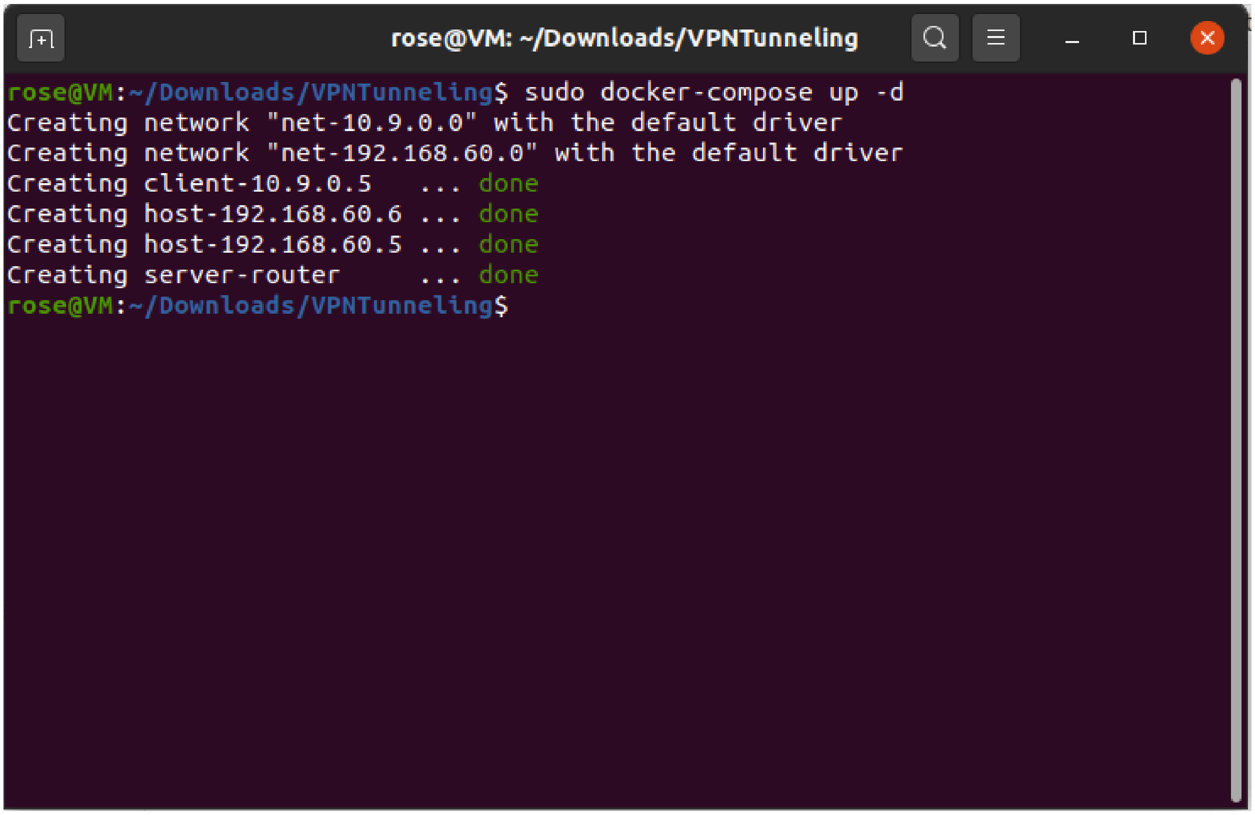
Samantha Jackson
CSCI4321
04-14-2025

Lab 9 - VPN Tunneling

Composing a Docker

I downloaded the files for the lab into my downloads folder in Ubuntu under the name ‘VPNTunneling’ and composed a docker using ‘sudo docker-compose up -d’.

Observations/Issues: None

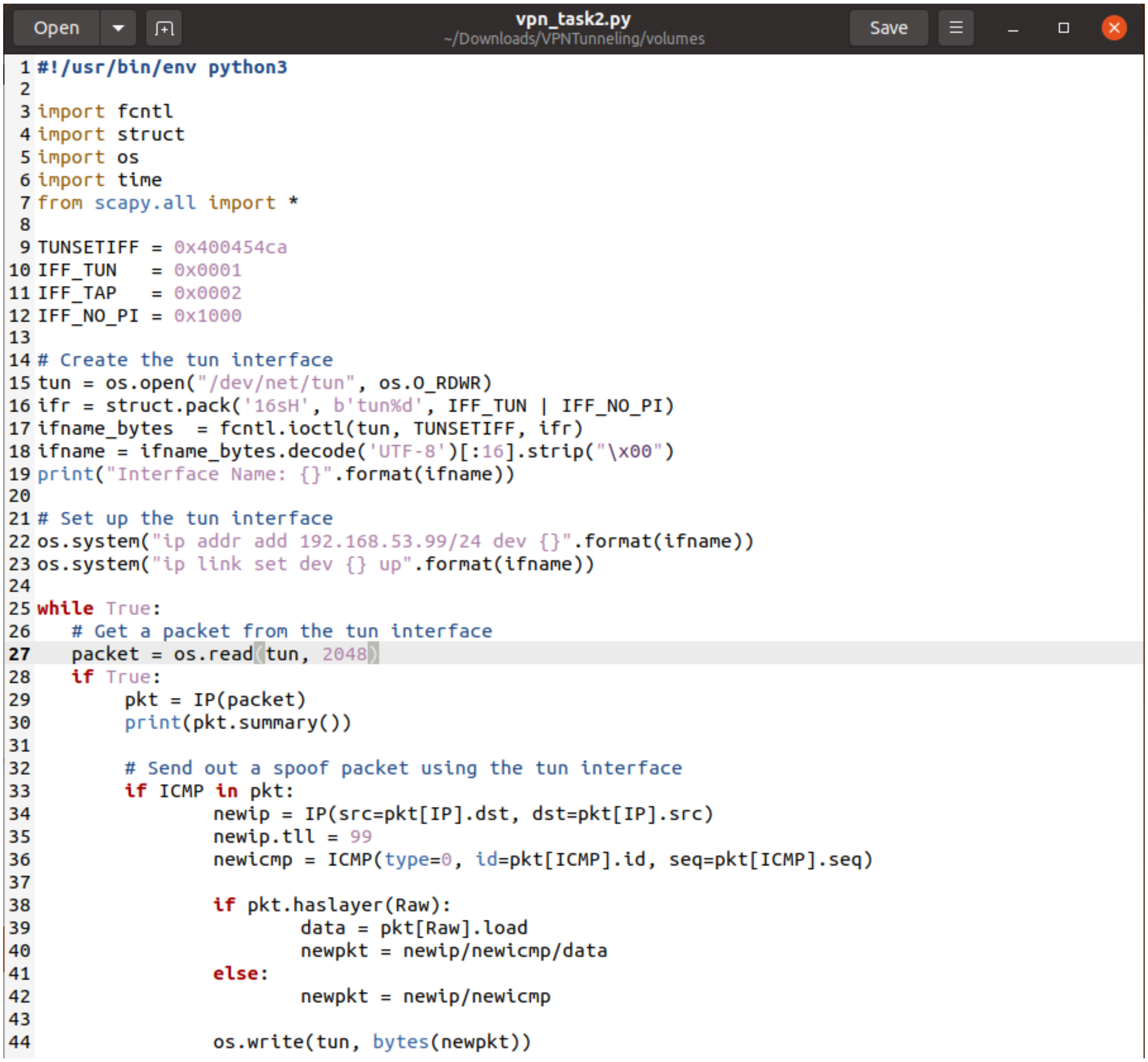


```
rose@VM:~/Downloads/VPNTunneling$ sudo docker-compose up -d
Creating network "net-10.9.0.0" with the default driver
Creating network "net-192.168.60.0" with the default driver
Creating client-10.9.0.5    ... done
Creating host-192.168.60.6 ... done
Creating host-192.168.60.5 ... done
Creating server-router      ... done
rose@VM:~/Downloads/VPNTunneling$
```

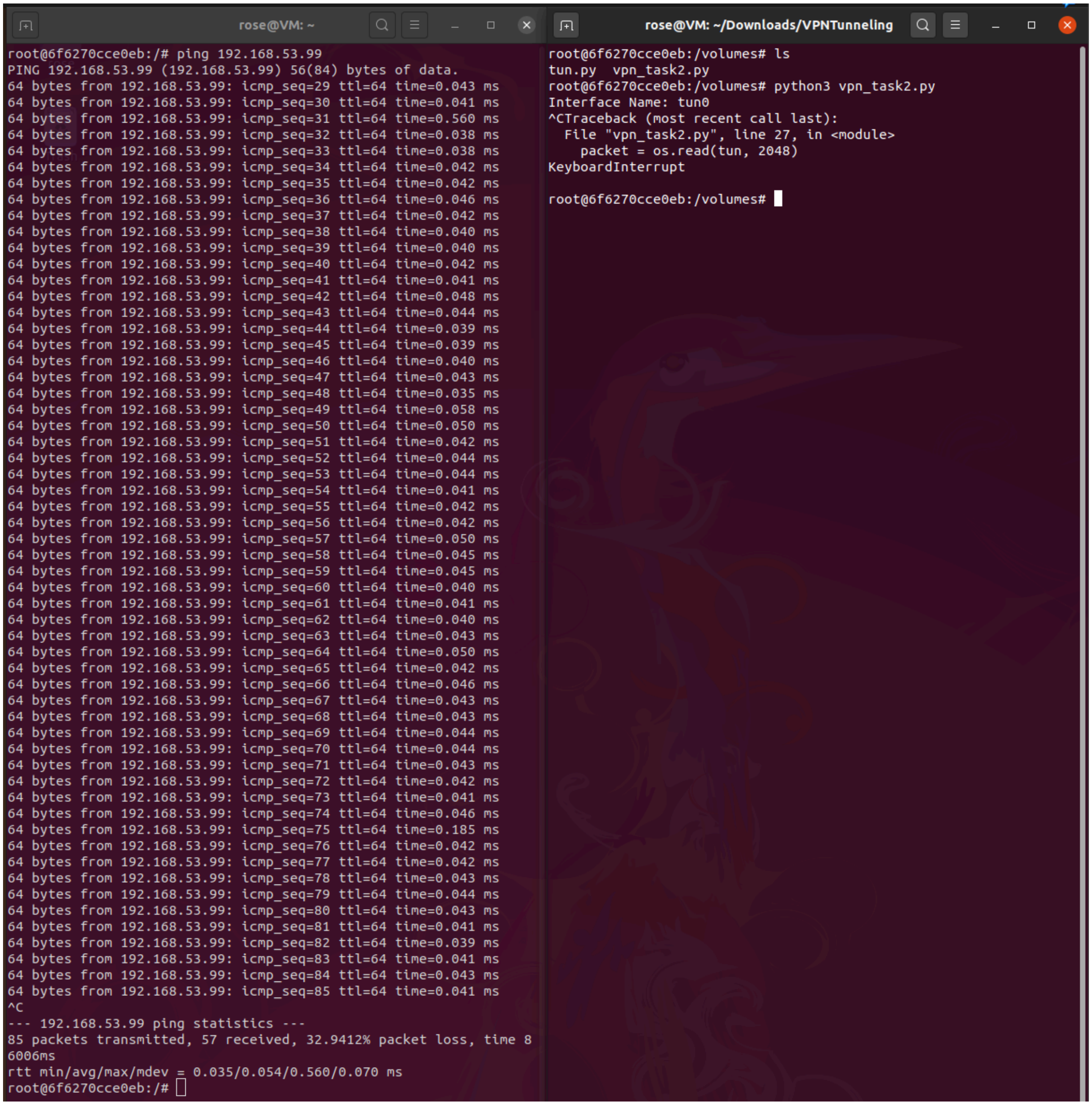
Task-02: VPN Tunneling

I created a program ‘vpn_task2.py’ that creates a virtual interface ‘tun0’ used to intercept traffic. I logged into the VPN Client terminal on two separate instances of the terminal using the command ‘sudo docker exec -it client-10.9.0.5 bash’. On the right I’m executing the vpn_task2.py script in /volumes. On the left I’m pinging 192.168.53.99 and I’m getting continuous replies for this IP that shouldn’t exist demonstrating that the ‘tun0’ interface is working.

Observations/Issues: None



```python
#!/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'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up the tun interface
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 True:
        pkt = IP(packet)
        print(pkt.summary())

        # Send out a spoof packet using the tun interface
        if ICMP in pkt:
                newip = IP(src=pkt[IP].dst, dst=pkt[IP].src)
                newip.ttl = 99
                newicmp = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq)

                if pkt.haslayer(Raw):
                        data = pkt[Raw].load
                        newpkt = newip/newicmp/data
                else:
                        newpkt = newip/newicmp

                os.write(tun, bytes(newpkt))
```



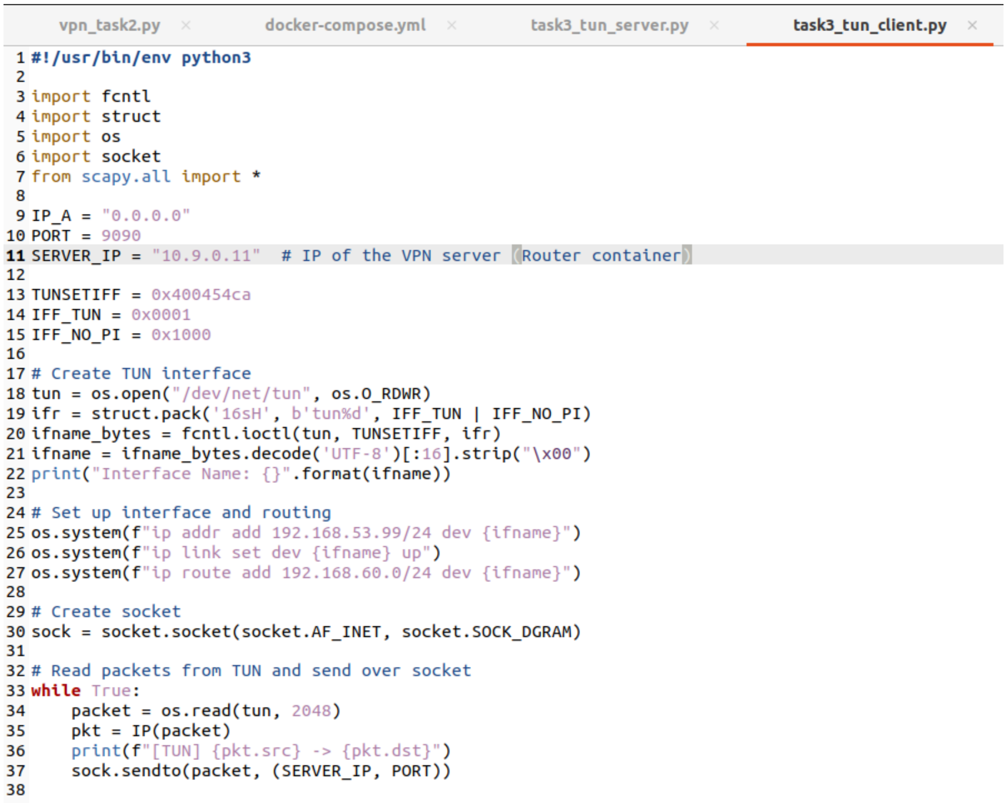
```
root@6f6270cce0eb:/# ping 192.168.53.99
PING 192.168.53.99 (192.168.53.99) 56(84) bytes of data.
64 bytes from 192.168.53.99: icmp_seq=29 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=30 ttl=64 time=0.041 ms
64 bytes from 192.168.53.99: icmp_seq=31 ttl=64 time=0.560 ms
64 bytes from 192.168.53.99: icmp_seq=32 ttl=64 time=0.038 ms
64 bytes from 192.168.53.99: icmp_seq=33 ttl=64 time=0.038 ms
64 bytes from 192.168.53.99: icmp_seq=34 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=35 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=36 ttl=64 time=0.046 ms
64 bytes from 192.168.53.99: icmp_seq=37 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=38 ttl=64 time=0.040 ms
64 bytes from 192.168.53.99: icmp_seq=39 ttl=64 time=0.040 ms
64 bytes from 192.168.53.99: icmp_seq=40 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=41 ttl=64 time=0.041 ms
64 bytes from 192.168.53.99: icmp_seq=42 ttl=64 time=0.048 ms
64 bytes from 192.168.53.99: icmp_seq=43 ttl=64 time=0.044 ms
64 bytes from 192.168.53.99: icmp_seq=44 ttl=64 time=0.039 ms
64 bytes from 192.168.53.99: icmp_seq=45 ttl=64 time=0.039 ms
64 bytes from 192.168.53.99: icmp_seq=46 ttl=64 time=0.040 ms
64 bytes from 192.168.53.99: icmp_seq=47 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=48 ttl=64 time=0.035 ms
64 bytes from 192.168.53.99: icmp_seq=49 ttl=64 time=0.058 ms
64 bytes from 192.168.53.99: icmp_seq=50 ttl=64 time=0.050 ms
64 bytes from 192.168.53.99: icmp_seq=51 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=52 ttl=64 time=0.044 ms
64 bytes from 192.168.53.99: icmp_seq=53 ttl=64 time=0.044 ms
64 bytes from 192.168.53.99: icmp_seq=54 ttl=64 time=0.041 ms
64 bytes from 192.168.53.99: icmp_seq=55 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=56 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=57 ttl=64 time=0.050 ms
64 bytes from 192.168.53.99: icmp_seq=58 ttl=64 time=0.045 ms
64 bytes from 192.168.53.99: icmp_seq=59 ttl=64 time=0.045 ms
64 bytes from 192.168.53.99: icmp_seq=60 ttl=64 time=0.040 ms
64 bytes from 192.168.53.99: icmp_seq=61 ttl=64 time=0.041 ms
64 bytes from 192.168.53.99: icmp_seq=62 ttl=64 time=0.040 ms
64 bytes from 192.168.53.99: icmp_seq=63 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=64 ttl=64 time=0.050 ms
64 bytes from 192.168.53.99: icmp_seq=65 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=66 ttl=64 time=0.046 ms
64 bytes from 192.168.53.99: icmp_seq=67 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=68 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=69 ttl=64 time=0.044 ms
64 bytes from 192.168.53.99: icmp_seq=70 ttl=64 time=0.044 ms
64 bytes from 192.168.53.99: icmp_seq=71 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=72 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=73 ttl=64 time=0.041 ms
64 bytes from 192.168.53.99: icmp_seq=74 ttl=64 time=0.046 ms
64 bytes from 192.168.53.99: icmp_seq=75 ttl=64 time=0.185 ms
64 bytes from 192.168.53.99: icmp_seq=76 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=77 ttl=64 time=0.042 ms
64 bytes from 192.168.53.99: icmp_seq=78 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=79 ttl=64 time=0.044 ms
64 bytes from 192.168.53.99: icmp_seq=80 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=81 ttl=64 time=0.041 ms
64 bytes from 192.168.53.99: icmp_seq=82 ttl=64 time=0.039 ms
64 bytes from 192.168.53.99: icmp_seq=83 ttl=64 time=0.041 ms
64 bytes from 192.168.53.99: icmp_seq=84 ttl=64 time=0.043 ms
64 bytes from 192.168.53.99: icmp_seq=85 ttl=64 time=0.041 ms
^C
--- 192.168.53.99 ping statistics ---
85 packets transmitted, 57 received, 32.9412% packet loss, time 8
6006ms
rtt min/avg/max/mdev = 0.035/0.054/0.560/0.070 ms
root@6f6270cce0eb:/# 
```

```
root@6f6270cce0eb:/volumes# ls
tun.py  vpn_task2.py
root@6f6270cce0eb:/volumes# python3 vpn_task2.py
Interface Name: tun0
^CTraceback (most recent call last):
  File "vpn_task2.py", line 27, in <module>
    packet = os.read(tun, 2048)
KeyboardInterrupt

root@6f6270cce0eb:/volumes# 
```

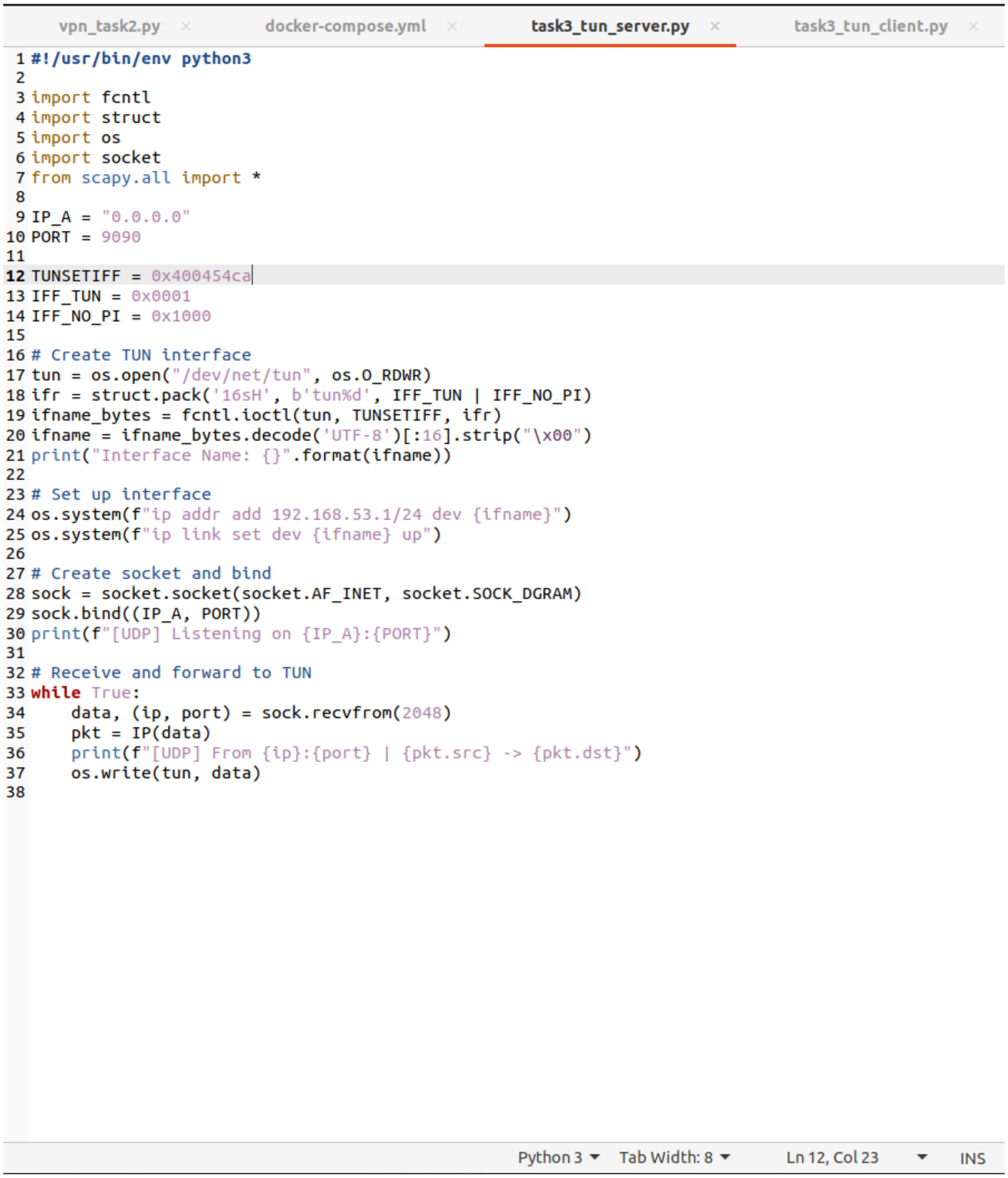
Task-03: Send the IP Packets to VPN Server Through a Tunnel

I created two programs ‘task3_tun_server.py’ and ‘task3_tun_client.py’ one that handles the server side script and the other handling the client side script. On the right I’m executing the task3_tun_server.py and task3_tun_client.py script in /volumes. On the left I’m pinging 192.168.60.5. You can see that this is entirely unidirectional and doesn’t go to the server side with 100% packet loss on the ping side.

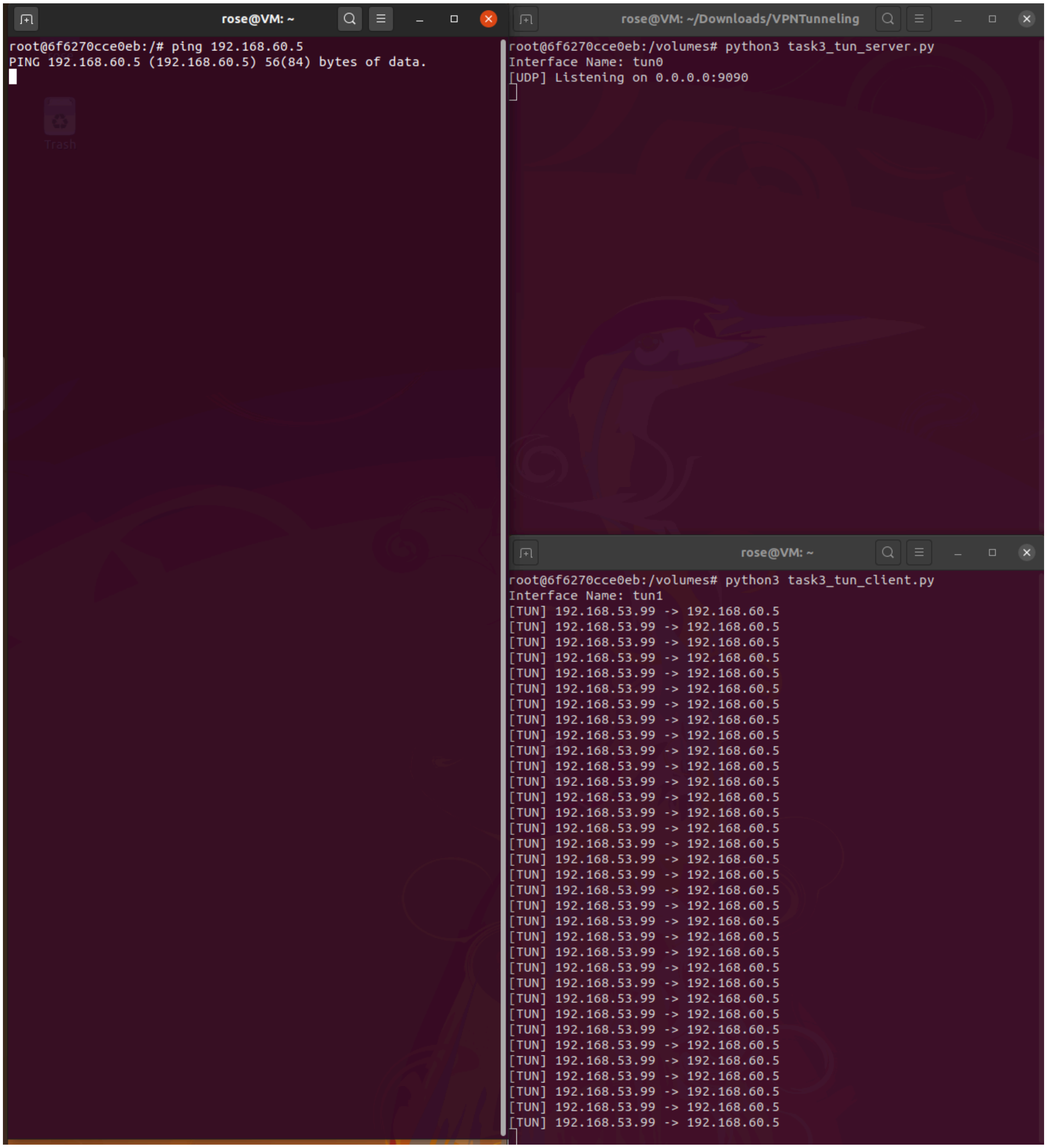
Observations/Issues: Stuff to fix in Task-04.



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

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

IP_A = "0.0.0.0"
PORT = 9090
SERVER_IP = "10.9.0.11"  # IP of the VPN server (Router container)

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

# Create TUN interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up interface and routing
os.system(f"ip addr add 192.168.53.99/24 dev {ifname}")
os.system(f"ip link set dev {ifname} up")
os.system(f"ip route add 192.168.60.0/24 dev {ifname}")

# Create socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Read packets from TUN and send over socket
while True:
    packet = os.read(tun, 2048)
    pkt = IP(packet)
    print(f"[TUN] {pkt.src} -> {pkt.dst}")
    sock.sendto(packet, (SERVER_IP, PORT))

```



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

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

IP_A = "0.0.0.0"
PORT = 9090

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

# Create TUN interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'tun%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

# Set up interface
os.system(f"ip addr add 192.168.53.1/24 dev {ifname}")
os.system(f"ip link set dev {ifname} up")

# Create socket and bind
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((IP_A, PORT))
print(f"[UDP] Listening on {IP_A}:{PORT}")

# Receive and forward to TUN
while True:
    data, (ip, port) = sock.recvfrom(2048)
    pkt = IP(data)
    print(f"[UDP] From {ip}:{port} | {pkt.src} -> {pkt.dst}")
    os.write(tun, data)

```



```
root@6f6270cce0eb:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
```

```
root@6f6270cce0eb:/volumes# python3 task3_tun_server.py
Interface Name: tun0
[UDP] Listening on 0.0.0.0:9090
```

```
root@6f6270cce0eb:/volumes# python3 task3_tun_client.py
Interface Name: tun1
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
[TUN] 192.168.53.99 -> 192.168.60.5
```