Seth Pennebaker
Tom Barrett
Kevin Hearty

Sniff and Spoof Lab

Task 1

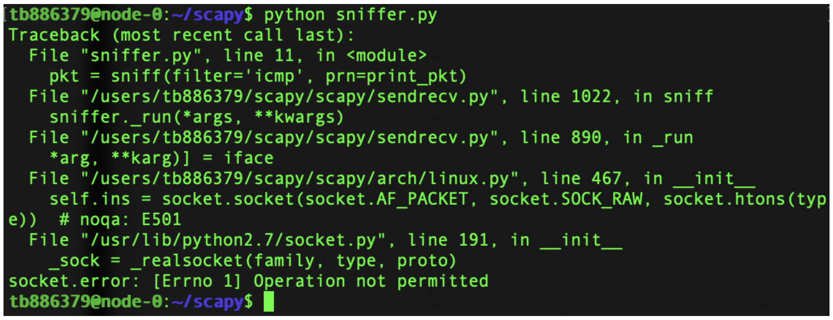
```
tb886379@node-0:~/scapy$ sudo python mycode.py
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = hopopt
  chksum    = None
  src       = 127.0.0.1
  dst       = 127.0.0.1
  \options   \
```

Task 1.1A

Sudo python sniffer.py

```
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 36
     id        = 10691
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x6846
     src       = 130.127.132.211
     dst       = 13.52.212.73
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0x4924
        id        = 0x15
        seq       = 0x221a
###[ Raw ]###
           load      = '\x15\xd4\xeb\x188\x87[8'
```

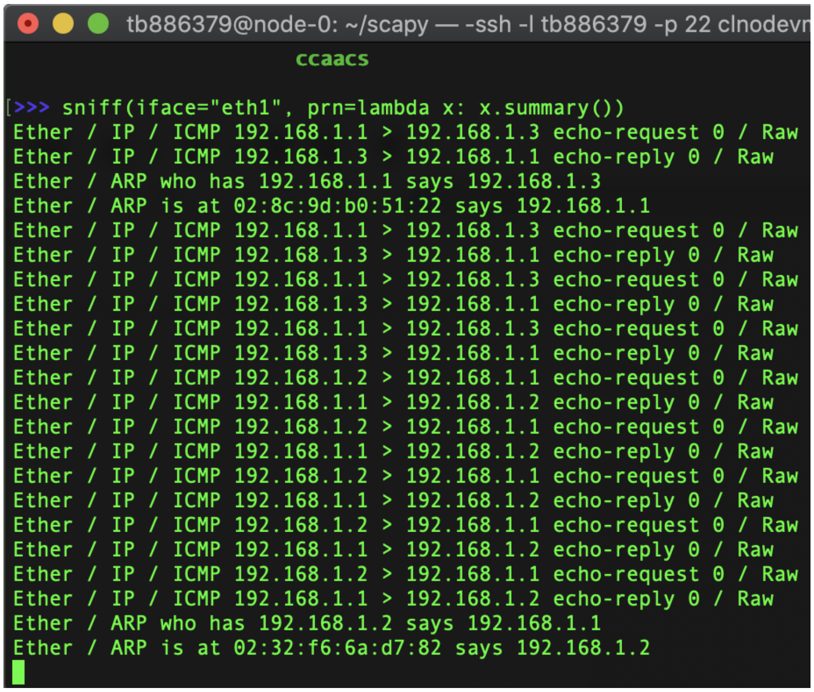
python sniffer.py



```
[tb886379@node-0:~/scapy$ python sniffer.py
Traceback (most recent call last):
  File "sniffer.py", line 11, in <module>
    pkt = sniff(filter='icmp', prn=print_pkt)
  File "/users/tb886379/scapy/scapy/sendrecv.py", line 1022, in sniff
    sniffer._run(*args, **kwargs)
  File "/users/tb886379/scapy/scapy/sendrecv.py", line 890, in _run
    *arg, **karg)] = iface
  File "/users/tb886379/scapy/scapy/arch/linux.py", line 467, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(typ
e))  # noqa: E501
  File "/usr/lib/python2.7/socket.py", line 191, in __init__
    _sock = _realsocket(family, type, proto)
socket.error: [Errno 1] Operation not permitted
tb886379@node-0:~/scapy$
```

<u>Task 1.1B</u>

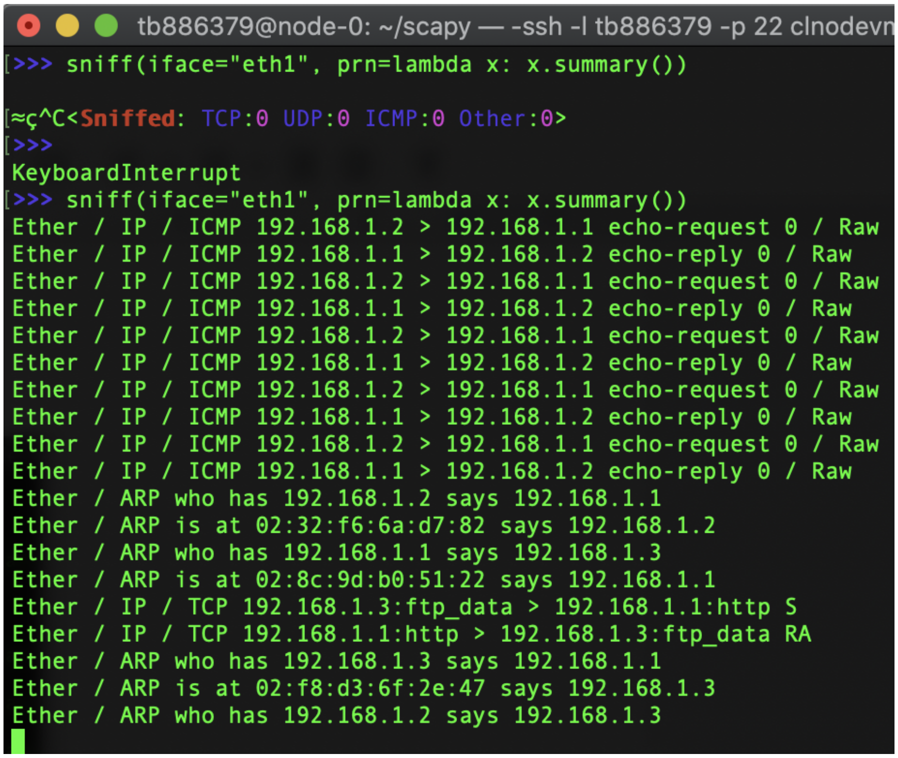
Only ICMP Packet



```
ccaacs

[>>> sniff(iface="eth1", prn=lambda x: x.summary())
Ether / IP / ICMP 192.168.1.1 > 192.168.1.3 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.3 > 192.168.1.1 echo-reply 0 / Raw
Ether / ARP who has 192.168.1.1 says 192.168.1.3
Ether / ARP is at 02:8c:9d:b0:51:22 says 192.168.1.1
Ether / IP / ICMP 192.168.1.1 > 192.168.1.3 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.3 > 192.168.1.1 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.1.1 > 192.168.1.3 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.3 > 192.168.1.1 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.1.1 > 192.168.1.3 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.3 > 192.168.1.1 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
Ether / ARP who has 192.168.1.2 says 192.168.1.1
Ether / ARP is at 02:32:f6:6a:d7:82 says 192.168.1.2
```

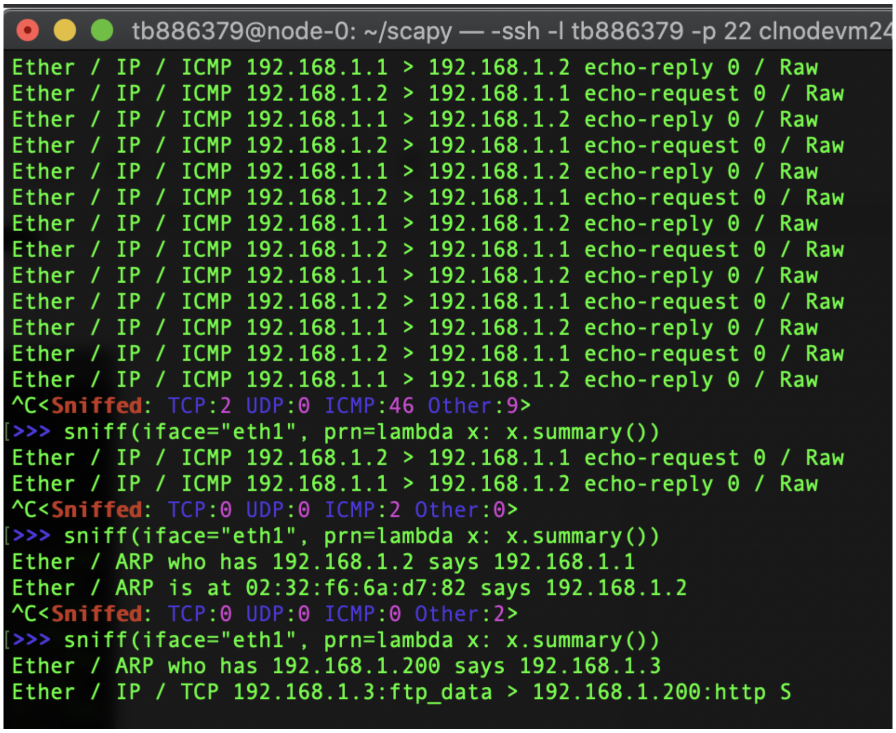
TCP on Port 23



```
[>>> sniff(iface="eth1", prn=lambda x: x.summary())

[≈ç^C<Sniffed: TCP:0 UDP:0 ICMP:0 Other:0>
[>>>
 KeyboardInterrupt
[>>> sniff(iface="eth1", prn=lambda x: x.summary())
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / ARP who has 192.168.1.2 says 192.168.1.1
 Ether / ARP is at 02:32:f6:6a:d7:82 says 192.168.1.2
 Ether / ARP who has 192.168.1.1 says 192.168.1.3
 Ether / ARP is at 02:8c:9d:b0:51:22 says 192.168.1.1
 Ether / IP / TCP 192.168.1.3:ftp_data > 192.168.1.1:http S
 Ether / IP / TCP 192.168.1.1:http > 192.168.1.3:ftp_data RA
 Ether / ARP who has 192.168.1.3 says 192.168.1.1
 Ether / ARP is at 02:f8:d3:6f:2e:47 says 192.168.1.3
 Ether / ARP who has 192.168.1.2 says 192.168.1.3
```

Capture packets comes from or to go to a particular subnet.



```
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 ^C<Sniffed: TCP:2 UDP:0 ICMP:46 Other:9>
[>>> sniff(iface="eth1", prn=lambda x: x.summary())
 Ether / IP / ICMP 192.168.1.2 > 192.168.1.1 echo-request 0 / Raw
 Ether / IP / ICMP 192.168.1.1 > 192.168.1.2 echo-reply 0 / Raw
 ^C<Sniffed: TCP:0 UDP:0 ICMP:2 Other:0>
[>>> sniff(iface="eth1", prn=lambda x: x.summary())
 Ether / ARP who has 192.168.1.2 says 192.168.1.1
 Ether / ARP is at 02:32:f6:6a:d7:82 says 192.168.1.2
 ^C<Sniffed: TCP:0 UDP:0 ICMP:0 Other:2>
[>>> sniff(iface="eth1", prn=lambda x: x.summary())
 Ether / ARP who has 192.168.1.200 says 192.168.1.3
 Ether / IP / TCP 192.168.1.3:ftp_data > 192.168.1.200:http S
```

Task 1.2

sp876427@node-1: ~/scapy

```
                     aSPY//YASa
             apyyyyCY//////////YCa       |
            sY//////YSpcs  scpCY//Pp     | Welcome to Scapy
 ayp ayyyyyyySCP//Pp           syY//C    | Version 2.4.3.dev161
 AYAsAYYYYYYYY///Ps              cY//S   |
         pCCCCY//p          cSSps y//Y   | https://github.com/secdev/scapy
         SPPPP///a          pP///AC//Y   |
              A//A            cyP////C   | Have fun!
              p///Ac            sC///a   |
              P////YCpc           A//A   | We are in France, we say Skappee.
       scccccp///pSP///p          p//Y   | OK? Merci.
      sY/////////y  caa           S//P   |             -- Sebastien Chabal
       cayCyayP//Ya              pY/Ya   |
        sY/PsY////YCc           aC//Yp
         sc  sccaCY//PCypaapyCP//YSs
                  spCPY//////YPSps
                       ccaacs

>>> from scapy.all import *
>>> a = IP()
>>> a.dst = '130.127.132.227'
>>> b = ICMP()
>>> p = a/b
>>> send(p)
.
Sent 1 packets.
>>>
[8]+  Stopped                 sudo ./run_scapy
sp876427@node-1:~/scapy$
```

sp876427@node-2: ~/scapy

```
        seq       = 0x0

###[ Ethernet ]###
  dst       = 02:67:4b:15:1e:0d
  src       = 02:70:fe:f1:bd:8c
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 30568
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0xf4c2
     src       = 130.127.132.227
     dst       = 130.127.132.212
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xffff
        id        = 0x0
        seq       = 0x0

^Z
[12]+  Stopped                 sudo python sniffer.py
sp876427@node-2:~/scapy$
```

1.3 Traceroute

```
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 1
     proto     = icmp
     chksum    = 0xab2a
     src       = 130.127.132.227
     dst       = 130.127.132.212
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xf7ff
        id        = 0x0
        seq       = 0x0

###[ Ethernet ]###
  dst       = 02:70:fe:f1:bd:8c
  src       = 02:67:4b:15:1e:0d
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 16723
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x2ad8
     src       = 130.127.132.212
     dst       = 130.127.132.227
```

```
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 2
     proto     = icmp
     chksum    = 0xaa2a
     src       = 130.127.132.227
     dst       = 130.127.132.212
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xf7ff
        id        = 0x0
        seq       = 0x0

###[ Ethernet ]###
  dst       = 02:70:fe:f1:bd:8c
  src       = 02:67:4b:15:1e:0d
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 16724
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x2ad7
     src       = 130.127.132.212
     dst       = 130.127.132.227
```

```
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 3
     proto     = icmp
     chksum    = 0xa92a
     src       = 130.127.132.227
     dst       = 130.127.132.212
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xf7ff
        id        = 0x0
        seq       = 0x0

###[ Ethernet ]###
  dst       = 02:70:fe:f1:bd:8c
  src       = 02:67:4b:15:1e:0d
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 16725
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x2ad6
     src       = 130.127.132.212
     dst       = 130.127.132.227
```

```
  GNU nano 2.5.3                File: tra

from scapy.all import *
a = IP()
a.dst = '130.127.132.212'
b = ICMP()

for x in range(3):
        a.ttl = (x+1)
        sr1(a/b)
```

Task 1.4

```
.
Sent 1 packets.
###[ Ethernet ]###
  dst       = 02:67:4b:15:1e:0d
  src       = 02:70:fe:f1:bd:8c
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x6c2a
     src       = 130.127.132.227
     dst       = 130.127.132.212
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xf7ff
        id        = 0x0
        seq       = 0x0
```

```
Sent 1 packets.
###[ Ethernet ]###
  dst       = 02:70:fe:f1:bd:8c
  src       = 02:67:4b:15:1e:0d
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 28
     id        = 21487
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x183c
     src       = 130.127.132.212
     dst       = 130.127.132.227
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xffff
        id        = 0x0
        seq       = 0x0

.
Sent 1 packets.
```

```
  GNU nano 2.5.3            File: sniffspoof.py

from scapy.all import *

def print_pkt(pkt):
        a = IP()
        a.dst = '130.127.132.212'
        b = ICMP()
        p = a/b
        send(p)
        pkt.show()


pkt = sniff(filter='icmp',prn=print_pkt)
```

Task 2.1A

```
sp876427@node-2: ~/scapy
   pcap_compile(handle, &fp, filter_exp, 0, net);
   ^
sp876427@node-2:~/scapy$ nano sniff.c
sp876427@node-2:~/scapy$ sp876427@node-2:~/scapy$
sp876427@node-2:~/scapy$
sp876427@node-2:~/scapy$ gcc -o sniff sniff.c -lpcap
sp876427@node-2:~/scapy$ ./sniff
Segmentation fault
sp876427@node-2:~/scapy$ nano sniff.c
sp876427@node-2:~/scapy$ sp876427@node-2:~/scapy$
sp876427@node-2:~/scapy$
sp876427@node-2:~/scapy$
sp876427@node-2:~/scapy$
sp876427@node-2:~/scapy$
sp876427@node-2:~/scapy$
sp876427@node-2:~/scapy$ sudo ./sniff
Got a packet
Got a packet
Got a packet
Got a packet
Got a packet
Got a packet
Got a packet
```

Question 1

pcap_lookupdev: Finds a capture device to sniff on
pcap_lookupnet: Returns the network number and mask for the capture device
pcap_open_live: Starts sniffing on the capture device
pcap_datalink: Returns the kind of device we're capturing on
Pcap_compile: Compiles the filter from a string
pcap_setfilter: Sets the compiled filter
Pcap_next: Sniff one packet at a time
Pcap_loop: Sniffs packets continuously.
pcap_freecode: Frees up allocated memory generated by pcap_compile
pcap_close: Closes the sniffing session

Question 2

Pcap_open_live is our issue for not being able to run the program without root. To be able to run sniffer programs you must have root access to the NIC.

Quesiton 3

char filter_exp[] = "icmp and (src host 192.168.0.38 and dst host 8.8.8.8) or (src host 8.8.8.8 and dst host 192.168.0.38)"