

NAME: YASH SNEHAL SHETIYA

SUID: 9276568741

LAB: VPN TUNNELING

TASK 1:

```
seed@VM: ~/.../Labsetup
[03/05/23] seed@VM:~/.../Labsetup$ dockps
444483f8aa26 host-192.168.60.6
eba6ecd7978f client-10.9.0.5
d3c56f1642e5 host-192.168.60.5
8d0eff81e4e9 server-router
[03/05/23] seed@VM:~/.../Labsetup$
```

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup x seed@VM: ~/.../Labsetup x seed@VM: ~/.../Labsetup x seed@VM: ~/.../Labsetup x
[03/05/23] seed@VM:~/.../Labsetup$ docksh 8d
root@8d0eff81e4e9:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.9.0.5: icmp_seq=6 ttl=64 time=0.079 ms
64 bytes from 10.9.0.5: icmp_seq=7 ttl=64 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=8 ttl=64 time=0.045 ms
64 bytes from 10.9.0.5: icmp_seq=9 ttl=64 time=0.051 ms
^C
--- 10.9.0.5 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8168ms
rtt min/avg/max/mdev = 0.045/0.052/0.079/0.009 ms
root@8d0eff81e4e9:/#
```

```
seed@VM: ~/.../Labsetup  seed@VM: ~/.../Labsetup  seed@VM: ~/.../Labsetup  seed@VM: ~/.../Labsetup
[03/05/23]seed@VM:~/.../Labsetup$ docksh 8d
root@8d0eff81e4e9:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.9.0.5: icmp_seq=6 ttl=64 time=0.079 ms
64 bytes from 10.9.0.5: icmp_seq=7 ttl=64 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=8 ttl=64 time=0.045 ms
64 bytes from 10.9.0.5: icmp_seq=9 ttl=64 time=0.051 ms
^C
--- 10.9.0.5 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8168ms
rtt min/avg/max/mdev = 0.045/0.052/0.079/0.009 ms
root@8d0eff81e4e9:/#
```

```
seed@VM: ~/.../Labsetup  seed@VM: ~/.../Labsetup  seed@VM: ~/.../Labsetup  seed@VM: ~/.../Labsetup
[03/05/23]seed@VM:~/.../Labsetup$ docksh client-10.9.0.5
root@eba6ecd7978f:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.071 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=7 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=8 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=9 ttl=64 time=0.048 ms
^C
--- 10.9.0.11 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8172ms
rtt min/avg/max/mdev = 0.048/0.050/0.071/0.007 ms
root@eba6ecd7978f:/#
```

```
seed@VM: ~/.../Labsetup
root@8d0eff81e4e9:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=0.071 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.050 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=64 time=0.044 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=64 time=0.048 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=64 time=0.048 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=64 time=0.048 ms
^C
--- 192.168.60.5 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9199ms
rtt min/avg/max/mdev = 0.044/0.050/0.071/0.007 ms
root@8d0eff81e4e9:/#
```

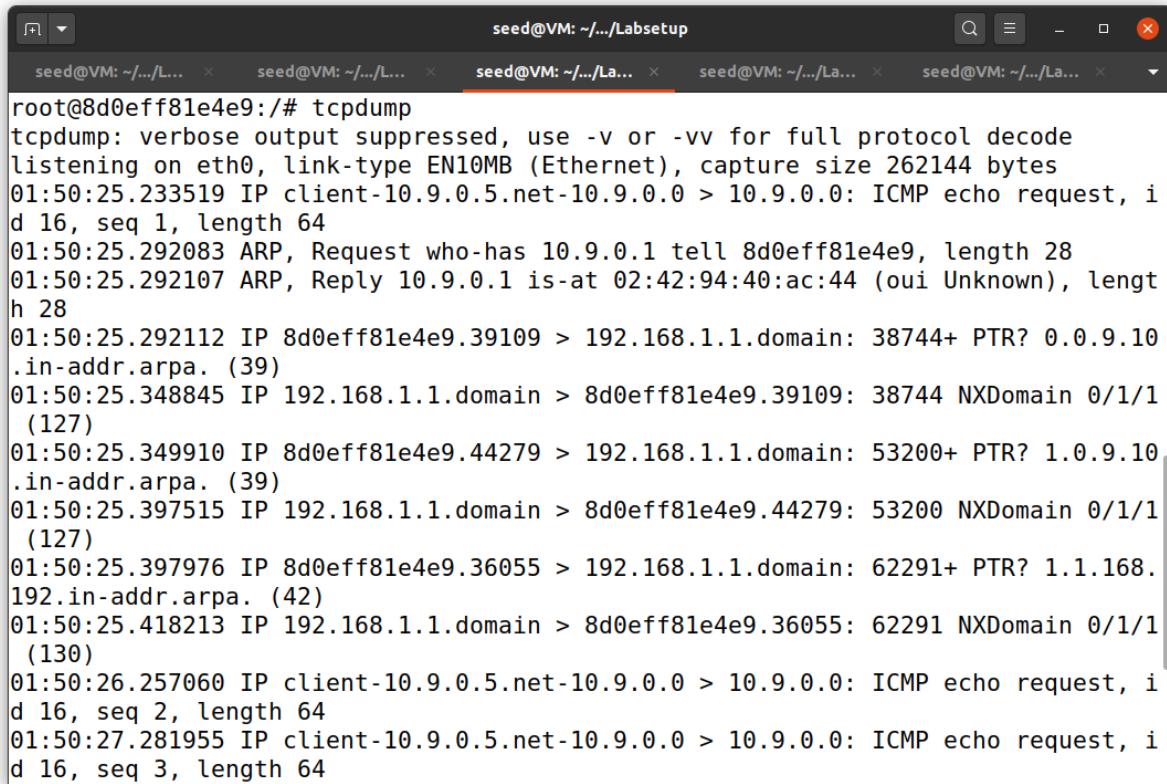
```
seed@VM: ~/.../Labsetup
[03/05/23] seed@VM:~/.../Labsetup$ docksh host-192.168.60.5
root@d3c56f1642e5:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.045 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.051 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.090 ms
64 bytes from 10.9.0.11: icmp_seq=7 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=9 ttl=64 time=0.046 ms
64 bytes from 10.9.0.11: icmp_seq=10 ttl=64 time=0.090 ms
64 bytes from 10.9.0.11: icmp_seq=11 ttl=64 time=0.050 ms
^C
--- 10.9.0.11 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10232ms
rtt min/avg/max/mdev = 0.045/0.056/0.090/0.016 ms
root@d3c56f1642e5:/#
```

Host U to Host V

```
[03/05/23]seed@VM:~/.../Labsetup$ docksh host-192.168.60.5
root@d3c56f1642e5:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.045 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.051 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.090 ms
64 bytes from 10.9.0.11: icmp_seq=7 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=9 ttl=64 time=0.046 ms
64 bytes from 10.9.0.11: icmp_seq=10 ttl=64 time=0.090 ms
64 bytes from 10.9.0.11: icmp_seq=11 ttl=64 time=0.050 ms
^C
--- 10.9.0.11 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10232ms
rtt min/avg/max/mdev = 0.045/0.056/0.090/0.016 ms
root@d3c56f1642e5:/#
```

Host V to Host U:

TCP dump at Host U:

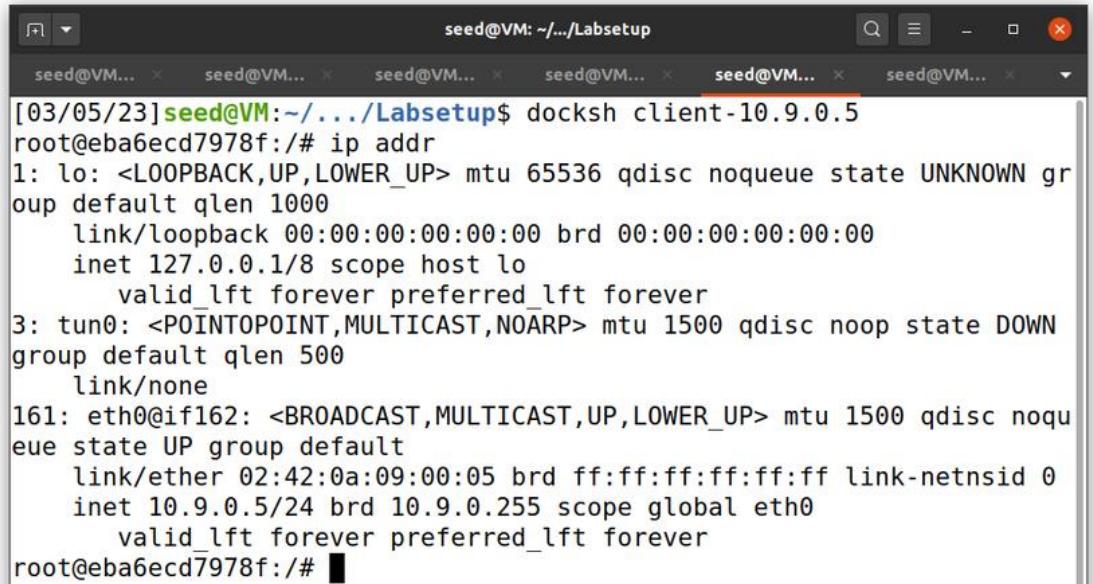


A terminal window titled "seed@VM: ~.../Labsetup" showing the output of the "tcpdump" command. The output shows network traffic on interface eth0, including ICMP echo requests and replies, ARP requests and replies, and DNS queries for domain names like "192.168.1.1.domain".

```
root@8d0eff81e4e9:# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
01:50:25.233519 IP client-10.9.0.5.net-10.9.0.0 > 10.9.0.0: ICMP echo request, id 16, seq 1, length 64
01:50:25.292083 ARP, Request who-has 10.9.0.1 tell 8d0eff81e4e9, length 28
01:50:25.292107 ARP, Reply 10.9.0.1 is-at 02:42:94:40:ac:44 (oui Unknown), length 28
01:50:25.292112 IP 8d0eff81e4e9.39109 > 192.168.1.1.domain: 38744+ PTR? 0.0.9.10.in-addr.arpa. (39)
01:50:25.348845 IP 192.168.1.1.domain > 8d0eff81e4e9.39109: 38744 NXDomain 0/1/1(127)
01:50:25.349910 IP 8d0eff81e4e9.44279 > 192.168.1.1.domain: 53200+ PTR? 1.0.9.10.in-addr.arpa. (39)
01:50:25.397515 IP 192.168.1.1.domain > 8d0eff81e4e9.44279: 53200 NXDomain 0/1/1(127)
01:50:25.397976 IP 8d0eff81e4e9.36055 > 192.168.1.1.domain: 62291+ PTR? 1.1.168.192.in-addr.arpa. (42)
01:50:25.418213 IP 192.168.1.1.domain > 8d0eff81e4e9.36055: 62291 NXDomain 0/1/1(130)
01:50:26.257060 IP client-10.9.0.5.net-10.9.0.0 > 10.9.0.0: ICMP echo request, id 16, seq 2, length 64
01:50:27.281955 IP client-10.9.0.5.net-10.9.0.0 > 10.9.0.0: ICMP echo request, id 16, seq 3, length 64
```

2 a)

Before:



A terminal window titled "seed@VM: ~.../Labsetup" showing the output of the "ip addr" command. It lists network interfaces including lo, tun0, and eth0, detailing their MTU, queueing discipline (qdisc), and link layer information.

```
[03/05/23] seed@VM:~/.../Labsetup$ docksh client-10.9.0.5
root@eba6ecd7978f:/# ip addr
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
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
161: eth0@if162: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
root@eba6ecd7978f:/# █
```

```
seed@VM: ~/.../Labsetup
[03/05/23] seed@VM:~/.../Labsetup$ docksh client-10.9.0.5
root@eba6ecd7978f:/# vi tun.py
bash: vi: command not found
root@eba6ecd7978f:/# cd volumes
root@eba6ecd7978f:/volumes# vi tun.py
bash: vi: command not found
root@eba6ecd7978f:/volumes# tun.py
Interface Name: tun0
```

```
seed@VM: ~/.../Labsetup
[03/05/23] seed@VM:~/.../Labsetup$ docksh client-10.9.0.5
root@eba6ecd7978f:/# vi tun.py
bash: vi: command not found
root@eba6ecd7978f:/# cd volumes
root@eba6ecd7978f:/volumes# vi tun.py
bash: vi: command not found
root@eba6ecd7978f:/volumes# tun.py
Interface Name: tun0
^CTraceback (most recent call last):
  File "./tun.py", line 24, in <module>
    time.sleep(10)
KeyboardInterrupt
^X
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes# tun.py
Interface Name: Sheti0
```

```
seed@VM: ~/.../Labsetup
seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM...
oup 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
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN
group default qlen 500
    link/none
161: eth0@if162: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
        inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
            valid_lft forever preferred_lft forever
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/# ip addr
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
4: Sheti0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
161: eth0@if162: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
        inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
            valid_lft forever preferred_lft forever
root@eba6ecd7978f:/# █
```

2 b)

Assigning ip

```
seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM...
root@eba6ecd7978f:/#
root@eba6ecd7978f:/# ip addr add 192.168.53.99/24 dev Sheti0
root@eba6ecd7978f:/# ip addr
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
4: Sheti0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 scope global Sheti0
        valid_lft forever preferred_lft forever
161: eth0@if162: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/# ip link set dev Sheti0 up
root@eba6ecd7978f:/# ip addr
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
4: Sheti0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 scope global Sheti0
        valid_lft forever preferred_lft forever
```

Bringing it up:

```
seed@VM... x seed@VM... x seed@VM... x seed@VM... x seed@VM... x seed@VM... x seed@VM...
root@eba6ecd7978f:/# ip link set dev Sheti0 up
root@eba6ecd7978f:/# ip addr
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
4: Sheti0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
        inet 192.168.53.99/24 scope global Sheti0
            valid_lft forever preferred_lft forever
161: eth0@if162: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
        inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
            valid_lft forever preferred_lft forever
root@eba6ecd7978f:/#
```

Code:

```
seed@VM... x seed@VM... x seed@VM... x seed@VM... x seed@VM... x seed@VM...
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x4000454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'Sheti%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

while True:
    time.sleep(10)

~
~
~

-- INSERT (paste) --
```

2 c)

Code:

```
#!/usr/bin/env python3

import fcntl
import struct
import os
import time
from scapy.all import *

TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'Sheti%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
        ip = IP(packet)
        print(ip.summary())
-- INSERT (paste) --
```

External network:

```
seed@VM: ~/Labsetup
seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM...
root@eba6ecd7978f:/volumes# 
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes# tun.py
Interface Name: Sheti0

^CTraceback (most recent call last):
  File "./tun.py", line 24, in <module>
    time.sleep(10)
KeyboardInterrupt

root@eba6ecd7978f:/volumes# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
^C
--- 192.168.53.1 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4103ms

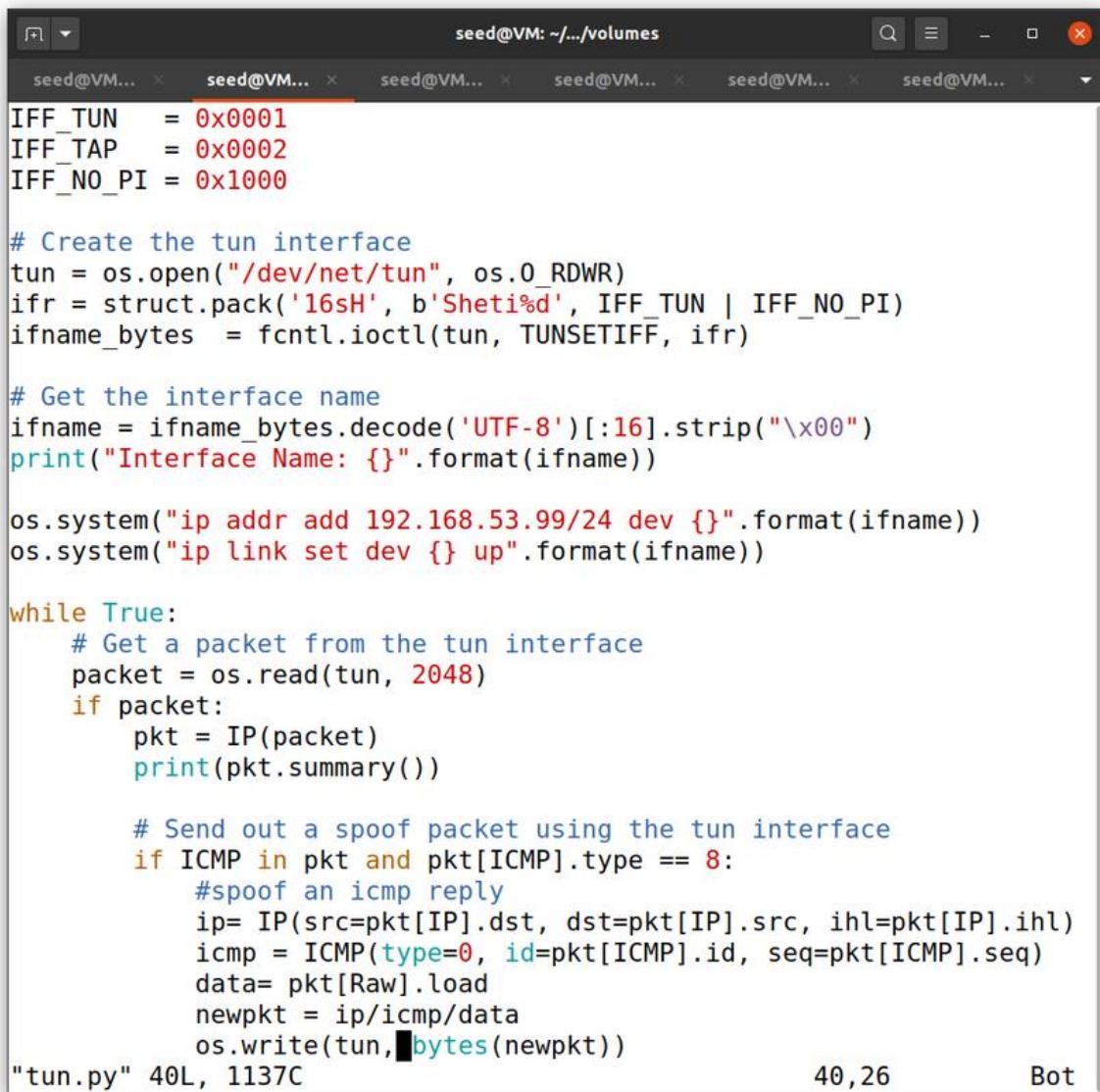
root@eba6ecd7978f:/volumes# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
^C
--- 192.168.53.1 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5119ms

root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8186ms

root@eba6ecd7978f:/volumes#
```

2 d)

Code: Sending spoof packet



The screenshot shows a terminal window titled "seed@VM: ~.../volumes" with multiple tabs open. The current tab contains Python code for creating a TUN interface and sending a spoofed ICMP packet. The code uses the `os` and `struct` modules to handle file operations and interface configuration. It then reads packets from the TUN interface, identifies ICMP type 8 (echo request), and creates a spoofed ICMP reply packet to send back. The terminal also displays the file name and line count ("tun.py" 40L, 1137C) and some status information ("40,26 Bot").

```
IFF_TUN    = 0x0001
IFF_TAP    = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'Sheti%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
        pkt = IP(packet)
        print(pkt.summary())

    # Send out a spoof packet using the tun interface
    if ICMP in pkt and pkt[ICMP].type == 8:
        #spoof an icmp reply
        ip= IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
        icmp = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq)
        data= pkt[Raw].load
        newpkt = ip/icmp/data
        os.write(tun, bytes(newpkt))
```

Pinging:

```
root@eba6ecd7978f:/volumes# ping 192.168.53.3
PING 192.168.53.3 (192.168.53.3) 56(84) bytes of data.
64 bytes from 192.168.53.3: icmp_seq=1 ttl=64 time=2.70 ms
64 bytes from 192.168.53.3: icmp_seq=2 ttl=64 time=2.59 ms
64 bytes from 192.168.53.3: icmp_seq=3 ttl=64 time=2.24 ms
64 bytes from 192.168.53.3: icmp_seq=4 ttl=64 time=2.26 ms
64 bytes from 192.168.53.3: icmp_seq=5 ttl=64 time=2.39 ms
64 bytes from 192.168.53.3: icmp_seq=6 ttl=64 time=2.39 ms
64 bytes from 192.168.53.3: icmp_seq=7 ttl=64 time=2.33 ms
64 bytes from 192.168.53.3: icmp_seq=8 ttl=64 time=2.43 ms
^C
--- 192.168.53.3 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7016ms
rtt min/avg/max/mdev = 2.244/2.418/2.698/0.147 ms
root@eba6ecd7978f:/volumes#
```

Code exec:

```
root@eba6ecd7978f:/volumes# chmod a+x tun.py
root@eba6ecd7978f:/volumes# tun.py
Interface Name: Sheti0
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
```

We can see that packets were sent and received:

```
seed@VM: ~.../Labsetup
seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM...
root@eba6ecd7978f:/volumes# 
root@eba6ecd7978f:/volumes# 
root@eba6ecd7978f:/volumes# 
root@eba6ecd7978f:/volumes# 
root@eba6ecd7978f:/volumes# ping 192.168.53.3
PING 192.168.53.3 (192.168.53.3) 56(84) bytes of data.
64 bytes from 192.168.53.3: icmp_seq=1 ttl=64 time=1.90 ms
64 bytes from 192.168.53.3: icmp_seq=2 ttl=64 time=1.61 ms
64 bytes from 192.168.53.3: icmp_seq=3 ttl=64 time=2.77 ms
64 bytes from 192.168.53.3: icmp_seq=4 ttl=64 time=1.67 ms
64 bytes from 192.168.53.3: icmp_seq=5 ttl=64 time=1.63 ms
64 bytes from 192.168.53.3: icmp_seq=6 ttl=64 time=1.74 ms
64 bytes from 192.168.53.3: icmp_seq=7 ttl=64 time=1.70 ms
^C
--- 192.168.53.3 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6010ms
rtt min/avg/max/mdev = 1.611/1.860/2.769/0.381 ms
-----
```

3)

Server code:

```
seed@VM: ~.../Labsetup
seed@VM: ~.../volumes
seed@VM: ~.../Labsetup
seed@VM: ~.../volumes

#!/usr/bin/env python3
from scapy.all import*

IP_A = "0.0.0.0"
PORT = 9090
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))

while True:
    data, (ip, port) = sock.recvfrom(2048)
    print("{}:{} --> {}:{}".format(ip, port, IP_A, PORT))
    pkt = IP(data)
    print("  Inside: {} --> {}".format(pkt.src, pkt.dst))
```

Client side:

```
seed@VM: ~.../volumes
seed@VM: ~.../Labsetup          seed@VM: ~.../volumes

import time
from scapy.all import*
TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'Sheti%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

SERVER_PORT = 9090
SERVER_IP = "10.9.0.11"

# Create UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
        # Send the packet via the tunnel
        sock.sendto(packet, (SERVER_IP, SERVER_PORT))

"tun_client.py" 36L, 924C           36,0-1      Bot
```

Server side output:

```
seed@VM: ~/.../Labsetup
[03/06/23] seed@VM:~/.../Labsetup$ docksh server-router
root@8d0eff81e4e9:/# cd volumes
root@8d0eff81e4e9:/volumes# chmod a+x tun_server.py
root@8d0eff81e4e9:/volumes# tun_server.py
10.9.0.5:43835 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.53.3
```

Client side output:

```
seed@VM: ~.../Labsetup
[03/06/23] seed@VM:~/.../Labsetup$ docksh client-10.9.0.5
root@eba6ecd7978f:/# cd volumes
root@eba6ecd7978f:/volumes# chmod a+x tun_client.py
root@eba6ecd7978f:/volumes# tun_client.py
File "./tun_client.py", line 15
    ifr = struct.pack('16sH', b'Sheti%d', IFF_TUN | IFF_NO_PI)
                                         ^
SyntaxError: invalid character in identifier
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes# tun_client.py
Interface Name: Sheti0
```

Pinging interface:

```
seed@VM: ~.../Labsetup
[03/06/23] seed@VM:~/.../Labsetup$ docksh e9
Error: No such container: e9
[03/06/23] seed@VM:~/.../Labsetup$ docksh eb
root@eba6ecd7978f:/# ping 192.168.53.3
bash: ping: command not found
root@eba6ecd7978f:/# ping 192.168.53.3
PING 192.168.53.3 (192.168.53.3) 56(84) bytes of data.
^C
--- 192.168.53.3 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11256ms
root@eba6ecd7978f:/#
```

Pinging in another network:

```
seed@VM: ~.../Labsetup
[03/06/23] seed@VM:~/.../Labsetup$ docksh e9
Error: No such container: e9
[03/06/23] seed@VM:~/.../Labsetup$ docksh eb
root@eba6ecd7978f:/# ping 192.168.53.3
bash: ping: command not found
root@eba6ecd7978f:/# ping 192.168.53.3
PING 192.168.53.3 (192.168.53.3) 56(84) bytes of data.
^C
--- 192.168.53.3 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11256ms
root@eba6ecd7978f:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
```

No response outputs:

Adding ip route:

```
seed@VM: ~/Labsetup
seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM... seed@VM...
root@eba6ecd7978f:/# ing 192.168.53.3
bash: ing: command not found
root@eba6ecd7978f:/# ping 192.168.53.3
PING 192.168.53.3 (192.168.53.3) 56(84) bytes of data.
^C
--- 192.168.53.3 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11256ms

root@eba6ecd7978f:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
177 packets transmitted, 0 received, 100% packet loss, time 180208ms

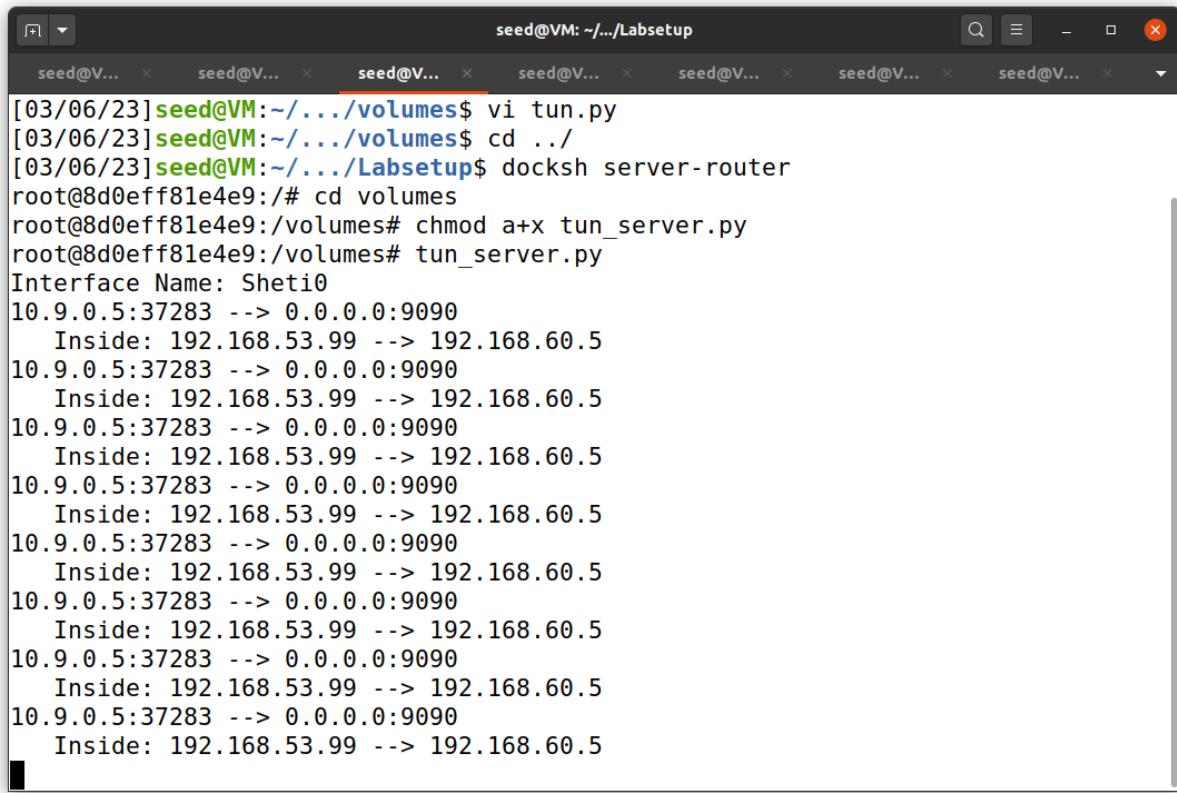
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/# ip route add 192.168.60.5 dev Sheti0
root@eba6ecd7978f:/# ip route
default via 10.9.0.1 dev eth0
10.9.0.0/24 dev eth0 proto kernel scope link src 10.9.0.5
192.168.53.0/24 dev Sheti0 proto kernel scope link src 192.168.53.99
192.168.60.5 dev Sheti0 scope link
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
44 packets transmitted, 0 received, 100% packet loss, time 44010ms

root@eba6ecd7978f:/#
```

We can see once we added the route, it is responsive:

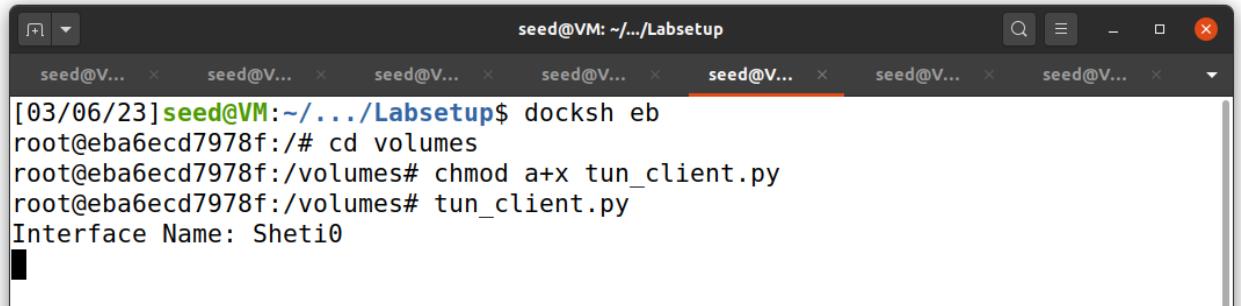
Task 4:

Server side :



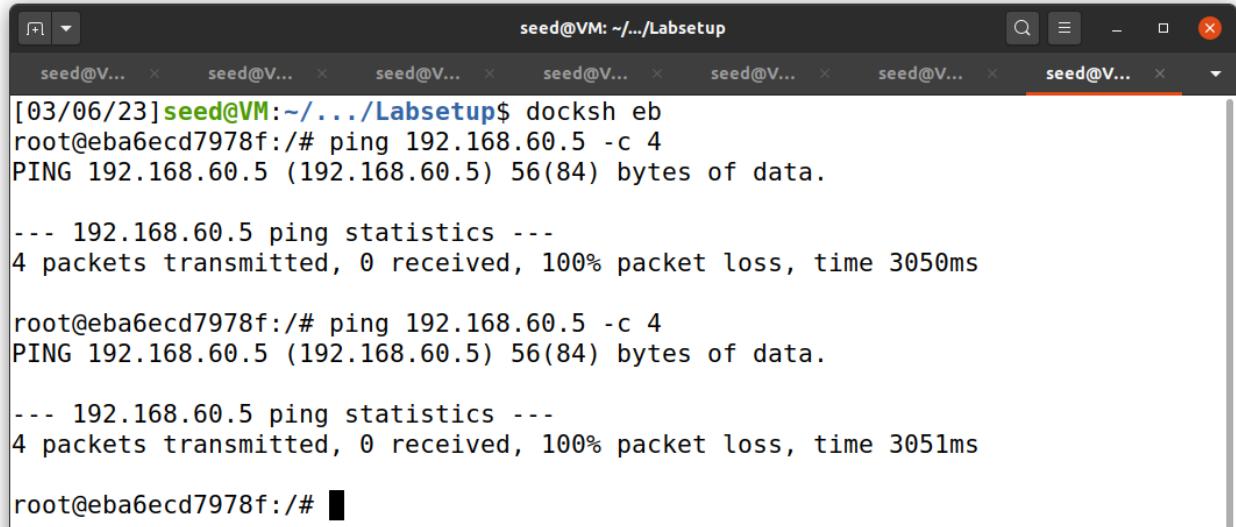
```
seed@VM: ~/.../Labsetup
[03/06/23] seed@VM:~/.../volumes$ vi tun.py
[03/06/23] seed@VM:~/.../volumes$ cd ../
[03/06/23] seed@VM:~/.../Labsetup$ docksh server-router
root@8d0eff81e4e9:/# cd volumes
root@8d0eff81e4e9:/volumes# chmod a+x tun_server.py
root@8d0eff81e4e9:/volumes# tun_server.py
Interface Name: Sheti0
10.9.0.5:37283 --> 0.0.0.0:9090
    Inside: 192.168.53.99 --> 192.168.60.5
```

Client side:



```
seed@VM: ~/.../Labsetup
[03/06/23] seed@VM:~/.../Labsetup$ docksh eb
root@eba6ecd7978f:/# cd volumes
root@eba6ecd7978f:/volumes# chmod a+x tun_client.py
root@eba6ecd7978f:/volumes# tun_client.py
Interface Name: Sheti0
```

Pinging:



A terminal window titled "seed@VM: ~/.../Labsetup" showing the output of a ping command. The command "ping 192.168.60.5 -c 4" is run twice. The first ping shows 100% packet loss. The second ping also shows 100% packet loss. The terminal has multiple tabs open, all labeled "seed@V..." except for the active tab.

```
[03/06/23]seed@VM:~/.../Labsetup$ docksh eb
root@eba6ecd7978f:/# ping 192.168.60.5 -c 4
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.

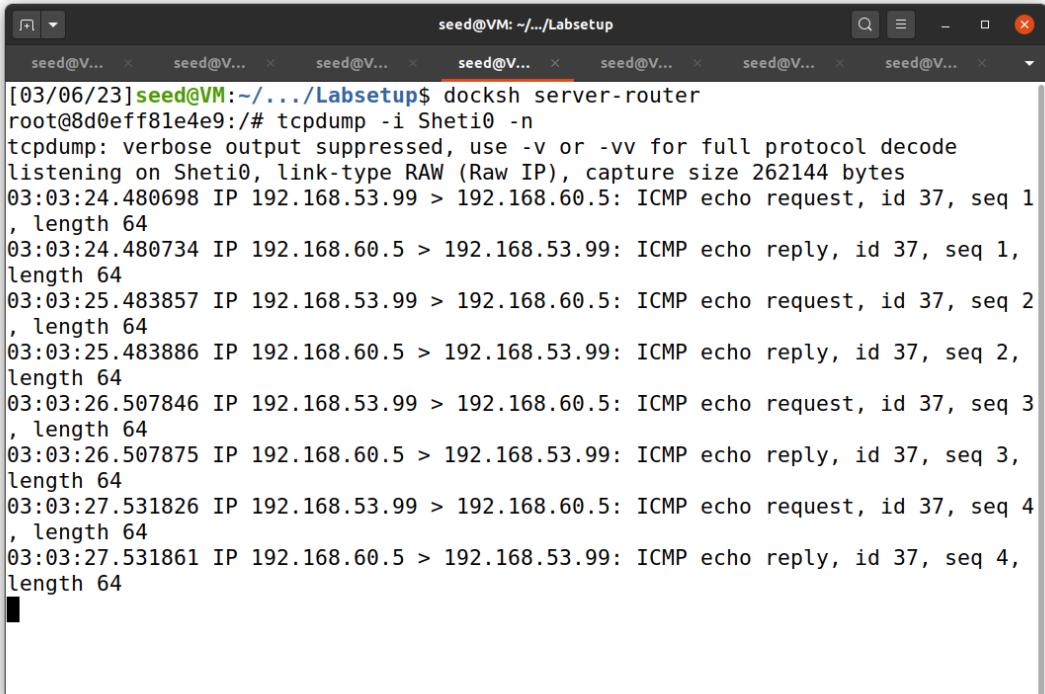
--- 192.168.60.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3050ms

root@eba6ecd7978f:/# ping 192.168.60.5 -c 4
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.

--- 192.168.60.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3051ms

root@eba6ecd7978f:/#
```

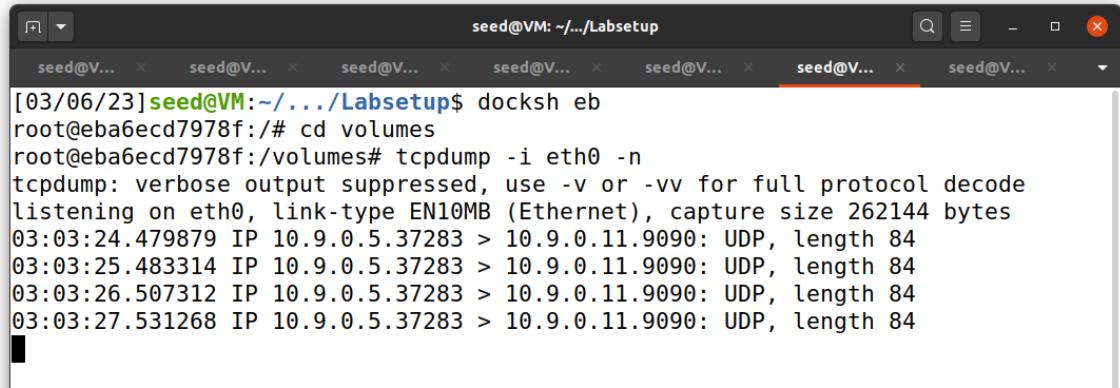
Tcp dump at server:



A terminal window titled "seed@VM: ~/.../Labsetup" showing the output of a tcpdump command. The command "tcpdump -i Sheti0 -n" is run. It captures several ICMP echo requests and replies between two hosts. The terminal has multiple tabs open, all labeled "seed@V..." except for the active tab.

```
[03/06/23]seed@VM:~/.../Labsetup$ docksh server-router
root@8d0eff81e4e9:/# tcpdump -i Sheti0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on Sheti0, link-type RAW (Raw IP), capture size 262144 bytes
03:03:24.480698 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 37, seq 1
, length 64
03:03:24.480734 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 37, seq 1,
length 64
03:03:25.483857 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 37, seq 2
, length 64
03:03:25.483886 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 37, seq 2,
length 64
03:03:26.507846 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 37, seq 3
, length 64
03:03:26.507875 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 37, seq 3,
length 64
03:03:27.531826 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 37, seq 4
, length 64
03:03:27.531861 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 37, seq 4,
length 64
```

Tcp dump:



A screenshot of a terminal window titled "seed@VM: ~.../Labsetup". The window contains a command-line session where the user runs "docksh eb", "cd volumes", and "tcpdump -i eth0 -n". The output shows several UDP packets being captured on interface eth0, all originating from IP 10.9.0.5 and destined for IP 10.9.0.11, with lengths of 84 bytes.

```
[03/06/23]seed@VM:~/.../Labsetup$ docksh eb
root@eba6ecd7978f:/# cd volumes
root@eba6ecd7978f:/volumes# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
03:03:24.479879 IP 10.9.0.5.37283 > 10.9.0.11.9090: UDP, length 84
03:03:25.483314 IP 10.9.0.5.37283 > 10.9.0.11.9090: UDP, length 84
03:03:26.507312 IP 10.9.0.5.37283 > 10.9.0.11.9090: UDP, length 84
03:03:27.531268 IP 10.9.0.5.37283 > 10.9.0.11.9090: UDP, length 84
```

At host V:

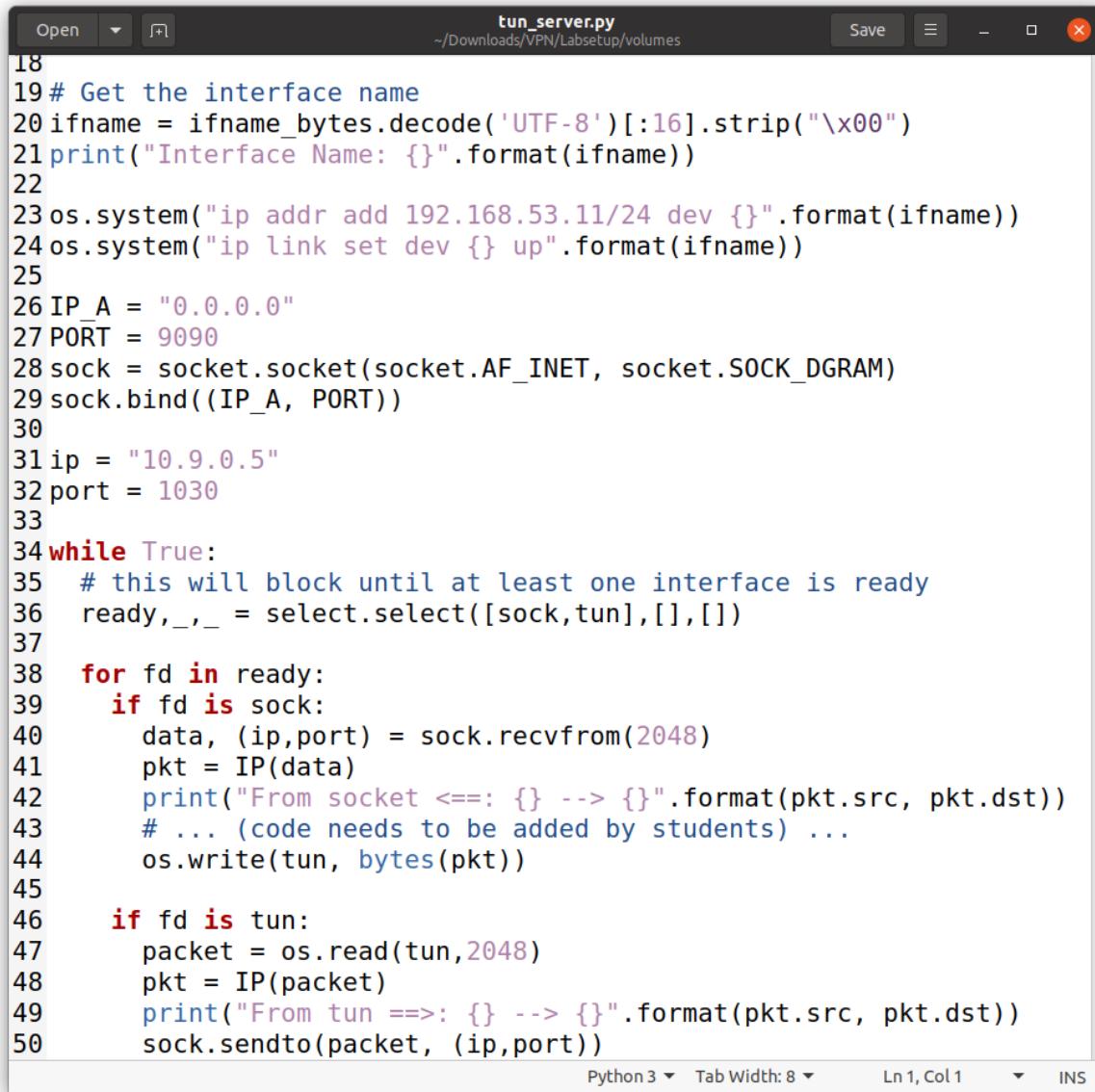
When tried to ping a private network, we can see that we are able to ping and get replies in the server but it is not showing on the host as the dual traffic hasn't been configured yet.

It can be seen that packets have arrived at host V

```
03:11:48.479401 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 38, seq 1
, length 64
03:11:48.479417 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 38, seq 1,
length 64
03:11:49.484560 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 38, seq 2
, length 64
03:11:49.484577 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 38, seq 2,
length 64
03:11:50.507828 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 38, seq 3
, length 64
03:11:50.507843 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 38, seq 3,
length 64
03:11:51.531841 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 38, seq 4
, length 64
03:11:51.531856 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 38, seq 4,
length 64
03:11:53.483211 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
03:11:53.483587 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
03:11:53.483601 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
03:11:53.483604 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
```

Task 5:

CODES:



The screenshot shows a code editor window with the file `tun_server.py` open. The code is a Python script for a TUN/TAP interface server. It starts by getting the interface name, creating a TUN interface, and binding it to a port. It then enters a loop where it waits for data on either the socket or the TUN interface. If data comes from the socket, it prints the source and destination IP addresses. If data comes from the TUN interface, it reads it, prints the source and destination IP addresses, and then sends it back to the socket. The code uses the `socket` and `select` modules.

```
18
19 # Get the interface name
20 ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
21 print("Interface Name: {}".format(ifname))
22
23 os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
24 os.system("ip link set dev {} up".format(ifname))
25
26 IP_A = "0.0.0.0"
27 PORT = 9090
28 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
29 sock.bind((IP_A, PORT))
30
31 ip = "10.9.0.5"
32 port = 1030
33
34 while True:
35     # this will block until at least one interface is ready
36     ready, _, _ = select.select([sock,tun],[],[])
37
38     for fd in ready:
39         if fd is sock:
40             data, (ip,port) = sock.recvfrom(2048)
41             pkt = IP(data)
42             print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))
43             # ... (code needs to be added by students) ...
44             os.write(tun, bytes(pkt))
45
46         if fd is tun:
47             packet = os.read(tun,2048)
48             pkt = IP(packet)
49             print("From tun ==>: {} --> {}".format(pkt.src, pkt.dst))
50             sock.sendto(packet, (ip,port))
```

A screenshot of a code editor window titled "tun_client.py". The file path is shown as "~/Downloads/VPN/Labsetup/volumes". The code is a Python script for a network client. It uses the socket module to handle UDP traffic. The script starts by setting up a TUN interface and adding routes. It then enters a loop where it waits for data on either the socket or the TUN interface. If data comes from the socket, it prints the source and destination IP addresses. If data comes from the TUN interface, it prints the source and destination IP addresses and then sends the packet back to the server. The script uses the IP library to parse network packets.

```
22
23 os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
24 os.system("ip link set dev {} up".format(ifname))
25
26 SERVER_PORT = 9090
27 SERVER_IP = "10.9.0.11"
28
29 ip = "10.9.0.5"
30 port = 1030
31
32 os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))
33
34 # Create UDP socket
35 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
36
37
38 while True:
39     # this will block until at least one interface is ready
40     ready, _, _ = select.select([sock,tun],[],[])
41
42     for fd in ready:
43         if fd is sock:
44             data, (ip,port) = sock.recvfrom(2048)
45             pkt = IP(data)
46             print("From socket ==>: {} --> {}".format(pkt.src, pkt.dst))
47             # ... (code needs to be added by students) ...
48             os.write(tun, bytes(pkt))
49
50         if fd is tun:
51             packet = os.read(tun,2048)
52             pkt = IP(packet)
53             print("From tun ==>: {} --> {}".format(pkt.src, pkt.dst))
54             sock.sendto(packet, (SERVER_IP, SERVER_PORT))
```

Client side:

A screenshot of a terminal window titled "seed@VM: ~/.../Labsetup". The command "tun_client.py" is being run. The output shows the client sending and receiving data over the TUN interface. The client's IP is 192.168.53.99 and the server's IP is 192.168.60.5. The client is sending data to the server and receiving data back from the server.

```
root@eba6ecd7978f:/volumes# tun_client.py
Interface Name: Sheti0
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
```

Server side:

```
root@8d0eff81e4e9:/volumes# tun_server.py
Interface Name: Sheti0
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
```

Pinging:

We can see that its successful as Host U and V are both getting requests and replies.

```
root@eba6ecd7978f:/# ping 192.168.60.5 -c 4
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=4.59 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=2.49 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=2.74 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=2.51 ms

--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.485/3.079/4.588/0.876 ms
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
root@eba6ecd7978f:/#
```

Tcp dump at server:

```
seed@VM: ~/Labsetup
root@d3c56f1642e5:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
05:58:42.208044 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 110, seq
1, length 64
05:58:42.208060 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 110, seq 1,
length 64
05:58:43.208179 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 110, seq
2, length 64
05:58:43.208192 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 110, seq 2,
length 64
05:58:44.210353 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 110, seq
3, length 64
05:58:44.210364 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 110, seq 3,
length 64
05:58:45.211721 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 110, seq
4, length 64
05:58:45.211732 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 110, seq 4,
length 64
05:58:47.435205 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
05:58:47.435324 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
05:58:47.435329 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
05:58:47.435340 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
```

Tcp dump:

```
seed@VM: ~/Labsetup
root@8d0eff81e4e9:/# tcpdump -i Sheti0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on Sheti0, link-type RAW (Raw IP), capture size 262144 bytes
05:58:42.208017 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 110, seq
1, length 64
05:58:42.208068 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 110, seq 1,
length 64
05:58:43.208157 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 110, seq
2, length 64
05:58:43.208198 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 110, seq 2,
length 64
05:58:44.210308 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 110, seq
3, length 64
05:58:44.210370 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 110, seq 3,
length 64
05:58:45.211701 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 110, seq
4, length 64
05:58:45.211738 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 110, seq 4,
length 64
```

Establishing telnet connection:

```
root@eba6ecd7978f:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^].
Ubuntu 20.04.1 LTS
d3c56f1642e5 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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

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.

seed@d3c56f1642e5:~$
```

Now when I Run “who am I” on the telnet connection, it gave me reply as “seed”. This was done multiple times and the result of this can be seen in this manner:

WE CAN SEE THAT IN THE SCREENSHOT OF TASK 6.

```
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
```

Wireshark proof

[SEED Labs] Capturing from any						
No.	Time	Source	Destination	Protocol	Length	Info
73	2023-03-07 01:08:35.1...	192.168.60.5	192.168.53.99	TELNET	70	Telnet Data ...
74	2023-03-07 01:08:35.1...	192.168.60.5	192.168.53.99	TCP	70	[TCP Retransmission] 23 → 35076 [PSH, ACK] Seq=794547354 Ack=...
75	2023-03-07 01:08:35.1...	10.9.0.11	10.9.0.5	UDP	98	9990 → 42969 Len=54
76	2023-03-07 01:08:35.1...	10.9.0.11	10.9.0.5	UDP	98	9990 → 42969 Len=54
77	2023-03-07 01:08:35.1...	10.9.0.5	10.9.0.11	UDP	96	42969 → 9990 Len=52
78	2023-03-07 01:08:35.1...	10.9.0.5	10.9.0.11	UDP	96	42969 → 9990 Len=52
79	2023-03-07 01:08:35.1...	192.168.53.99	192.168.60.5	TCP	68	35076 → 23 [ACK] Seq=3992575264 Ack=794547356 Win=501 Len=0 T...
80	2023-03-07 01:08:35.1...	192.168.53.99	192.168.60.5	TCP	68	[TCP Dup ACK 79#1] 35076 → 23 [ACK] Seq=3992575264 Ack=794547356
81	2023-03-07 01:08:35.1...	192.168.60.5	192.168.53.99	TELNET	95	Telnet Data ...
82	2023-03-07 01:08:35.1...	192.168.60.5	192.168.53.99	TCP	95	[TCP Retransmission] 23 → 35076 [PSH, ACK] Seq=794547356 Ack=...
83	2023-03-07 01:08:35.1...	10.9.0.11	10.9.0.5	UDP	123	9990 → 42969 Len=79
84	2023-03-07 01:08:35.1...	10.9.0.11	10.9.0.5	UDP	123	9990 → 42969 Len=79
85	2023-03-07 01:08:35.1...	10.9.0.5	10.9.0.11	UDP	96	42969 → 9990 Len=52
86	2023-03-07 01:08:35.1...	10.9.0.5	10.9.0.11	UDP	96	42969 → 9990 Len=52
87	2023-03-07 01:08:35.1...	192.168.53.99	192.168.60.5	TCP	68	35076 → 23 [ACK] Seq=3992575264 Ack=794547383 Win=501 Len=0 T...
88	2023-03-07 01:08:35.1...	192.168.53.99	192.168.60.5	TCP	68	[TCP Dup ACK 87#1] 35076 → 23 [ACK] Seq=3992575264 Ack=794547383
89	2023-03-07 01:08:40.3...	02:42:0a:09:00:0b		ARP	44	Who has 10.9.0.5? Tell 10.9.0.11
90	2023-03-07 01:08:40.3...	02:42:0a:09:00:0b		ARP	44	Who has 10.9.0.5? Tell 10.9.0.11

Task 6:

Connection freezes once I disconnect from telnet connection from one side:

```
seed@VM: ~.../Labsetup
seed@... x seed@... x
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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

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.

seed@d3c56f1642e5:~$ whoami
seed
seed@d3c56f1642e5:~$ whoami
seed
seed@d3c56f1642e5:~$ whoami
seed
seed@d3c56f1642e5:~$
```

```
seed@VM: ~/.../Labsetup
seed@... x seed@...
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
^CTraceback (most recent call last):
  File "./tun_client.py", line 40, in <module>
    ready,_,_ = select.select([sock,tun],[],[])
KeyboardInterrupt

root@eba6ecd7978f:/volumes#
```

Re-establishing connection:

```
Keyboard (most recent call last):
  File "./tun_client.py", line 40, in <module>
    ready,_,_ = select.select([sock,tun],[],[])
KeyboardInterrupt

root@eba6ecd7978f:/volumes# tun_client.py
Interface Name: Sheti0
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
```

The screenshot shows a terminal window with multiple tabs open, all labeled "seed@...". The active tab displays a series of messages from the Ubuntu system:

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

This system has been minimized by removing packages and content that are not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

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.

At the bottom of the terminal, there is a loop of the "whoami" command output:

```
seed@d3c56f1642e5:~$ whoami  
seed  
seed@d3c56f1642e5:~$ █
```

After breaking the connection at the client side, we can observe that the telnet froze, then when we re-establish the side we can see we are able to type in the telnet.