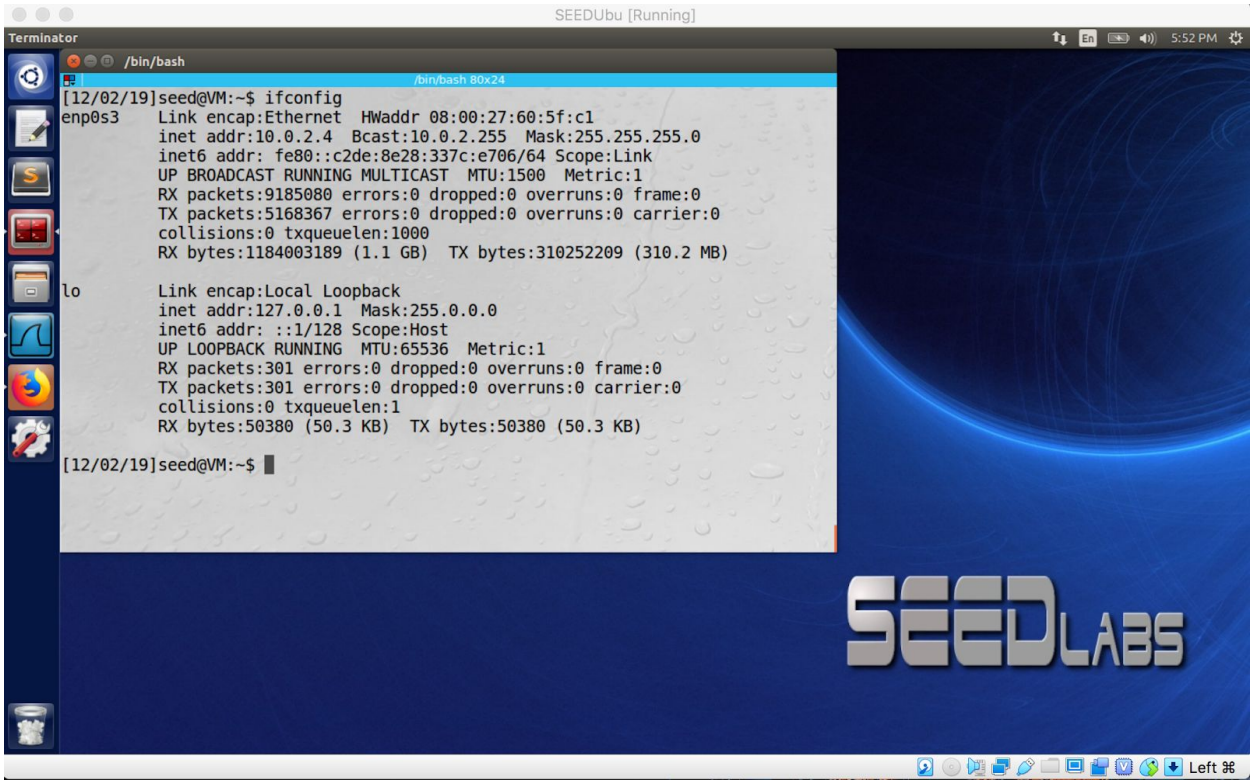
Greg Gertsen
Nazim Zerrouki
TCSS 481
TCP/IP Lab

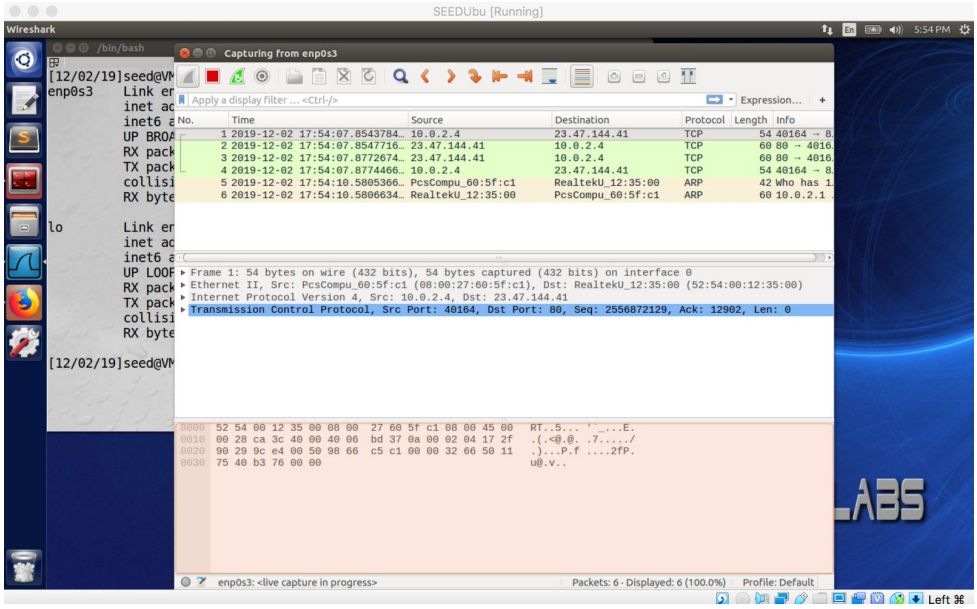
3.1 Task 1: SYN Flooding Attack

For this task we used 3 machines - The victim, attacker, and observer. The attack was launched against the victim machine's default Apache web server from the attacker's machine(Kali) using HPING. We found that the attack was initially unsuccessful against the victim due to the fact that the SYN cookies were enabled. Once the SYN cookies were disabled and the attack was run again, it was successful. The reason that this was the case was because with SYN cookies enabled, the TCP queue on the victim's machine gets cleared when the queue is nearly full. When the cookies are disabled, the queue fills up and then the server has to drop new incoming packets and therefore the attack becomes successful. Attack successful because victim web server was made unavailable.

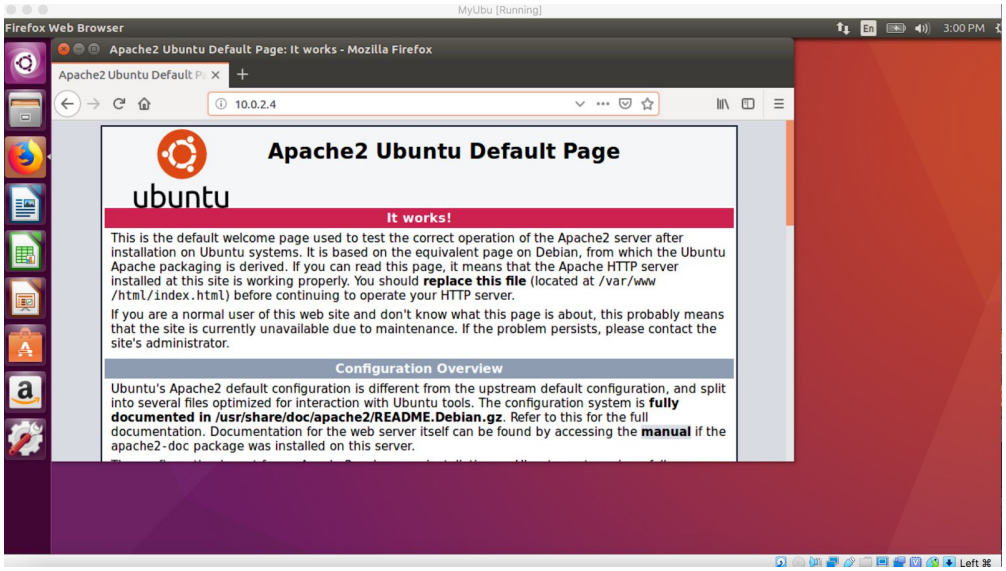
The victim machine



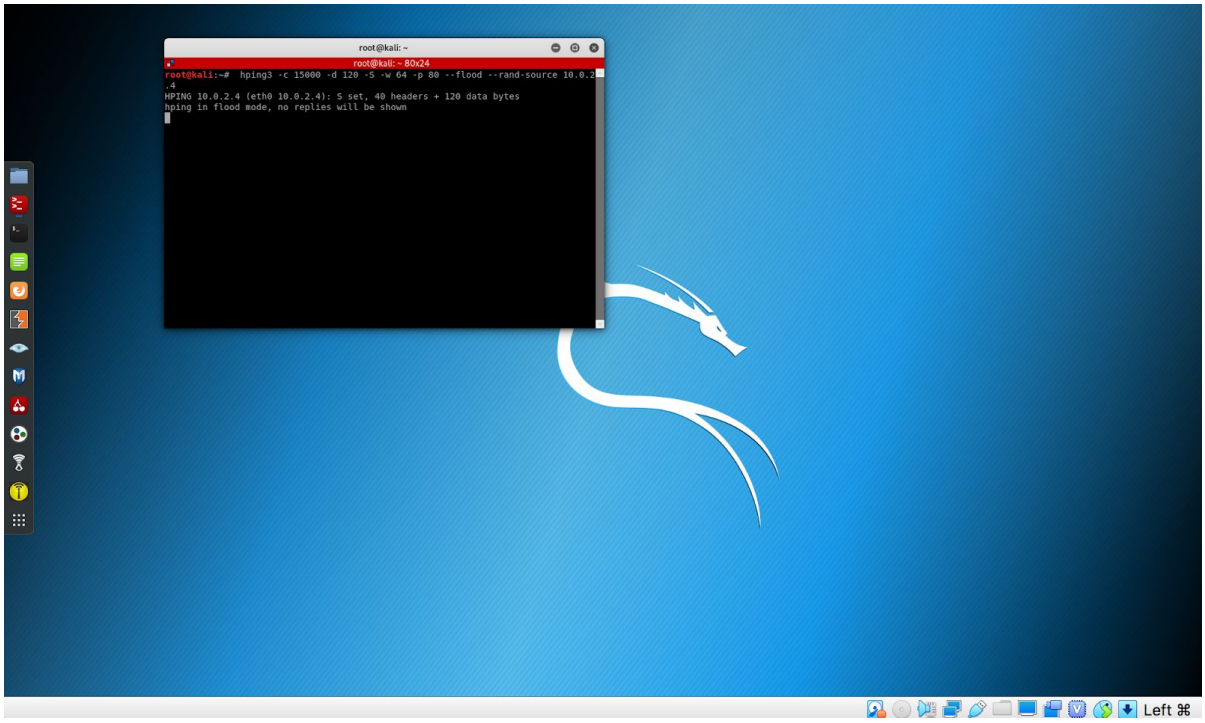
Wireshark on victim machine prior to attack



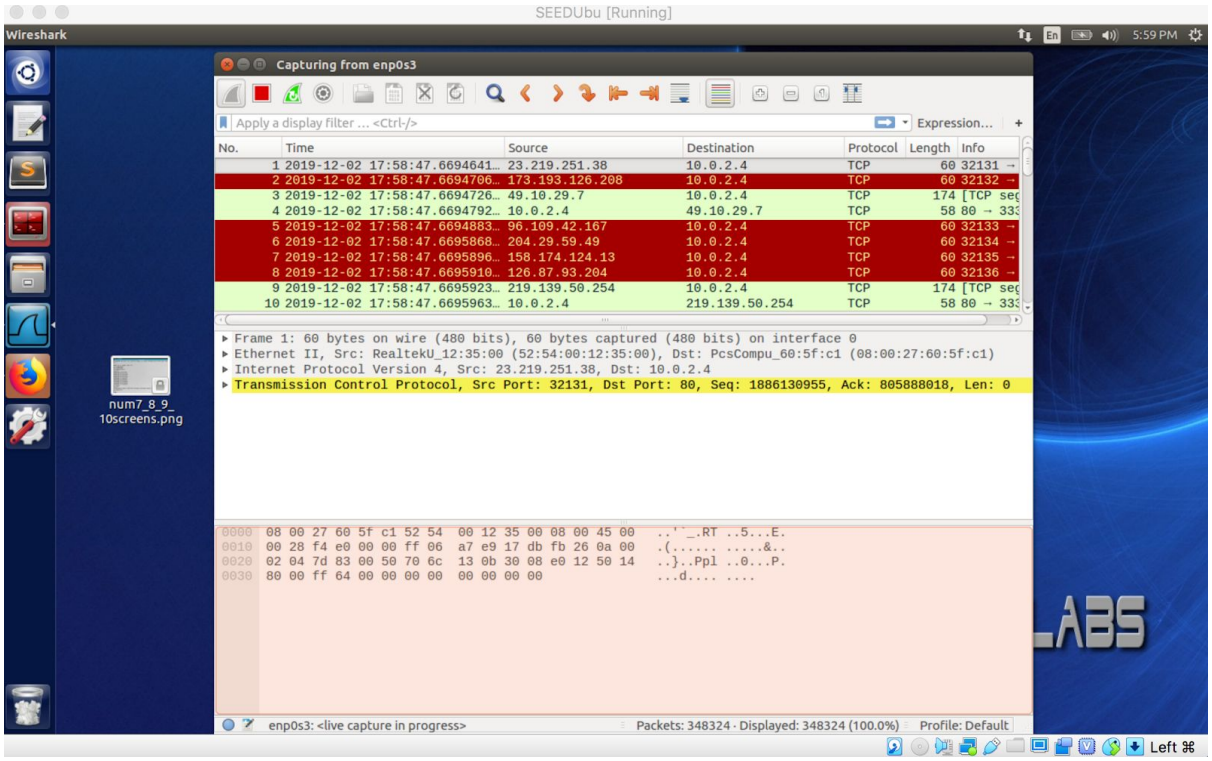
Observer machine accessing victim's server



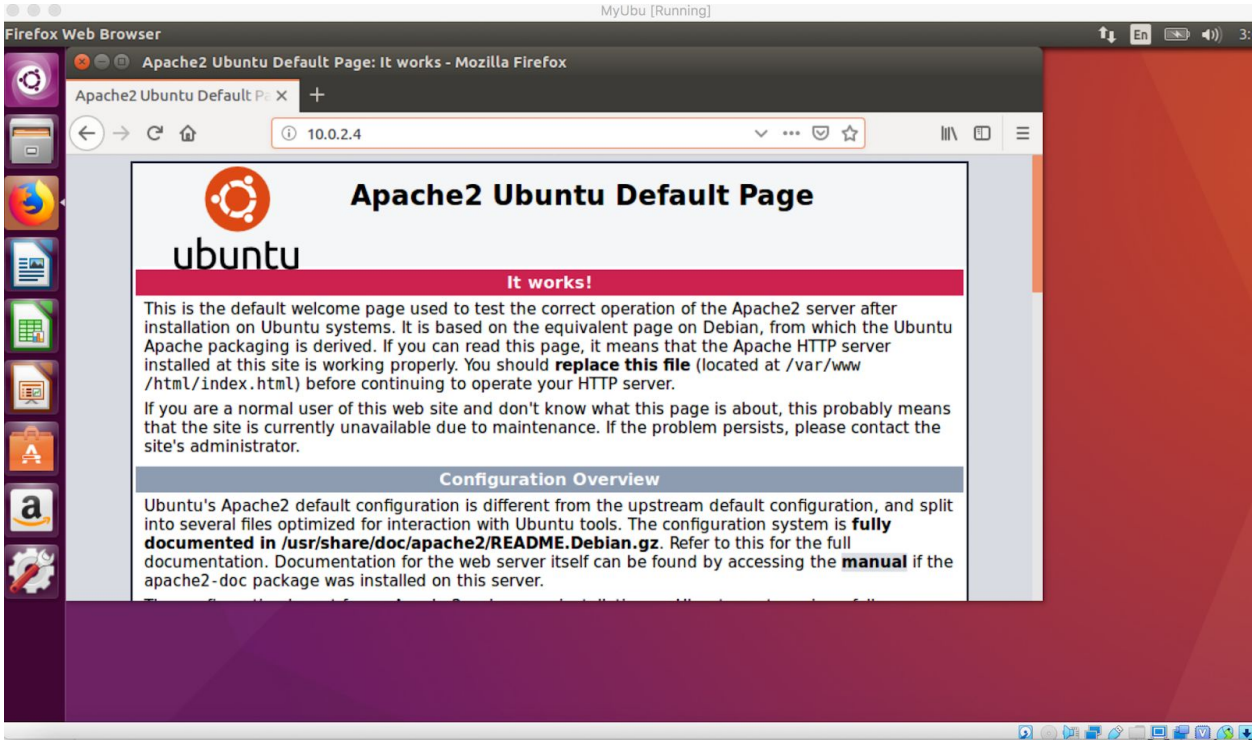
Attacker machine running HPING against victim machine.



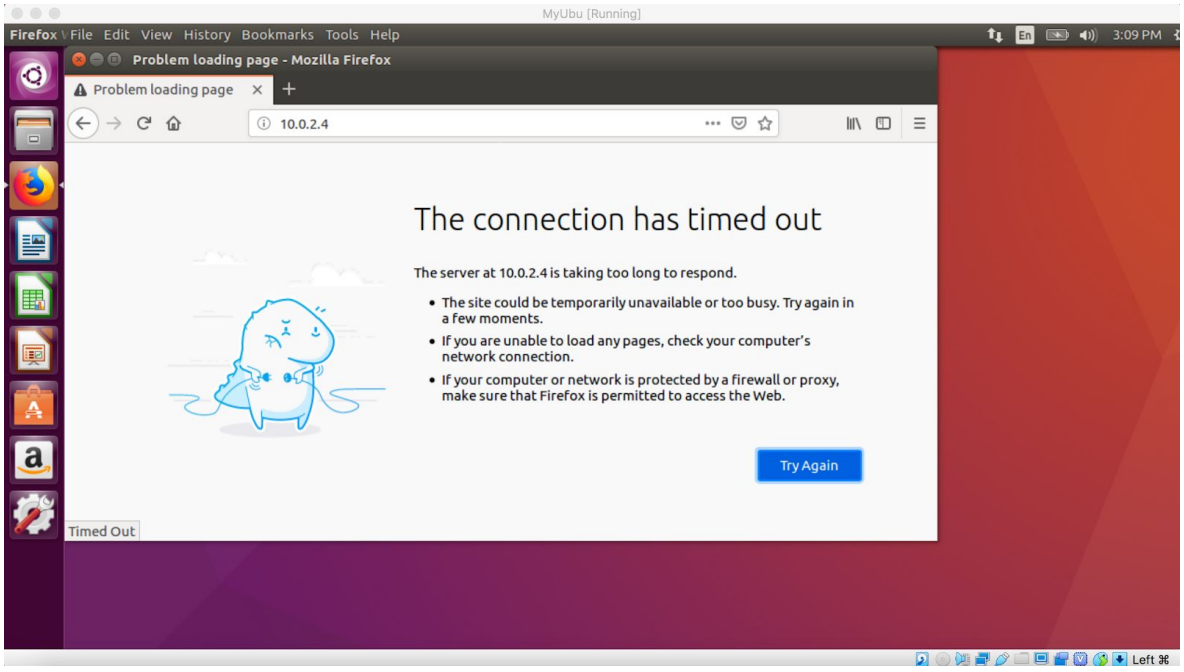
Wireshark on victim machine during attack



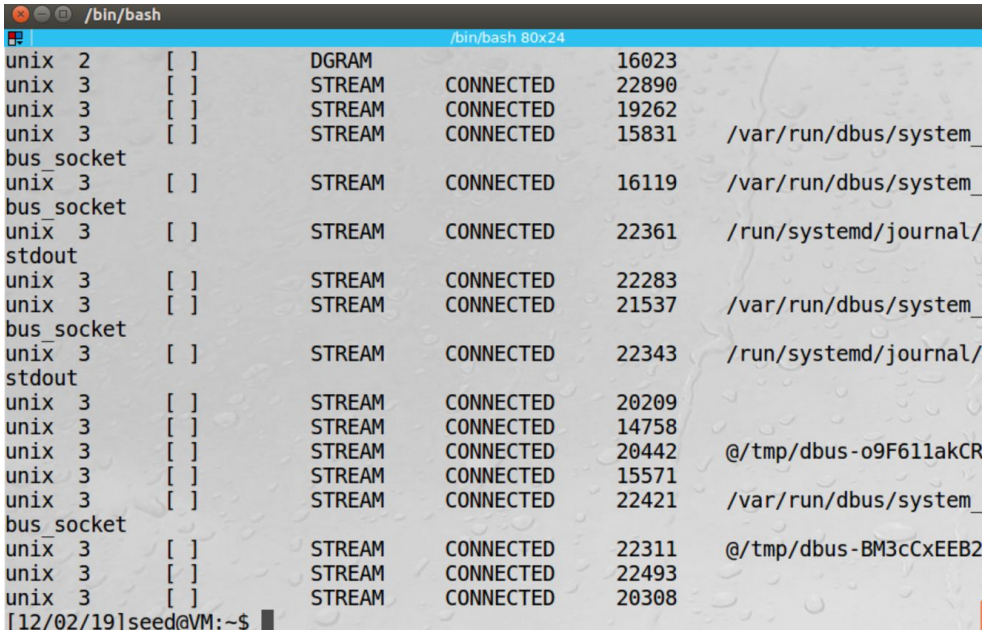
Observer during attack with cookies switched on, was still able to access server. Attack unsuccessful.



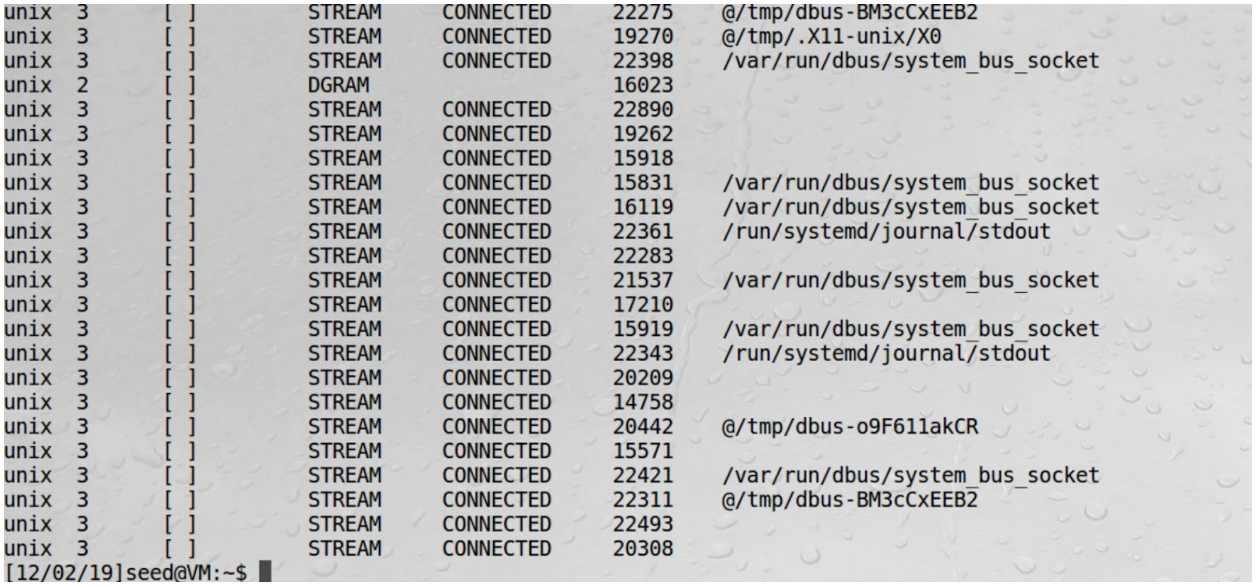
Observer during attack with cookies switched off, attack worked successfully. Was unable to access server.



Netstat -na command run on victim machine before attack, there appeared to be less connections, it was sparser looking.



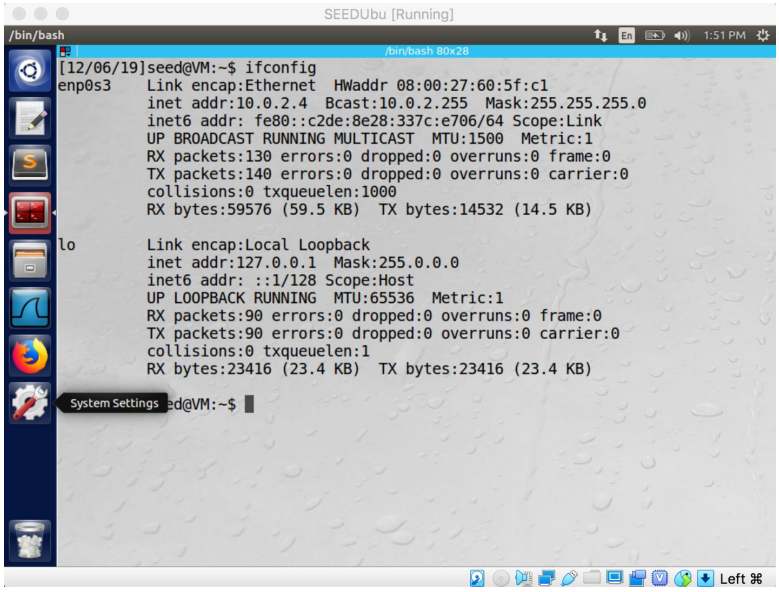
Netstat -na command run on victim machine during the attack. Observed that all the connections were being used.



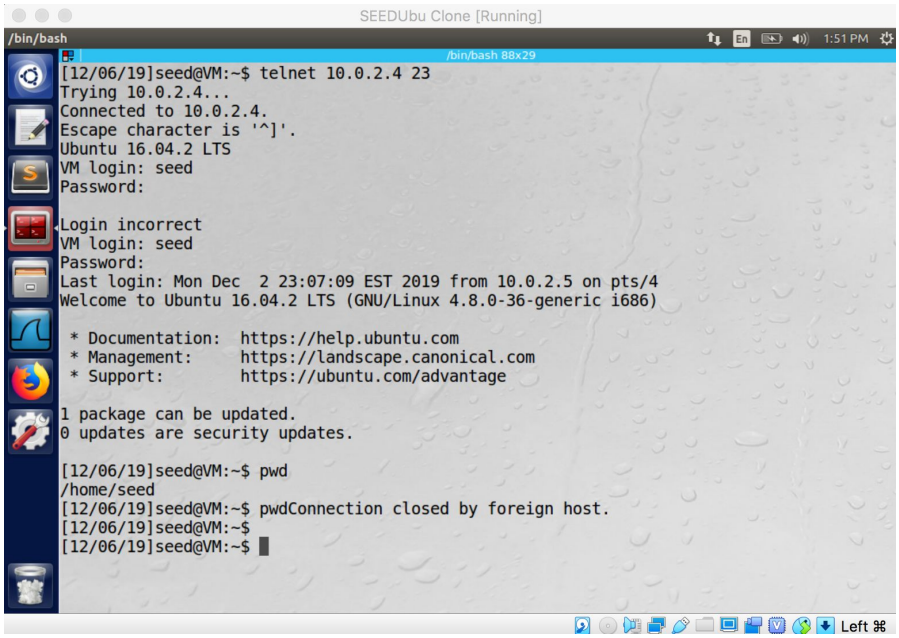
3.2 Task 2: TCP RST Attacks on telnet and ssh Connections

For this attack, a telnet connection is established between A and B using standard telnet protocol. The attacker computers goal is to terminate the TCP connection between A and B, so it sends out a TCP packet with an RST request to the other computer. The TCP packet is a reset packet and is made by the attacker to appear to be sent between parties A and B. The attack works exactly the same for SSH and is demonstrated below. Attack was successful. And this was evident because message appeared on victim machine “Connection closed by foreign host”.

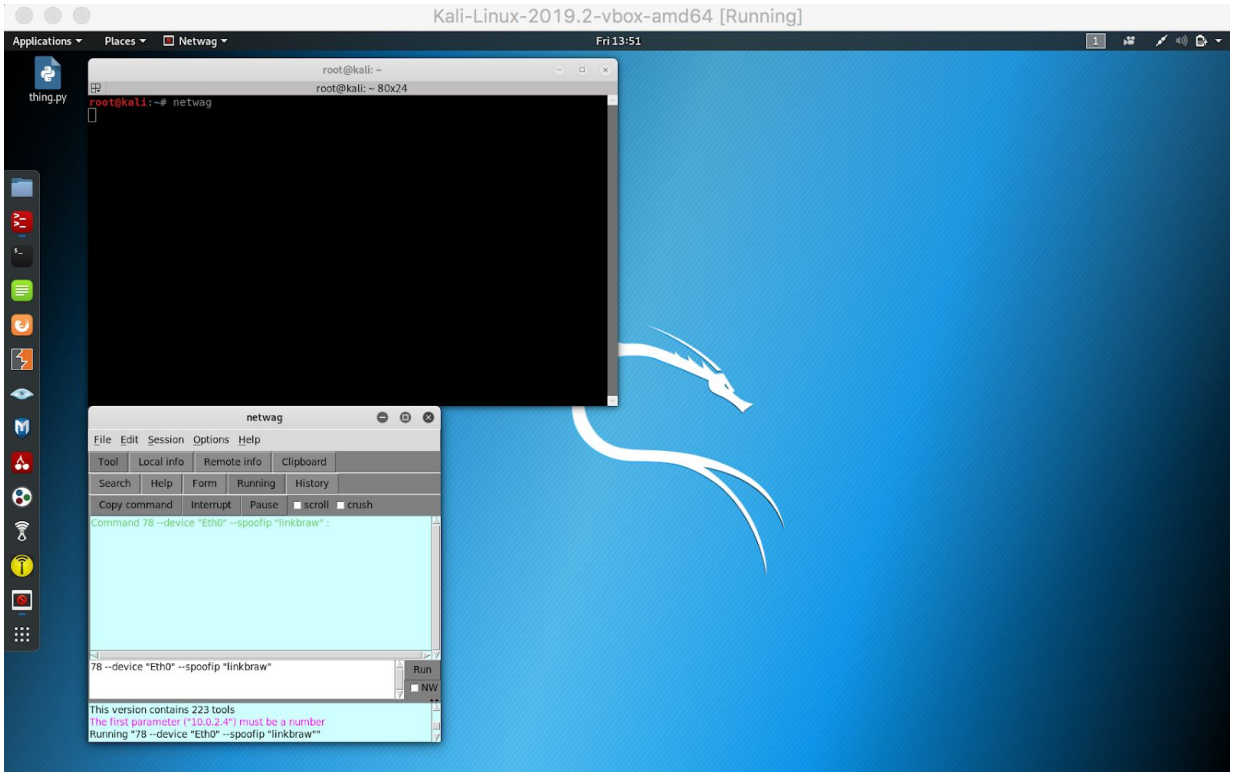
Computer A host for telnet



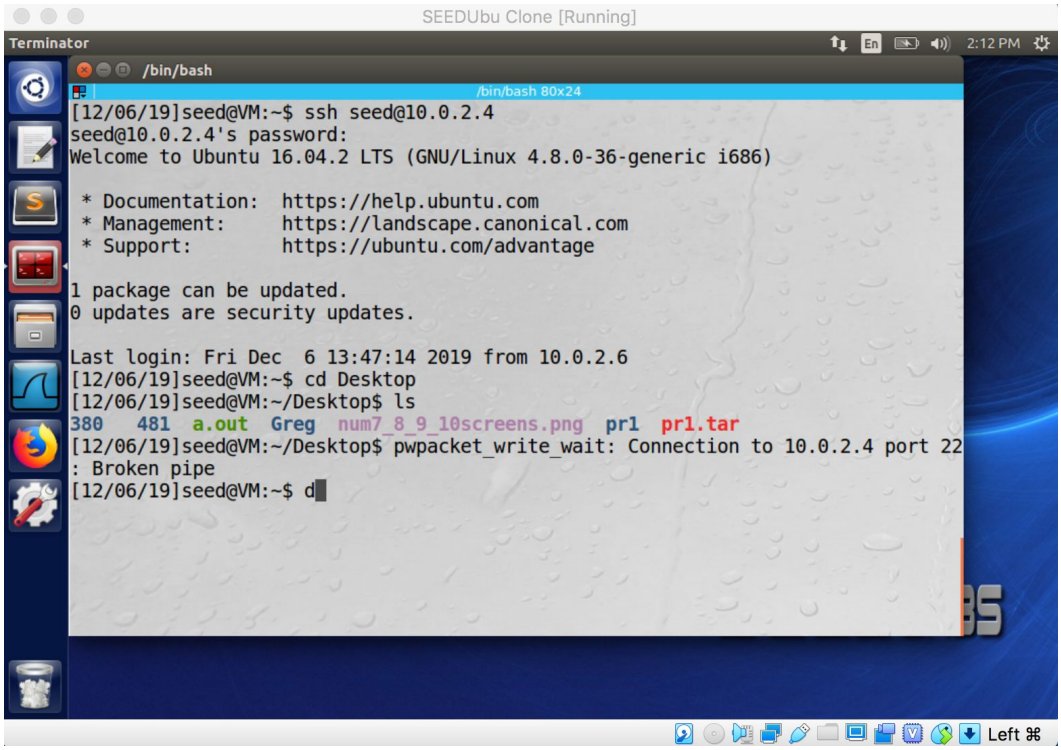
Computer B user, telnet established and broken by attacker



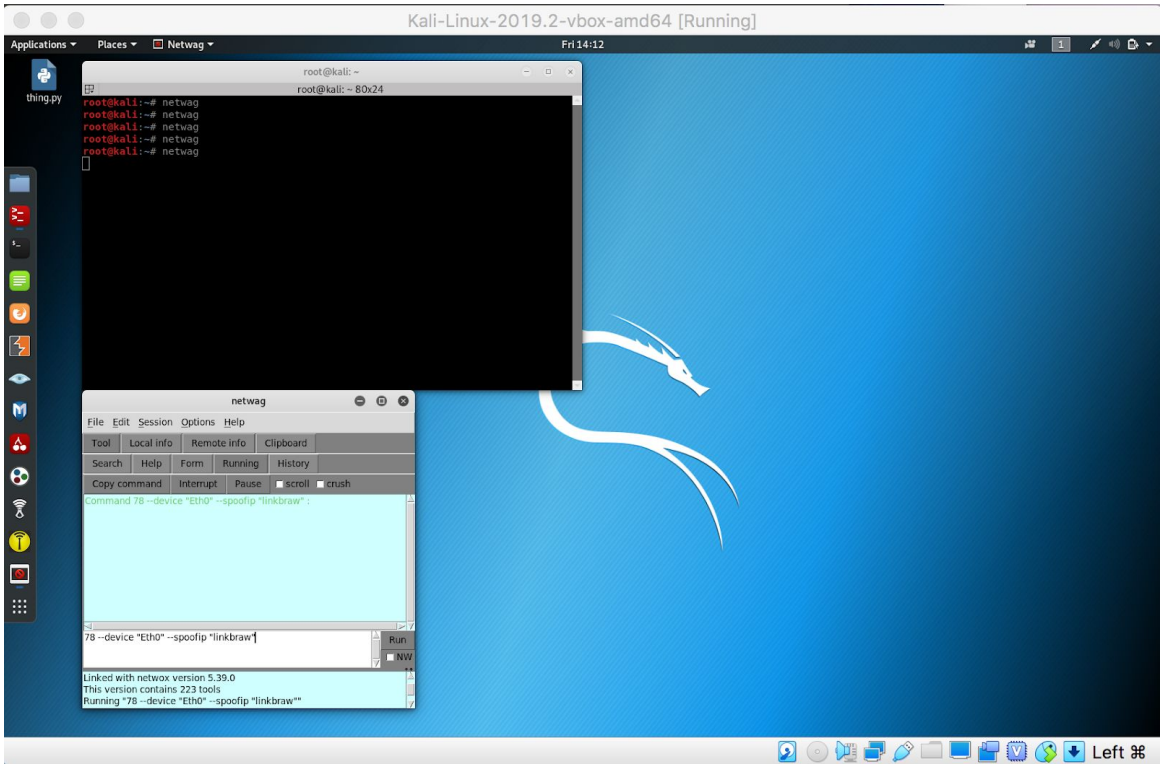
Attacker computer running netwag GUI and command 78



SSH connection established between A and B and broken by attacker.

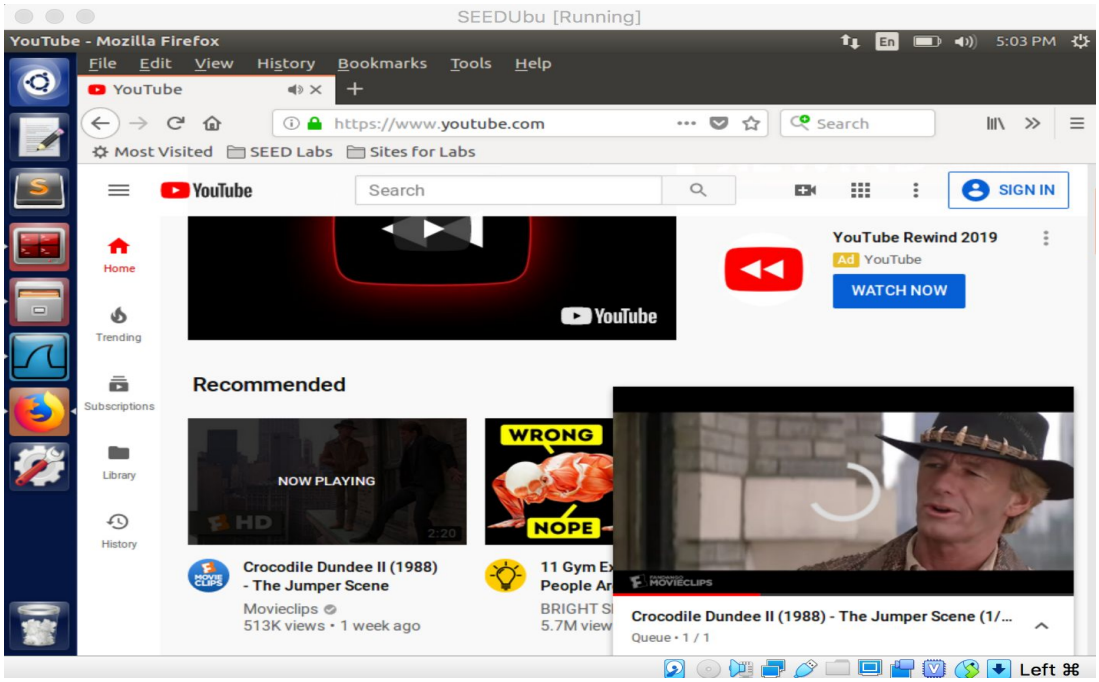


SSH connection between A and B terminated by attacker using netwag GUI command 78

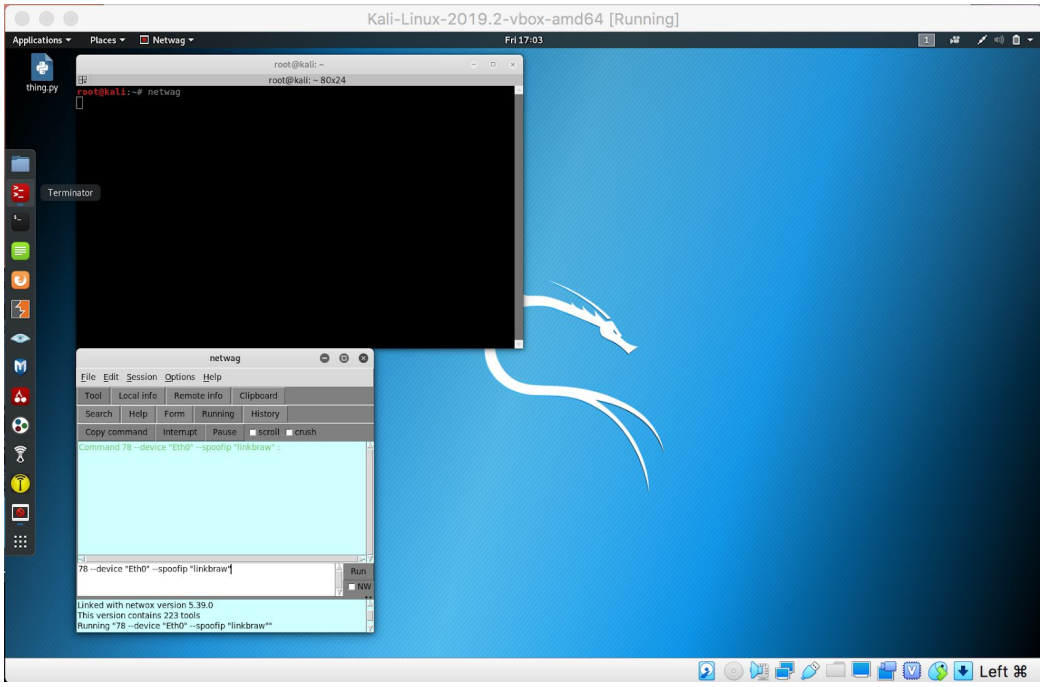


3.3 Task 3: TCP RST Attacks on Video Streaming Applications

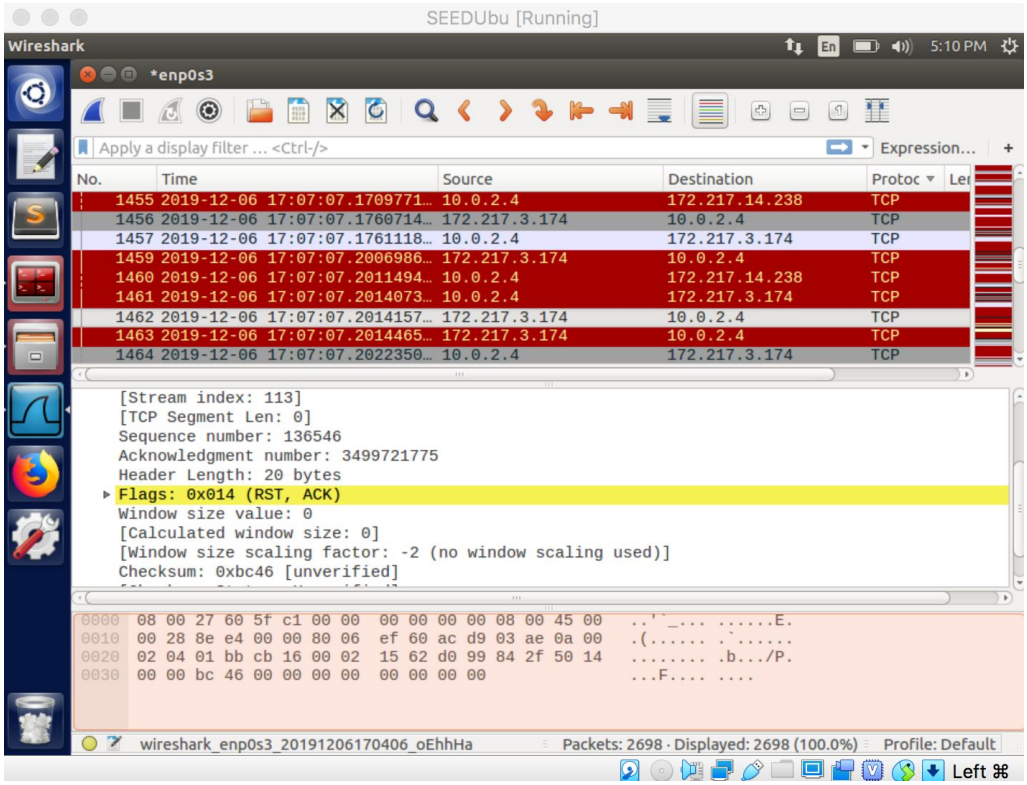
The principal behind this attack is the same for the telnet and SSH TCP attack. Basically, the video streaming TCP link is broken because the attacker is sending RST packets. This prevents the video from buffering any further then from the initial loaded state.Attack successful- Victim computer opens up a video stream, and can see stream is stopped, trying to buffer.



Attacker computer uses netwag GUI using command 78 to disrupt stream



Wireshark on victim's computer showing the RST packets coming from attacker

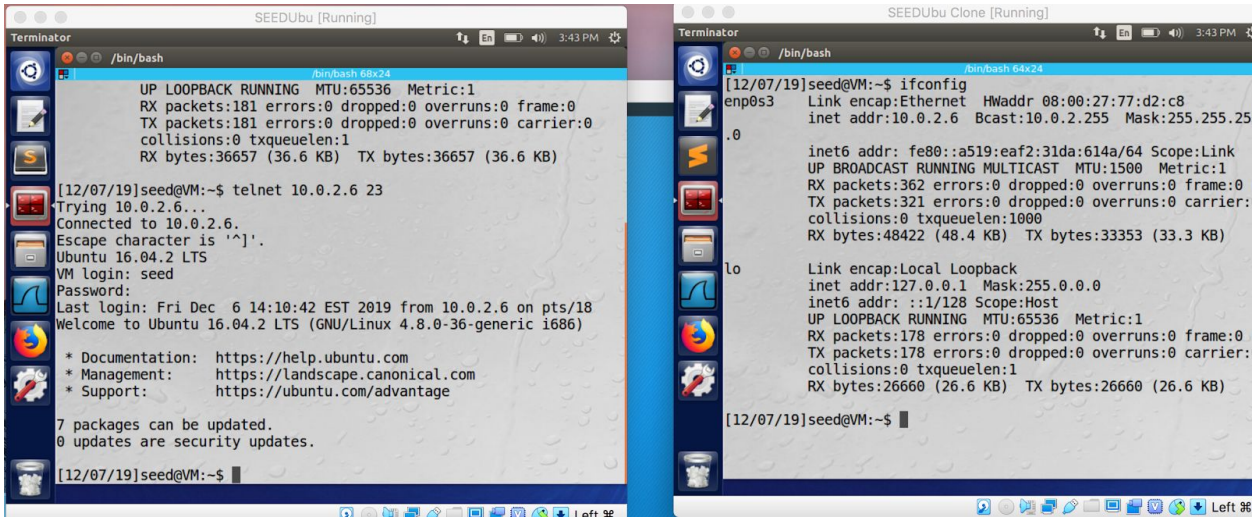


3.4 Task 4: TCP Session Hijacking

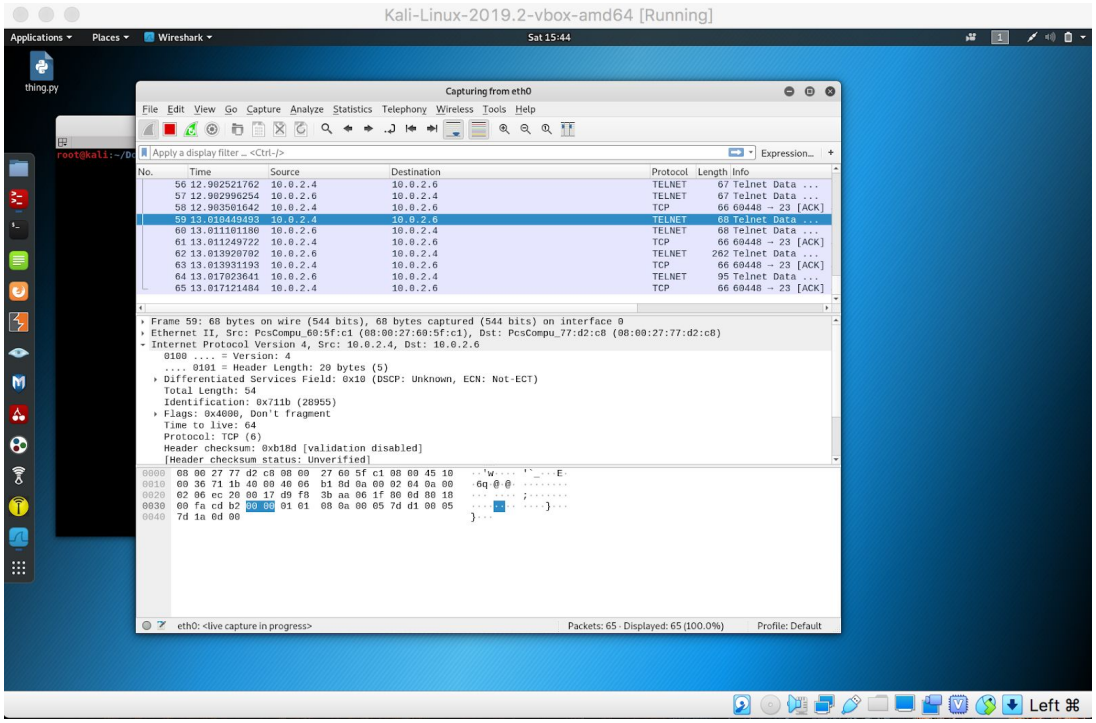
In this attack, victim and observer establish a telnet connection.

The attacker forges a TCP packet which has a next sequence and also an acknowledgement number, it then sends this to one of the two machines. In my implementation I used a program called shijack to hijack the computer which established the telnet connection. The computer that established the connection now has a frozen terminal and the attacker now has full access to the victim's computer. Attack successful - I was able to create a new file on desktop of victim.

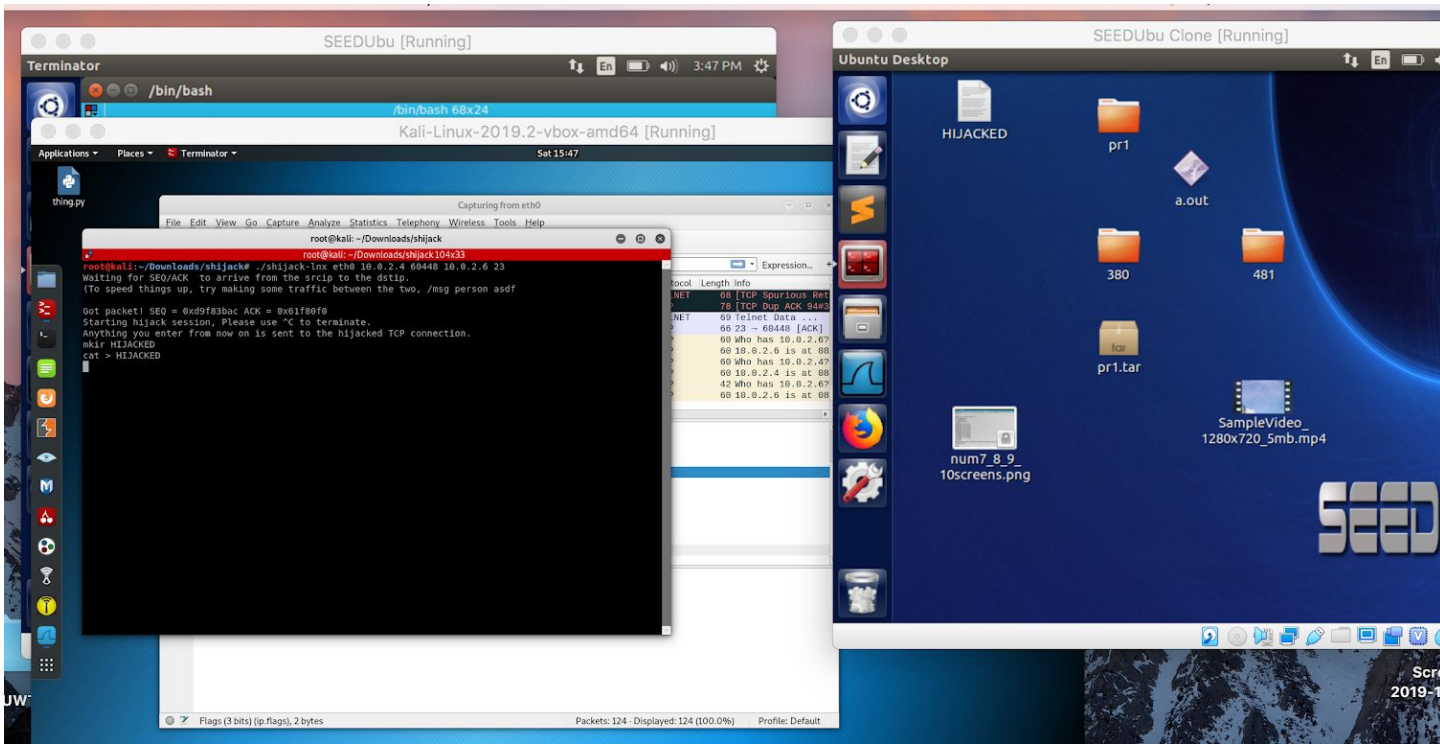
Victim and observer are connected via telnet



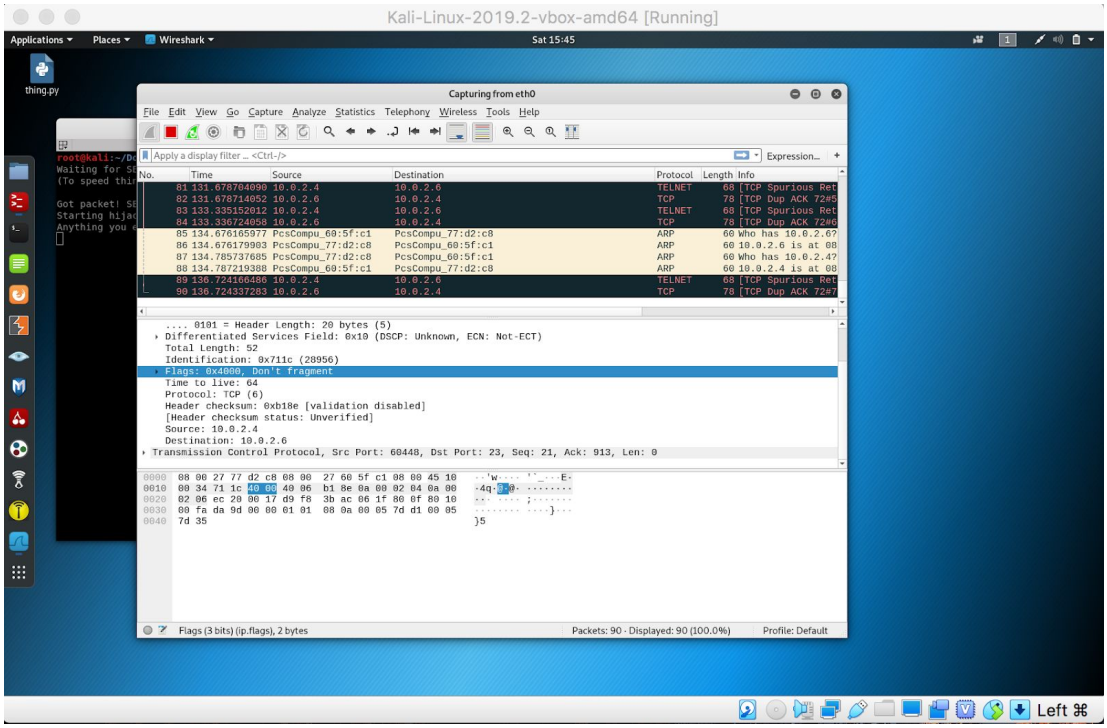
Attacker is observing traffic on Wireshark



Attacker hijacks connection using shijack program. Creates a file called HIJACKED.txt and it shows up on victim's computer.



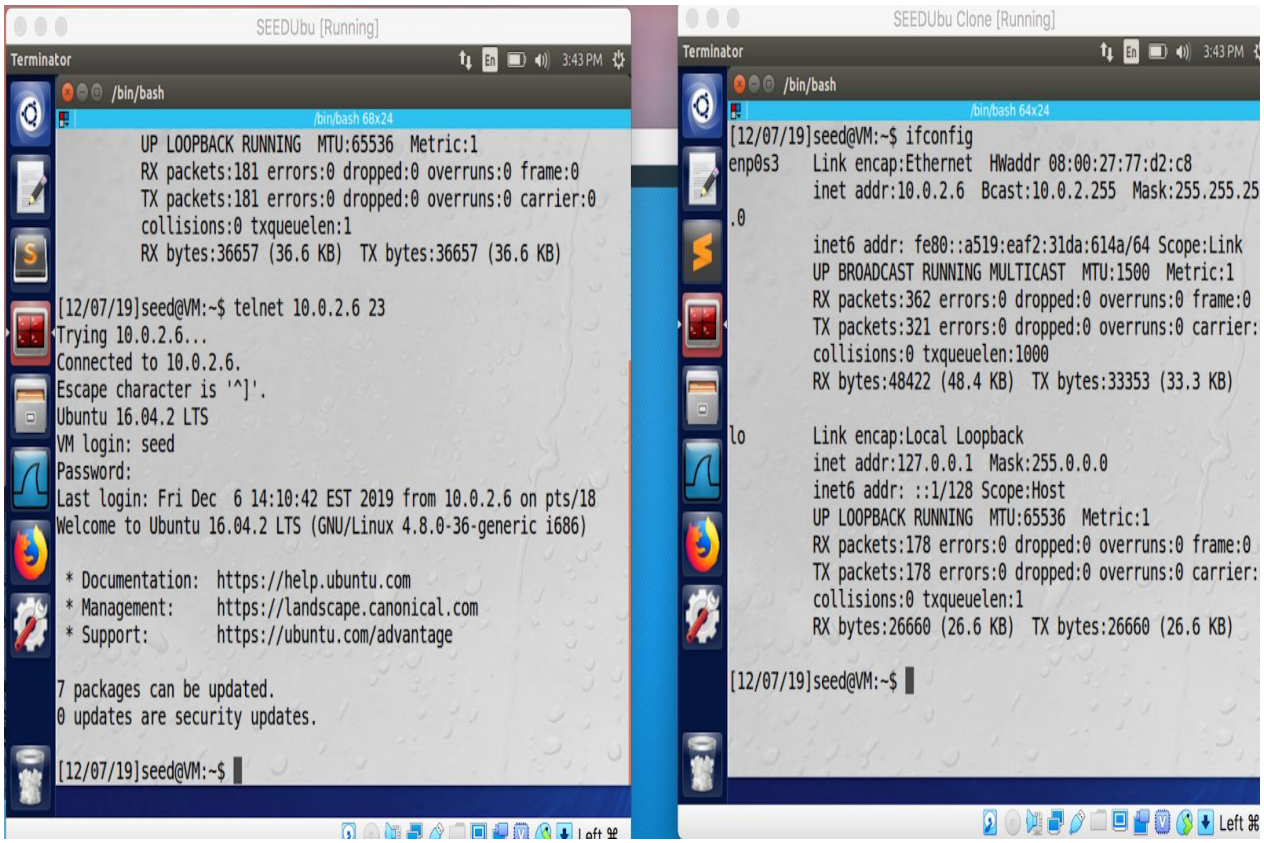
Wireshark view from attacker's computer during hijacking attack



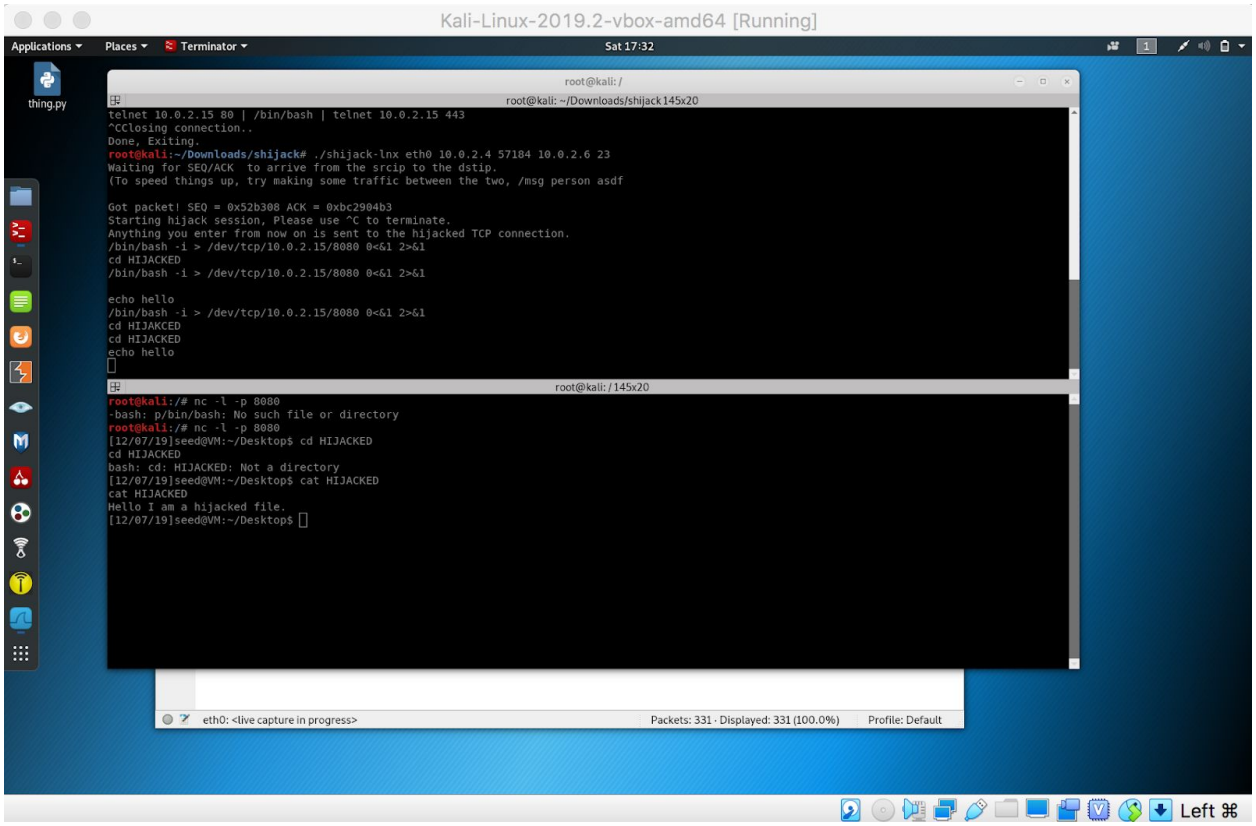
3.5 Task 5: Creating Reverse Shell using TCP Session Hijacking

With this attack the setup is more or less the same as the previous attacks. A telnet connection is initiated between A and B. The attacker hijacks the connection, but this time instead of sending only a few commands he sends the command that creates a reverse shell. I used the nc command to start listening on a selected port on attacker machine. I used shijack to hijack the tcp connection. I sent the command to the victim machine telling it to send the attacker machine all the output from its shell so everything that would normally be printed out goes to attacker and nobody else. Attack successful. Can observe cat command prints out file contents from victim computer on attacker's terminal.

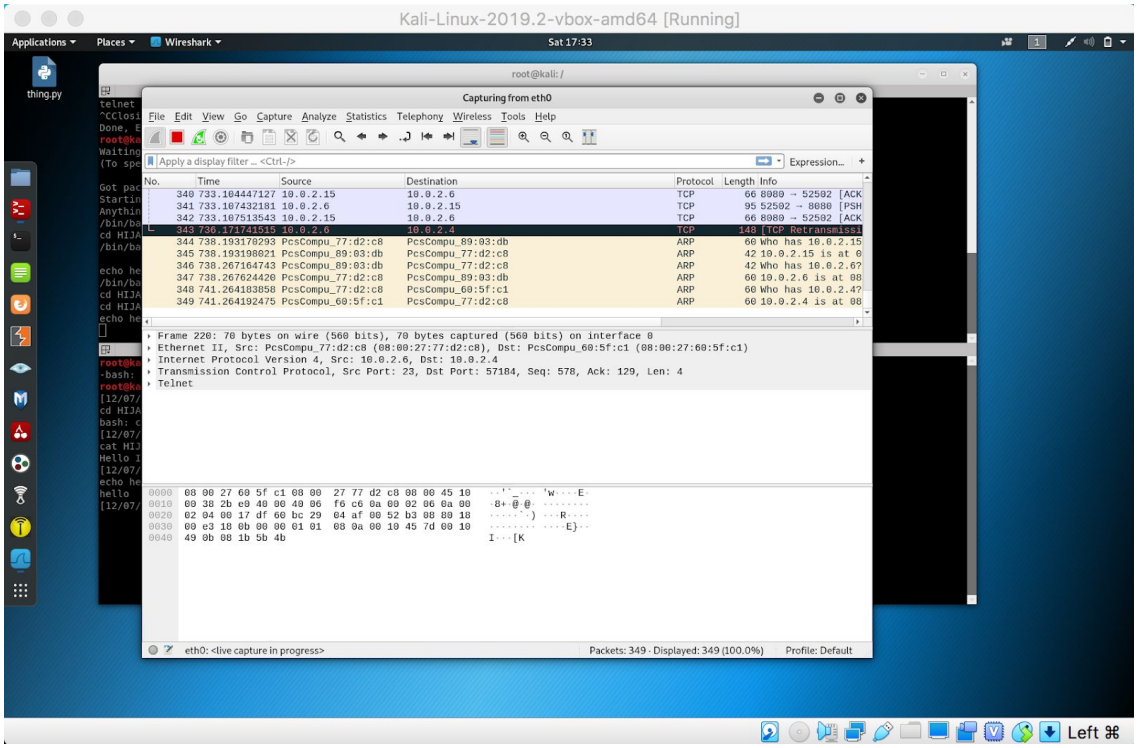
Establishing telnet connection between A and B



Attacker using shijack to hijack the tcp connection and send commands, and netcat for port listening. Took a few wrong commands to get it right.



Wireshark from Attacker's computer documenting network activity



END