

Computer Network Security

Name	Naman Choudhary
SRN	PES2UG20CS209
Section	D

VPN Tunneling Lab

Task 1: Network Setup

On Client-10.9.0.5

```
ping server-router
```

Bash

```
#_ seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
root@Naman209-client:/# ping server-router
PING server-router (10.9.0.11) 56(84) bytes of data.
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=
1 ttl=64 time=7.68 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=
2 ttl=64 time=0.611 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=
3 ttl=64 time=0.557 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=
4 ttl=64 time=0.575 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=
5 ttl=64 time=0.528 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=
6 ttl=64 time=0.605 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=
7 ttl=64 time=0.574 ms
^C
--- server-router ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6028ms
rtt min/avg/max/mdev = 0.528/1.590/7.684/2.487 ms
root@Naman209-client:/#
```

In router

```
Bash
ping 192.168.60.5
```

The screenshot shows a terminal window titled "seed@Naman209: ~". The window has a menu bar with "File", "Actions", "Edit", "View", and "Help". Below the menu is a sub-menu window titled "seed@Naman209: ~" with a close button. The main terminal area displays the output of a "ping" command:

```
root@Naman209-router:/# 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=2.31 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.600 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.578 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=64 time=0.613 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=64 time=0.721 ms
^C
--- 192.168.60.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4029ms
rtt min/avg/max/mdev = 0.578/0.964/2.310/0.674 ms
root@Naman209-router:/#
```

On Client-10.9.0.5

The screenshot shows a terminal window on a client machine. The window title is "Bash". The input field contains the command "ping 192.168.60.5".

```
seed@Naman209: ~
PING server-router (10.9.0.11) 56(84) bytes of data.
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=1 ttl=64 time=7.68 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=2 ttl=64 time=0.611 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=3 ttl=64 time=0.578 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=4 ttl=64 time=0.613 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=5 ttl=64 time=0.721 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=6 ttl=64 time=0.575 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=7 ttl=64 time=0.528 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=8 ttl=64 time=0.605 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=9 ttl=64 time=0.574 ms
^C
--- server-router ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6028ms
rtt min/avg/max/mdev = 0.528/1.598/7.684/2.487 ms
root@Naman209-client:/# 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 ---
27 packets transmitted, 0 received, 100% packet loss, time 2602ms
root@Naman209-client:/#
```

```
root@Naman209-router:/# 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=2.31 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.600 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.578 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=64 time=0.613 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=64 time=0.721 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=64 time=0.575 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=64 time=0.528 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=64 time=0.605 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=64 time=0.574 ms
^C
--- 192.168.60.5 ping statistics ---
13:44:36.198312 IP6 fe80::b888:40ff:fe3c:9d2a.5353 > ff02::fb.5
7 packets transmitted, 5 received, 0% packet loss, time 4029ms
rtt min/avg/max/mdev = 0.578/0.964/2.310/0.674 ms
root@Naman209-router:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full prot
listening on eth0, link-type EN10MB (Ethernet), capture size 26
2144 bytes
al. (45)
13:44:36.198312 IP6 fe80::b888:40ff:fe3c:9d2a.5353: 0 [2q] PTR (QM)? _ipp._tcp.local. PTR (QM)? _ipp._tcp.loc
al. (45)
```

Bash

```
tcpdump -i eth0 -n
```

Bash

```
ping server-router
```

```
root@Naman209-client:/# ping server-router
PING server-router (10.9.0.11) 56(84) bytes of data.
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=1 ttl=64 time=2.39 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=2 ttl=64 time=0.625 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=3 ttl=64 time=0.583 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=4 ttl=64 time=0.706 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=5 ttl=64 time=0.541 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=6 ttl=64 time=0.536 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=7 ttl=64 time=0.775 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=8 ttl=64 time=0.694 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=9 ttl=64 time=0.684 ms
64 bytes from server-router.net-10.9.0.0 (10.9.0.11): icmp_seq=10 ttl=64 time=0.629 ms
root@Naman209-router:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full prot
listening on eth0, link-type EN10MB (Ethernet), capture size 26
13:45:21.337666 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 18
13:45:21.337992 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 18
13:45:22.339795 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 18
13:45:22.340629 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 18
13:45:23.341979 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 18
13:45:23.342215 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 18
13:45:24.345799 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 18
13:45:24.346010 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 18
```

Task 2: Create and Configure TUN Interface

Task 2.a: Name of the Interface

```
chmod a+x tun.py
./tun.py &
ip addr
```

Bash

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
root@Naman209-client:/volumes# ./tun.py &
[3] 35
root@Naman209-client:/volumes# Interface Name: tun2
root@Naman209-client:/volumes# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNK
NOWN 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
2: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop stat
e DOWN group default qlen 500
    link/none
3: tun1: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop stat
e DOWN group default qlen 500
    link/none
4: tun2: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop stat
e DOWN group default qlen 500
    link/none
76: eth0@if77: <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-net
nsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
root@Naman209-client:/volumes# 
```

Change tun to SRN

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
GNU nano 4.8          tun.py
#!/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'CS209\x0d', 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))

while True:
    [ Wrote 25 lines ]
^G Get Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
^X Exit      ^R Read File^\\ Replace   ^U Paste Tex^T To Spell
```

Bash

```
chmod a+x tun.py
./tun.py &
ip addr
```

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
[1]  Terminated      ./tun.py
root@Naman209-client:/volumes# ./tun.py &
[4] 42  I
root@Naman209-client:/volumes# Interface Name: CS2090

root@Naman209-client:/volumes# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNK
NOWN 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: tun1: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop stat
e DOWN group default qlen 500
    link/none
4: tun2: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop stat
e DOWN group default qlen 500
    link/none
5: CS2090: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop st
ate DOWN group default qlen 500
    link/none
76: eth0@if77: <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-net
nsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
root@Naman209-client:/volumes#
```

Task 2.b: Set up the TUN Interface

```
Bash
ip addr add 192.168.53.99/24 dev <SRN>0
ip link set dev <SRN>0 up
```

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
root@Naman209-client:/volumes# ip addr add 192.168.53.99/24 dev
CS2090
root@Naman209-client:/volumes# ip link set dev CS2090 up
root@Naman209-client:/volumes#
```

Task 2.c: Read from the TUN Interface

Replace code :

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

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
GNU nano 4.8          tun.py          Modified
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'CS209%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))

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

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

```
Bash
ip addr add 192.168.53.99/24 dev <SRN>0
ip link set dev <SRN>0 up
```

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
root@Naman209-client:/volumes# ip addr add 192.168.53.99/24 dev CS2090
root@Naman209-client:/volumes# ip link set dev CS2090 up
root@Naman209-client:/volumes# ./tun.py &
[2] 75
root@Naman209-client:/volumes# Interface Name: CS2091

root@Naman209-client:/volumes# ping 192.168.53.5
PING 192.168.53.5 (192.168.53.5) 56(84) bytes of data.
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
^C
--- 192.168.53.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3062ms

root@Naman209-client:/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 ---
11 packets transmitted, 0 received, 100% packet loss, time 10249ms

root@Naman209-client:/volumes#
```

Observation: Yes, `ping 192.168.53.5` prints out

`IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw`

No, `ping 192.168.60.5` loses all packets

Task 2.d: Write to the TUN Interface

Change tun to SRN

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
GNU nano 4.8          tun1.py
#!/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'CS209%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))
[ Read 44 lines ]
^G Get Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
^X Exit      ^R Read File^L Replace  ^U Paste Tex^T To Spell
```

```
Bash
chmod a+x tun.py
./tun1.py &
ip addr
```

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
root@Naman209-client:/volumes# chmod a+x tun1.py
root@Naman209-client:/volumes# ./tun1.py &
[1] 93
root@Naman209-client:/volumes# Interface Name: CS2090

root@Naman209-client:/volumes# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNK
NOWN 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
9: CS2090: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 q
disc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 scope global CS2090
        valid_lft forever preferred_lft forever
76: eth0@if77: <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-net
nsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
root@Naman209-client:/volumes#
```

I

ping 192.168.53.5

Bash

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
root@Naman209-client:/volumes# ping 192.168.53.5
PING 192.168.53.5 (192.168.53.5) 56(84) bytes of data.
CS2090: IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 /
Raw
64 bytes from 192.168.53.5: icmp_seq=1 ttl=99 time=34.9 ms
CS2090: IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 /
Raw
64 bytes from 192.168.53.5: icmp_seq=2 ttl=99 time=33.2 ms
CS2090: IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 /
Raw
64 bytes from 192.168.53.5: icmp_seq=3 ttl=99 time=48.8 ms
CS2090: IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 /
Raw
64 bytes from 192.168.53.5: icmp_seq=4 ttl=99 time=32.7 ms
CS2090: IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 /
Raw
64 bytes from 192.168.53.5: icmp_seq=5 ttl=99 time=25.8 ms
CS2090: IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 /
Raw
64 bytes from 192.168.53.5: icmp_seq=6 ttl=99 time=19.0 ms
^C
--- 192.168.53.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5011ms
rtt min/avg/max/mdev = 19.017/32.400/48.816/9.121 ms
root@Naman209-client:/volumes#
```

Task 3: Send the IP Packet to VPN Server Through a Tunnel

Change tun to SRN

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
GNU nano 4.8          tun_client.py      Modified
#!/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'CS209%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))

# Configure the interface

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

```
Bash
chmod a+x tun_server.py
./tun_server.py
```

```
Bash
chmod a+x tun_client.py
./tun_client.py &
ip addr
```

The screenshot shows a Lubuntu 20.04 desktop environment with a purple and blue gradient background. At the bottom is a dock with icons for a terminal, file manager, and other applications. Two terminal windows are open in the qterminal application. The left terminal window (root@Naman209-router) shows the command `ping 192.168.53.5` being run. The right terminal window (root@Naman209-client) shows the command `ping 192.168.60.5` being run. Both terminals show the output of their respective ping commands.

```
root@Naman209-router:/volumes# chmod a+x tun_server.py
root@Naman209-router:/volumes# ./tun_server.py
[1] 109
root@Naman209-client:/volumes# nano tun_client
root@Naman209-client:/volumes# chmod a+x tun_client.py
root@Naman209-client:/volumes# ./tun_client.py &
[1] 109
root@Naman209-client:/volumes# Interface Name: CS2090

root@Naman209-client:/volumes# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    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
10: CS2090: <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 CS2090
            valid_lft forever preferred_lft forever
76: eth0@f77: <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@Naman209-client:/volumes#
```

The screenshot shows a Lubuntu 20.04 desktop environment with a purple and blue gradient background. At the bottom is a dock with icons for a terminal, file manager, and other applications. A single terminal window (root@Naman209) is open in the qterminal application. It shows the command `ping 192.168.53.5` being run. The window has a "Bash" tab at the top right.

```
ping 192.168.53.5
ping 192.168.60.5
```

The screenshot shows a Lubuntu 20.04 desktop environment with a purple and blue gradient background. At the bottom is a dock with icons for a terminal, file manager, and other applications. Two terminal windows are open in the qterminal application. The left terminal window (root@Naman209-router) shows the command `ping 192.168.53.5` being run. The right terminal window (root@Naman209-client) shows the command `ping 192.168.60.5` being run. Both terminals show the output of their respective ping commands.

```
root@Naman209-router:/volumes# chmod a+x tun_server.py
root@Naman209-router:/volumes# ./tun_server.py
[1] 109
root@Naman209-client:/volumes# ping 192.168.53.5
PING 192.168.53.5 (192.168.53.5) 56(84) bytes of data.
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.5 echo-request 0 / Raw
^C
--- 192.168.53.5 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4093ms
root@Naman209-client:/volumes# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
^C
--- 192.168.60.5 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4096ms
```

Bash

```
ip route
```

```
root@Naman209-client:/volumes# 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 CS2090 proto kernel scope link src 192.168.
53.99
192.168.60.0/24 dev CS2090 scope link
root@Naman209-client:/volumes#
```

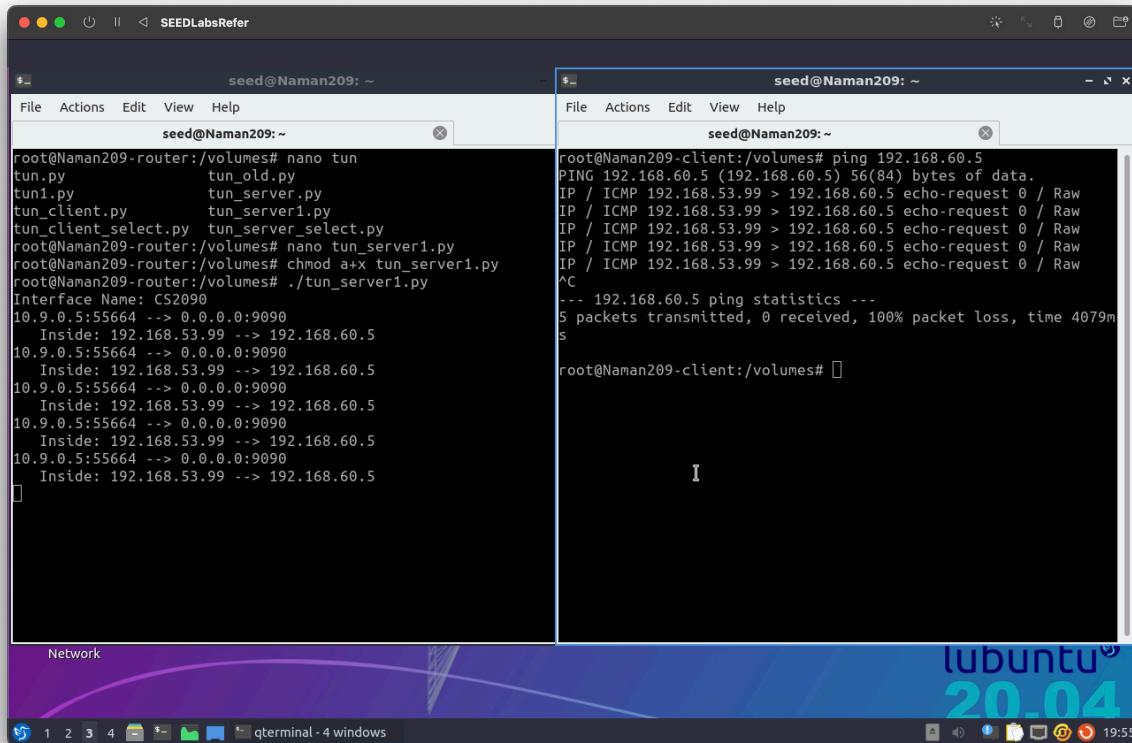
Task 4: Set Up the VPN Server

Bash

```
chmod a+x tun_server1.py
./tun_server1.py
```

Bash

```
ping 192.168.60.5
```



Bash

```
tcpdump -i eth0 -n
```

```
$ seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
14:24:56.447032 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 125, seq 1, length 64
14:24:57.446097 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 125, seq 2, length 64
14:24:57.446356 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 125, seq 2, length 64
14:24:58.473410 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 125, seq 3, length 64
14:24:58.473738 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 125, seq 3, length 64
14:24:59.497700 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 125, seq 4, length 64
14:24:59.497947 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 125, seq 4, length 64
14:25:00.520313 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 125, seq 5, length 64
14:25:00.520538 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 125, seq 5, length 64
14:25:01.529644 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
14:25:01.530101 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
14:25:01.530122 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
14:25:01.530162 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
```

Task 5: Handling Traffic in Both Directions

Change tun to SRN

```
seed@Naman209: ~
File Actions Edit View Help
seed@Naman209: ~
GNU nano 4.8 tun_server_select.py
#!/usr/bin/python3

#import select
import fcntl
import struct
import os
from scapy.all import *
IP_A = "0.0.0.0"
PORT = 9090

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

# Create a tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'CS209\x00', 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))

[G Get Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
^X Exit      ^R Read File^V Replace ^U Paste Tex^I To Spell

[ Read 51 lines ]
et Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
xit      ^R Read File^V Replace ^U Paste Tex^I To Spell

seed@Naman209: ~
Actions Edit View Help
seed@Naman209: ~
GNU nano 4.8 tun_client_select.py
sr/bin/python3

rt fcntl
rt struct
rt os
scapy.all import *

ETIFF = 0x400454ca
TUN   = 0x0001
TAP   = 0x0002
NO_PI = 0x1000

eate a tun interface
= os.open("/dev/net/tun", os.O_RDWR)
= struct.pack('16sH', b'CS209\x00', IFF_TUN | IFF_NO_PI)
me_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
me = ifname_bytes.decode("UTF-8")[:16].strip("\x00")
t("Interface Name: {}".format(ifname))

t up the tun interface and routing
ystem("ip addr add 192.168.53.99/24 dev {}".format(ifname))
ystem("ip link set dev {} up".format(ifname))

[Wrote 46 lines]
et Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
xit      ^R Read File^V Replace ^U Paste Tex^I To Spell
```

Bash

```
chmod a+x tun_client_select.py
./tun_client_select.py
```

Bash

```
ping 192.168.60.5
```

The screenshot shows two terminal windows side-by-side. The left window displays the output of the command `./tun_client_select.py`, which lists various network connections between interfaces like tun and socket. The right window shows the output of the command `ping 192.168.60.5`, displaying ping statistics for 6 packets transmitted.

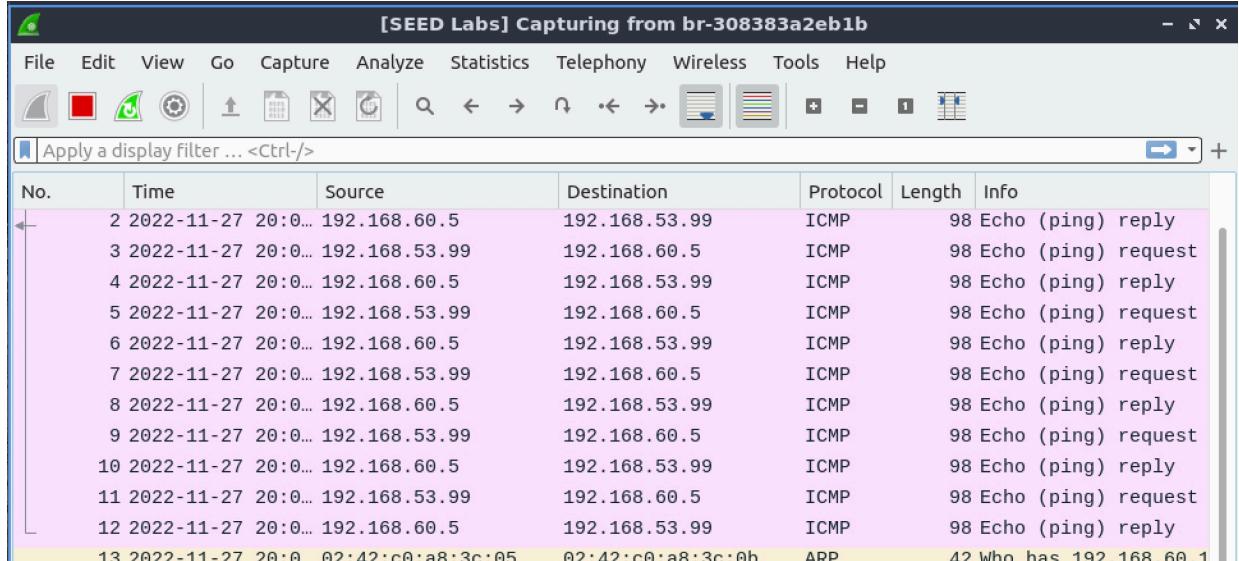
```
root@Naman209-client:/volumes# ./tun_client_select.py
Interface Name: CS2090
From tun ==>: 192.168.53.99 -> 192.168.60.5
From socket <==: 192.168.60.5 -> 192.168.53.99
From tun ==>: 192.168.53.99 -> 192.168.60.5
From socket <==: 192.168.60.5 -> 192.168.53.99
From tun ==>: 192.168.53.99 -> 192.168.60.5
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
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
[...]

```

```
root@Naman209-client:/volumes# 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=63 time=33.7 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=26.5 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=21.1 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=18.5 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=18.7 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=24.7 ms
^C
--- 192.168.60.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5012ms
rtt min/avg/max/mdev = 18.475/23.884/33.725/5.289 ms
root@Naman209-client:/volumes#
```

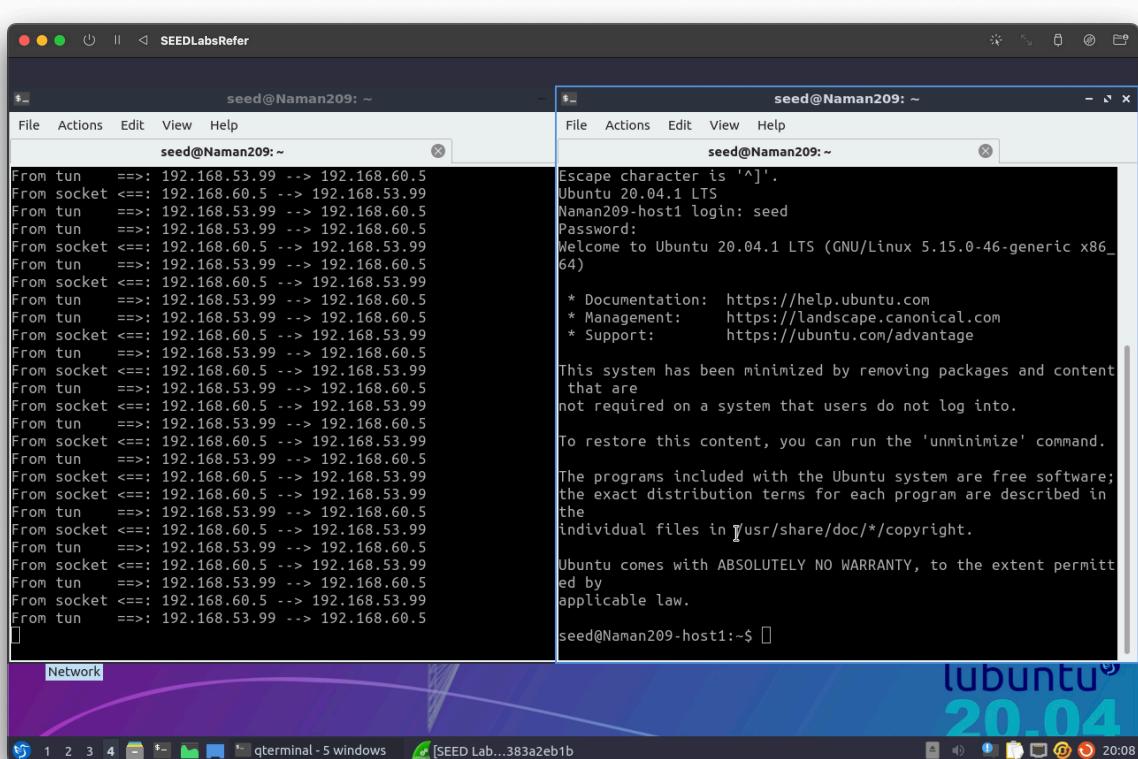
```
chmod a+x tun_server_select.py  
./tun_server_select.py
```

Wireshark:



telnet 192.168.60.5

Bash



Wireshark:

[SEED Labs] Capturing from br-308383a2eb1b

No.	Time	Source	Destination	Protocol	Length	Info
49	2022-11-27 20:00...	192.168.60.5	192.168.53.99	TCP	66	23 → 33260 [ACK] Seq
50	2022-11-27 20:00...	192.168.53.99	192.168.60.5	TELNET	67	Telnet Data ...
51	2022-11-27 20:00...	192.168.60.5	192.168.53.99	TCP	66	23 → 33260 [ACK] Seq
52	2022-11-27 20:00...	192.168.53.99	192.168.60.5	TELNET	68	Telnet Data ...
53	2022-11-27 20:00...	192.168.60.5	192.168.53.99	TCP	66	23 → 33260 [ACK] Seq
54	2022-11-27 20:00...	192.168.60.5	192.168.53.99	TELNET	68	Telnet Data ...
55	2022-11-27 20:00...	192.168.53.99	192.168.60.5	TCP	66	33260 → 23 [ACK] Seq
56	2022-11-27 20:00...	192.168.60.5	192.168.53.99	TELNET	477	Telnet Data ...
57	2022-11-27 20:00...	192.168.53.99	192.168.60.5	TCP	66	33260 → 23 [ACK] Seq
58	2022-11-27 20:00...	192.168.60.5	192.168.53.99	TELNET	341	Telnet Data ...
59	2022-11-27 20:00...	192.168.53.99	192.168.60.5	TCP	66	33260 → 23 [ACK] Seq
60	2022-11-27 20:00...	192.168.60.5	192.168.53.99	TELNET	89	Telnet Data ...
61	2022-11-27 20:00...	192.168.53.99	192.168.60.5	TCP	66	33260 → 23 [ACK] Seq
62	2022-11-27 20:00... 02:42:c0:a8:3c:0b		02:42:c0:a8:3c:05	ARP	42	Who has 192.168.60.5
63	2022-11-27 20:00... 02:42:c0:a8:3c:0b		02:42:c0:a8:3c:05	ARP	42	192.168.60.5 is at 0

Task 6: Tunnel-Breaking Experiment

Bash
telnet 192.168.60.5

and then breaking the connection off [last 7 lines]

```

seed@Naman209: ~
$ nano tun_server_select.py
#!/usr/bin/python3

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

IP_A = "0.0.0.0"
PORT = 9090

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

# Create a tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'CS209\0', 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))

[ Read 51 lines ]
^G Get Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File^U Replace ^U Paste Tex^T To Spell

```

```

seed@Naman209: ~
$ nano tun_client_select.py
#!/usr/bin/python3

rt fcntl
rt struct
rt os
scapy.all import *

ETIFF = 0x400454ca
TUN = 0x0001
TAP = 0x0002
NO_PI = 0x1000

#create a tun interface
= os.open("/dev/net/tun", os.O_RDWR)
= struct.pack('16sH', b'CS209\0', IFF_TUN | IFF_NO_PI)
me_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
me = ifname_bytes.decode('UTF-8')[16:].strip("\x00")
t("Interface Name: {}".format(ifname))

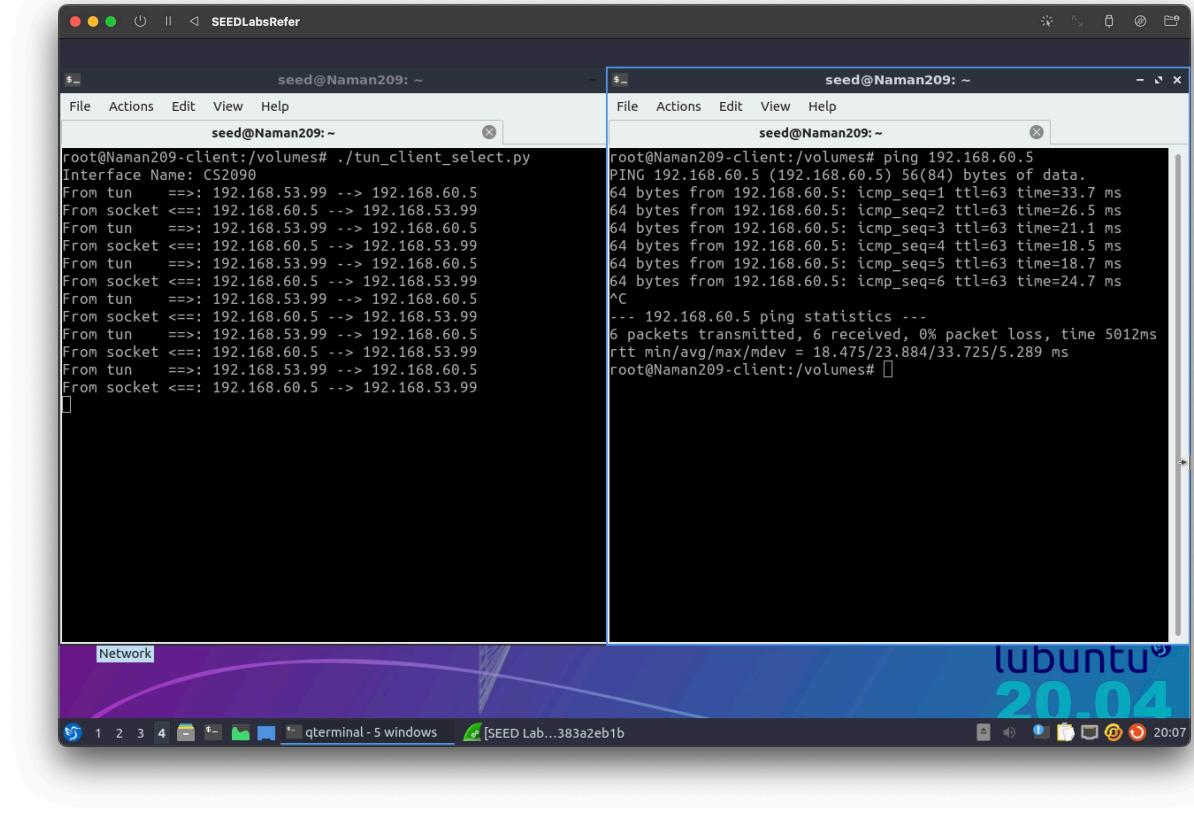
t up the tun interface and routing
system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
system("ip link set dev {} up".format(ifname))

[ Wrote 46 lines ]
et Help ^O Write Out^W Where Is ^K Cut Text ^J Justify
xit ^R Read File^U Replace ^U Paste Tex^T To Spell

```

Bringing connection back

```
./tun_server_select.py
```



The screenshot shows a Wireshark interface capturing traffic from a bridge interface. The packet list pane displays 13 captured frames. The first 12 frames are ICMP Echo requests (Type 8, Code 0) sent from 192.168.60.5 to 192.168.53.99, with sizes ranging from 64 to 68 bytes. The 13th frame is an ARP request (Type 1, Code 1) sent from 02:42:60:08:3c:05 to 02:42:60:a8:3c:0b, asking for the MAC address of 192.168.60.1.

No.	Time	Source	Destination	Protocol	Length	Info
2	2022-11-27 20:0...	192.168.60.5	192.168.53.99	ICMP	98	Echo (ping) reply
3	2022-11-27 20:0...	192.168.53.99	192.168.60.5	ICMP	98	Echo (ping) request
4	2022-11-27 20:0...	192.168.60.5	192.168.53.99	ICMP	98	Echo (ping) reply
5	2022-11-27 20:0...	192.168.53.99	192.168.60.5	ICMP	98	Echo (ping) request
6	2022-11-27 20:0...	192.168.60.5	192.168.53.99	ICMP	98	Echo (ping) reply
7	2022-11-27 20:0...	192.168.53.99	192.168.60.5	ICMP	98	Echo (ping) request
8	2022-11-27 20:0...	192.168.60.5	192.168.53.99	ICMP	98	Echo (ping) reply
9	2022-11-27 20:0...	192.168.53.99	192.168.60.5	ICMP	98	Echo (ping) request
10	2022-11-27 20:0...	192.168.60.5	192.168.53.99	ICMP	98	Echo (ping) reply
11	2022-11-27 20:0...	192.168.53.99	192.168.60.5	ICMP	98	Echo (ping) request
12	2022-11-27 20:0...	192.168.60.5	192.168.53.99	ICMP	98	Echo (ping) reply
13	2022-11-27 20:0:02:42:60:08:3c:05		02:42:60:a8:3c:0b	ARP	42	Who has 192.168.60.1