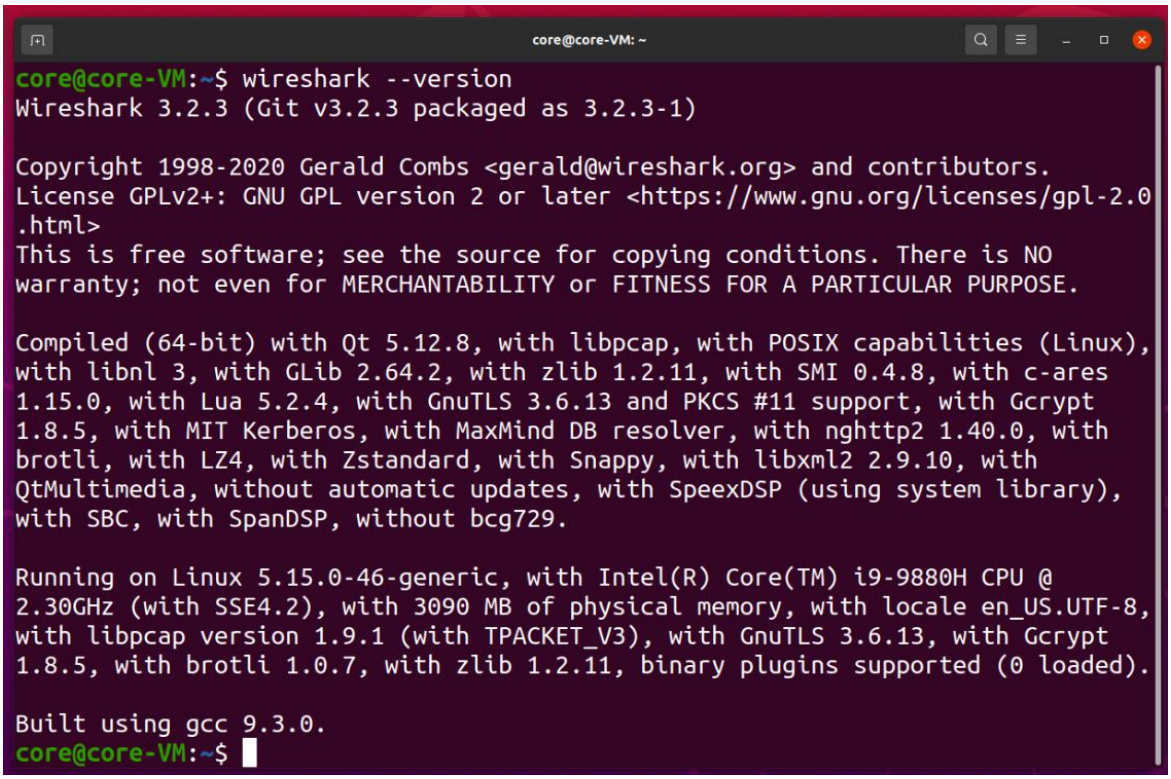
Investigate and Characterize the Impact of DoS Attacks from Attackers on the Webserver using Wireshark, the Protocol Analyzer

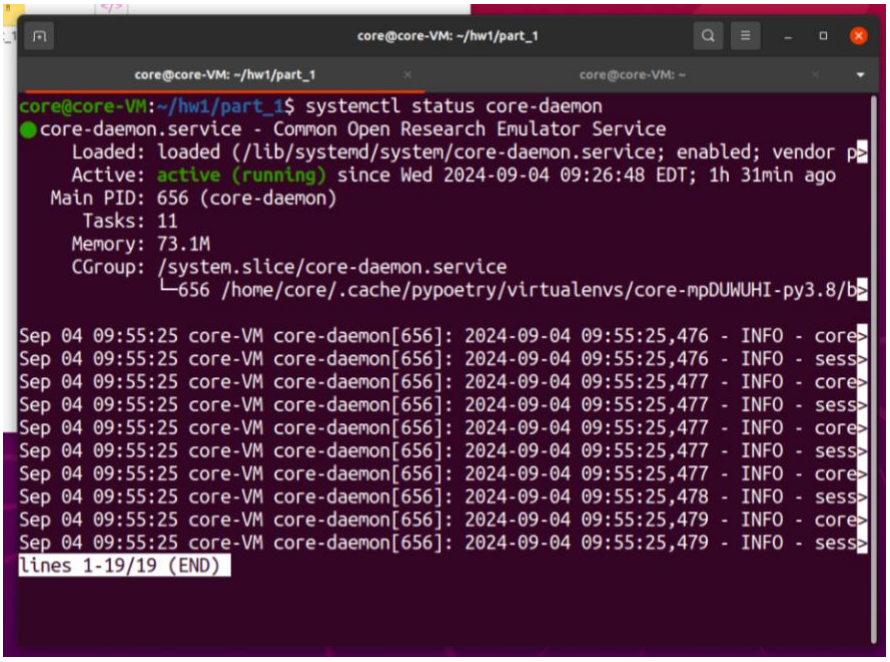
Install Wireshark

To install Wireshark, I used the following commands, `sudo apt-get update`, `sudo apt-get install wireshark` and `wireshark --version`.

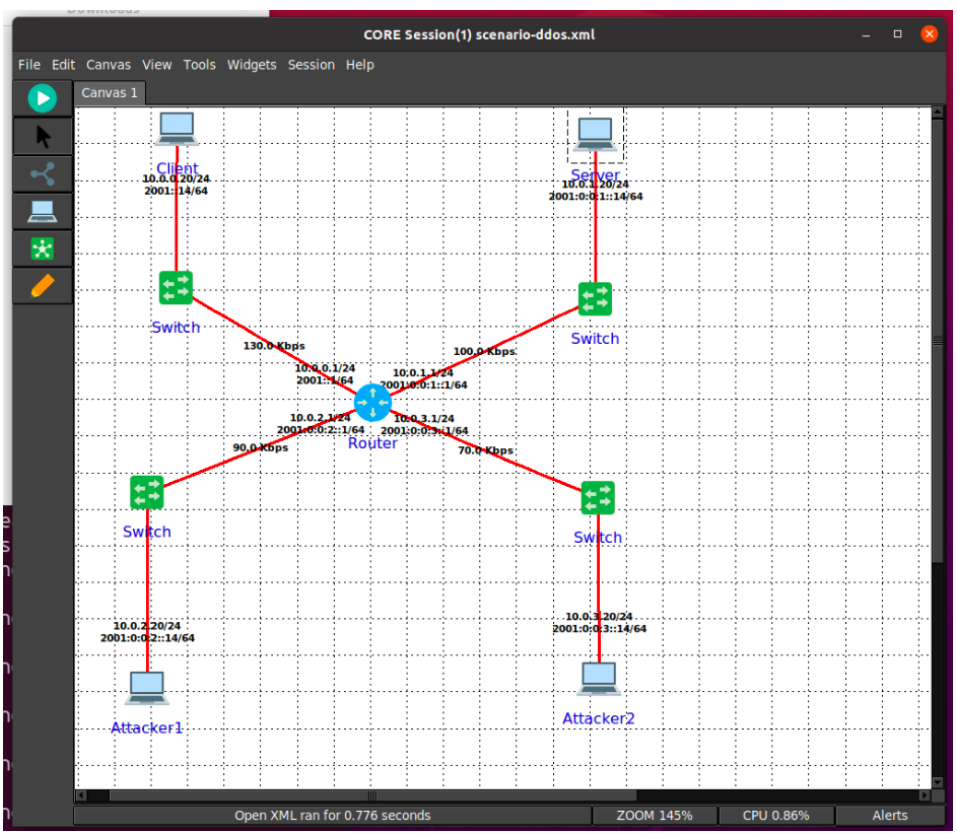


Setting Up the CORE Scenario

I first ensured the CORE daemon was running to set up the core scenario. To check that, I used the command, `systemctl status core-daemon`

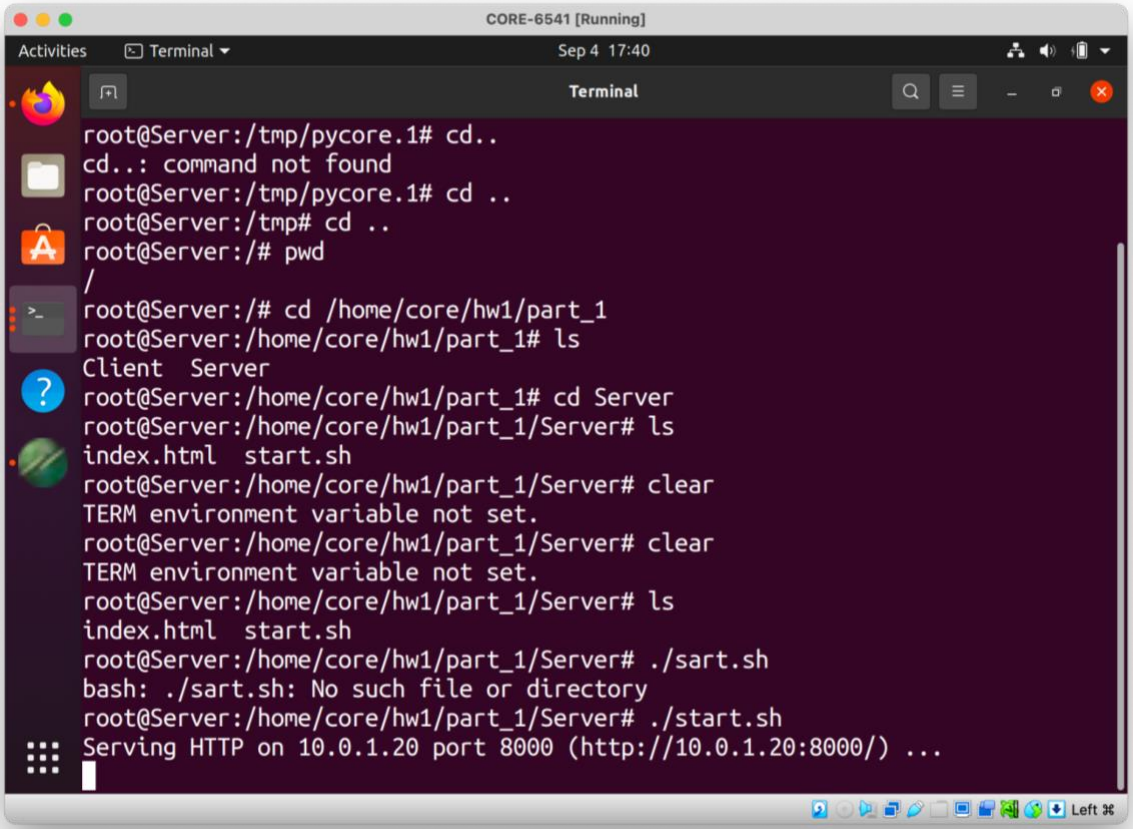


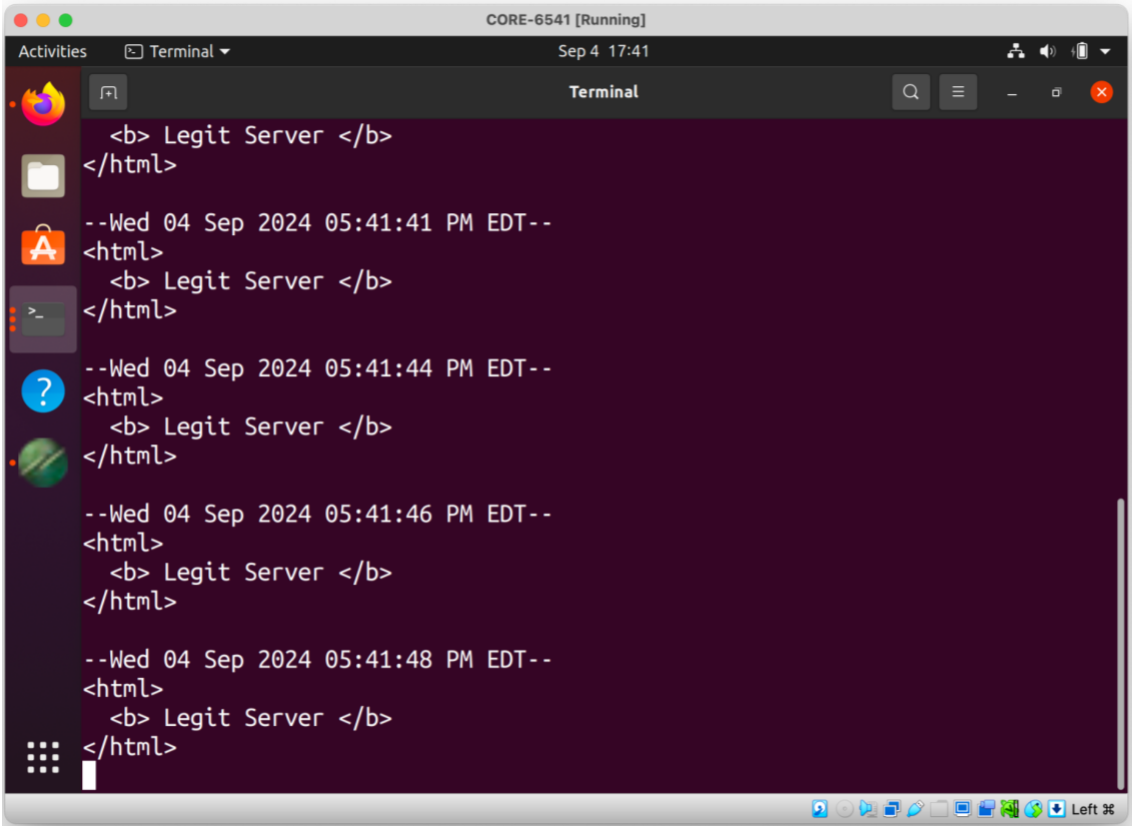
Then, I loaded the file “scenario-ddos.xml” in the CORE GUI.

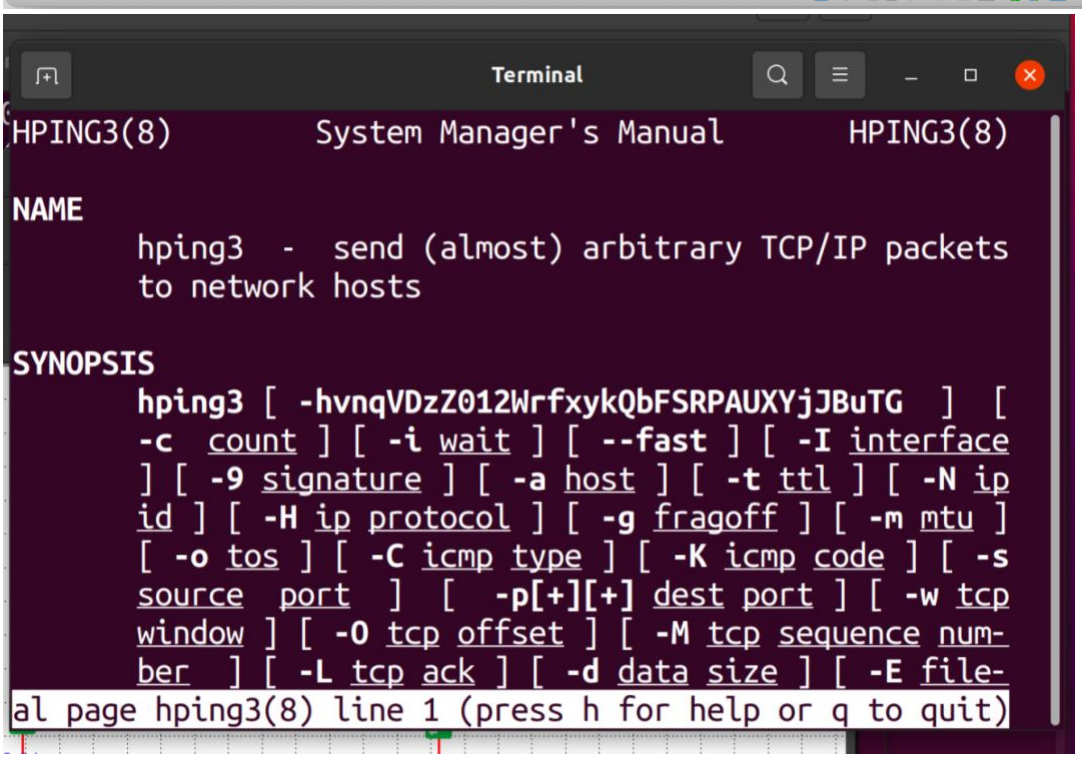


Running the Scenarios

First, I ran the scenario by clicking the green play button. Then, I ran start.sh in the server node using the command, `./start.sh`. Next, I ran the run_curl.sh in the client node using the command, `./run_curl.sh`. Followed I ran the `man hping3` command to read the manual of the hping3.

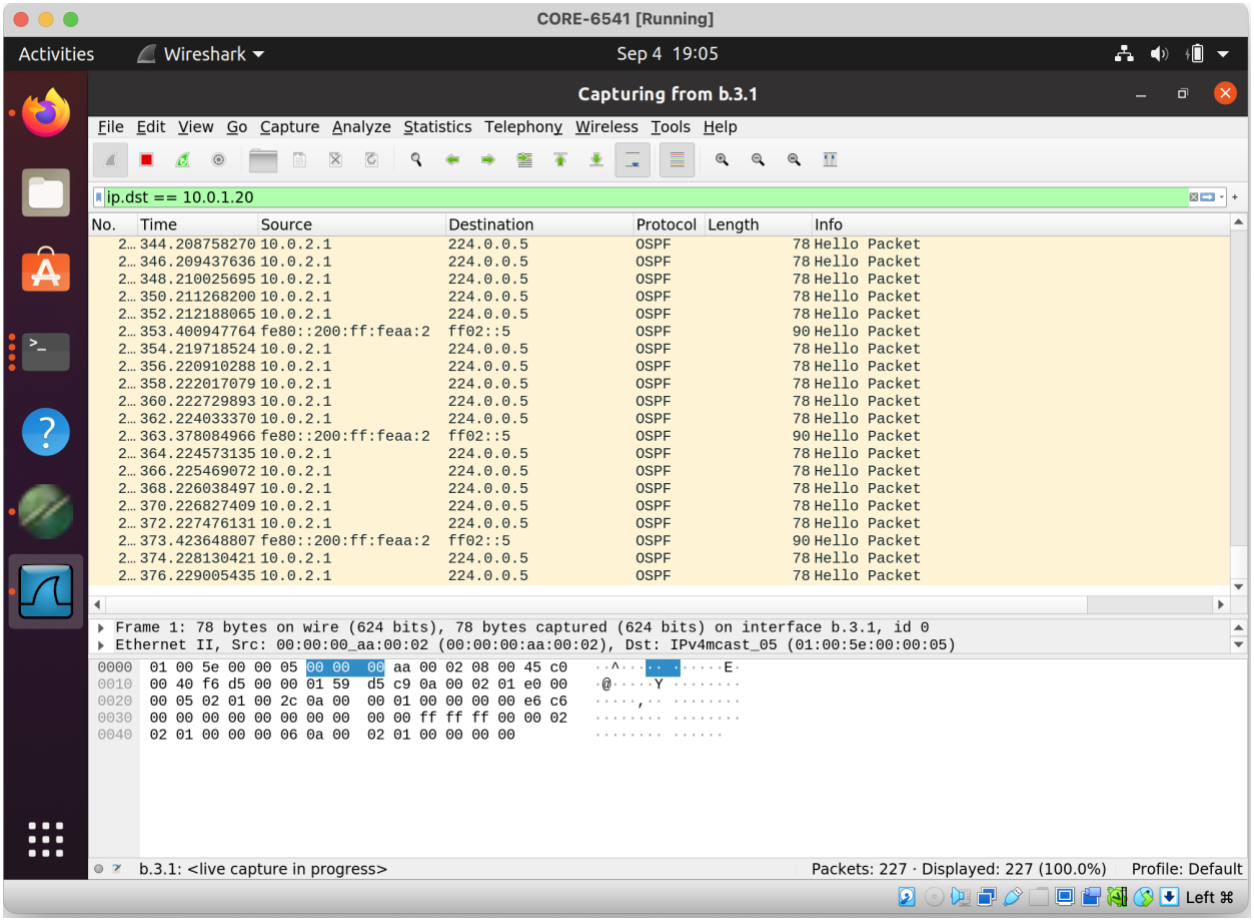



CORE-6541 [Running]
Activities
Terminal
Sep 4 17:41
Terminal
 Legit Server
</html>
--Wed 04 Sep 2024 05:41:41 PM EDT--
<html>
 Legit Server
</html>
--Wed 04 Sep 2024 05:41:44 PM EDT--
<html>
 Legit Server
</html>
--Wed 04 Sep 2024 05:41:46 PM EDT--
<html>
 Legit Server
</html>
--Wed 04 Sep 2024 05:41:48 PM EDT--
<html>
 Legit Server
</html>
Left ⌘


Terminal
HPING3(8) System Manager's Manual HPING3(8)
NAME
hping3 - send (almost) arbitrary TCP/IP packets to network hosts
SYNOPSIS
hping3 [-hvnqVDzZ012WrfxykQbFSRPAUXYjJBuTG] [-c count] [-i wait] [--fast] [-I interface] [-9 signature] [-a host] [-t ttl] [-N ip id] [-H ip protocol] [-g fragoff] [-m mtu] [-o tos] [-C icmp type] [-K icmp code] [-s source port] [-p[+][+] dest port] [-w tcp window] [-O tcp offset] [-M tcp sequence number] [-L tcp ack] [-d data size] [-E file-
al page hping3(8) line 1 (press h for help or q to quit)

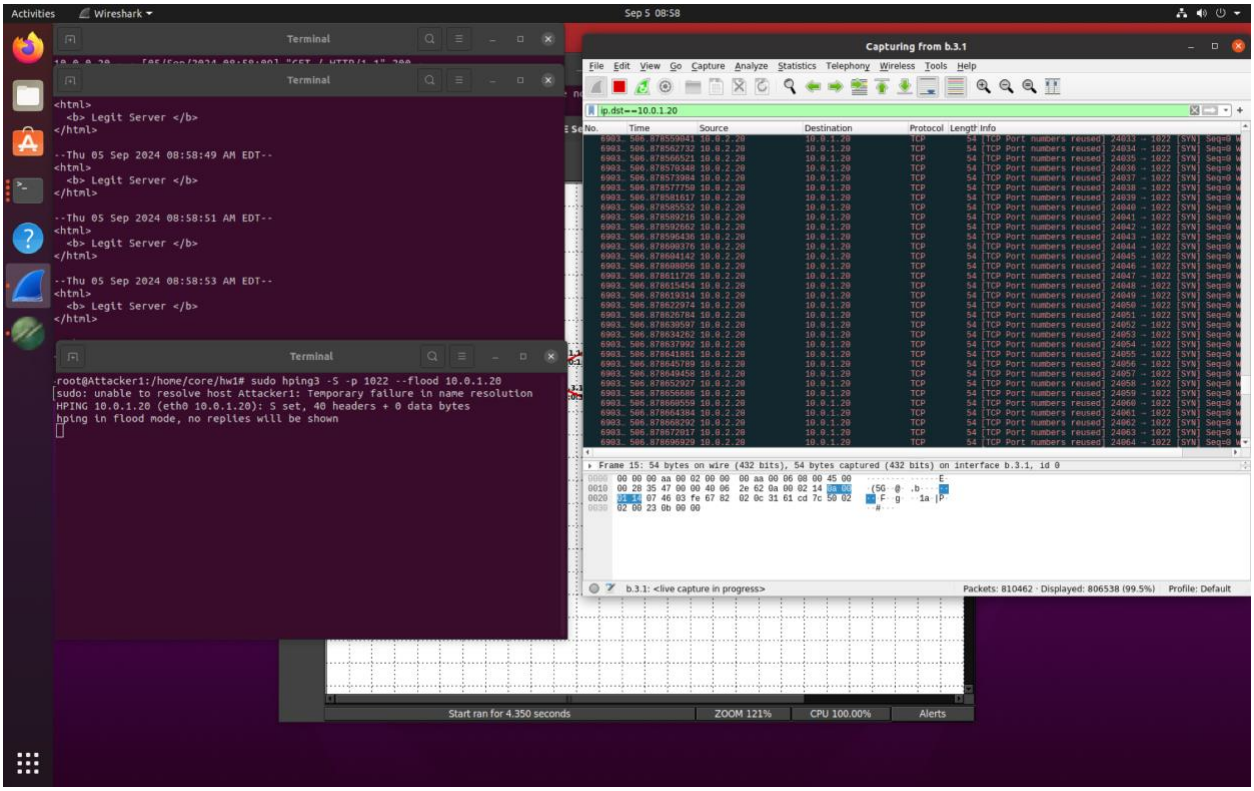
1. Run the Wireshark:

First, run the Wireshark with the filter `ip.dst==10.0.1.20` and then run the `./run_curl.sh`, command to see the traffic through the client and server using Wireshark.



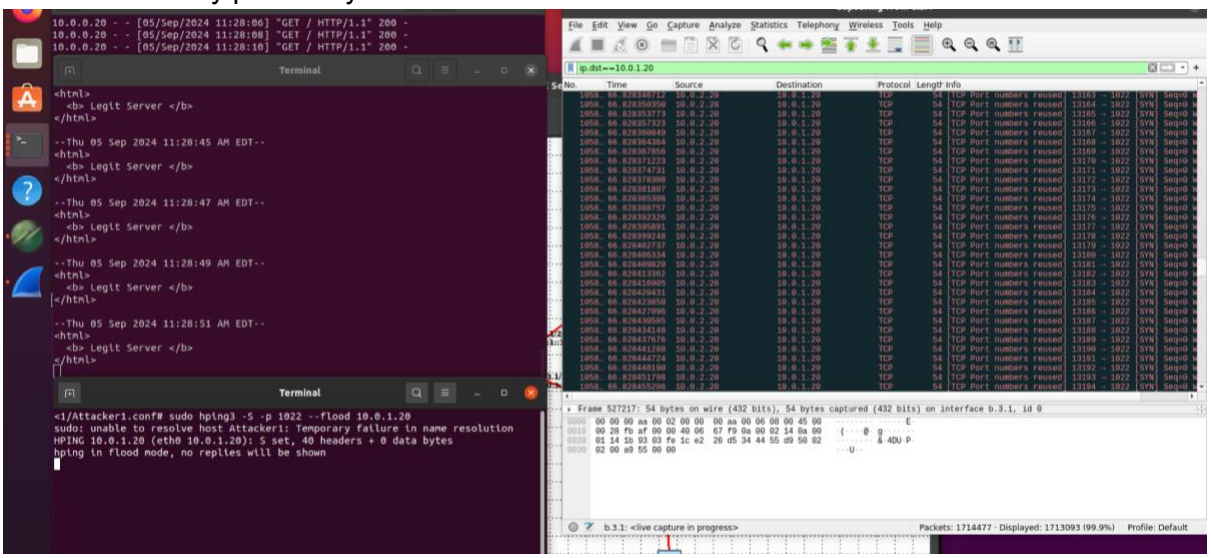
a) SYN Flood DoS Attack

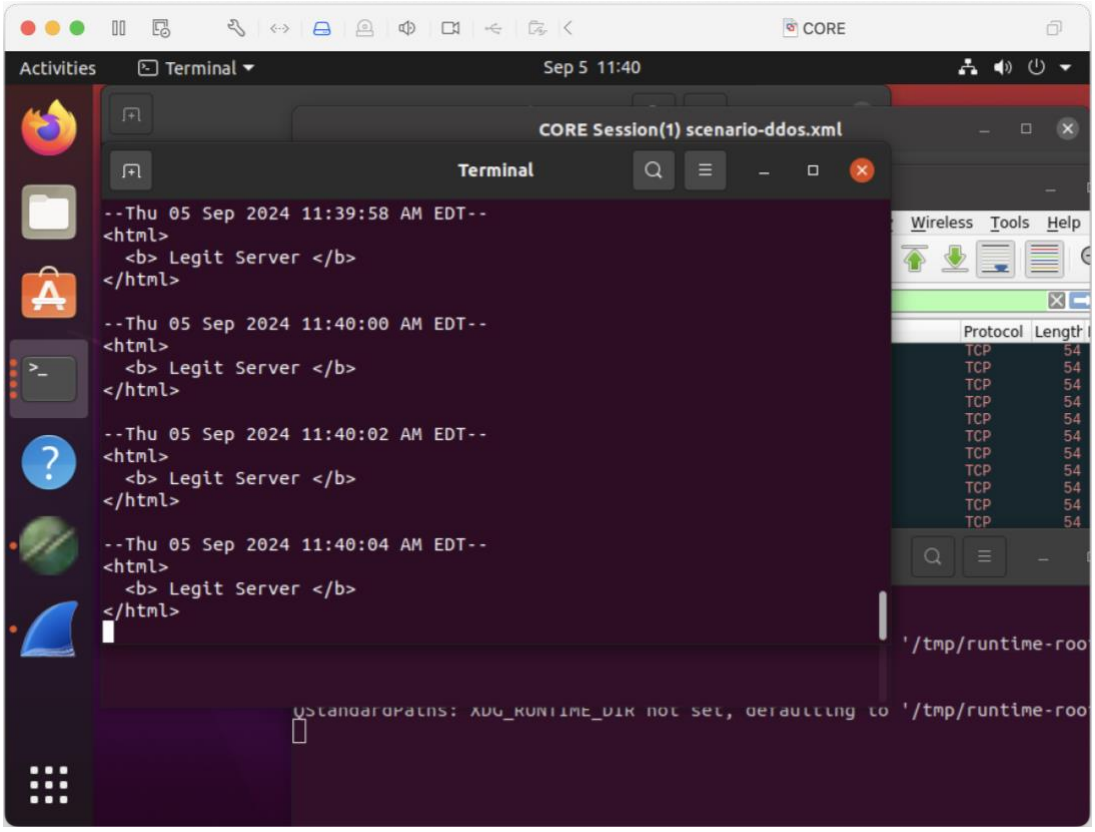
Did an SYN flood DoS attack on the server from attacker 1 using the command, `sudo hping3 -S -p 1022 --flood 10.0.1.20.` The -S flag is used to provide SYN packets, and -p sets the port i.e. 1022.



b) Is the Attack Effective or Not?

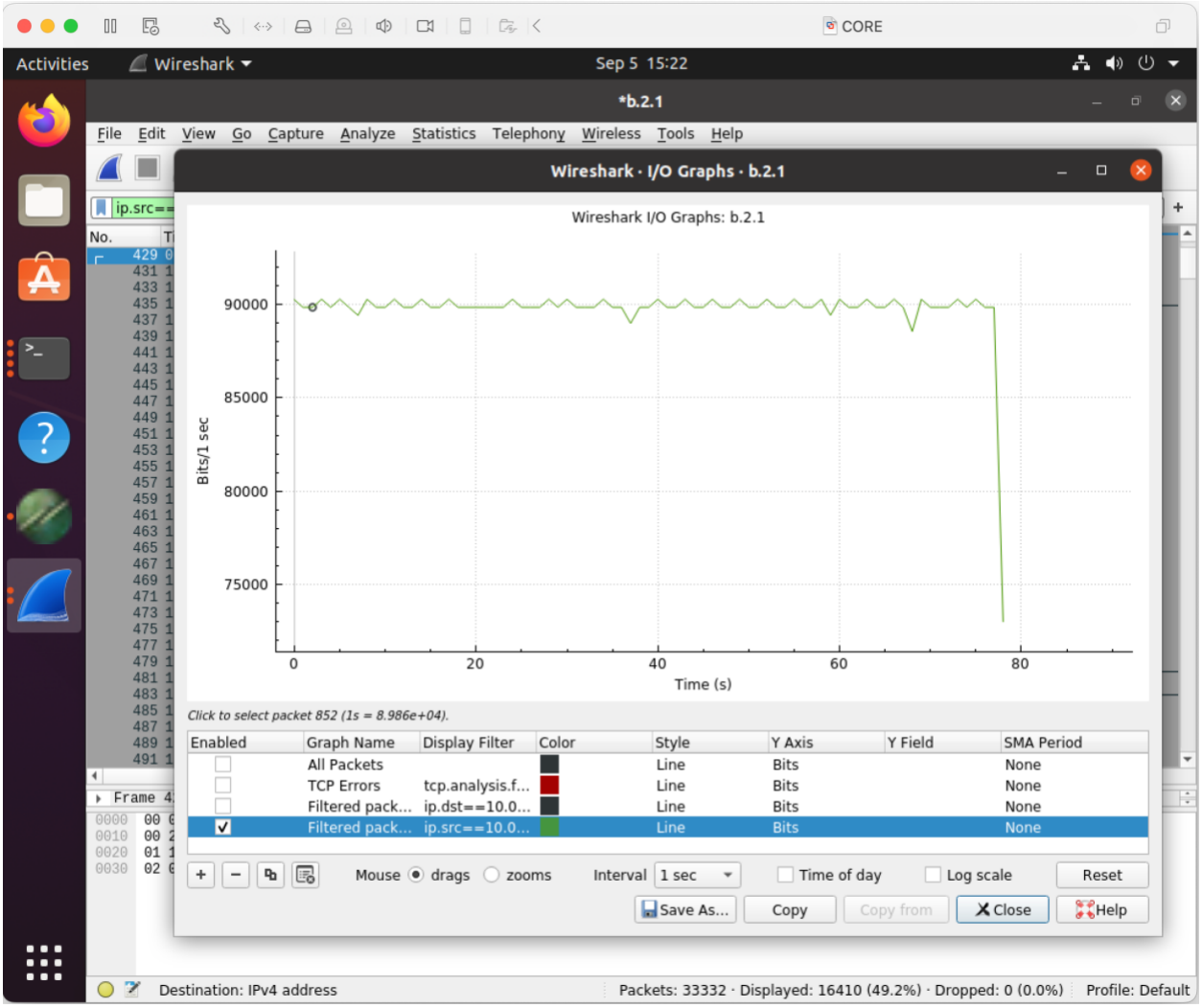
The attack is not effective because the threshold of the server that it can accommodate is higher than the packets being sent. As a result, the Client kept receiving the response for its curl command. The document said to stop the attack after 1 minute but I stopped after 10 mins to see if there's any possibility for the attack to be effective but it's not effective.





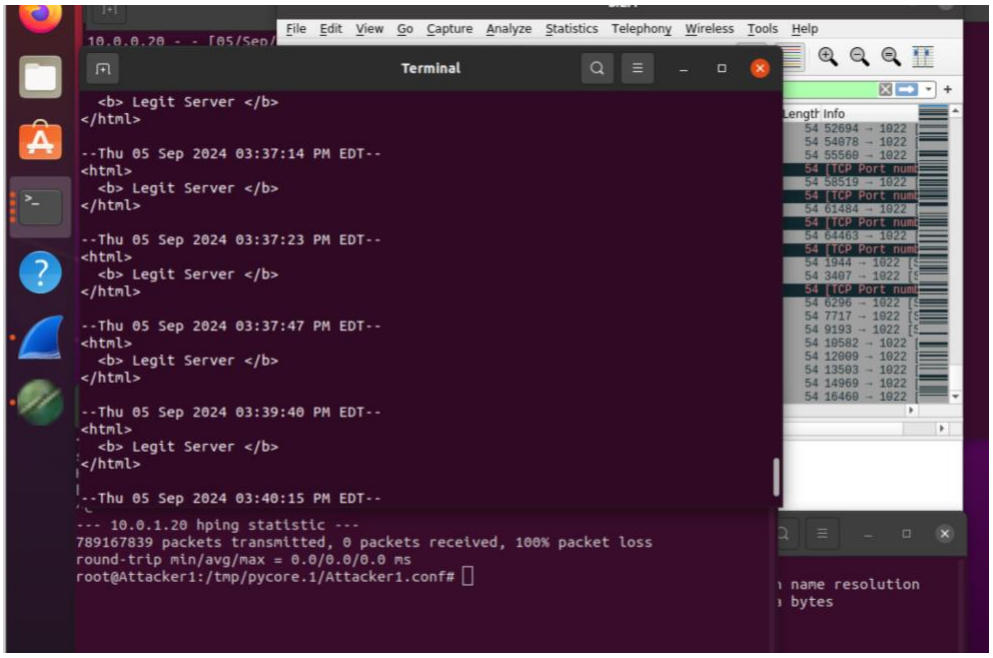
c) Magnitude of the Attack

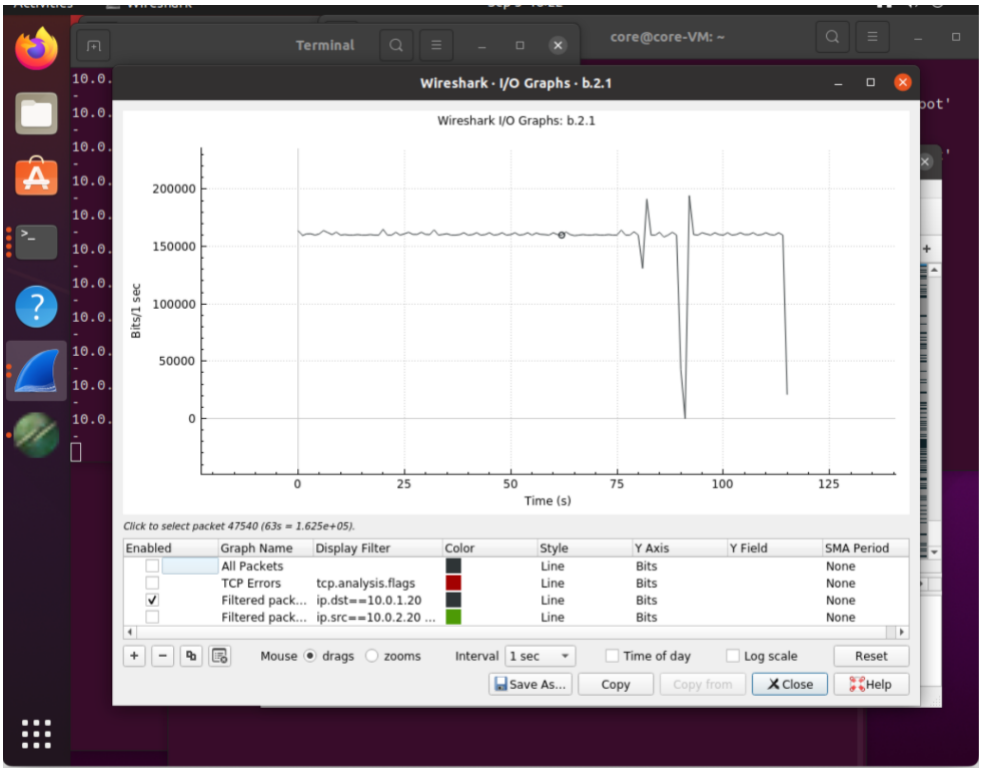
To find the magnitude of the attack, we've to find the peak value on the Y-axis. This represents the highest bandwidth consumption of the attack. From, the graph the values range between `75000 bps to 90000 bps`. The peak value on the graph seems to be close to 90000 bps. This means the magnitude of the attack is approximately 90 kbps. Therefore, the attack is consuming approximately `90000 bps/90 kbps` at its peak as shown in the graph.



d) SYN Flood Attack from Attacker 1 and Attacker 2

I did a SYN Flood attack on the server node but this time from both Attacker 1 and Attacker 2 nodes. As a result, the attack was effective and the client node stopped receiving responses from the server for almost `2 minutes` because we made a SYN Flood attack (DoS) on the server. The graph shows a range between `0 - 200000 bps` and a stable reliable line of traffic of around `150000 bps` with some fluctuations. This concludes the magnitude of the attack is approximately `200000 bps/200 kbps`.

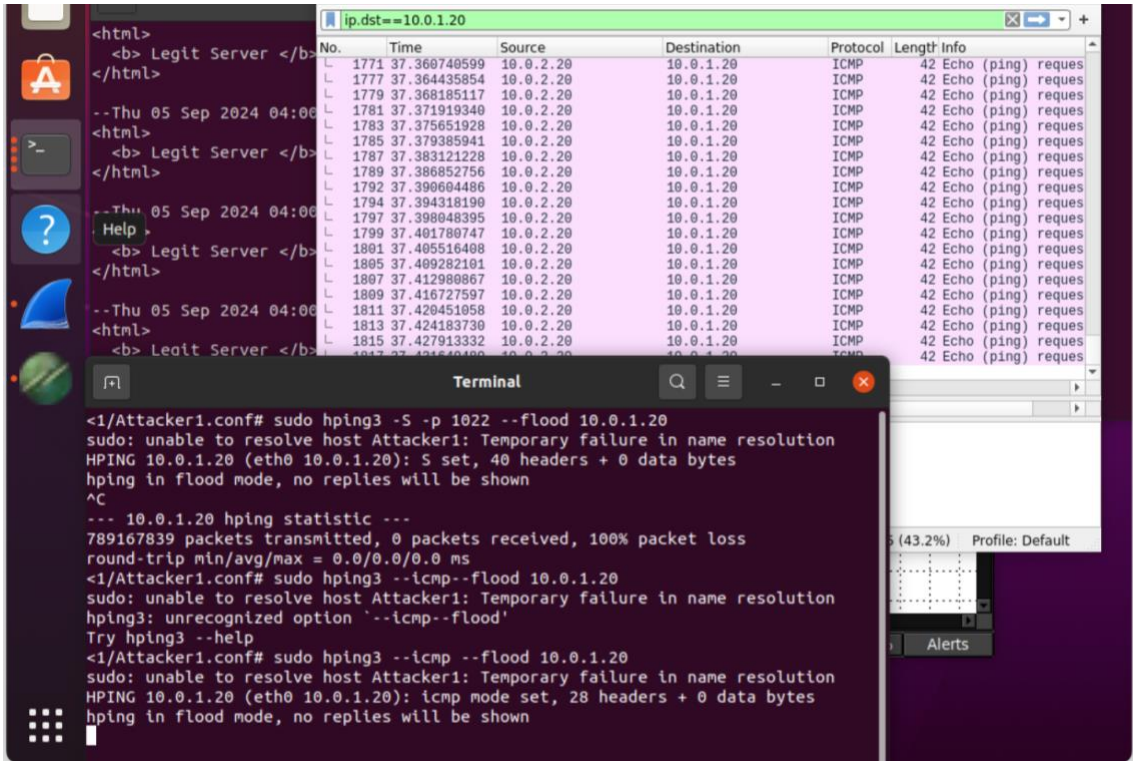


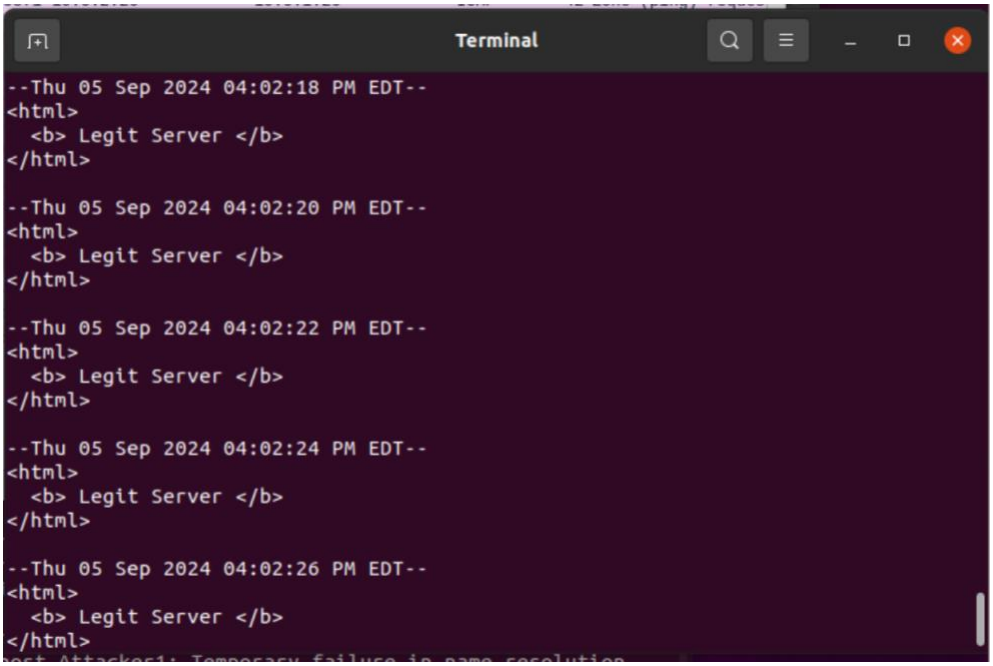


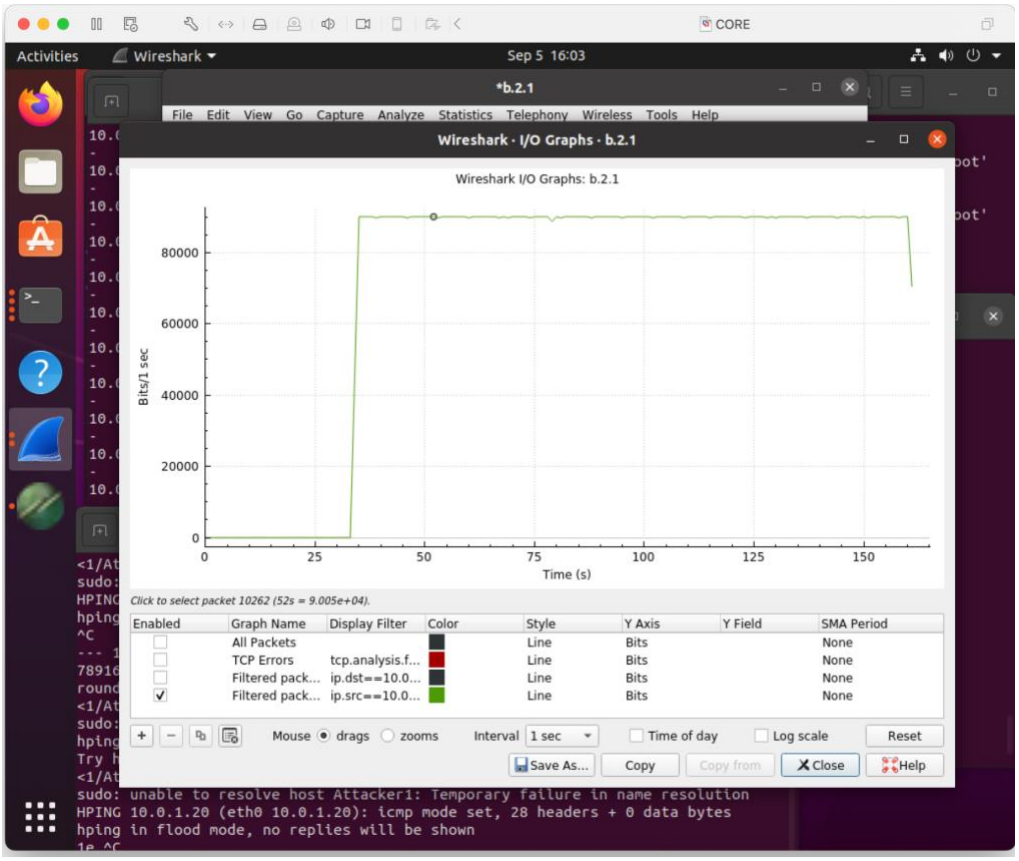
e) ICMP Flooding Attack

This time I did an ICMP Flood attack on the server from Attacker 1 and the attack was not effective because the client kept receiving the response from the server. The graph ranges

between `0 and 100000 bps` and shows a stable line of traffic pattern at approximately `80000 bps` i.e. 80 kbps and also the peak traffic rate during the attack is approximately the same. Therefore, the magnitude of the attack is `80000 bps/80 kbps`.

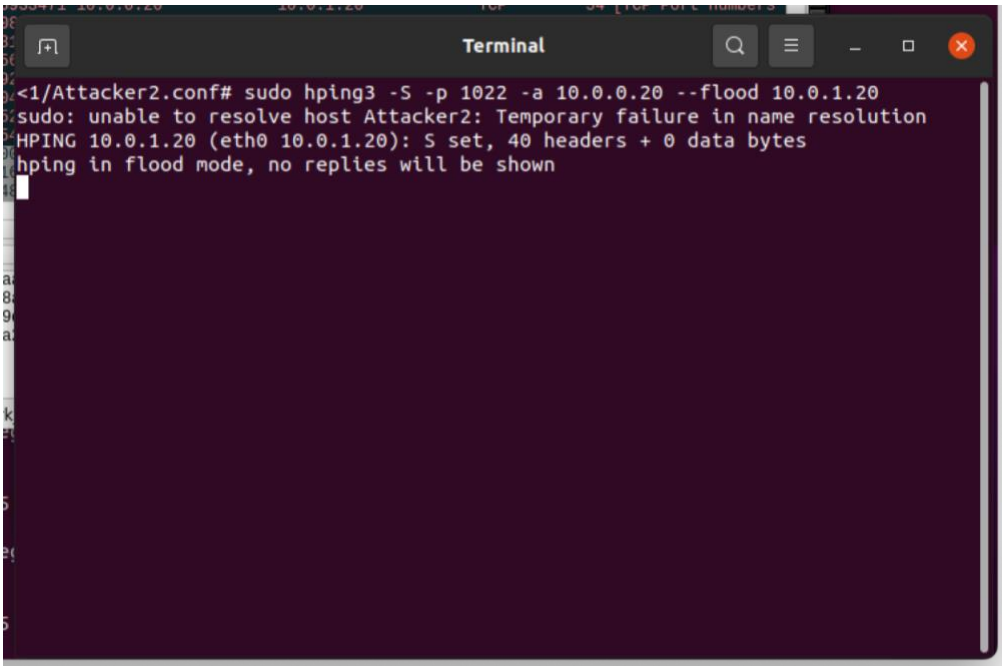


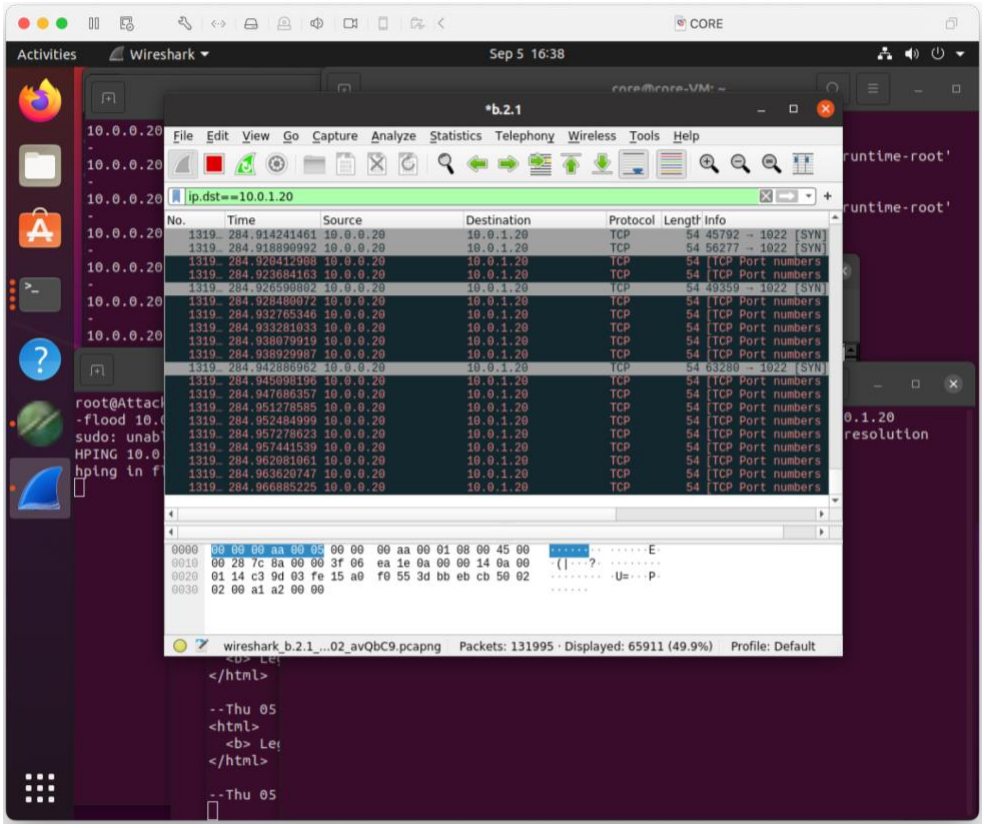


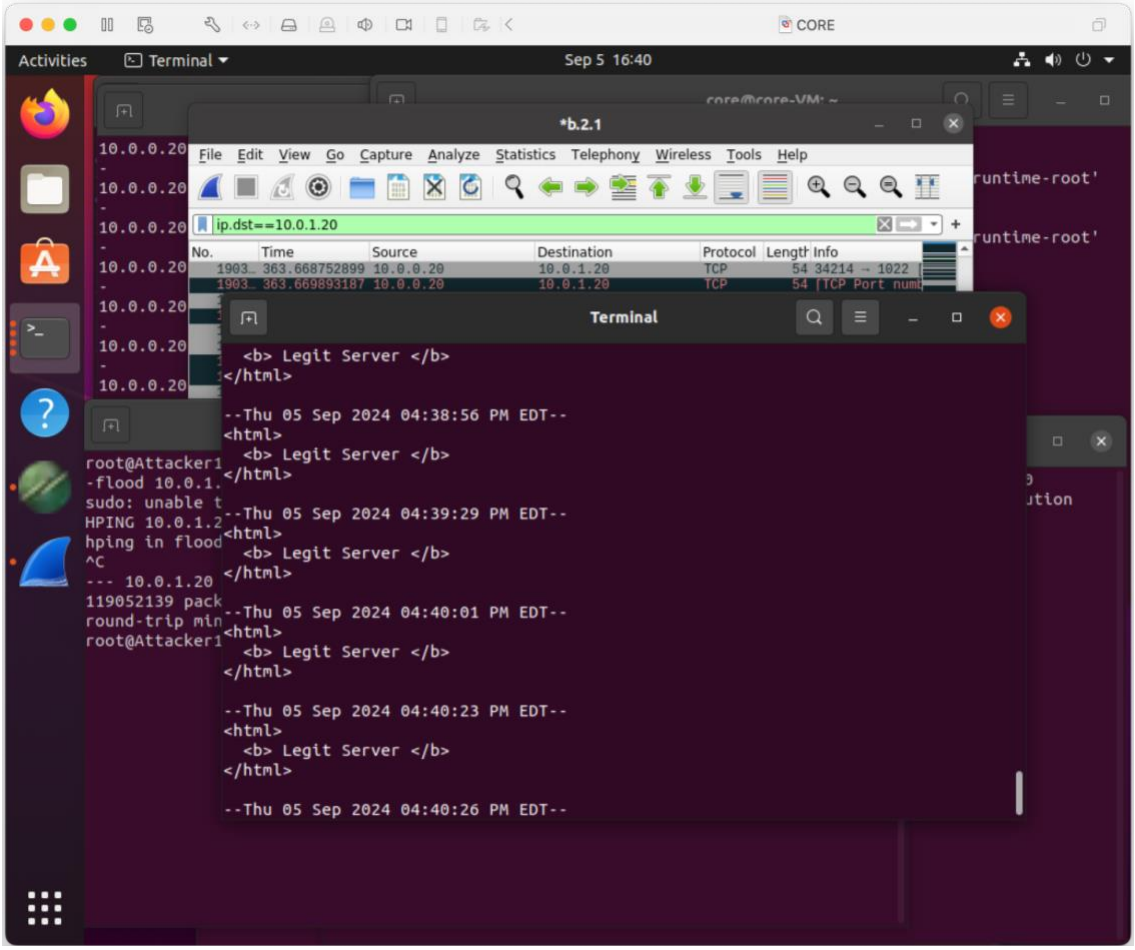


f) SYN Flooding Attack with Address Spoofing

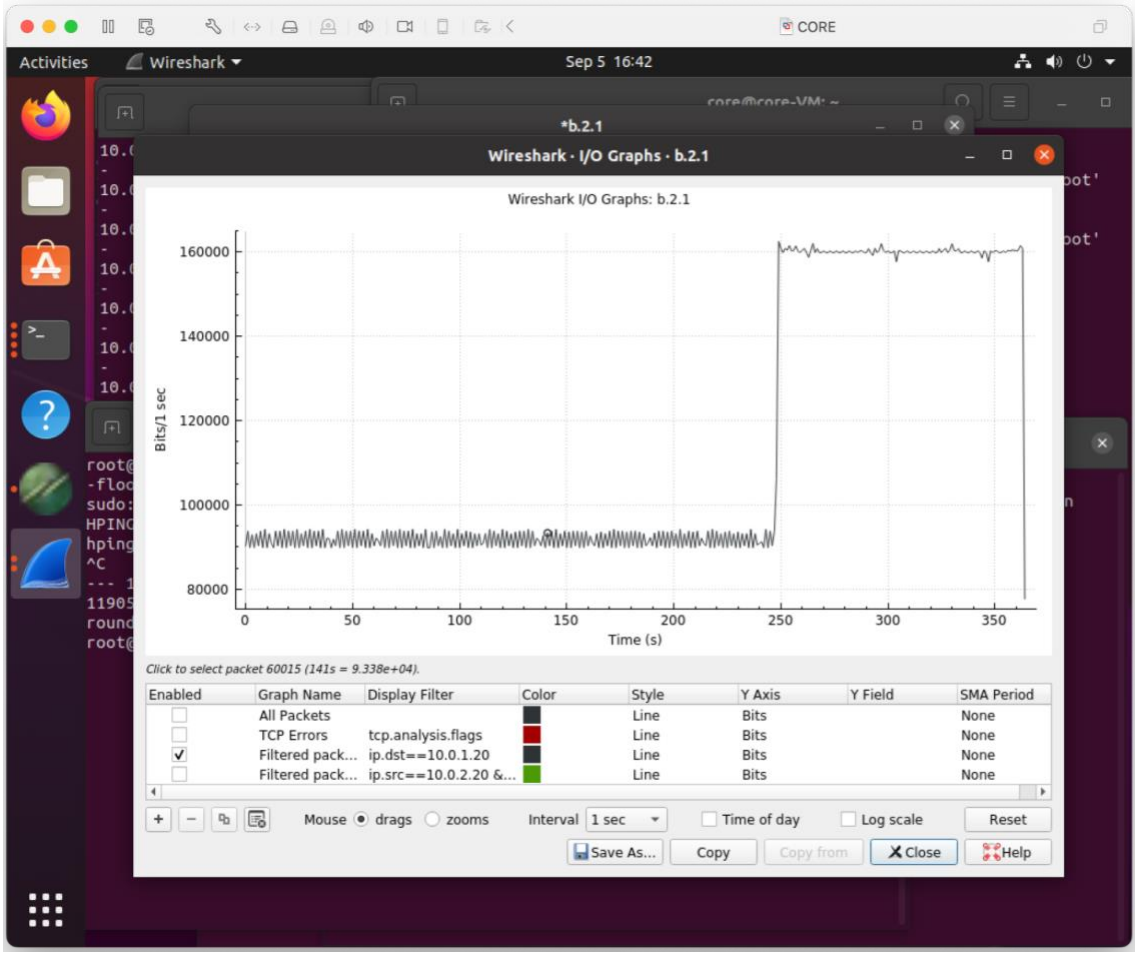
I did the SYN Flood attack on the server but changed the command so that the attack reflects the source address as client but originally its source address is attacker 1 node and the same is done from attacker 2 node too. I used the command `sudo hping3 -S -p 1022 -a 10.0.0.20 --flood 10.0.1.20.` I can also see there is a lag in response to the client in 1-minute intervals which means the attack is effective.


Terminal
<1/Attacker2.conf# sudo hping3 -S -p 1022 -a 10.0.0.20 --flood 10.0.1.20
sudo: unable to resolve host Attacker2: Temporary failure in name resolution
HPING 10.0.1.20 (eth0 10.0.1.20): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown


CORE
Activities
Wireshark
Sep 5 16:38
*b.2.1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.dst==10.0.1.20
No. Time Source Destination Protocol Length Info
wireshark_b.2.1_...02_avQbC9.pcapng
Packets: 131995 · Displayed: 65911 (49.9%)
Profile: Default



The IO graph shows traffic pattern ranges between `80000 bps to 160000` bps. Initially, Attacker 1 node is spoofed to the client's IP, generating traffic at approx. `90000 bps/90 kbps`. After some time, Attacker 2 node is spoofed to the client's IP, and this resulted in a significant increase in traffic with the combined attack reaching a peak i.e. magnitude of `160000 bps/160 kbps`. See the graph below for a better view.


CORE
Activities
Wireshark
Sep 5 16:42
*b.2.1
Wireshark · I/O Graphs · b.2.1
Wireshark I/O Graphs: b.2.1
Bits/1 sec
160000
140000
120000
100000
80000
0
50
100
150
200
250
300
350
Time (s)
Click to select packet 60015 (141s = 9.338e+04).
Enabled
Graph Name
Display Filter
Color
Style
Y Axis
Y Field
SMA Period
All Packets
Line
Bits
None
TCP Errors
tcp.analysis.flags
Line
Bits
None
Filtered pack...
ip.dst==10.0.1.20
Line
Bits
None
Filtered pack...
ip.src==10.0.2.20 &...
Line
Bits
None
Mouse
drags
zooms
Interval
1 sec
Time of day
Log scale
Reset
Save As...
Copy
Copy from
Close
Help