Cyber and Network Security

Project – Mitnick Attack Lab

Submitted by:

Muhammad Osama Khalid

20I-1955 (MS-CNS)

Section: 1

Table of Contents

List of Figures

Part-1 Setting up the lab

In this part, I am going to set up the Attack Lab in which I will perform the Mitnick Attack in net parts.

1. First, I download the Seed Virtual Machine from the SEED Lab Website and then open the virtual box and click on the add button. Then I add details of my VM and click Next. After that, I set the memory size (RAM) to 1 GB and click next. After that from the options, I select “Use an existing virtual hard disk file” and go to the path where I download the Seed VM and select that. After that, I click create to create my VM.

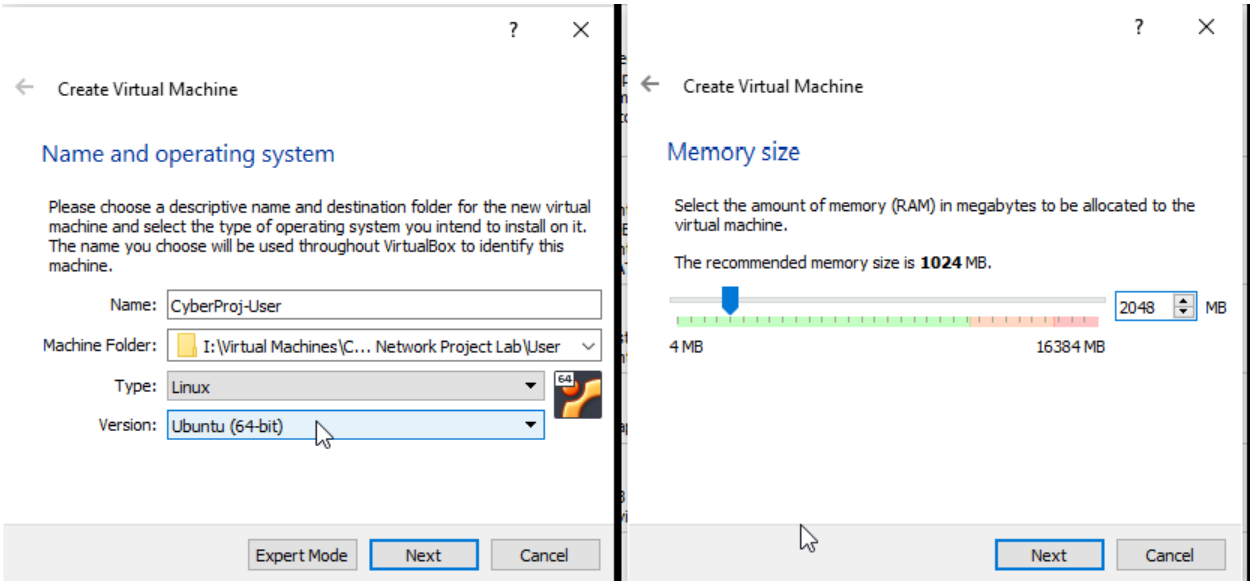

Figure 1: Creating the Virtual Machine

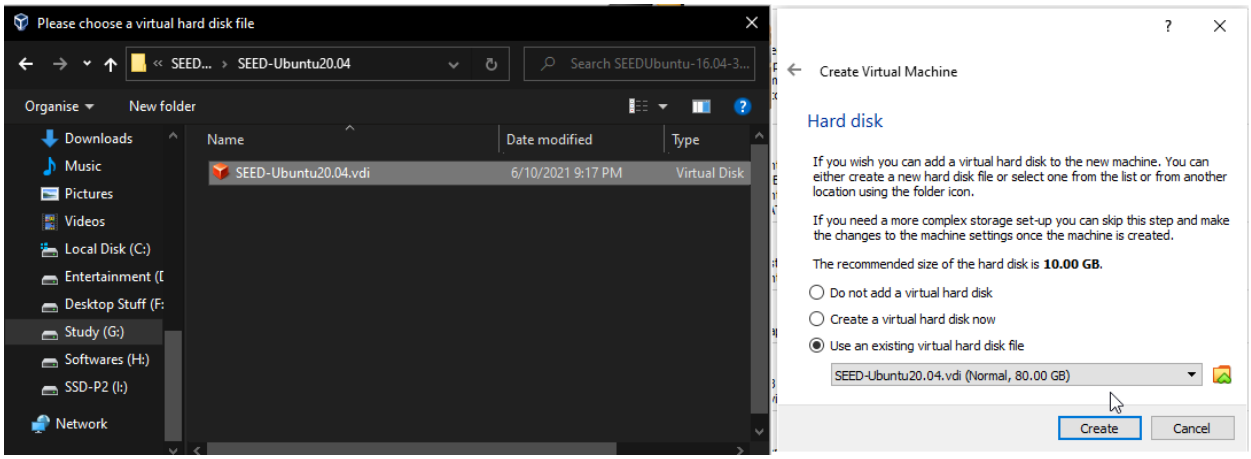

Figure 2: Creating Virtual Machine

2. After that, I go to the VM settings and click on the shared folder to create the shared folder between the host and the virtual machine so that every time I put something in that folder, it can be accessed by both VM and host machine.

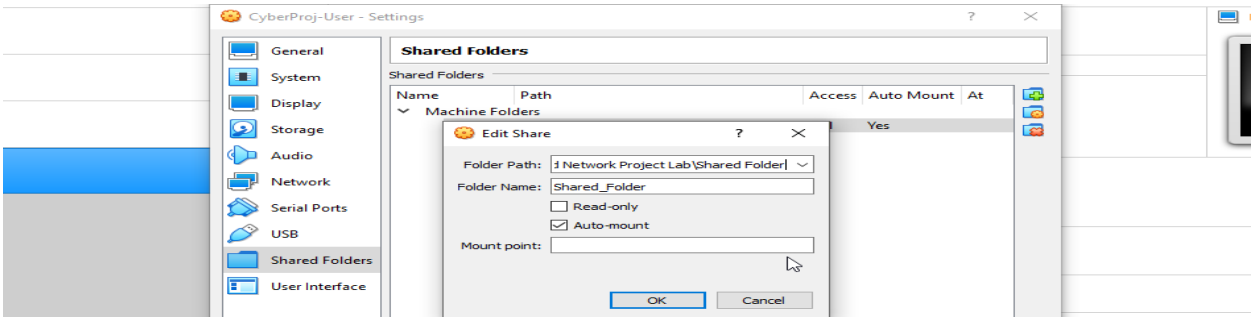

Figure 3: Go to Virtual Machine Shared Folder Setting

3. After that, I start the virtual machine, and then to mount the shared folder to the home directory of the Virtual machine I create a folder called “Share” in the home directory and then mount the shared folder to this “Share” folder.

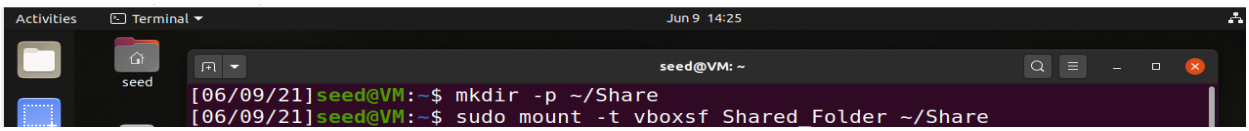

Figure 4: Mount the shared folder to the home directory folder

4. Then I power off the VM and go to the virtual box setting and select Network from the tab on the left panel. Then I click on the “+” button to create a new NAT Networks adapter. After that, I open the virtual machine setting, select Network from the tab on the left panel, and staying on Adapter 1 and under enable network adapter I click on the “Attah to” drop-down menu and select NAT Network adapter that I just created and click ok.

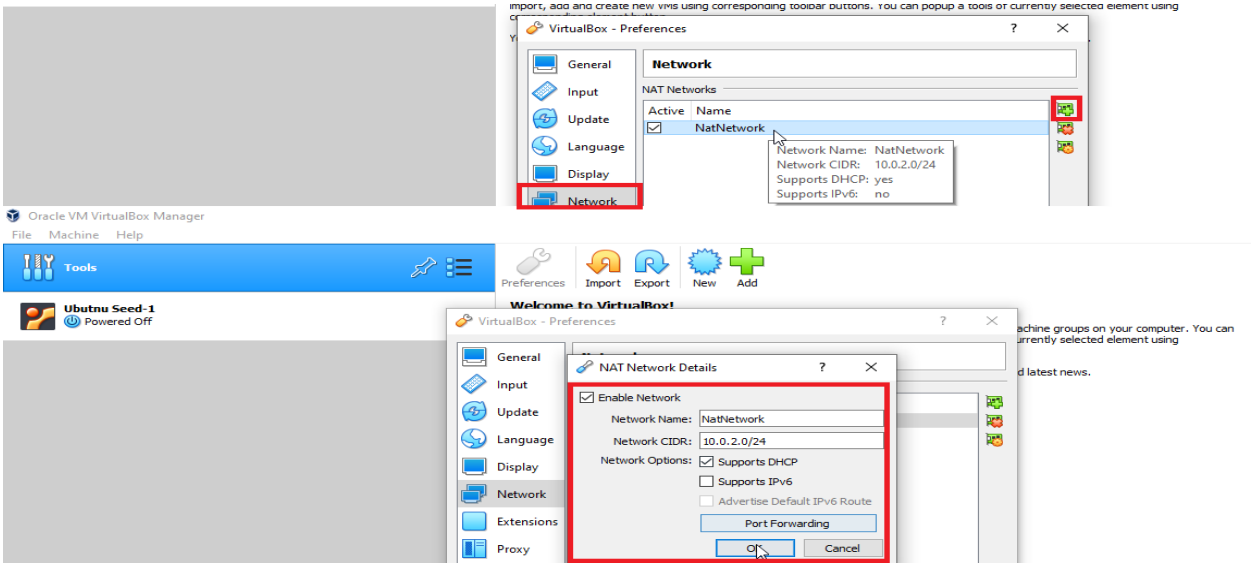

Figure 5: Creating new Nat Adapter from Virtual Box Setting

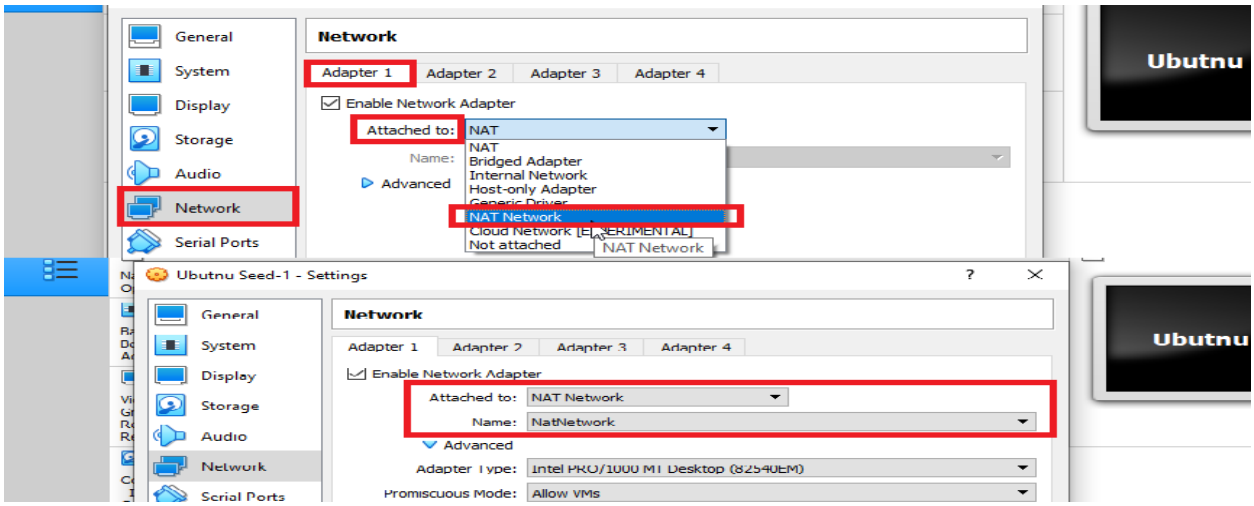

Figure 6: Adding Nat adapter in Virtual Machine setting

5. Then I right-click on the virtual machine and then click on the clone. Then in the popup dialogue box, I enter the clone machine name and select the path where I want to store my clone virtual machine and then I click on the Next button. Then from the clone type options, I select “Full done” and click on the clone button. I repeat this step one more time so that I have one actual machine and two clone machines.
6. Then I change the names of my virtual machines to CyberProj-User, CyberProj-Attacker, and CyberProj-Server.

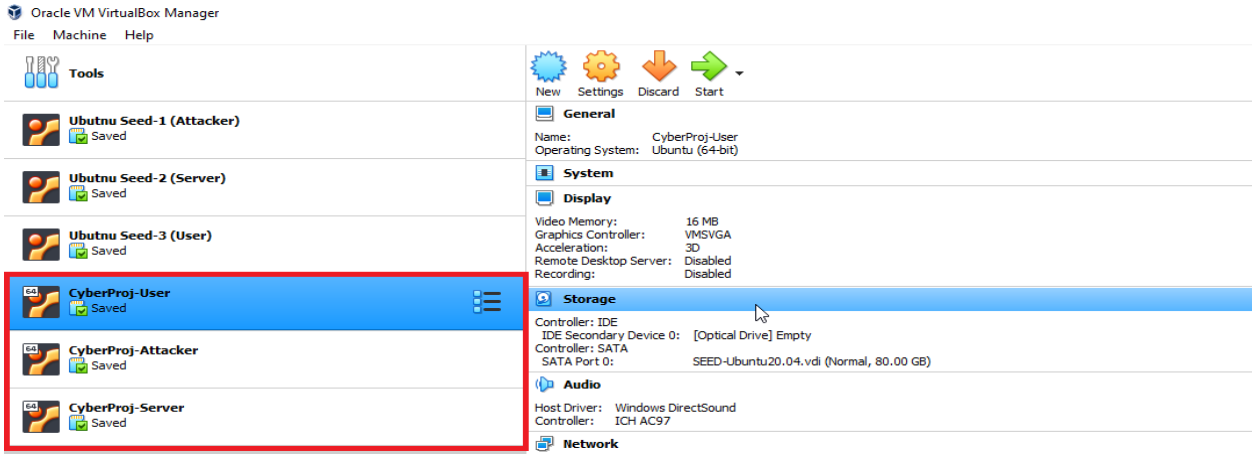

Figure 7: Virtual Machines to be used in Attack Lab

7. After that, before doing anything I also change the name of the host machines to the x-terminaluser, Server and Attacker by using the command “**sudo hostnamectl set-hostname newNameOfTheHostMachine**”. I perform this because while performing the lab it will become easy for me to check on which machine I am currently on and doing my task. I change the hostname of the machine as follows
    - On the user machine, I change the hostname to “x-terminaluser”.
    - On the server machine, I change the hostname to “Server”.

- On the attacker machine, I change the hostname to “Attacker”.

8. After that, I run all my virtual machines and open the terminal to check the IP address of all my virtual machines.

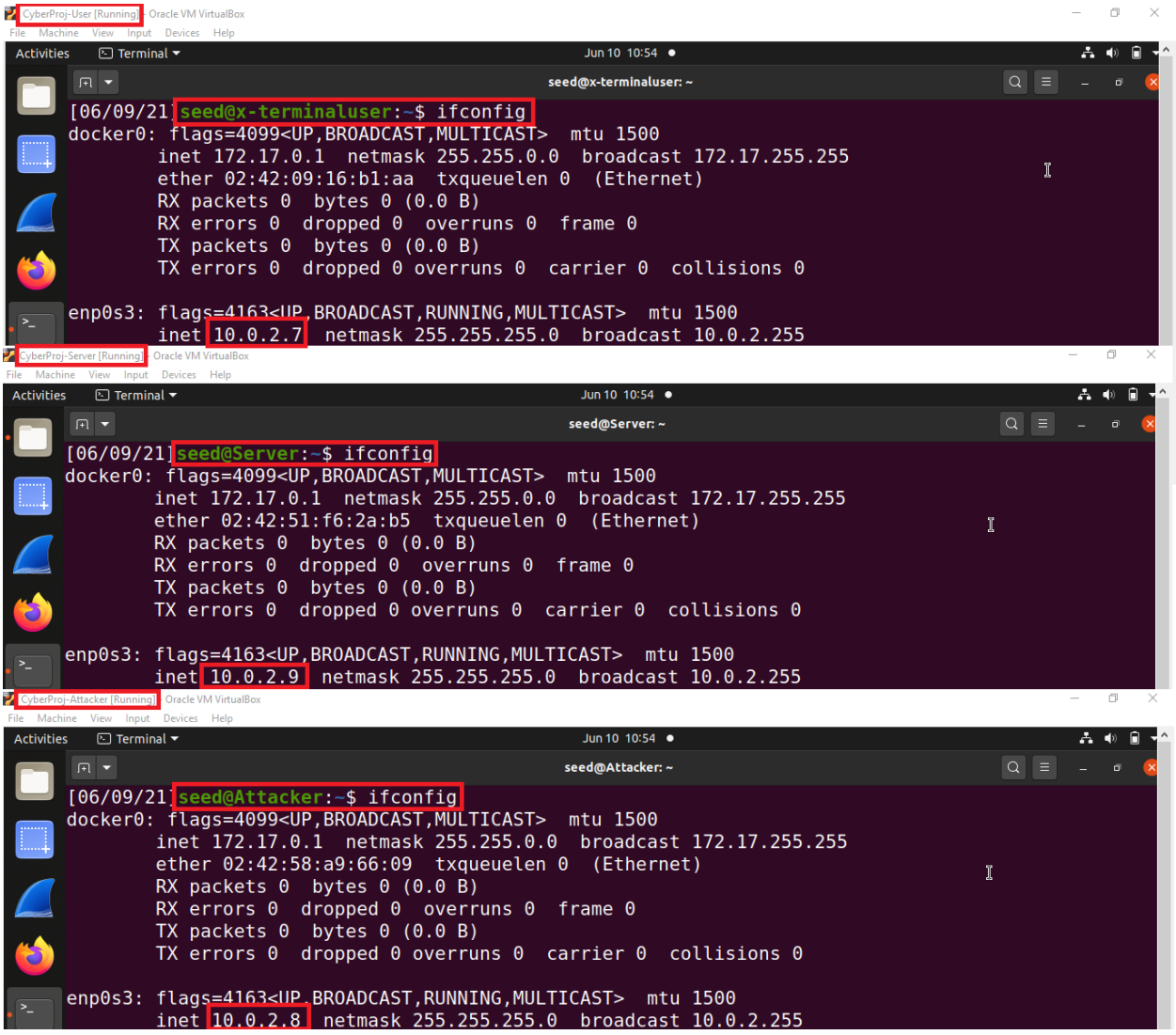

Figure 8: IP address of my virtual machines

9. Then from the user machine, I open the terminal and ping the server machine to check that all the machines are connected or not.

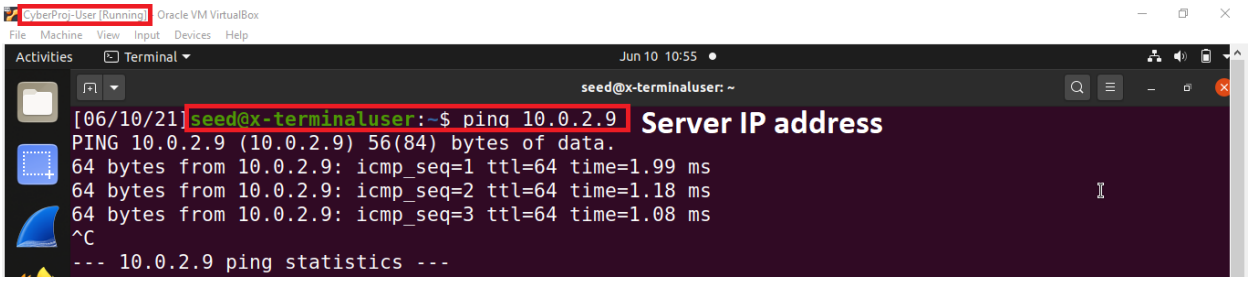

Figure 9: Ping request to Server machine

10. IP addresses of my machines
    - User-machine IP: 10.0.2.7
    - Server machine IP: 10.0.2.9
    - Attacker machine IP: 10.0.2.8
11. Now to perform the Mitnick attack I first need to install the insecure version of the remote shell (rsh) program called rsh-redone for client and server on virtual machines. On the user machine, I install both client and server programs of rsh and on the server and attacker machine, I only install the client program of rsh using these commands.
    - For server program: sudo apt-get install rsh-redone-server
    - For client program: sudo apt-get install rsh-redone-client

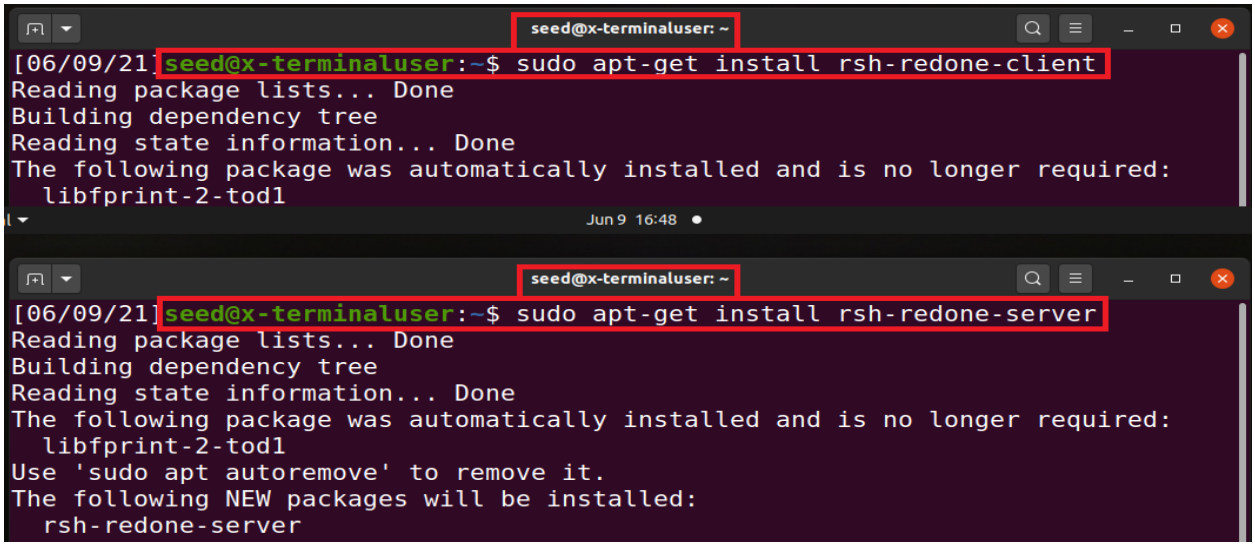

Figure 10: Installing rsh client and server program on User-machine

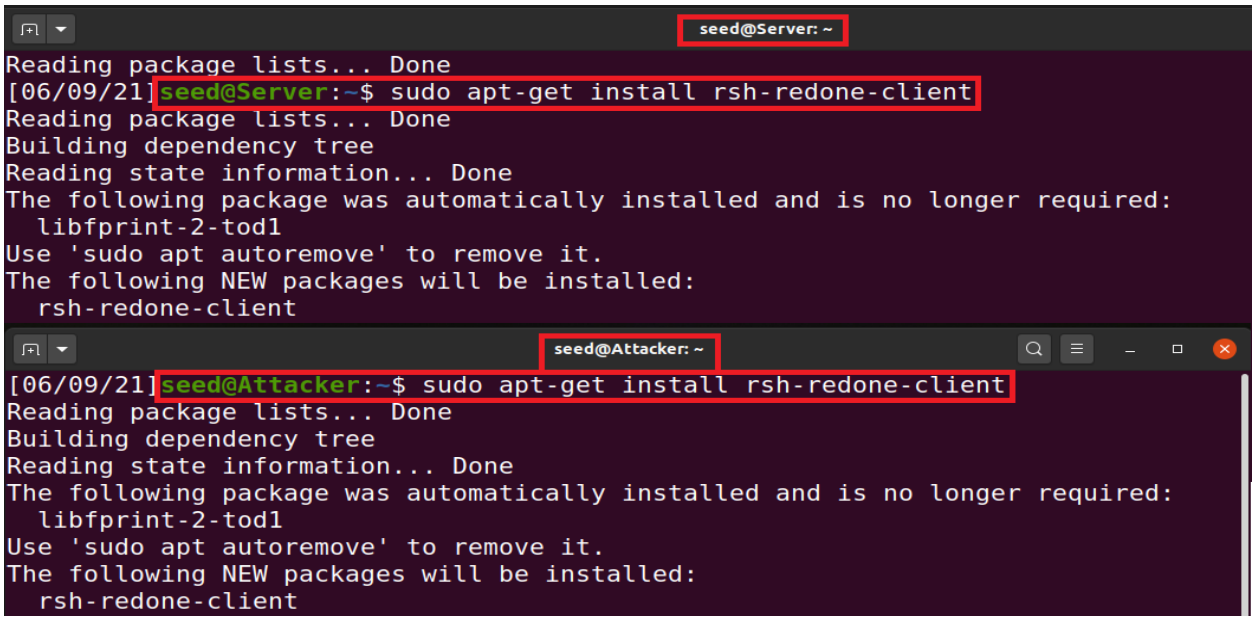

Figure 11: Installing rsh client and server program on User-machine

12. I install the rsh server program on the user machine because in the actual attack the server machine is allowed to log in to the user machine without needing the password of the user machine. To do that I need to create a ".rhosts" file on the user machine and enter the IP address of the server machine in that file. Now rsh server on the user machine will use this file to authenticate the server machine by checking the server IP address in the ".rhosts" file and if it found the IP address of the server machine in that file then the rsh server allows the server machine to login to the user machine without needing the password of the user machine.

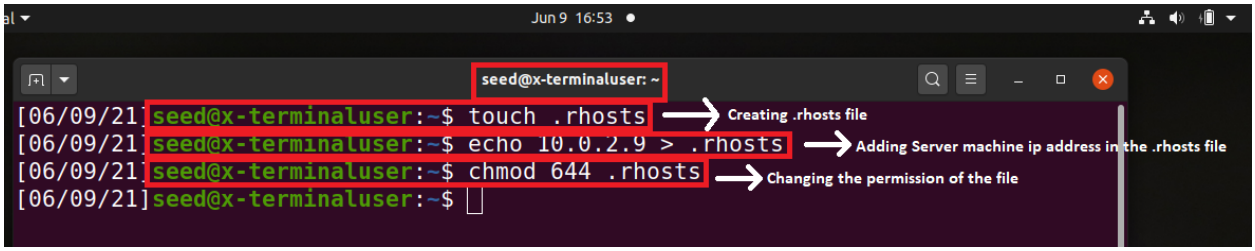

Figure 12: Creating .rhosts file and adding server IP to that file

13. Now to check that the server can log in to the user machine without need the password of the user machine I run the rsh command to ask the user machine its current date and time. If the rsh server on the user machine finds the IP address of the server machine in the ".rhosts" file it replies to the server machine with the current date and time of the user machine otherwise it will display an "Authentication Failure" error.

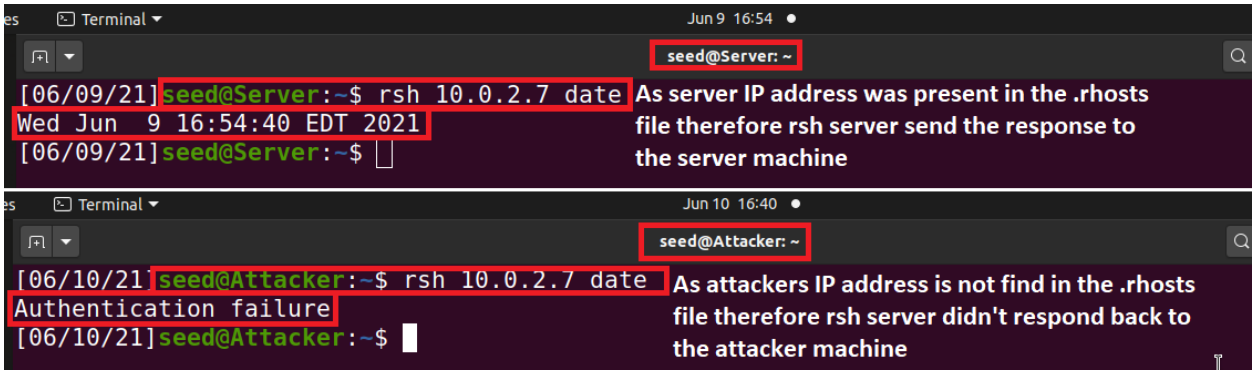

Figure 13: Checking the configuration

Part-2 Simulating SYN Flooding

In this task to simulate the effect of SYN Flooding attack on the server machine, I need to disconnect the server from the network and add the server's IP and MAC address in the x-terminal's (victim machine) ARP cache table because when I send the SYN packet from the attacker machine on the behalf of the server to the x-terminal, x-terminal will respond with the SYN-ACK packet but before sending that packet it will check its ARP cache table to find the MAC address of the server machine. If x-terminal finds the server's MAC address in the ARP cache table it will send the SYN-ACK packet but if it didn't find the server's MAC address then before sending the SYN-ACK packet, it first sends ARP request packet to the server to ask for its MAC address and as the server is disconnected from the network it will not respond to the x-terminal request so x-terminal will not be able to send the SYN-ACK packet and our attack will not work therefore we need to add server's MAC address in the x-terminals ARP cache table.

Now to add the server's MAC address in the x-terminal ARP cache table I first ping the server machine from the x-terminal and then use the "**arp -an**" command to check the entries in the ARP cache table and find that server's MAC address is temporarily present in that table. Now here is the issue that when I disconnect the server from the network, after some time on x-terminal OS delete the MAC address of the server machine from the ARP Cache table because OS fails to communicate with the server machine. To overcome that issue, I need to permanently add the server's IP and MAC address in the ARP cache table on the x-terminal and to do that I use the command "**sudo arp -s 'server IP address' 'server MAC address'**". This command permanently adds the server's MAC address in the x-terminal's ARP cache table. To confirm that I again use the "**arp -an**" command and find that server's IP and MAC address is stored permanently in the ARP cache table.

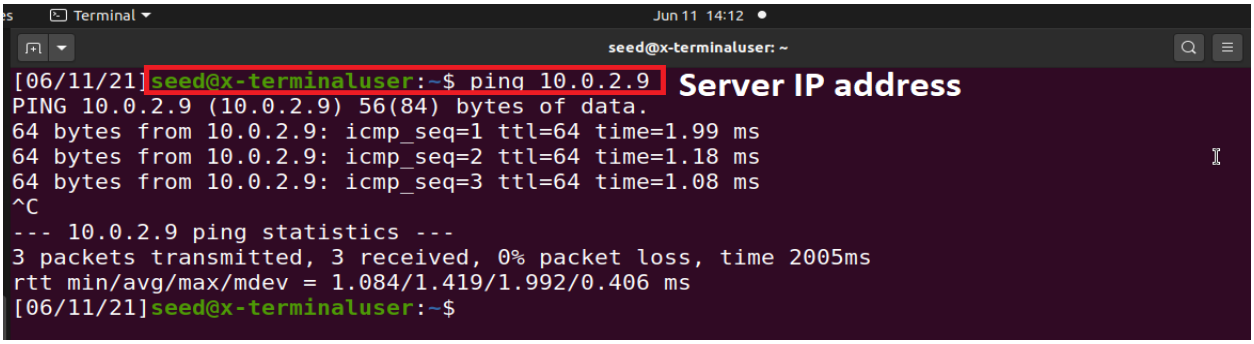

Figure 14: Ping the server machine from x-terminal

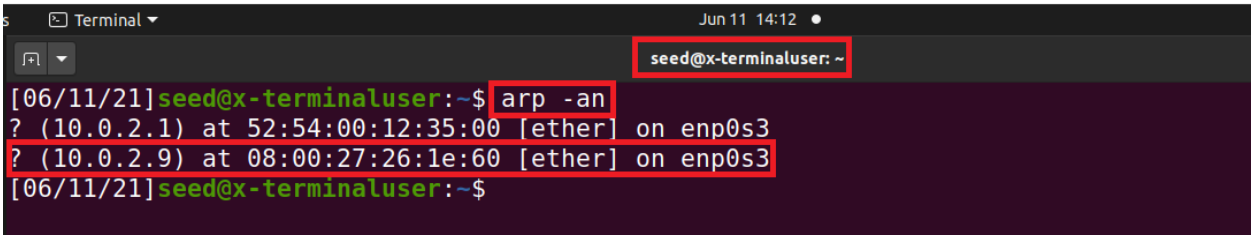

Figure 15: After running the ping command server's mac address temporarily stored in the ARP cache table

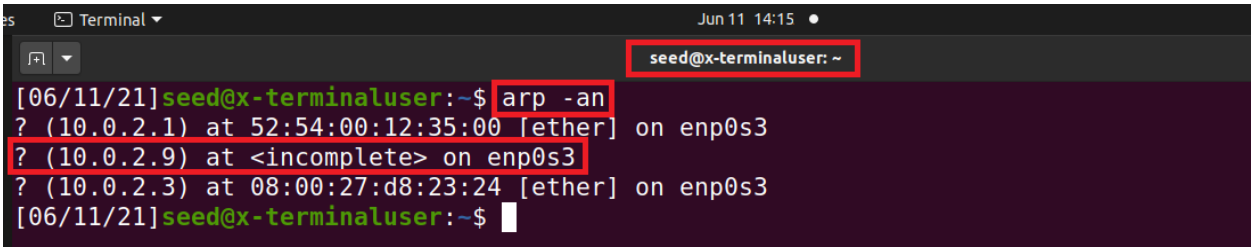

Figure 16: OS deletes MAC address of the Server machine as it is disconnected from the network

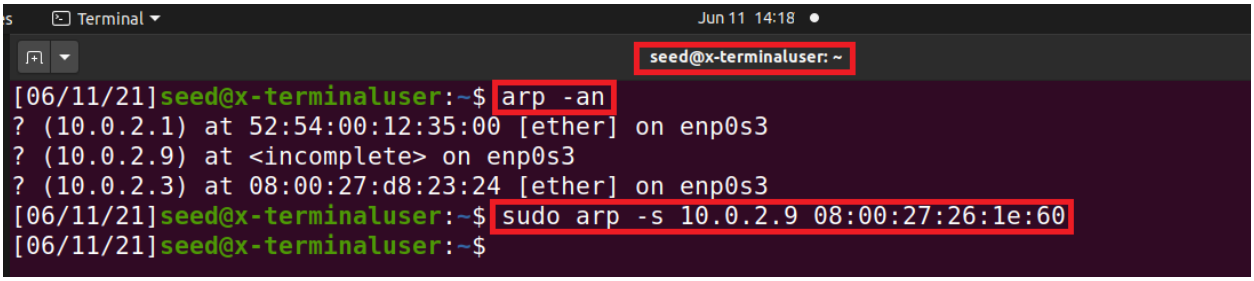

Figure 17: Adding Server's IP and MAC address permanently in the ARP Cache Table

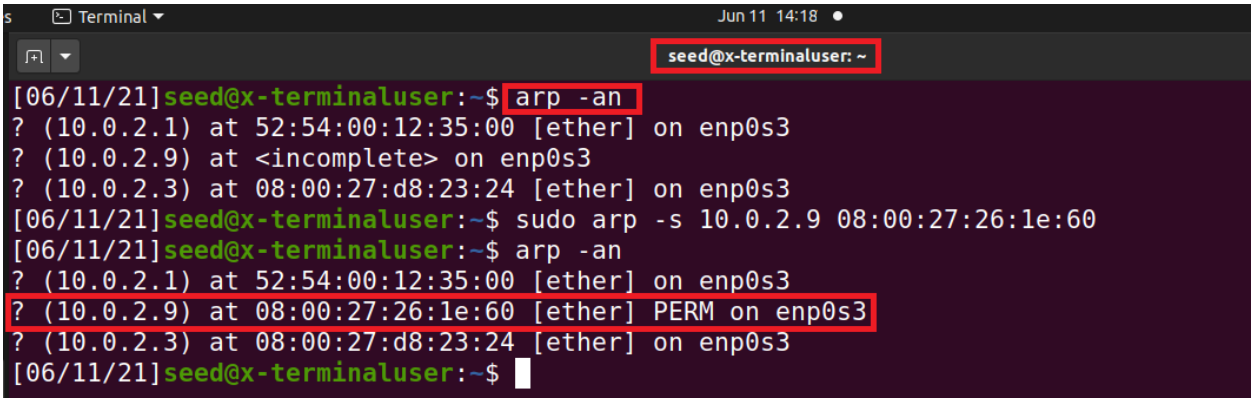

Figure 18: Server's IP and MAC address permanently stored in the ARP Cache Table

Part-3 Spoof TCP Connections and rsh Sessions

As the server is disconnected from the network and the server's IP and MAC address is permanently stored in the x-terminal's ARP cache table, now we can launch the attack on the x-terminal (victim machine). To do that I need to establish the rsh connection with the x-terminal on behalf of the server so that I will able to run commands on the x-terminal and as rsh runs on the TCP therefore I first need to create the TCP connection between the server machine and x-terminal. Now to create the TCP connection I first need to spoof the SYN packet from the server to the x-terminal to get the sequence number from the response by the x-terminal.

Task-1 Spoof SYN Packet

Now to initiate the three-way handshake of the TCP connection I first need to spoof the SYN packet and to do that I send the spoofed SYN packet to the x-terminal. I write the python code through which I send the spoofed SYN packet to the x-terminal by pretending that it is sent by the actual server to the x-terminal.

Figure 19: Attacker: Code used to send the spoofed SYN packet

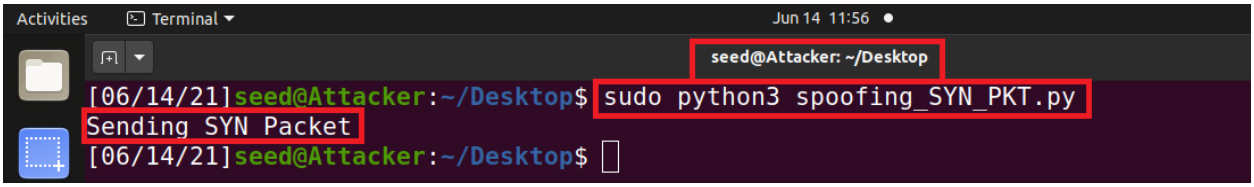

Figure 20: Attacker: Run the Code

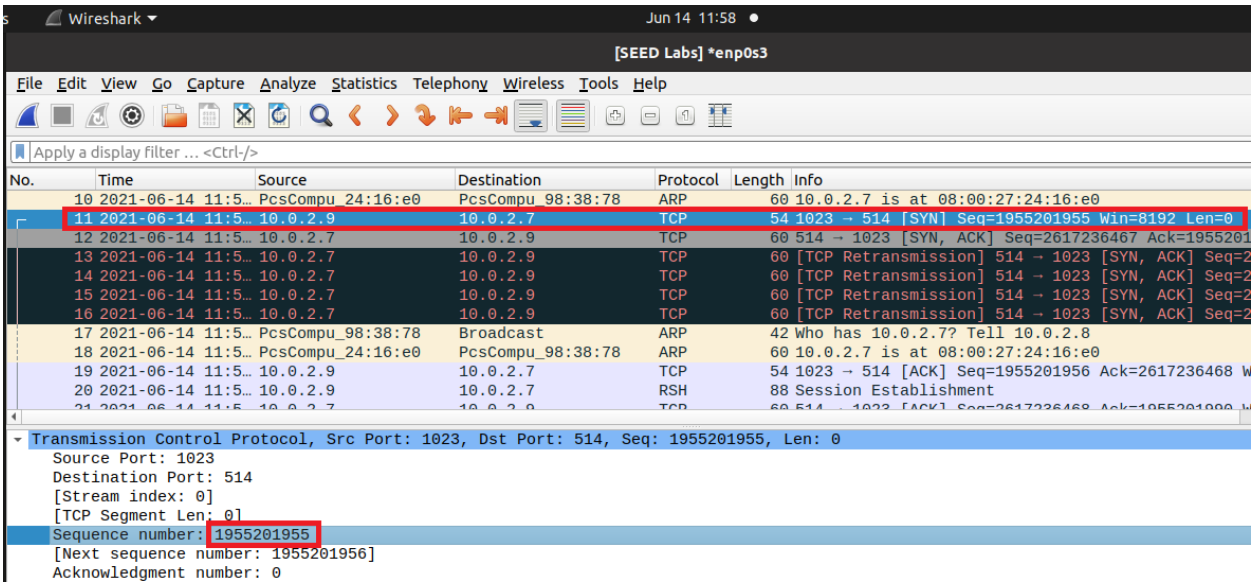

Figure 21: Attacker: SYN packet send to x-terminal successfully

Now as the x-terminal receives the SYN packet, it responds with the SYN-ACK packet to the server.

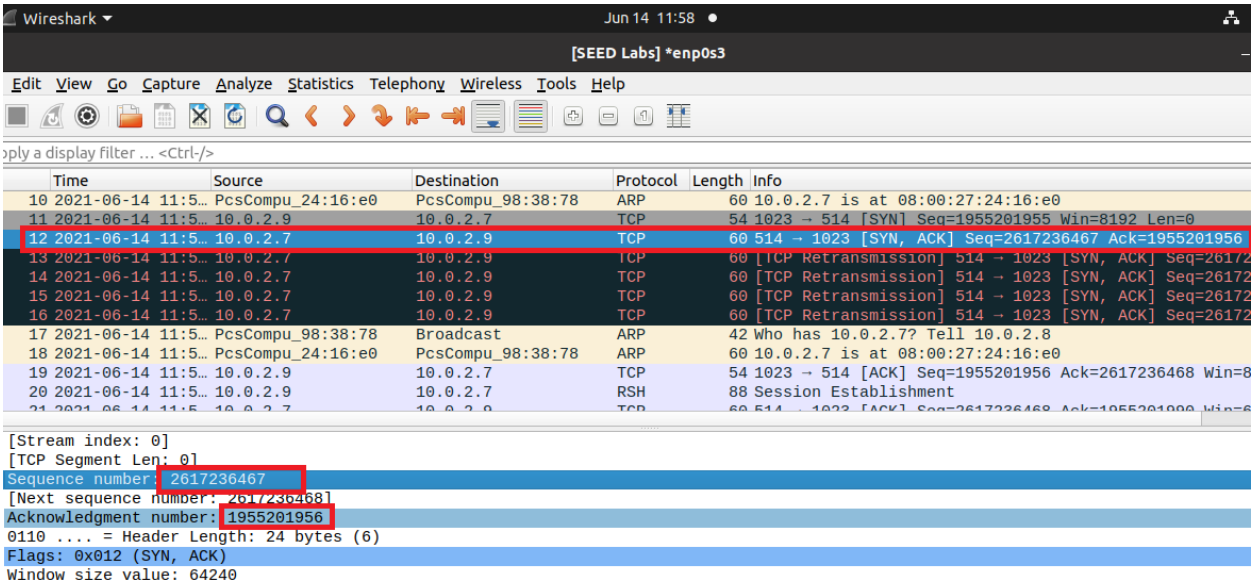

Figure 22: X-terminal respond with SYN-ACK packet

Task-2 Respond to SYN-ACK Packet and also spoof the rsh data packet

As the x-terminal responds with the SYN-ACK packet attacker sniff that packet, extracts the sequence number of the x-terminal, and responds with the ACK packet to complete the three-way handshake. As the three-way handshake of TCP connection completes attacker on behalf of server tells the x-terminal that I want to create an rsh connection with you, I am going to listen on port 9090 and want to run that command "**touch /tmp/20i1955**".

I write the python code through which, sends the ACK packet back to the x-terminal with the rsh data packet on behalf of the server from the attacker machine.

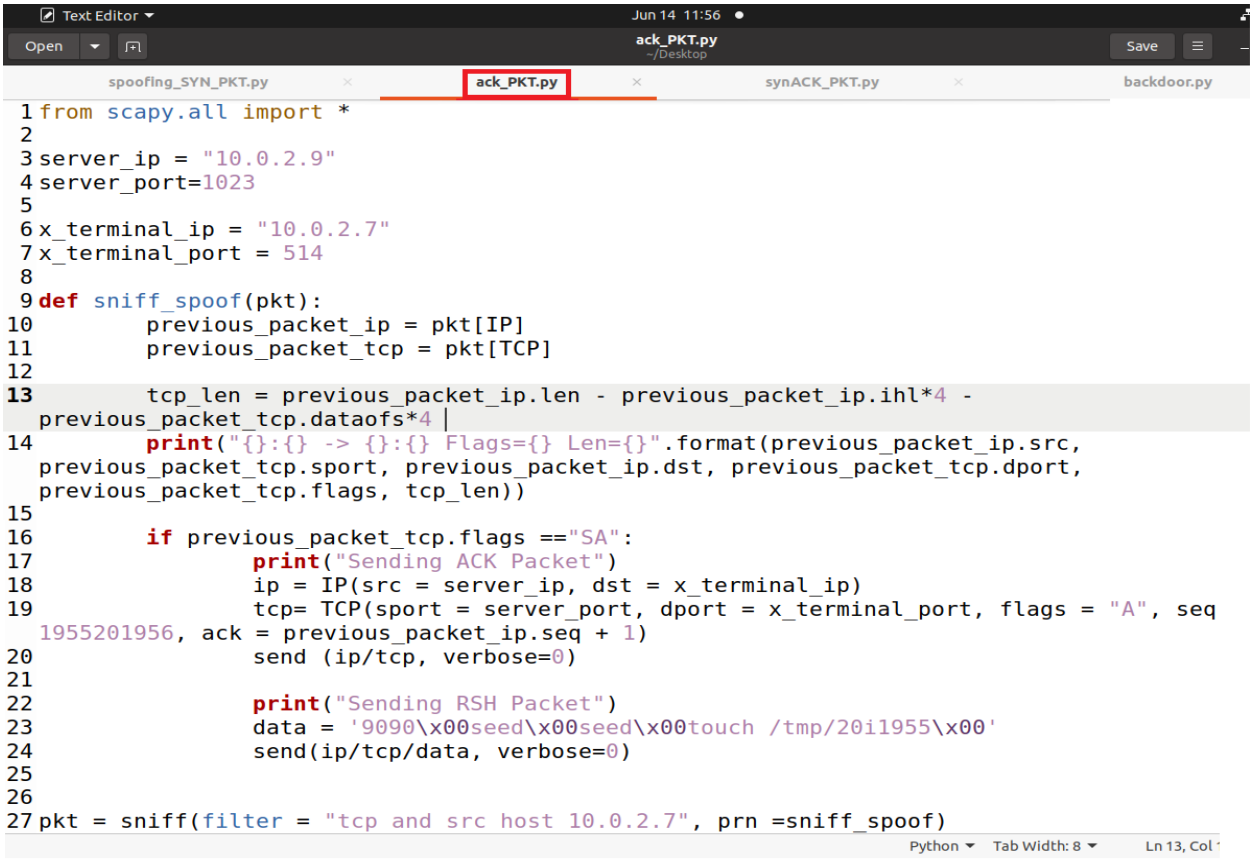

Figure 23: Attacker: Code used to send the ACK packet with the rsh data packet

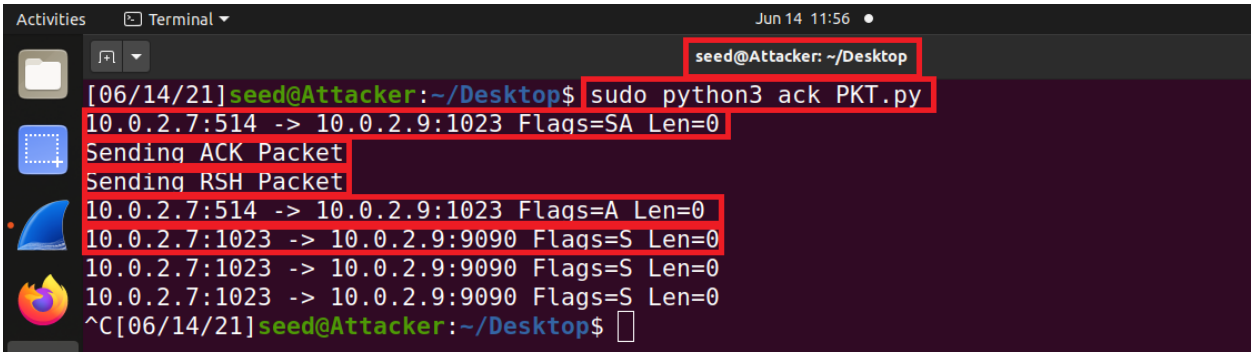

Figure 24: Attacker: Run the code

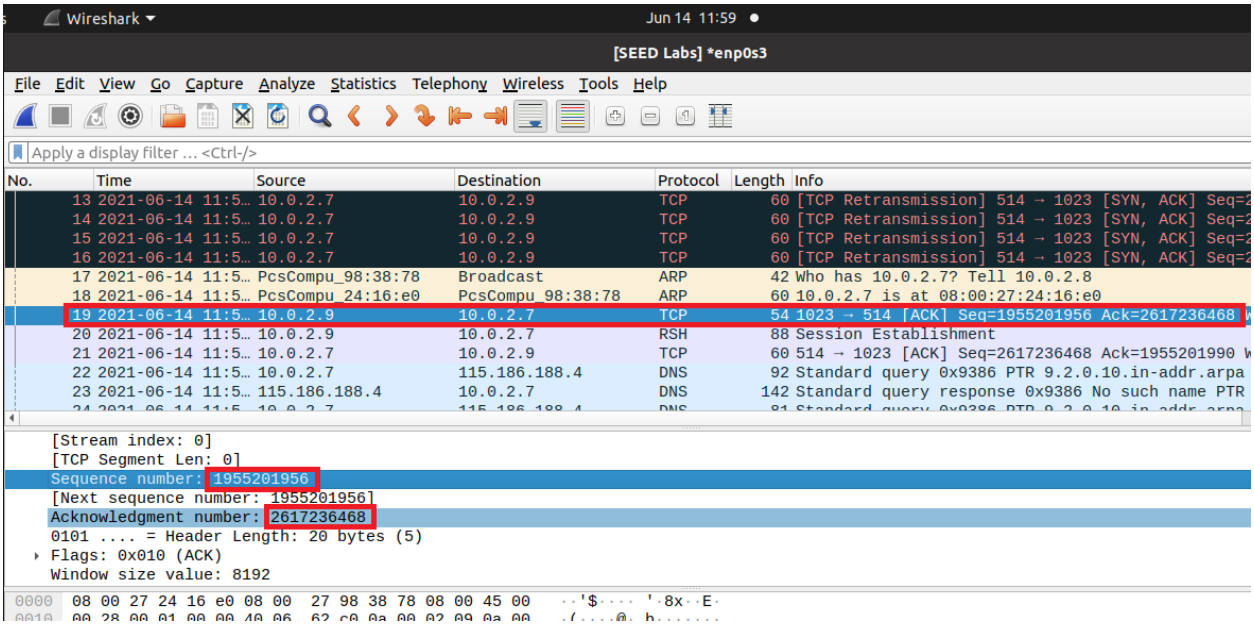

Figure 25: Attacker: ACK packet send to x-terminal successfully

As the x-terminal receives the ACK packet from the attacker the three-way handshake will complete and a TCP connection will be established. Now attacker sends the rsh data packet to the x-terminal containing the command that I need to run on the x-terminal. On receiving the rsh data packet x-terminal establish the rsh session between itself and the server.

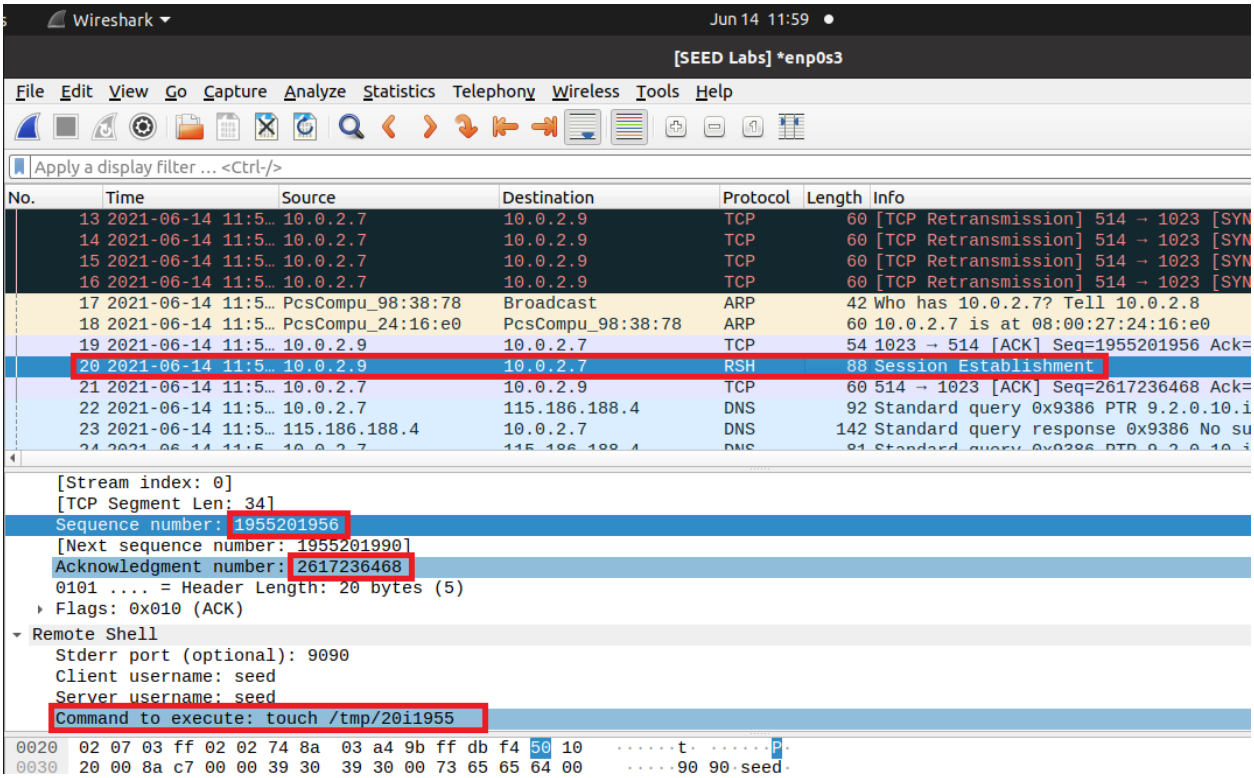

Figure 26: RSH session established

As the RSH session is established I go to the user machine to check whether my command runs on x-terminal or not and find that the command was not executed yet because here only the session is established and a three-way handshake for rsh was not completed.

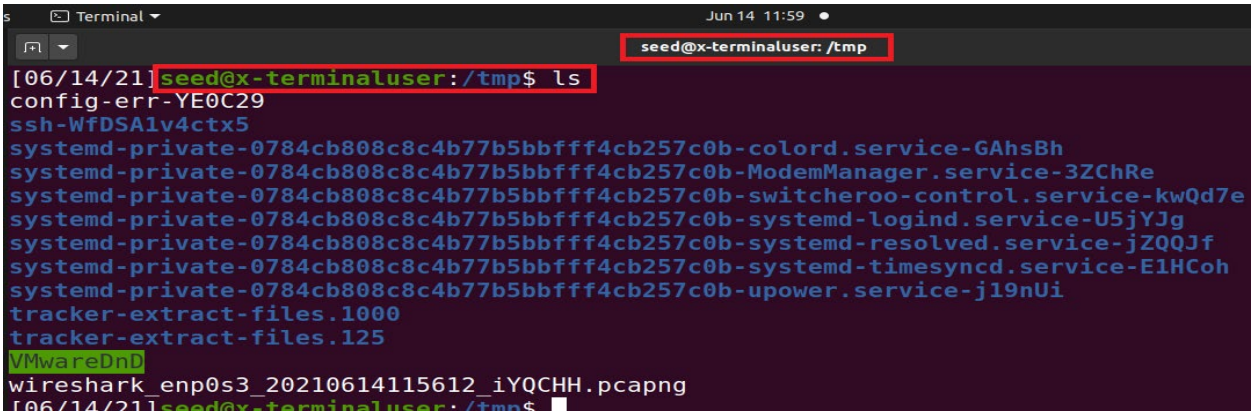

Figure 27: RSH Session Established but command not executed yet

Task-3 Spoof the Second TCP Connection

After the first connection was established, the x-terminal is going to initiate the second connection which is used by rshd to send out error messages. This connection will need to be established because if it was not established rshd will stop and our command will not execute, therefore we also need to spoof this connection so that x-terminal and server establish that connection successfully. To establish that connection attacker needs to sniff the SYN packet from the x-terminal to the server and after extracting the sequence number from the SYN packet, respond with the SYN-ACK packet to the x-terminal. When x-terminal receivers the SYN-ACK packet by the server it responds with the ACK packet, which in result completes the three-way handshake for the second connection. Now when both connections have established rshd will execute our command and our attack will be successful.

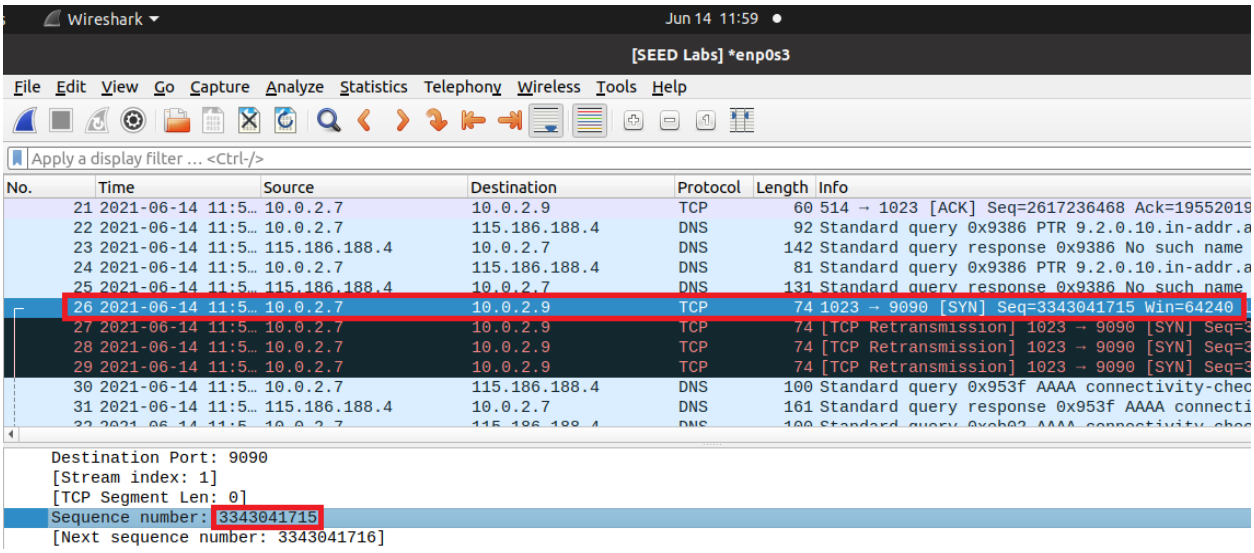

Figure 28: SYN packet from X-Terminal

For sniffing the SYN packet and respond with the SYN-ACK packet I write the python code through which, after sniffing the SYN packet responds with SYN-ACK packet to the x-terminal.

Figure 29: Attacker: Code used to sniff SYN packet and respond with SYN-ACK packet

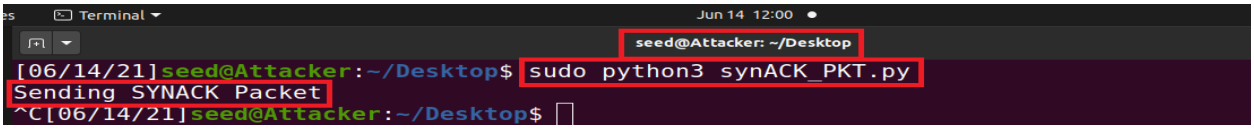

Figure 30: Attacker: Run the code

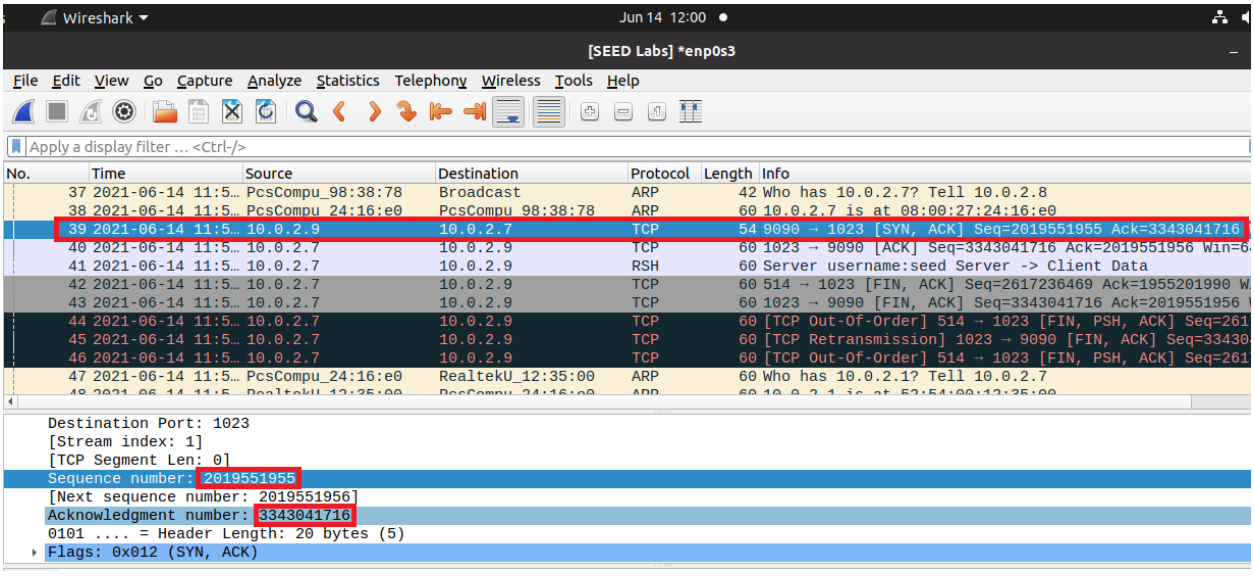

Figure 31: Attacker: SYN-ACK packet send to X-Terminal

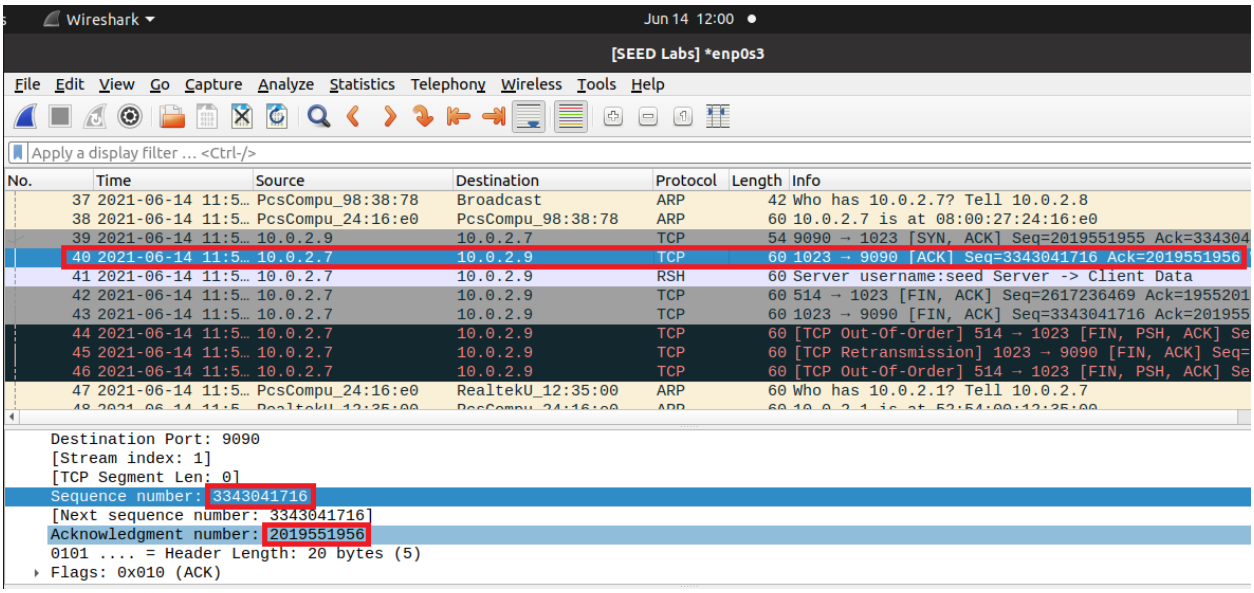

Figure 32: ACK packet from X-Terminal

Now here, after successful completion of a three-way handshake and connection establishment, my command "**touch /tmp/20i1955**" that I send in the rsh data packet executed successfully on the x-terminal machine.

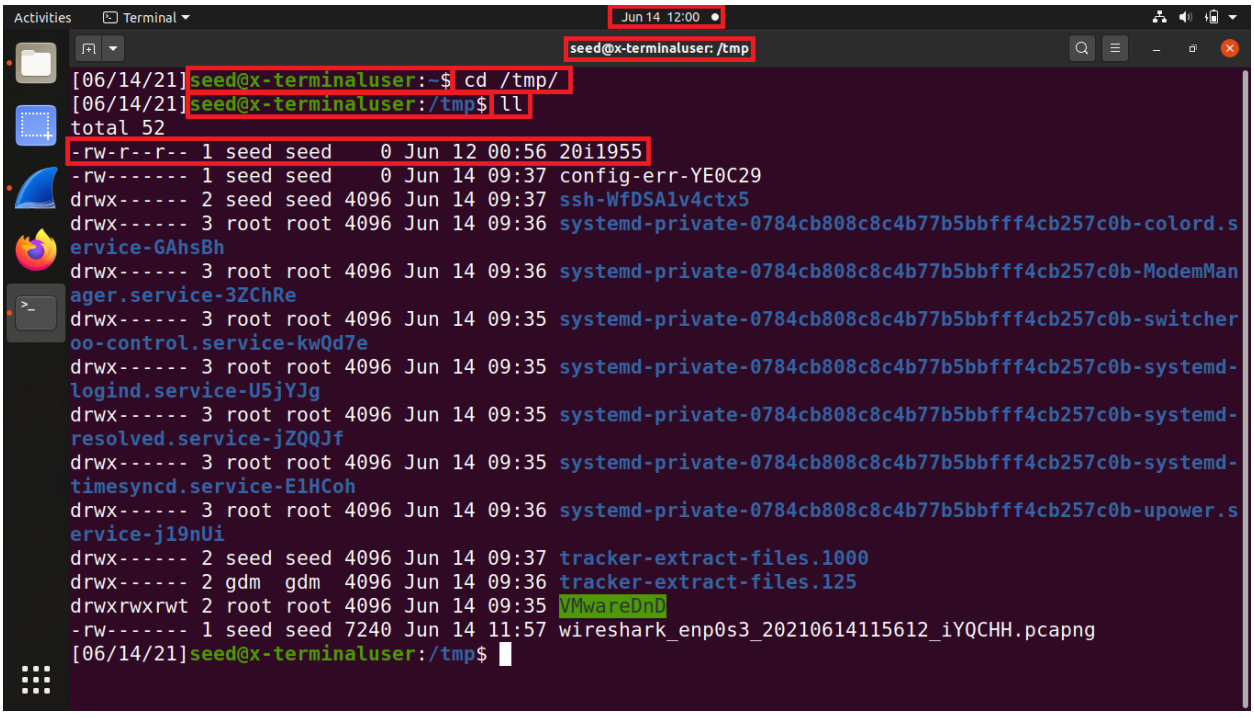

Figure 33: RSH command successfully executed on x-terminal

Part-4 Set up a Backdoor

Now in the previous part, I only run the touch command and see whether my command runs on the x-terminal after the successful connection establishment and find that my command successfully runs on the victim machine. Now to set up the backdoor on the x-terminal I am again going to follow the steps in the previous part but now instead of using the touch command to create the file on the x-terminal, I am going to enter the "+ +" string in the .rhosts file on the x-terminal using the echo command "**echo + + > .rhosts**" and again send this command with the rsh data packet. Now, this is the vulnerability in the rsh server that if it finds that there is a "+ +" string in the .rhosts file it will allow all devices to log in to the victim machine without entering the victim machine password. In an actual attack, Kevin Mitnick takes advantage of this vulnerability to perform the attack on Shimomura's computer. Now to set up the backdoor I combine all the codes using in the previous task in one code and run that code to perform the attack on the x-terminal.

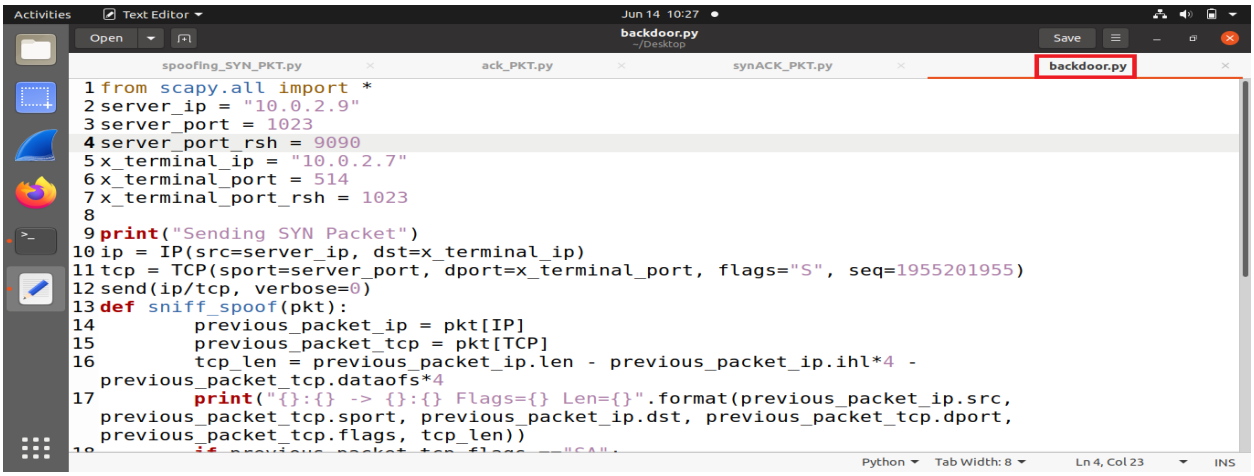

Figure 34: Attacker: Code to setup backdoor on x-terminal

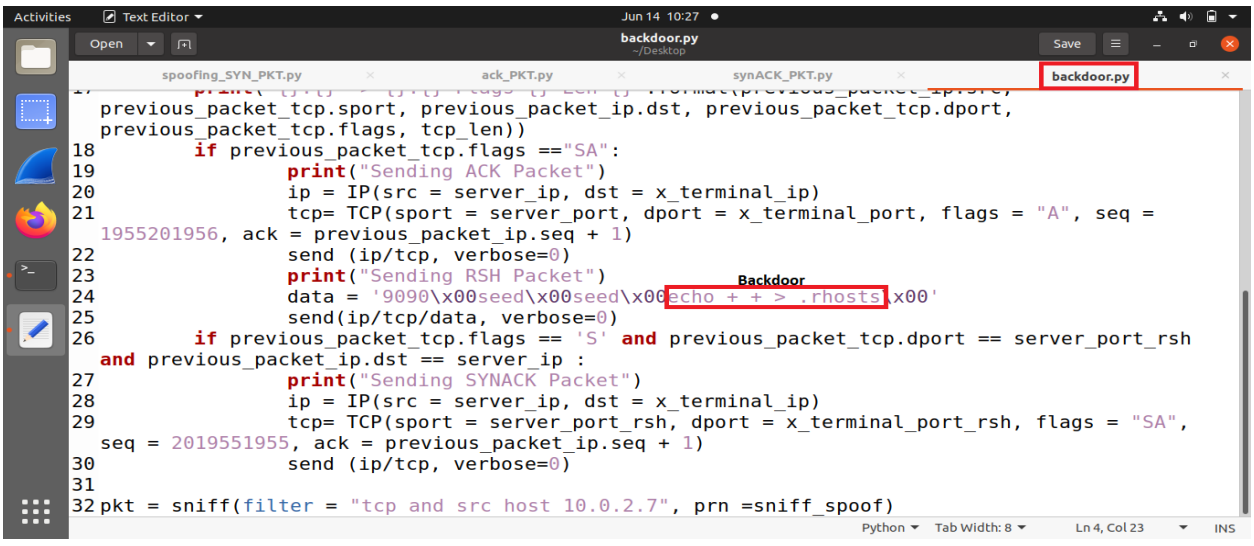

Figure 35: Attacker: Code to set up a backdoor on the victim machine

Now, this code first spoof's the SYN packet to the x-terminal, then x-terminal on receiving the SYN packet respond with SYN-ACK packet which it sniffs and extract the sequence number from that packet and then respond to x-terminal with ACK packet to complete the three-way handshake. Now after sending the ACK packet this code automatically sends another packet which is the rsh data packet that contains the backdoor command that I want to run on the x-terminal. On receiving the rsh data packet x-terminal initiate another connection which rshd uses to send out error messages and run our command on the x-terminal. To help the x-terminal successfully establish that connection this code sniff the SYN packet from the x-terminal and after extracting the sequence number responds with the SYN-ACK packet. When x-terminal receives that SYN-ACK packet it responds with the ACK packet which again completes a three-way handshake for the second connection. On the successful establishment of our second connection rshd runs the backdoor code and our attack will become successful.

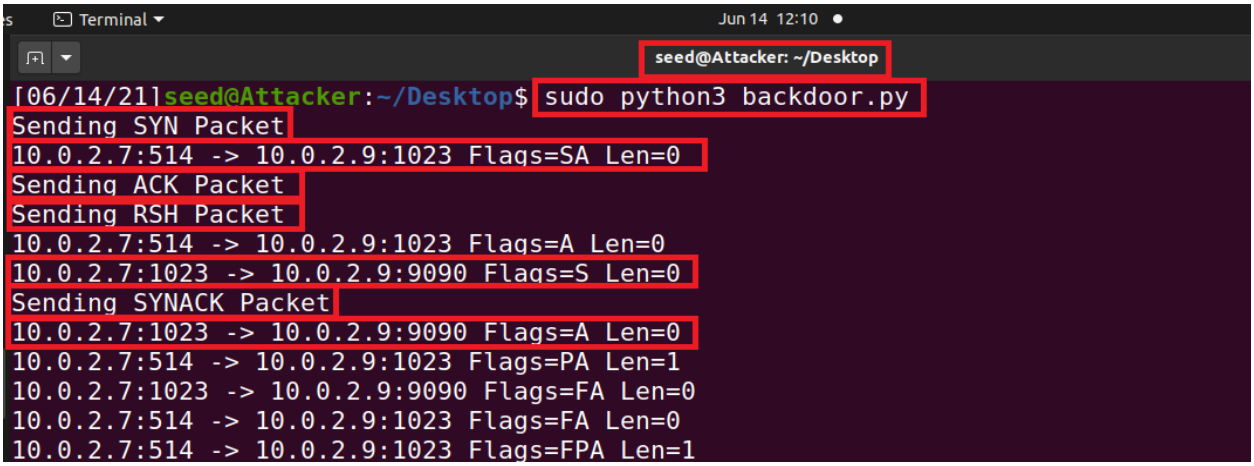

Figure 36: Attacker: Run the code

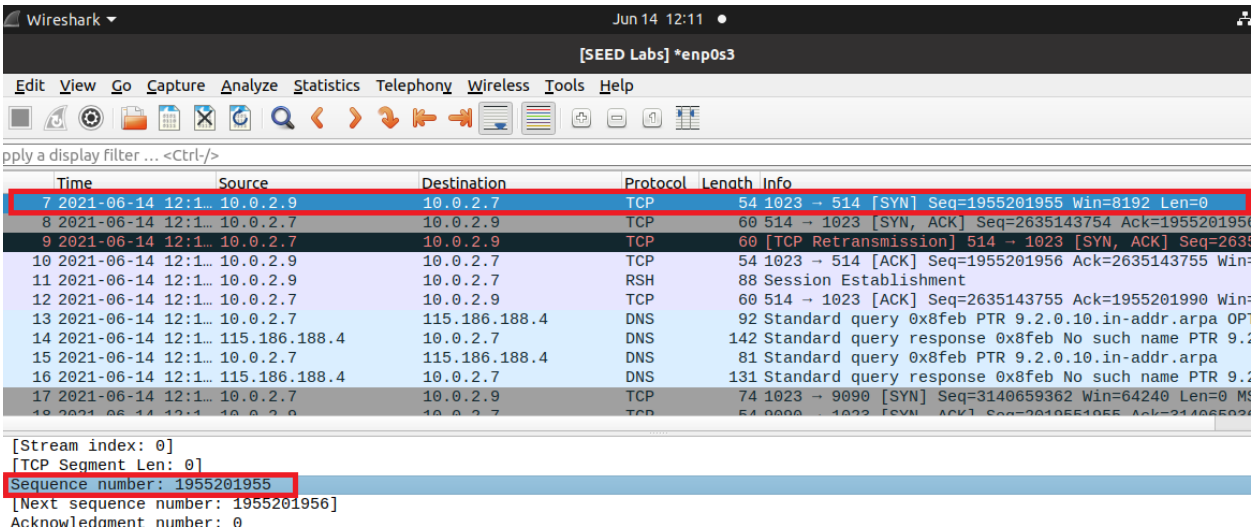

Figure 37: Attacker: SYN packet send to x-terminal successfully

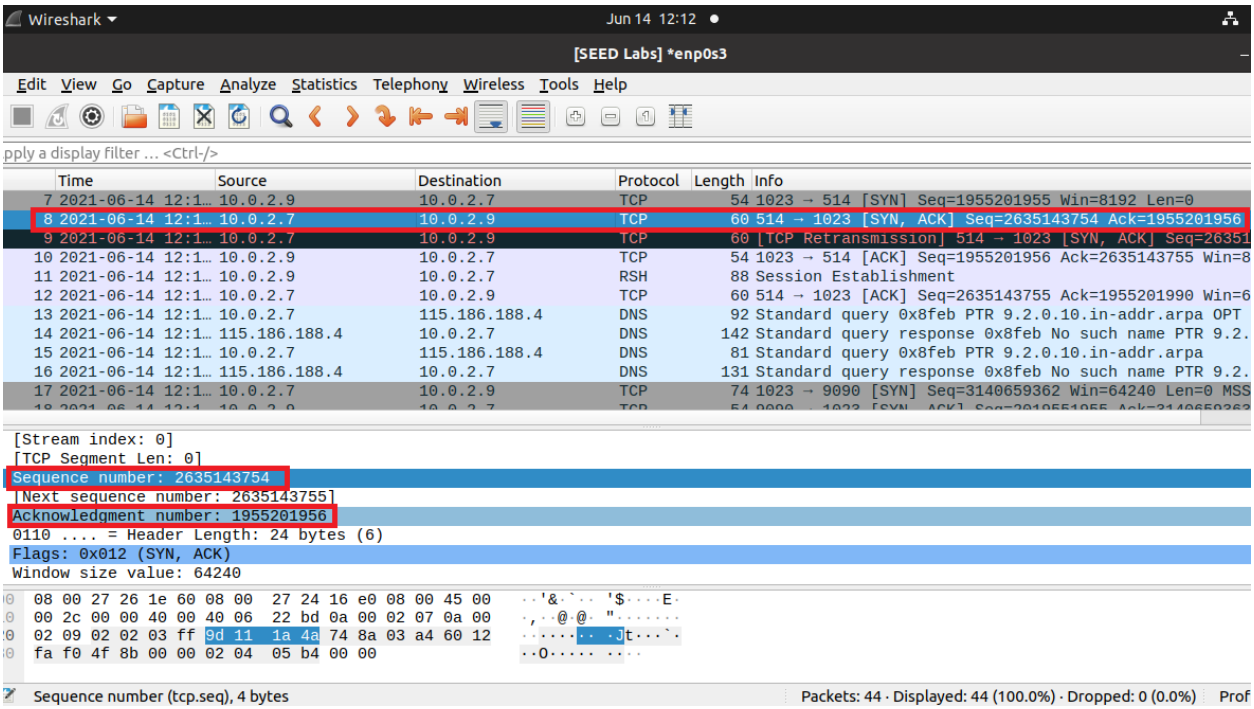

Figure 38: X-terminal respond with SYN-ACK packet

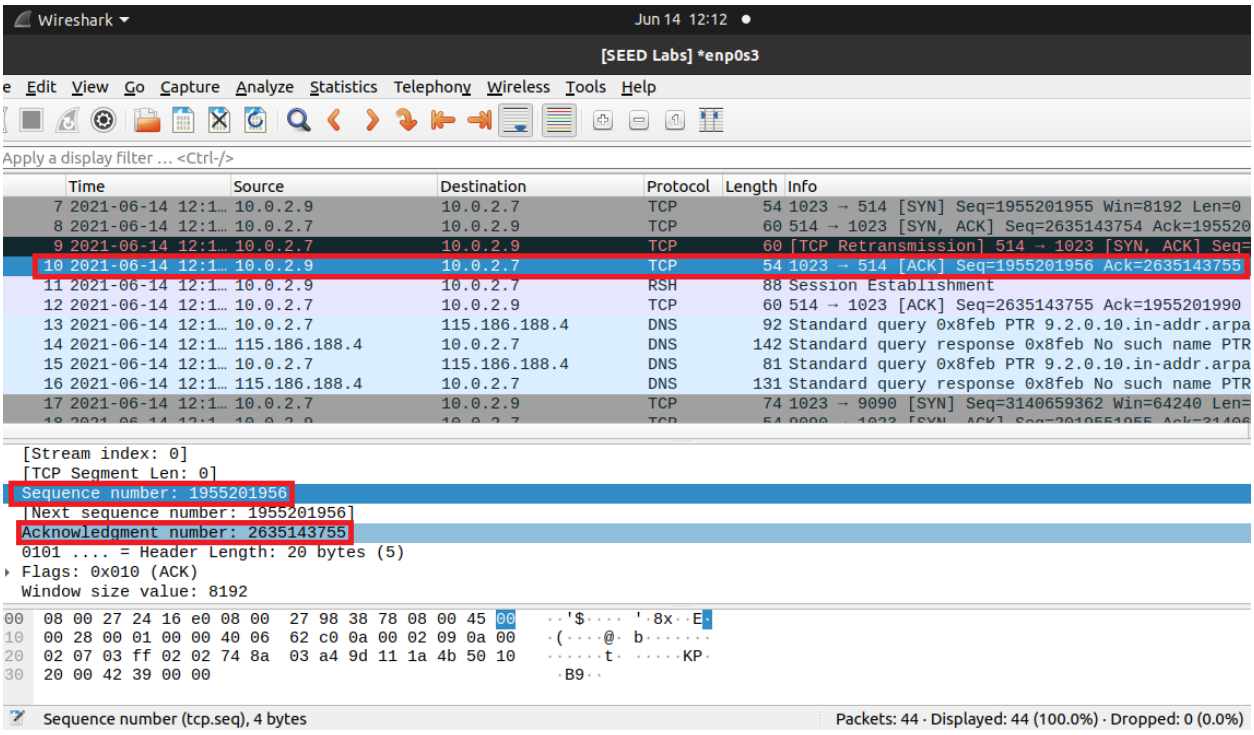

Figure 39: Attacker: ACK packet send to x-terminal successfully

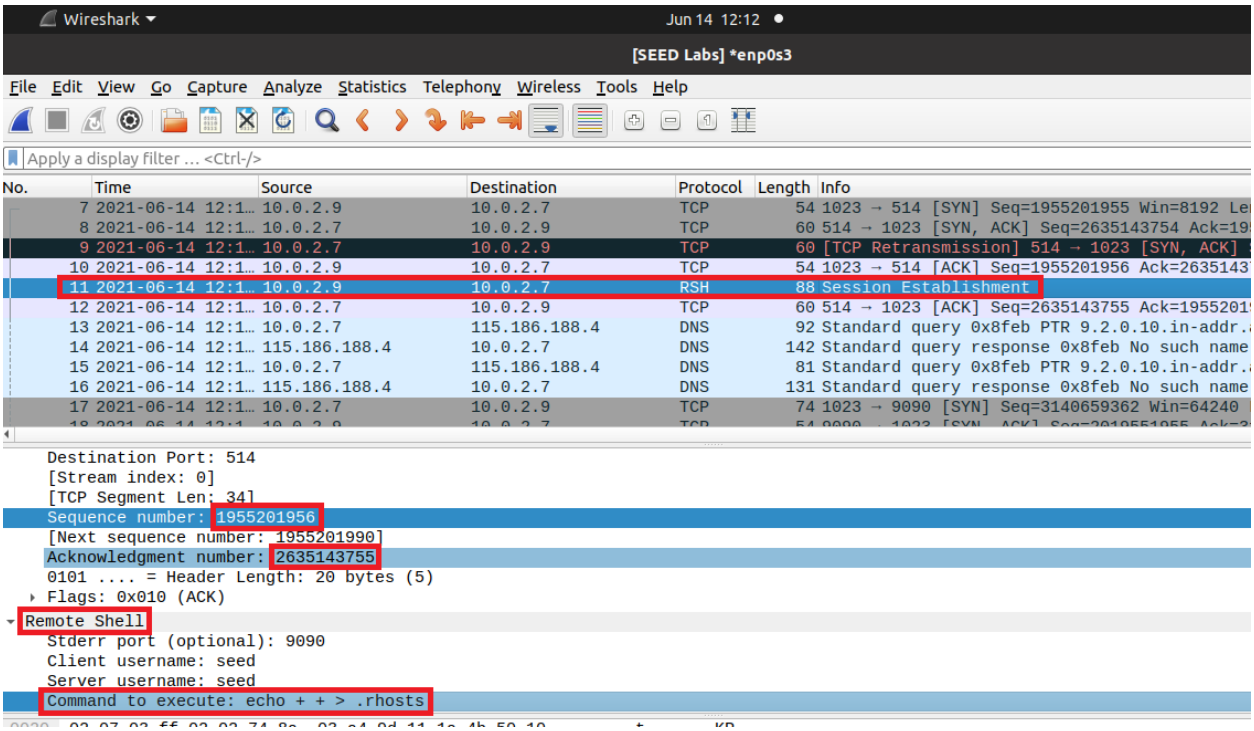

Figure 40: RSH session established

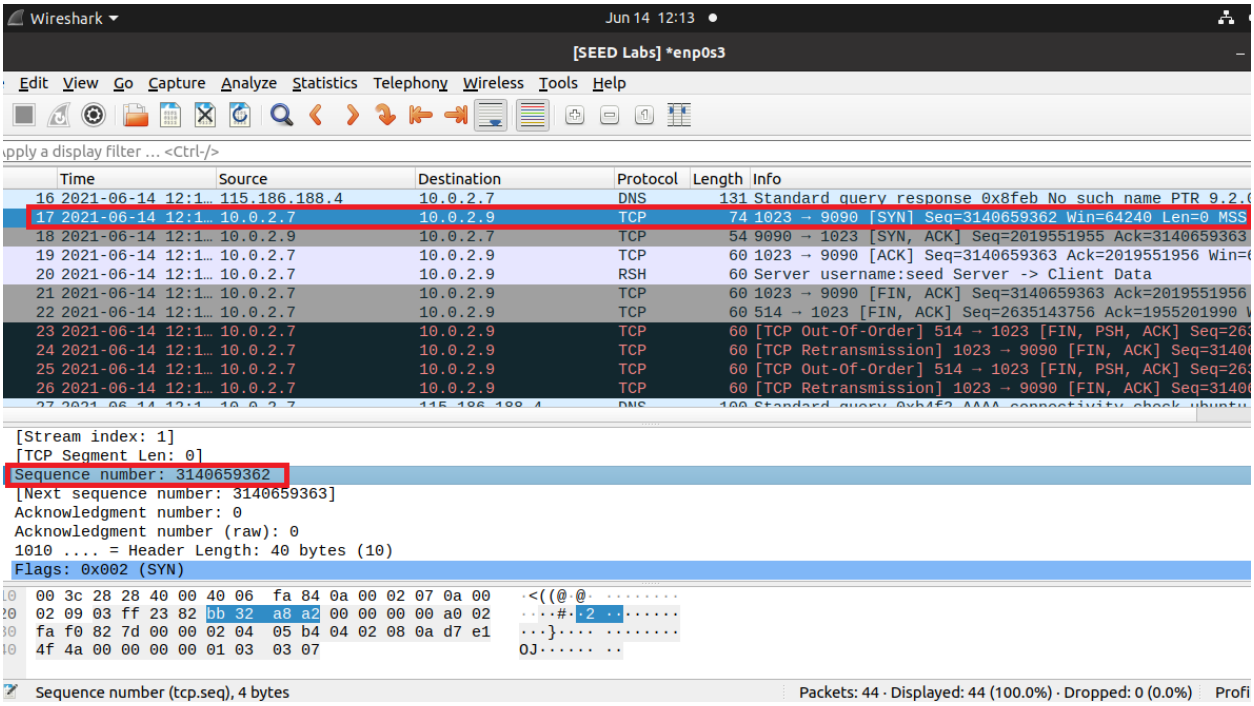

Figure 41: SYN packet from X-Terminal for second TCP connection for rshd

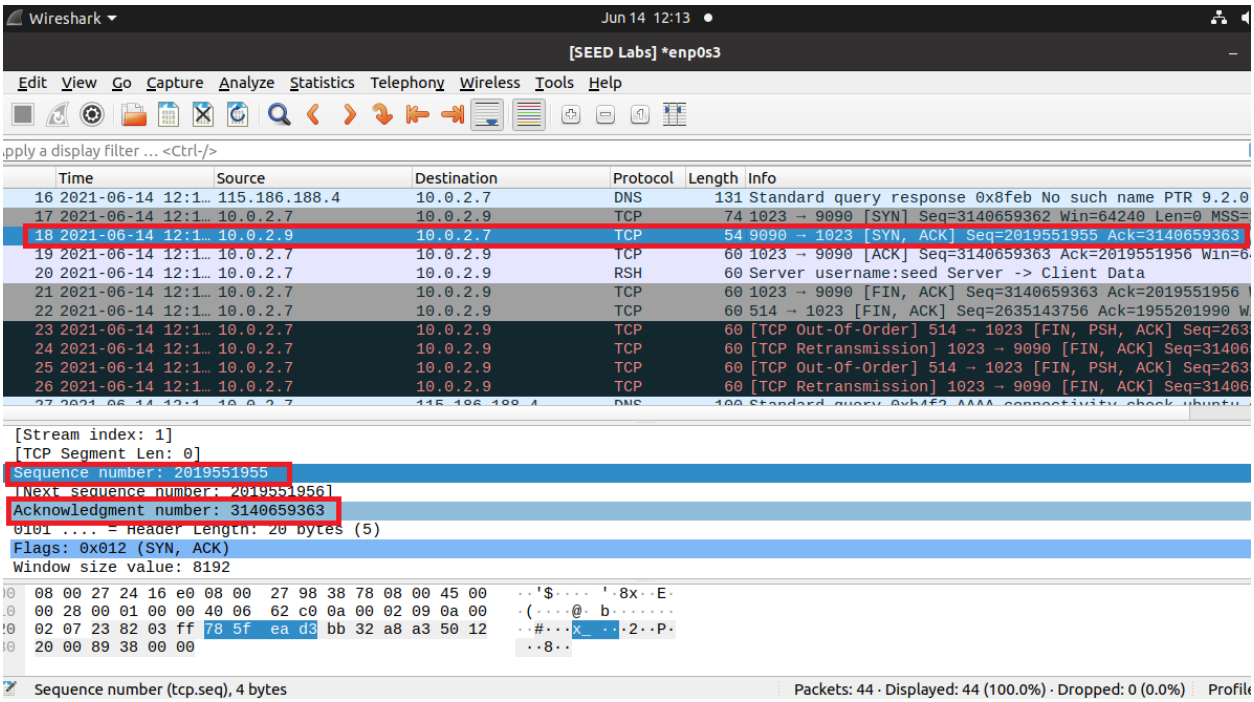

Figure 42: Attacker: SYN-ACK packet send to X-Terminal for second TCP connection for rshd

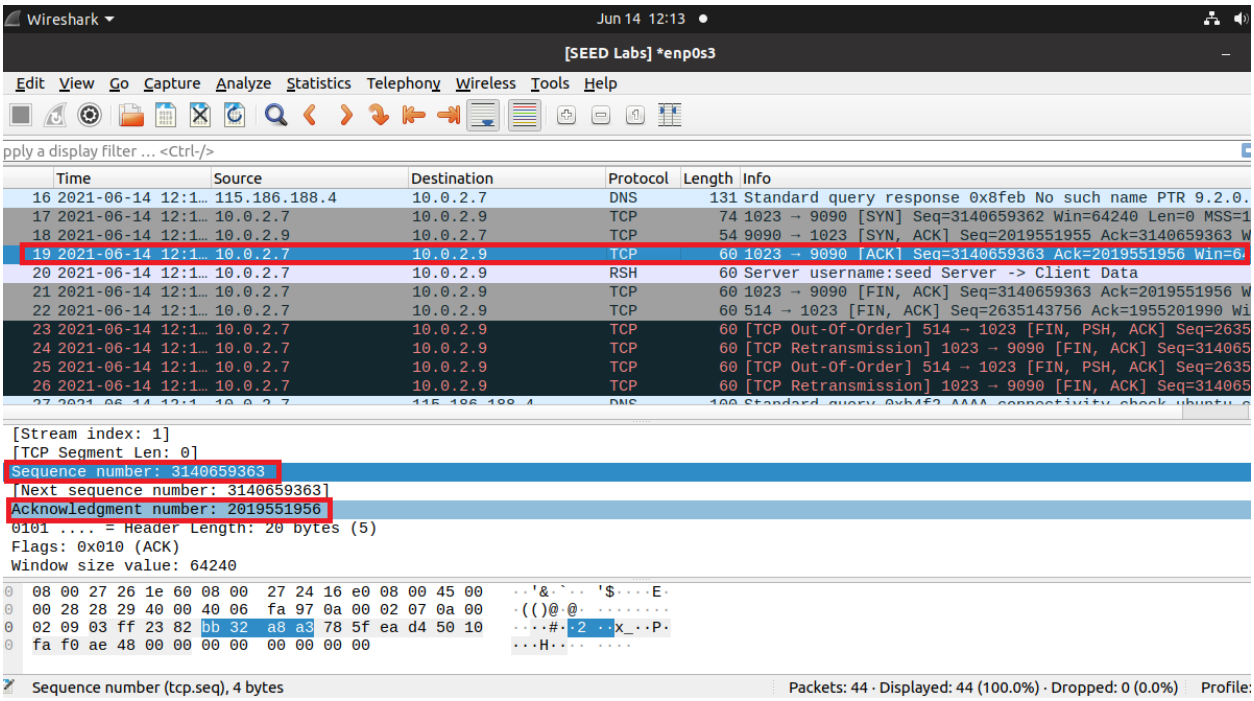

Figure 43: ACK packet from X-Terminal for second TCP connection for rshd

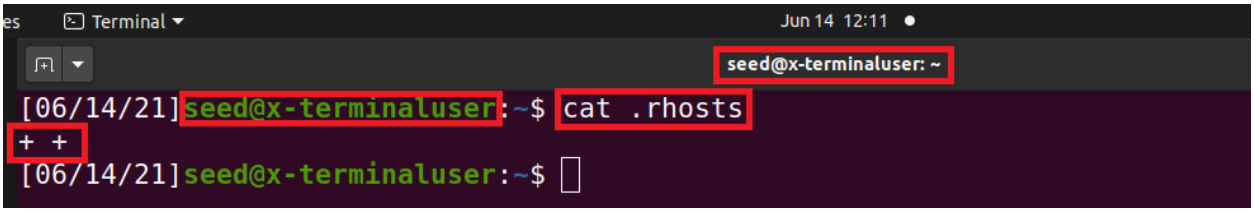

Figure 44: RSH command runs successfully and backdoor inserted in .rhosts successfully

As the backdoor command runs successfully, now I try to login to the x-terminal machine by using the rsh command "**rsh 'x-terminal IP**" and logged in to the x-terminal successfully without needing the x-terminal machine password.

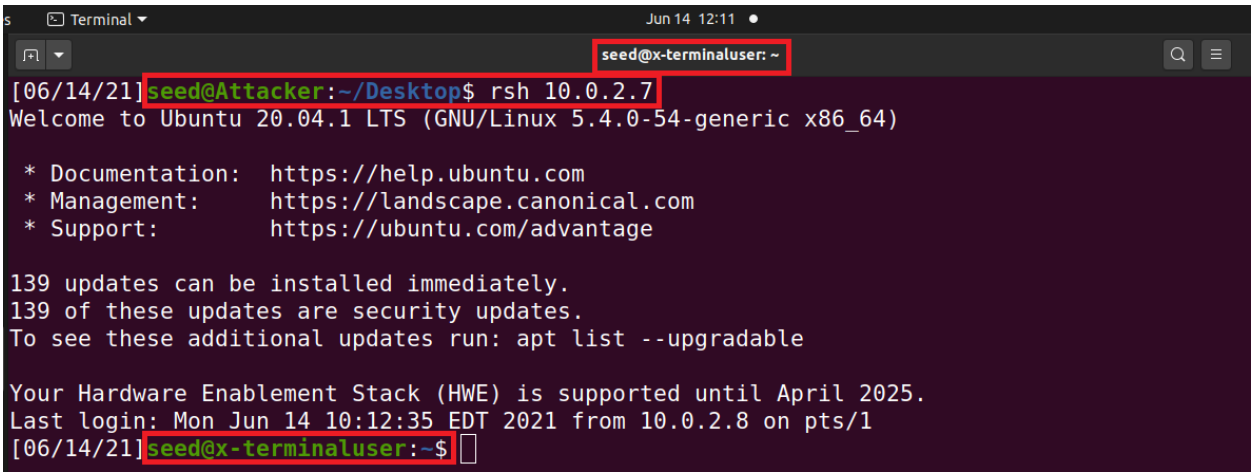

Figure 45: Attacker: Logged in to the x-terminal machine successfully without needing the password