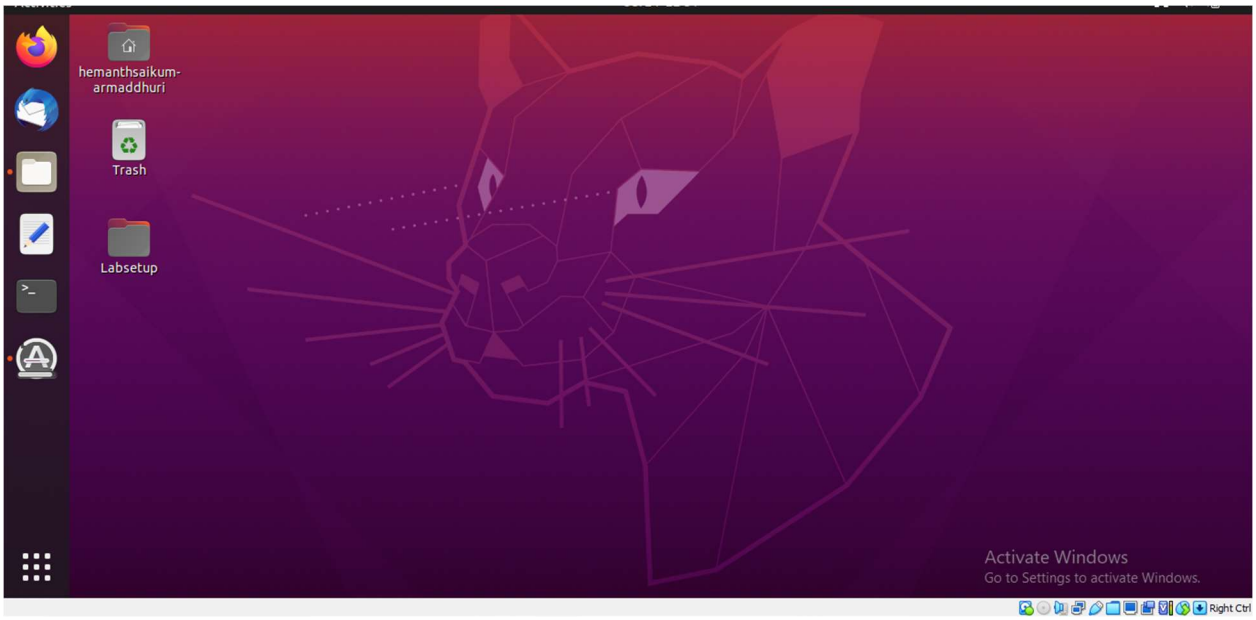
Information and Networking Security

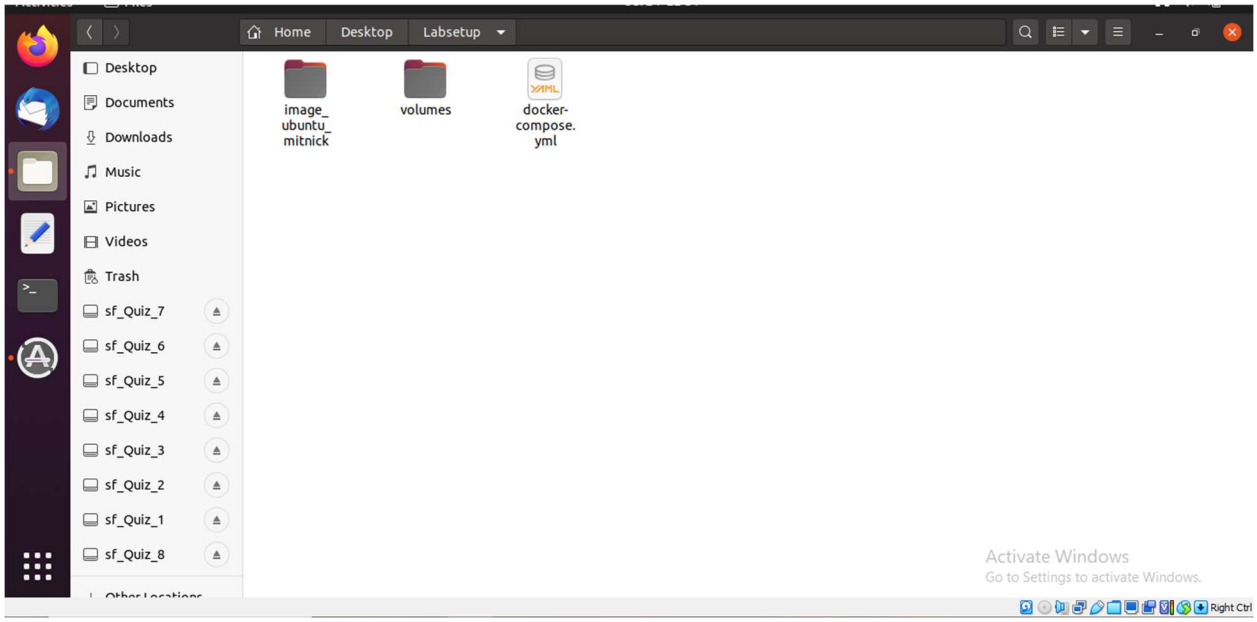
Quiz – 8

Name: Hemanth Sai Kumar Maddhuri **ID:** 999902480

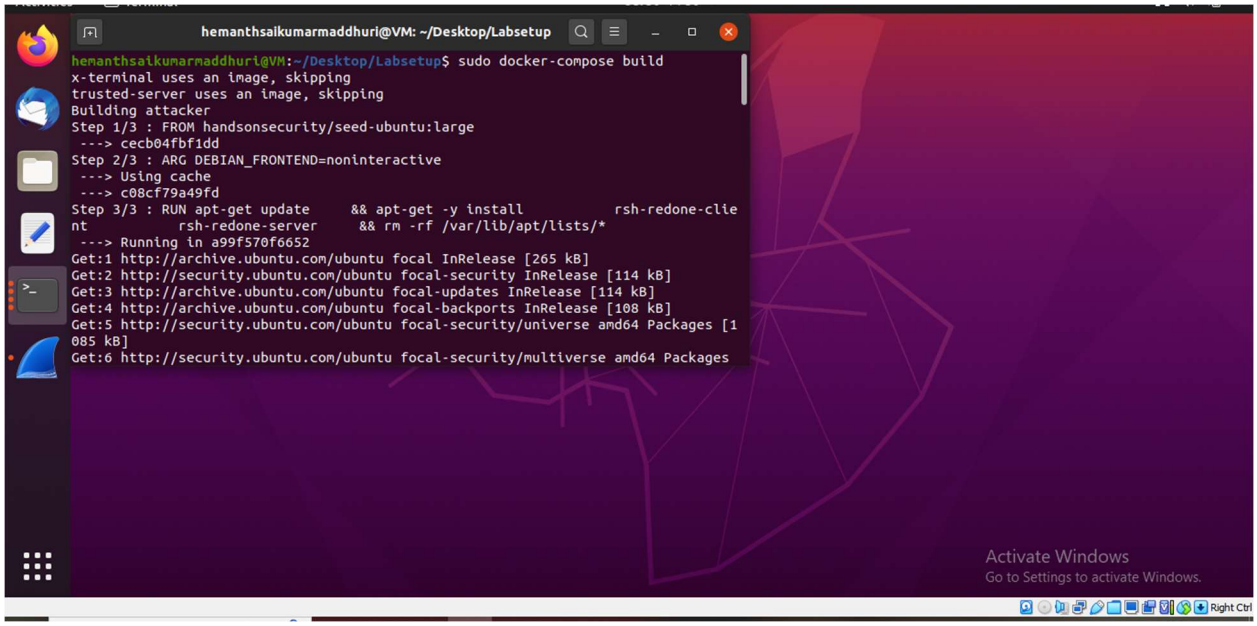
As Instructed, I have downloaded the Labsetup file from the following URL: https://seedsecuritylabs.org/Labs_20.04/Files/Mitnick_Attack/Labsetup.zip.



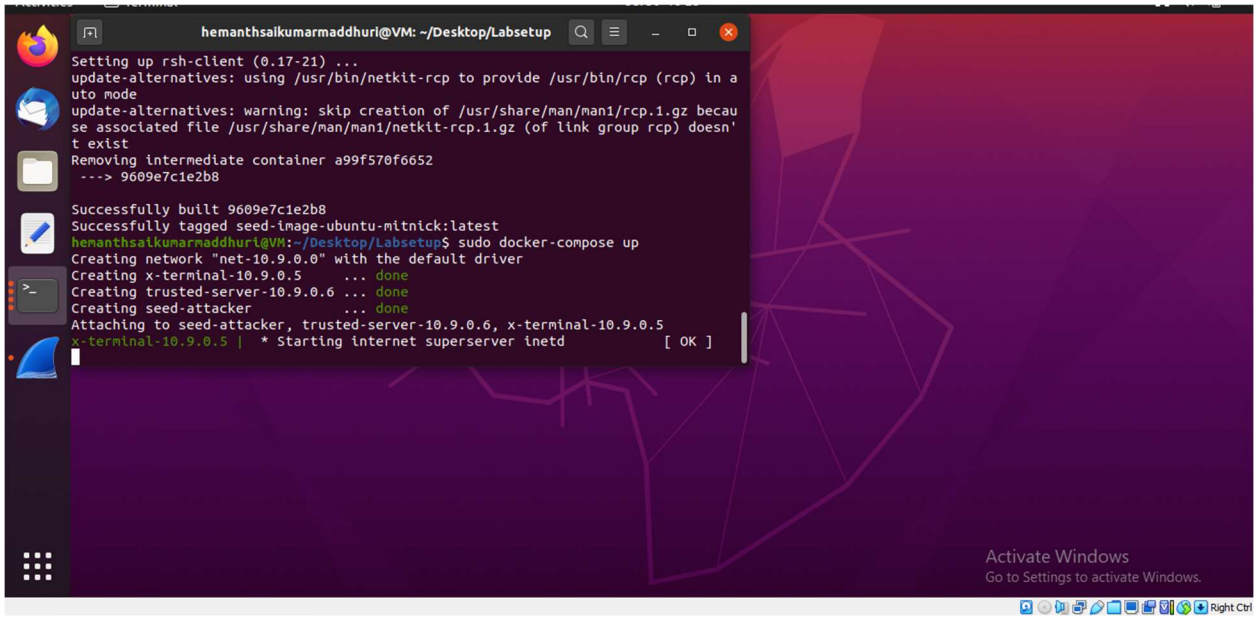
The downloaded Labsetup files are as shown below.



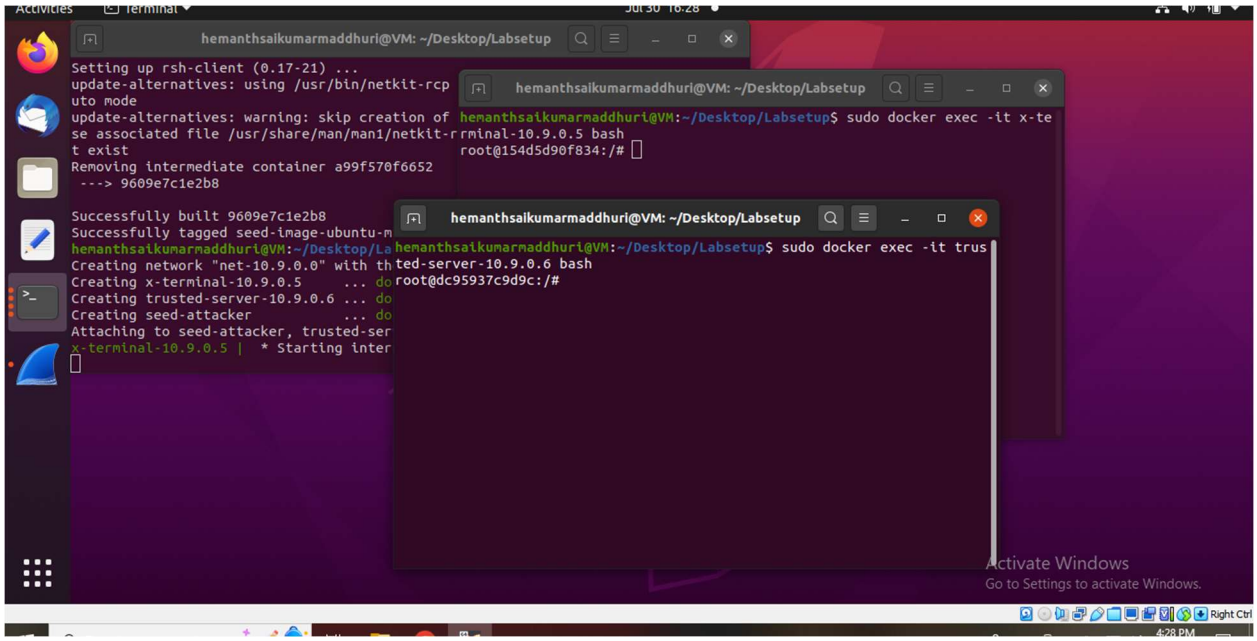
Setting up the docker using the command **"sudo docker-compose build"**.



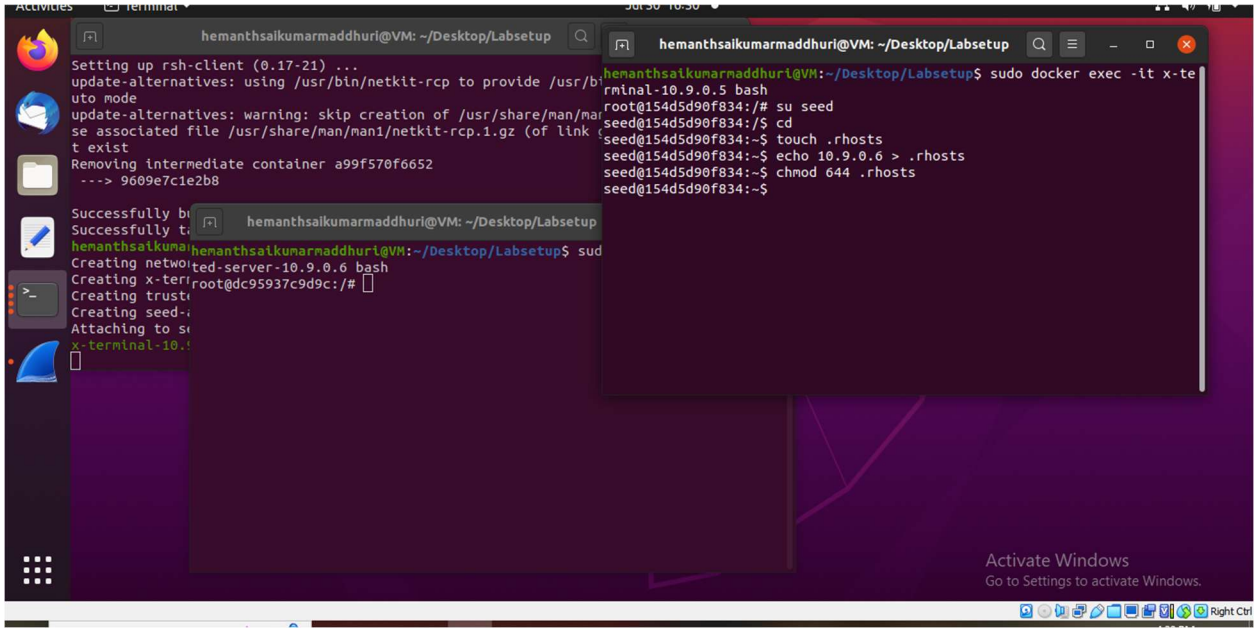
Turning the dockers up using the command **“sudo docker-compose up”**.



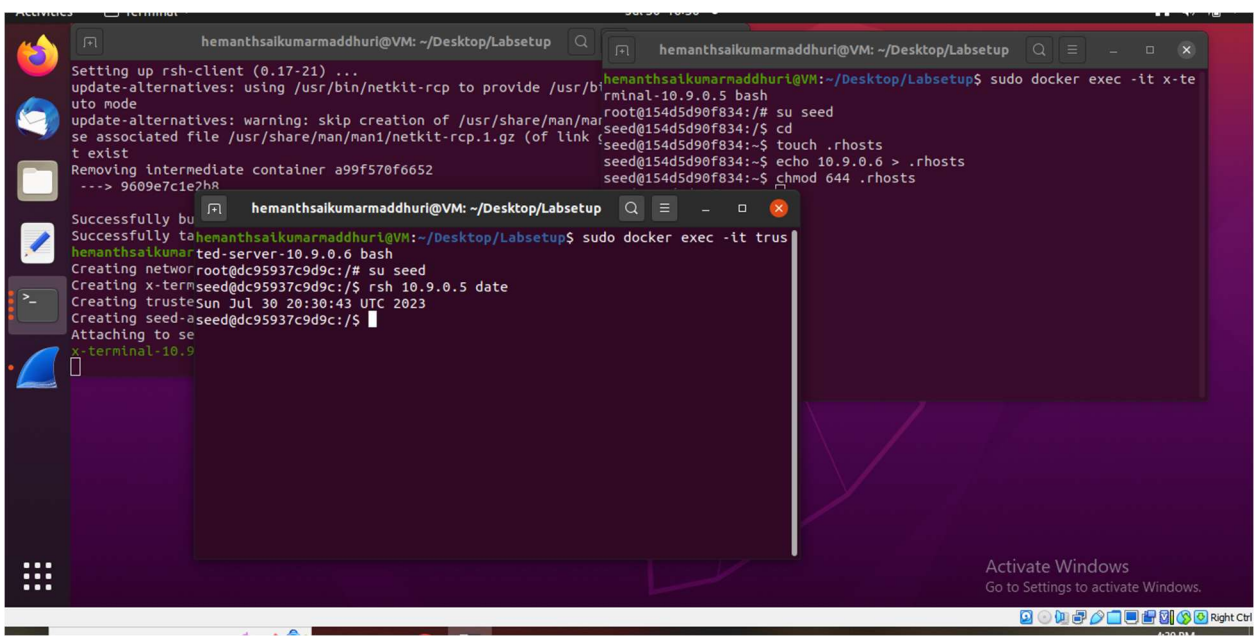
Logging into trusted server and x-terminal machines using the command **“sudo docker exec -it <machine info> bash”**.



Later setting up trust relationship on x-terminal as shown below.

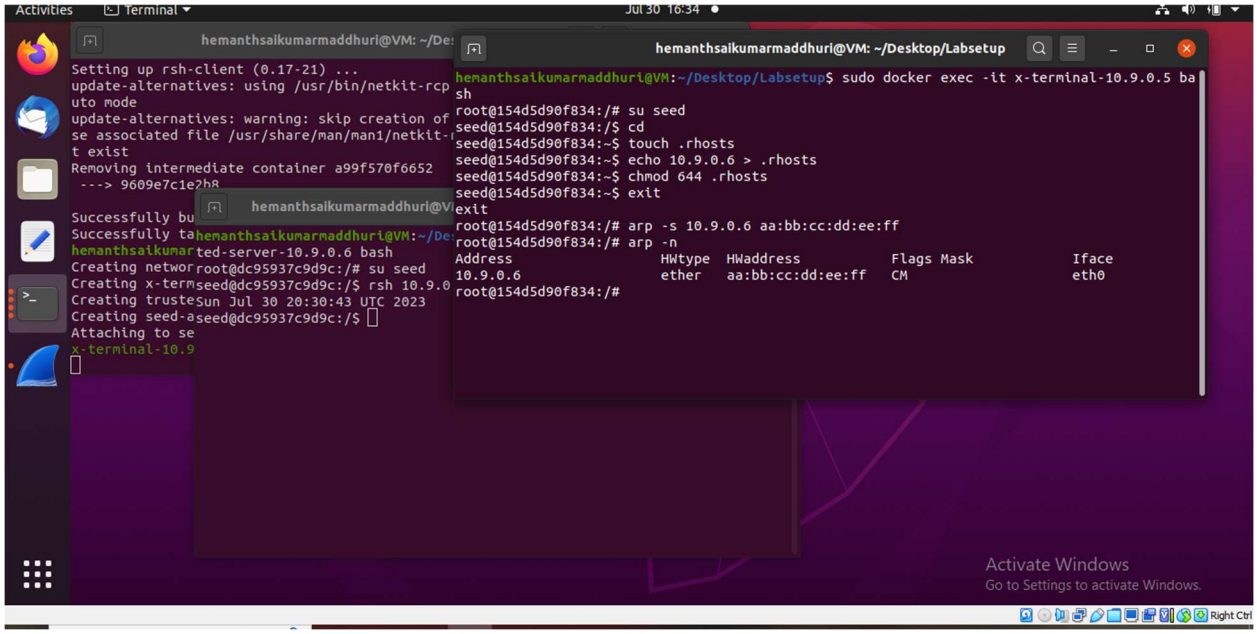


Verifying the trust relationship on trusted server as shown below.



Task 1: Simulated SYN flooding

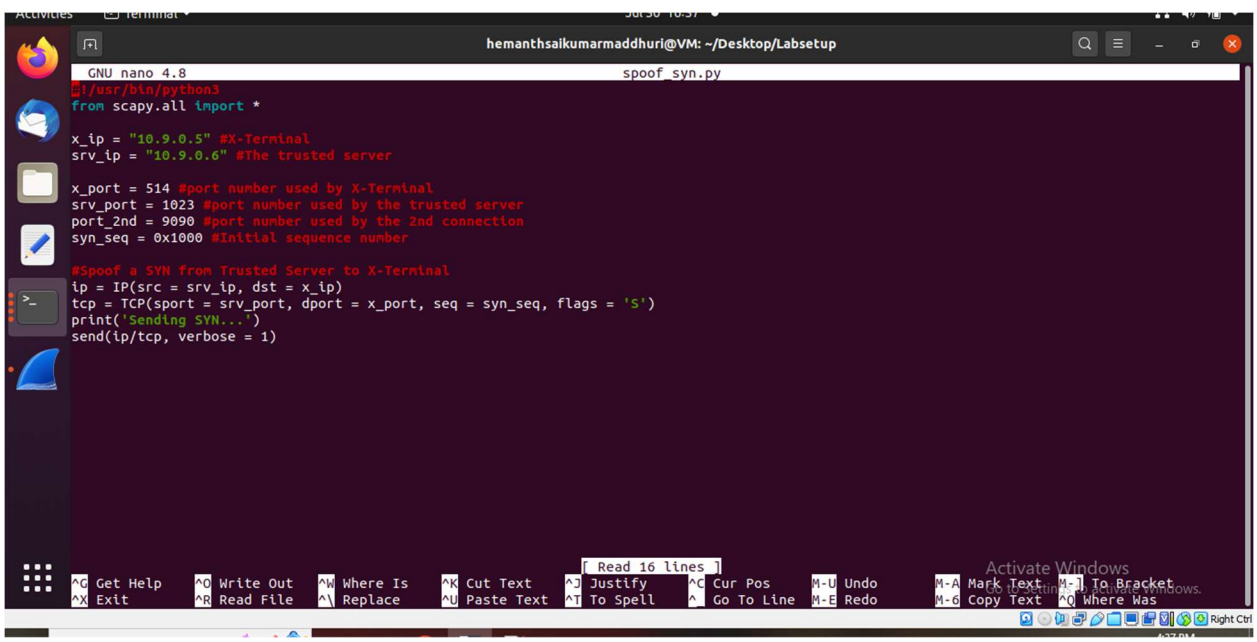
Performing Simulated SYN flooding on X-terminal and adding an ARP entry for 10.9.0.6(trusted server) with a fake MAC address.



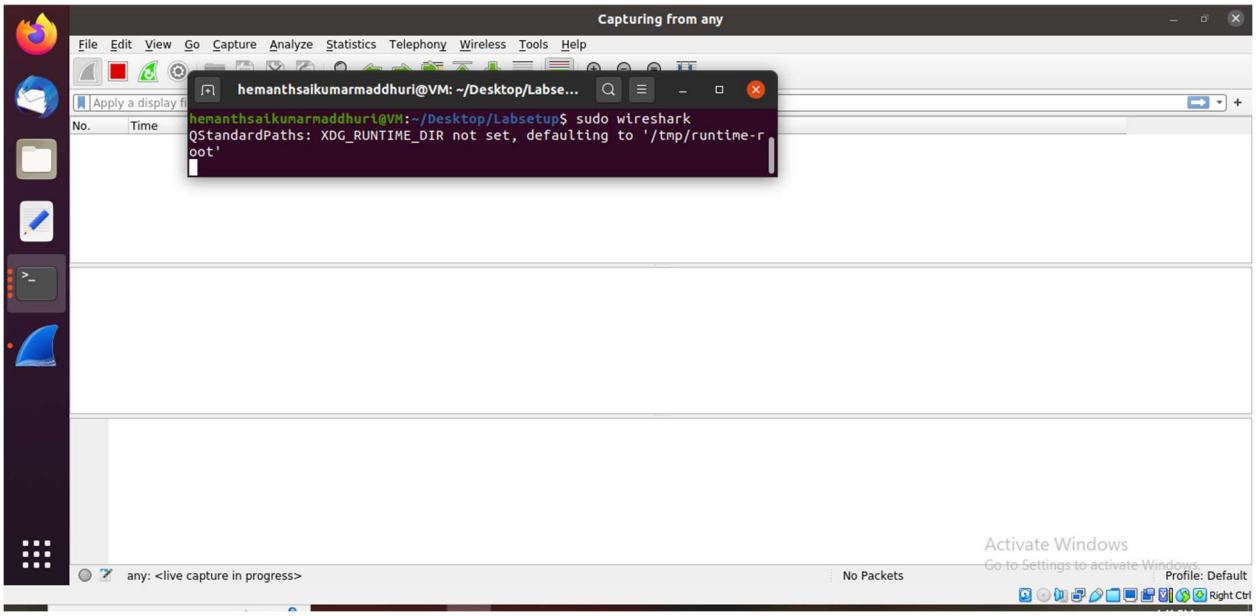
Task 2: Spoof TCP Connections and rsh Sessions

Step 1: Spoof a SYN packet

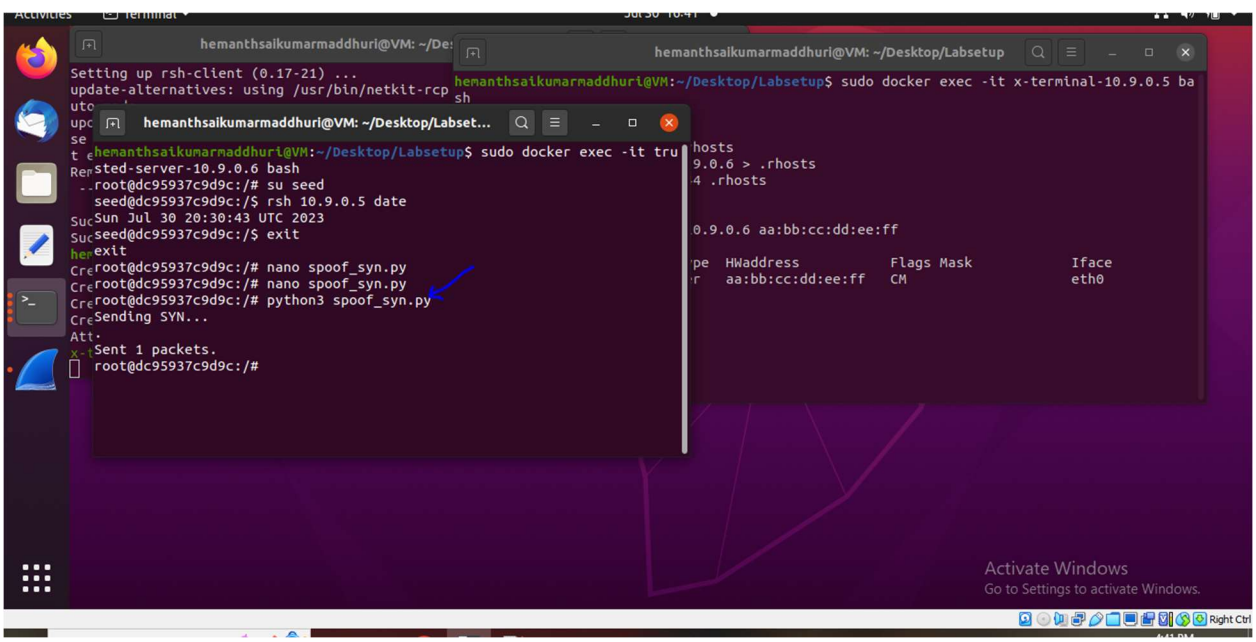
Here in this step, we have written a file called spoof_syn.py file to spoof a SYN packet.



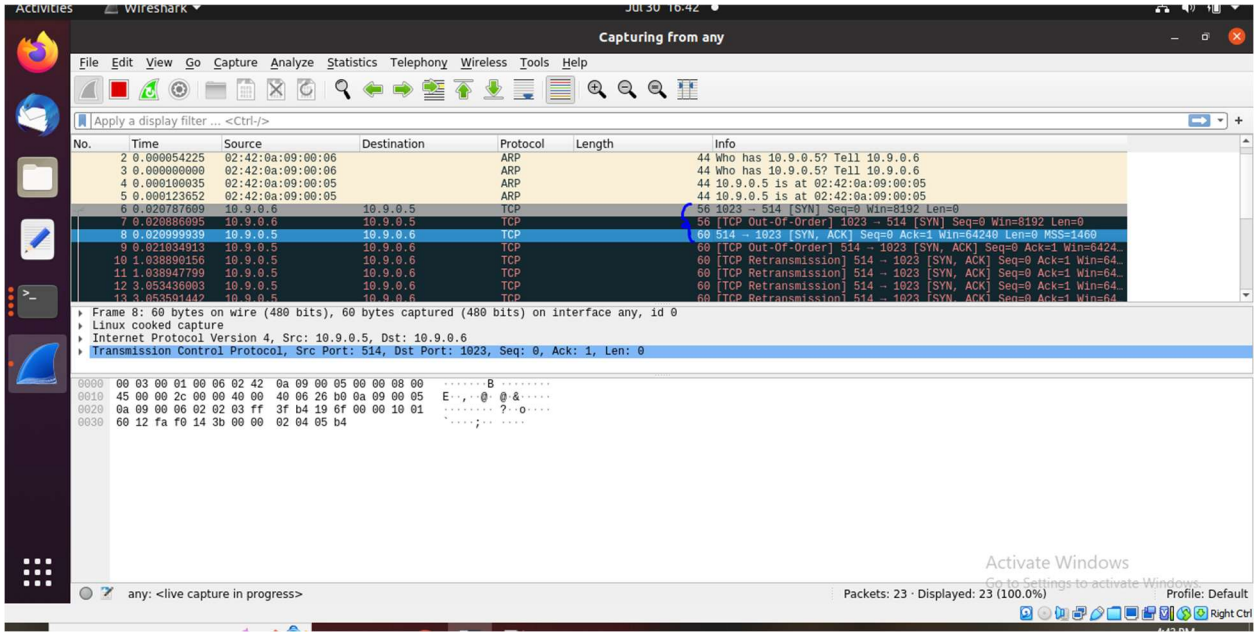
Starting Wireshark to capture the network packets.



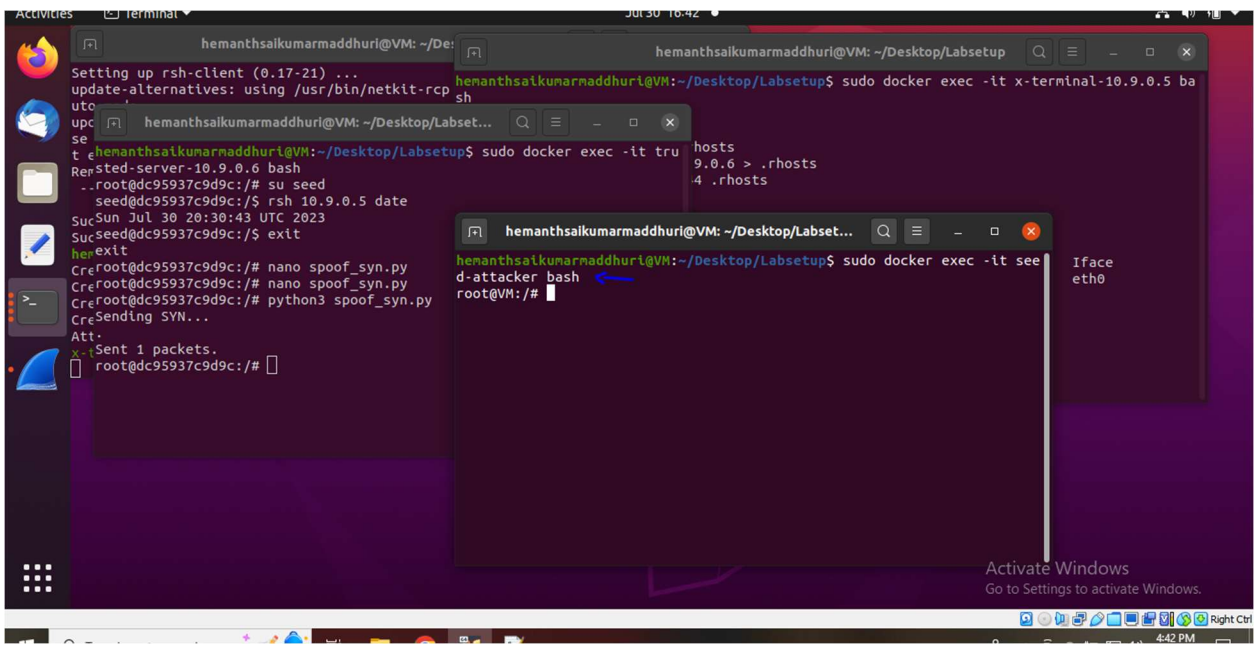
Later we run the python file using command **"python3 spoof_syn.py"** and observe that a packet was sent.



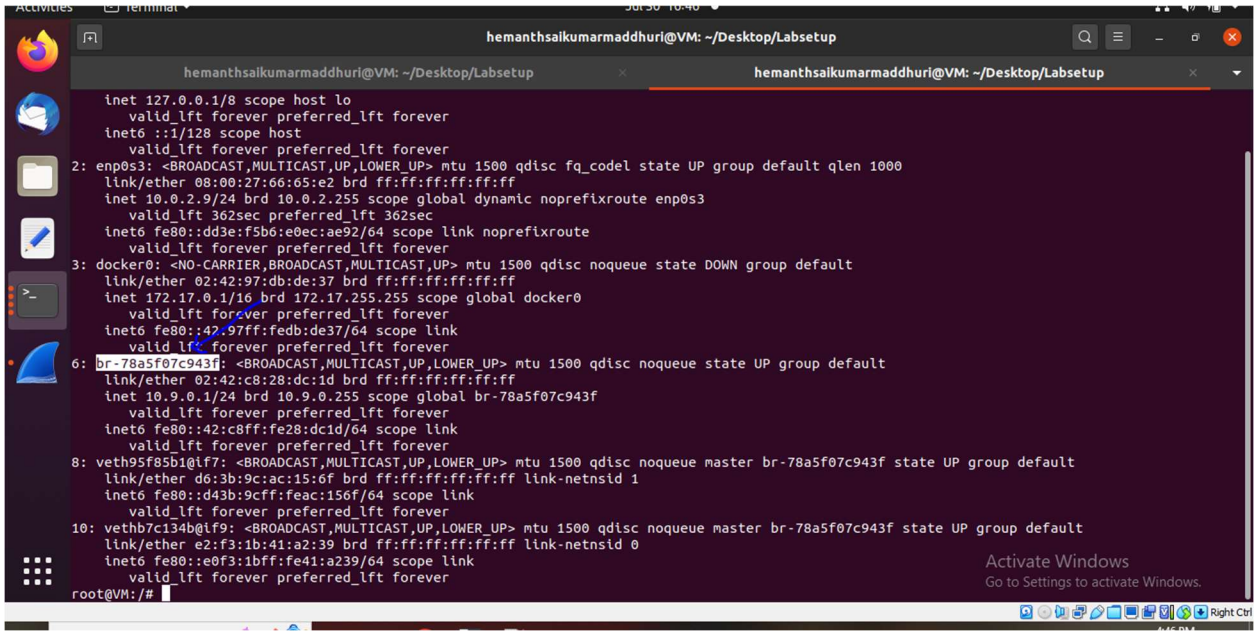
The Wireshark capture after sending the SYN packet is as shown below.



Logging into attacker machine using command **"sudo docker exec -it seed-attacker bash"** for next steps.

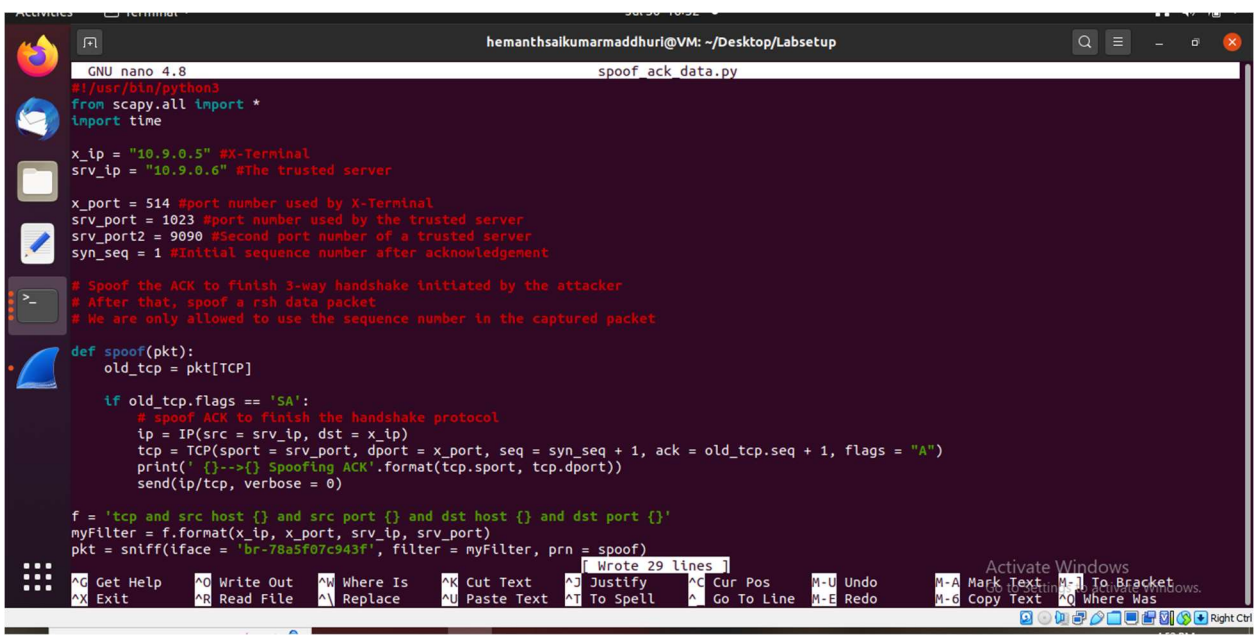


In the meanwhile, we check for the machine id using command **“ip addr”** for the SYN + ACK packet reply as shown below.

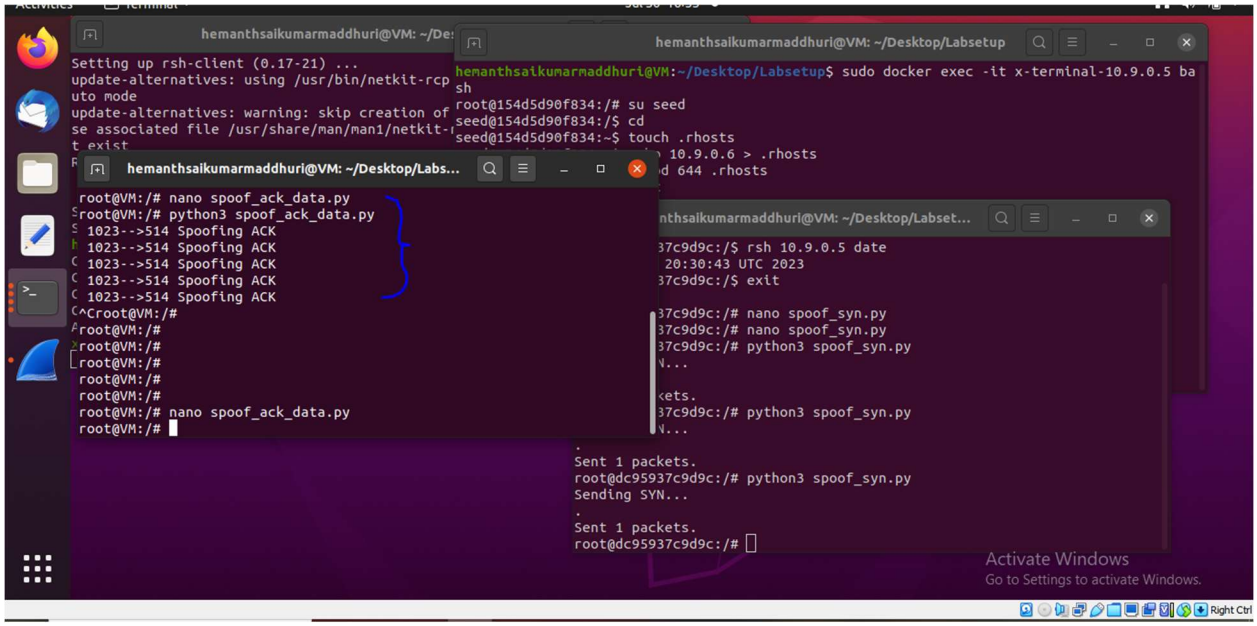


Step 2: Respond to the SYN + ACK packet

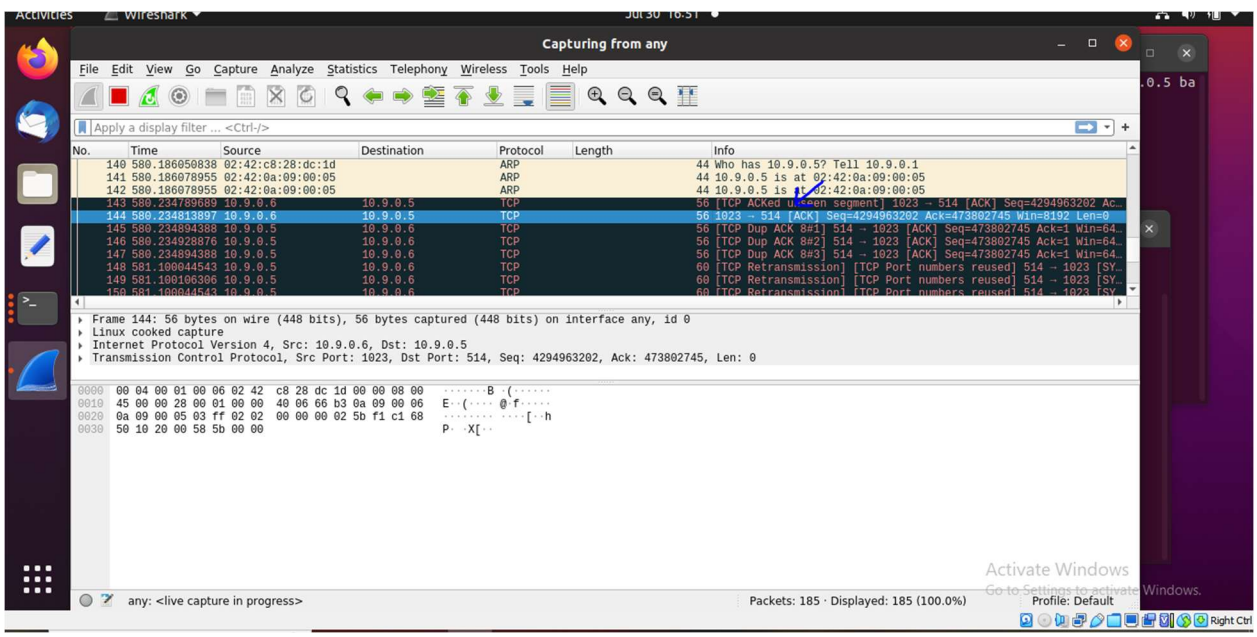
After X-Terminal sends out a SYN + ACK, we need to use trusted server needs to send out an ACK packet to complete the 3-way handshake protocol. So, we take the same sequence number as captured in the last packet and provide the same while responding to SYN + ACK.



Later we run the file to respond for SYN + ACK as shown below, and we get the output as spoofing ACK from 1023 to 514.

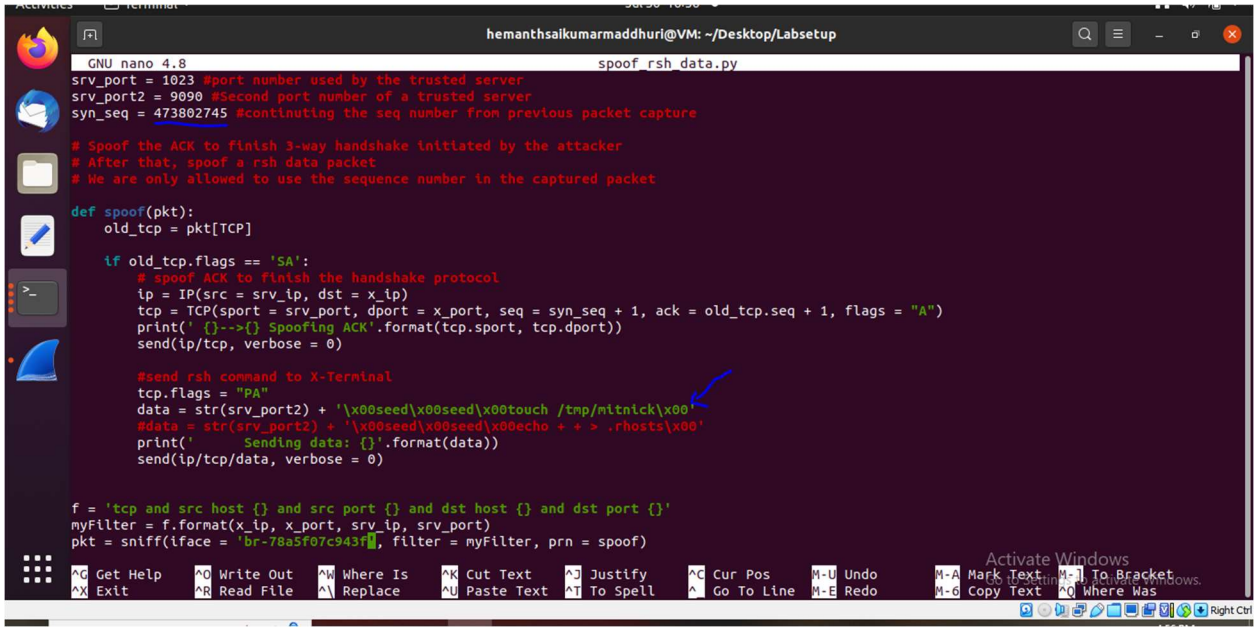


The same is captured in Wireshark as shown below.

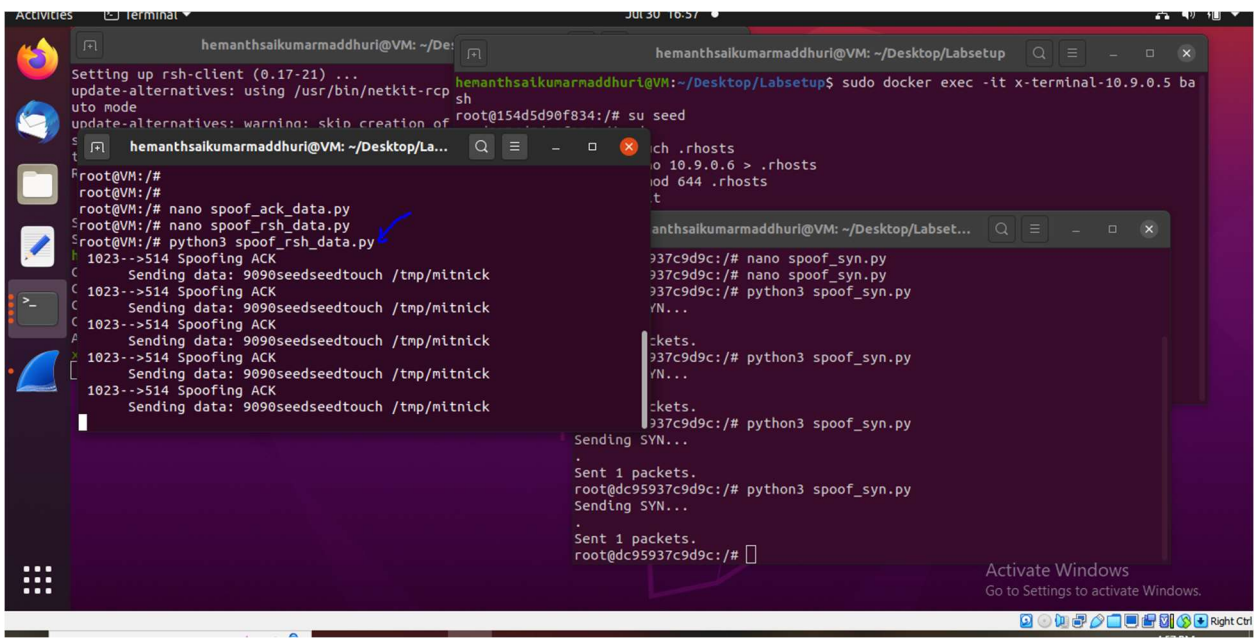


Step 3: Spoof the rsh data packet

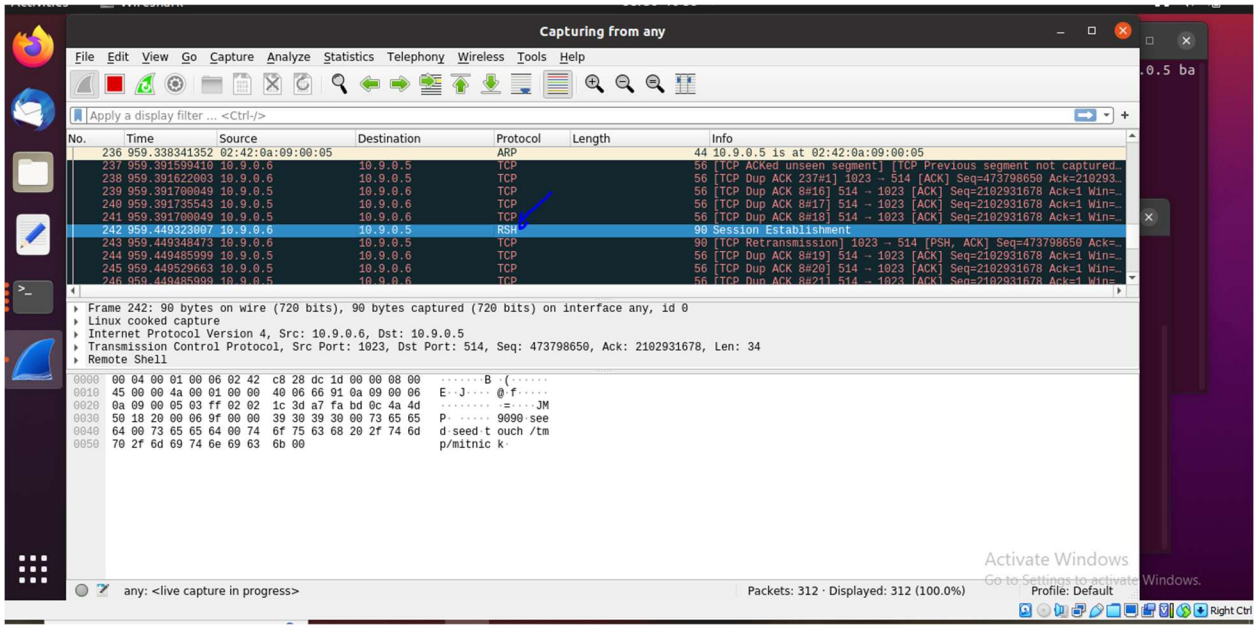
As we want to pass the data packet using the rsh command we now add the following part in a new file as spoof_rsh_data.py and the sequence number is followed as shown below.



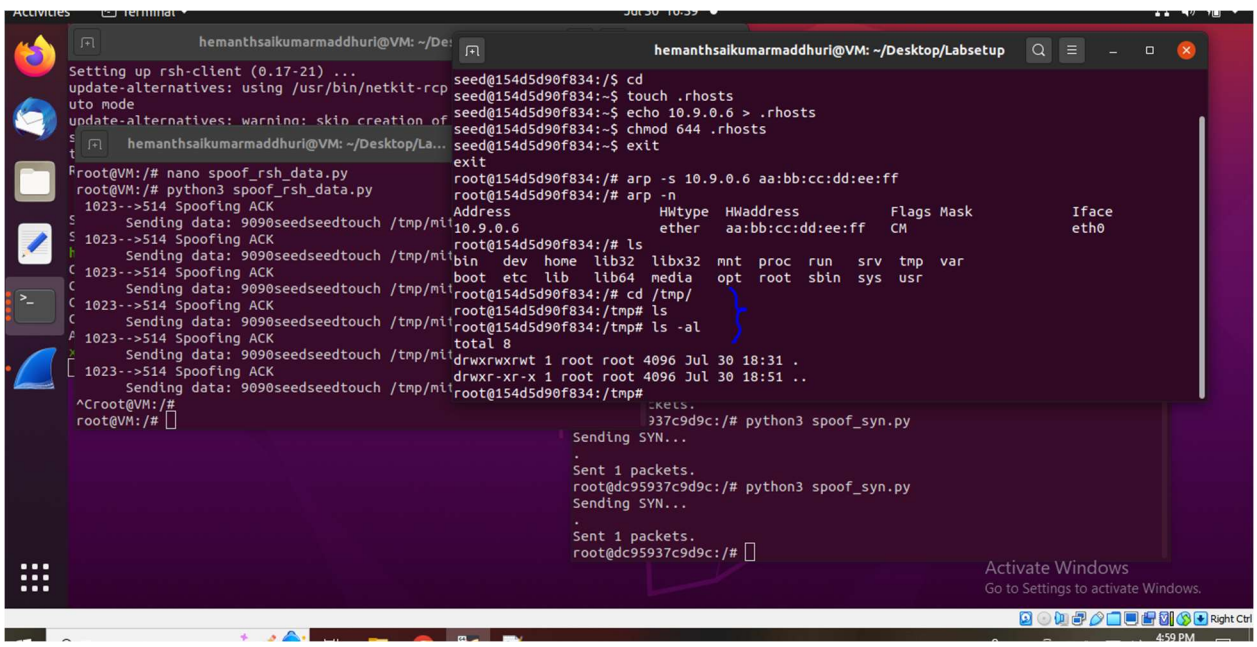
Now when we run the python file consisting of the rsh data packet command we see that data is spoofed and sent via 1023 -> 514 as shown below.



The Wireshark capture can be seen below which says that RSH connection is established as in screenshot below.

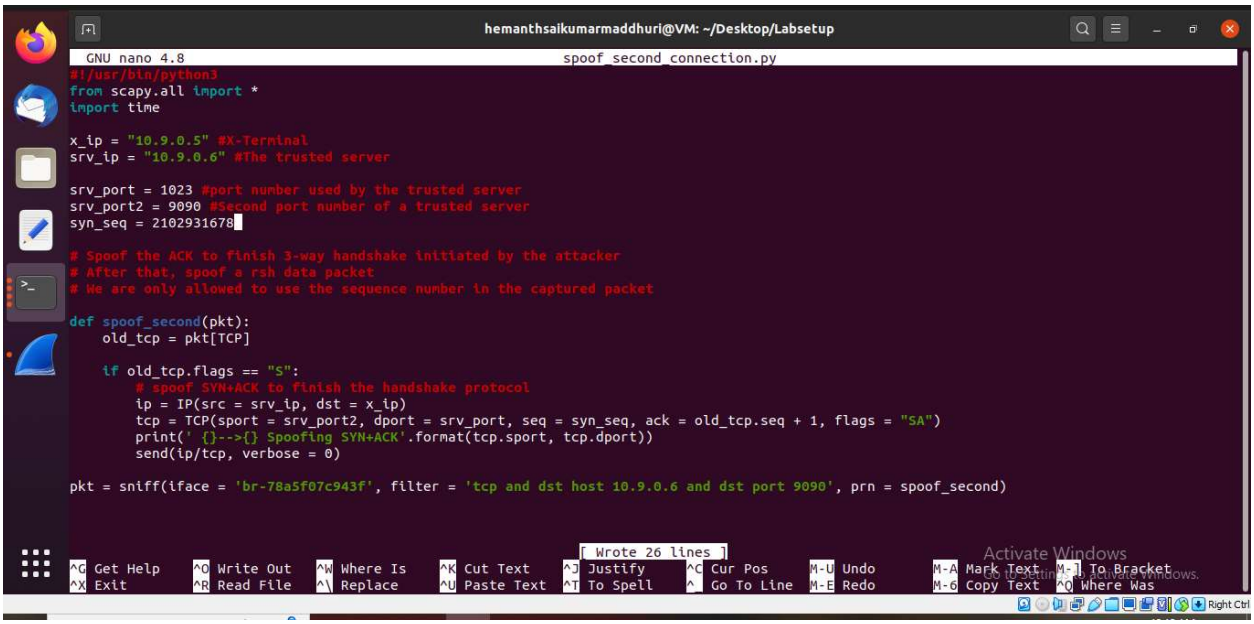


As instructed we check for Mitnick file but it is still created in X-terminal as we just established the connection and need to send the data again using rshd command.

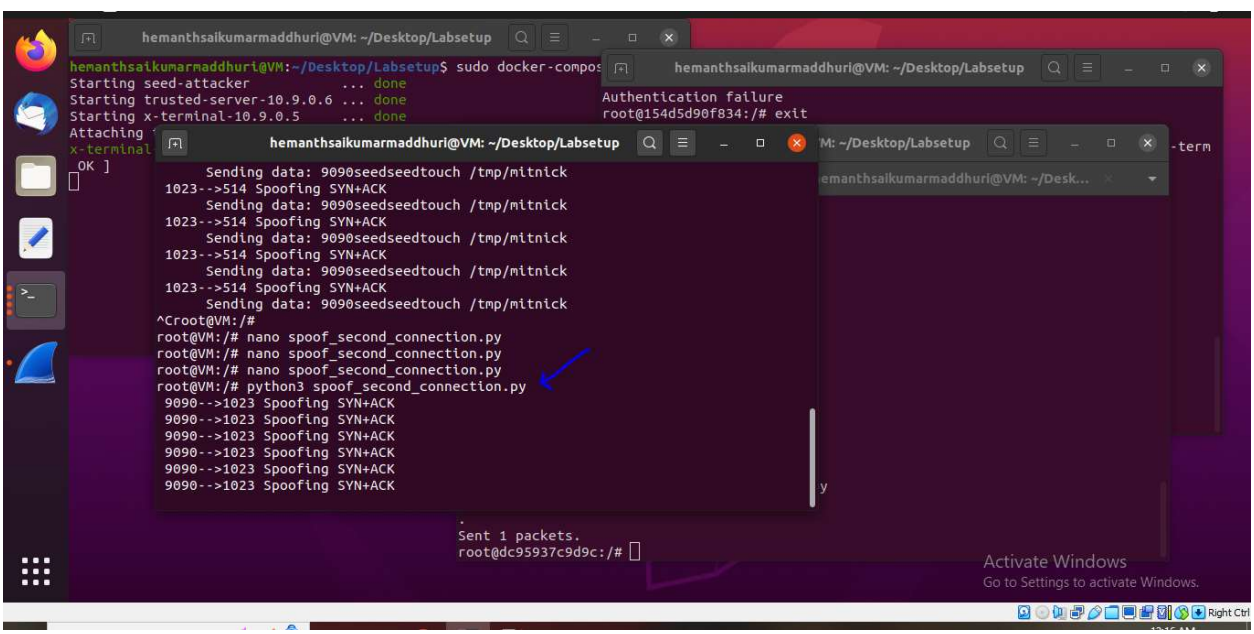


Task 2.2: Spoof the Second TCP Connection

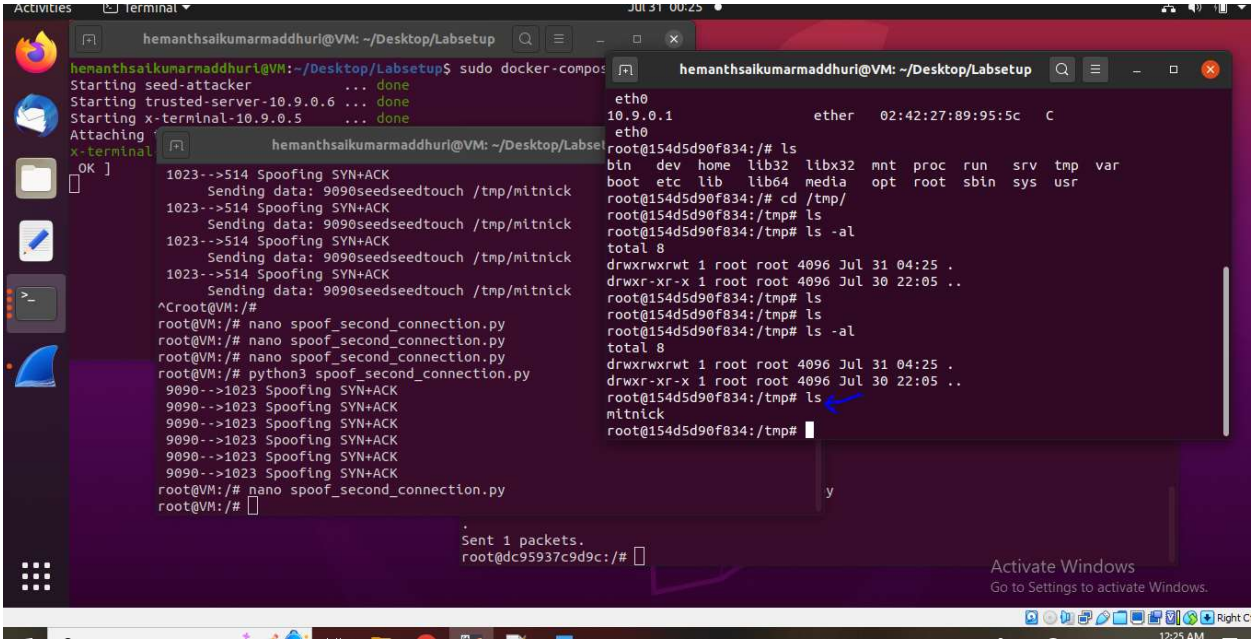
As instructed in the lab manual, we now write a new sniff-spoof program for setting up second connection as shown below.



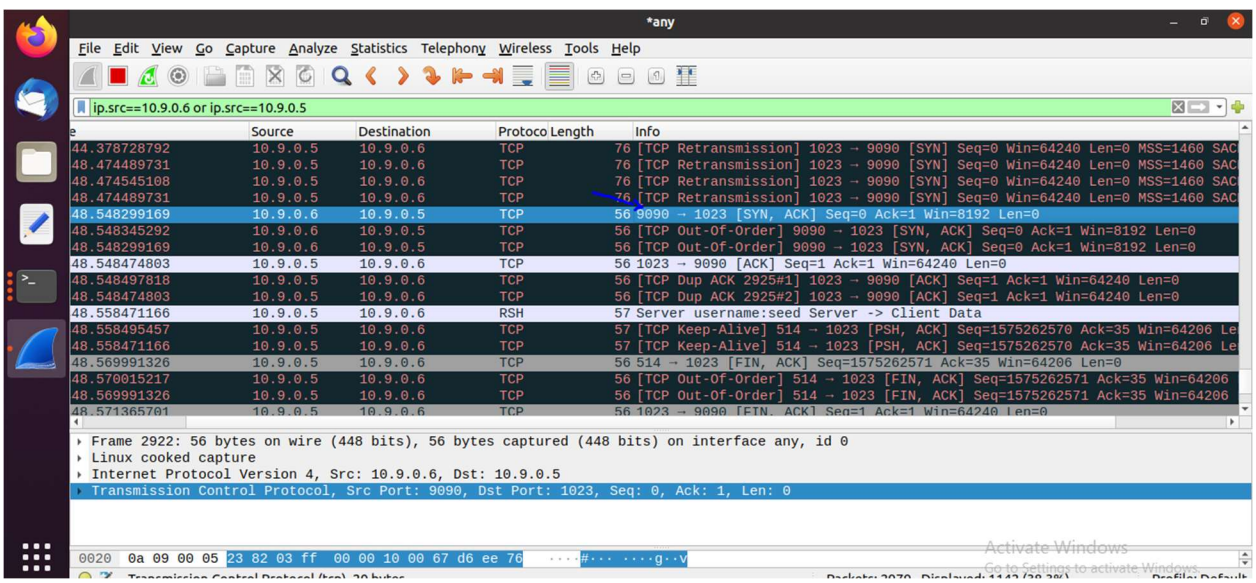
Later we run the code for second connection as shown below and can observe that spoofing is being done from the 9090 -> 1023.



Later we checked the /tmp folder in X-terminal for Mitnick file, and the attack was successful, Mitnick file was created in X-terminal as shown in screenshot below.



The Wireshark capture for the same is attached below and we can observe that the data is moved from 1023 -> 9090, 9090-> 1023 and the Mitnick file is created on X-terminal.



Summary:

Firstly, the backstory of Mitnick was quite interesting and encouraged me to do this lab. In the Mitnick attack lab I have been revised with interesting topics like 3-way handshake which I have learnt in computer networking. I had faced a few issues initially with following the sequence numbers and maintaining them to continue the packet transfer. Going through steps like SYN flooding, spoofing a TCP connection to sniff and spoof SYN+ACK steps was a bit confusing but later I sorted them out and successfully completed the attack.