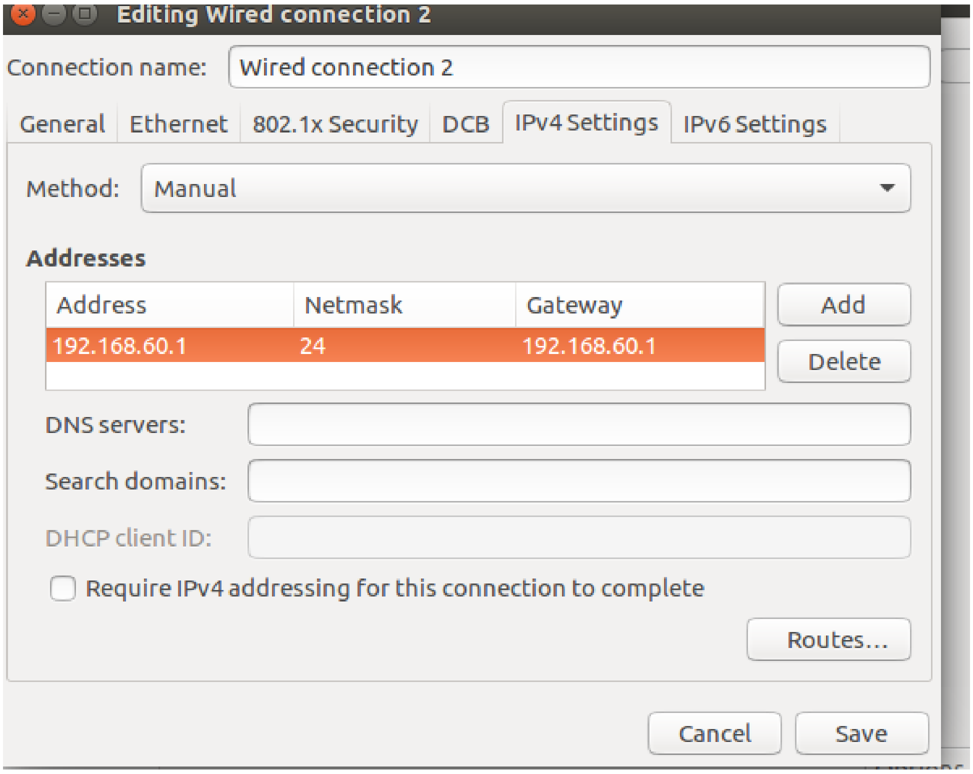
VPN Tunneling Lab

Task1: Network Setup

- **VPN server**



Connection name: Wired connection 2

Method: Manual

Addresses

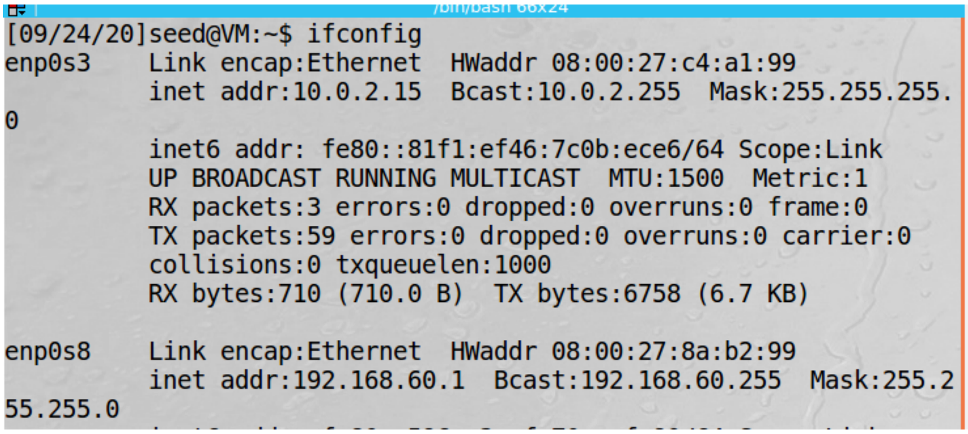
Address	Netmask	Gateway
192.168.60.1	24	192.168.60.1

DNS servers:

Search domains:

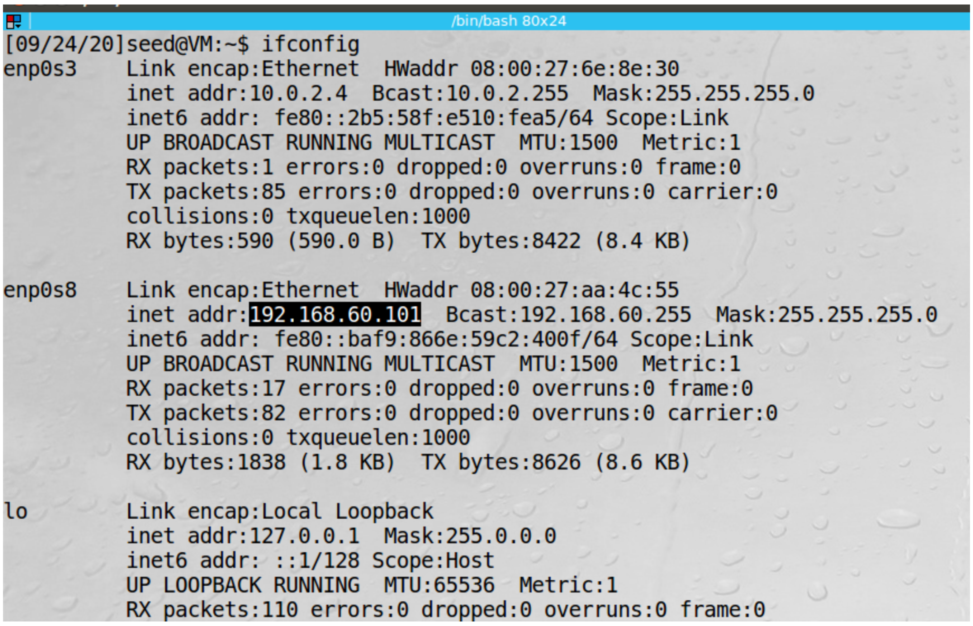
DHCP client ID:

□ Require IPv4 addressing for this connection to complete



```
[09/24/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:c4:a1:99
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.
0
          inet6 addr: fe80::81f1:ef46:7c0b:ece6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:59 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:710 (710.0 B)  TX bytes:6758 (6.7 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:8a:b2:99
          inet addr:192.168.60.1  Bcast:192.168.60.255  Mask:255.2
55.255.0
```

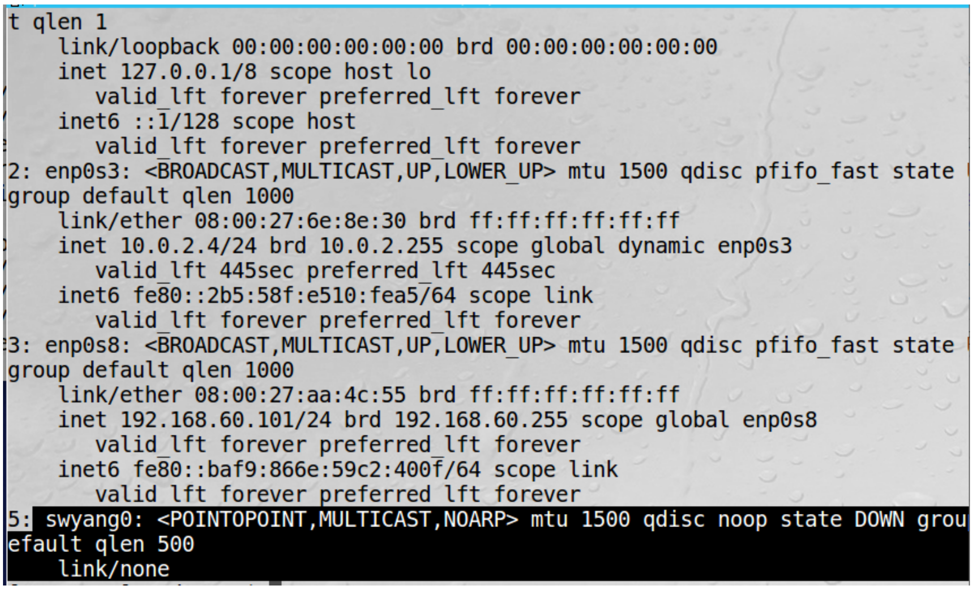
- **Host V**



```
[09/24/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:6e:8e:30
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::2b5:58f:e510:fea5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:85 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:590 (590.0 B)  TX bytes:8422 (8.4 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:aa:4c:55
          inet addr:192.168.60.101  Bcast:192.168.60.255  Mask:255.255.255.0
          inet6 addr: fe80::baf9:866e:59c2:400f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17 errors:0 dropped:0 overruns:0 frame:0
          TX packets:82 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1838 (1.8 KB)  TX bytes:8626 (8.6 KB)

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:110 errors:0 dropped:0 overruns:0 frame:0
```

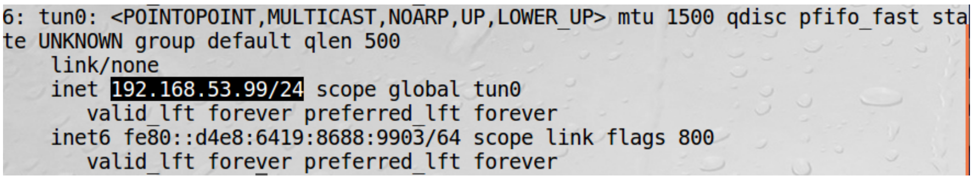
Task2: Create and Configure TUN Interface

- **taska**



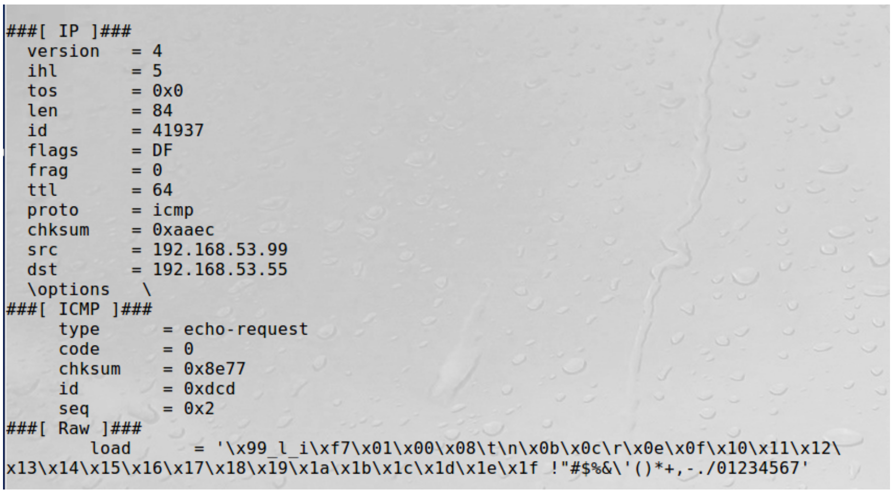
```
t qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
group default qlen 1000
    link/ether 08:00:27:6e:8e:30 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 445sec preferred_lft 445sec
    inet6 fe80::2b5:58f:e510:fea5/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
group default qlen 1000
    link/ether 08:00:27:aa:4c:55 brd ff:ff:ff:ff:ff:ff
    inet 192.168.60.101/24 brd 192.168.60.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::baf9:866e:59c2:400f/64 scope link
       valid_lft forever preferred_lft forever
5: swyang0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN grou
efault qlen 500
    link/none
```

- **taskb**



```
6: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast sta
te UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::d4e8:6419:8688:9903/64 scope link flags 800
       valid_lft forever preferred_lft forever
```

- **taskc**
    1. ping a host in network 192.168.53.0/24

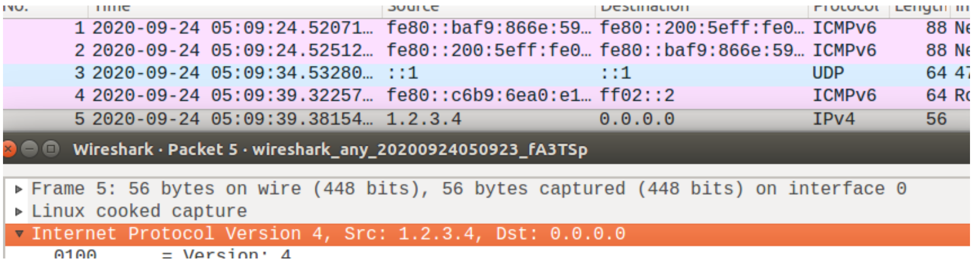


```
###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 84
  id        = 41937
  flags     = DF
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = 0xaaec
  src       = 192.168.53.99
  dst       = 192.168.53.55
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = 0x8e77
     id        = 0xdcd
     seq       = 0x2
###[ Raw ]###
        load      = '\x99_l_i\xf7\x01\x00\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\
x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'
```

2. ping a host in network 192.168.60.0/24

   print nothing

- **taskd**



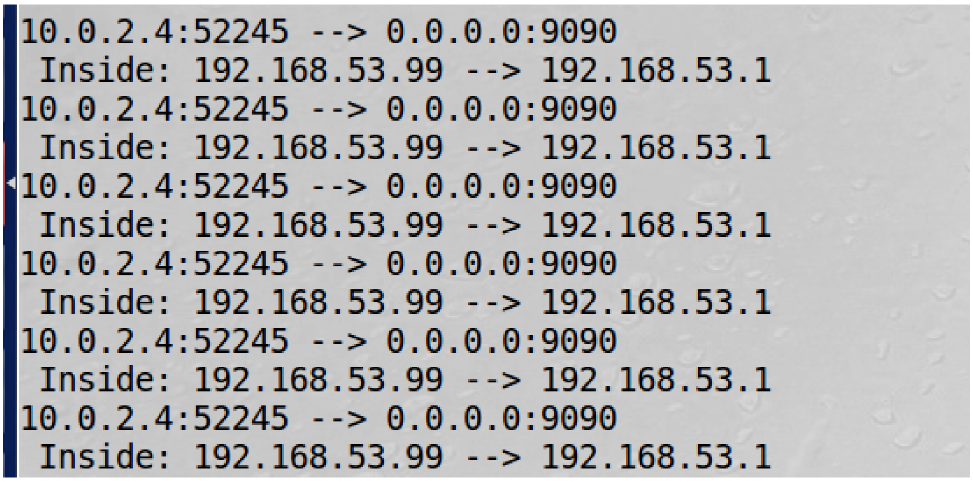
No.	Time	Source	Destination	Protocol	Length	In
1	2020-09-24 05:09:24.52071…	fe80::baf9:866e:59…	fe80::200:5eff:fe0…	ICMPv6	88	Ne
2	2020-09-24 05:09:24.52512…	fe80::200:5eff:fe0…	fe80::baf9:866e:59…	ICMPv6	88	Ne
3	2020-09-24 05:09:34.53280…	::1	::1	UDP	64	47
4	2020-09-24 05:09:39.32257…	fe80::c6b9:6ea0:e1…	ff02::2	ICMPv6	64	Ro
5	2020-09-24 05:09:39.38154…	1.2.3.4	0.0.0.0	IPv4	56	

Wireshark · Packet 5 · wireshark_any_20200924050923_fA3TSp

```
▸ Frame 5: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0
▸ Linux cooked capture
▾ Internet Protocol Version 4, Src: 1.2.3.4, Dst: 0.0.0.0
    0100 .... = Version: 4
```

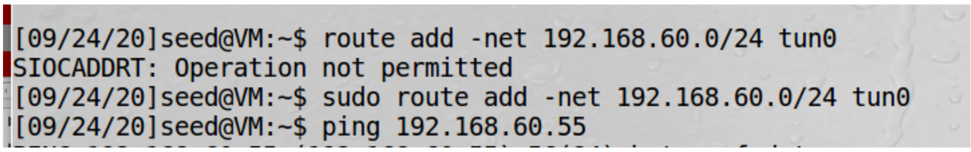
Task3:Send the IP Packet to VPN Server Through a Tunnel

1. ping any IP address belonging to the 192.168.53.0/24

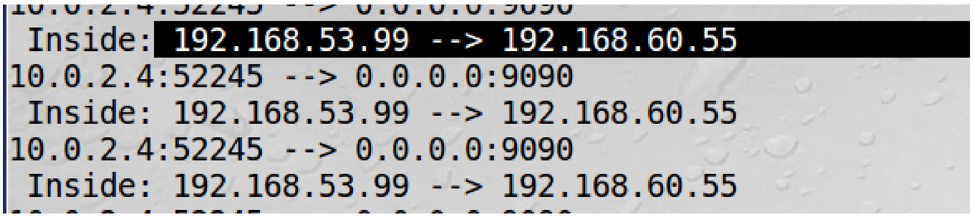


```
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.53.1
```

2. ping any ip address belonging to the 192.168.60.0/24

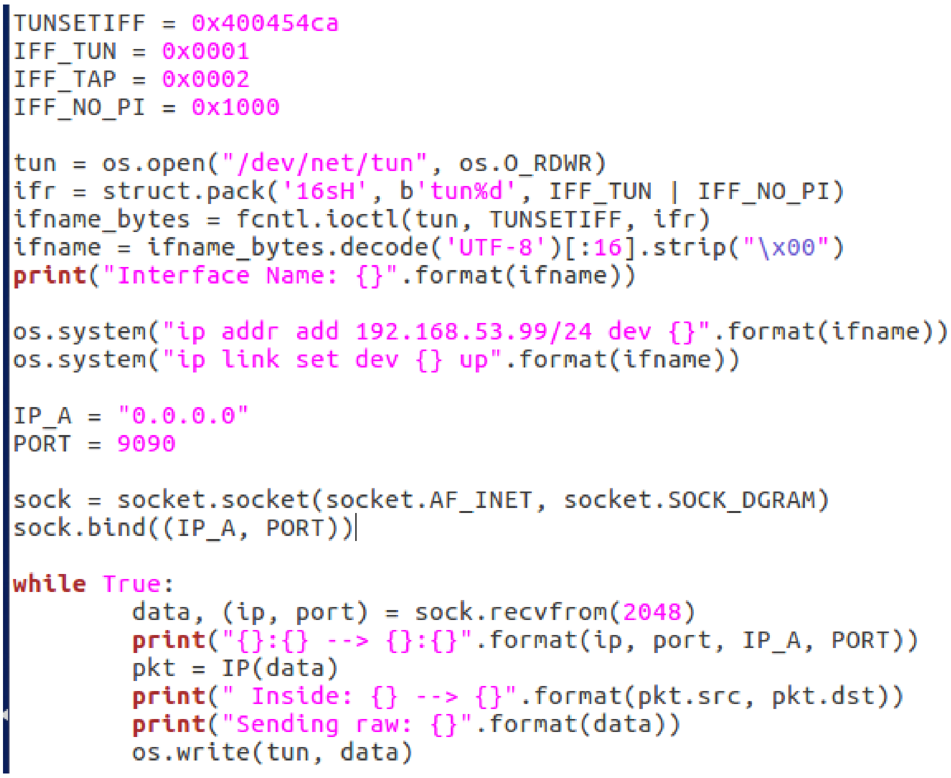


```
[09/24/20]seed@VM:~$ route add -net 192.168.60.0/24 tun0
SIOCADDRT: Operation not permitted
[09/24/20]seed@VM:~$ sudo route add -net 192.168.60.0/24 tun0
[09/24/20]seed@VM:~$ ping 192.168.60.55
```

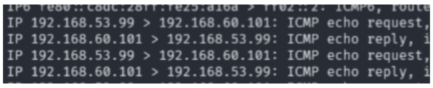


```
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.55
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.55
10.0.2.4:52245 --> 0.0.0.0:9090
 Inside: 192.168.53.99 --> 192.168.60.55
```

Task4: Set Up the VPN Server



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

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))

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

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))
        print("Sending raw: {}".format(data))
        os.write(tun, data)
```



```
IP6 fe80::c8dc:28ff:fe25:a16a > ff02::2: ICMP6, route
IP 192.168.53.99 > 192.168.60.101: ICMP echo request,
IP 192.168.60.101 > 192.168.53.99: ICMP echo reply, i
IP 192.168.53.99 > 192.168.60.101: ICMP echo request,
IP 192.168.60.101 > 192.168.53.99: ICMP echo reply, i
```

Task5: Handling Traffic in Both Directions

- **program**
  - **tun_server:**

```
#!/usr/bin/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

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))
```

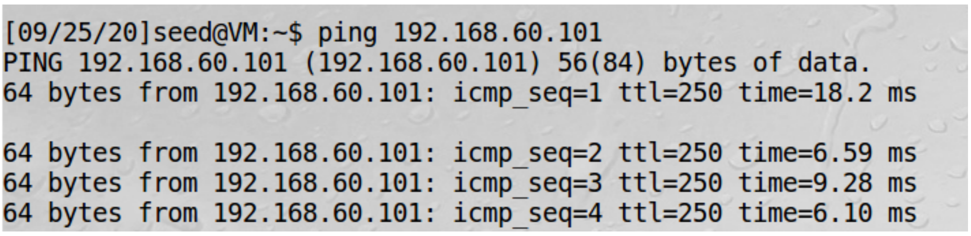
```
os.system("ip addr add 192.168.53.98/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

IP_A = "0.0.0.0"
PORT = 9090

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

while True:
    ready, _, _ = select.select([sock, tun], [], [])
    for fd in ready:
        if fd is sock:
            data, (ip, port) = sock.recvfrom(2048)
            pkt = IP(data)
            print("From socket <==: {} --> {}".format(pkt.src,
pkt.dst))
            os.write(tun, data)
        if fd is tun:
            packet = os.read(tun, 2048)
            pkt = IP(packet)
            print("From tun    ==>: {} --> {}".format(pkt.src,
pkt.dst))
            sock.sendto(packet, ("10.0.2.5", port))
```

- **tun_client**

```
#!/usr/bin/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)

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

#configure the interface
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
os.system("route add -net 192.168.60.0/24 {}".format(ifname))

print("Interface Name: {}".format(ifname))

# Create UDP socket
```

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)


while True:
    ready, _, _ = select.select([sock, tun], [], [])
    # Get a packet from the tun interface
    for fd in ready:
        if fd is sock:
            data, (ip, port) = sock.recvfrom(2048)
            pkt = IP(data)
            print("From socket <==: {} --> {}".format(pkt.src,
pkt.dst))
            os.write(tun, data)
        if fd is tun:
            packet = os.read(tun, 2048)
            pkt = IP(packet)
            print("From tun    ==>: {} --> {}".format(pkt.src,
pkt.dst))                        sock.sendto(packet, ("10.0.2.15",
9090))
```

- **machine U ping machine V**



```
[09/25/20]seed@VM:~$ ping 192.168.60.101
PING 192.168.60.101 (192.168.60.101) 56(84) bytes of data.
64 bytes from 192.168.60.101: icmp_seq=1 ttl=250 time=18.2 ms

64 bytes from 192.168.60.101: icmp_seq=2 ttl=250 time=6.59 ms
64 bytes from 192.168.60.101: icmp_seq=3 ttl=250 time=9.28 ms
64 bytes from 192.168.60.101: icmp_seq=4 ttl=250 time=6.10 ms
```

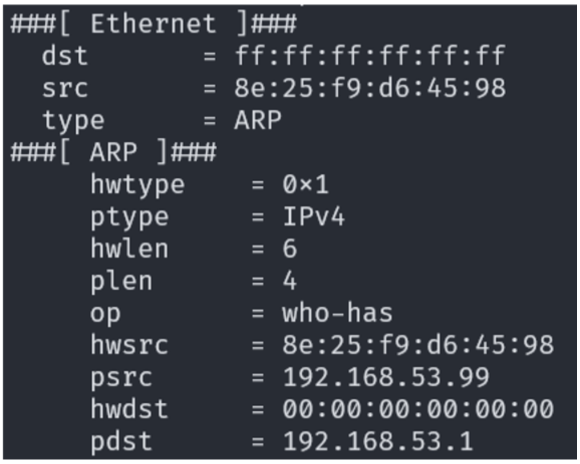
Task 6: Tunnel-Breaking Experiment

- when we stop run tun_client.py, the machine U print nothing. when we run tun_client.py again,we found that the character we type into Reappear

Task 8: Experiment with TUN IP Address

- when we change the tun's ip to 192.168.30.99. we can not make ping connection
- we found that the udp packet lose when it was sent to vpn server

Task 9: Experiment with the TAP Interface



```
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 8e:25:f9:d6:45:98
  type      = ARP
###[ ARP ]###
     hwtype    = 0×1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 8e:25:f9:d6:45:98
     psrc      = 192.168.53.99
     hwdst     = 00:00:00:00:00:00
     pdst      = 192.168.53.1
```