.136  192.168.85.137  UDP           94 39735 → 555
  27 2020-03-01 10:28:14.9278561…  192.168.85.137  192.168.85.136  UDP          206 55555 → 397
  28 2020-03-01 10:28:14.9281727…  192.168.85.136  192.168.85.137  UDP           94 39735 → 555
```

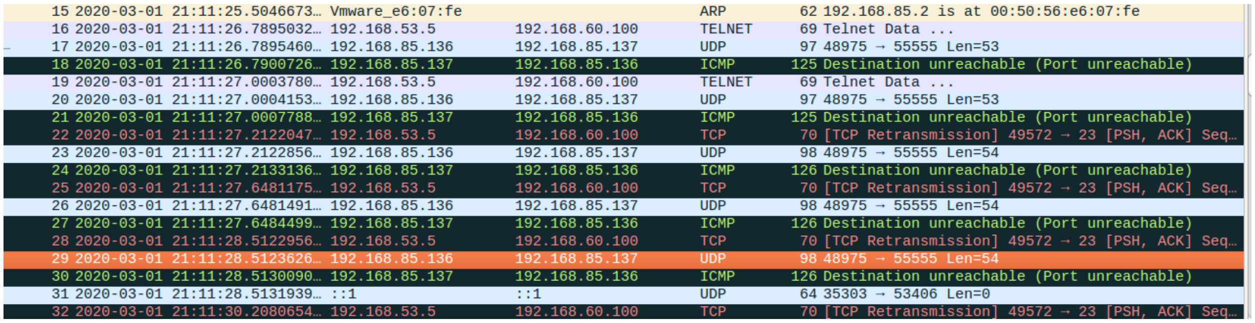
As I disconnected the VPN, the session with the terminal froze but then was functional when I reconnected the VPN.



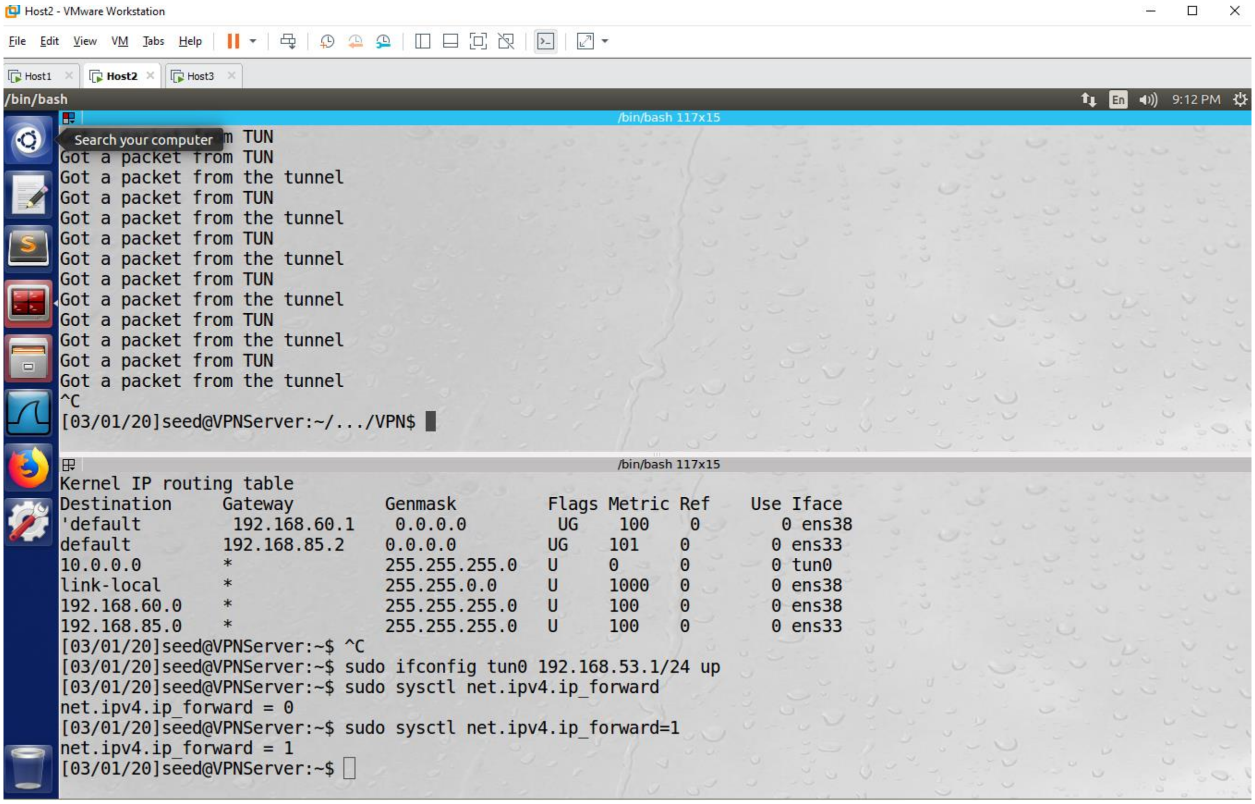
```
[03/01/20]seed@HostU:~/.../VPN$ telnet 192.168.60.100
Trying 192.168.60.100...
Connected to 192.168.60.100.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
HostV login: seed
Password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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



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

[03/01/20]seed@HostV:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:50:56:30:57:0e
          inet addr:192.168.60.100  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::51cd:3e4f:6d12:1b66/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```



```
15 2020-03-01 21:11:25.5046673… Vmware_e6:07:fe                    ARP       62 192.168.85.2 is at 00:50:56:e6:07:fe
16 2020-03-01 21:11:26.7895032… 192.168.53.5     192.168.60.100    TELNET    69 Telnet Data ...
17 2020-03-01 21:11:26.7895460… 192.168.85.136   192.168.85.137    UDP       97 48975 → 55555 Len=53
18 2020-03-01 21:11:26.7900726… 192.168.85.137   192.168.85.136    ICMP     125 Destination unreachable (Port unreachable)
19 2020-03-01 21:11:27.0003780… 192.168.53.5     192.168.60.100    TELNET    69 Telnet Data ...
20 2020-03-01 21:11:27.0004153… 192.168.85.136   192.168.85.137    UDP       97 48975 → 55555 Len=53
21 2020-03-01 21:11:27.0007788… 192.168.85.137   192.168.85.136    ICMP     125 Destination unreachable (Port unreachable)
22 2020-03-01 21:11:27.2122047… 192.168.53.5     192.168.60.100    TCP       70 [TCP Retransmission] 49572 → 23 [PSH, ACK] Seq…
23 2020-03-01 21:11:27.2122856… 192.168.85.136   192.168.85.137    UDP       98 48975 → 55555 Len=54
24 2020-03-01 21:11:27.2133136… 192.168.85.137   192.168.85.136    ICMP     126 Destination unreachable (Port unreachable)
25 2020-03-01 21:11:27.6481175… 192.168.53.5     192.168.60.100    TCP       70 [TCP Retransmission] 49572 → 23 [PSH, ACK] Seq…
26 2020-03-01 21:11:27.6481491… 192.168.85.136   192.168.85.137    UDP       98 48975 → 55555 Len=54
27 2020-03-01 21:11:27.6484499… 192.168.85.137   192.168.85.136    ICMP     126 Destination unreachable (Port unreachable)
28 2020-03-01 21:11:28.5122956… 192.168.53.5     192.168.60.100    TCP       70 [TCP Retransmission] 49572 → 23 [PSH, ACK] Seq…
29 2020-03-01 21:11:28.5123626… 192.168.85.136   192.168.85.137    UDP       98 48975 → 55555 Len=54
30 2020-03-01 21:11:28.5130090… 192.168.85.137   192.168.85.136    ICMP     126 Destination unreachable (Port unreachable)
31 2020-03-01 21:11:28.5131939… ::1              ::1               UDP       64 35303 → 53406 Len=0
32 2020-03-01 21:11:30.2080654… 192.168.53.5     192.168.60.100    TCP       70 [TCP Retransmission] 49572 → 23 [PSH, ACK] Seq…
```

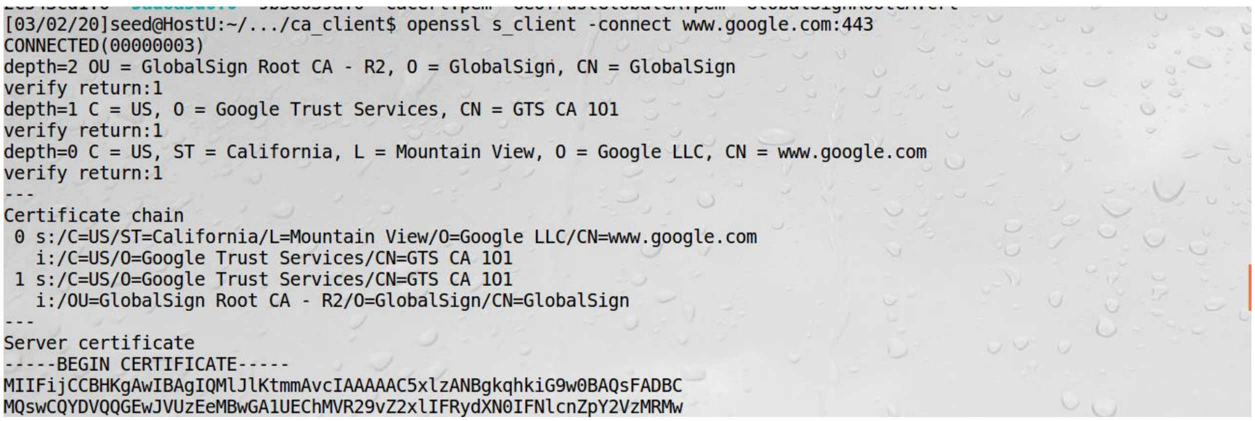


```
                    m TUN
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
Got a packet from TUN
Got a packet from the tunnel
^C
[03/01/20]seed@VPNServer:~/.../VPN$
```

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
'default         192.168.60.1    0.0.0.0         UG    100    0        0 ens38
default         192.168.85.2    0.0.0.0         UG    101    0        0 ens33
10.0.0.0        *               255.255.255.0   U     0      0        0 tun0
link-local      *               255.255.0.0     U     1000   0        0 ens38
192.168.60.0    *               255.255.255.0   U     100    0        0 ens38
192.168.85.0    *               255.255.255.0   U     100    0        0 ens33
[03/01/20]seed@VPNServer:~$ ^C
[03/01/20]seed@VPNServer:~$ sudo ifconfig tun0 192.168.53.1/24 up
[03/01/20]seed@VPNServer:~$ sudo sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
[03/01/20]seed@VPNServer:~$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[03/01/20]seed@VPNServer:~$
```

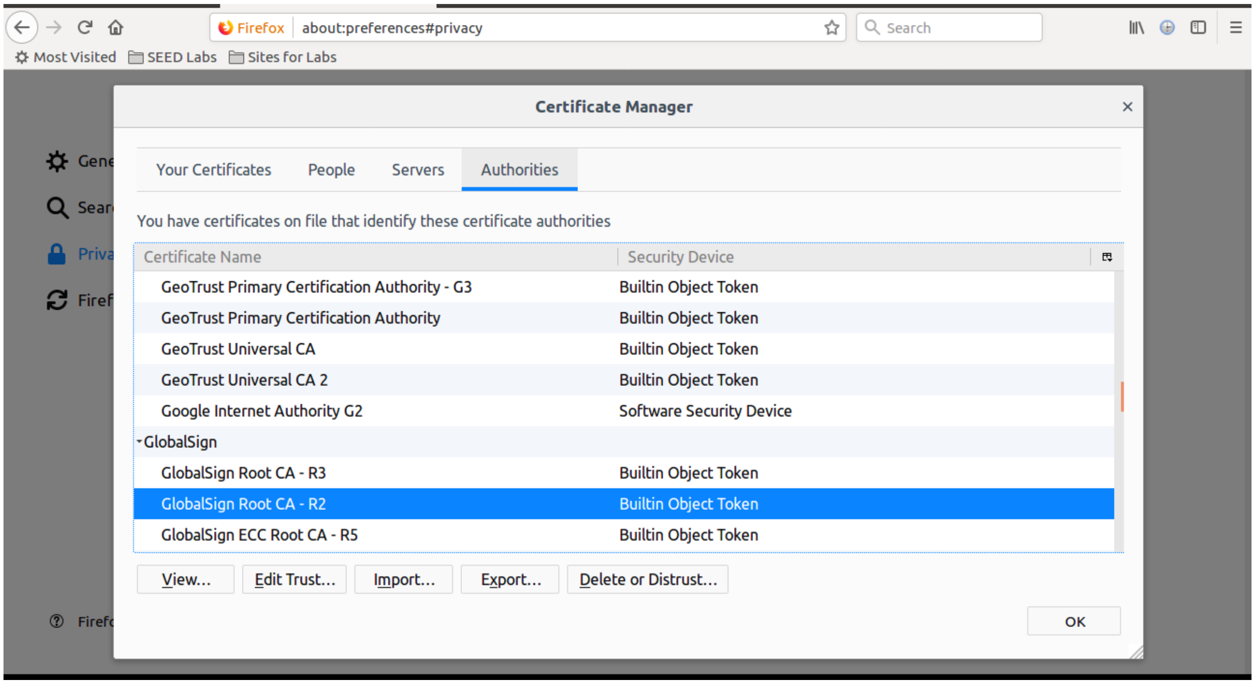
Task 4

Ran the open SSL command to see what root certificate ww.google.com used so I can grab the cert and authenticate myself to google.com.



```
[03/02/20]seed@HostU:~/.../ca_client$ openssl s_client -connect www.google.com:443
CONNECTED(00000003)
depth=2 OU = GlobalSign Root CA - R2, O = GlobalSign, CN = GlobalSign
verify return:1
depth=1 C = US, O = Google Trust Services, CN = GTS CA 1O1
verify return:1
depth=0 C = US, ST = California, L = Mountain View, O = Google LLC, CN = www.google.com
verify return:1
---
Certificate chain
 0 s:/C=US/ST=California/L=Mountain View/O=Google LLC/CN=www.google.com
   i:/C=US/O=Google Trust Services/CN=GTS CA 1O1
 1 s:/C=US/O=Google Trust Services/CN=GTS CA 1O1
   i:/OU=GlobalSign Root CA - R2/O=GlobalSign/CN=GlobalSign
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIFijCCBHKgAwIBAgIQMlJlKtmmAvcIAAAAAC5xlzANBgkqhkiG9w0BAQsFADBC
MQswCQYDVQQGEwJVUzEeMBwGA1UEChMVR29vZ2xlIFRydXN0IFNlcnZpY2VzMRMw
```

Found it uses a globalsign root ca-r2 Certificate, now going to my browser and export it to my certificate folder.



Certificate Manager

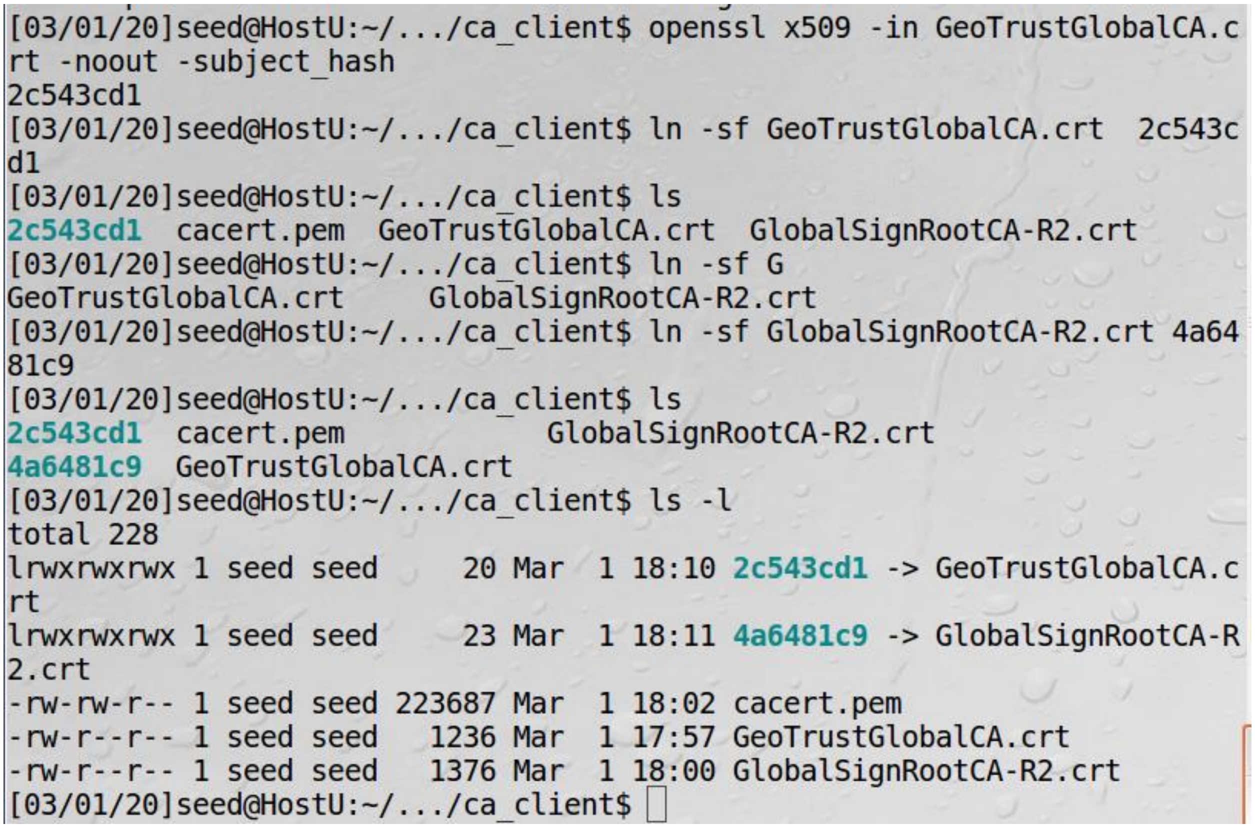
Your Certificates | People | Servers | **Authorities**

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
GeoTrust Primary Certification Authority - G3	Builtin Object Token
GeoTrust Primary Certification Authority	Builtin Object Token
GeoTrust Universal CA	Builtin Object Token
GeoTrust Universal CA 2	Builtin Object Token
Google Internet Authority G2	Software Security Device
˅GlobalSign	
GlobalSign Root CA - R3	Builtin Object Token
GlobalSign Root CA - R2	Builtin Object Token
GlobalSign ECC Root CA - R5	Builtin Object Token

View... | Edit Trust... | Import... | Export... | Delete or Distrust...

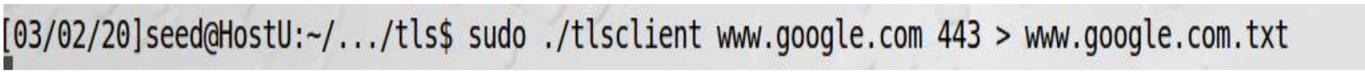
OK



```
[03/01/20]seed@HostU:~/.../ca_client$ openssl x509 -in GeoTrustGlobalCA.c
rt -noout -subject_hash
2c543cd1
[03/01/20]seed@HostU:~/.../ca_client$ ln -sf GeoTrustGlobalCA.crt  2c543c
d1
[03/01/20]seed@HostU:~/.../ca_client$ ls
2c543cd1  cacert.pem  GeoTrustGlobalCA.crt  GlobalSignRootCA-R2.crt
[03/01/20]seed@HostU:~/.../ca_client$ ln -sf G
GeoTrustGlobalCA.crt     GlobalSignRootCA-R2.crt
[03/01/20]seed@HostU:~/.../ca_client$ ln -sf GlobalSignRootCA-R2.crt 4a64
81c9
[03/01/20]seed@HostU:~/.../ca_client$ ls
2c543cd1  cacert.pem            GlobalSignRootCA-R2.crt
4a6481c9  GeoTrustGlobalCA.crt
[03/01/20]seed@HostU:~/.../ca_client$ ls -l
total 228
lrwxrwxrwx 1 seed seed     20 Mar  1 18:10 2c543cd1 -> GeoTrustGlobalCA.c
rt
lrwxrwxrwx 1 seed seed     23 Mar  1 18:11 4a6481c9 -> GlobalSignRootCA-R
2.crt
-rw-rw-r-- 1 seed seed 223687 Mar  1 18:02 cacert.pem
-rw-r--r-- 1 seed seed   1236 Mar  1 17:57 GeoTrustGlobalCA.crt
-rw-r--r-- 1 seed seed   1376 Mar  1 18:00 GlobalSignRootCA-R2.crt
[03/01/20]seed@HostU:~/.../ca_client$ 
```

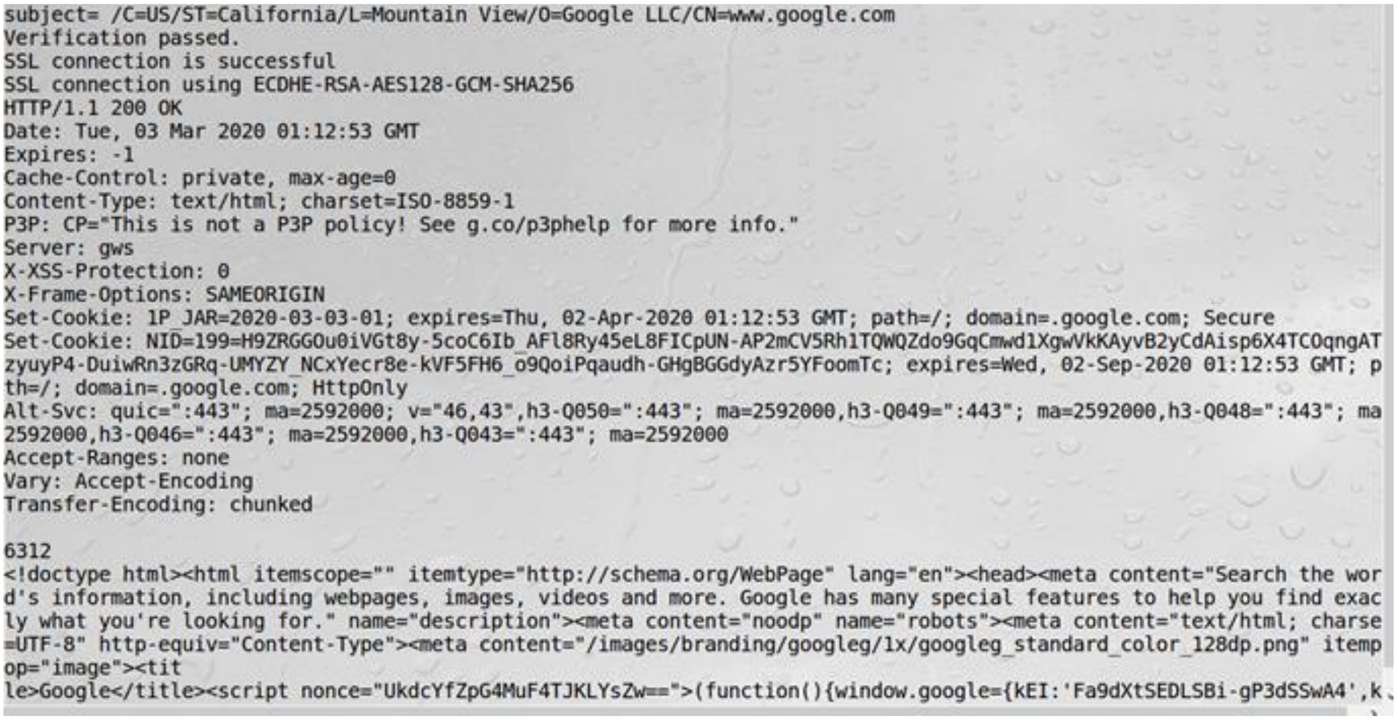
Now with the file I used openssl to create a hash for the server to verify the machine say who it is.

It generate a hash value, then I used LN command to create a link between the hash value for the server to points to the file and its hash.



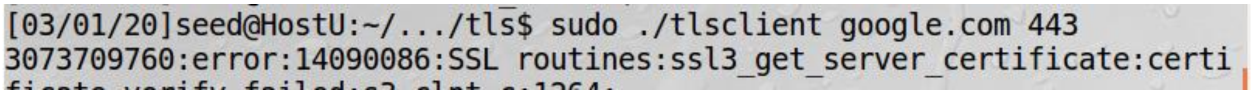
```
[03/02/20]seed@HostU:~/.../tls$ sudo ./tlsclient www.google.com 443 > www.google.com.txt
```

Looking inside the text file and the verification check was successful because of the content displaying inside the file.



```
subject= /C=US/ST=California/L=Mountain View/O=Google LLC/CN=www.google.com
Verification passed.
SSL connection is successful
SSL connection using ECDHE-RSA-AES128-GCM-SHA256
HTTP/1.1 200 OK
Date: Tue, 03 Mar 2020 01:12:53 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2020-03-03-01; expires=Thu, 02-Apr-2020 01:12:53 GMT; path=/; domain=.google.com; Secure
Set-Cookie: NID=199=H9ZRGGOu0iVGt8y-5coC6Ib_AFl8Ry45eL8FICpUN-AP2mCV5Rh1TQWQZdo9GqCmwd1XgwVkKAyvB2yCdAisp6X4TCOqngAT
zyuyP4-DuiwRn3zGRq-UMYZY_NCxYecr8e-kVF5FH6_o9QoiPqaudh-GHgBGGdyAzr5YFoomTc; expires=Wed, 02-Sep-2020 01:12:53 GMT; p
th=/; domain=.google.com; HttpOnly
Alt-Svc: quic=":443"; ma=2592000; v="46,43",h3-Q050=":443"; ma=2592000,h3-Q049=":443"; ma=2592000,h3-Q048=":443"; ma
2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked

6312
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta content="Search the wor
d's information, including webpages, images, videos and more. Google has many special features to help you find exac
ly what you're looking for." name="description"><meta content="noodp" name="robots"><meta content="text/html; charse
=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemp
op="image"><tit
le>Google</title><script nonce="UkdcYfZpG4MuF4TJKLYsZw==">(function(){window.google={kEI:'Fa9dXtSEDLSBi-gP3dSSwA4',k
```

Here is a full example if I did not verified the certificate properly.

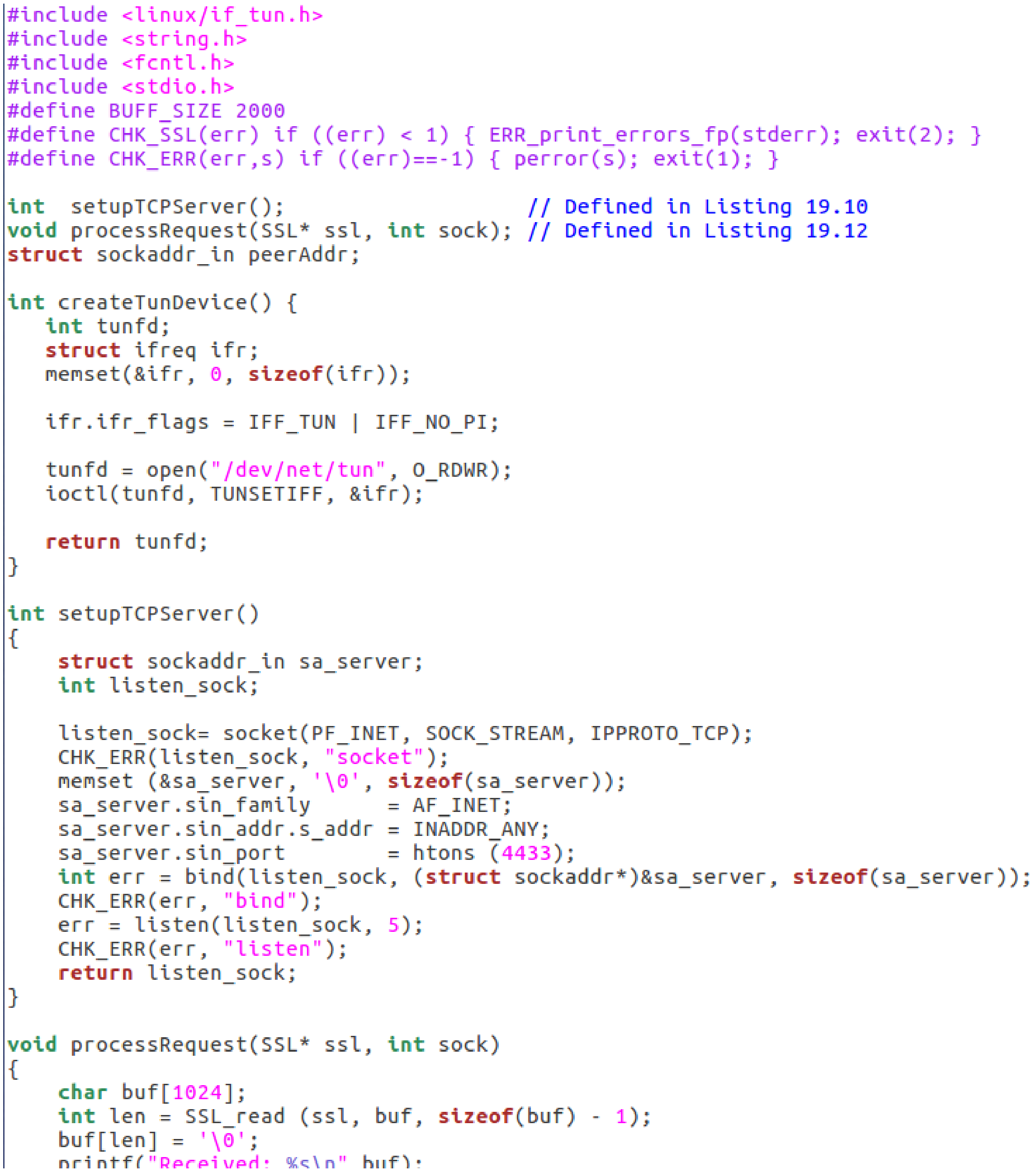


```
[03/01/20]seed@HostU:~/.../tls$ sudo ./tlsclient google.com 443
3073709760:error:14090086:SSL routines:ssl3_get_server_certificate:certi
ficate verify failed:s3_clnt.c:1264:
```

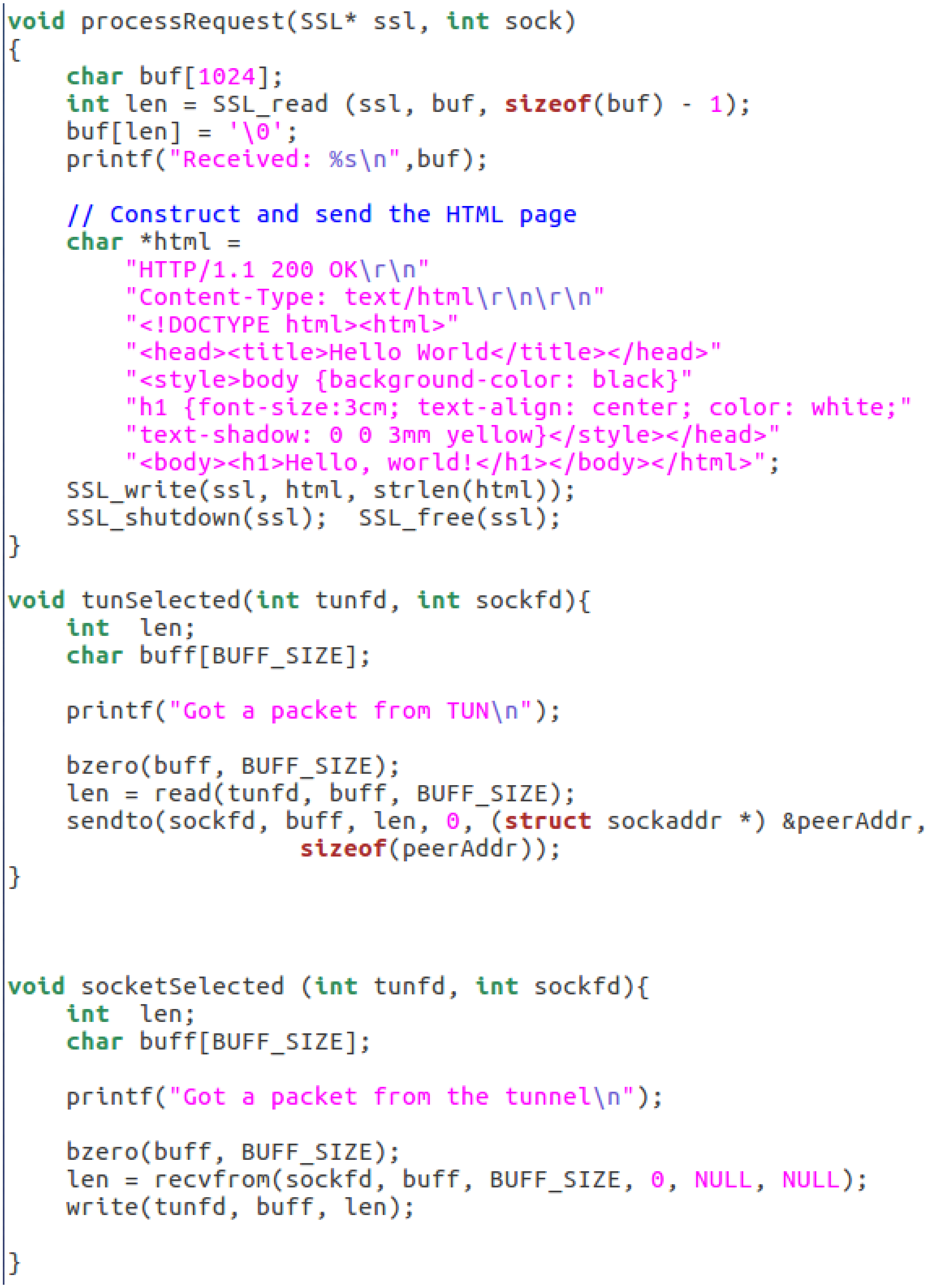
Here is an attemet to make my vpn serer script and tls/server script to gether to create a vpn that has a tunnel for encription.

I tried to add the interface script to the tls/server script and tried to run the script but had issues with segmentation failure that associate with the handling of memeory when I ran the script.

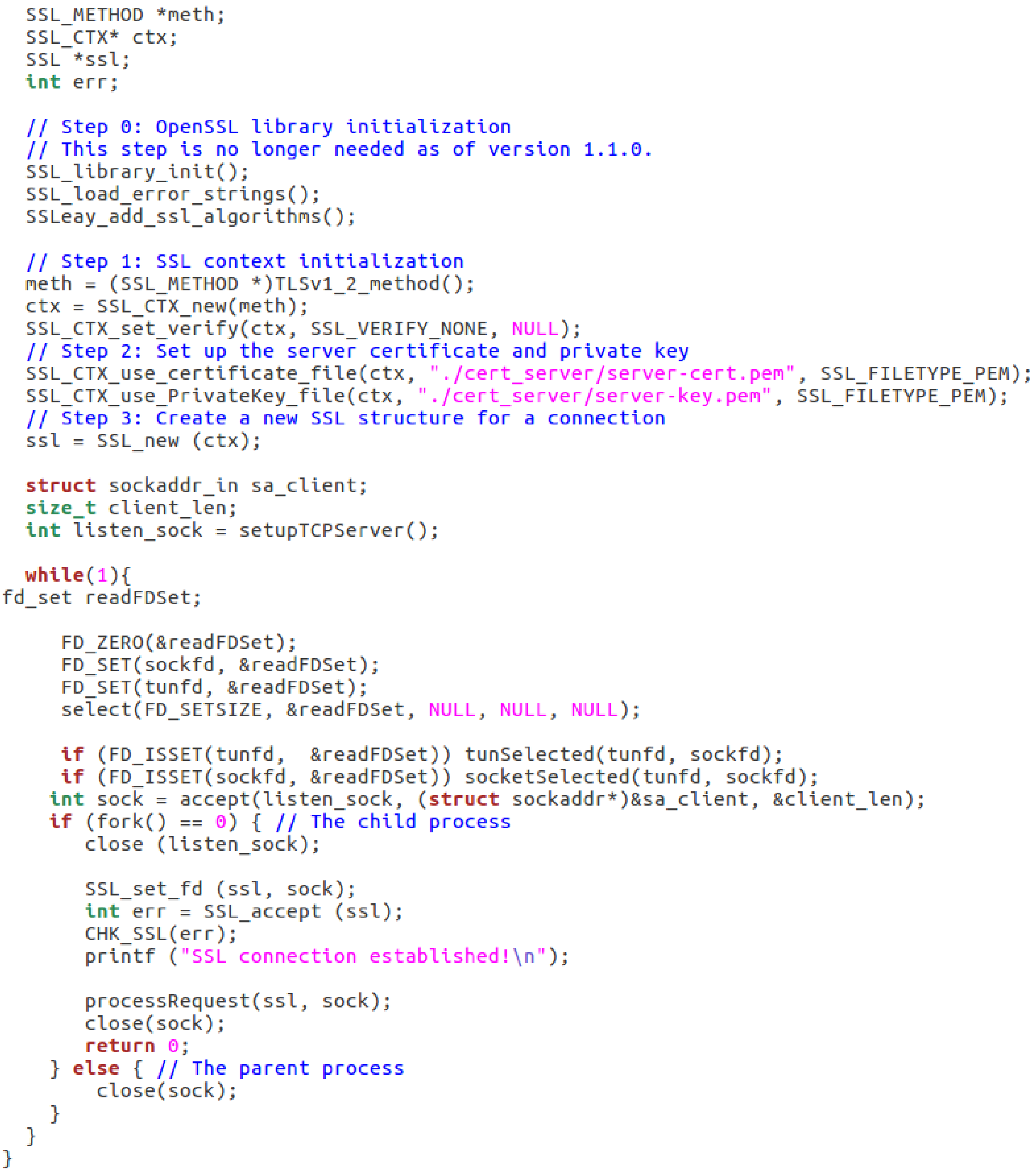
Here is the code I used.



```
#include <linux/if_tun.h>
#include <string.h>
#include <fcntl.h>
#include <stdio.h>
#define BUFF_SIZE 2000
#define CHK_SSL(err) if ((err) < 1) { ERR_print_errors_fp(stderr); exit(2); }
#define CHK_ERR(err,s) if ((err)==-1) { perror(s); exit(1); }

int  setupTCPServer();                   // Defined in Listing 19.10
void processRequest(SSL* ssl, int sock); // Defined in Listing 19.12
struct sockaddr_in peerAddr;

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

   ifr.ifr_flags = IFF_TUN | IFF_NO_PI;

   tunfd = open("/dev/net/tun", O_RDWR);
   ioctl(tunfd, TUNSETIFF, &ifr);

   return tunfd;
}

int setupTCPServer()
{
    struct sockaddr_in sa_server;
    int listen_sock;

    listen_sock= socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);
    CHK_ERR(listen_sock, "socket");
    memset (&sa_server, '\0', sizeof(sa_server));
    sa_server.sin_family      = AF_INET;
    sa_server.sin_addr.s_addr = INADDR_ANY;
    sa_server.sin_port        = htons (4433);
    int err = bind(listen_sock, (struct sockaddr*)&sa_server, sizeof(sa_server));
    CHK_ERR(err, "bind");
    err = listen(listen_sock, 5);
    CHK_ERR(err, "listen");
    return listen_sock;
}

void processRequest(SSL* ssl, int sock)
{
    char buf[1024];
    int len = SSL_read (ssl, buf, sizeof(buf) - 1);
    buf[len] = '\0';
    printf("Received: %s\n",buf);
```

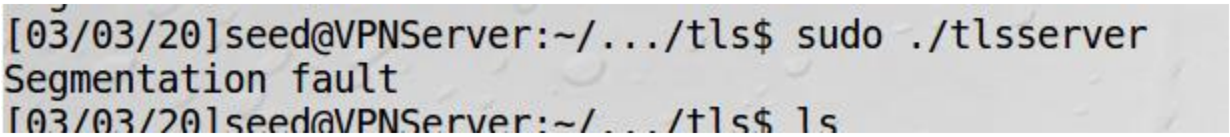


```
void processRequest(SSL* ssl, int sock)
{
    char buf[1024];
    int len = SSL_read (ssl, buf, sizeof(buf) - 1);
    buf[len] = '\0';
    printf("Received: %s\n",buf);

    // Construct and send the HTML page
    char *html =
        "HTTP/1.1 200 OK\r\n"
        "Content-Type: text/html\r\n\r\n"
        "<!DOCTYPE html><html>"
        "<head><title>Hello World</title></head>"
        "<style>body {background-color: black}"
        "h1 {font-size:3cm; text-align: center; color: white;"
        "text-shadow: 0 0 3mm yellow}</style></head>"
        "<body><h1>Hello, world!</h1></body></html>";
    SSL_write(ssl, html, strlen(html));
    SSL_shutdown(ssl);  SSL_free(ssl);
}

void tunSelected(int tunfd, int sockfd){
    int  len;
    char buff[BUFF_SIZE];

    printf("Got a packet from TUN\n");

    bzero(buff, BUFF_SIZE);
    len = read(tunfd, buff, BUFF_SIZE);
    sendto(sockfd, buff, len, 0, (struct sockaddr *) &peerAddr,
                    sizeof(peerAddr));
}



void socketSelected (int tunfd, int sockfd){
    int  len;
    char buff[BUFF_SIZE];

    printf("Got a packet from the tunnel\n");

    bzero(buff, BUFF_SIZE);
    len = recvfrom(sockfd, buff, BUFF_SIZE, 0, NULL, NULL);
    write(tunfd, buff, len);

}
```



```
  SSL_METHOD *meth;
  SSL_CTX* ctx;
  SSL *ssl;
  int err;

  // Step 0: OpenSSL library initialization
  // This step is no longer needed as of version 1.1.0.
  SSL_library_init();
  SSL_load_error_strings();
  SSLeay_add_ssl_algorithms();

  // Step 1: SSL context initialization
  meth = (SSL_METHOD *)TLSv1_2_method();
  ctx = SSL_CTX_new(meth);
  SSL_CTX_set_verify(ctx, SSL_VERIFY_NONE, NULL);
  // Step 2: Set up the server certificate and private key
  SSL_CTX_use_certificate_file(ctx, "./cert_server/server-cert.pem", SSL_FILETYPE_PEM);
  SSL_CTX_use_PrivateKey_file(ctx, "./cert_server/server-key.pem", SSL_FILETYPE_PEM);
  // Step 3: Create a new SSL structure for a connection
  ssl = SSL_new (ctx);

  struct sockaddr_in sa_client;
  size_t client_len;
  int listen_sock = setupTCPServer();

  while(1){
fd_set readFDSet;

     FD_ZERO(&readFDSet);
     FD_SET(sockfd, &readFDSet);
     FD_SET(tunfd, &readFDSet);
     select(FD_SETSIZE, &readFDSet, NULL, NULL, NULL);

     if (FD_ISSET(tunfd,  &readFDSet)) tunSelected(tunfd, sockfd);
     if (FD_ISSET(sockfd, &readFDSet)) socketSelected(tunfd, sockfd);
    int sock = accept(listen_sock, (struct sockaddr*)&sa_client, &client_len);
    if (fork() == 0) { // The child process
       close (listen_sock);

       SSL_set_fd (ssl, sock);
       int err = SSL_accept (ssl);
       CHK_SSL(err);
       printf ("SSL connection established!\n");

       processRequest(ssl, sock);
       close(sock);
       return 0;
    } else { // The parent process
        close(sock);
    }
  }
}
```

Here is the output of the failure.



```
[03/03/20]seed@VPNServer:~/.../tls$ sudo ./tlsserver
Segmentation fault
[03/03/20]seed@VPNServer:~/.../tls$ ls
```