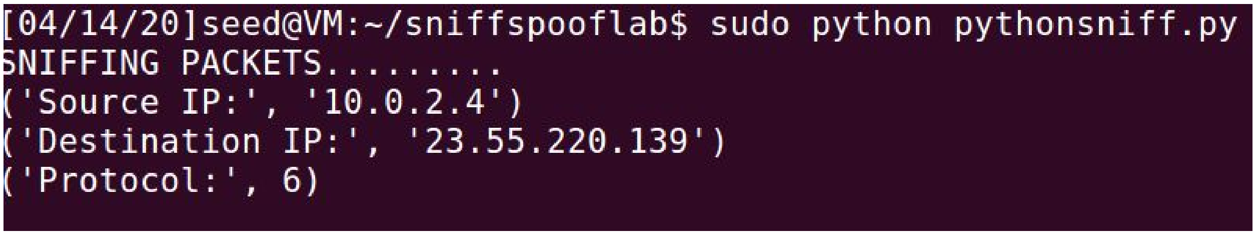
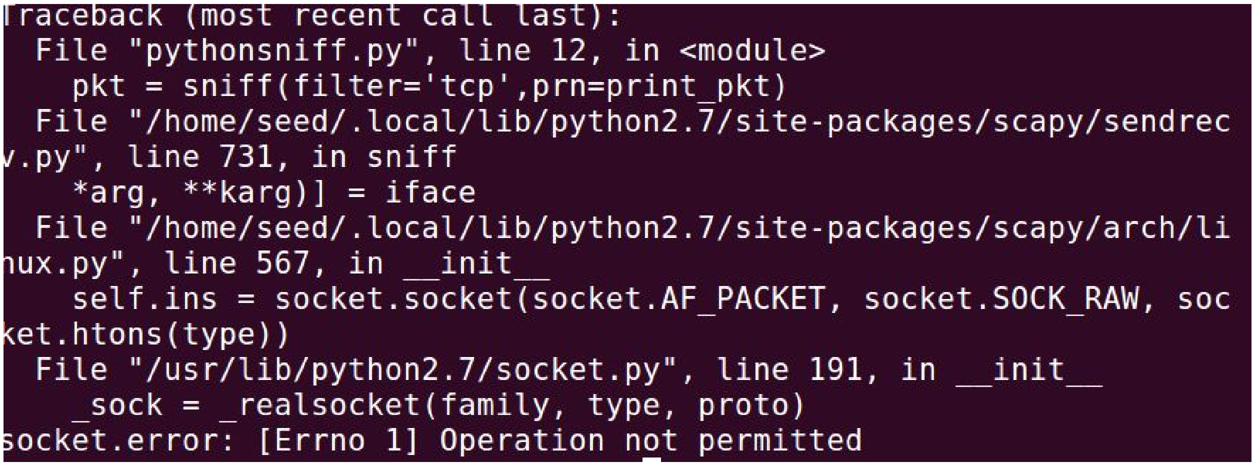
Packet sniffing lab
Jared Ren
4/13/20

Task 1.1A)



```
[04/14/20]seed@VM:~/sniffspooflab$ sudo python pythonsniff.py
SNIFFING PACKETS.........
('Source IP:', '10.0.2.4')
('Destination IP:', '23.55.220.139')
('Protocol:', 6)
```

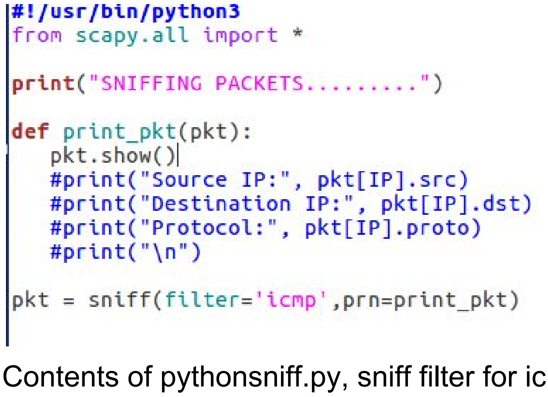
Packet has been sniffed



```
Traceback (most recent call last):
  File "pythonsniff.py", line 12, in <module>
    pkt = sniff(filter='tcp',prn=print_pkt)
  File "/home/seed/.local/lib/python2.7/site-packages/scapy/sendrec
v.py", line 731, in sniff
    *arg, **karg)] = iface
  File "/home/seed/.local/lib/python2.7/site-packages/scapy/arch/li
nux.py", line 567, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, soc
ket.htons(type))
  File "/usr/lib/python2.7/socket.py", line 191, in __init__
    _sock = _realsocket(family, type, proto)
socket.error: [Errno 1] Operation not permitted
```

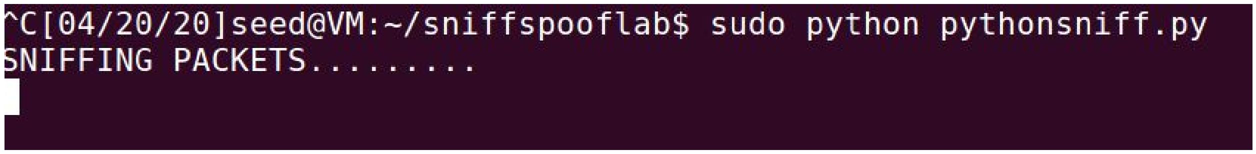
Error when running without the sudo keyword.
The reason why it did not work is because root privilege is needed to put the NIC into promiscuous mode

Task1.1B)



```
#!/usr/bin/python3
from scapy.all import *

print("SNIFFING PACKETS.........")

def print_pkt(pkt):
    pkt.show()
    #print("Source IP:", pkt[IP].src)
    #print("Destination IP:", pkt[IP].dst)
    #print("Protocol:", pkt[IP].proto)
    #print("\n")

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

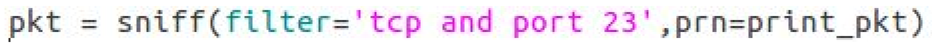
Contents of pythonsniff.py, sniff filter for icmp applied.



```
^C[04/20/20]seed@VM:~/sniffspooflab$ sudo python pythonsniff.py
SNIFFING PACKETS.........
```

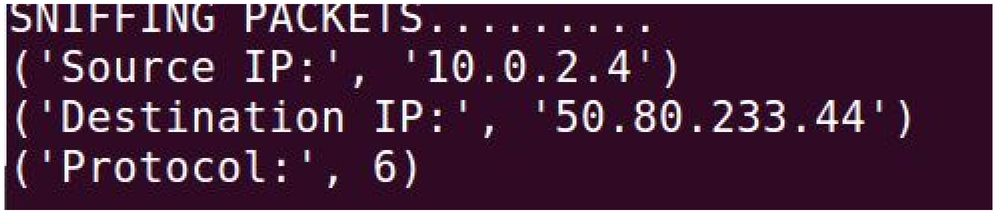
Result of running pythonsniff.py

Bullet 2.



```
pkt = sniff(filter='tcp and port 23',prn=print_pkt)
```

Filter applied to pythonsniff.py



```
SNIFFING PACKETS.........
('Source IP:', '10.0.2.4')
('Destination IP:', '50.80.233.44')
('Protocol:', 6)
```

Packet sniffed.

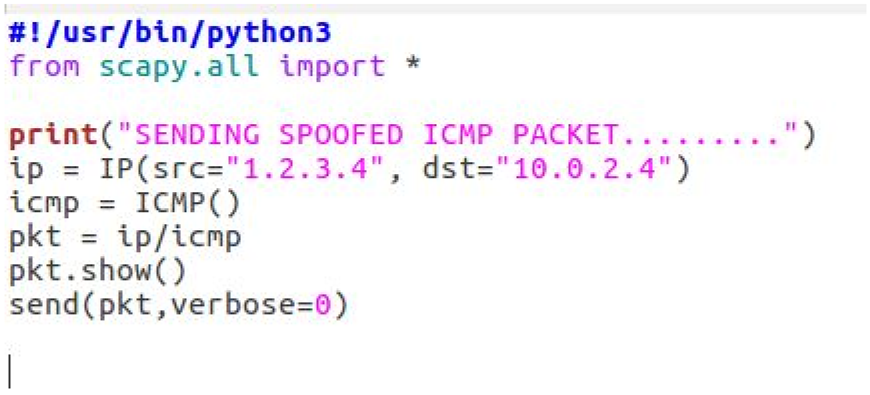
Bullet 3

159.65.249.155

Central web server IP

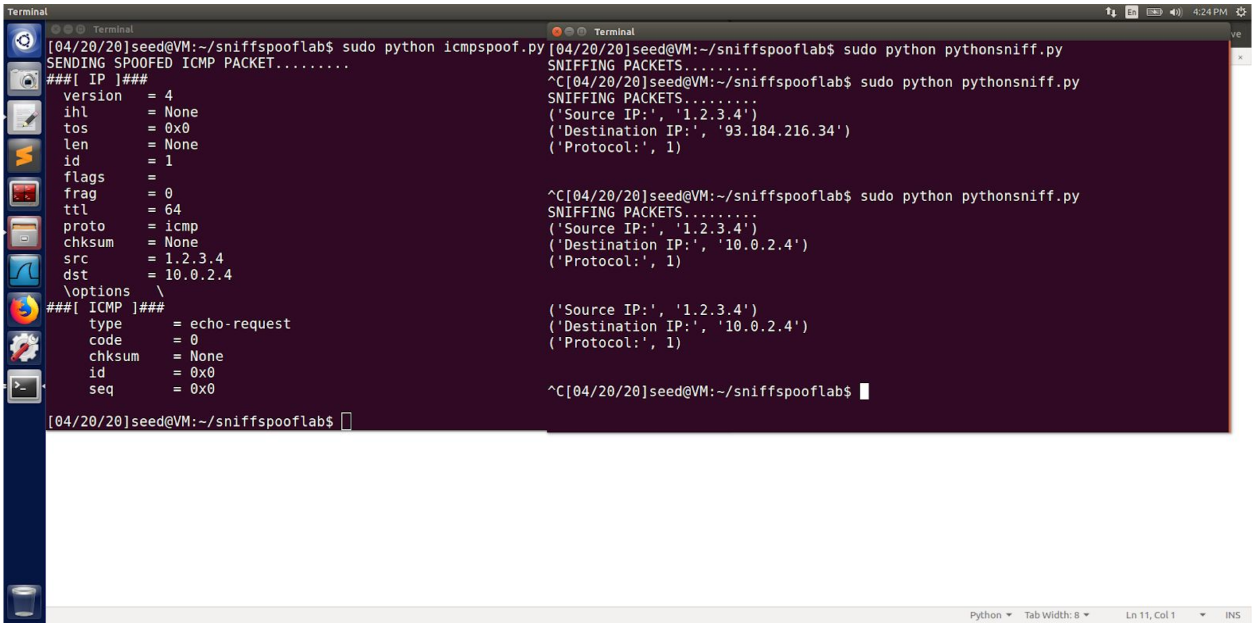
Subnet 249

1.2)



```
#!/usr/bin/python3
from scapy.all import *

print("SENDING SPOOFED ICMP PACKET.........")
ip = IP(src="1.2.3.4", dst="10.0.2.4")
icmp = ICMP()
pkt = ip/icmp
pkt.show()
send(pkt,verbose=0)
```

Spoofing program

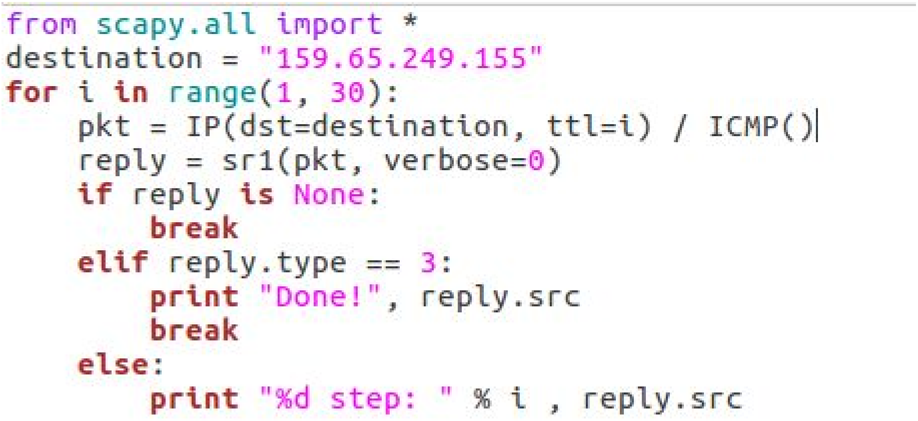


```
[04/20/20]seed@VM:~/sniffspooflab$ sudo python icmpspoof.py
SENDING SPOOFED ICMP PACKET.........
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = None
  src       = 1.2.3.4
  dst       = 10.0.2.4
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

[04/20/20]seed@VM:~/sniffspooflab$
```

```
[04/20/20]seed@VM:~/sniffspooflab$ sudo python pythonsniff.py
SNIFFING PACKETS.........
^C[04/20/20]seed@VM:~/sniffspooflab$ sudo python pythonsniff.py
SNIFFING PACKETS.........
('Source IP:', '1.2.3.4')
('Destination IP:', '93.184.216.34')
('Protocol:', 1)


^C[04/20/20]seed@VM:~/sniffspooflab$ sudo python pythonsniff.py
SNIFFING PACKETS.........
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.4')
('Protocol:', 1)


('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.4')
('Protocol:', 1)


^C[04/20/20]seed@VM:~/sniffspooflab$
```

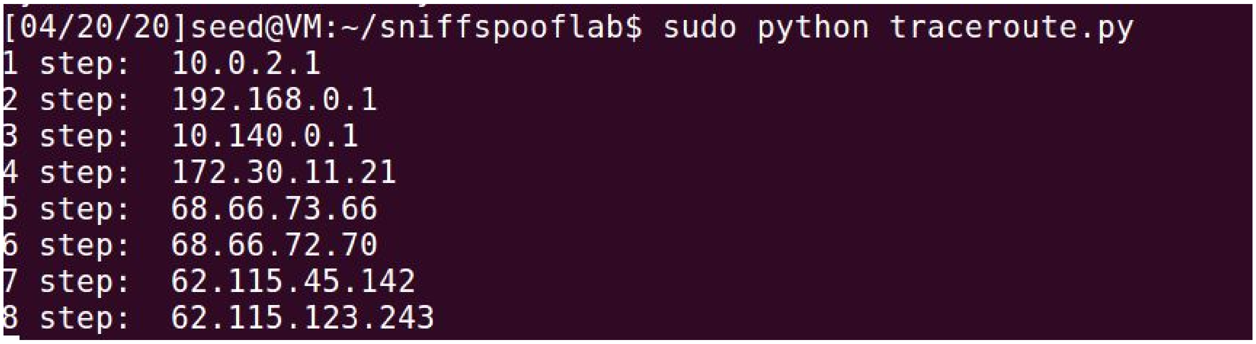
Both windows, left is spoof right is sniff.

1.3)



```
from scapy.all import *
destination = "159.65.249.155"
for i in range(1, 30):
    pkt = IP(dst=destination, ttl=i) / ICMP()
    reply = sr1(pkt, verbose=0)
    if reply is None:
        break
    elif reply.type == 3:
        print "Done!", reply.src
        break
    else:
        print "%d step: " % i , reply.src
```

Python traceroute program



```
[04/20/20]seed@VM:~/sniffspooflab$ sudo python traceroute.py
1 step:  10.0.2.1
2 step:  192.168.0.1
3 step:  10.140.0.1
4 step:  172.30.11.21
5 step:  68.66.73.66
6 step:  68.66.72.70
7 step:  62.115.45.142
8 step:  62.115.123.243
```

8 routers between me and central.edu

--EXERCISES---

15.13)

This is not fake news. ARP poisoning attacks are when attackers deliberately send fake ARP messages on a LAN.