

Question 1:

```
(noe@njt0060)-[~/hw2]
$ python3 mycode.py
###[ IP ]###
version      = 4
ihl          = None
tos          = 0x0
len          = None
id           = 1
flags        =
frag         = 0
ttl          = 64
proto        = hopopt
chksum       = None
src          = 127.0.0.1
dst          = 127.0.0.1
\options     \

(noe@njt0060)-[~/hw2]
$ s
```

Figure 1

Figure 1 is the results from the current state of my mycode.py code

```
[noe@njt0060]-[~/hw2]
$ python3 sniffer.py
Traceback (most recent call last):
  File "/home/noe/hw2/sniffer.py", line 6, in <module>
    sniff(prn = display_pkt)
  File "/usr/lib/python3/dist-packages/scapy/sendrecv.py", line 1424, in sniff
    sniffer._run(*args, **kwargs)
  File "/usr/lib/python3/dist-packages/scapy/sendrecv.py", line 1273, in _run
    sniff_sockets[_RL2(iface)(type=ETH_P_ALL, iface=iface,
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/scapy/arch/linux/__init__.py", line 218, in __init__
    self.ins = socket.socket(
^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.11/socket.py", line 232, in __init__
    _socket.socket.__init__(self, family, type, proto, fileno)
PermissionError: [Errno 1] Operation not permitted

[noe@njt0060]-[~/hw2]
$
```

Figure 2 is the failure of question 2, since we are not using sudo privileges. In the next photo, we will see how when you use sudo, it will work correctly.

Question 3:

[illegible]

Figure 3

Figure 3 output of my sniffer script, this is the tail of it, after opening unt.edu.

Question 4:

```
(noe@njt0000) ~/hw2
└─$ sudo python3 sniffer.py
####[ Ethernet ]####
dst      = 80:78:71:00:83:15
src      = 08:00:27:a3:c0:ea
type     = IPv4
####[ IP ]####
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 2279
flags    = DF
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0x3a4e
src      = 192.168.1.52
dst      = 20.225.32.183
\options \
####[ ICMP ]####
type     = echo-request
code     = 0
chksum   = 0x21aa
id       = 0x3c5e
seq      = 0x1
unused   = b''
####[ Raw ]####
load     = b'D9<g\x00\x00\x00\x00H\x83\x0c\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#%&'()*+,-./01234567'

####[ Ethernet ]####
dst      = 08:00:27:a3:c0:ea
src      = 80:78:71:00:83:15
type     = IPv4
####[ IP ]####
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 0
flags    = DF
frag     = 0
ttl      = 113
proto    = icmp
chksum   = 0x1235
src      = 20.225.32.183
dst      = 192.168.1.52
\options \
####[ ICMP ]####
```

Figure 4

Figure 4 this is the sniffer with the ICMP filter after pinging www.unt.edu

Question 5:

```
(noe@njt0060)-[~/hw2]
$ sudo python3 spoof.py
Sent 1 packets.
(noe@njt0060)-[~/hw2]
$
```

Figure 5a

Figure 5a is the spoof script sending 1 packet.

```
noe@njt0060: ~/hw2
(noe@njt0060)-[~/hw2]
$ sudo python3 sniffer.py
###[ Ethernet ]###
dst      = 00:00:00:00:00:00
src      = 00:00:00:00:00:00
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 28
id       = 1
flags    = 
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0x9280
src      = 129.120.231.230
dst      = 127.0.0.1
\options
###[ ICMP ]###
type     = echo-request
code     = 0
chksum   = 0xf7ff
id       = 0x0
seq      = 0x0
unused   = b''
###[ Ethernet ]###
dst      = 00:00:00:00:00:00
src      = 00:00:00:00:00:00
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 28
id       = 1
flags    = 
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0x9280
src      = 129.120.231.230
dst      = 127.0.0.1
\options
###[ ICMP ]###
type     = echo-request
code     = 0
chksum   = 0xf7ff
id       = 0x0
seq      = 0x0
unused   = b''
```

Figure 5b

Figure 5b is the sniffer script capturing the ICMP echo request.

Question 6:

```
>>> ip=IP(dst="127.0.0.1")
...: syn=TCP(dport=22, sport= 45600, flags="S", seq=100)
...: resp=srl((ip/syn), timeout=10)
Begin emission

Finished sending 1 packets

Received 2 packets, got 1 answers, remaining 0 packets
>>> resp.show()
###[ IP ]###
  version    = 4
  ihl        = 5
  tos        = 0x0
  len        = 40
  id         = 0
  flags      = DF
  frag       = 0
  ttl        = 64
  proto      = tcp
  chksum     = 0x3cce
  src        = 127.0.0.1
  dst        = 127.0.0.1
  \options   \
###[ TCP ]###
  sport      = ssh
  dport      = 45600
  seq        = 0
  ack        = 101
  dataofs    = 5
  reserved   = 0
  flags      = RA
  window     = 0
  chksum     = 0xff32
  urgptr     = 0
  options    = []

>>> exit()

(noel@njt0060)-[~/hw2]
```

Figure 6

Figure 6 shows a tcp response with a ack number of 101, but a flag of RA, which means reset acknowledgement. This means the ssh port was pinged, but decided to reject the connection.

Question 7:

```
(noe@njt0060)-[~/hw2]
$ sudo python3 dns.py
Begin emission
Finished sending 1 packets
.*
Received 3 packets, got 1 answers, remaining 0 packets
###[ DNS ]###
  id      = 0
  qr      = 1
  opcode  = QUERY
  aa      = 0
  tc      = 0
  rd      = 1
  ra      = 1
  z       = 0
  ad      = 0
  cd      = 0
  rcode   = ok
  qdcount = 1
  ancount = 2
  nscount = 0
  arcount = 0
  \qd     \
  |###[ DNS Question Record ]###
  | qname  = b'www.unt.edu.'
  | qtype  = A
  | unicastresponse= 0
  | qclass = IN
  \an     \
  |###[ DNS Resource Record ]###
  | rrname = b'www.unt.edu.'
  | type   = CNAME
  | cacheflush= 0
  | rclass = IN
  | ttl    = 300
  | rdlen  = None
  | rdata  = b'prd.farm01.azure.cws.vip.dtc.untsystem.edu.'
  |###[ DNS Resource Record ]###
  | rrname = b'prd.farm01.azure.cws.vip.dtc.untsystem.edu.'
  | type   = A
  | cacheflush= 0
  | rclass = IN
  | ttl    = 5
  | rdlen  = None
  | rdata  = 20.225.32.183
  \ns     \
  \ar     \
None

(noe@njt0060)-[~/hw2]
$ ping www.unt.edu
PING prd.farm01.azure.cws.vip.dtc.untsystem.edu (20.225.32.183) 56(84) bytes of data:
64 bytes from 20.225.32.183: icmp_seq=1 ttl=113 time=31.5 ms
```

Figure 7a

In figure 7a I am verifying the ip address of unt via the ping command in the second command.

```
ns.py mycode.py sniffer.py spoof.py

(noe@njt0060)-[~/hw2]
$ ip route | grep default
default via 192.168.1.1 dev eth0 proto dhcp src 192.168.1.52 metric 100

(noe@njt0060)-[~/hw2]
```

Figure 7b

In Figure 7b, I am verifying that my own ip address is also the one that the professor provided, 192.168.1.1.