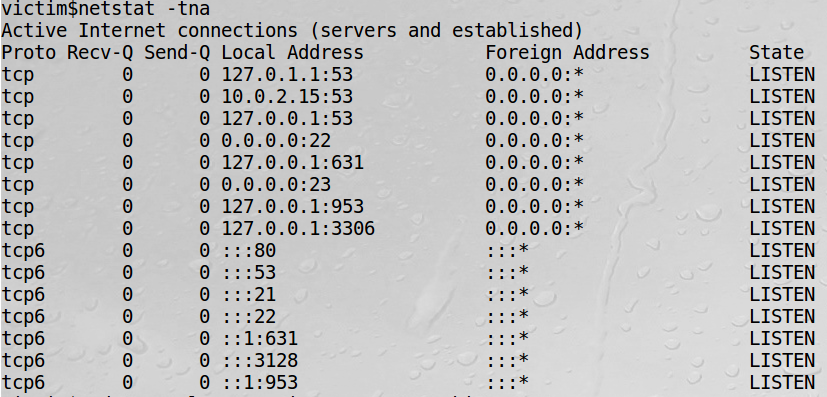
**3 – Lab Tasks**

**3.1 – Task 1: SYN Flooding Attack (30 points)**

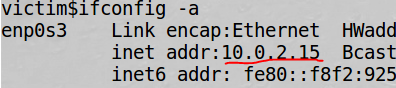
SYN flood attack with SYN cookies turned off…



Nobody listening to the victim machine beforehand!



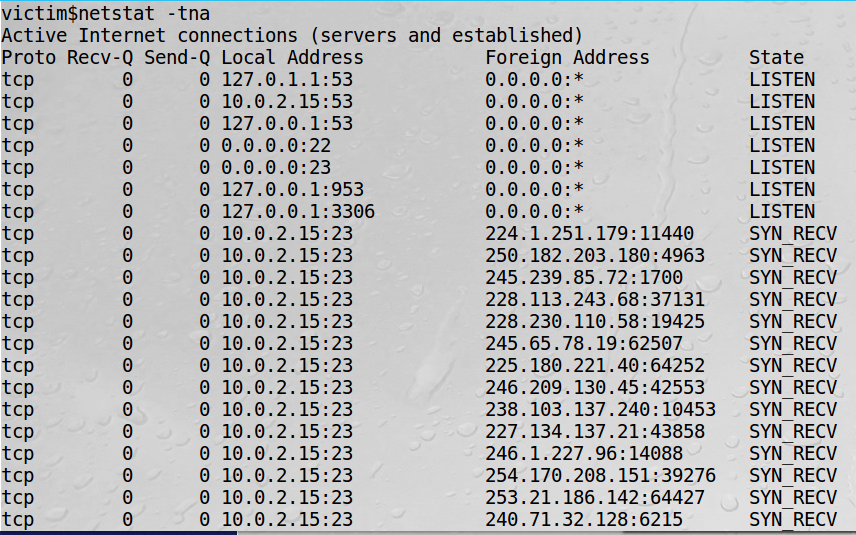
The following is the IP address of the victim terminal:



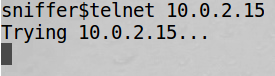
Then we go to our attacker terminal and type the following command:



And the attack is now ongoing. We can now see on our victim terminal that it is flooded with half-open three-way handshakes (indicated by the SYN\_RECV status).



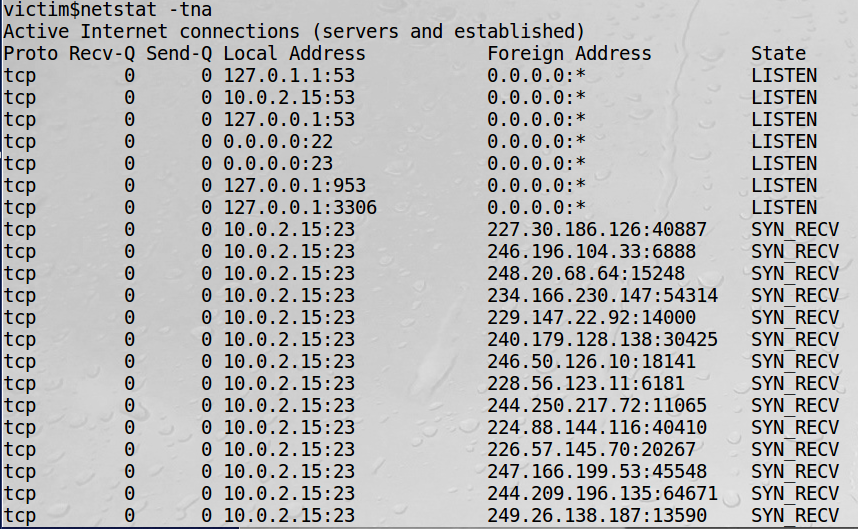
We can then try to connect to our victim terminal but it doesn’t work because all these ports are “occupied” by the SYN flood attack.



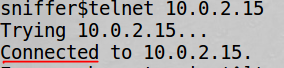
The SYN flood attack was therefore successful!

However, with SYN cookies turned back on….





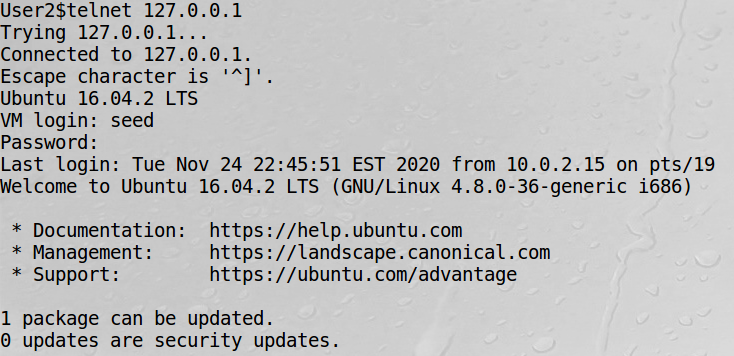
While the SYN flood attack still seems to occupy the same number of ports (seemingly successful), the sniffer terminal is able to connect to the victim terminal right away without any trouble.



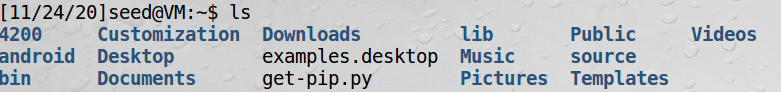
So it would seem that with the SYN cookies countermeasure turned on, the SYN flood attack is only successful in appearance, but not in actually hindering the functionality of the victim.

**3.2 – Task 2: TCP RST Attacks on telnet and ssh Connections (40 points)**

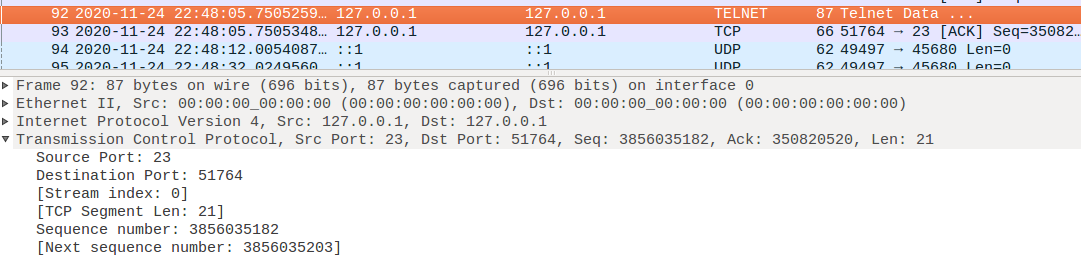
First, we establish a telnet connection between User and User2.



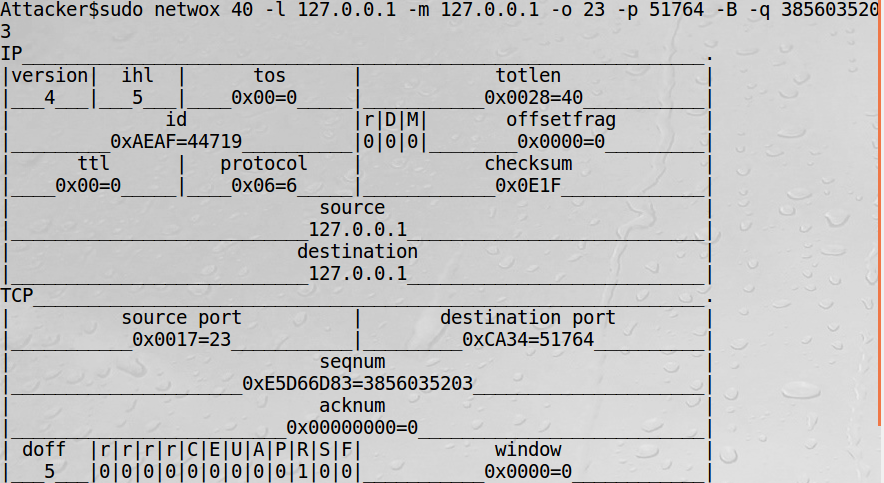
Now the two are connected. Then we do some command on User 2 to generate some traffic.



Then we view this traffic in Wireshark. Find the 2nd to last packet (TELNET) and view its information (source port, destination port, and the next sequence number).



Now we run our attack using this information to create a spoofed RST packet.



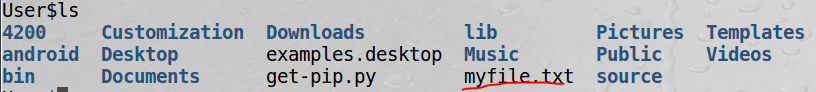
And as expected, this spoofed RST packet attack has caused the connection between User and User2 to be closed.



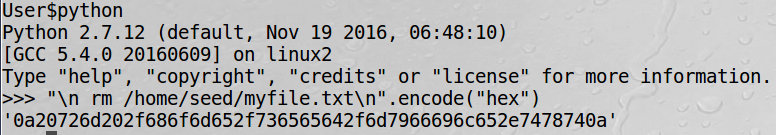
(NOTE: Dr. Cheng told us in lecture on 11/24 that we were not required to use Scapy for this task.)

**3.3 – Task 3: TCP Session Hijacking on Telnet Connection (40 points)**

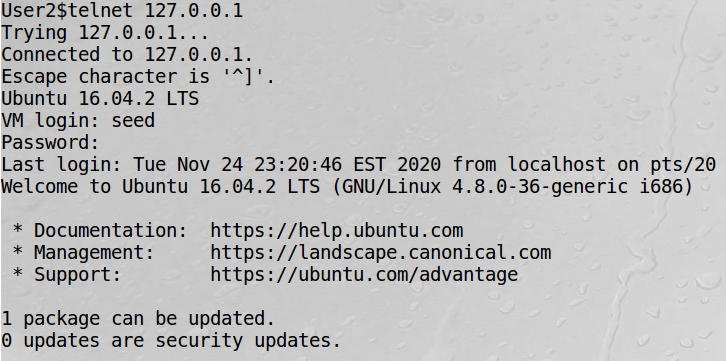
First, we create the file /home/seed/myfile.txt.



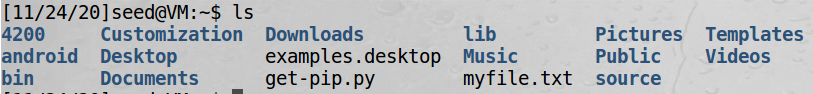
Then we run the python command to get the hex code for the command to delete this file.



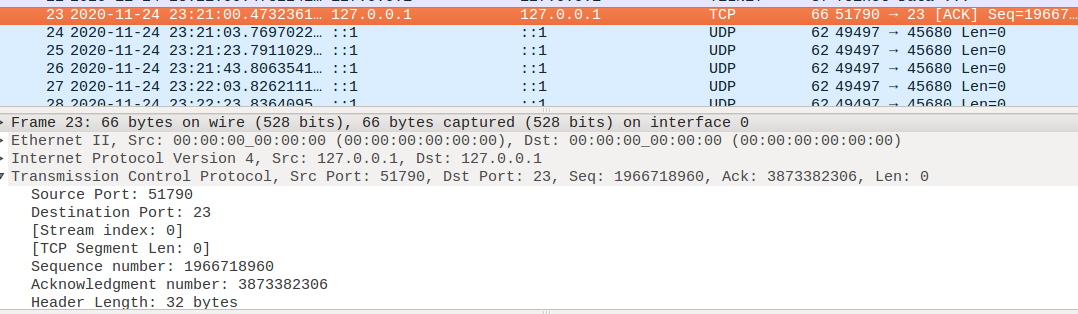
After this, we establish a telnet connection between User and User2 again.



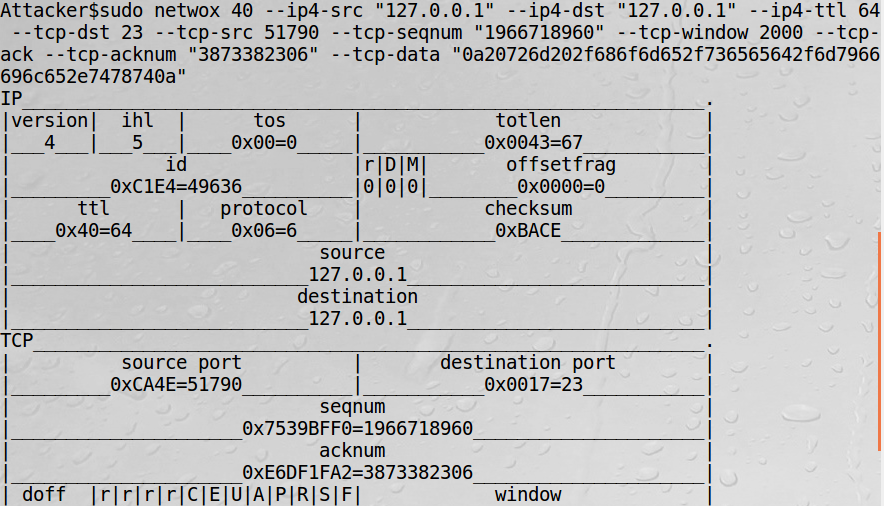
Then we create traffic using a simple ls command.

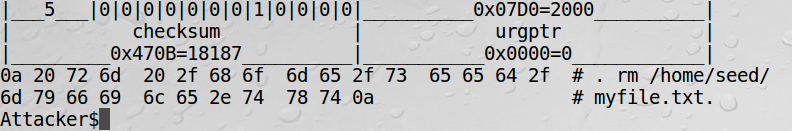


Then we view this traffic via Wireshark again. We click on the last TCP packet sent this time and view its info (source port, destination port, sequence number, and acknowledgement number).

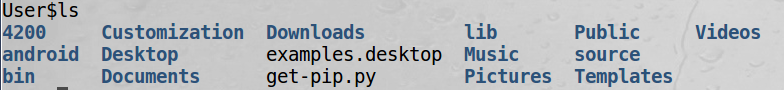


We then use this information to launch our hijacking attack on the telnet connection between User and User2. The --tcp-data flag contains the hex code to execute the command to delete myfile.txt on User’s home directory.





By hijacking the telnet connection, we are able to successfully run the malicious command and delete myfile.txt from User’s home directory.



The file now no longer exists.