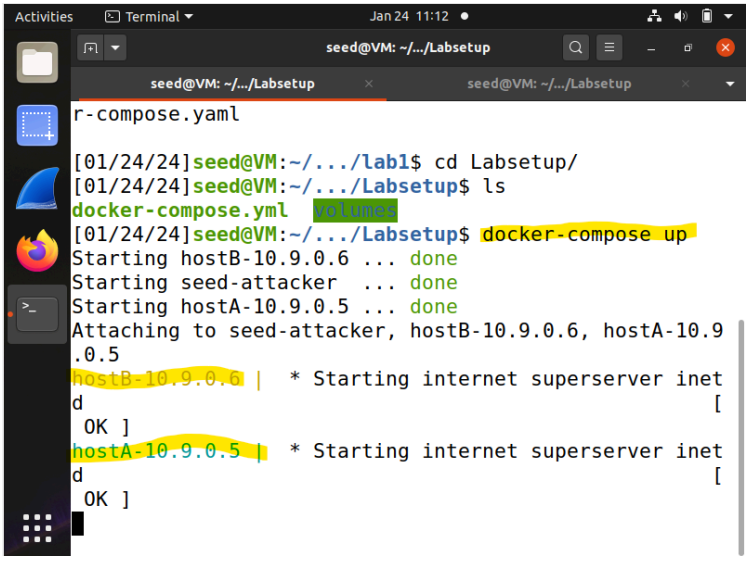
Darion Kwasnitza
3122890
Assignment 2

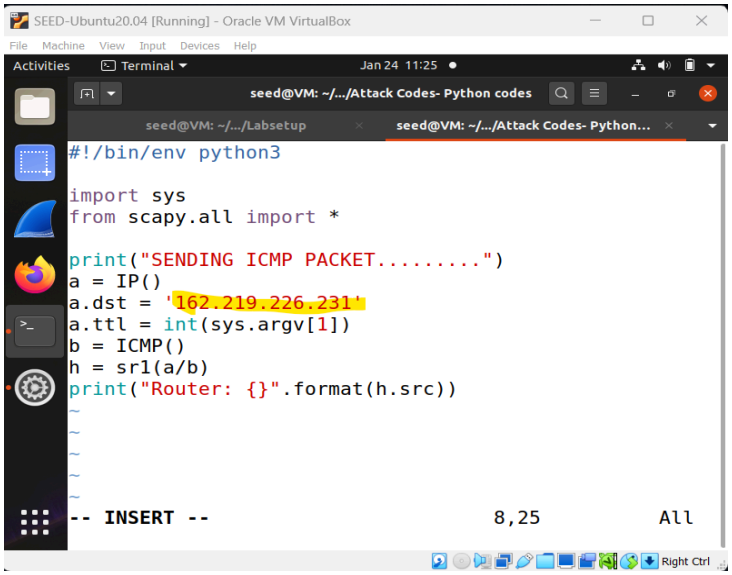
<u>Part One</u>

Task 1.3

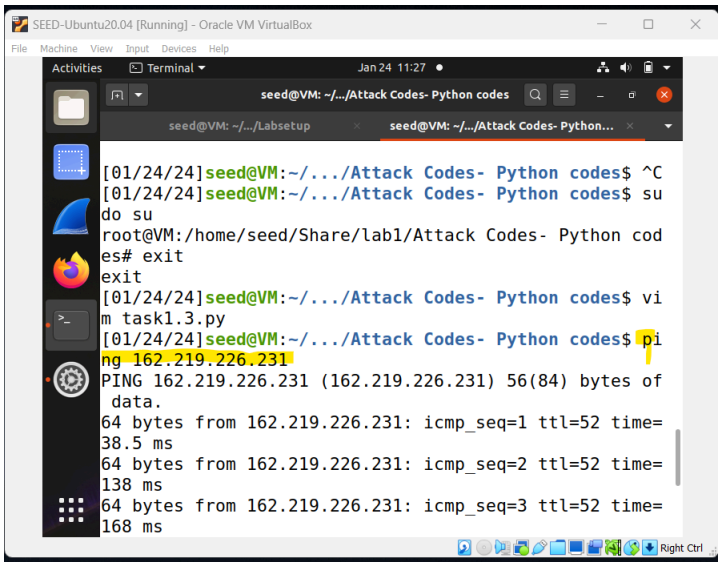
Ran docker-compose up



Changed the Ip destination address to 162.219.226.231 in task1.3.py

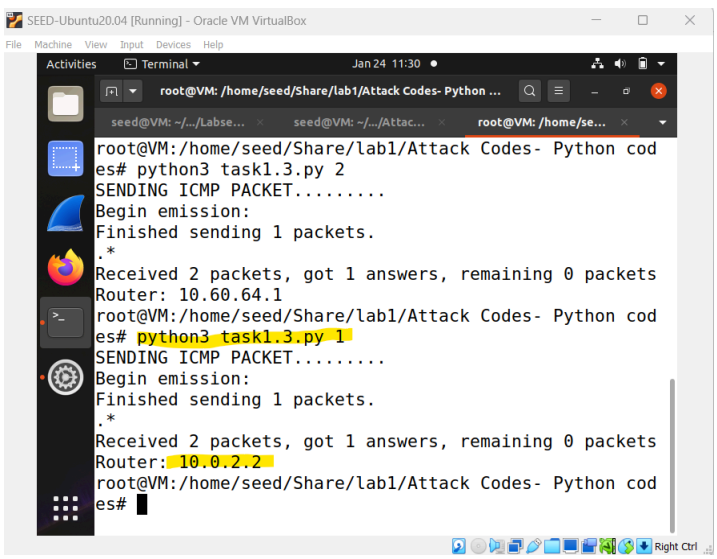


Pinged the address 162.219.226.231

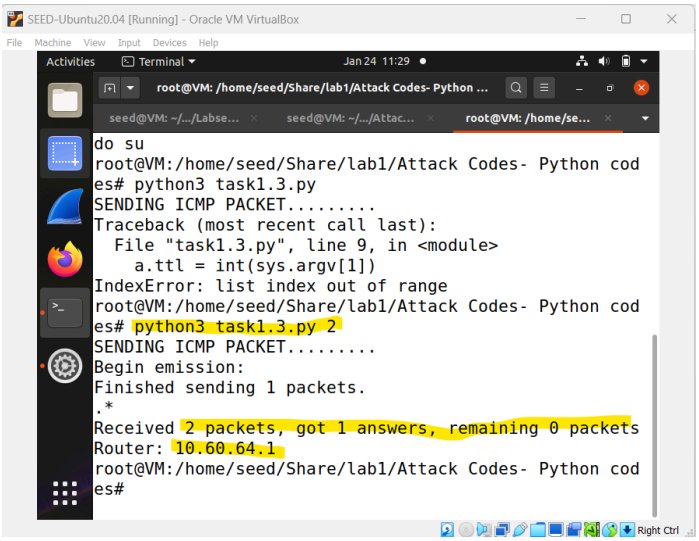


Ran the code with different TTL numbers, starting from one and increasing by one each time until I found the destination IP address as the Router.

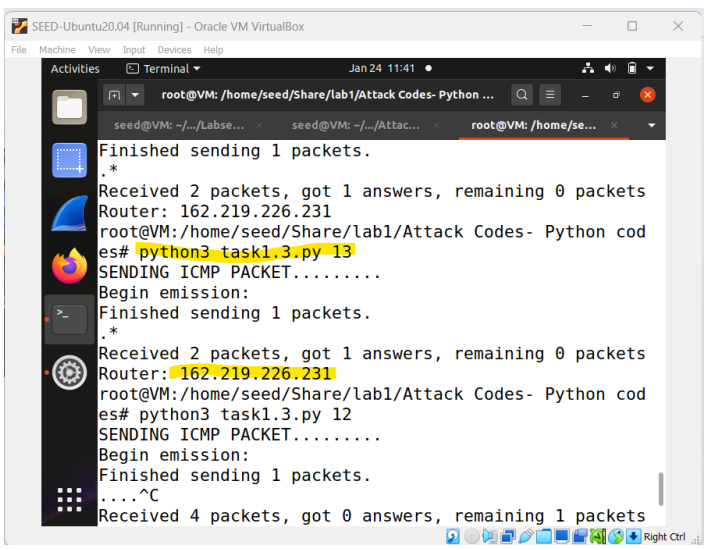
1)



2)

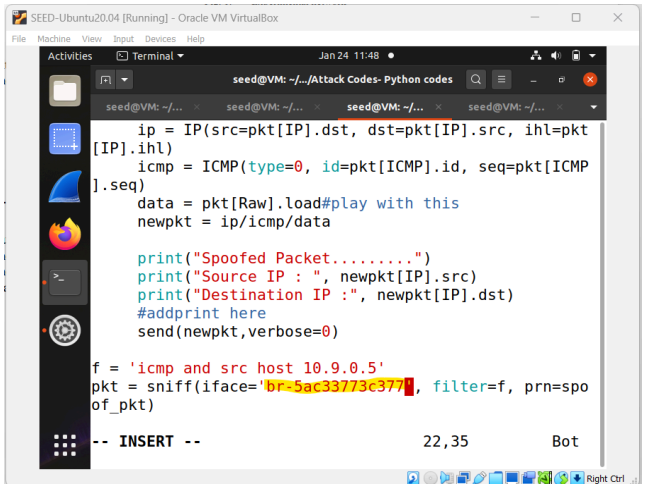


I got to TTP 13, and the Router IP address matched my destination address.

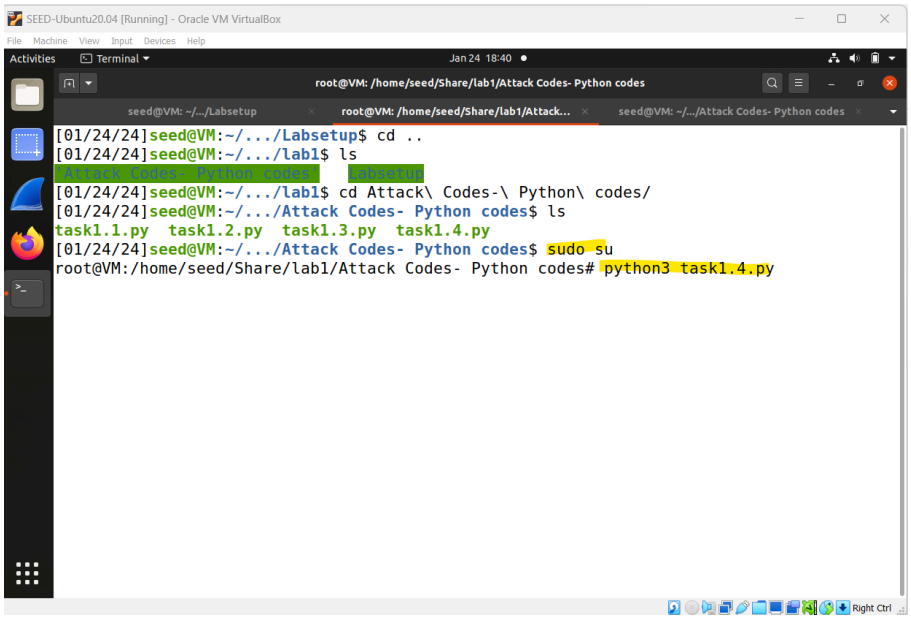


Task 1.4

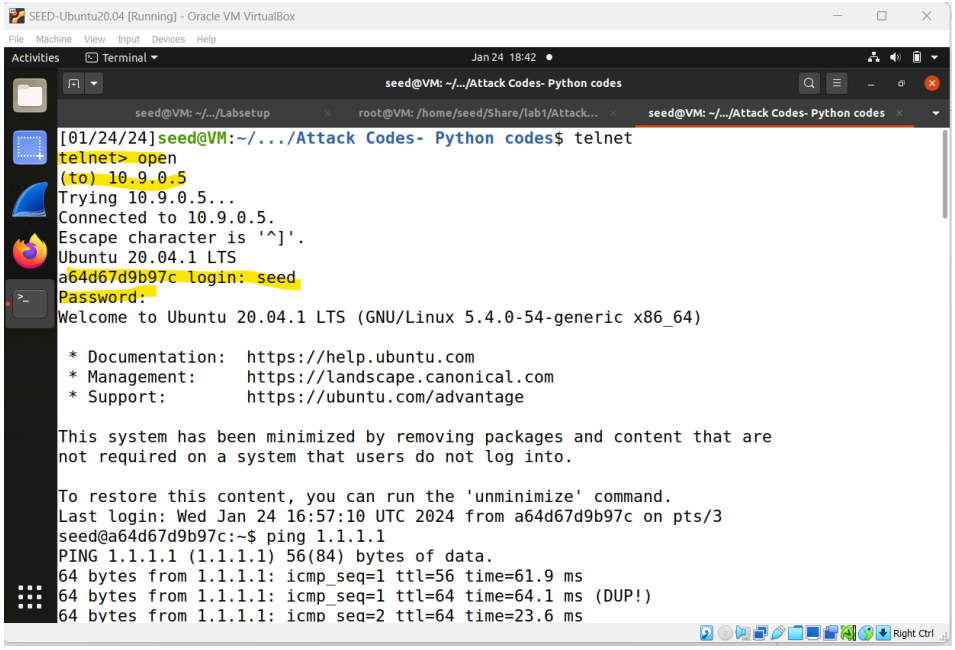
Changed iface to my computer address



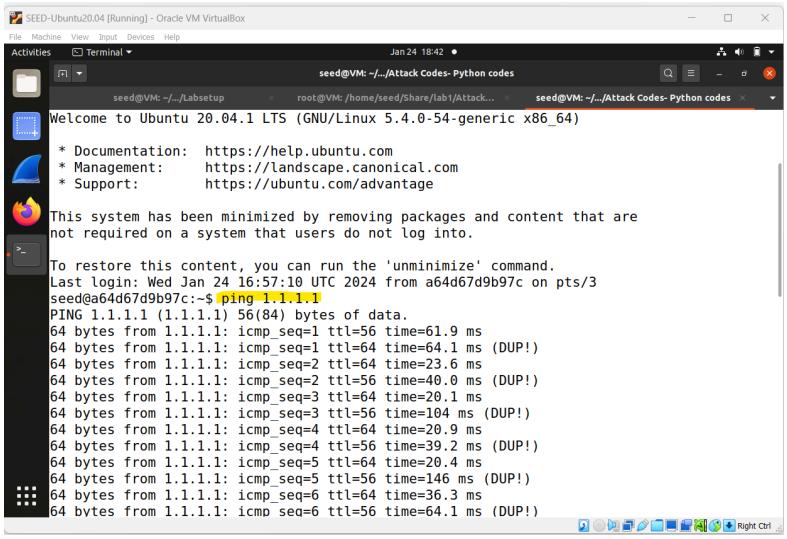
Ran task 1.4



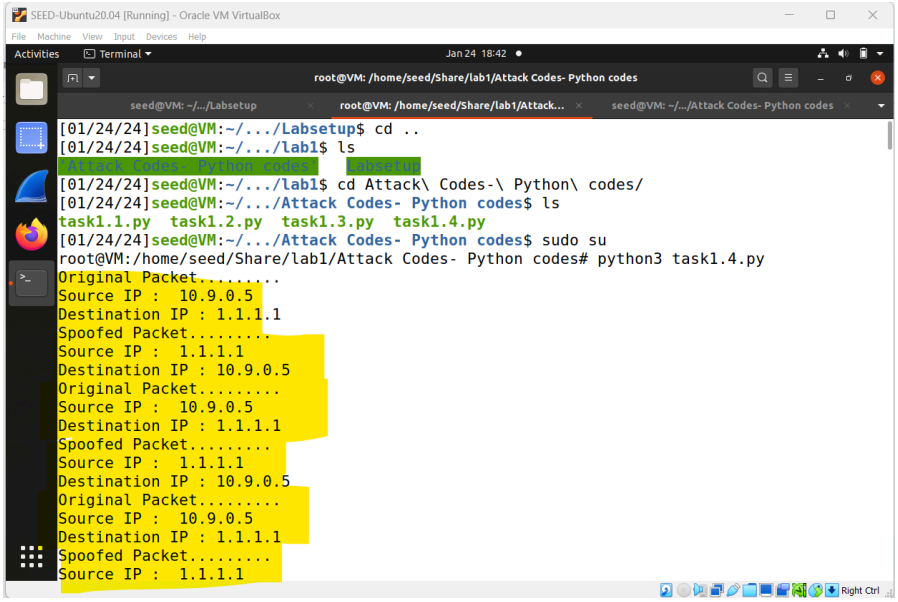
Logged in using seed credentials and using telnet to open 10.9.0.5



Sent a ping using 1.1.1.1

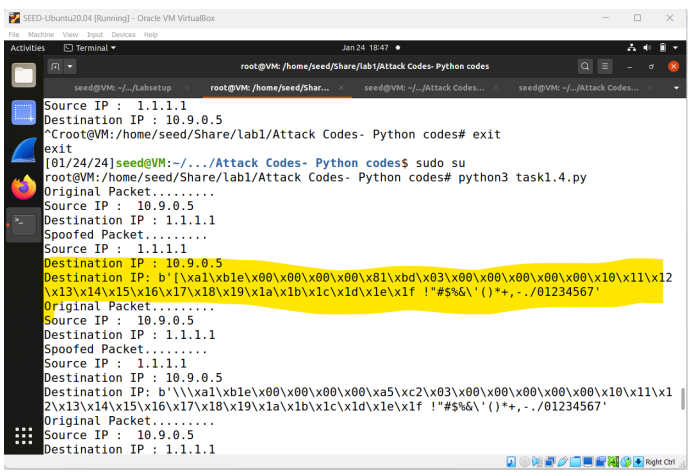


Packets showed up while running task 1.4, completed successfully.



Part Two

I was just messing around with the code, and I was able to send random gibberish as a destination IP



More importantly, I caused many errors in the spoofed packet when I pinged 1.1.1.1, and I am not sure exactly what it all means, but there is an overload in the size response and the wrong response. I found some interesting things to do in the documentation of Scapy; the highlighted parts of the code are from the Scapy documentation.

seed@VM: ~/.../Attack Codes- Python codes

seed@VM: ~/.../Labsetup | root@VM: /home/seed/Share/lab1/... | seed@VM: ~/.../Attack Codes- Pytho...

```
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a64d67d9b97c login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Wed Jan 24 23:59:37 UTC 2024 from a64d67d9b97c on pts/2
seed@a64d67d9b97c:~$ ping
ping: usage error: Destination address required
seed@a64d67d9b97c:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=56 time=114 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=56 time=86.4 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=56 time=55.9 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=56 time=77.3 ms
64 bytes from 1.1.1.1: icmp_seq=5 ttl=56 time=98.8 ms
```

root@VM: /home/seed/Share/lab1/Attack Codes- Python codes

seed@VM: ~/.../Labsetup | root@VM: /home/seed/Share/lab1/... | seed@VM: ~/.../Attack Codes- Pytho...

```
_run
    session.on_packet_received(p)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sessions.py", line 82, in o
n_packet_received
    result = self.prn(pkt)
  File "task1.4.py", line 21, in spoof_pkt
    icmp_echo_request = IP(dst="target_ip")/ICMP()
  File "/usr/local/lib/python3.8/dist-packages/scapy/base_classes.py", line 266,
 in __call__
    i.__init__(*args, **kargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/packet.py", line 170, in __
init__
    self.fields[fname] = self.get_field(fname).any2i(self, value)
  File "/usr/local/lib/python3.8/dist-packages/scapy/fields.py", line 568, in an
y2i
    return self.h2i(pkt, x)
  File "/usr/local/lib/python3.8/dist-packages/scapy/fields.py", line 543, in h2
i
    x = Net(x)
  File "/usr/local/lib/python3.8/dist-packages/scapy/base_classes.py", line 108,
 in __init__
    self.parsed, self.netmask = self._parse_net(net)
  File "/usr/local/lib/python3.8/dist-packages/scapy/base_classes.py", line 101,
 in _parse_net
```

seed@VM: ~/.../Attack Codes- Python codes

seed@VM: ~/.../Labsetup | root@VM: /home/seed/... | seed@VM: ~/.../Attack C... | seed@VM: ~/.../Attack C...

```
  if ICMP in pkt and pkt[ICMP].type == 8:
      print("Original Packet.........")
      print("Source IP : ", pkt[IP].src)
      print("Destination IP :", pkt[IP].dst)

      ip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
      icmp = ICMP(type=8, id=pkt[ICMP].id, seq=pkt[ICMP].seq)
      data =b'X' * 10000 # pkt[Raw].load
      newpkt = ip/icmp/data

      print("Spoofed Packet.........")
      print("Source IP : ", newpkt[IP].src)
      print("Destination IP :", newpkt[IP].dst)
      print("size: ",len( data))
      send(newpkt,verbose=0)
     # Craft an ICMP Echo Request packet
      icmp_echo_request = IP(dst="target_ip")/ICMP()

     # Send the crafted packet and receive the response
      response = sr1(icmp_echo_request, timeout=2, verbose=0)
f = 'icmp and src host 10.9.0.5'
pkt = sniff(iface='br-5ac33773c377', filter=f, prn=spoof_pkt)

"task1.4.py" [readonly] 27L, 916C                          25,1          Bot
```