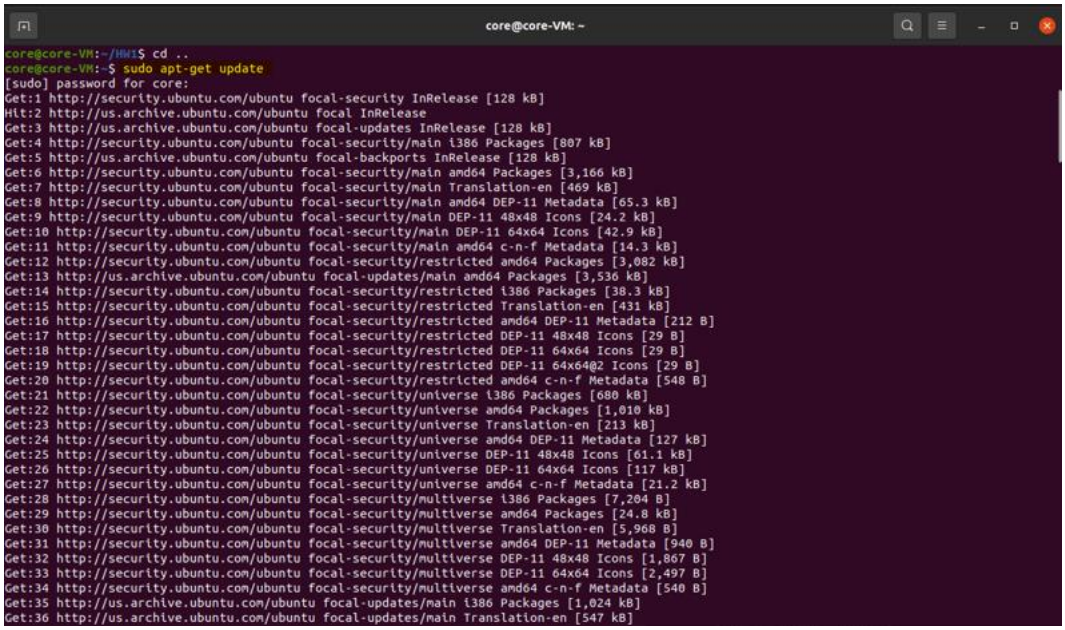
Network Security – CSCI_6541_80

Namana Y Tarikere – G21372717

Homework Assignment – 1

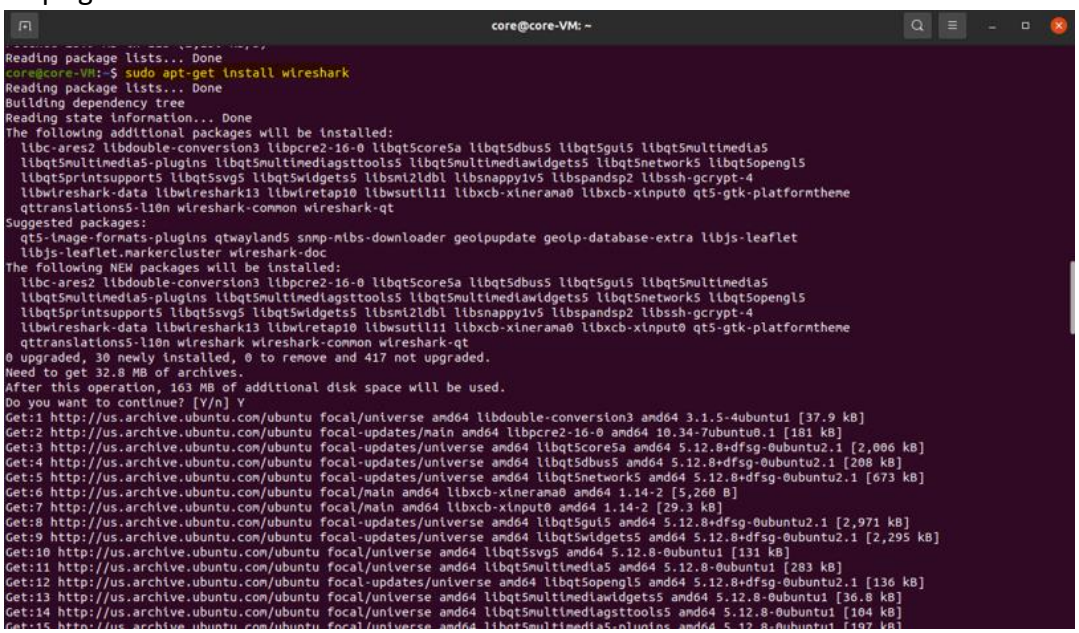
Install Wireshark

- Open a terminal window
- To install Wireshark, enter the commands:
- sudo apt-get update



```
core@core-VM: ~
core@core-VM:~/HW1$ cd ..
core@core-VM:~$ sudo apt-get update
[sudo] password for core:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [807 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3,166 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [469 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [65.3 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main DEP-11 48x48 Icons [24.2 kB]
Get:10 http://security.ubuntu.com/ubuntu focal-security/main DEP-11 64x64 Icons [42.9 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [14.3 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [3,082 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,536 kB]
Get:14 http://security.ubuntu.com/ubuntu focal-security/restricted i386 Packages [38.3 kB]
Get:15 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [431 kB]
Get:16 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 DEP-11 Metadata [212 B]
Get:17 http://security.ubuntu.com/ubuntu focal-security/restricted DEP-11 48x48 Icons [29 B]
Get:18 http://security.ubuntu.com/ubuntu focal-security/restricted DEP-11 64x64 Icons [29 B]
Get:19 http://security.ubuntu.com/ubuntu focal-security/restricted DEP-11 64x64@2 Icons [29 B]
Get:20 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 c-n-f Metadata [548 B]
Get:21 http://security.ubuntu.com/ubuntu focal-security/universe i386 Packages [680 kB]
Get:22 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [1,010 kB]
Get:23 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [213 kB]
Get:24 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [127 kB]
Get:25 http://security.ubuntu.com/ubuntu focal-security/universe DEP-11 48x48 Icons [61.1 kB]
Get:26 http://security.ubuntu.com/ubuntu focal-security/universe DEP-11 64x64 Icons [117 kB]
Get:27 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [21.2 kB]
Get:28 http://security.ubuntu.com/ubuntu focal-security/multiverse i386 Packages [7,204 B]
Get:29 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [24.8 kB]
Get:30 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [5,968 B]
Get:31 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [940 B]
Get:32 http://security.ubuntu.com/ubuntu focal-security/multiverse DEP-11 48x48 Icons [1,867 B]
Get:33 http://security.ubuntu.com/ubuntu focal-security/multiverse DEP-11 64x64 Icons [2,497 B]
Get:34 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [540 B]
Get:35 http://us.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [1,024 kB]
Get:36 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [547 kB]
```

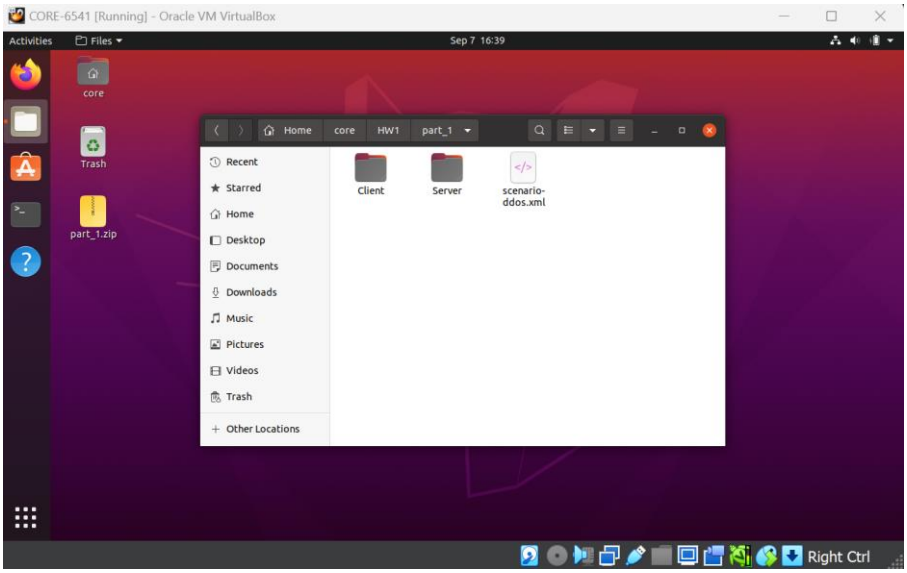
- sudo apt-get install wireshark



```
core@core-VM: ~
Reading package lists... Done
core@core-VM:~$ sudo apt-get install wireshark
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libdouble-conversion3 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5 libqt5multimedia5
  libqt5multimedia5-plugins libqt5multimediagsttools5 libqt5multimediawidgets5 libqt5network5 libqt5opengl5
  libqt5printsupport5 libqt5svg5 libqt5widgets5 libsmi2ldbl libsnappy1v5 libspandsp2 libssh-gcrypt-4
  libwireshark-data libwireshark13 libwiretap10 libwsutil11 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme
  qttranslations5-l10n wireshark-common wireshark-qt
Suggested packages:
  qt5-image-formats-plugins qtwayland5 snmp-mibs-downloader geoipupdate geoip-database-extra libjs-leaflet
  libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
  libc-ares2 libdouble-conversion3 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5 libqt5multimedia5
  libqt5multimedia5-plugins libqt5multimediagsttools5 libqt5multimediawidgets5 libqt5network5 libqt5opengl5
  libqt5printsupport5 libqt5svg5 libqt5widgets5 libsmi2ldbl libsnappy1v5 libspandsp2 libssh-gcrypt-4
  libwireshark-data libwireshark13 libwiretap10 libwsutil11 libxcb-xinerama0 libxcb-xinput0 qt5-gtk-platformtheme
  qttranslations5-l10n wireshark wireshark-common wireshark-qt
0 upgraded, 30 newly installed, 0 to remove and 417 not upgraded.
Need to get 32.8 MB of archives.
After this operation, 163 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libdouble-conversion3 amd64 3.1.5-4ubuntu1 [37.9 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpcre2-16-0 amd64 10.34-7ubuntu0.1 [181 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libqt5core5a amd64 5.12.8+dfsg-0ubuntu2.1 [2,006 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libqt5dbus5 amd64 5.12.8+dfsg-0ubuntu2.1 [208 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libqt5network5 amd64 5.12.8+dfsg-0ubuntu2.1 [673 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libxcb-xinerama0 amd64 1.14-2 [5,260 B]
Get:7 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libxcb-xinput0 amd64 1.14-2 [29.3 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libqt5gui5 amd64 5.12.8+dfsg-0ubuntu2.1 [2,971 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libqt5widgets5 amd64 5.12.8+dfsg-0ubuntu2.1 [2,295 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libqt5svg5 amd64 5.12.8-0ubuntu1 [131 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libqt5multimedia5 amd64 5.12.8-0ubuntu1 [283 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 libqt5opengl5 amd64 5.12.8+dfsg-0ubuntu2.1 [136 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libqt5multimediawidgets5 amd64 5.12.8-0ubuntu1 [36.8 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 libqt5multimediagsttools5 amd64 5.12.8-0ubuntu1 [104 kB]
```

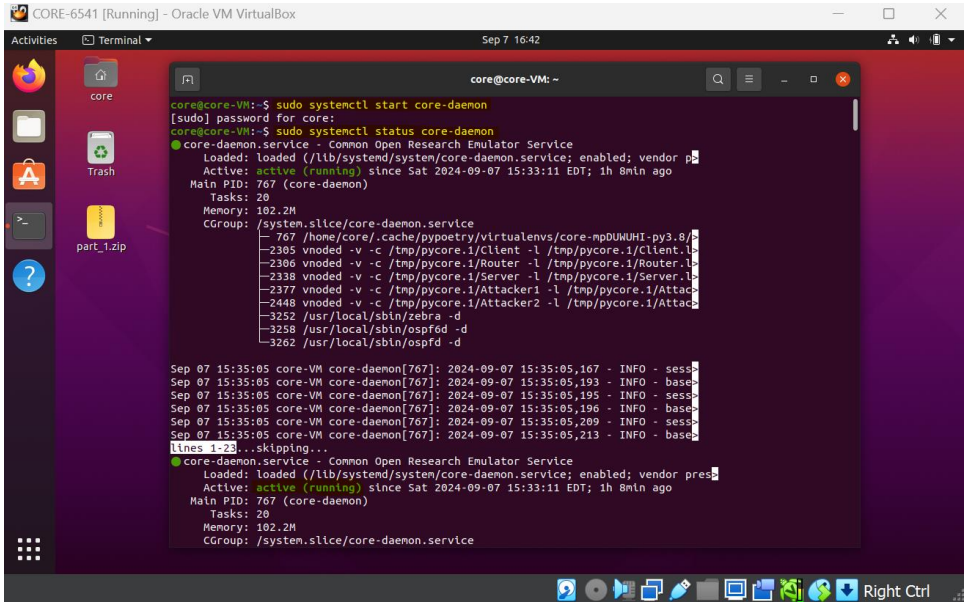
Accessing the files

- You can connect to the internet from your VM. You can access the blackboard and download the files
- Load the zipfile called part_1.zip
- Create a directory /home/core/HW1. Unzip the part_1.zip file in it.
    - You can use the command: unzip part_1.zip
- There should be one file and two directories under part_1/ directory: Client and Server.
    - Client includes “client” scripts
    - Server includes “server” scripts
    - An XML file that describes the scenario: scenario-ddos.xml

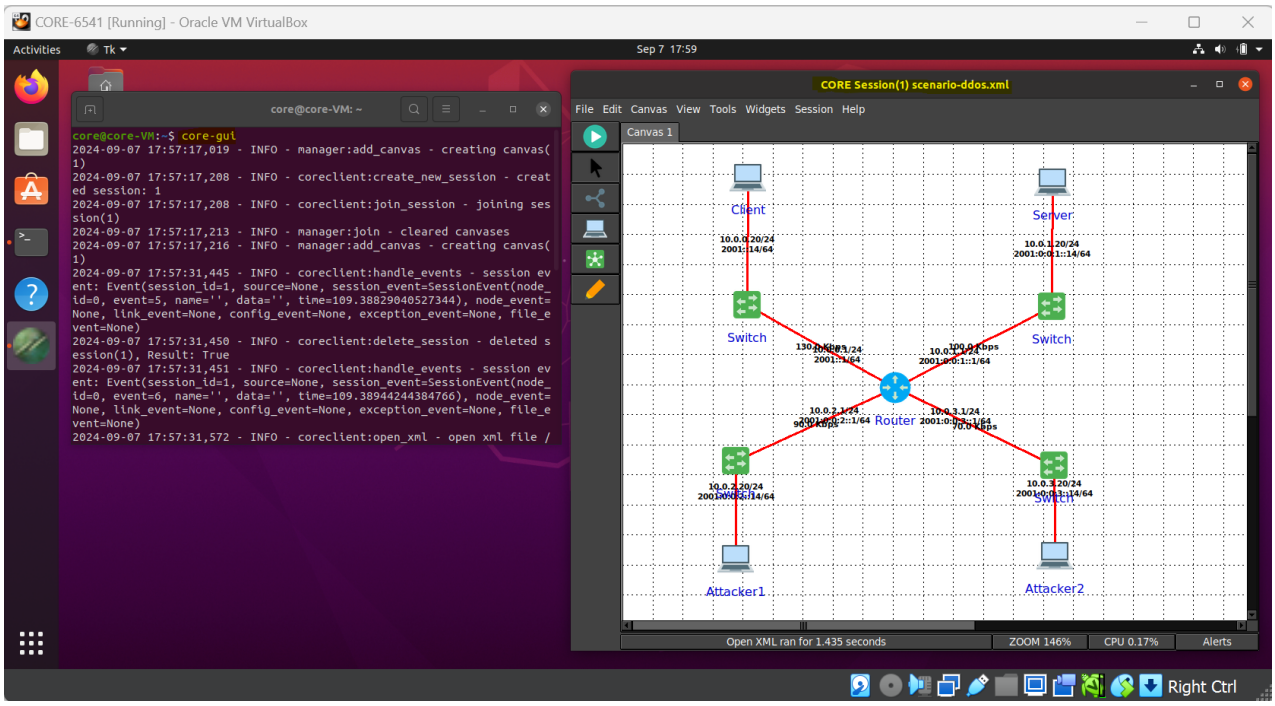


Setting up the CORE scenario

- Make sure the CORE daemon service is running:
    - To check status: systemctl status core-daemon, if it is not running do the next step
    - To actually run the service: sudo systemctl start core-daemon

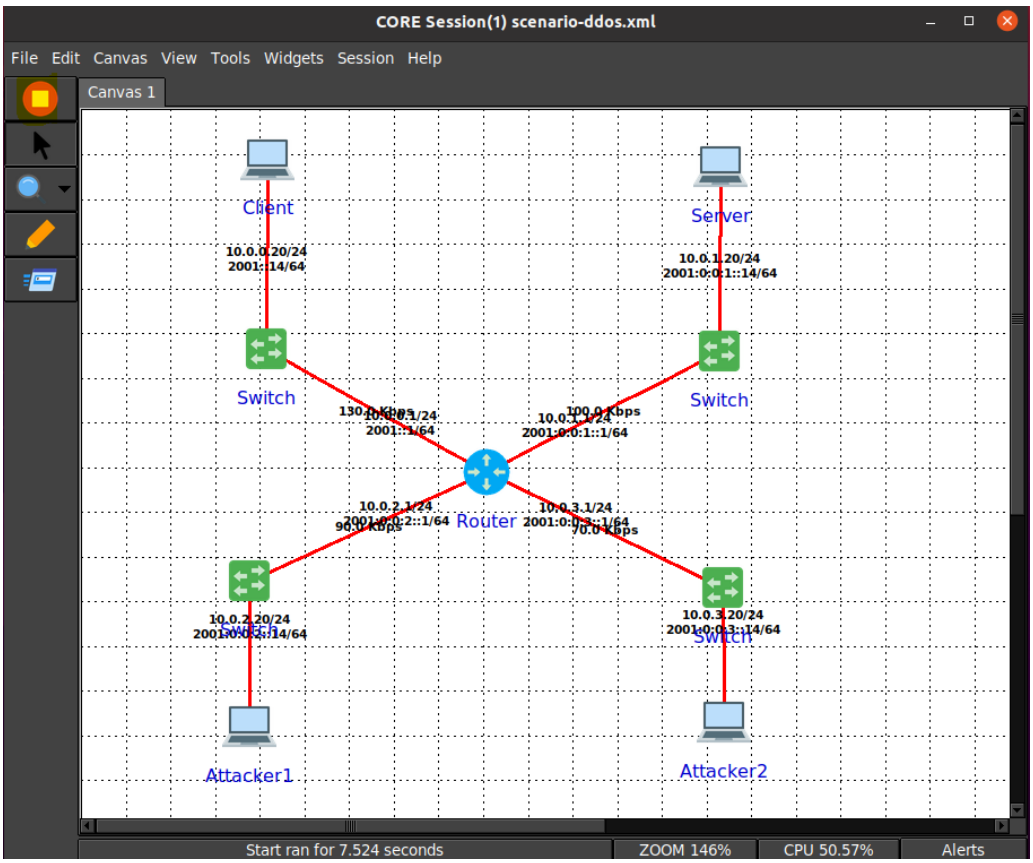


- Run the CORE GUI using the command: core-gui
- Load the file scenario-ddos.xml in the CORE GUI. You should see the following CORE scenario:



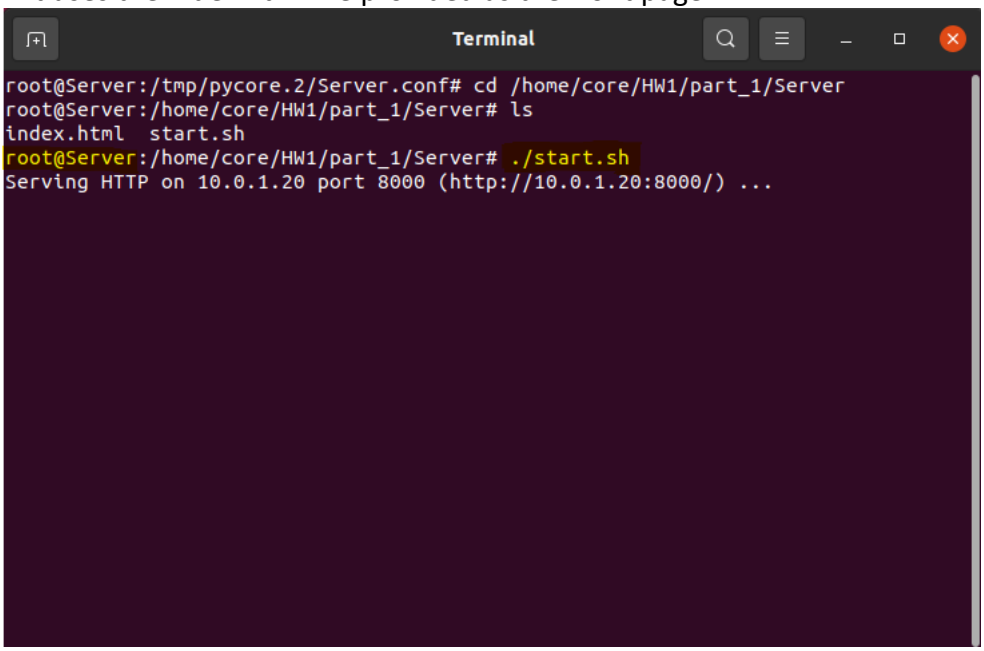
Running the CORE scenario

- Run the scenario by clicking on the green play button on the CORE GUI window



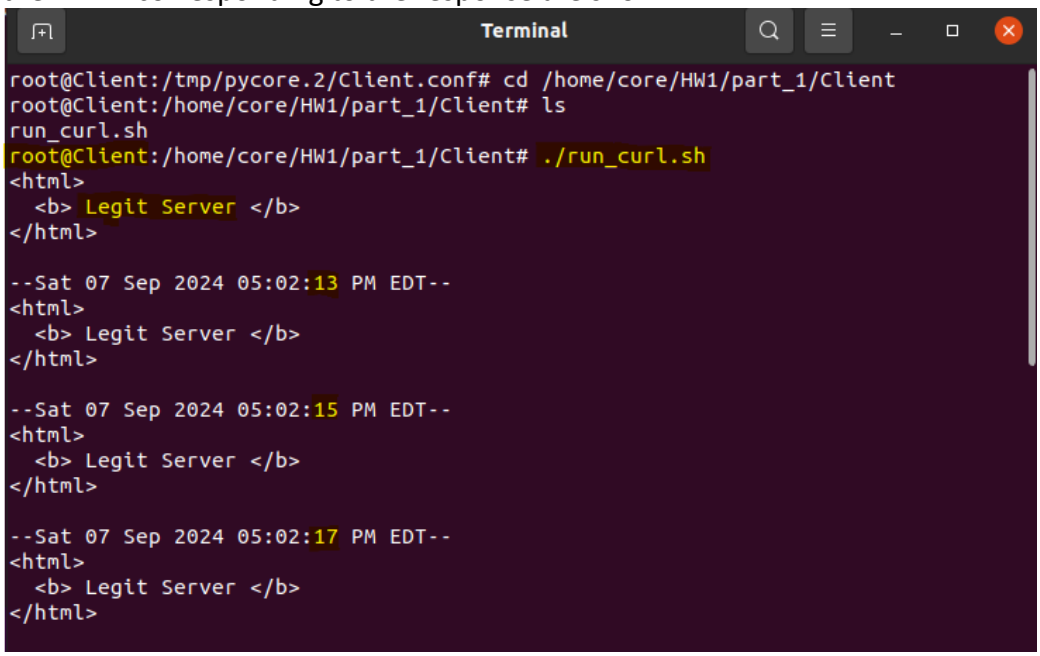
Server

- Run a terminal on the Server node and change directory to /home/core/HW1/part_1/Server
- Run start.sh script. This runs a python http.server module on the Server node on port 8000. It uses the index.html file provided as the front page.



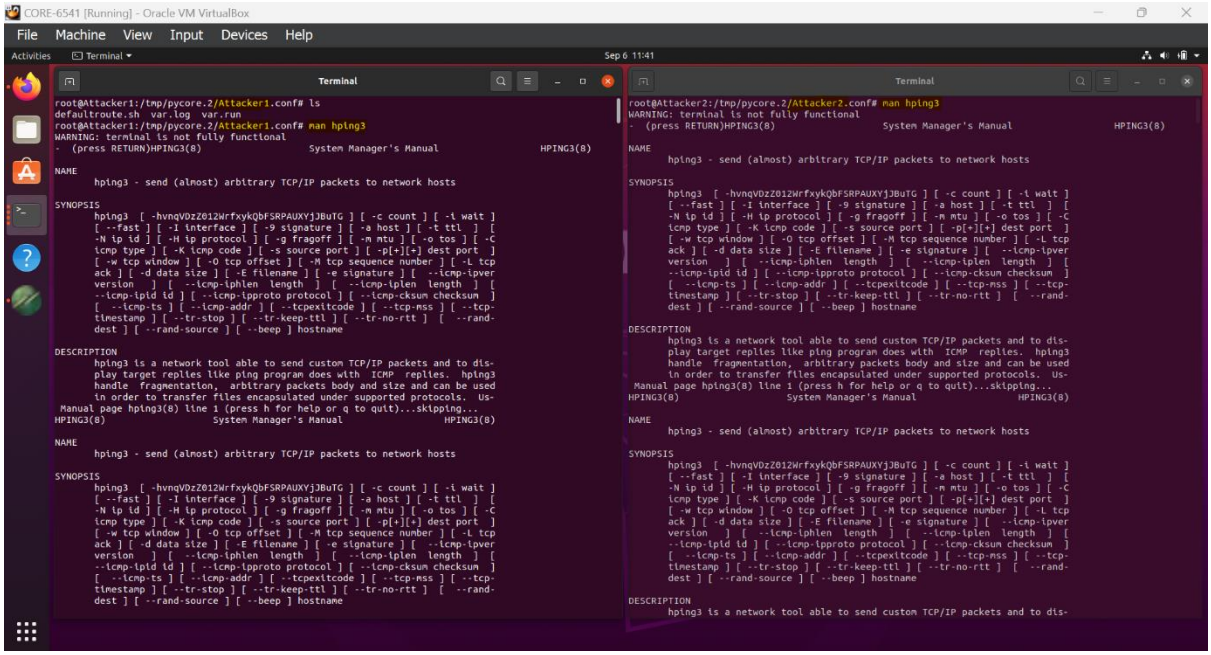
Client

- Run a terminal on the Client node and change directory to /home/core/HW1/part_1/Client
- Run the script *run_curl.sh*
  - This script mimics a web brower
  - The script uses curl command to automatically access the main page on the index.html page hosted by the Server every 2 seconds. The request time and the HTML corresponding to the response are shown.



Attacker

- Run a terminal on one of the Attacker nodes
- Run the command: man hping3. This will display the manual for hping3
- Hit control-C when done reading the manual.



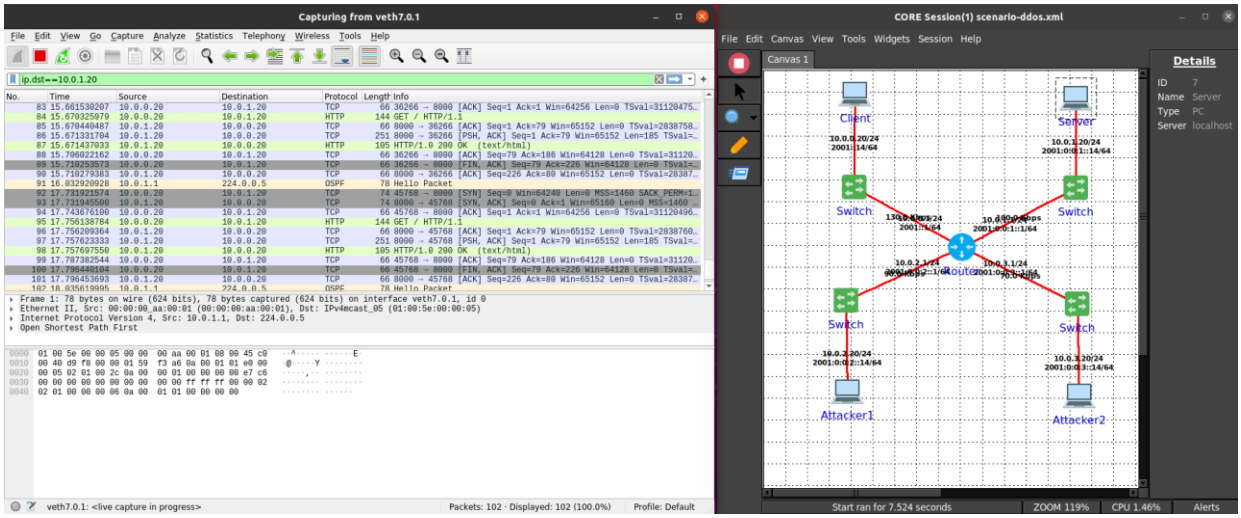
Your task is to investigate and characterize the impact of the DoS attack from Attackers on the webserver using Wireshark, the protocol analyzer.

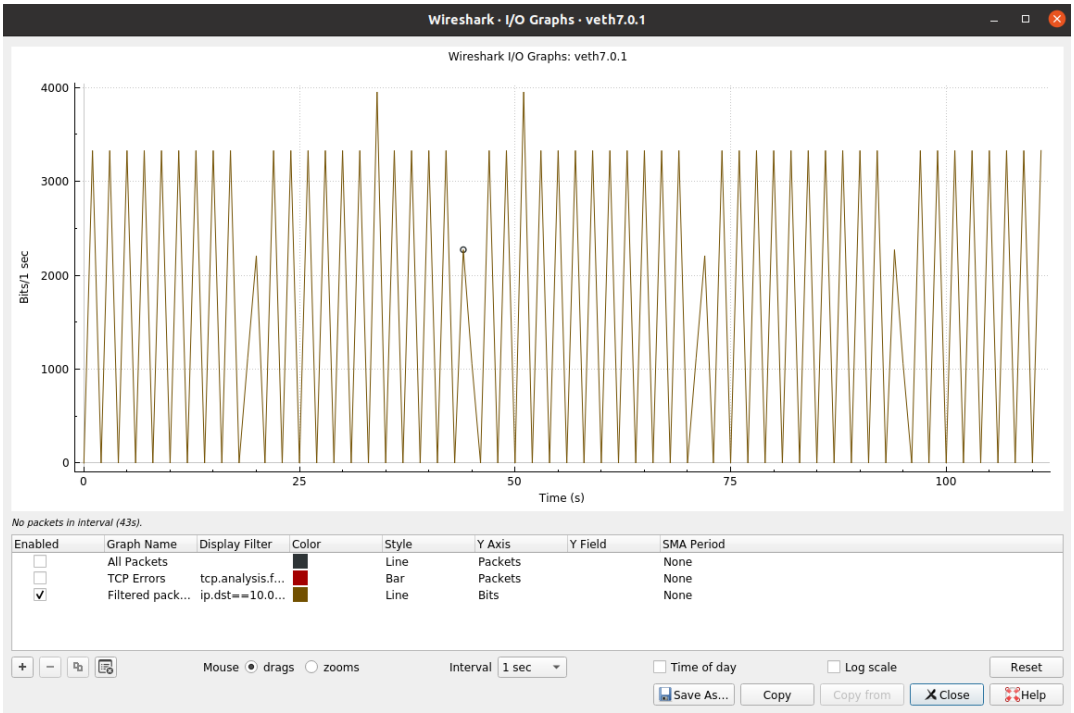
HW Steps:

1) **(10pts) Using the network as is: First, run Wireshark (from the Ubuntu VM) on the link connecting the Server. Add a filter in Wireshark: ip.dst == 10.0.1.20. This will limit the packets shown to the ones destined to the server (so we are only counting bytes to Server = Attack traffic + Client traffic). Next, run the client script *run_curl.sh*.**

   The following screenshots represent the Core-GUI session of the scenario-ddos.xml, the Wireshark panel showing the source and destination IPs of the Server and Client communicating via TCP and HTTP protocols. The IO graph activities of the node connecting to the Server (Node Number 7) is captured on the veth7.0.1 Wireshark interface.

   Before the attack was launched by Attacker 1, the Wireshark reported a network traffic of about **3250 bits/sec** and the client is receiving server's website information.



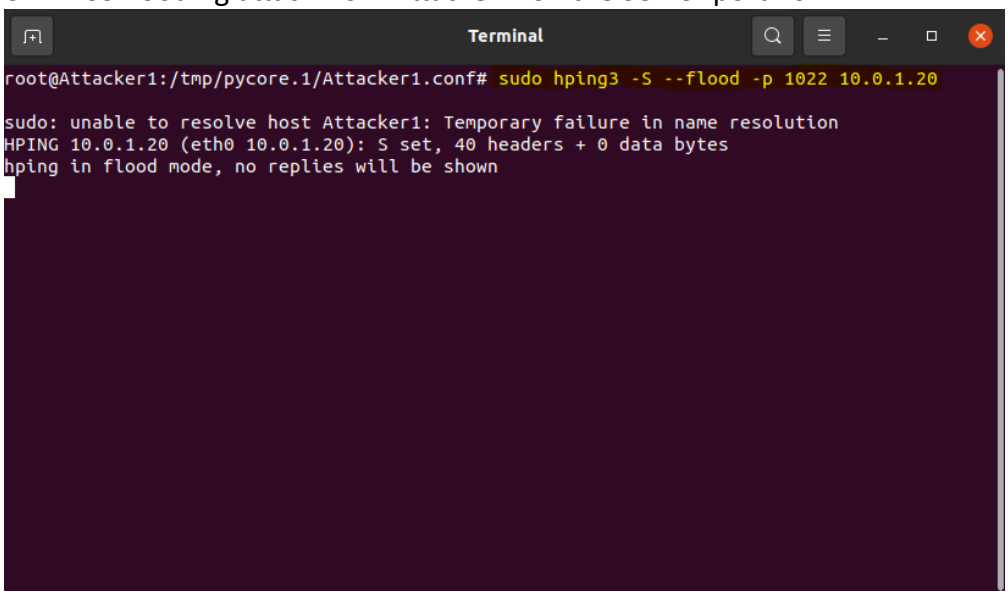


a. (0.5pt) Do a SYN flooding DoS attack on the server using hping3 (no address spoofing) from Attacker1 only. Specify destination port 1022.
What command did you use? Show a screenshot of the command

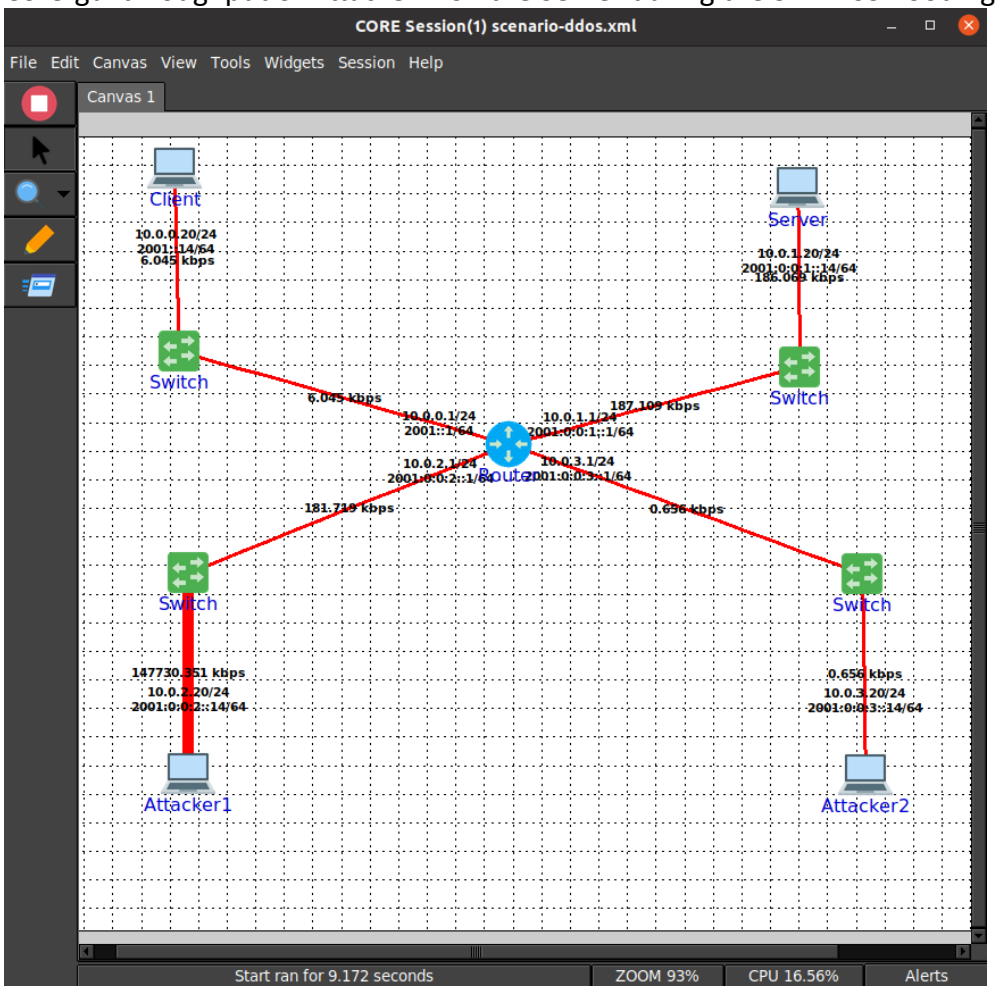
Solution: I used the **sudo hping3 -S –flood -p 1022 10.0.1.20** command to do a SYN flooding DoS attack on the server as shown below. Here:

- sudo is used for administrative privileges
- hping3 is the tool used to send/ping SYN packets to the server
- -S represents Syn packets
- --flood represents flooding of requests
- -p 1022 represents target port and port number of the destination system
- 10.0.1.20 is the target IP address (Server) which will be attacked

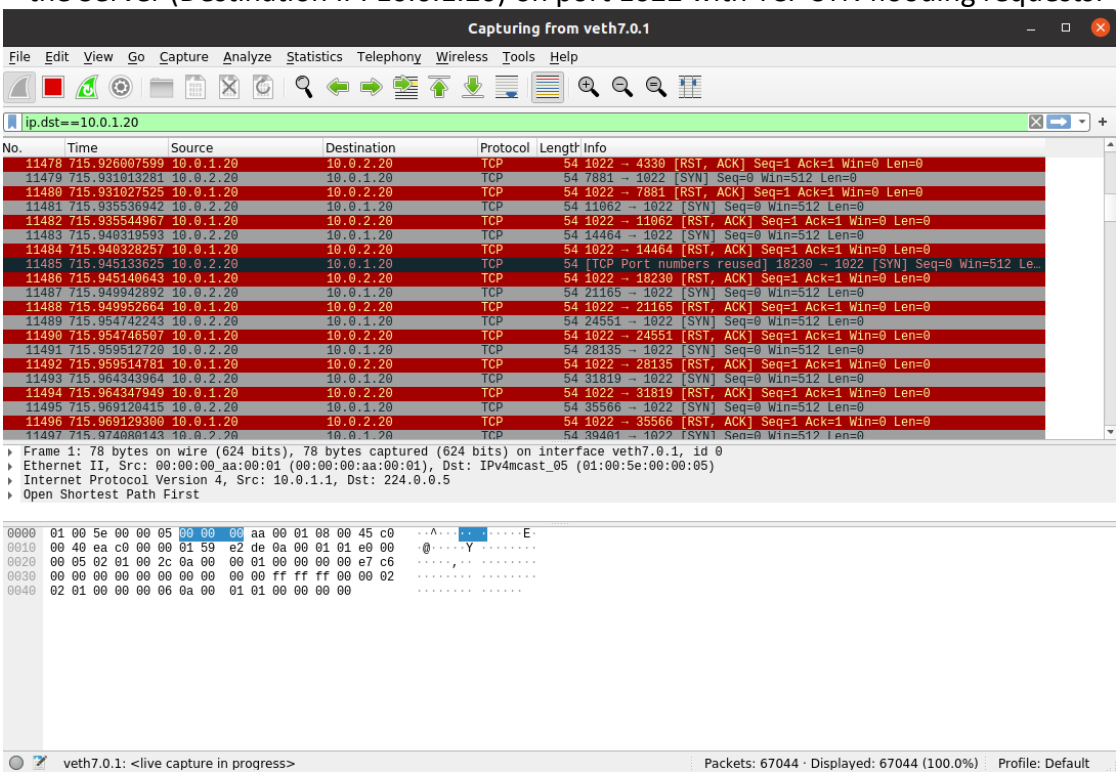
SYN DoS flooding attack from Attacker 1 on the Server port 1022:



Core-gui throughput of Attacker 1 on the Server during the SYN DoS flooding attack:

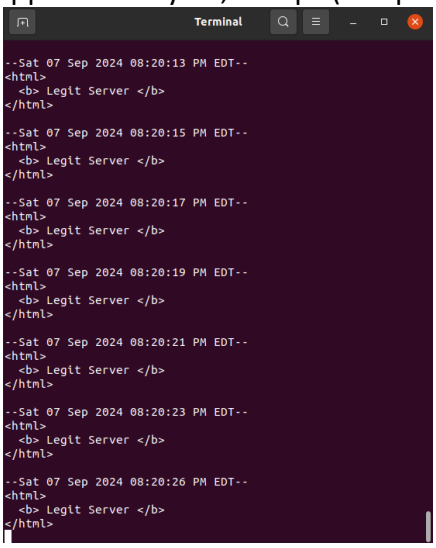


Wireshark console capturing the DoS attack from Attacker 1 (Source IP: 10.0.2.20) on the Server (Destination IP: 10.0.1.20) on port 1022 with TCP SYN flooding requests:



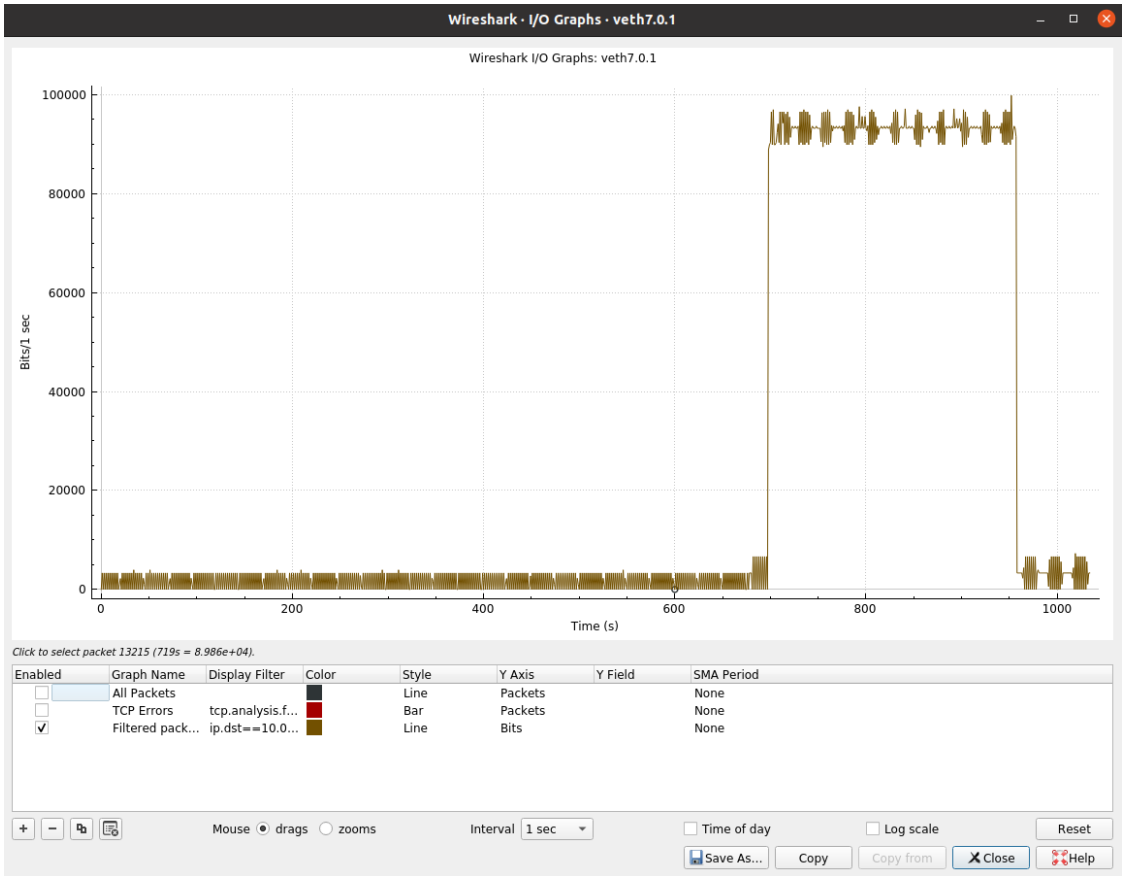
b. (1pt) Is the attack effective? Why or why not?
Show a screenshot of the client curl script indicating if the client is affected. If attack is effective, client should stop getting the HTML page from server every 2 seconds. If attack is effective, show a gap in time when the client was not receiving (start attack → client will stop receiving → stop attack after a minute → client will start receiving again → show the gap in time where client stopped receiving).

This Syn flooding DoS attack from Attacker1 is **not effective** as the client is still receiving the requests from the server every 2 seconds without any changes. This is because the link that connects the router to the Server can handle more traffic (100,000bps) than the throughput generated by Attacker 1 which is approximately 90,000bps (bits per second). Client response:



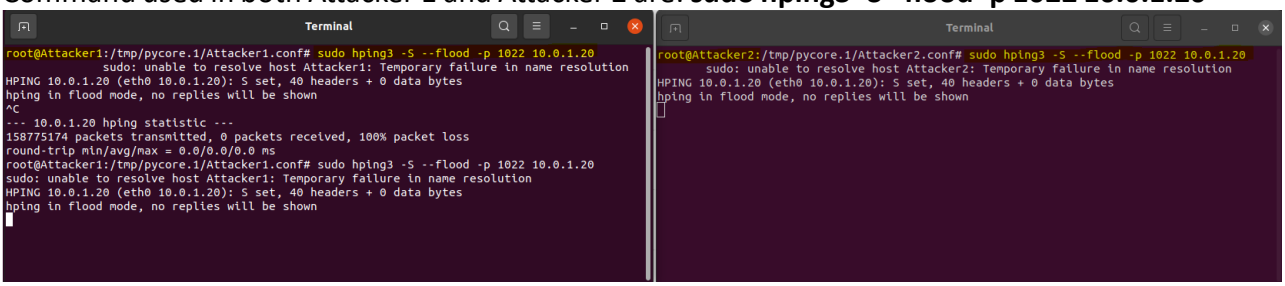
c. (1pt) What is the magnitude of the attack (in bps)? Use Wireshark IO Graphs to answer this question. Show screenshots to support your answer. Use "bits" in the y-axis as opposed to "packets".

The peak magnitude or maximum network traffic of the link connecting the Server is 100,000bps. But, during the attack, the magnitude or the network traffic of Attacker 1 on the server was approximately 90,000bps which was not strong enough to impact the client system. After the DoS flooding attack, the traffic is back to 3250 bps. Below is the screenshot:

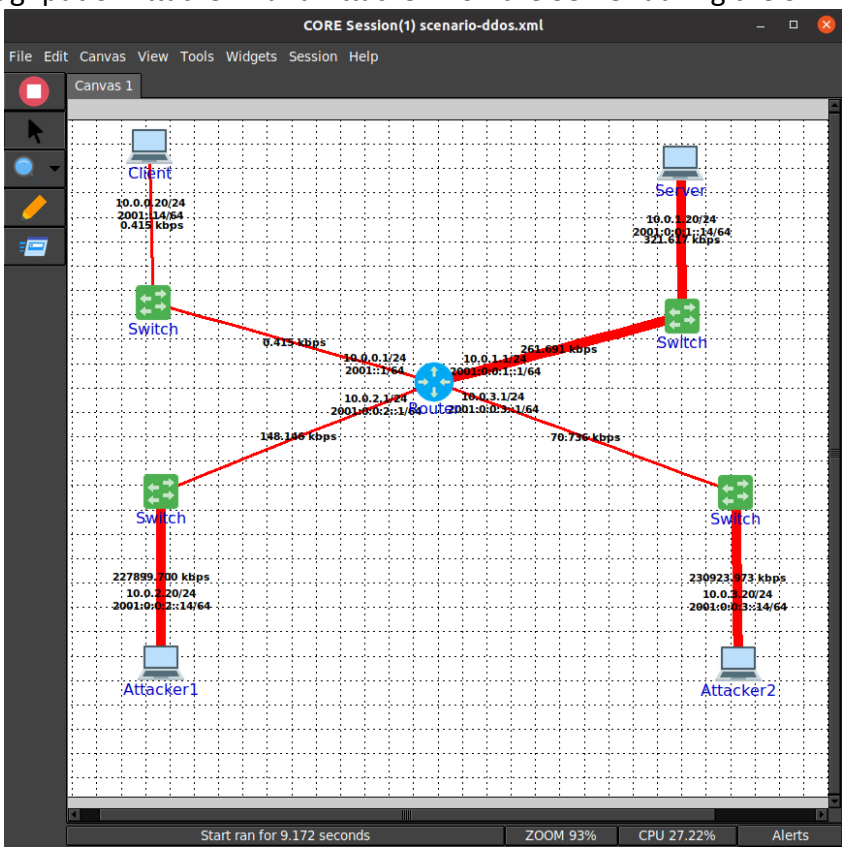


d. (2.5pts) Repeat a, b & c, this time run the attack from both Attacker1 and Attacker2.

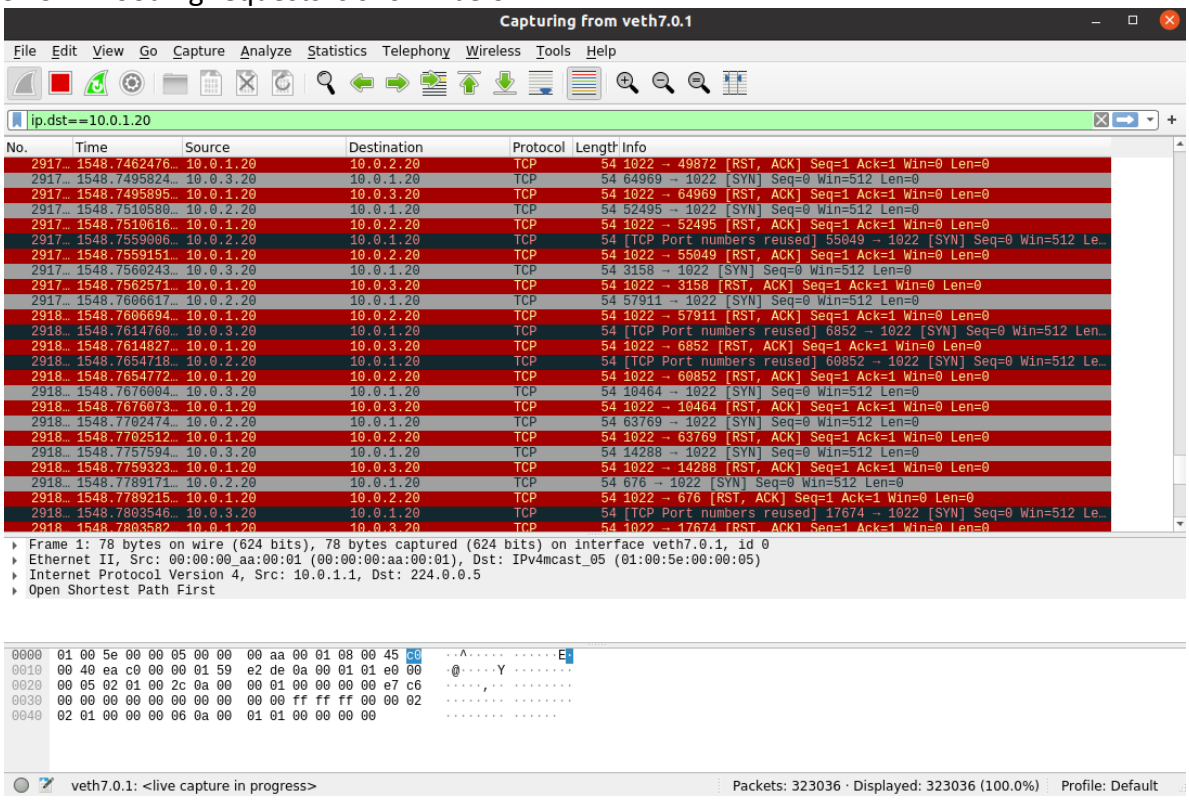
SYN DoS flooding attack from Attacker 1 and Attacker 2 on the Server port 1022 as shown:
Command used in both Attacker 1 and Attacker 2 are: **sudo hping3 -S –flood -p 1022 10.0.1.20**



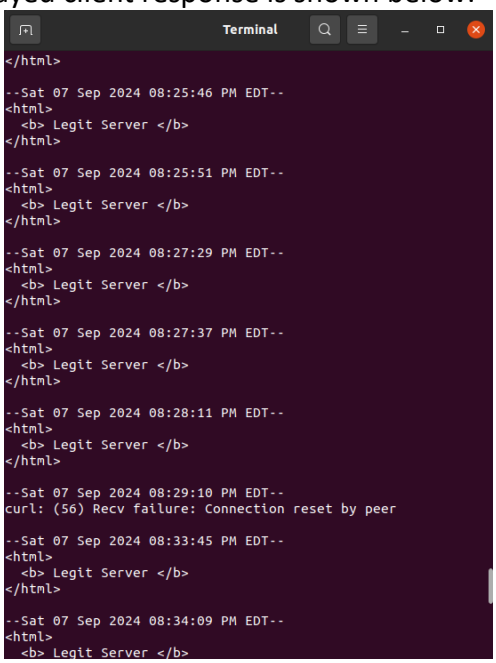
Core-gui throughput of Attacker 1 and Attacker 2 on the Server during the SYN DoS attack:



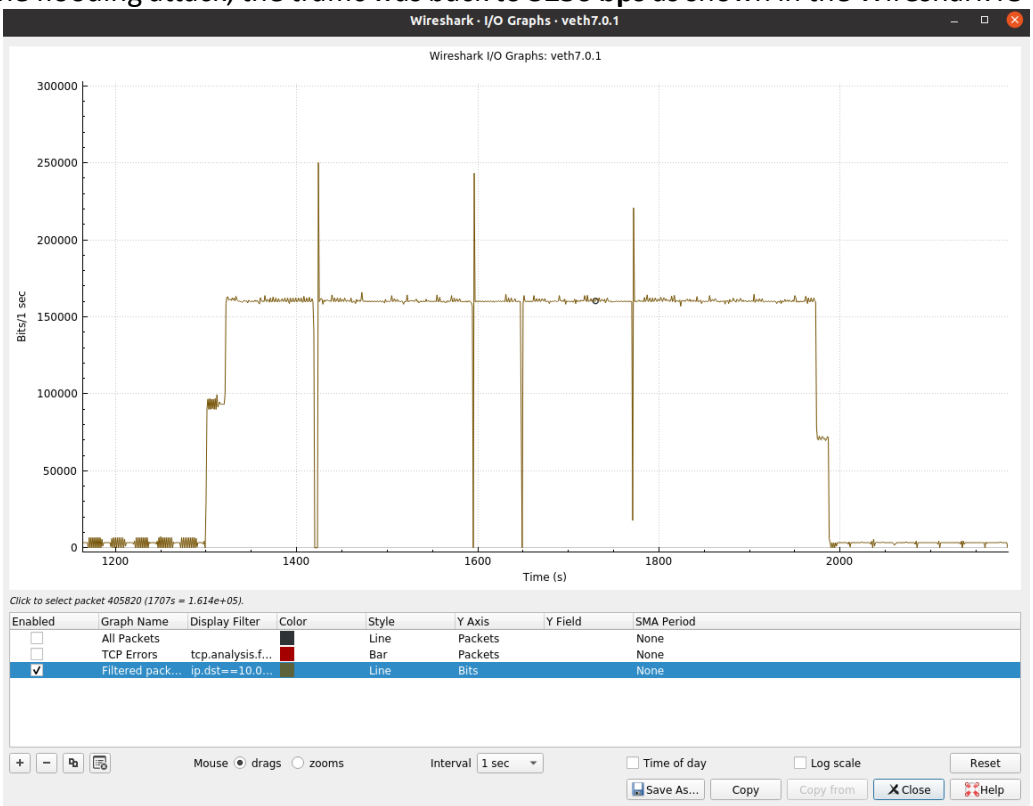
Wireshark console capturing the DoS attack from Attacker 1 (Source IP: 10.0.2.20) and Attacker 2 (Source IP: 10.0.3.20) on the Server (Destination IP: 10.0.1.20) on port 1022 with TCP SYN flooding requests is shown below:



The attack from both the attackers was **effective** as the response time of the Client system was delayed by 1 to 4 minutes showing successful Denial of Service attack. This is because the link that connects the router to the Server could only handle a traffic 100,000bps, but the throughput generated by both Attacker 1 and 2 was approximately 160,000bps which **overloaded the Server.** Delayed client response is shown below:

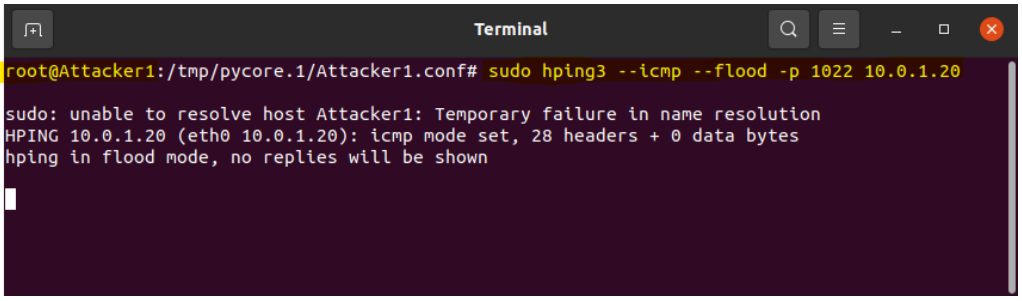


Since I started the Attacker 1 first, the network traffic from Attacker 1 is shown in the graph to be approximately **90,000bps** initially and later increased to **160,000bps** after I started Attacker 2. The network traffic from both attackers was strong enough to impact the client system. After the flooding attack, the traffic was back to **3250 bps** as shown in the Wireshark IO Graph:

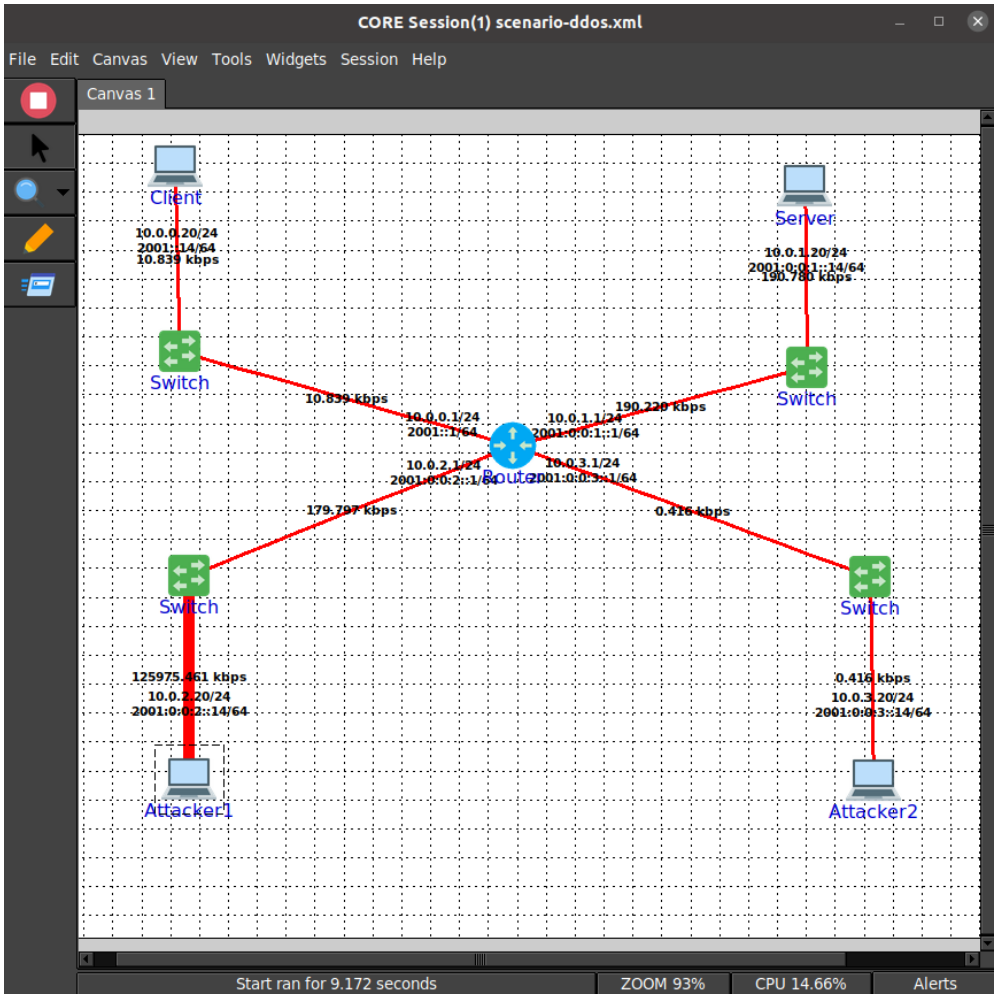


e. (2.5pts) Repeat a, b & c, this time run the attack from Attacker1 only and use an ICMP flooding attack.

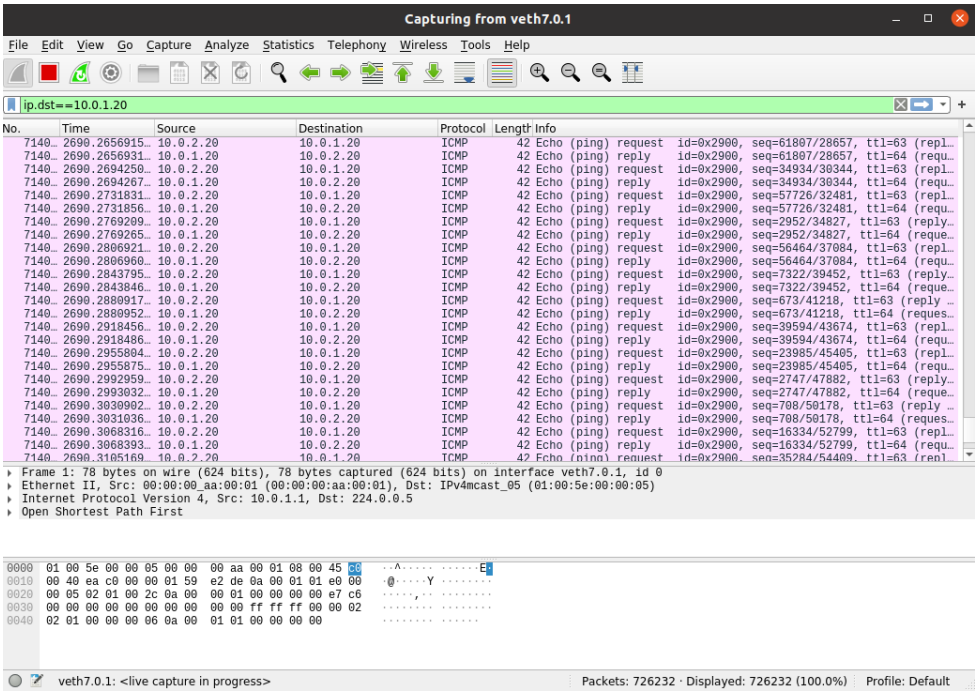
ICMP attack with flooding requests from Attacker 1 on the Server IP 10.0.1.20 using the command **sudo hping3 –icmp –flood -p 1022 10.0.1.20** as shown below:



Core-Gui throughput of Attacker 1 on the Server during the ICMP flooding attack:

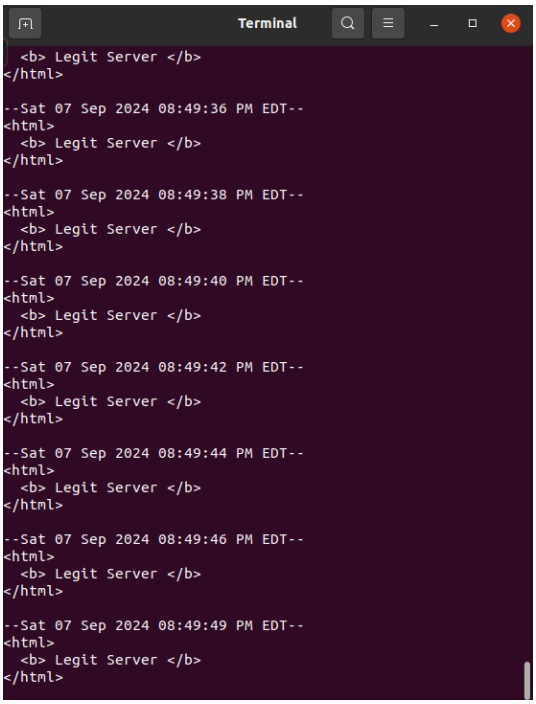


Wireshark console capturing the **ICMPS attack** from Attacker 1 (Source IP: 10.0.2.20) on the Server (Destination IP: 10.0.1.20) on port 1022 with flooding requests is shown below:



During the ICMP attack, the network traffic from Attacker 1 is approximately **90,000bps** which was not strong enough to impact the client system. After the flooding attack, the traffic was back to **3250 bps** as shown in the Wireshark IO Graph below:

The attack was **not effective** as the Client is still receiving server requests every 2 seconds. This is because the link that connects the router to the Server can handle more traffic (100,000bps) but the throughput generated by Attacker 1 was approximately 90,000bps. Client response is shown below:



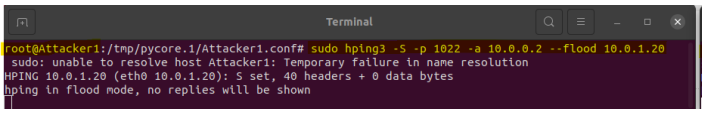
f. (2.5pts) Repeat d but change the command used at so

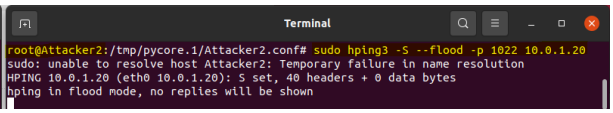
i. Attack packets from Attacker 1 are reflected off of the client (address spoofing is allowed).

DoS attack with flooding requests from Attacker 1 and Attacker 2 on the Server IP 10.0.1.20 where Attacker 1 is spoofed as Client is shown below along with the commands used:

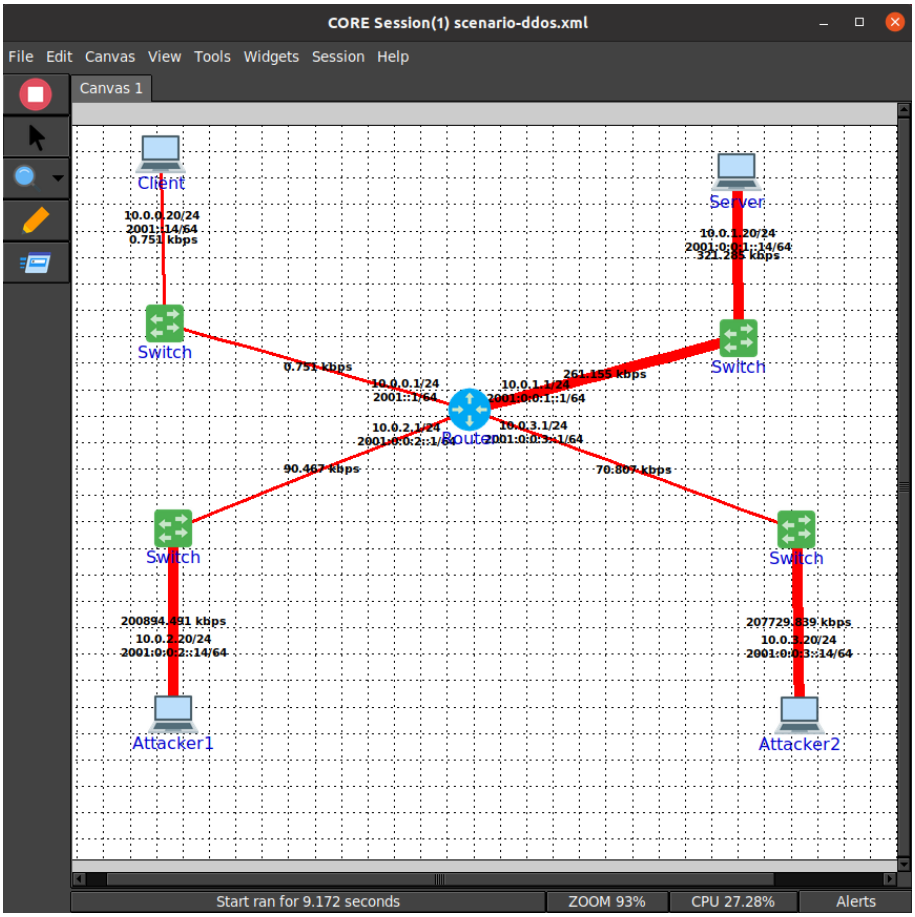
Attacker 1: sudo hping3 –S -p 1022 -a 10.0.0.2 –flood 10.0.1.20

Attacker 2: sudo hping3 -S –flood -p 1022 10.0.1.20

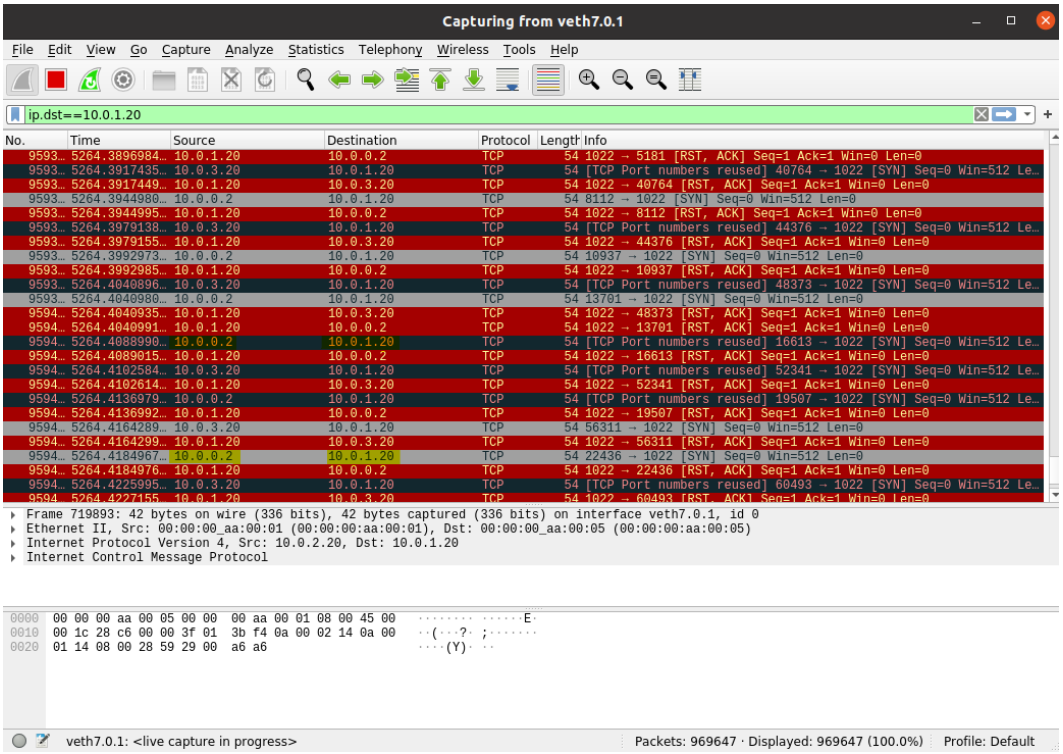




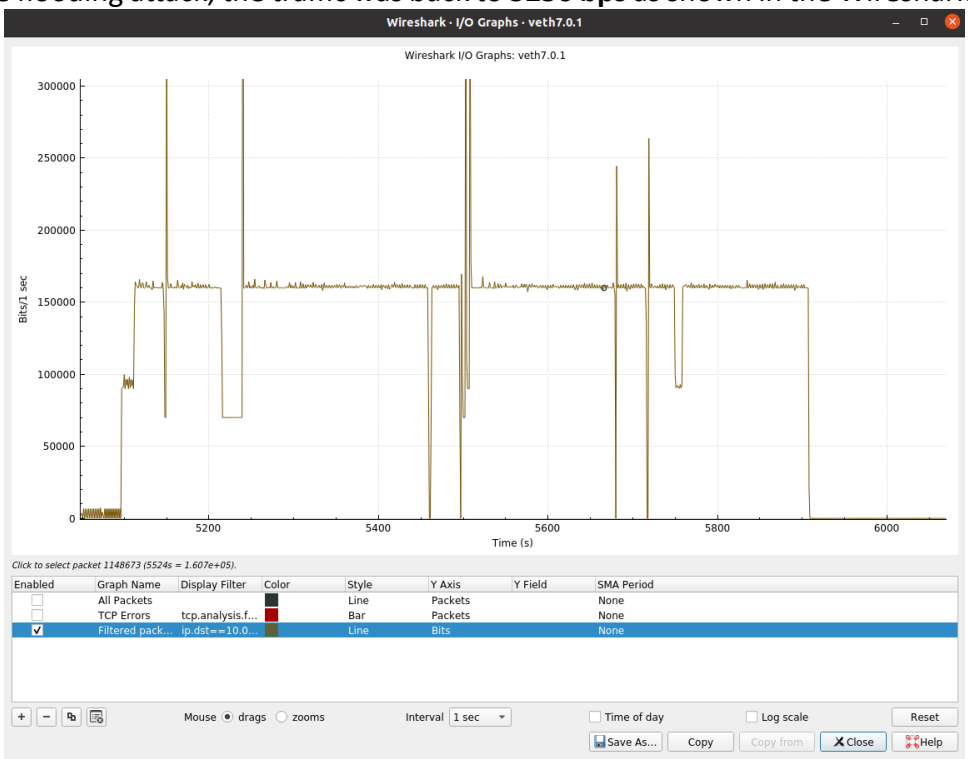
Core-GUI throughput of Attacker 1 and Attacker 2 on the Server during the SYN DOS flooding attack is shown below:



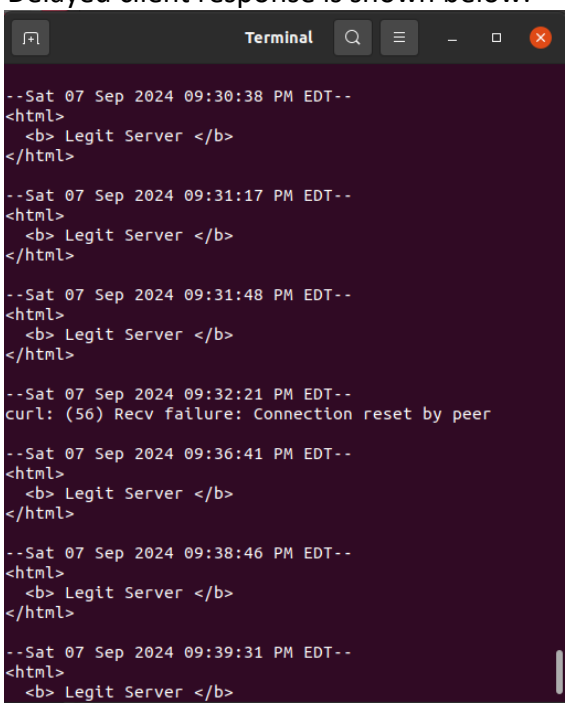
Wireshark console capturing the **DoS flooding attack** from **Attacker 1 spoofed as Client** (Source IP: 10.0.0.20) and Attacker 2 (Source IP: 10.0.3.20) on the Server (Destination IP: 10.0.1.20) on port 1022 with flooding requests is shown below:



Since I started the Attacker 1 first, the network traffic from Attacker 1 is shown in the graph to be approximately **90,000bps** initially and later increased to **160,000bps** after I started Attacker 2. The network traffic from both attackers was strong enough to impact the client system. After the flooding attack, the traffic was back to **3250 bps** as shown in the Wireshark IO Graph:



The attack from both the attackers was **effective** as the response time of the Client system was delayed by 1 to 4 minutes showing successful Denial of Service attack. This is because the link that connects the router to the Server could only handle a traffic 100,000bps, but the throughput generated by both Attacker 1 and 2 was approximately 160,000bps which **overloaded the Server.** Delayed client response is shown below:

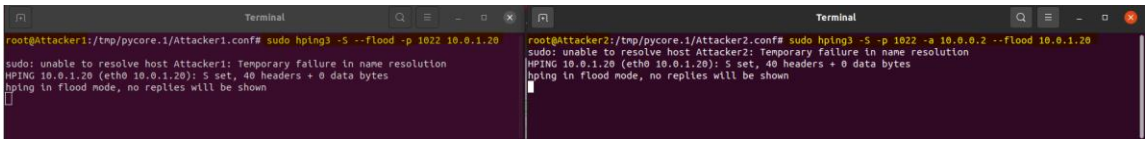


ii. **Attack packets from Attacker 2 look like they are coming from the client (address spoofing is allowed).**

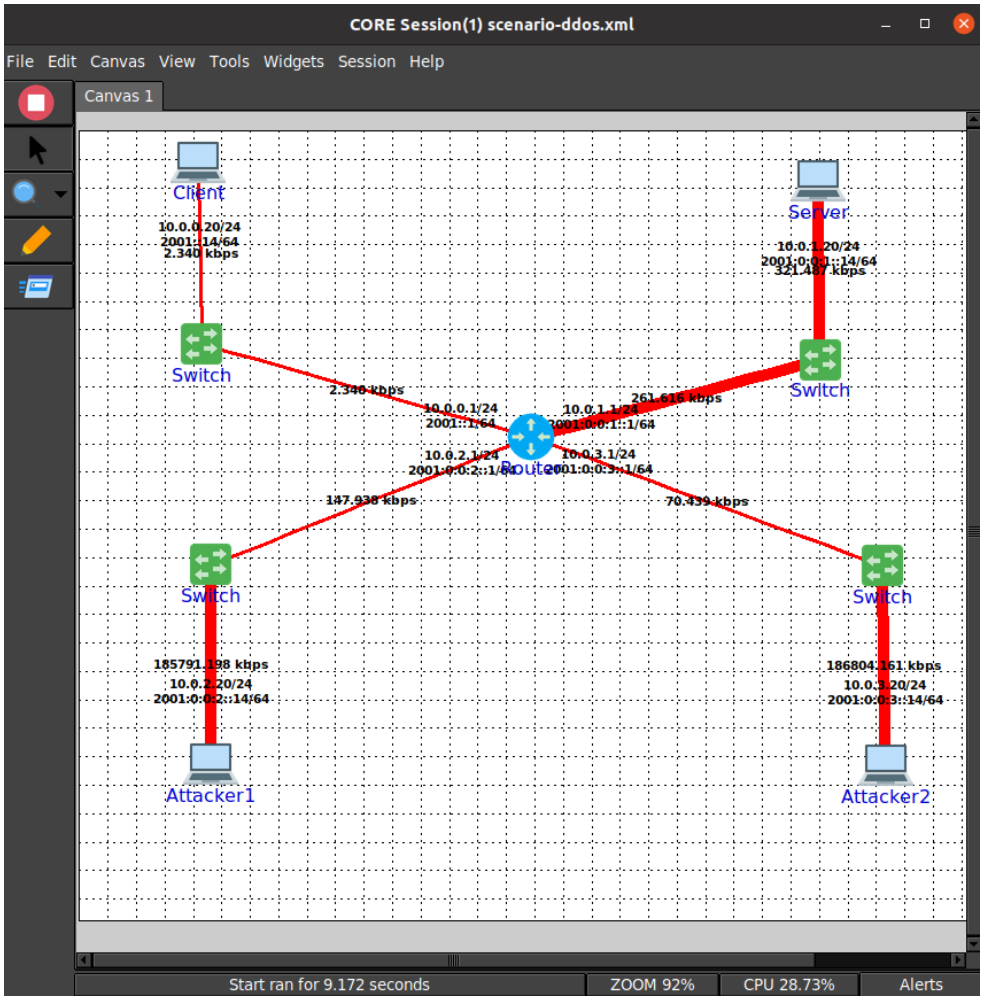
DoS attack with flooding requests from Attacker 1 and Attacker 2 on the Server IP 10.0.1.20 where Attacker 2 is spoofed as Client is shown below along with the commands used:

Attacker 1: sudo hping3 -S –flood -p 1022 10.0.1.20

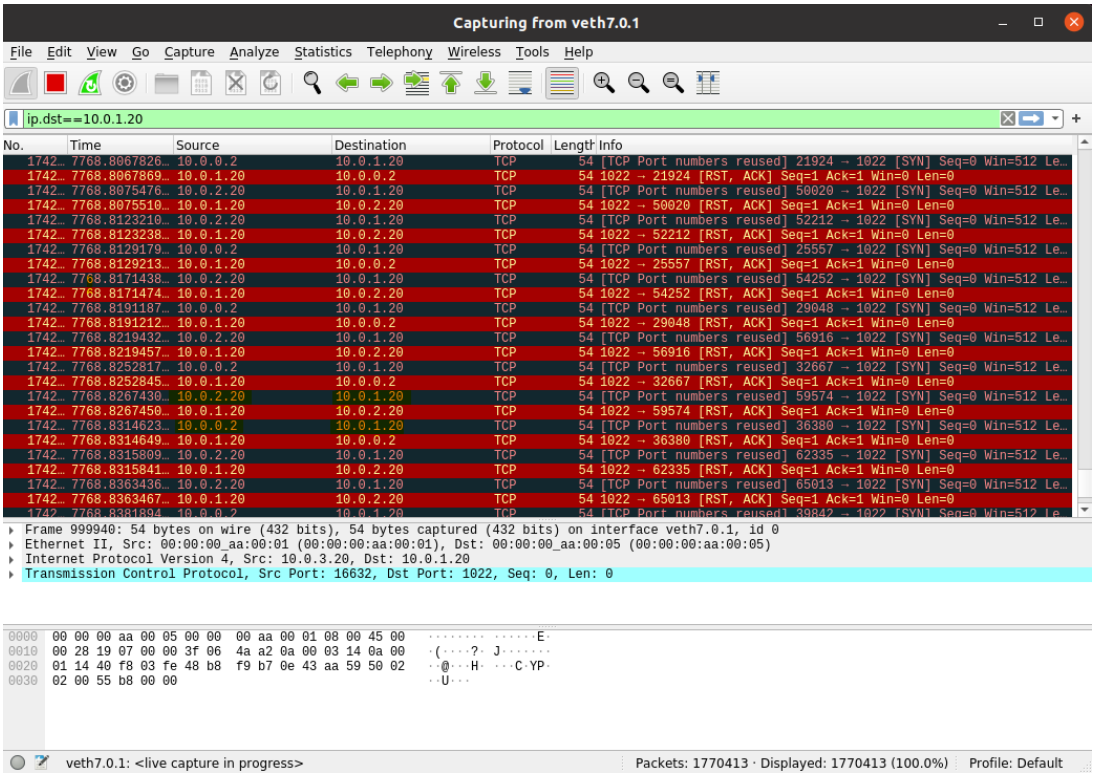
Attacker 2: sudo hping3 –S -p 1022 -a 10.0.0.2 –flood 10.0.1.20



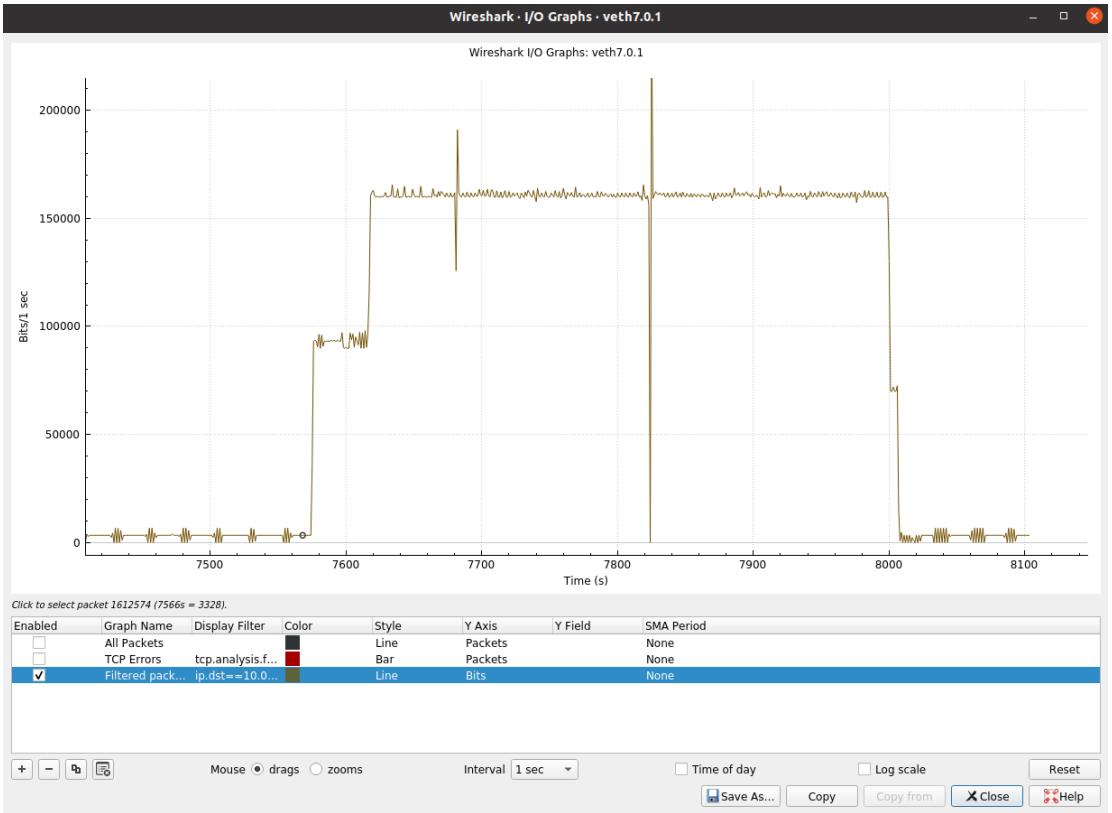
Core-GUI throughput of Attacker 1 and Attacker 2 on the Server during the SYN DOS flooding attack is shown below:



Wireshark console capturing the **DoS flooding attack** from Attacker 1 (Source IP: 10.0.2.20) and **Attacker 2 spoofed as Client** (Source IP: 10.0.0.20) on the Server (Destination IP: 10.0.1.20) on port 1022 with flooding requests is shown below:



Since I started the Attacker 1 first, the network traffic from Attacker 1 is shown in the graph to be approximately **90,000bps** initially and later increased to **160,000bps** after I started Attacker 2. The network traffic from both attackers was strong enough to impact the client system. After the flooding attack, the traffic was back to **3250 bps** as shown in the Wireshark IO Graph:



The attack from both the attackers was **effective** as the response time of the Client system was delayed by 1 to 3 minutes showing successful Denial of Service attack. This is because the link that connects the router to the Server could only handle a traffic 100,000bps, but the throughput generated by both Attacker 1 and 2 was approximately 160,000bps which **overloaded the Server.** Delayed client response is shown below: