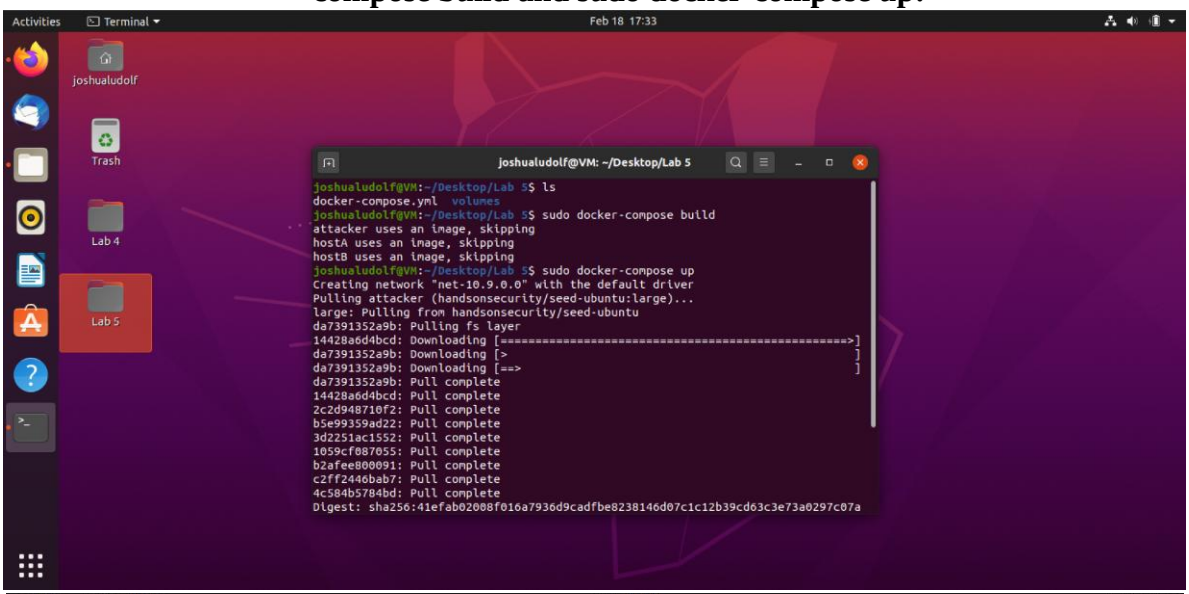
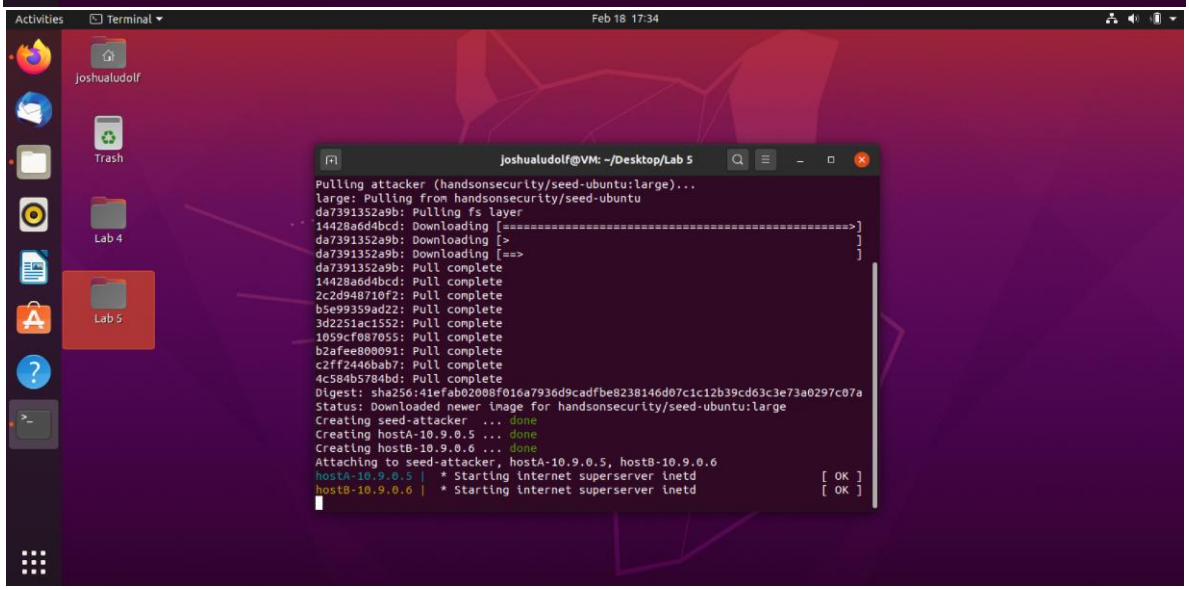
Lab 5: Packet Sniffing and Spoofing Lab

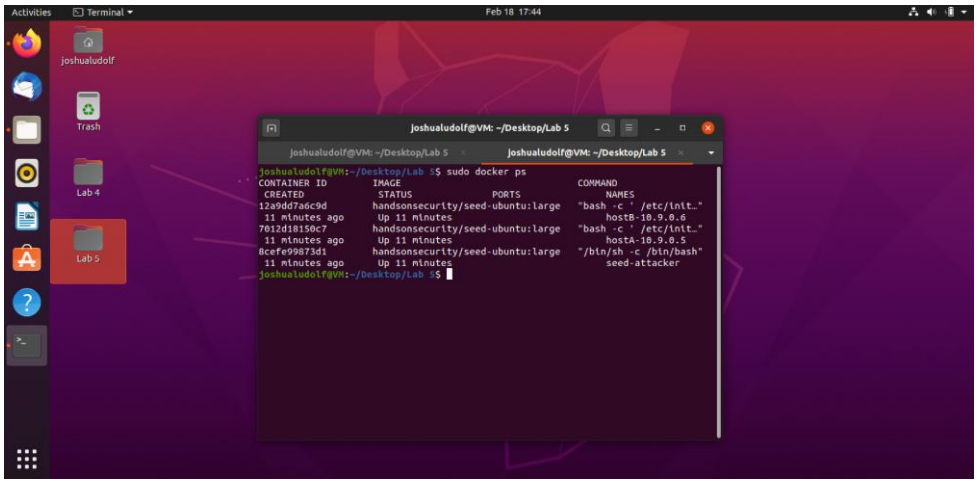
Joshua Ludolf
CSCI 4321
Computer Security

- **To start this lab, I booted up my vm and typed following command – sudo docker-compose build and sudo docker-compose up:**

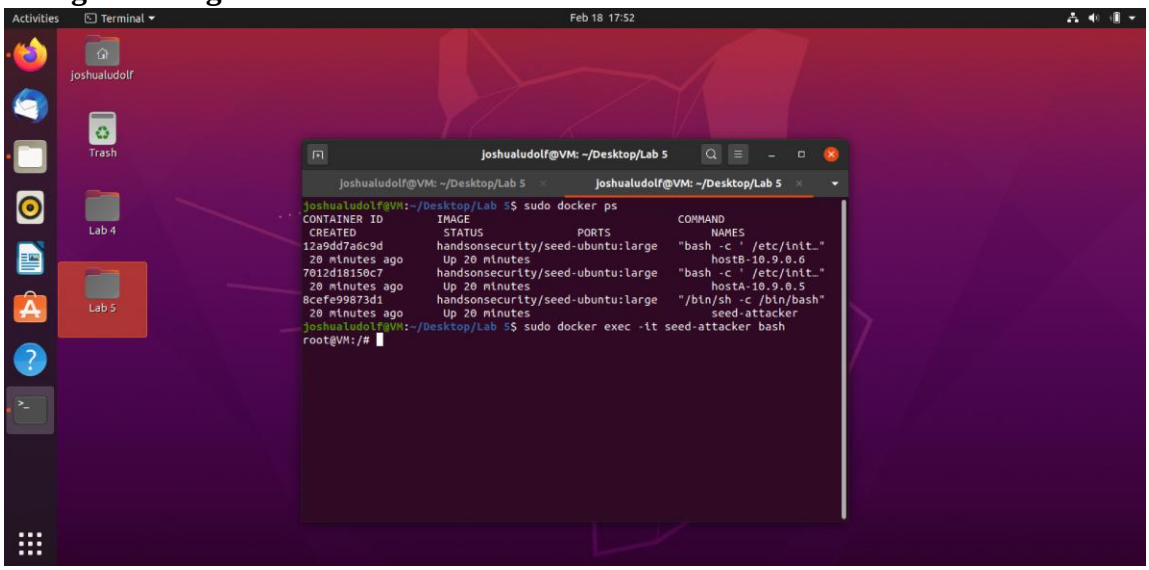




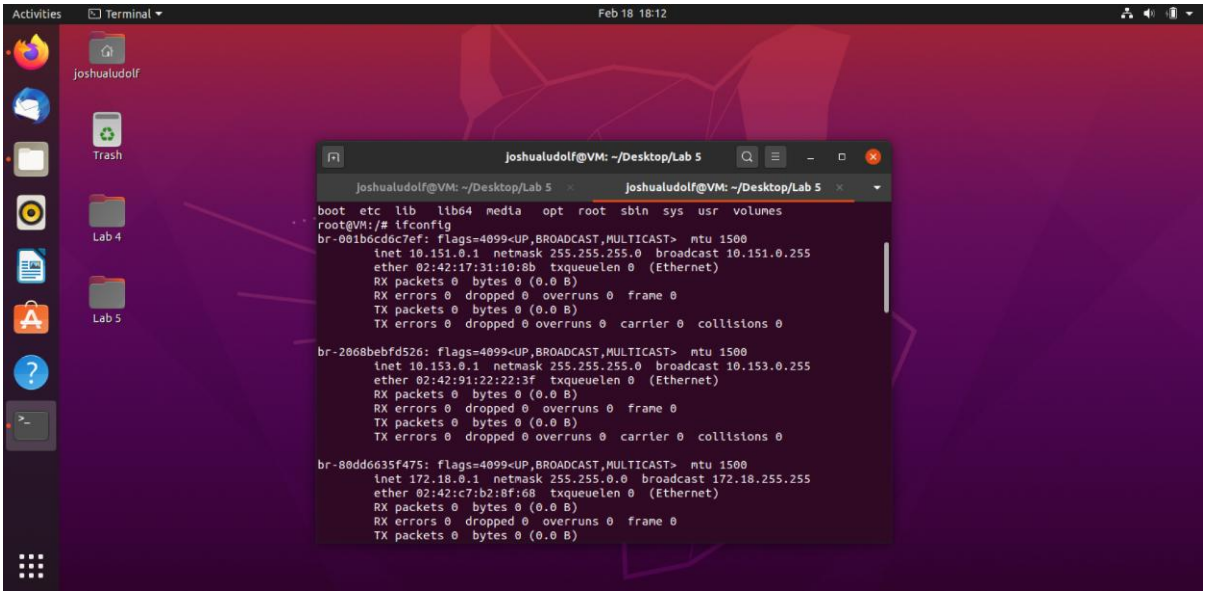
- **From there, I acquired the different ip address for this packet sniffing and spoofing attack by inputting following command – sudo docker ps (the names at the end of the output [ones to right most column]):**



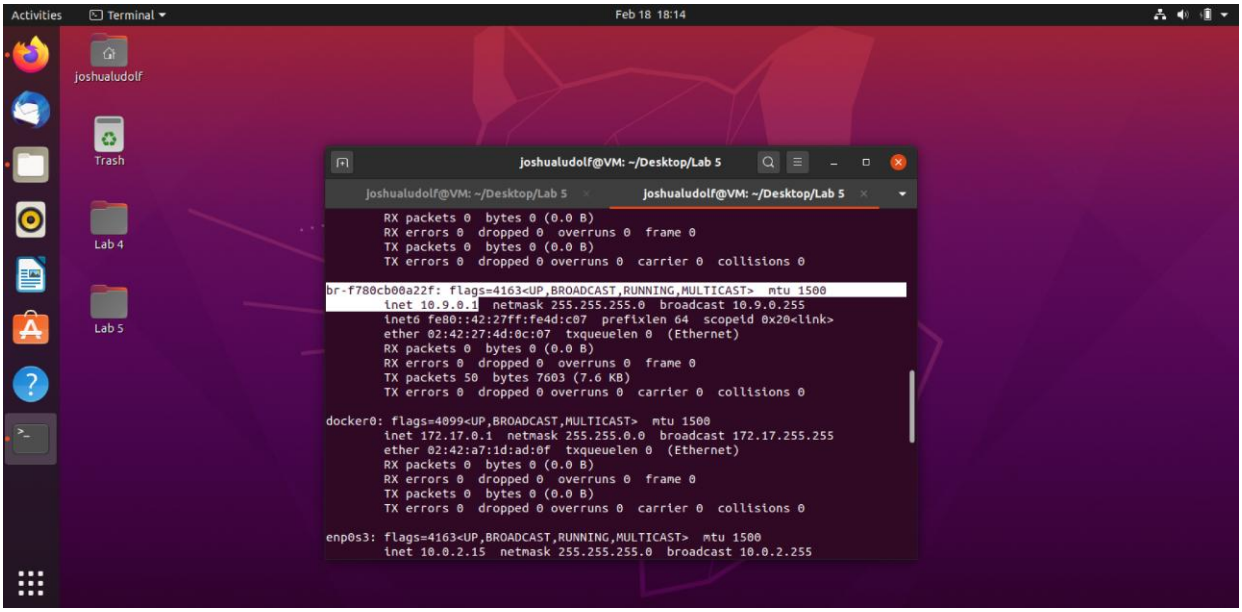
- **To exploit theses hosts, I need to open an interactive bash shell in the container; utilizing following command – sudo docker exec -it seed-attacker bash:**



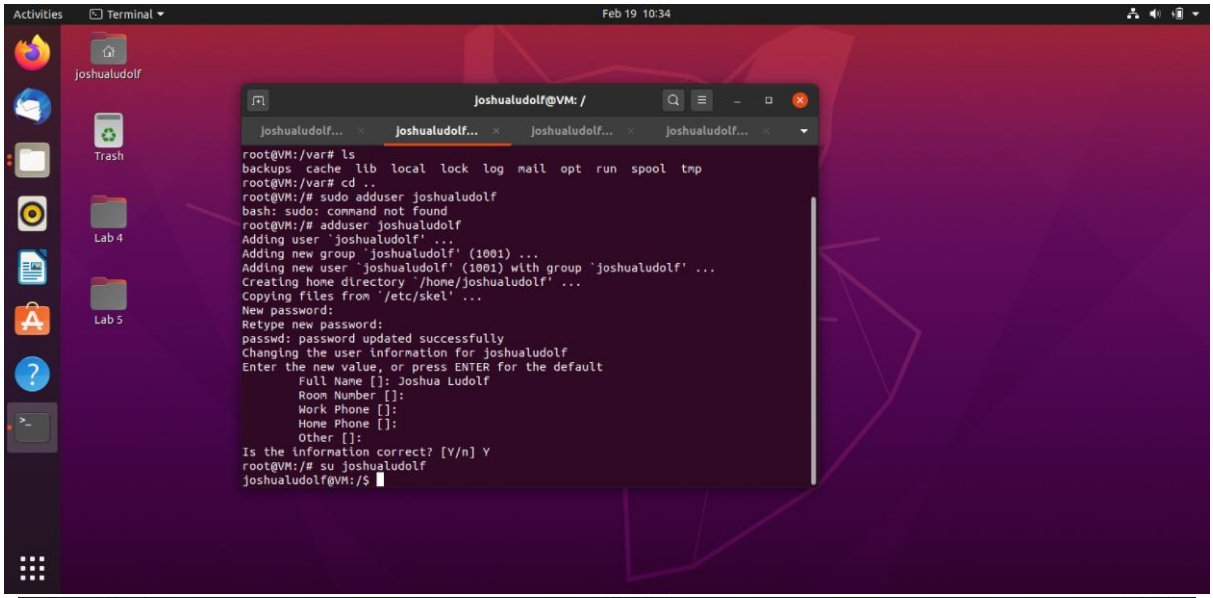
- **Following that, I acquired the bridge ID for the host internet protocol; using this command – ifconfig (luckily found where it talks about command from original manual😎):**

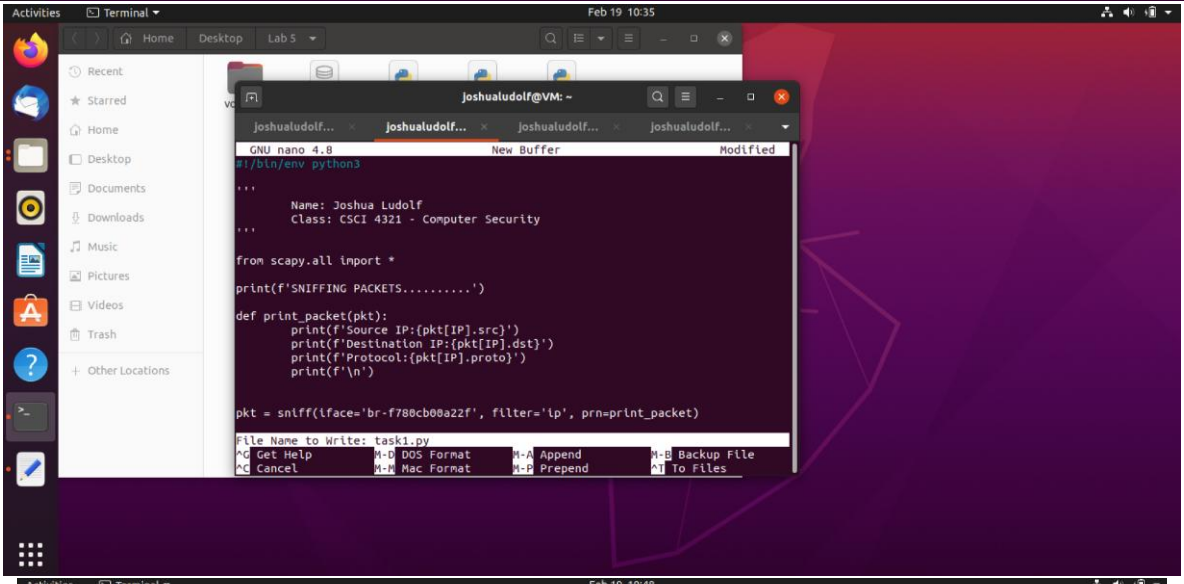


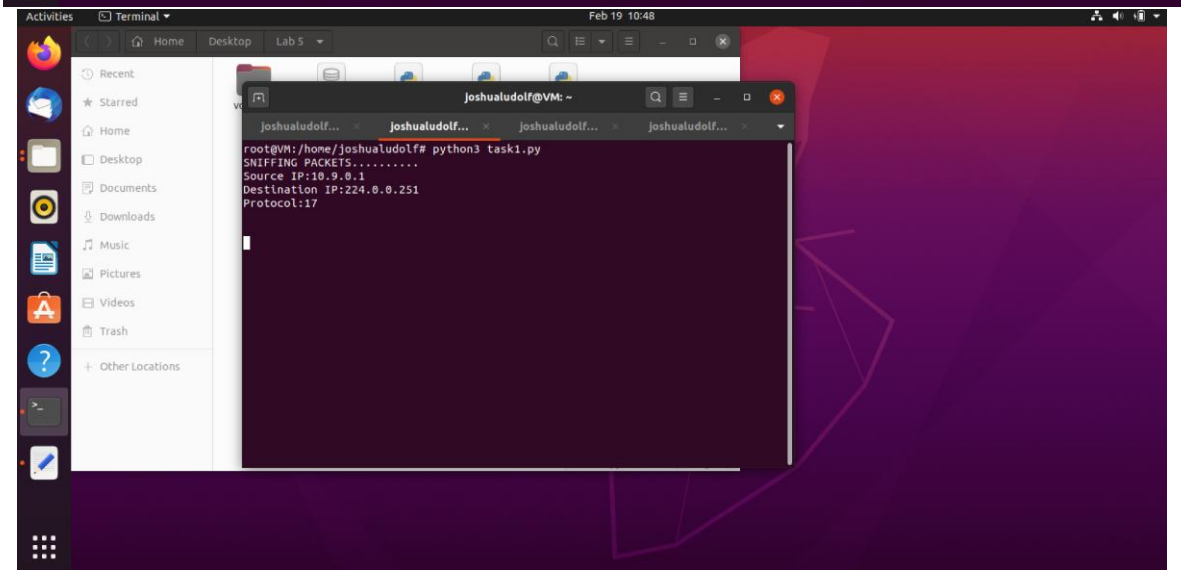
- **In my case its br-f780cb00a22f:**



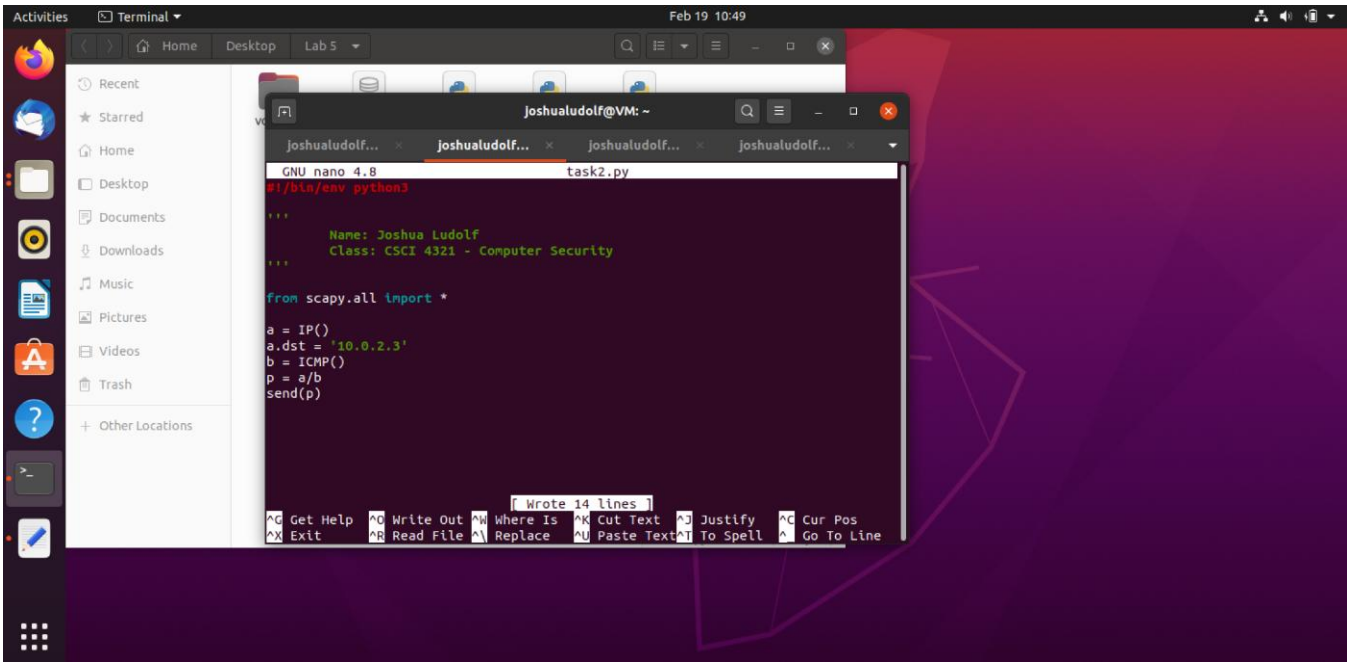
- **Now I am ready to start the actual tasks (1.1, 1.2, 1.3, & 1.4); to start I take the name that I acquired and edit task1.py (additionally needed to create user in the docker container, wasn't noted in either instruction, however I learned that it requires root privileges to run in the container and sudo doesn't work so have to be root user):**

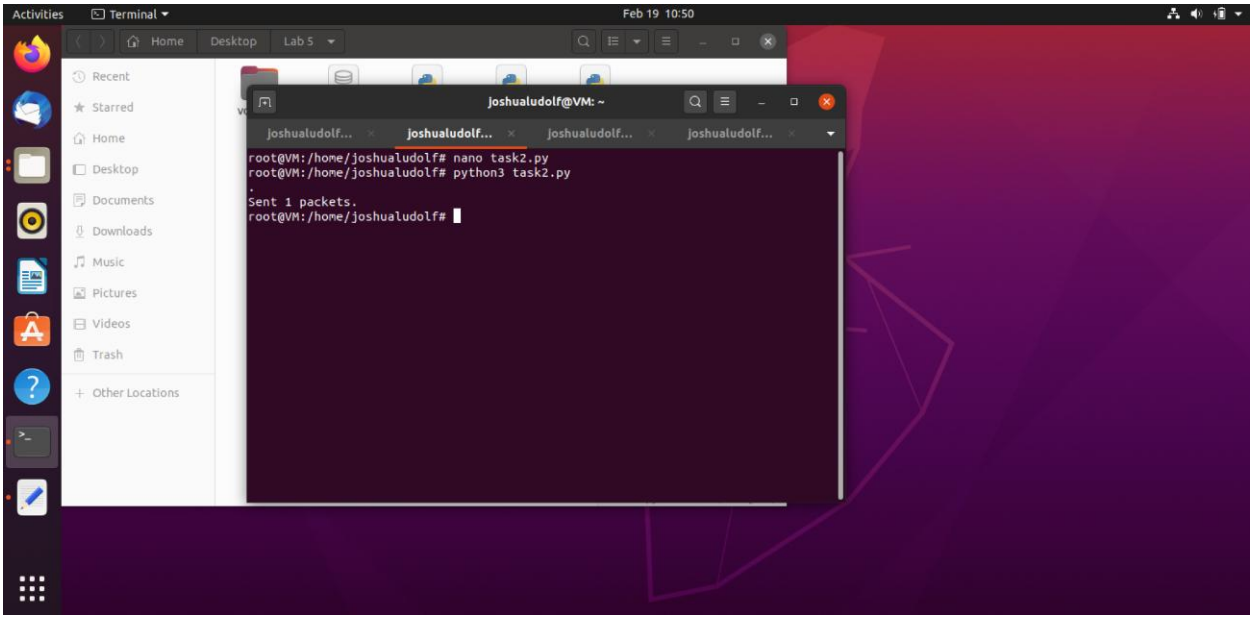

Activities
Terminal
Feb 19 10:34
joshualudolf
Trash
Lab 4
Lab 5
joshualudolf@VM: /
joshualudolf...
joshualudolf...
joshualudolf...
joshualudolf...
root@VM:/var# ls
backups cache lib local lock log mail opt run spool tmp
root@VM:/var# cd ..
root@VM:/# sudo adduser joshualudolf
bash: sudo: command not found
root@VM:/# adduser joshualudolf
Adding user `joshualudolf' ...
Adding new group `joshualudolf' (1001) ...
Adding new user `joshualudolf' (1001) with group `joshualudolf' ...
Creating home directory `/home/joshualudolf' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for joshualudolf
Enter the new value, or press ENTER for the default
Full Name []: Joshua Ludolf
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] Y
root@VM:/# su joshualudolf
joshualudolf@VM:/$


Activities
Terminal
Feb 19 10:35
Home
Desktop
Lab 5
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
joshualudolf@VM: ~
joshualudolf...
joshualudolf...
joshualudolf...
joshualudolf...
GNU nano 4.8
New Buffer
Modified
#!/bin/env python3
'''
Name: Joshua Ludolf
Class: CSCI 4321 - Computer Security
'''
from scapy.all import *
print(f'SNIFFING PACKETS.....')
def print_packet(pkt):
print(f'Source IP:{pkt[IP].src}')
print(f'Destination IP:{pkt[IP].dst}')
print(f'Protocol:{pkt[IP].proto}')
print(f'\n')
pkt = sniff(iface='br-f780cb08a22f', filter='ip', prn=print_packet)
File Name to Write: task1.py
^G Get Help
^C Cancel
M-D DOS Format
M-M Mac Format
M-A Append
M-P Prepend
M-B Backup File
^T To Files

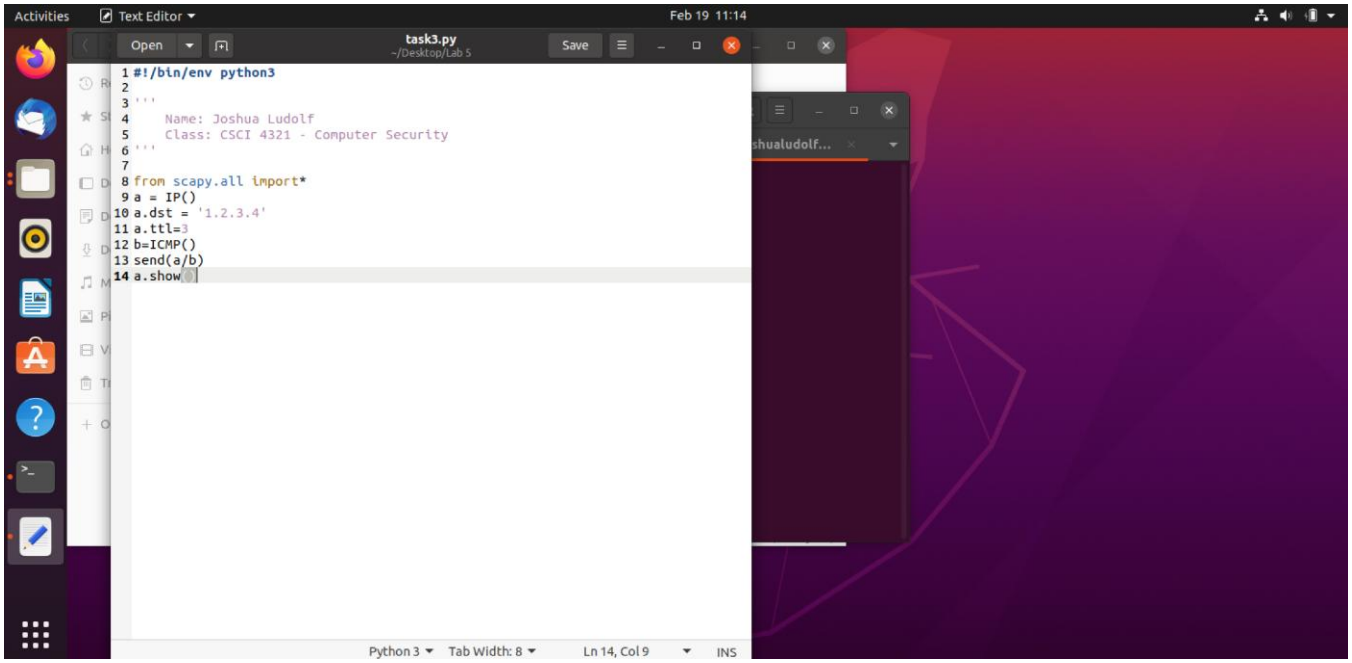

Activities
Terminal
Feb 19 10:48
Home
Desktop
Lab 5
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
joshualudolf@VM: ~
joshualudolf...
joshualudolf...
joshualudolf...
joshualudolf...
root@VM:/home/joshualudolf# python3 task1.py
SNIFFING PACKETS.....
Source IP:10.9.0.1
Destination IP:224.0.0.251
Protocol:17

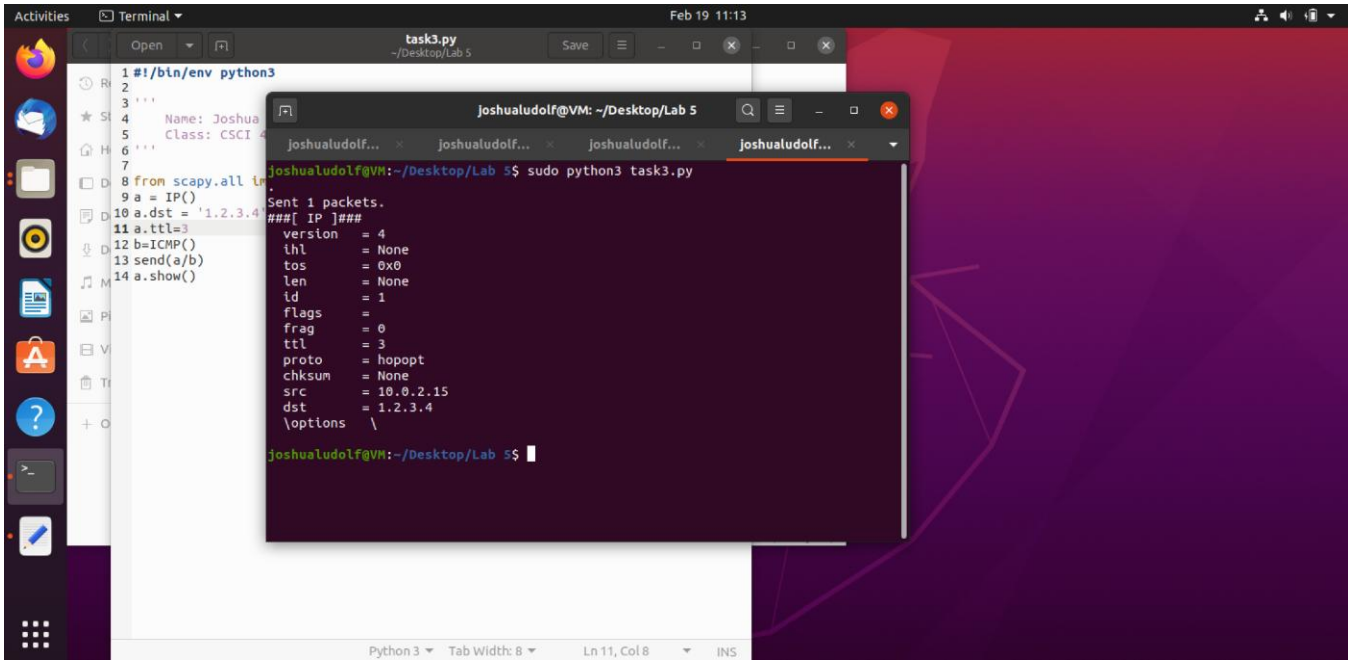
- **For task 1.2, I created another python script to send an ICMP packet to 10.0.2.3:**



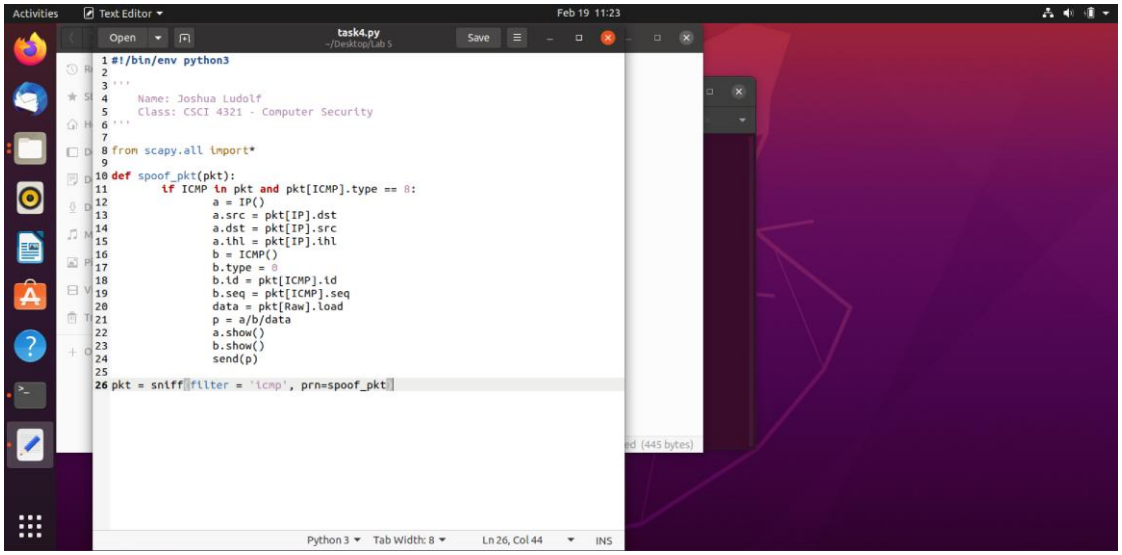


- **For task 1.3, I implemented traceroute to estimate the distance between the routers; utilizing another simple python script (learned that this only works in the vm so I open new terminal and ttl didn't matter in our case as it only affected ttl nothing else…):**



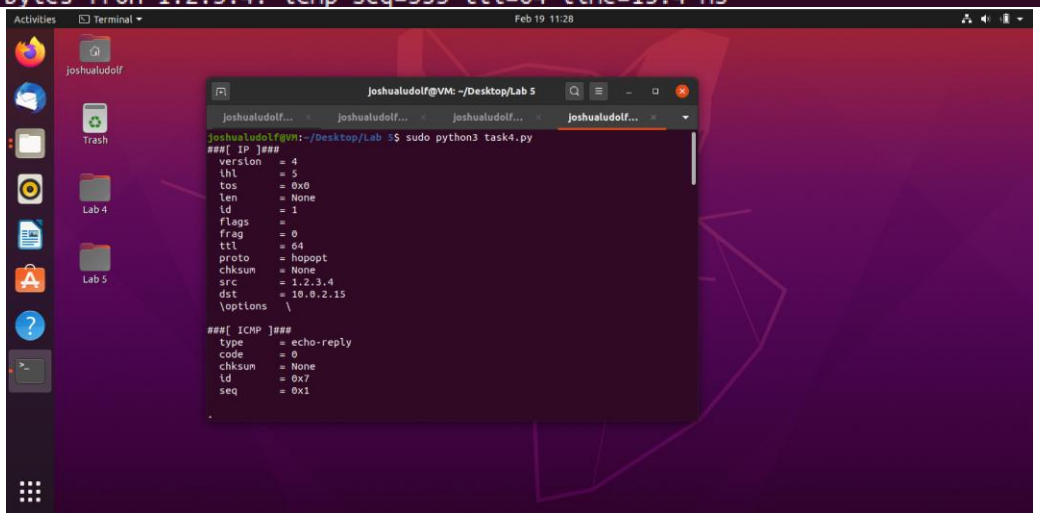


- **Finally, task 1.4 was to combine the spoofing and sniffing techniques, which this part required pinging from container and python script that runs on the main virtual machine to sniff the container:**



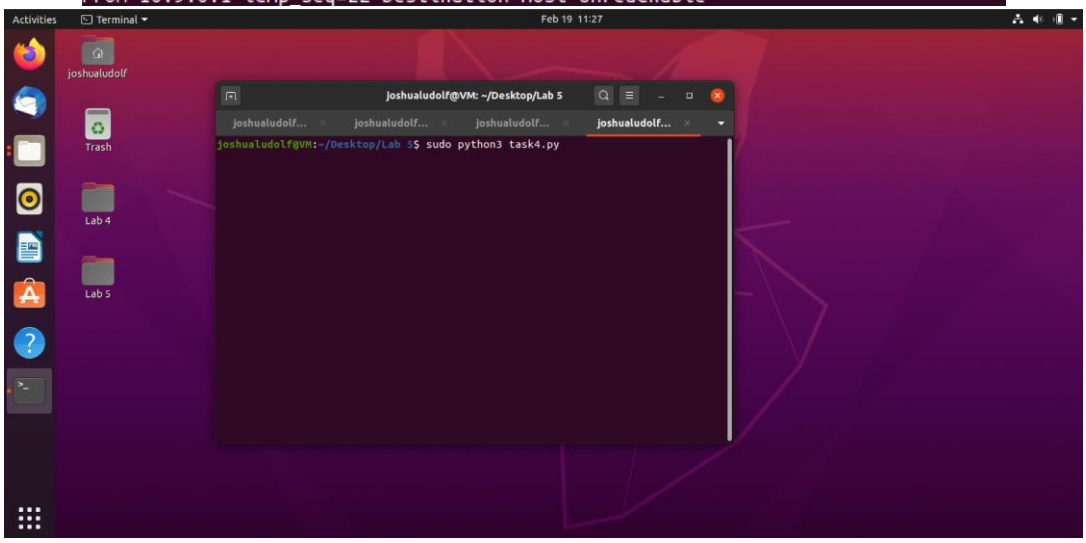
- **For ip address 1.2.3.4:**

```
root@VM:/home/joshualudolf# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=319 ttl=64 time=17.4 ms
64 bytes from 1.2.3.4: icmp_seq=320 ttl=64 time=17.4 ms
64 bytes from 1.2.3.4: icmp_seq=321 ttl=64 time=20.0 ms
64 bytes from 1.2.3.4: icmp_seq=322 ttl=64 time=17.5 ms
64 bytes from 1.2.3.4: icmp_seq=323 ttl=64 time=11.6 ms
64 bytes from 1.2.3.4: icmp_seq=324 ttl=64 time=16.5 ms
64 bytes from 1.2.3.4: icmp_seq=325 ttl=64 time=14.6 ms
64 bytes from 1.2.3.4: icmp_seq=326 ttl=64 time=20.1 ms
64 bytes from 1.2.3.4: icmp_seq=327 ttl=64 time=22.1 ms
64 bytes from 1.2.3.4: icmp_seq=328 ttl=64 time=23.7 ms
64 bytes from 1.2.3.4: icmp_seq=329 ttl=64 time=21.9 ms
64 bytes from 1.2.3.4: icmp_seq=330 ttl=64 time=28.4 ms
64 bytes from 1.2.3.4: icmp_seq=331 ttl=64 time=15.1 ms
64 bytes from 1.2.3.4: icmp_seq=332 ttl=64 time=25.1 ms
64 bytes from 1.2.3.4: icmp_seq=333 ttl=64 time=19.3 ms
64 bytes from 1.2.3.4: icmp_seq=334 ttl=64 time=19.1 ms
64 bytes from 1.2.3.4: icmp_seq=335 ttl=64 time=13.4 ms
```

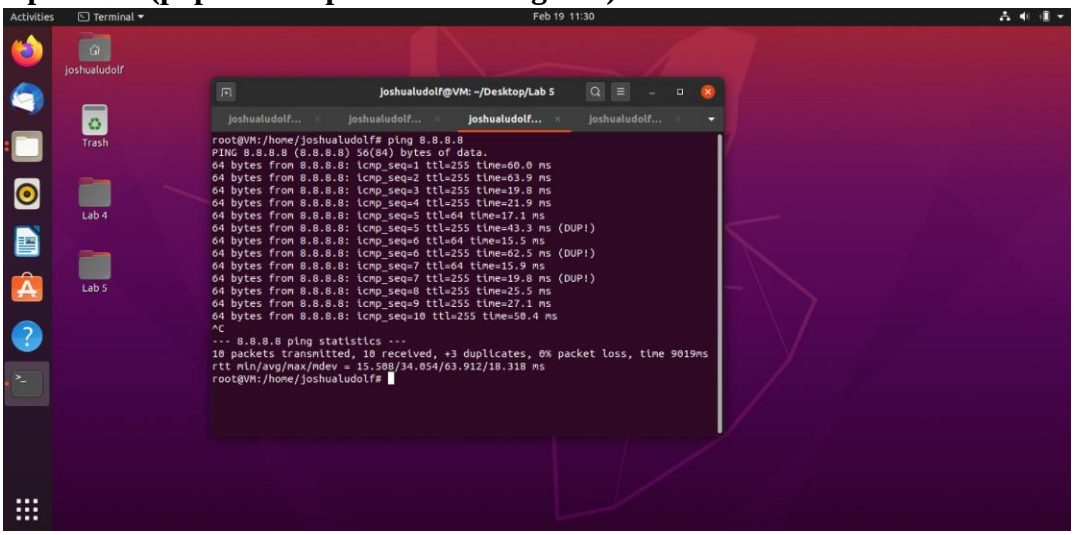


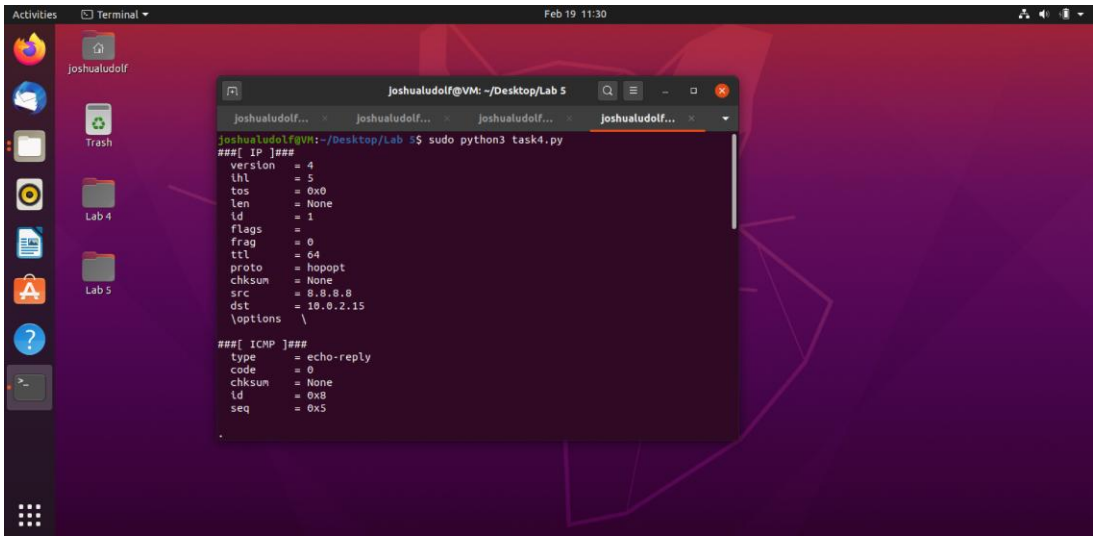
- **For ip address 10.9.0.99:**

```
root@VM:/home/joshualudolf# ping 10.9.0.99
PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.
From 10.9.0.1 icmp_seq=1 Destination Host Unreachable
From 10.9.0.1 icmp_seq=2 Destination Host Unreachable
From 10.9.0.1 icmp_seq=3 Destination Host Unreachable
From 10.9.0.1 icmp_seq=4 Destination Host Unreachable
From 10.9.0.1 icmp_seq=5 Destination Host Unreachable
From 10.9.0.1 icmp_seq=6 Destination Host Unreachable
From 10.9.0.1 icmp_seq=7 Destination Host Unreachable
From 10.9.0.1 icmp_seq=8 Destination Host Unreachable
From 10.9.0.1 icmp_seq=9 Destination Host Unreachable
From 10.9.0.1 icmp_seq=10 Destination Host Unreachable
From 10.9.0.1 icmp_seq=11 Destination Host Unreachable
From 10.9.0.1 icmp_seq=12 Destination Host Unreachable
From 10.9.0.1 icmp_seq=13 Destination Host Unreachable
From 10.9.0.1 icmp_seq=14 Destination Host Unreachable
From 10.9.0.1 icmp_seq=15 Destination Host Unreachable
From 10.9.0.1 icmp_seq=16 Destination Host Unreachable
From 10.9.0.1 icmp_seq=17 Destination Host Unreachable
From 10.9.0.1 icmp_seq=18 Destination Host Unreachable
From 10.9.0.1 icmp_seq=19 Destination Host Unreachable
From 10.9.0.1 icmp_seq=20 Destination Host Unreachable
From 10.9.0.1 icmp_seq=21 Destination Host Unreachable
From 10.9.0.1 icmp_seq=22 Destination Host Unreachable
```



- **For ip 8.8.8.8 (popular for port forwarding btw):**





- **What I learned from this lab:**

In completing tasks 1.1 to 1.4 of the SEED Labs' Packet Sniffing and Spoofing Lab, I gained a lot of valuable insights and hands-on experience. I started with packet sniffing, using tools like Tcpdump to capture and analyze network traffic. This helped me understand how data packets are transmitted over a network. Then, I moved on to packet spoofing, where I learned to create and send forged packets using Scapy. This was particularly eye-opening as it showed me how attackers might manipulate network traffic to deceive systems or users. Additionally, I wrote simple programs to perform packet sniffing and spoofing, which deepened my understanding of the underlying technologies and protocols. Through these exercises, I also explored the security implications of these techniques, realizing how crucial it is to implement security measures to protect against such threats. Overall, these tasks provided a comprehensive introduction to the practical aspects of network security, enhancing my ability to both understand and mitigate these common threats.