ECE 8860: Distributed Denial of Service Attacks

Spring 2023

Laboratory 2

Name: Sandali Sanjay Nemmaniwar

Email: sandaln@g.clemson.edu

Major: Computer Science

Abstract

This report discusses a study that explains about network background traffic, attack generation, and attack tools. The first section, part A, explains how to replay captured network traffic. Also, Part A aids in observing and comprehending the difference between operational network background traffic and simulated background traffic. Part B, on the other hand, explains the differences between the four different types of attacks and how they are launched. The final section, C, goes into further detail on the TCP handshake process and describes how a Syn flooding attack can affect a server or network.

Introduction:

The amount of background network traffic is crucial for DDoS attack detection research. The time of day, the permitted network services (such as email, cloud services, VoIP, and downloading/uploading large files), and the typical user count all affect how much traffic is created in a network. It serves as a basis for truth during a detection. Anomaly-based detection approaches employ the unexpected departure in the observable feature (such as the number of packets received in a second) of the background traffic for attack detection while signature-based detection approaches look for recognized patterns in background traffic. Thus, it is essential that these experiments use active network background traffic. Yet, access to a live network for testing is not always feasible. Researchers get around this problem by simulating background network traffic in their experiments, creating traffic in a computer cluster, or replaying packet traces gathered from an operating network.

Different DDoS attacks have different performances. In the lab sessions we are performing 4 types of attacks and measure their performance.

Syn flood is an attack that makes use of a vulnerability in the TCP protocol to deplete the resources of the target server. A three-way handshake is performed by the client and server to begin a TCP connection (See Figure 2.16). The handshake is initiated by the client sending the server a TC synchronization message (SYN) message. With a SYN-ACK message, the server acknowledges the request. The client sends an ACK message to end the handshake.

By starting the handshake with merely Syn messages and not sending ACK messages afterward, an attacker can take advantage of this process. A list of unfinished TCP connections will be generated at the server as a result. The server will use RAM to store the details of each TCP session. The memory will retain this information until they time out. A server will run out of memory, refuse any new incoming TCP connection requests, and result in a denial of service if an attacker can start enough TCP handshakes.

Network Traffic Generation

Methodology:

1. Connect to the security lab machines.
2. Capture campus traffic.
3. Replay the captured traffic.
4. Execute a python script to compare the real background traffic versus simulated background traffic.

Analysis:

1. Captured the campus network traffic data using Wireshark.
2. Compared the real background traffic and the simulated background traffic by plotting the time-series graph for both.

Questions:

1. Plot the background traffic datasets generated/used in the three scenarios where x-axis shows time, and the y-axis shows the number of packets received. Compare the figures and discuss their differences.

Ans: There are 3 scenarios for different background traffic datasets:

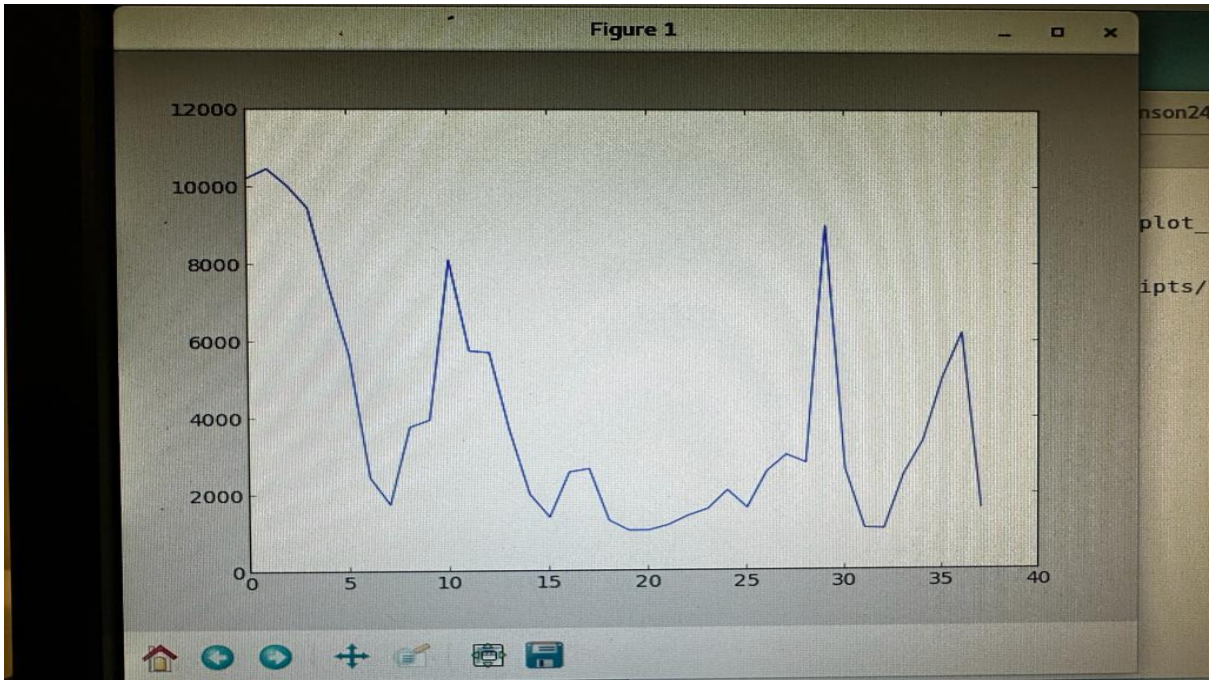


Figure 1: scenario 1

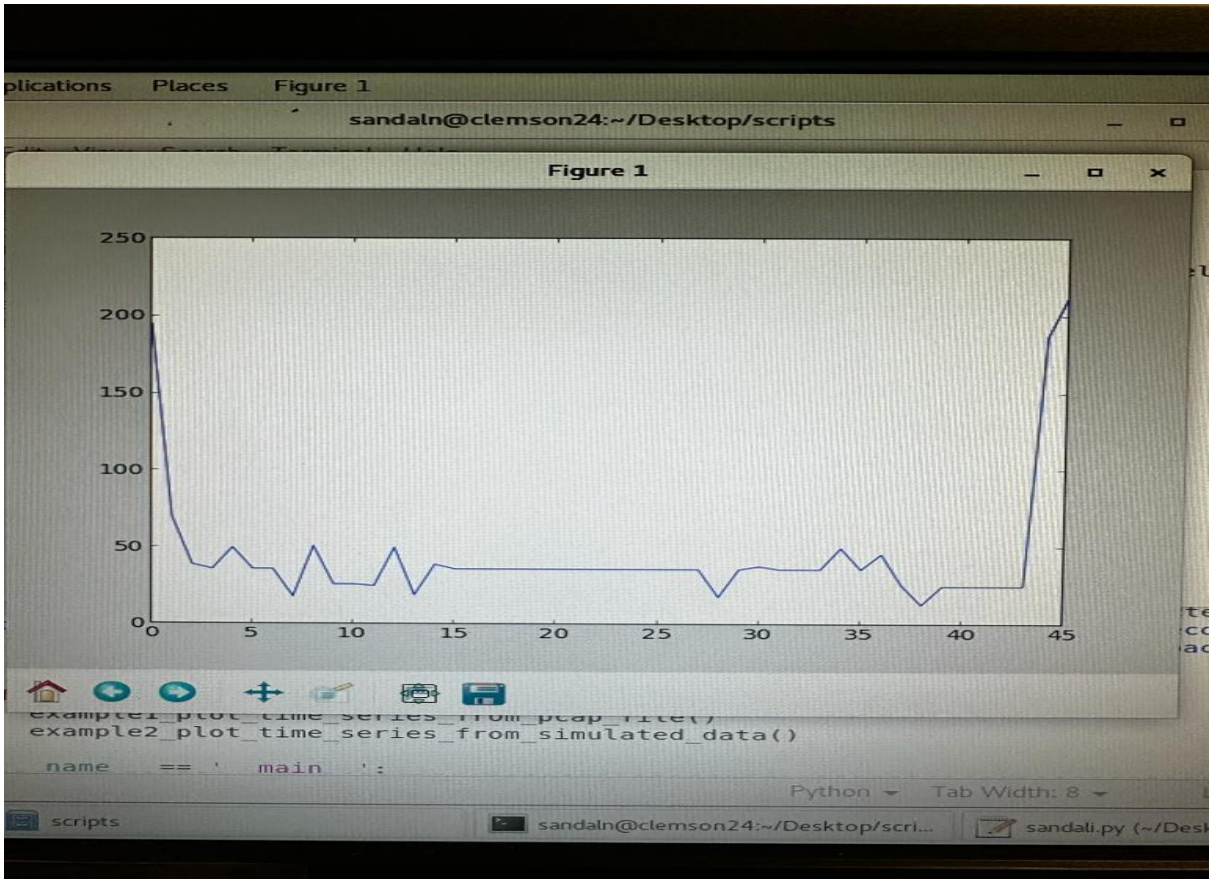


Figure 2: scenario 2

Figure 3: scenario 3

According to the aforementioned graphs, scenario 1's traffic flow was recorded for 40 seconds, with the majority of packets flowing within the first 2.0 to 10 seconds of the capture. The last packet is transferred at the 37 second mark, but the overall flow of traffic is moderate and there are no noticeable drops in the flow of packets over the captured network.

However, in scenario 2, the number of packets gathered is lower than in scenario 1, and the traffic flow lasts for 45 seconds. Here, at the start and end seconds show the highest packet flow rate, whereas between 5 and 40 seconds show a marked reduction in packet flow across the sample. The last packet for this transfer was transmitted on the 45th second.

So in scenario 3, the traffic flow was captured for 100 seconds. Here, between 20^{th} and 80^{th} seconds, the highest volume of packet traffic is seen. The final packet for this site was transmitted on the 100th second.

2. Network background traffic statistics generation script (plot_time_series_example.py) generates datasets using different probability distribution functions (PDF). Compare datasets generated using different PDFs. Discuss the difference between datasets and the data-set converted from operational network data.

Ans:

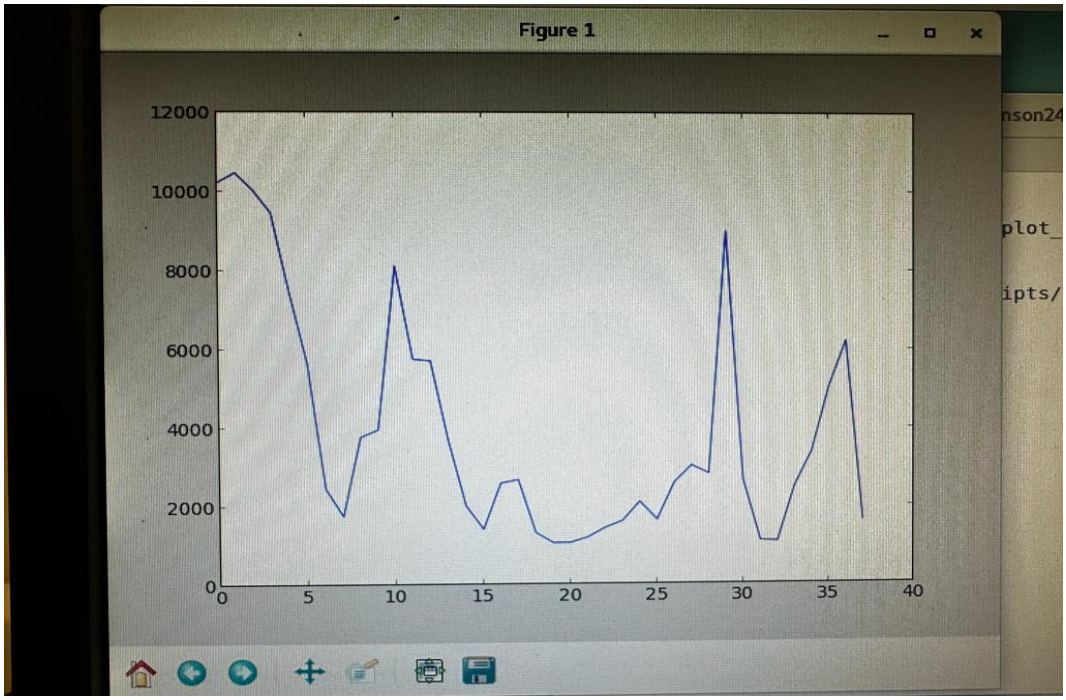


Figure 4: Dataset converted from operational network data

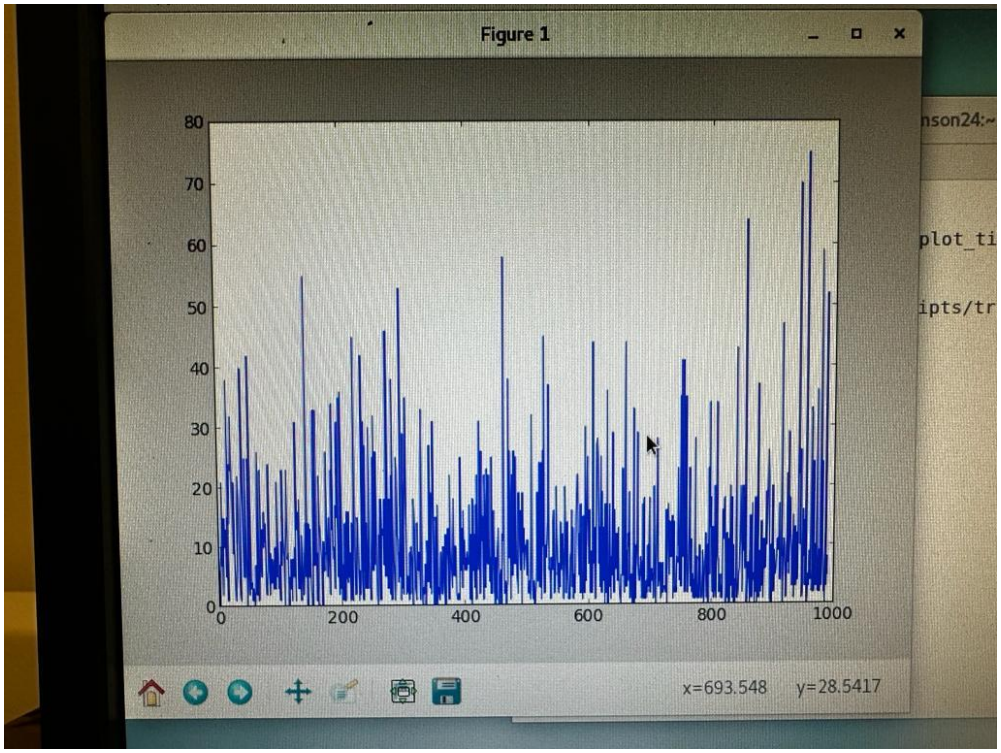


Figure 5: Data set generated using Tnorm probability distribution function.

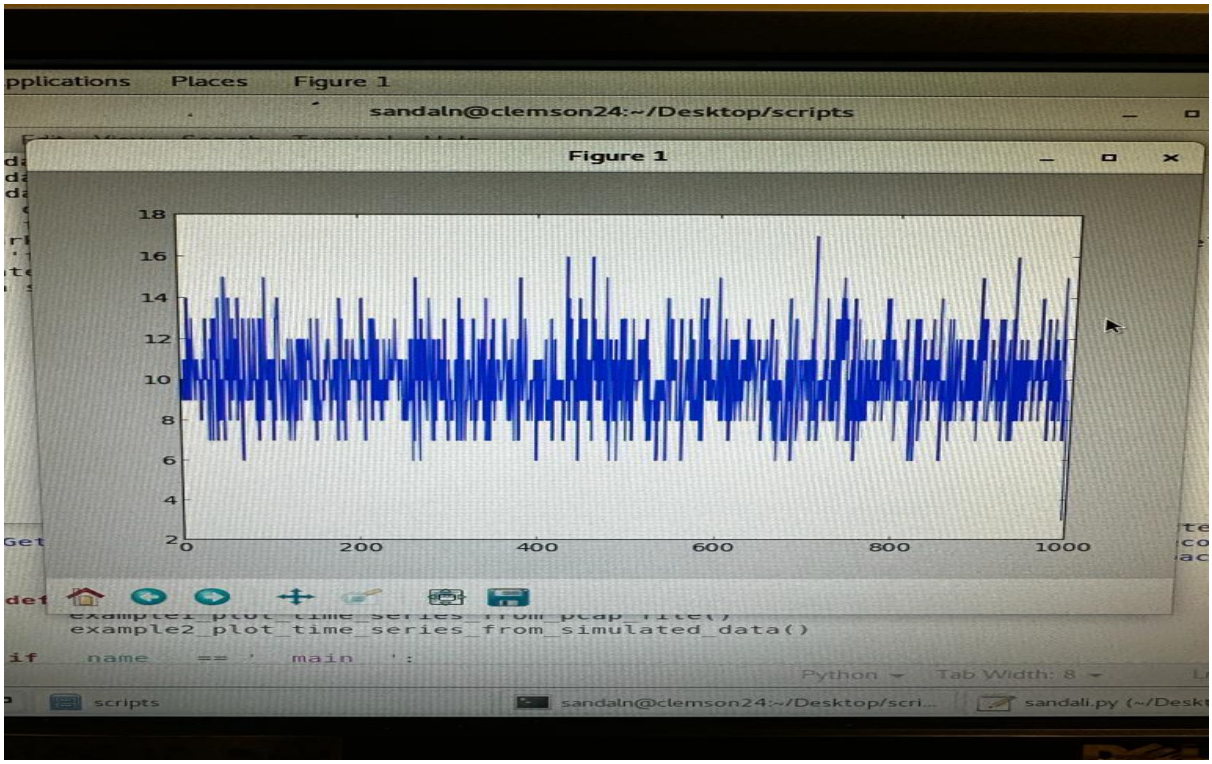


Figure 6: Data set generated using poisson probability distribution function.

Figure 4 shows that 40 seconds of traffic flow were captured. The final packet is transferred at the 37 second mark, however overall traffic flow is modest and there are no discernible drops in packet flow over the captured network.

In figure 5, the observed packets are all in the range of 0 and 1000. The traffic flow was recorded for 1000 seconds, and a total of 10,000 packets were plotted throughout that period. The greatest number of packet flows are observed during 1000th second. The last packet is sent on the 1000th second.

Figure 6 displays a traffic flow that was recorded during a time of 1000 seconds. In this instance, most of the packets, which are all in the 3 to 18 range, fall between the numbers 6 and 16, and the biggest packet flow is observed between 0 and 200 seconds. On the 1000th second, the final packet is transferred.

3. What kind of traffic did you run between hosts in the second scenario (Replay a pcap file with tcp replay). Justify why you think it can represent operational network traffic.

Ans: Simulated network traffic was sent between the hosts in the second instance. After operational network traffic, it is the second-best choice for network traffic that can be used for study. When the packets required to construct the replay traffic were recorded, they can be regarded as network traffic because they originated from a real, functioning network. Here, real-world data on the permitted network services, usual user count, and time of day all of which affect the amount of traffic produced have been used. As a result, the simulated traffic is as accurate as it can be to the actual network traffic.

4. You can replay and forward captured pcap files on a link in a controlled network Environment to use as background traffic. If you perform a DDoS attack on this link, you can observe the effects of the DDoS attack without jeopardizing the operational

network. However, some of the effects cannot be observed with replayed/forwarded background traffic. List some of these effects and explain the reason why they cannot be observed.

Ans: One effect that cannot be observed results from the background traffic being repeated or routed is responsive behavior. For instance, it is difficult to do a half-handshake during a SYN assault since the target would not respond in replayed traffic. This is because replayed traffic will convey data blindly and in a fixed order rather than adjusting to changing traffic conditions.

One of the extra effects that cannot be seen is coherence across the entire network. It is impossible to generate the network-wide coherence required for some observation sites to detect changes in traffic patterns when replayed traffic is occasionally too scarce or too congested.

5. Number of packets received by a node on the network is one of the popular metrics used in DDoS detection applications. In this assignment, we focused on packet count statistics. Discuss what other metrics that can be used for DDoS detection.

Ans: The Hartley, Shannon, Renyi, and generalized entropies can all be utilized to identify DDoS attacks. Using the proper metric makes it simpler to develop a model that can accurately detect DDoS attacks. These metrics can be used to characterize network traffic characteristics.

When the difference between the entropy of the flow count and the mean value of entropy is greater than the threshold value, which is adjusted adaptively based on traffic pattern, DDoS attacks can frequently be detected.

Attack Generation

Methodology:

1. Connect to the security lab machines(CNC machine and bot machine).
2. Run scripts to measure the performance of the attacks.
3. Capture traffic by launching a DNS amplification attack.
4. Launch 3 different combinations of attacks and measure the performance.

Analysis:

1. Measured the performance by performing Flood attack and slow http attack.

Questions:

1. Discuss the difference between each attack combination you choose.

Ans:

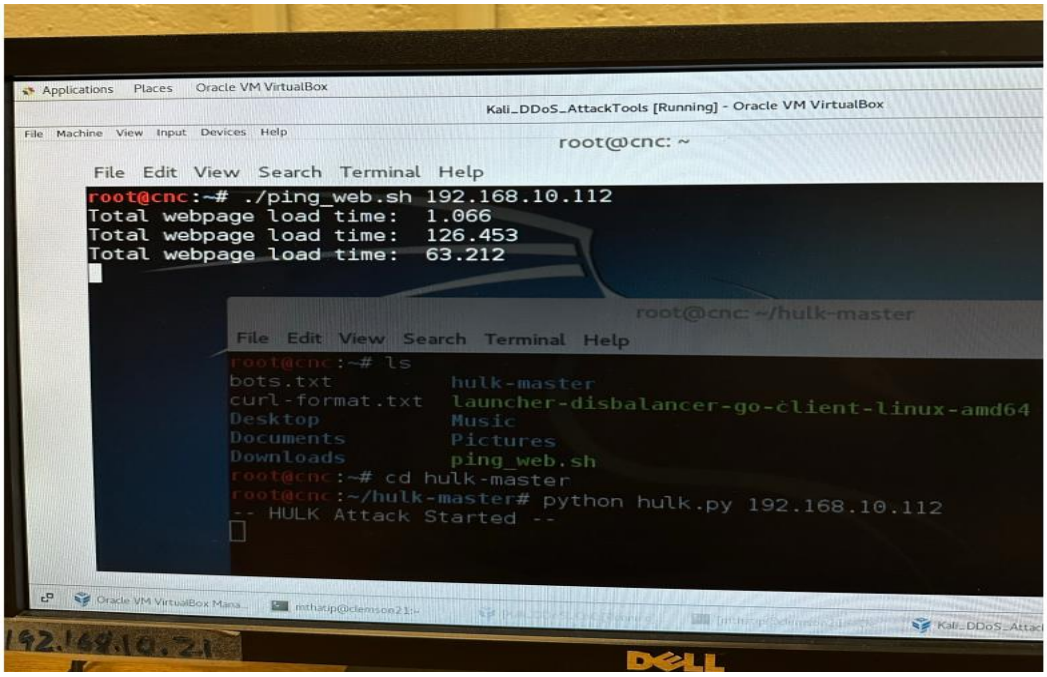


Figure 7: Response time of the victim during flood attack & SYN flood attack that was carried out simultaneously

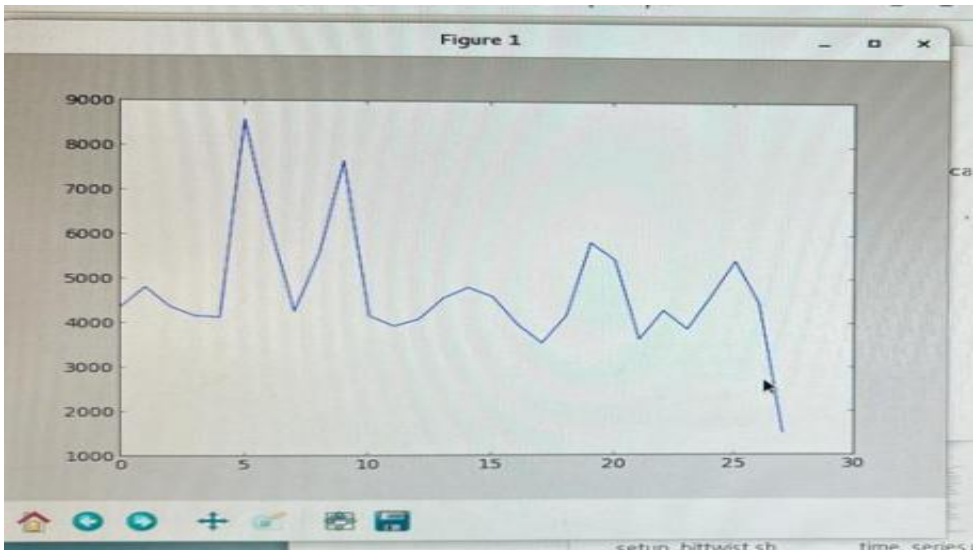


Figure 8: Time-series graph for the traffic captured during flood attack & SYN flood attack that was carried out simultaneously

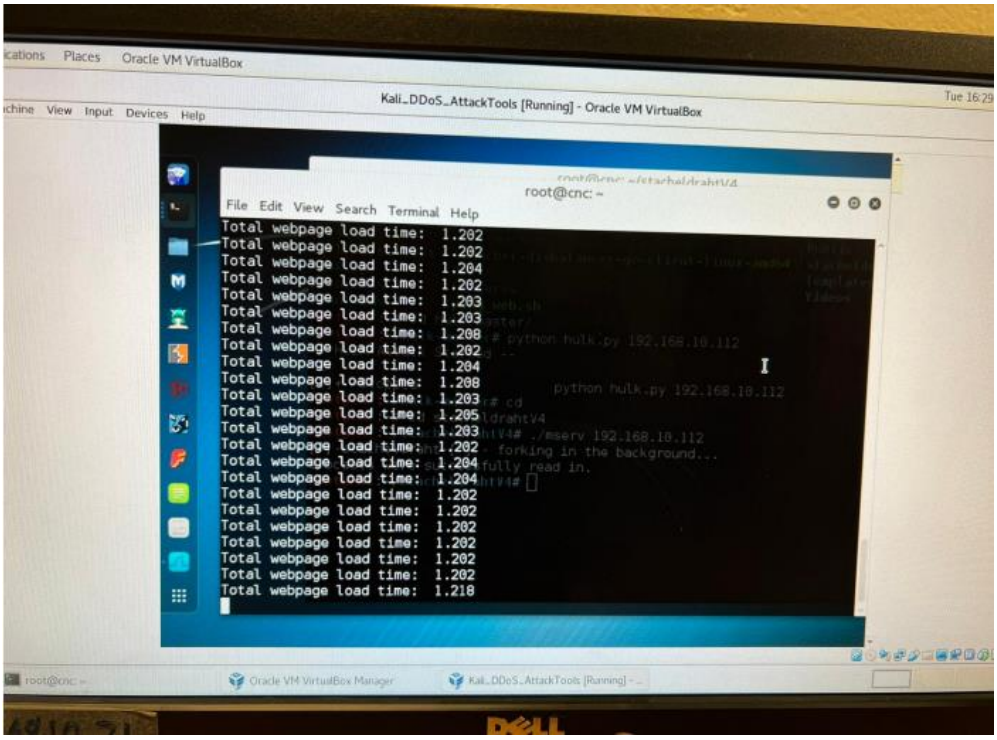


Figure 9: Response time of the victim during SYN flood attack & Slow HTTP attack that was carried out simultaneously

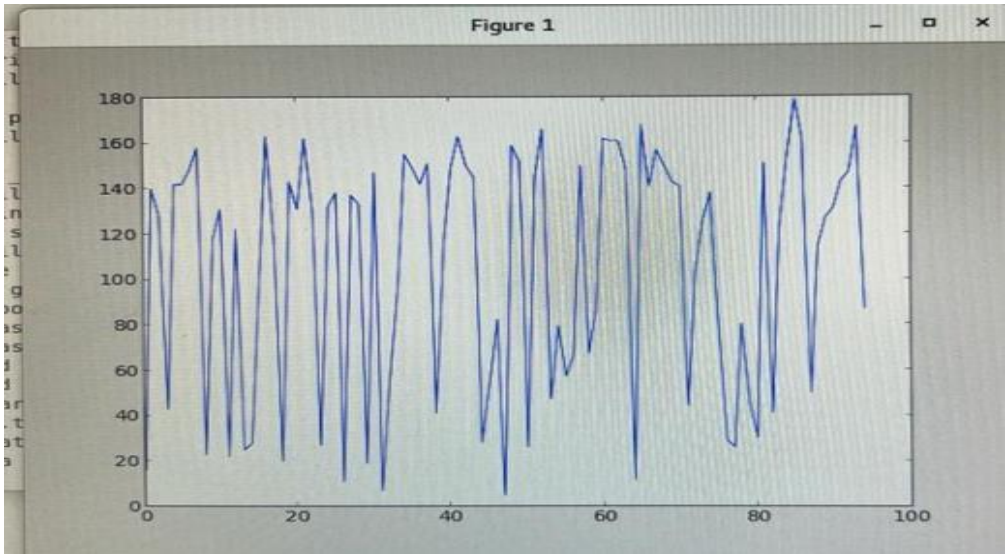


Figure 10: Time-series graph for the traffic capturing SYN flood attack & Slow HTTP attack that was carried out simultaneously

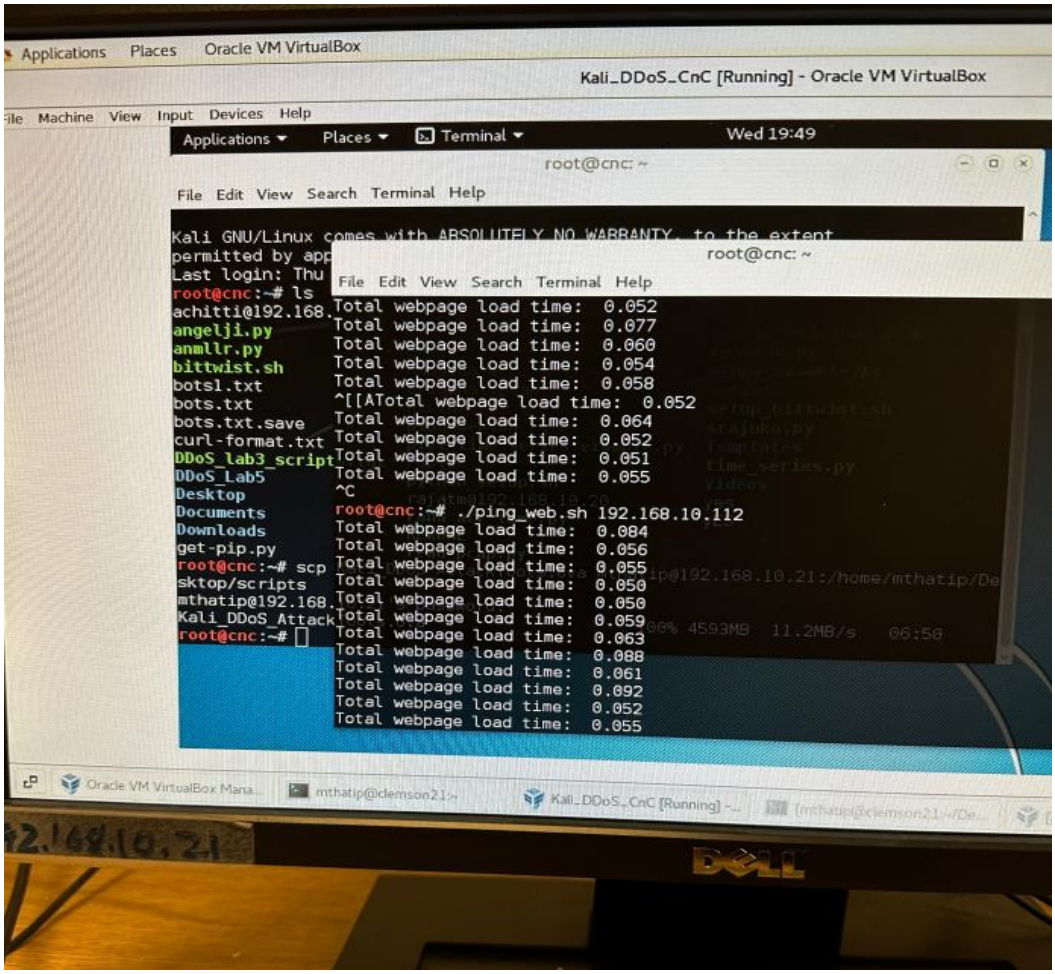


Figure 11: Response time of the victim during Flood attack & Slow HTTP attack that was carried out simultaneously

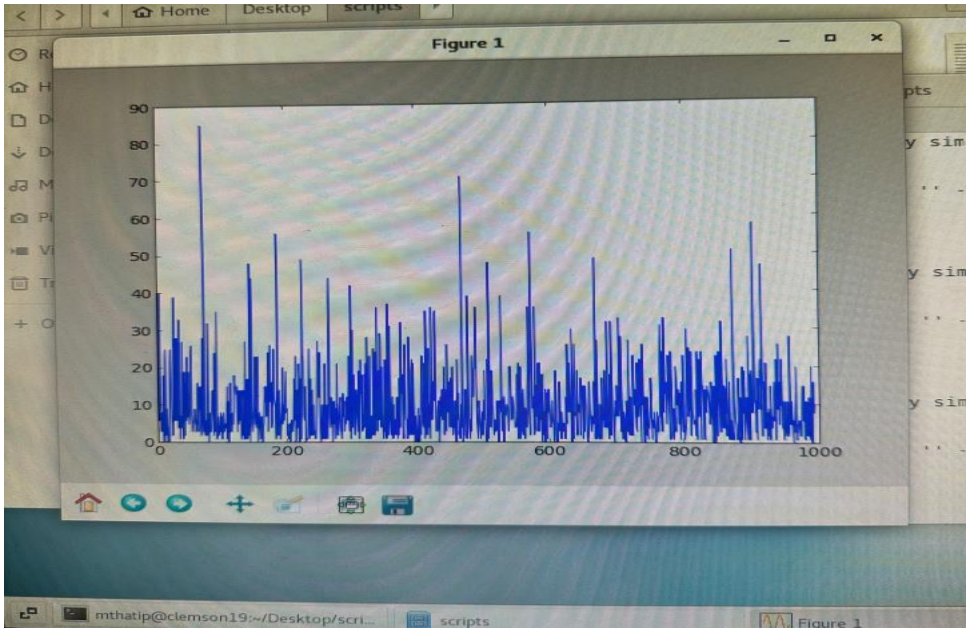


Figure 12: Time-series graph for the traffic capturing Flood attack & Slow HTTP attack that was carried out simultaneously

There are three attack combinations I may select from: 1) Flood attack and SYN attack, 2) SYN flood attack and Slow HTTP attack, and 3) Flood attack and Slow HTTP attack. These three combinations vary from one another in the following ways: Flood attack & SYN flood attack: The bandwidth of the target is used up by sending many packets during the attack. The victim's response time gradually extends beyond its typical range because of the other batch of packets delivered, keeping the TCP handshake partially open, until the attack is terminated when the maximum value (in my case, 126.453) is achieved. Also, you can see the time-series graph of the traffic captured during this combination of attack below (Figure: 8). We can notice how the traffic flow is moderate until the end. This is because the number of packets I sent out for the attack are low compared to an actual attack. Figure 8: Time-series graph for the traffic captured during flood attack & SYN flood attack that was carried out simultaneously SYN flood attack and slow HTTP attack: In this attack, many packets establish a partial 3-way TCP handshake, which depletes resources, and the remaining attack packets open http connections for every available socket with the victim, tying up the victim. Here, the response time does not increase gradually but rather reaches its maximum (in my case, 1.218) before the attack is terminated.

The time-series graph for the traffic recorded during this particular attack combination is displayed below. (Figure 10. The curves are sharper in this instance than they are in the other two attack combinations, as can be seen. Thus, more packets are sent more often and at a higher rate during the attack. Figure 10: Time-series graph for the traffic capturing SYN flood attack & Slow HTTP attack that was carried out simultaneously Flood attack and slow HTTP attack: In this attack, a large number of packets are delivered throughout the attack period, exhausting the victim's bandwidth, and another group of packets contains open http connections to all of the target's available sockets, tying up the victim. Reaction time is not increased gradually here; rather, it increases to a maximum value (in my example, 0.084) before the attack is terminated (Figure: 11). Figure 11: Response time of the victim during Flood attack & Slow HTTP attack that was carried out simultaneously Figure 12, below displays the time-series graph for this collection of intercepted attack communications. As we can see, this attack's packet volume is lower than that of the other two combinations. Figure 12: Time-series graph for the traffic capturing Flood attack & Slow HTTP attack that was carried out simultaneously Flood attack & Slow HTTP attack appears to be the greatest option out of all the combinations I've explored because it targets the victim's bandwidth as well as starving them of resources.

2. For SYN Flood attack, capture a complete TCP three-way handshake process and explain it in detail.

Ans: The TCP three-way handshake is the first thing that happens when a client tries to connect to a server using TCP. The following are the steps in the process:

SYN: The client sends a SYN (synchronize) packet to the server along with an initial sequence number to establish the connection (ISN). The ISN is a random number to protect against sequence number prediction attacks.

SYN-ACK: After receiving the SYN packet, the server sends the client a SYN-ACK (synchronize acknowledge) packet. This packet includes the ISN of the server together with an acknowledgment number (ACK) that is equal to the ISN of the client plus one. The client's request is acknowledged at this step, and the following one is organized.

ACK: When the client sends the server an ACK packet to verify receipt of the server's SYN-ACK packet, the connection is established.

In a SYN Flood attack, the attacker bombards the server with SYN packets without completing the last stage of the three-way handshake.

This happens as the server awaits the ACK packets to finish the connection gives it a deluge of partially open connections. As a result, authentic clients are unable to access the server, and its resources are rapidly used up.

In conclusion, the TCP three-way handshake procedure must be finished in order to create a connection between a client and a server. Yet, a SYN Flood attack could flood the server with connections that are only partially open, exhausting its resources and blocking legitimate connections.

3. For DNS amplification attack, capture the traffic sent by the attacker and the traffic received by the victim and calculate the amplification ratio.

Ans: The traffic sent by the attacker (Figure: 13) and the traffic received by the victim (Figure: 14) are captured and shown below. Here, in my case, the number of packets sent by the attacker are 10 and the number of packets received by the victim are 100. Which calculates to the amplification ratio of 10.0.

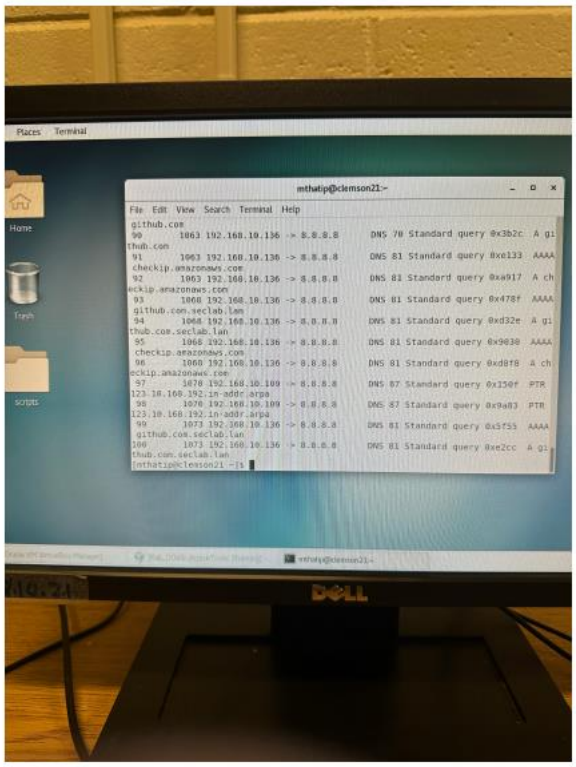


Figure 13: Traffic sent by attacker

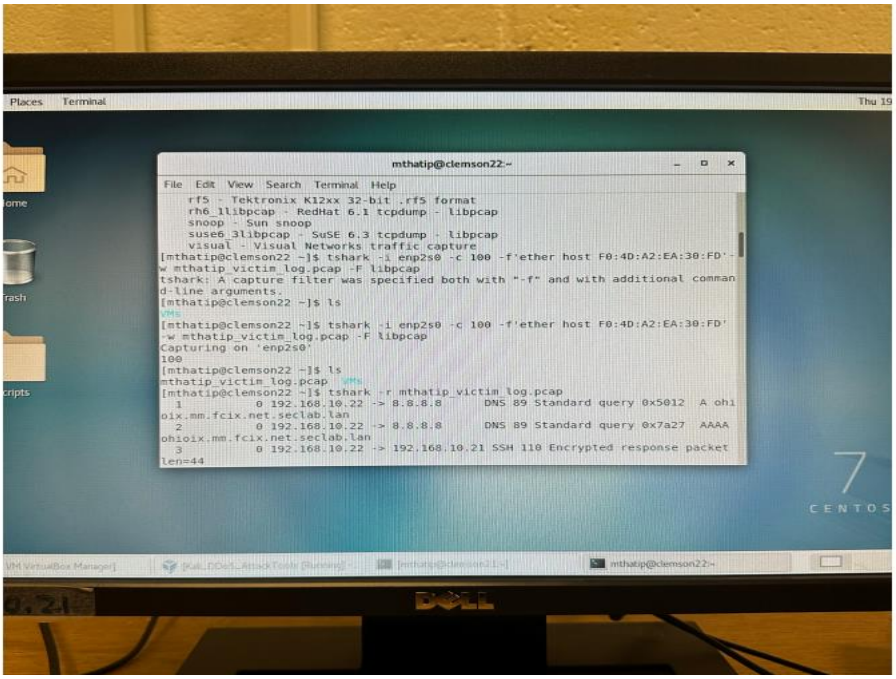


Figure 14: traffic received by the victim.

4. For each attack combination, discuss the following questions:
   - I. Capture attack traffic and explain its effect on the network and victim node.
   - II. Discuss the changes in system availability between before and after the attack.
   - III. Discuss the changes in system resource usage before and after the attack.

Ans: a) Flood attack and Syn flood attack

An illustration of the attack traffic's time series is shown below (Figure 16). Here, a flood attack uses up the victim's bandwidth, the network is congested with packets, restricting the victim's access to legitimate traffic, and the SYN flood attack exhausts the victim node's resources.

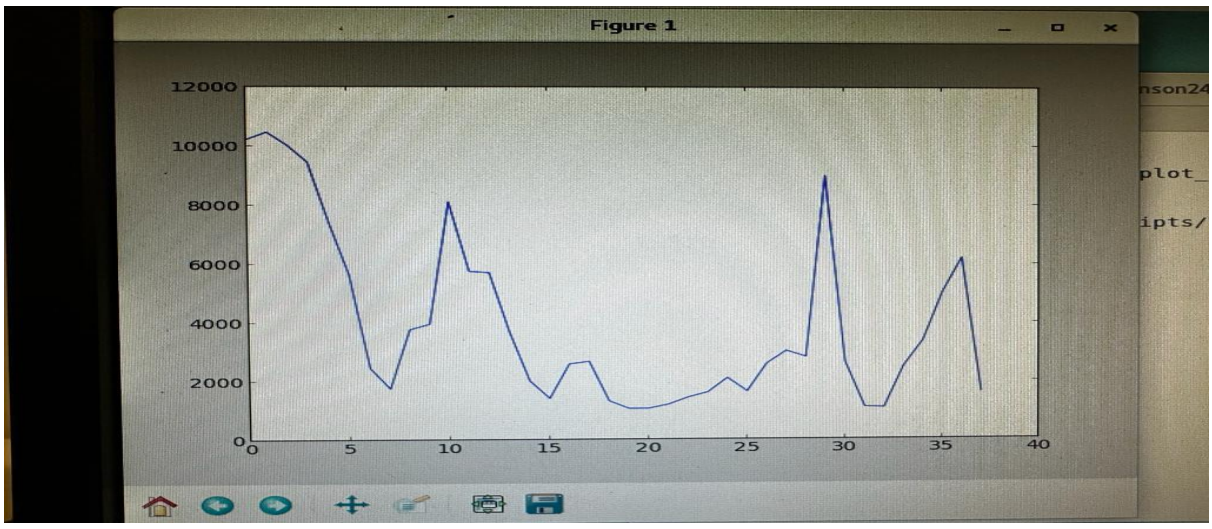


Figure 16: Captured Flood attack & SYN flood attack traffic

Before the assault, the victim system answers to requests with a load time of 0.045 to 0.055 and is prepared to accept new requests. After the assault, however, the victim's resources are completely used up, availability gradually falls as the load time increases, and eventually the system stops being ready for new requests. In my case, the assault started, the victim system ceased to function after a short period of time, and the load time climbed until it reached 31.110.

All legitimate traffic had its fair amount of resources prior to the attack, and the victim's resource allocations were functioning normally. Yet, open 3-way TCP connections were totally absorbed by the assault traffic following the attack, depriving the victim of any resources to allocate to legitimate traffic.

b)Syn flood attack and slow http attack

The attack traffic during a SYN flood attack and a slow HTTP attack is shown below (Figure 17). Here, the SYN flood attack causes packets to flood the network, using up all of the target node's bandwidth and depleting its resources, starving the victim of resources for legitimate traffic. The slow HTTP attack then accelerates the resource starvation.

Figure 17: Captured SYN flood attack & Slow HTTP attack

All of the target node's open sockets are devoured at the time of the attack, and the attack packets wait for the closed sockets to open up before successfully devouring them as well. The victim system is ready to accept new requests and responds to any within the load time range of 0.045 to 0.055 prior to the assault. When this is finished, access to the system is lost.

All legitimate traffic had its fair amount of resources prior to the attack, and the victim's resource allocations were functioning normally. Open 3-way TCP connections and partially

open http connections were totally occupied by the attack traffic after the attack, however, leaving the victim with no resources to allocate to legitimate traffic.

c) Flood attack & Slow HTTP attack

Below is a capture of the attack traffic with the combination of a flood attack and a slow HTTP attack (Figure 18). In this case, sluggish HTTP attack delivers partial HTTP requests to the victim in order to deplete resources, while flood attack consumes up the victim's bandwidth and jams packets throughout the network, obstructing genuine traffic to the victim.

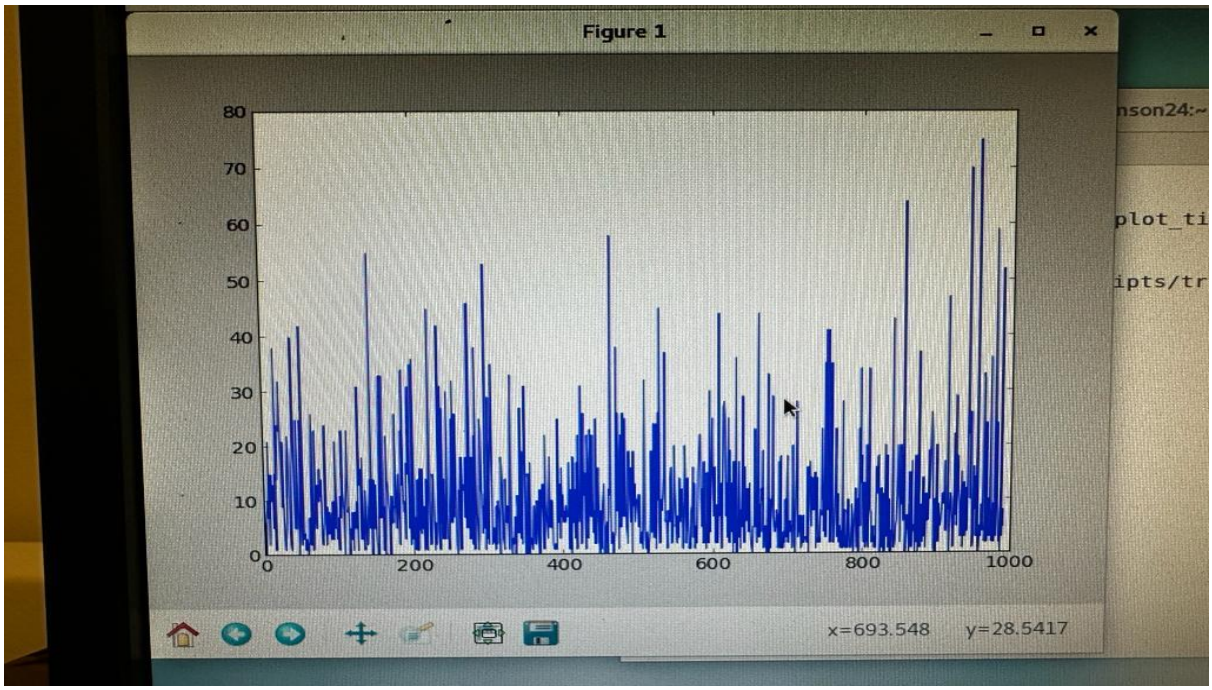


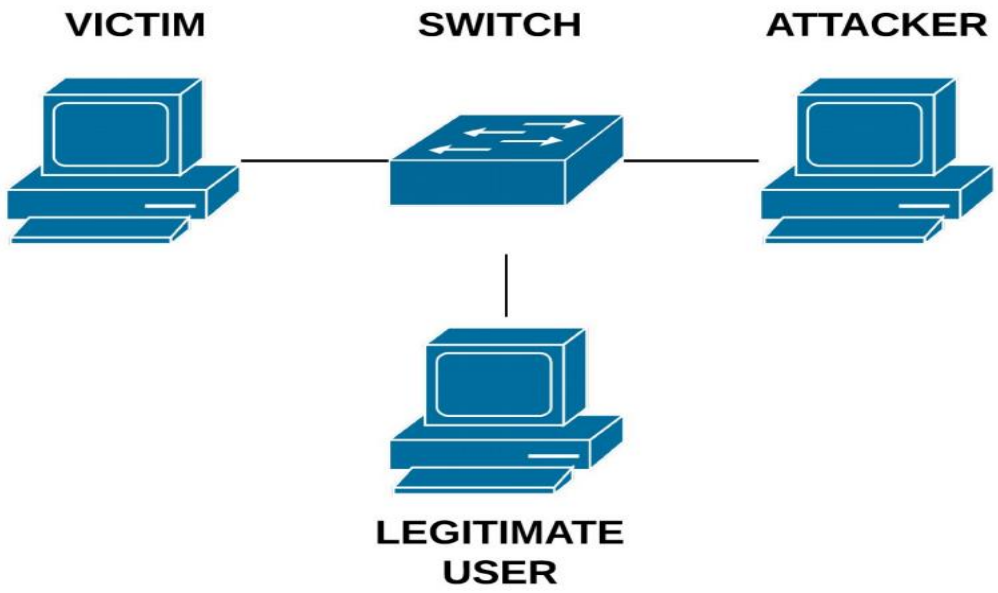
Figure 18: Captured Flood attack & Slow HTTP attack

All of the target node's open sockets are devoured at the time of the attack, and the attack packets wait for the closed sockets to open up before successfully devouring them as well. The victim system is ready to accept new requests and responds to any within the load time range of 0.045 to 0.055 prior to the assault. When this is finished, access to the system is lost.

All legitimate traffic had its fair amount of resources prior to the attack, and the victim's resource allocations were functioning normally. But, following the attack, broken http connections made it impossible for legitimate traffic to use the services. All resources were utilised by the assault traffic.

SYN Flood attack

Methodology:



Analysis:

1. With tools like Stacheldraht, Hulk, slowloris, and hping3, a Syn flooding attack can be carried out.
2. Acknowledged the TCP handshake procedure.
3. Saw how a typical flooding attack affected a server or network.

Questions:

1. a. Discuss and compare each attack tool.
   Ans: DDoS (Distributed Denial of Service) assaults can be launched using a number of tools, including Stacheldraht, Hulk, slowloris, and hping3. Both of them want to stop a particular service from being offered, but they approach the problem differently.

   Stacheldraht: Stacheldraht uses a combination of UDP, TCP, and ICMP flood assaults to completely overwhelm the target server. It can also be used to launch Smurf and Fraggles attacks. This program is quite dangerous and can be difficult to detect and block due to the multiple attack routes it uses.

   A Python-based software called Hulk bombards a server with HTTP GET requests. It generates a lot of HTTP requests with erratic user agents and referrers, which overwhelms the server and causes it to crash. Although it is a simple and helpful tool, its utility can be limited if used against servers that have strong security.

   Another HTTP-based utility, Slowloris establishes many connections to a server and keeps each one open for as long as is practical. By constantly transmitting small amounts of data, the software can cause the server to crash by using up all of the server's resources.Servers that are not built to manage a high amount of open connections are particularly vulnerable to Slowloris' efficacy.

Hping3: To generate different types of network traffic, use the Hping3 command-line utility. It can be used for a variety of attacks, such as SYN flooding, which involves flooding a server with SYN packets in an effort to clog up its connection queue. Hping3 may spoof IP addresses, which makes it difficult for the target server to figure out where the assault came.

Stacheldraht is possibly the most devastating instrument in terms of effectiveness because it can deploy several attack vectors at once and is tough to recognize and prevent. But, it is difficult to access and requires a high level of technical expertise.

In conclusion, it is crucial to keep in mind that DDoS assaults are illegal and that those who carry them out run the possibility of suffering serious consequences. It is always best to handle any complaints you may have with a website or service through legal and ethical means.

b) Discuss the most popular and still actively used is Disbalancer (also called "Liberator"), a DDoS tool used to take down infrastructure targets in Russia, and its effects.
Ans: Disbalancer, also referred to as "Liberator," is a DDoS technique that has primarily been employed in Russia to attack government infrastructure and organizations.
The tool has the potential to produce colossal amounts of traffic, which can cripple the systems being targeted and overwhelm them.
A extremely potent tool called Disbalancer has been employed in a number of big DDoS attacks in Russia. In 2016, a number of Russian banks were targeted, seriously affecting the country's financial systems. This incident is one of the most notable ones. The attack was so severe that the Russian Central Bank had to intervene to prevent a financial collapse.
Disbalancer is still in use today and is a significant threat to Russian infrastructure and organizations. Due to the program's high degree of customizability, specific system vulnerabilities can be targeted with it. Although it can be difficult to detect and block, it is also incredibly robust.
Disbalancer attacks have the potential to be destructive and have far-reaching impacts. DDoS assaults can seriously disrupt the targeted organizations as well as cause significant financial loss. In the future, businesses might need to invest more in resources to ward off attacks, and service interruptions could cost them money.
Disbalancer is a dangerous and incredibly powerful tool that highlights the ongoing danger of DDoS attacks. Organizations need to be vigilant and spend heavily in cybersecurity if they want to protect themselves from this kind of danger. Law enforcement agencies must act to identify and apprehend those responsible for such DDoS assaults in order to stop them.

Results:

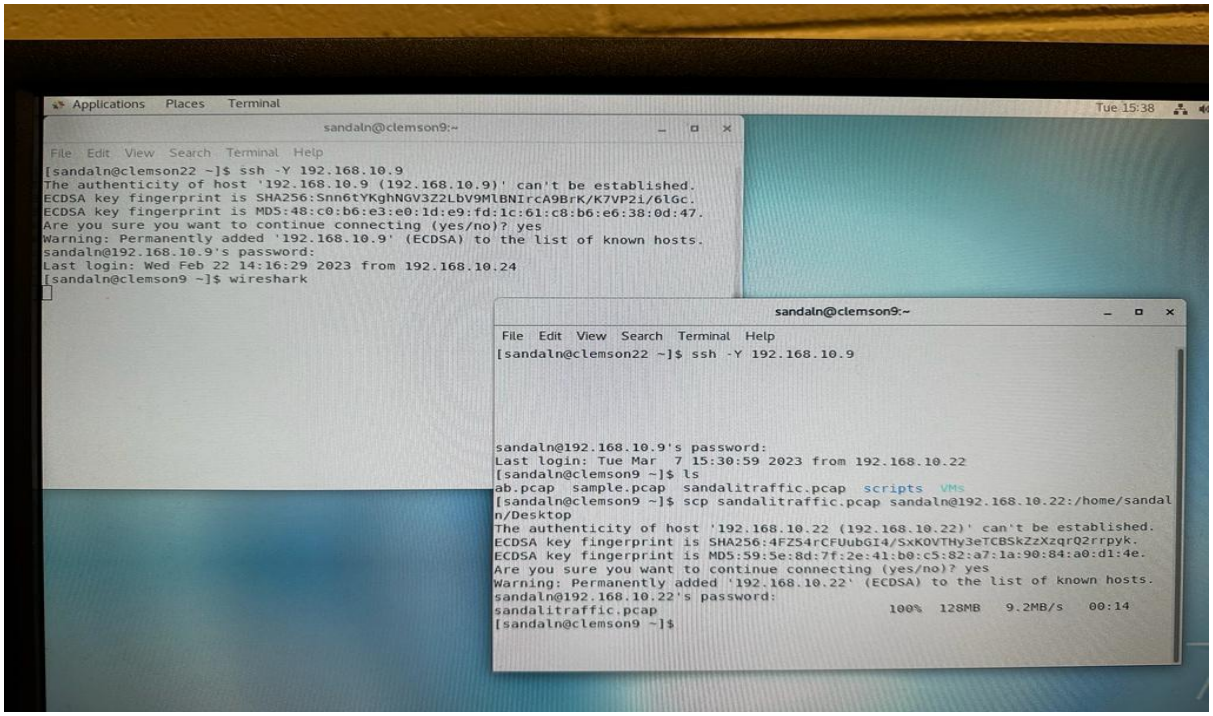


Figure 19: connected to sniffer, sent traffic file to the machine

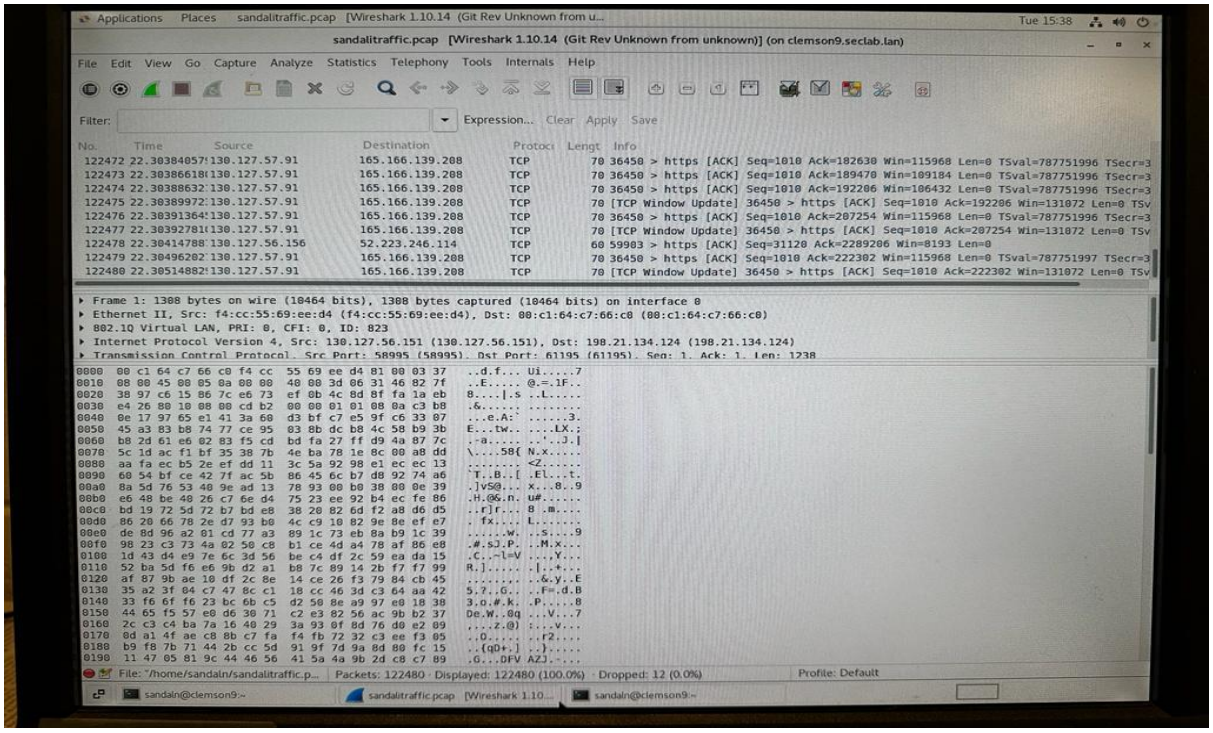


Figure 20: Captured more than 100000 packets on wireshark

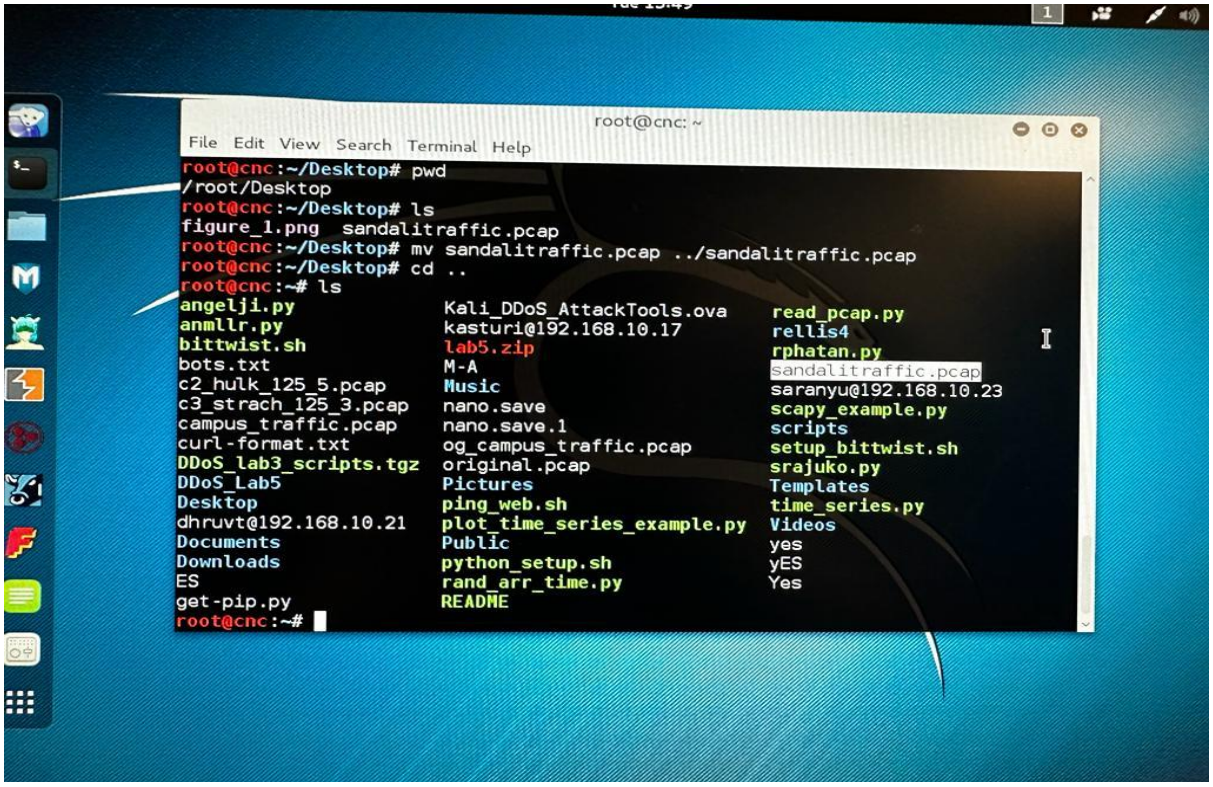


Figure 21: sent the traffic file to the VM

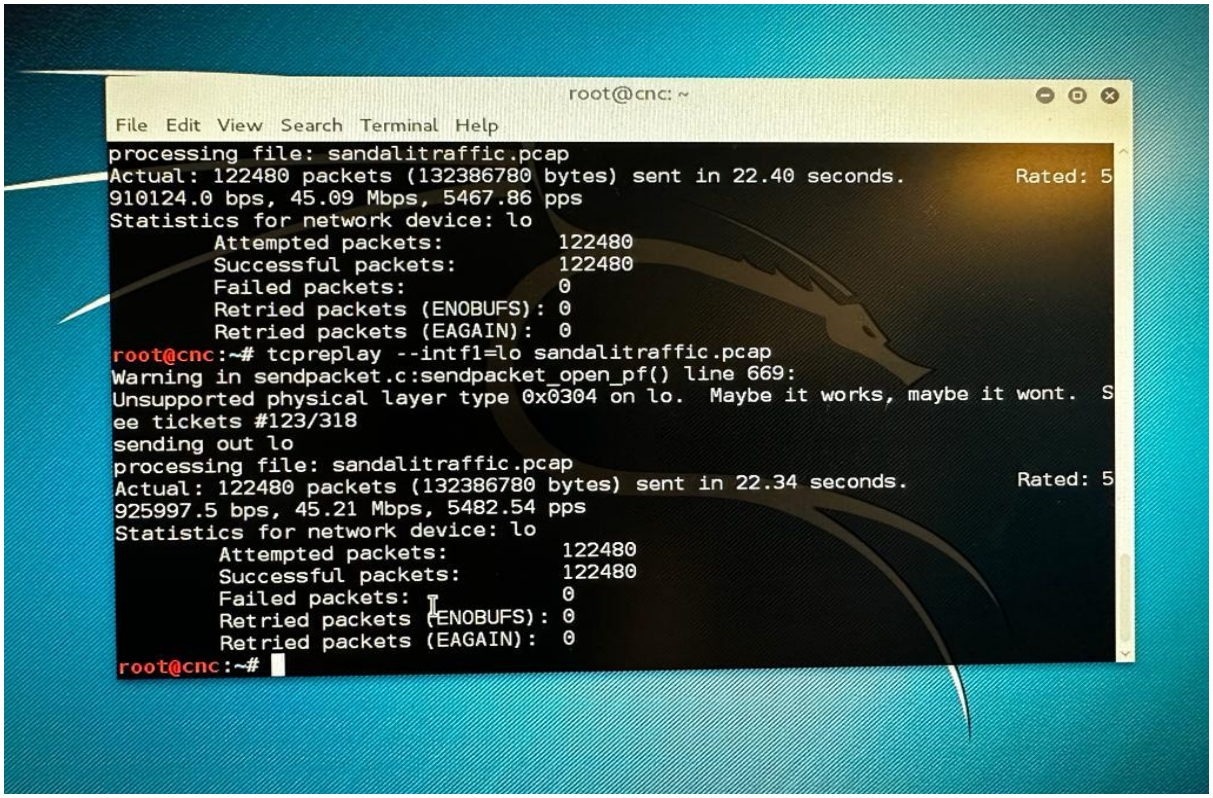


Figure 22: Replayed the traffic file

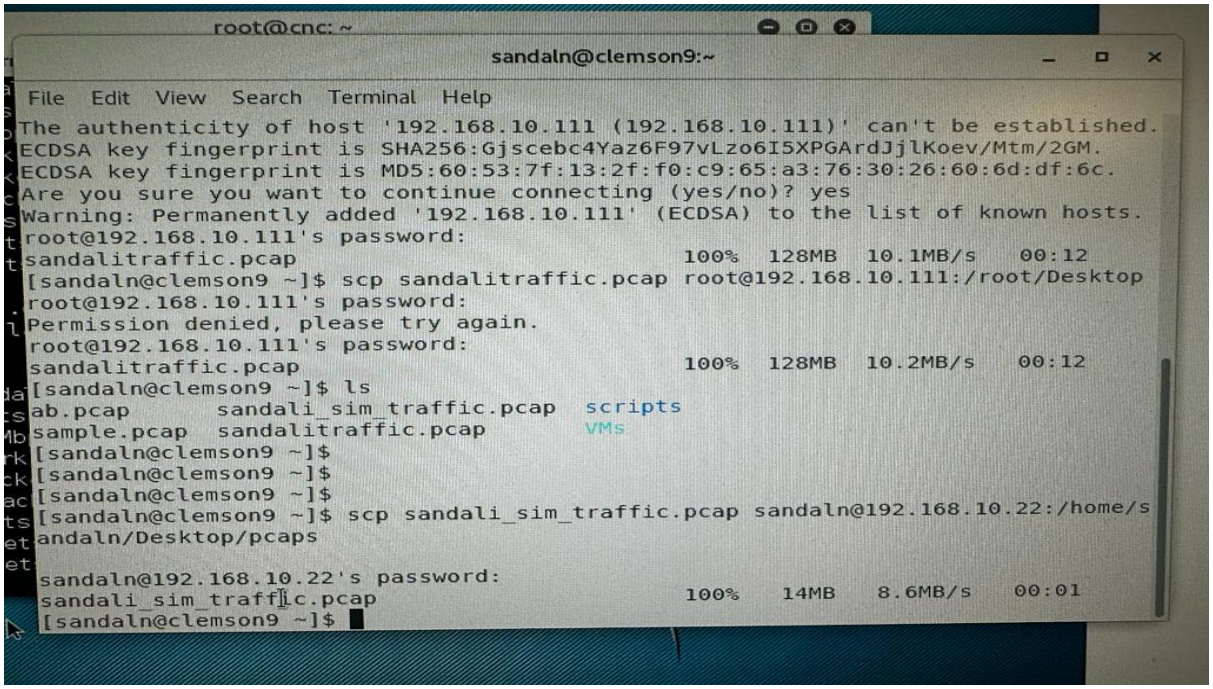
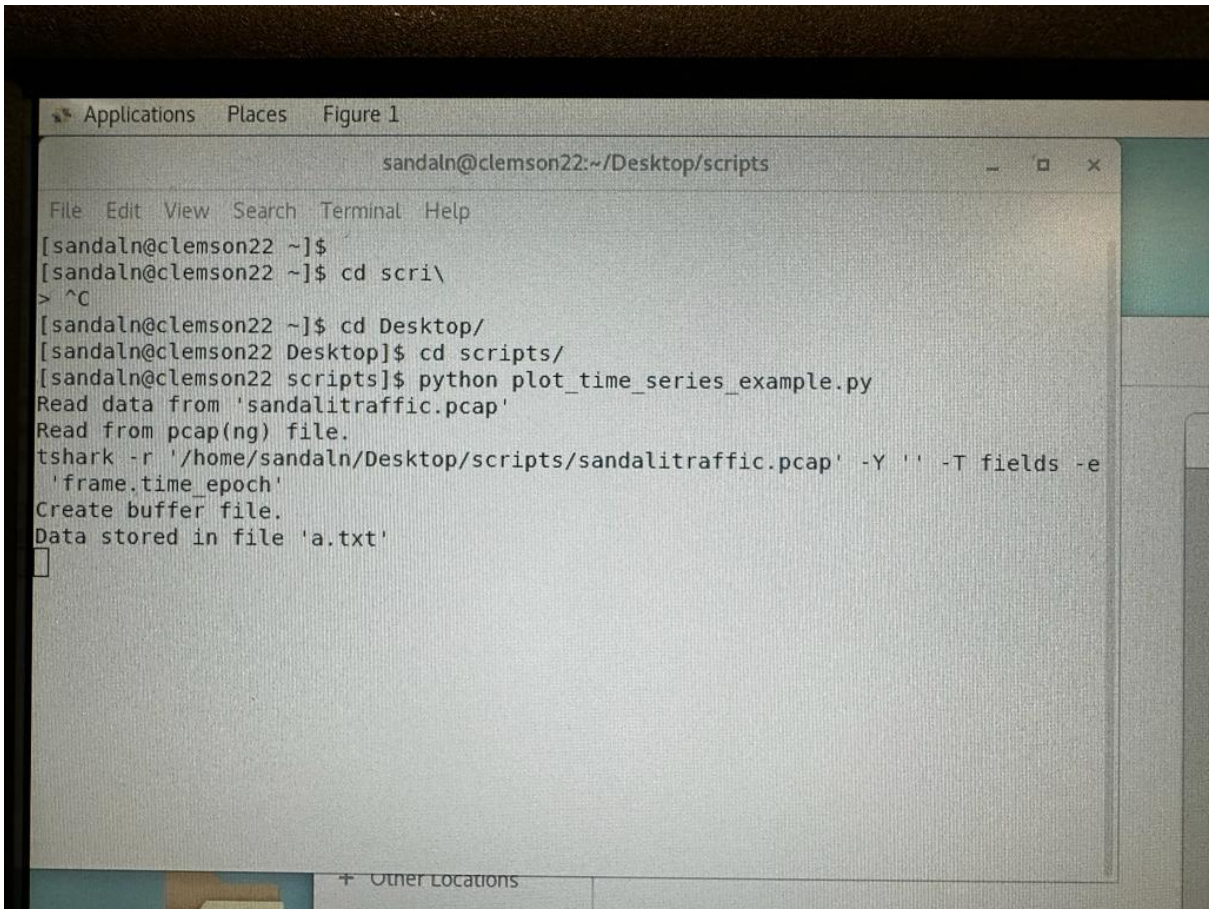


Figure 23: Transferred the reply.pcap and traffic files to the local machine.



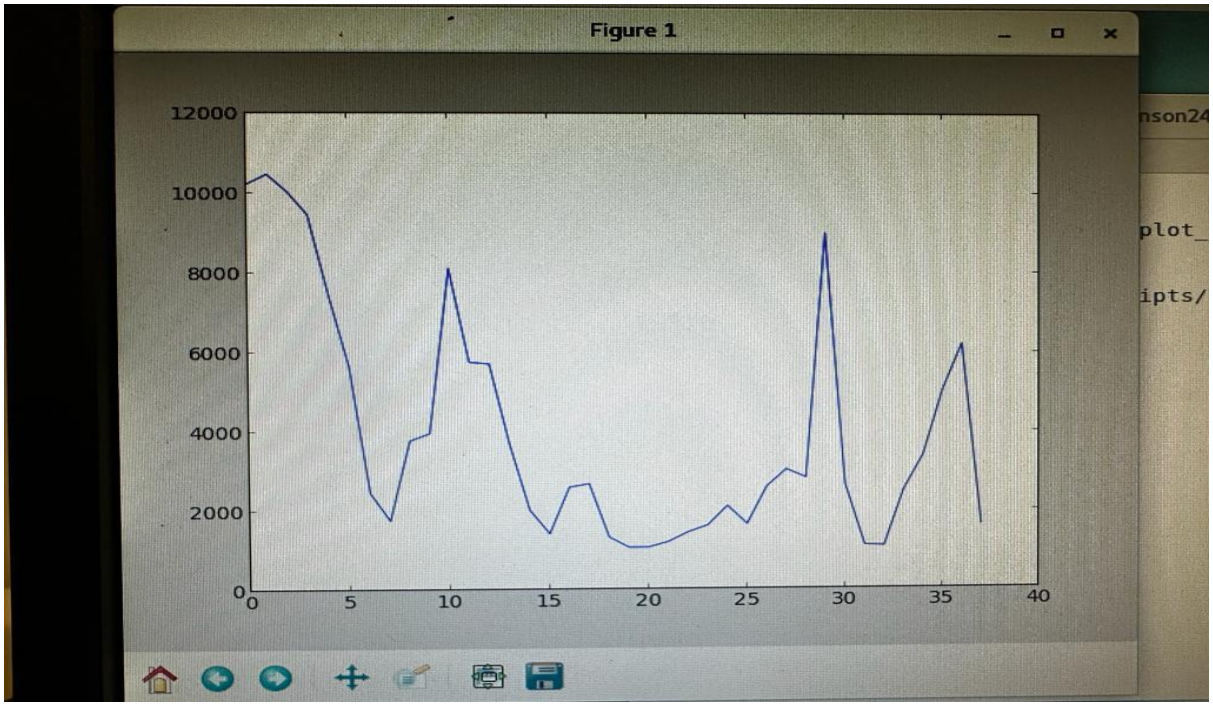
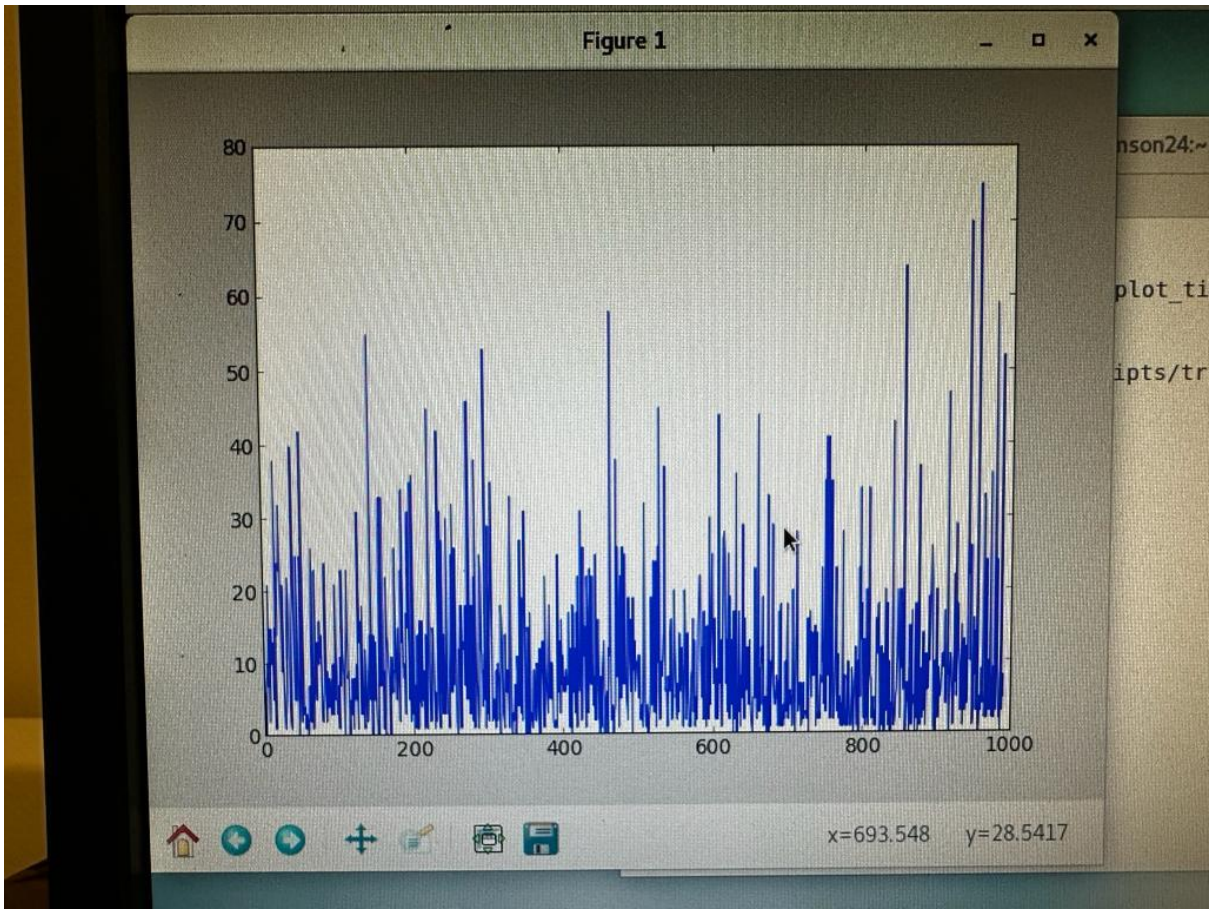


Figure 24: real time traffic plot



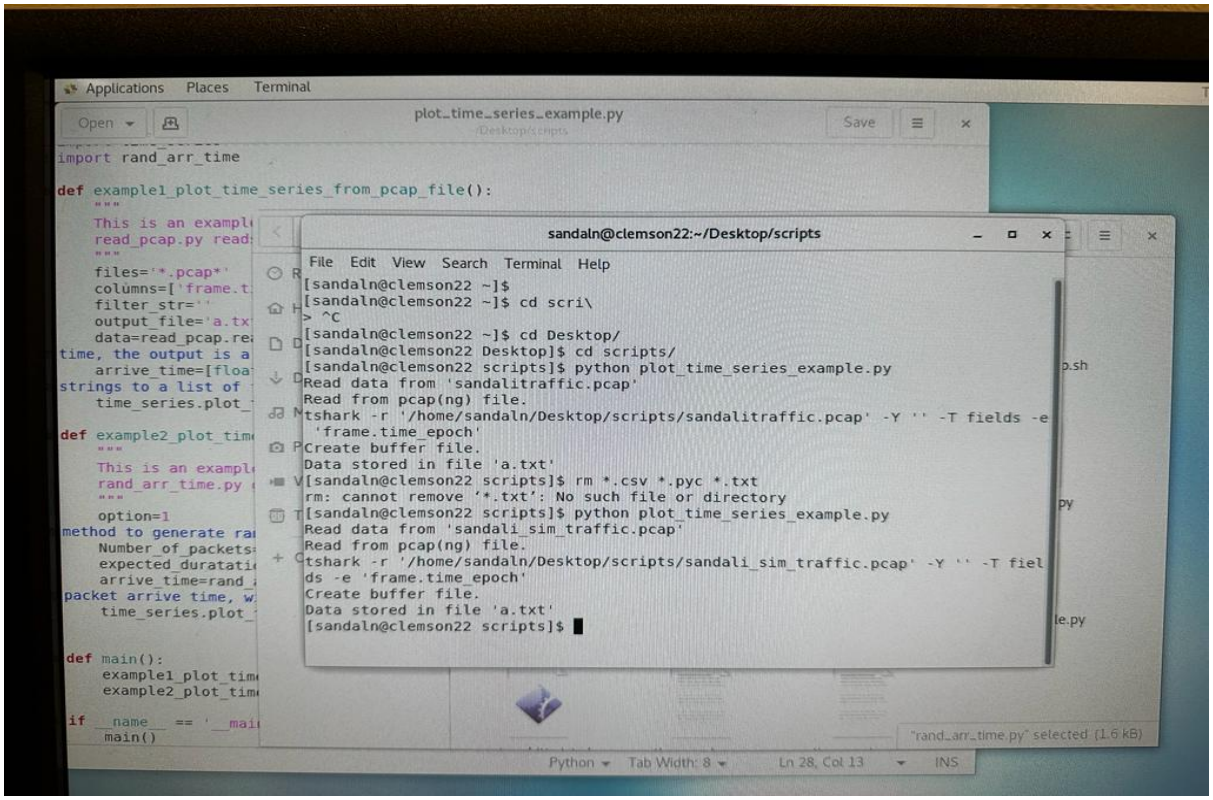


Figure 25: By changing the option plotting graph

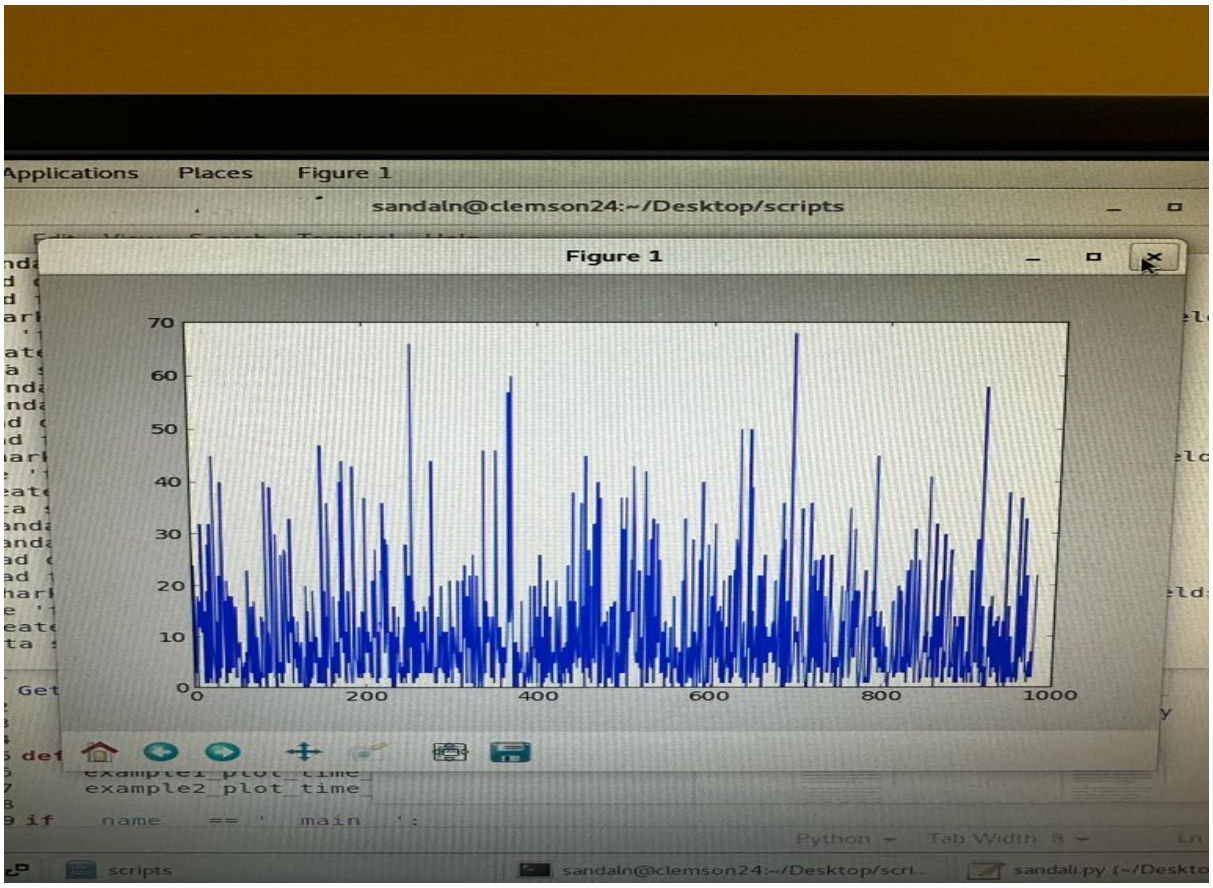


Figure 26: simulated traffic which is closer to the real

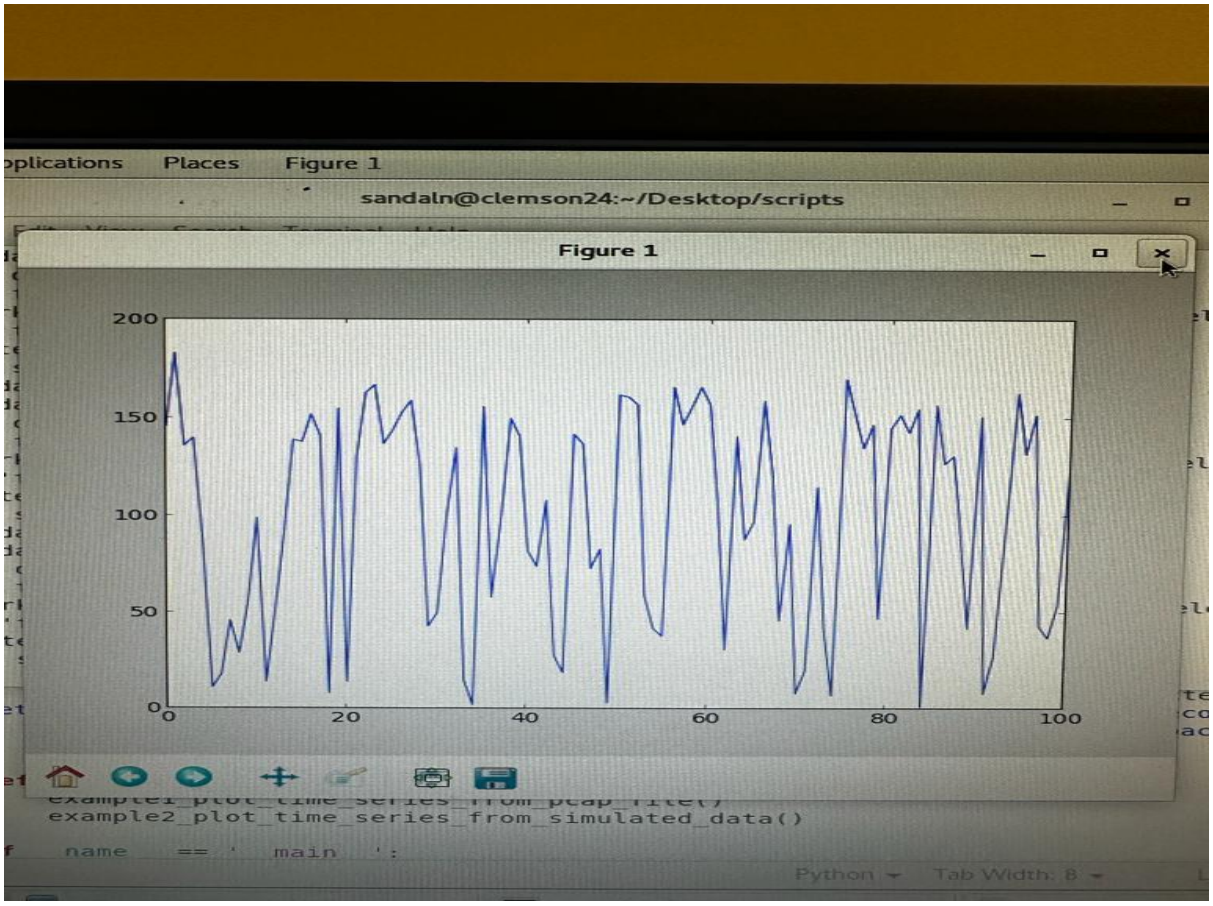


Figure 27: simulated traffic which is closer to the real

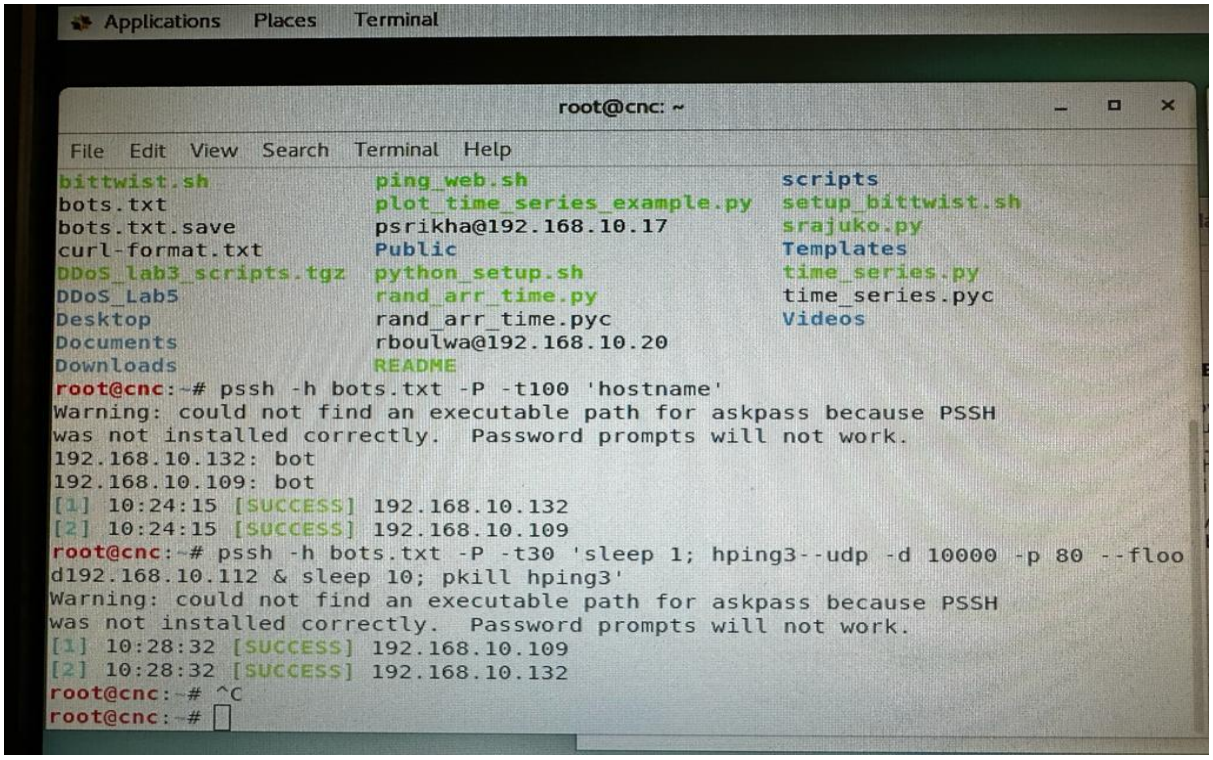


Figure 28: two bot machines added and connected successfully.

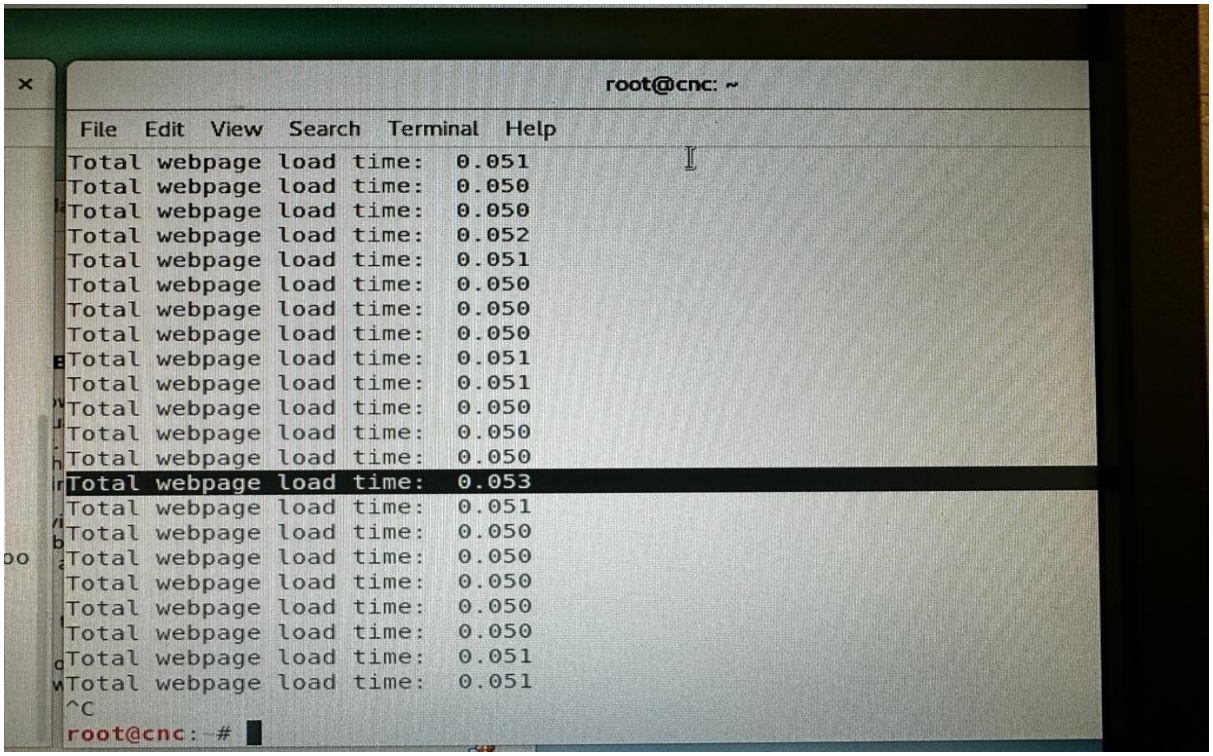


Figure 29: victim response time

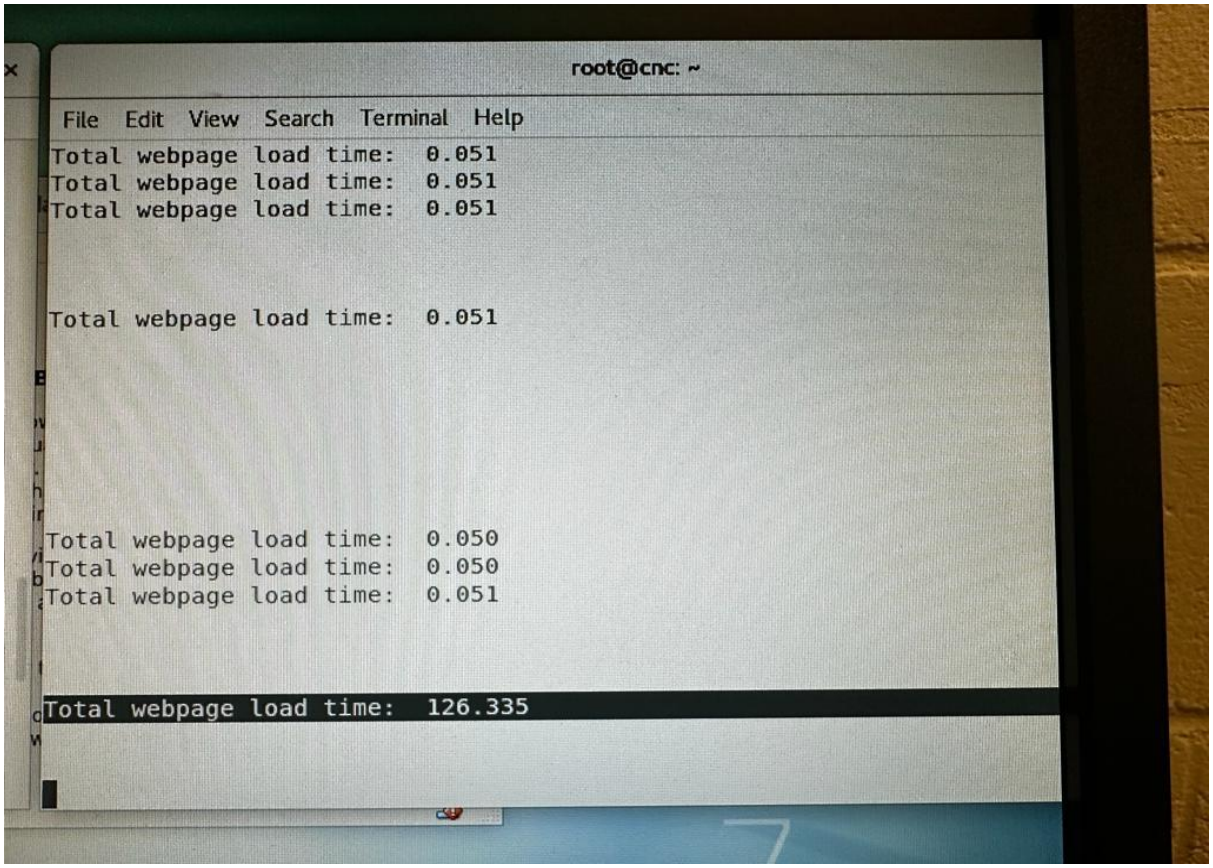


Figure 30: response time increased after attack

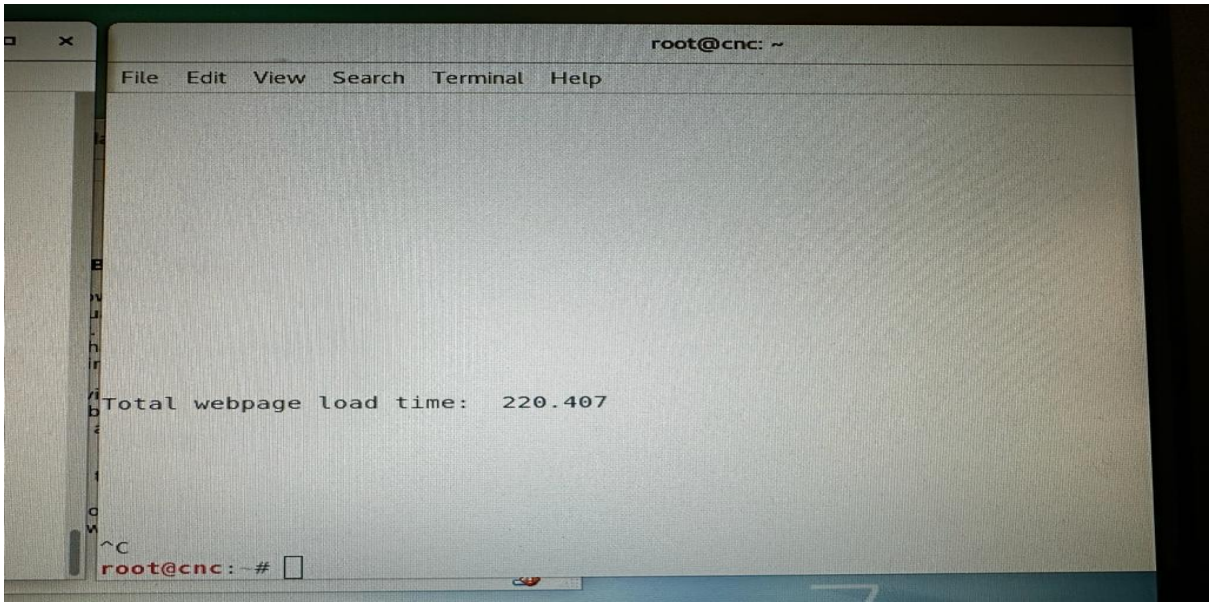


Figure 31: time increased after attack

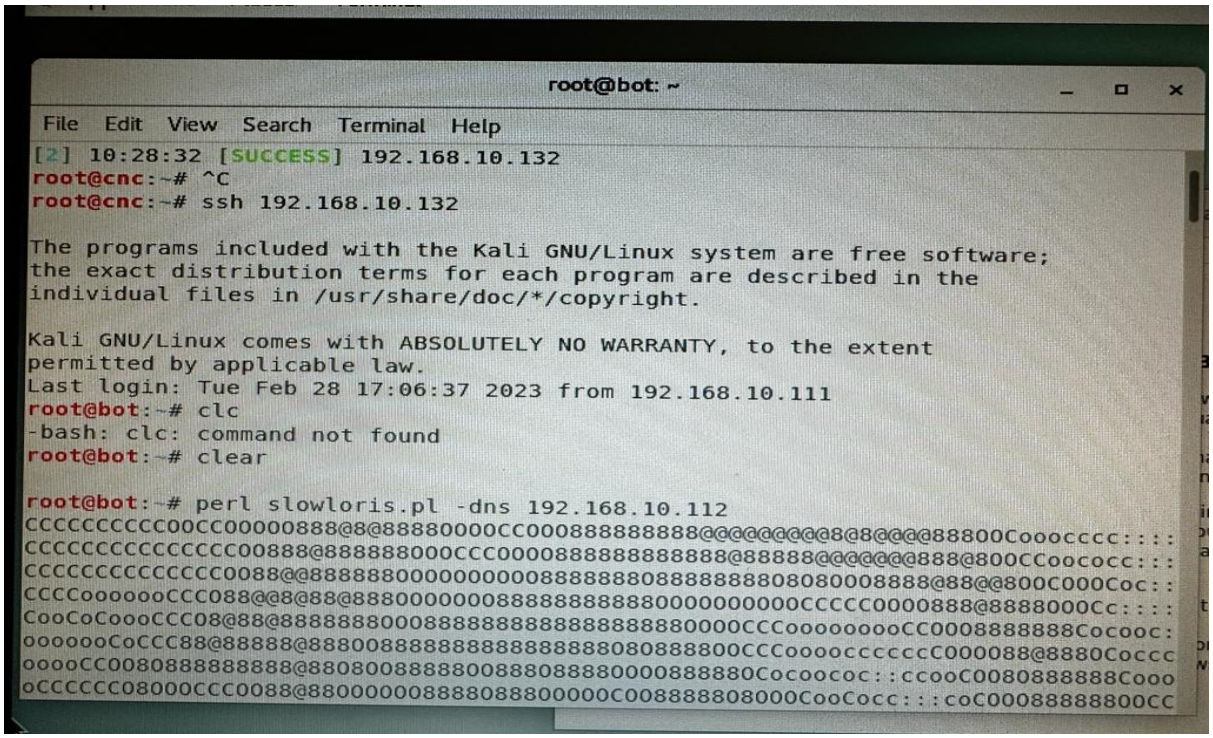


Figure 32: started slow http attack

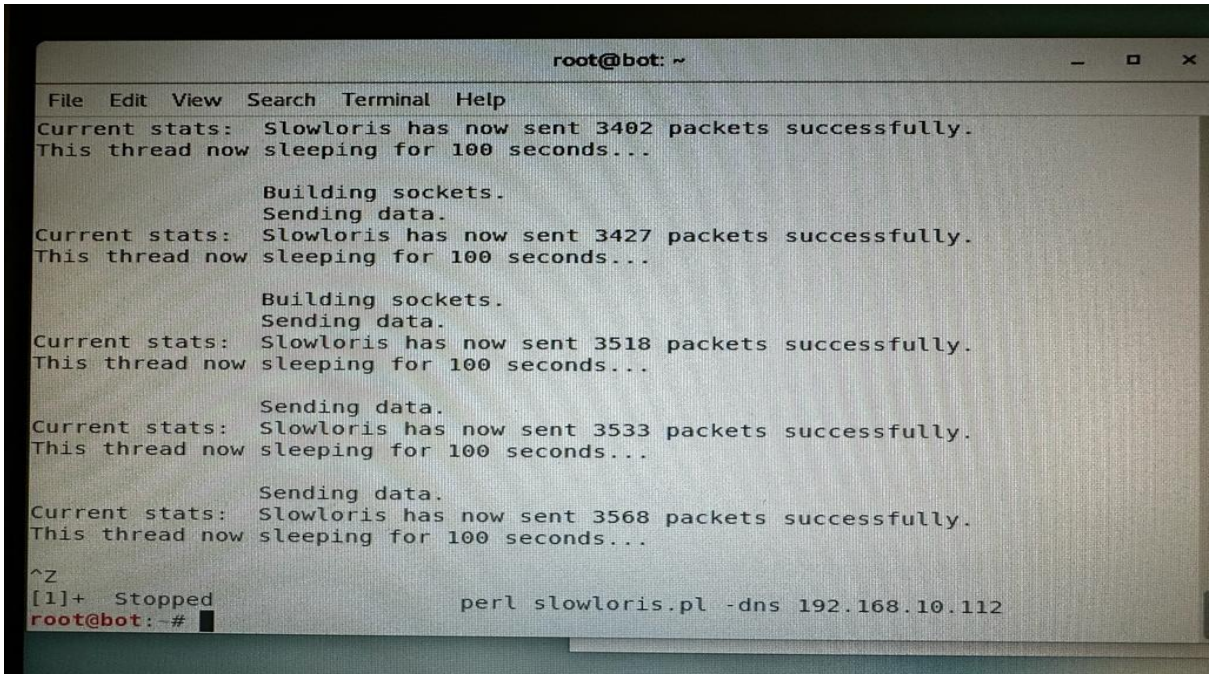


Figure 33: slow http attack

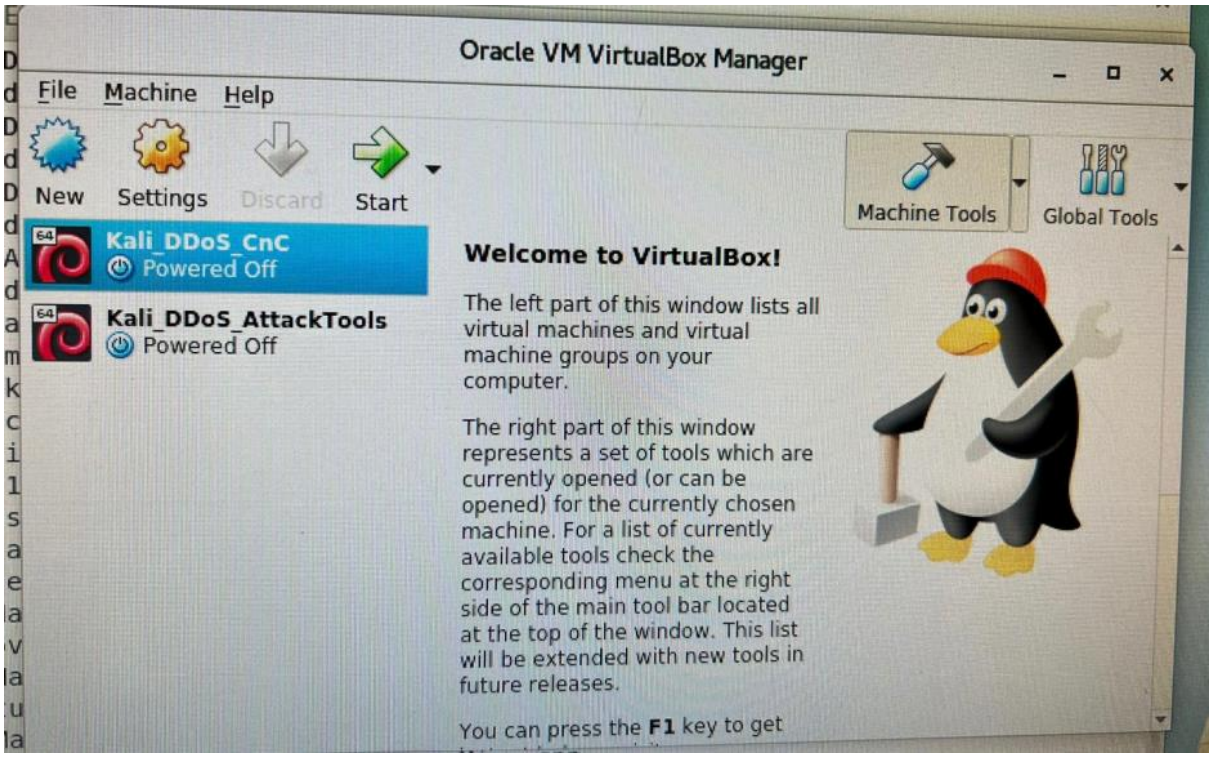


Figure 34: imported tools to VM

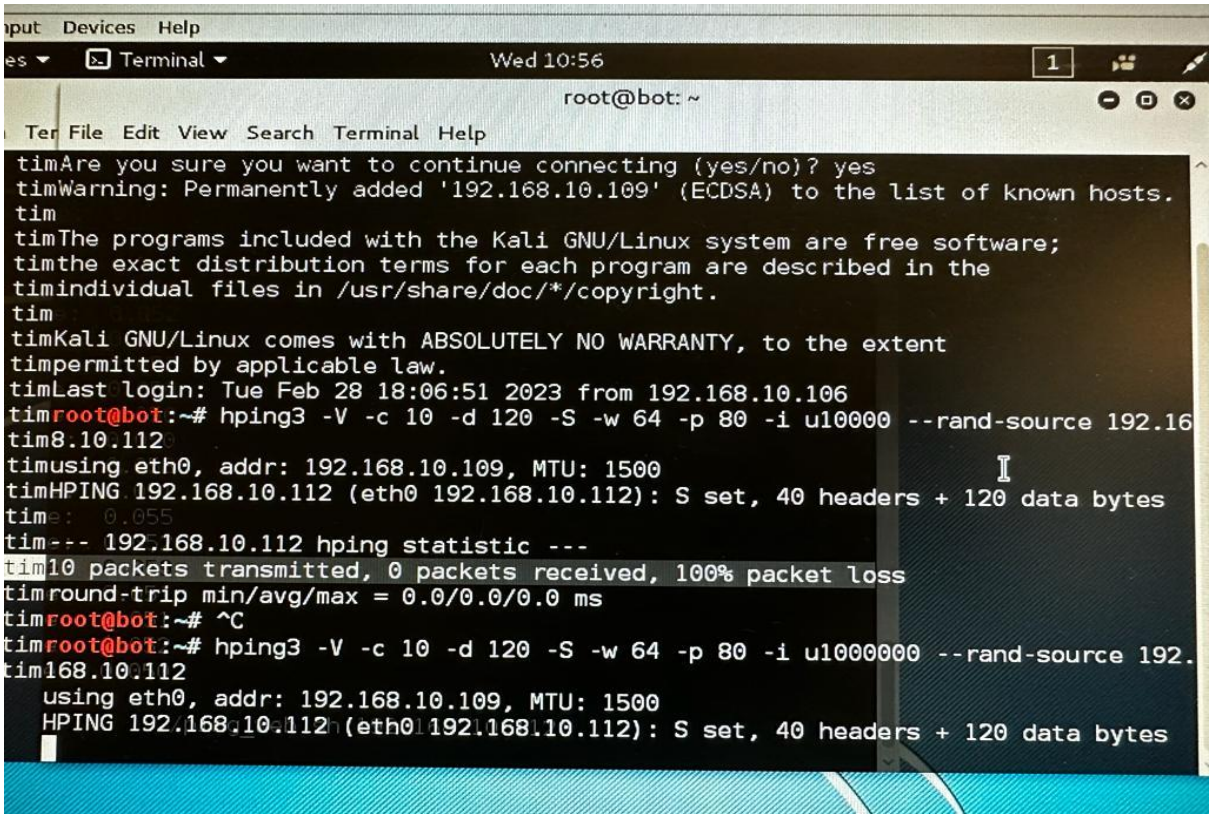


Figure 35: Did the hping attack

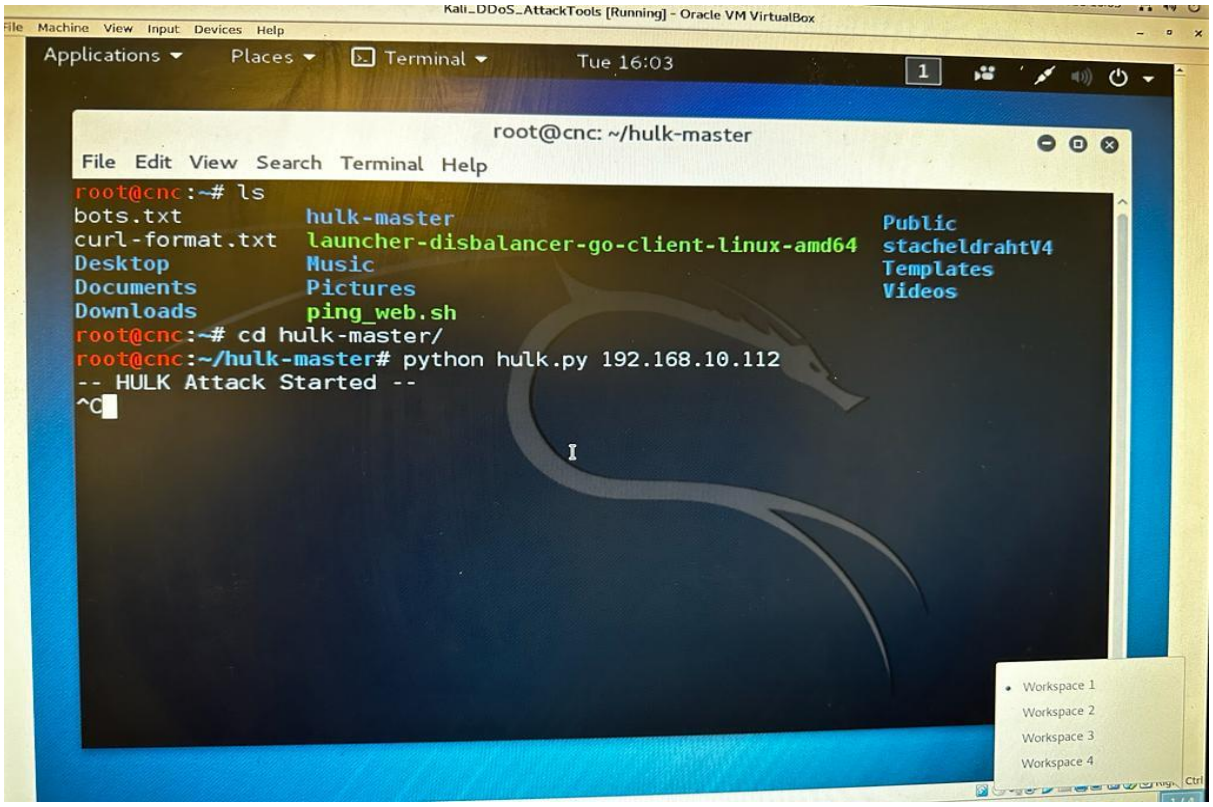


Figure 36: started the hulk attack

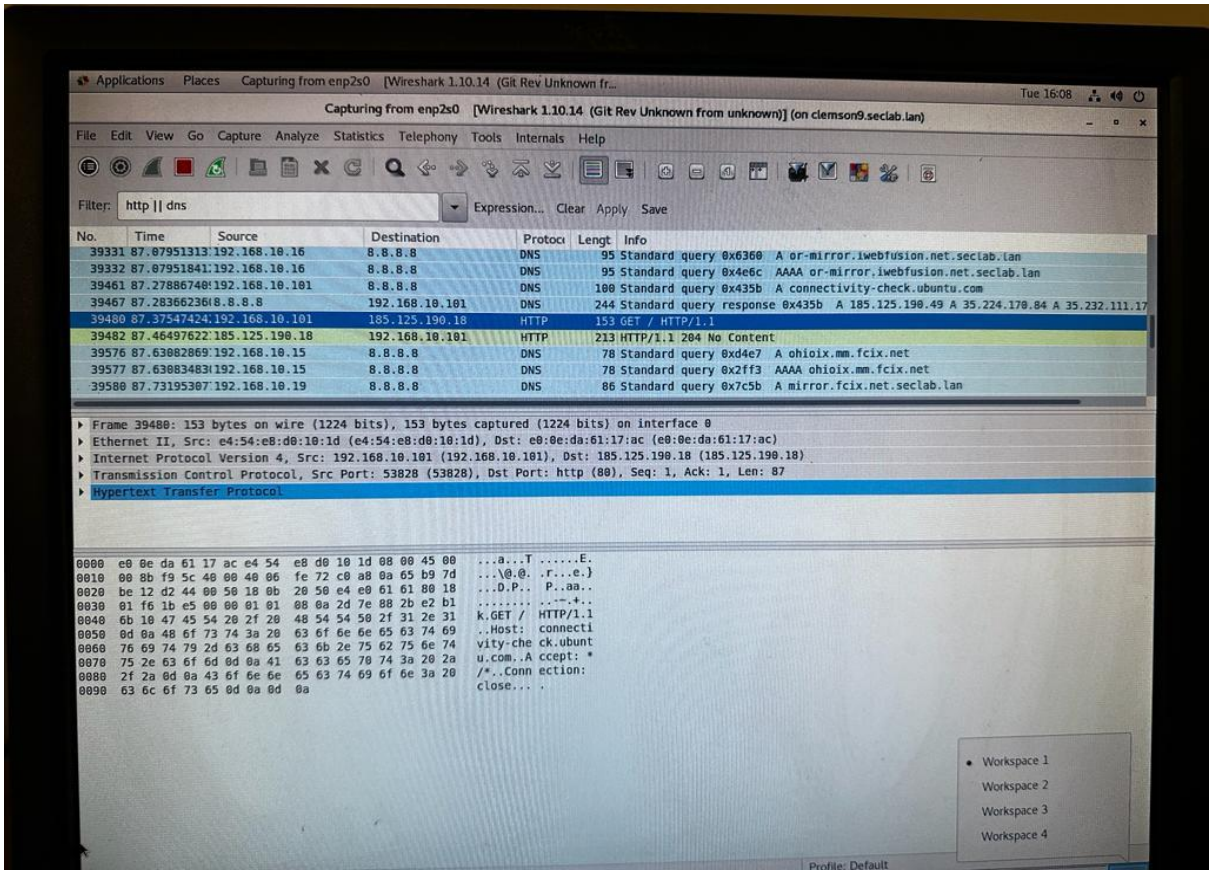


Figure 37: Http attack on victim machine in wireshark.

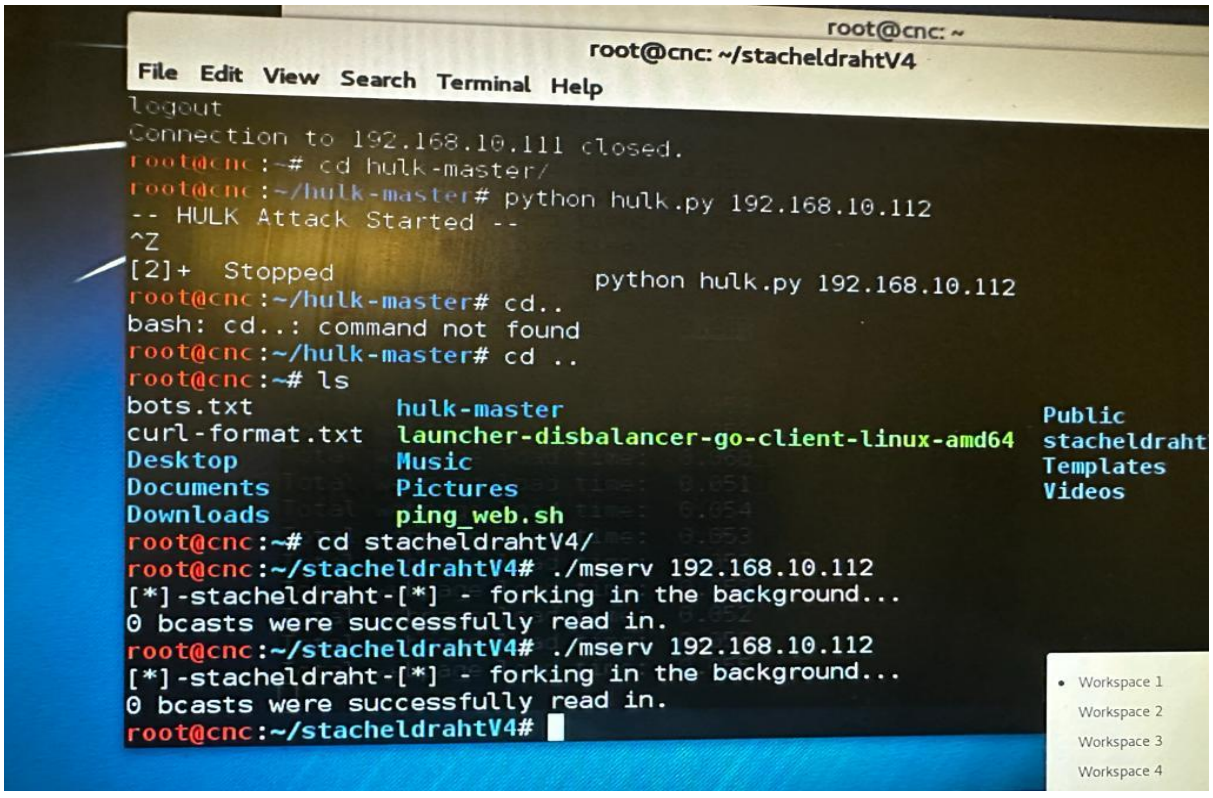


Figure 37: stacheldrahtV4 attack

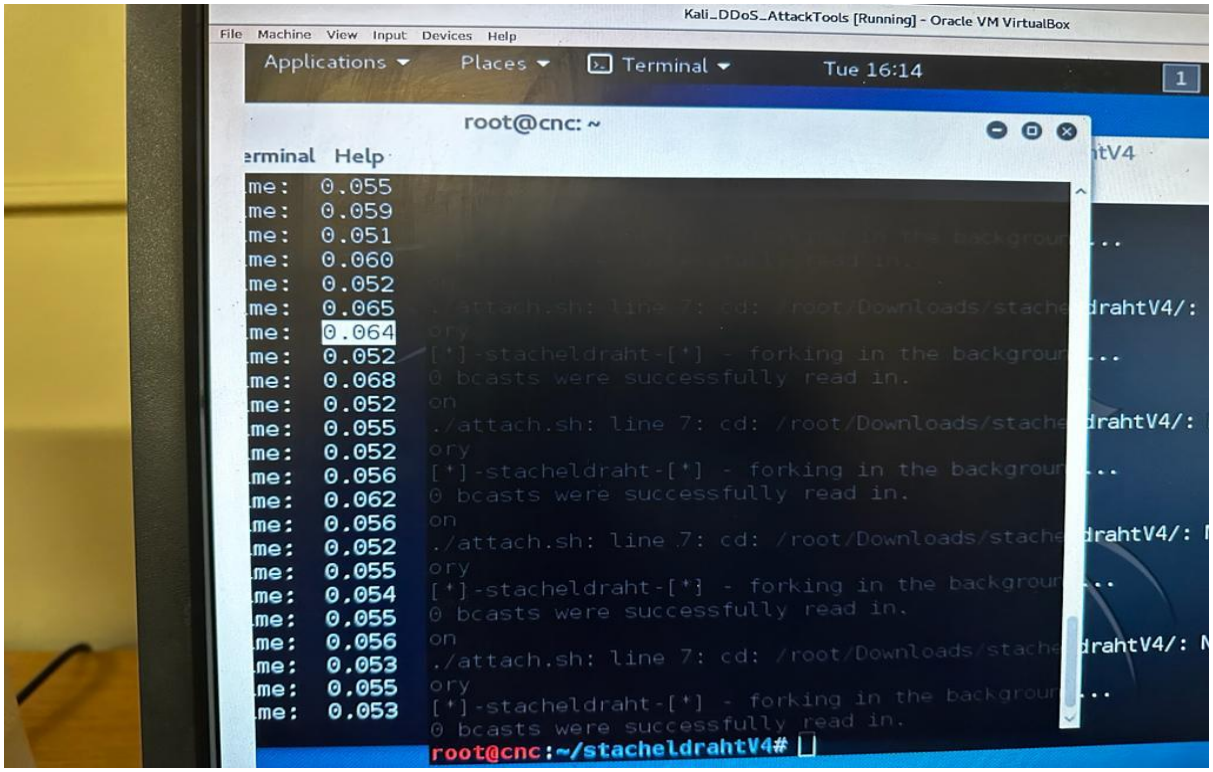


Figure 38: web page load time after attack

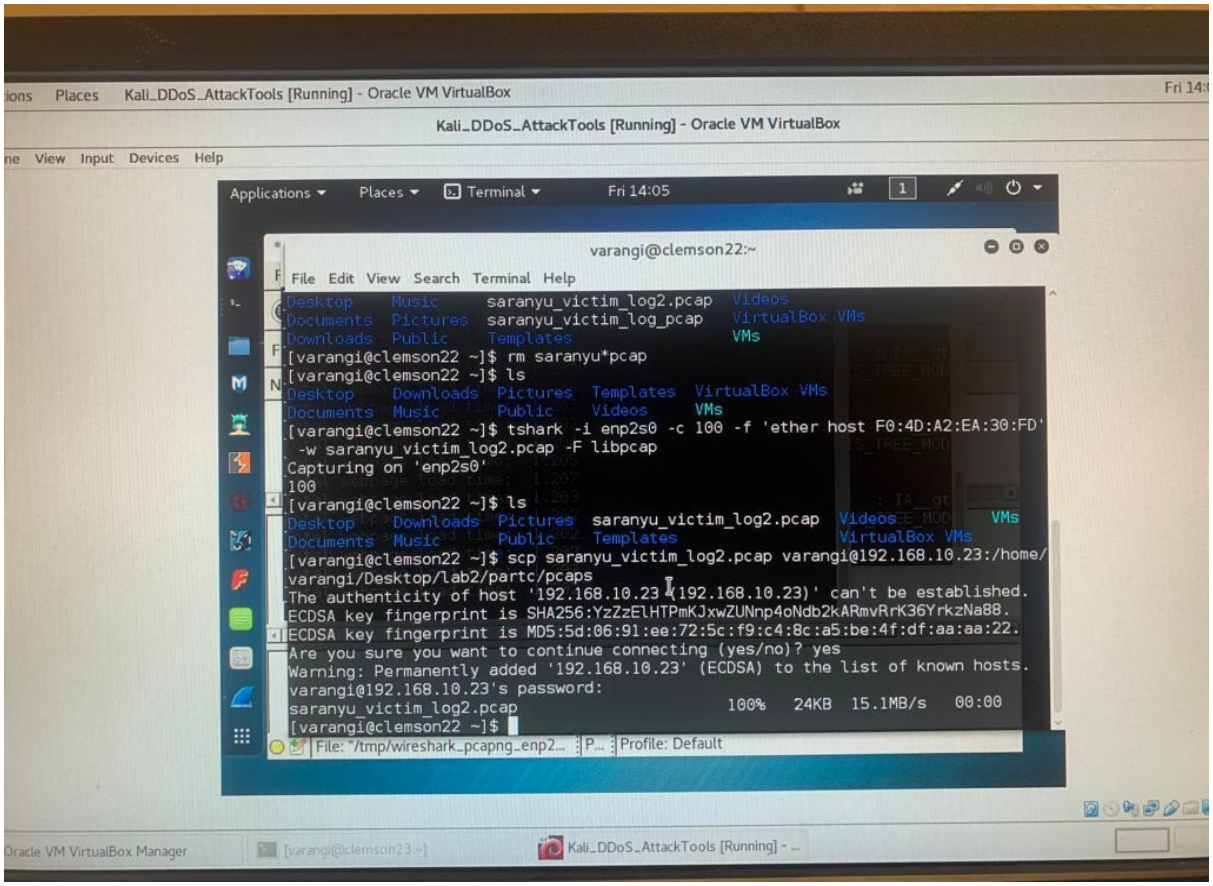


Figure 39: DNS amplification attack file

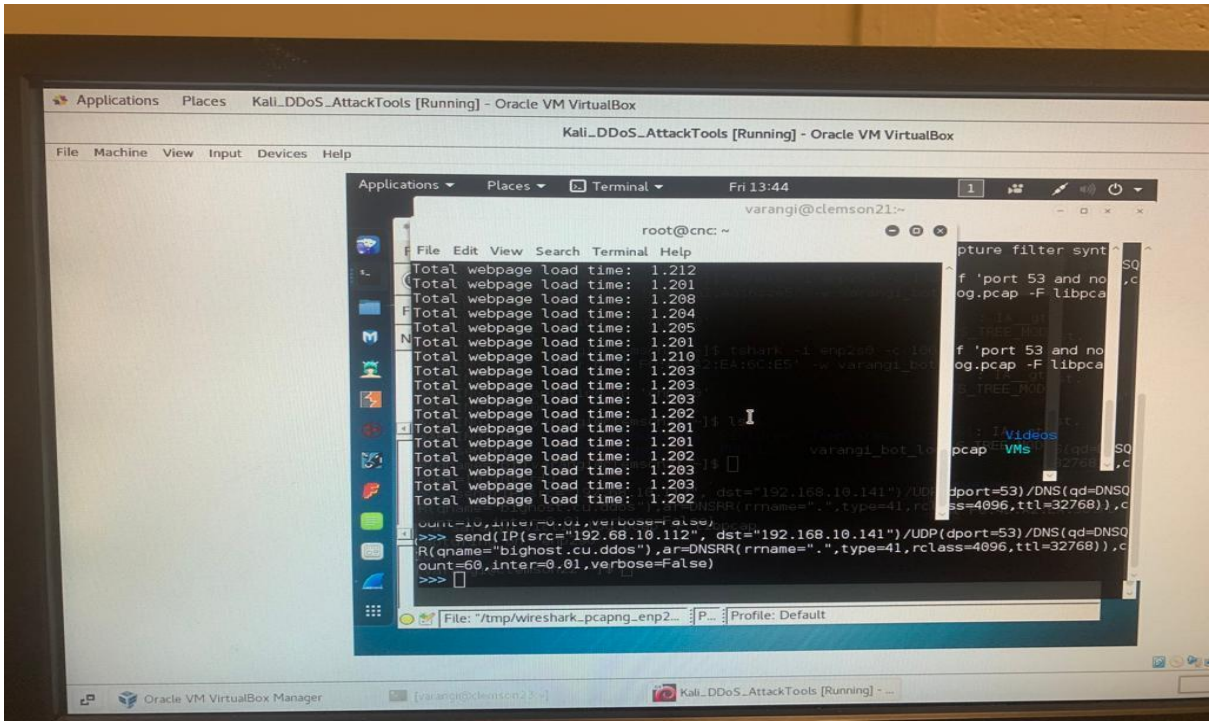


Figure 40: load time after attack

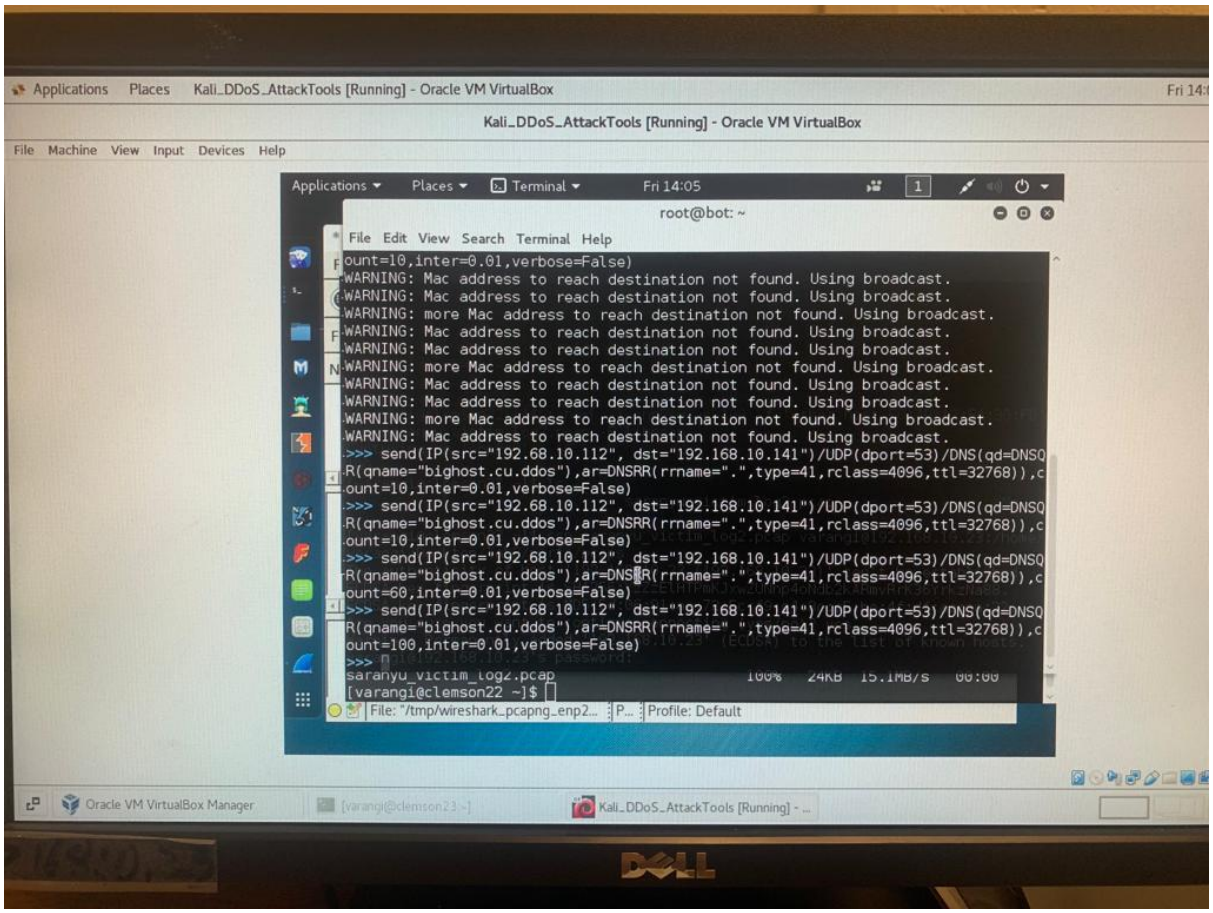


Figure 41: Ran the tshark command to read the traffic file

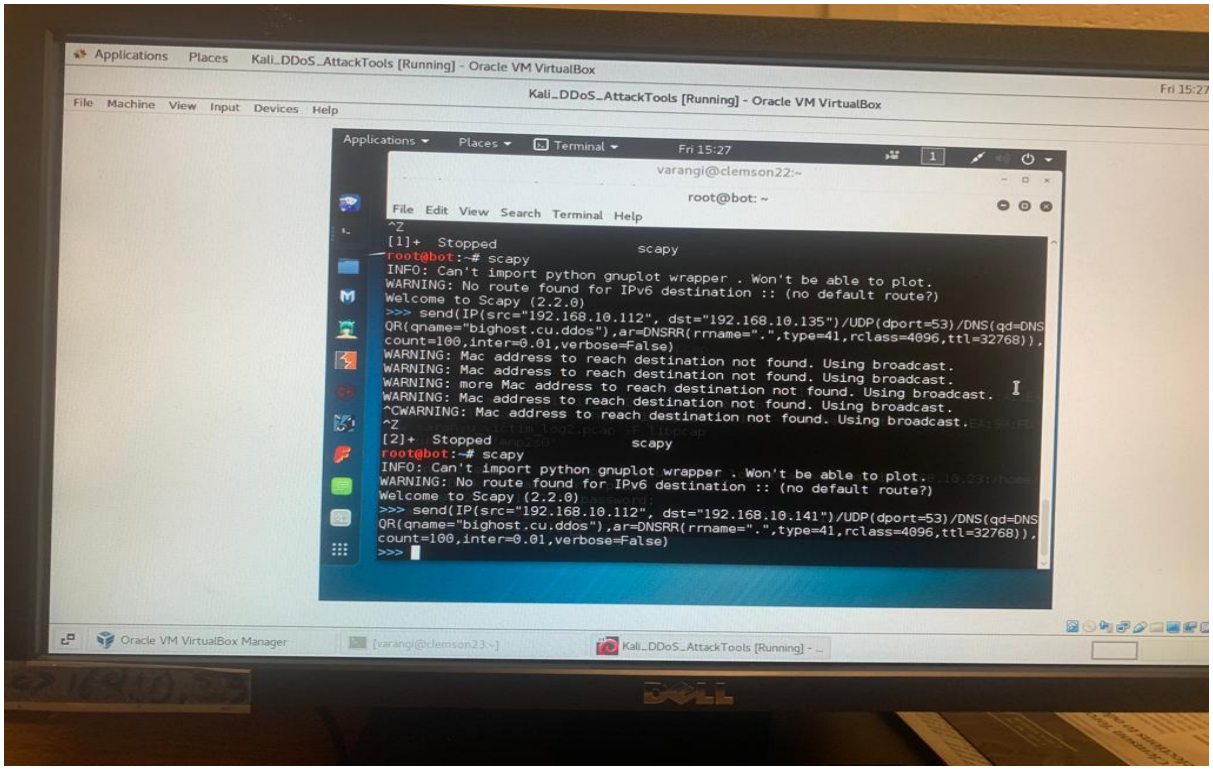


Figure 42: cmd proof for DNS amplification

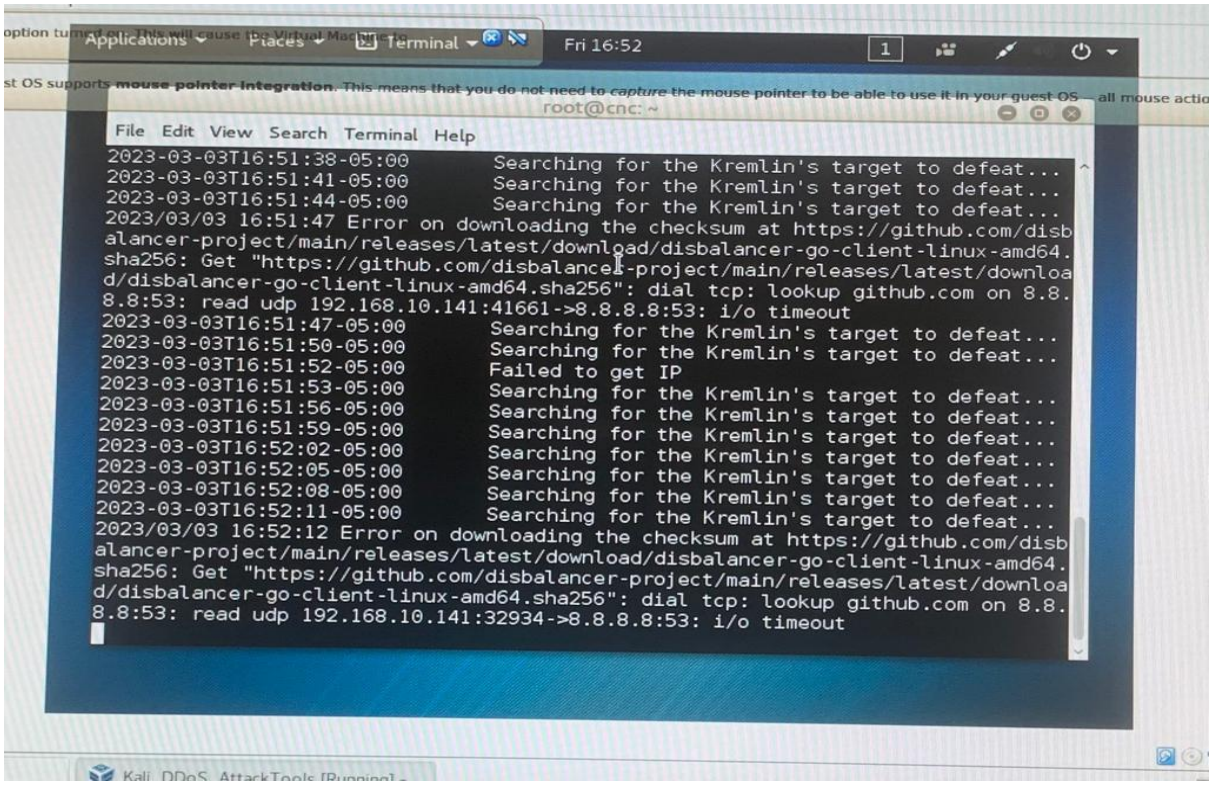


Figure 43: Captured Russia cyber-attack with Disbalancer

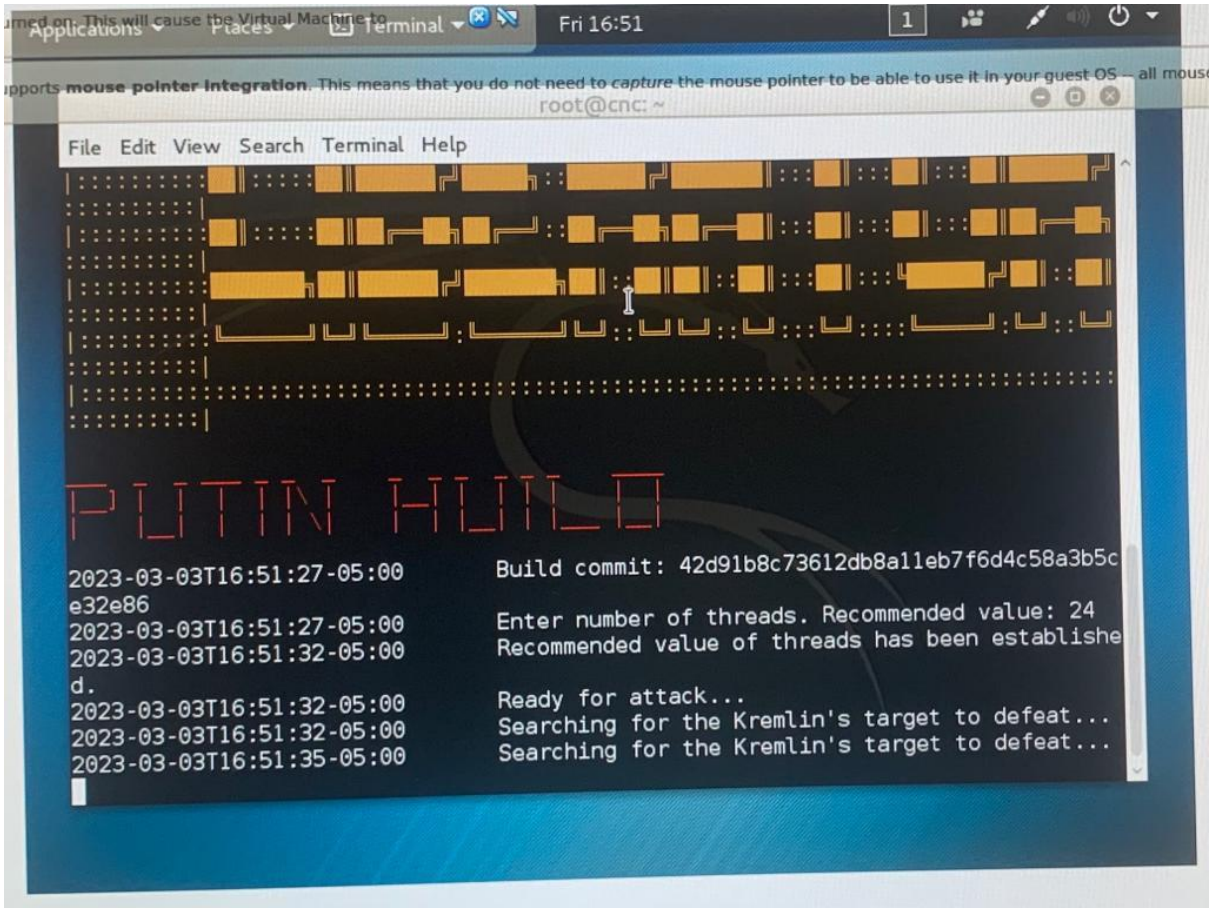


Figure 44: Captured Russia cyber-attack with Disbalancer

Conclusion:

As a result, network background traffic is crucial for detecting DDoS attacks since it provides the groundwork for detection. Both signature-based and anomaly-based detection methods use background traffic to look for recognizable patterns or sudden changes in the observed data. When access to a live network is not always feasible for testing, packet traces or simulated network background traffic may be used. In a lab setting, four different DDoS attacks were evaluated to determine how they affected the target websites, with Slowloris having the most impact and Stacheldraht having zero impact. I learned how to launch numerous DDoS and flood attacks during the trial, as well as how to monitor network data. There are some things to look out for or exploit in order to stop successful packet transmission between devices.

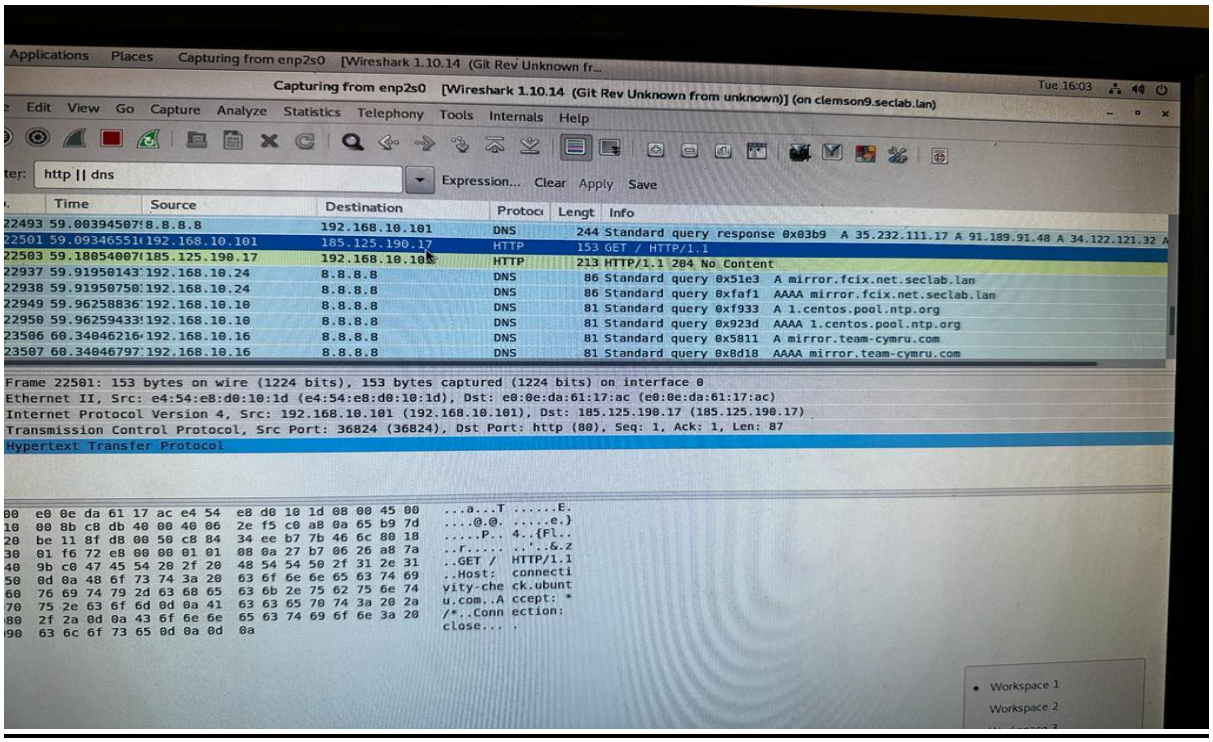
Collaboration:

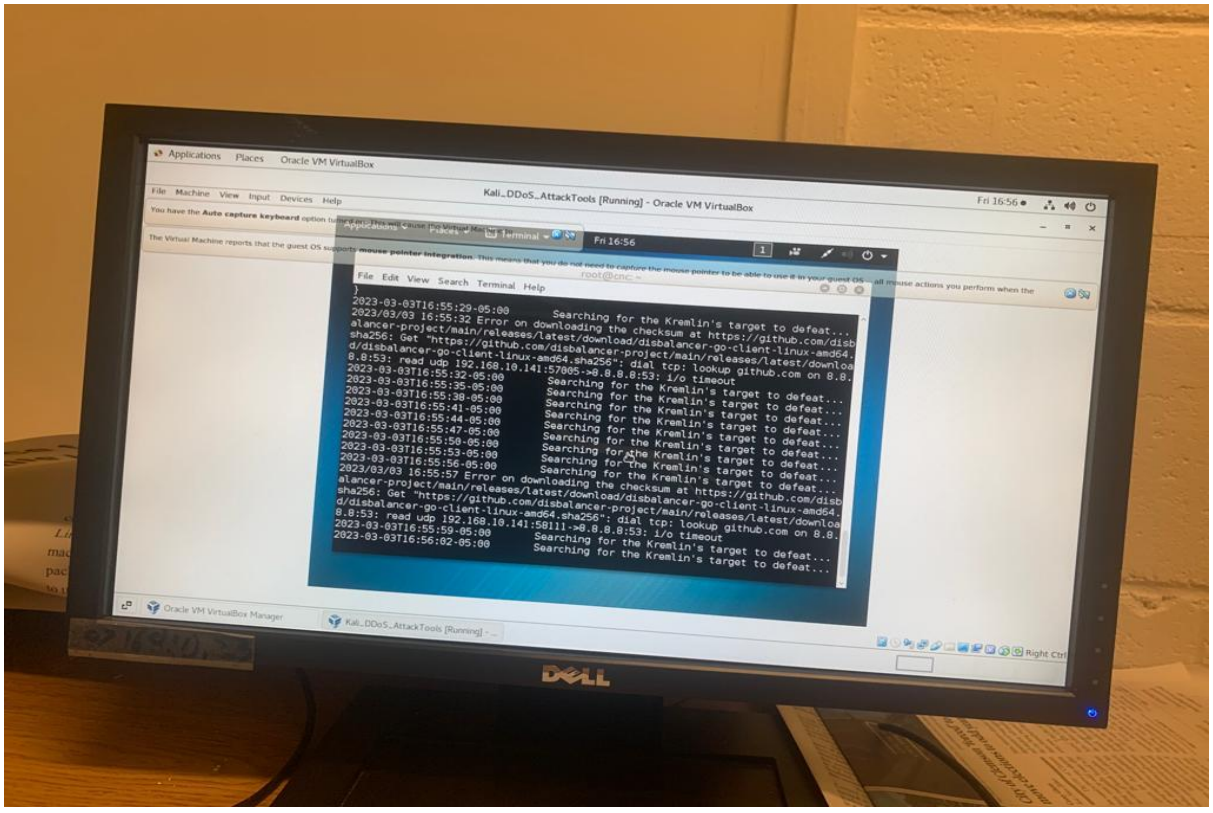
I collaborated with Manasa Thatipamula and Saranyu Arangi to do these lab experiments.

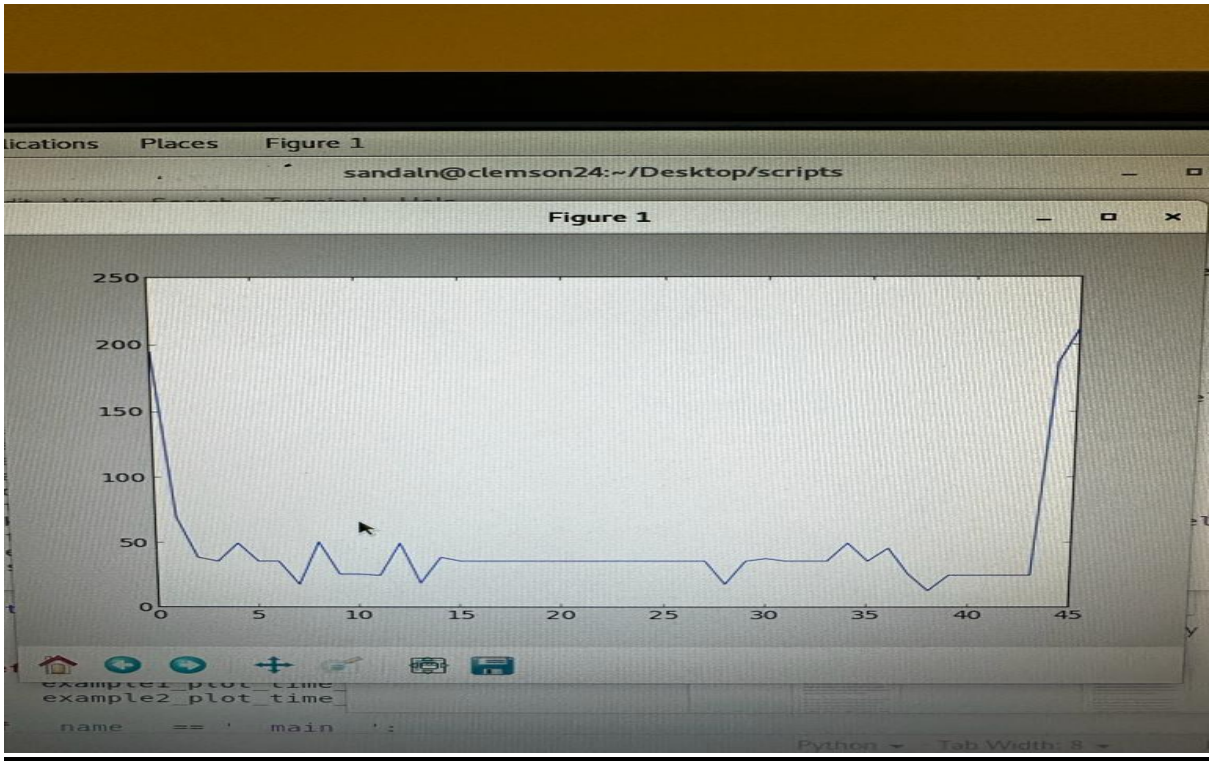
References:

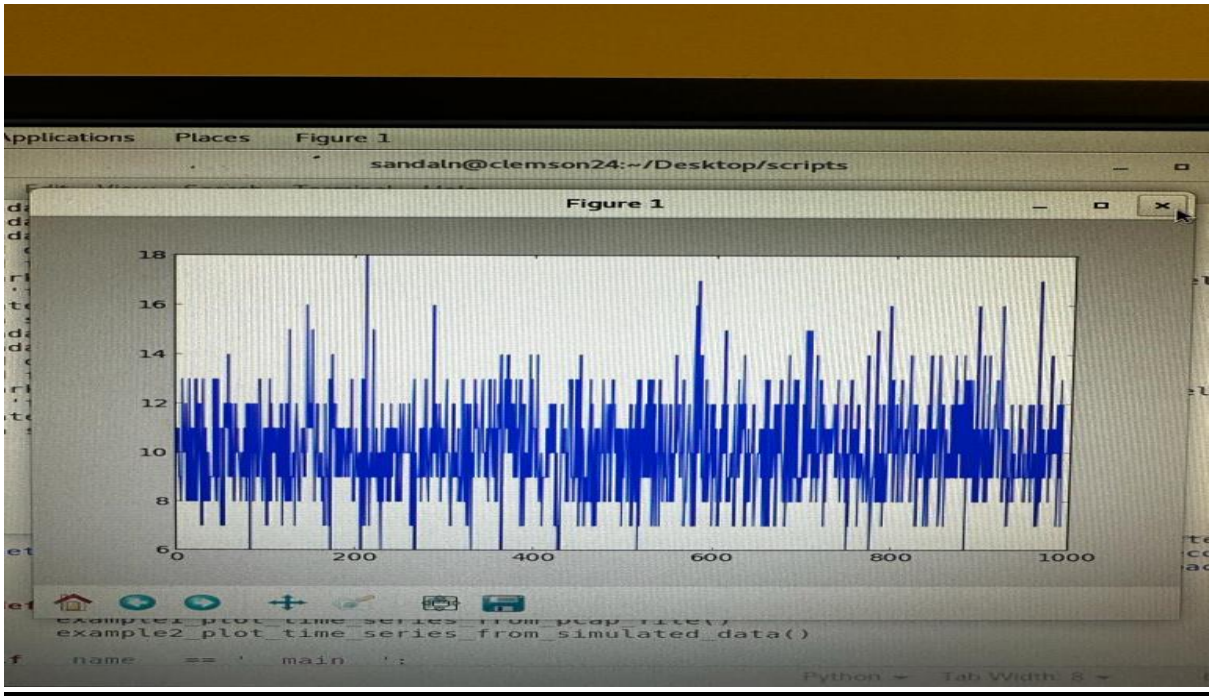
https://clemson.instructure.com/files/15917557/download?download_frd=1
https://clemson.instructure.com/files/15997385/download?download_frd=1
https://clemson.instructure.com/files/15986259/download?download_frd=1
https://clemson.instructure.com/files/15819731/download?download_frd=1
https://clemson.instructure.com/files/15986258/download?download_frd=1
https://clemson.instructure.com/files/15671198/download?download_frd=1

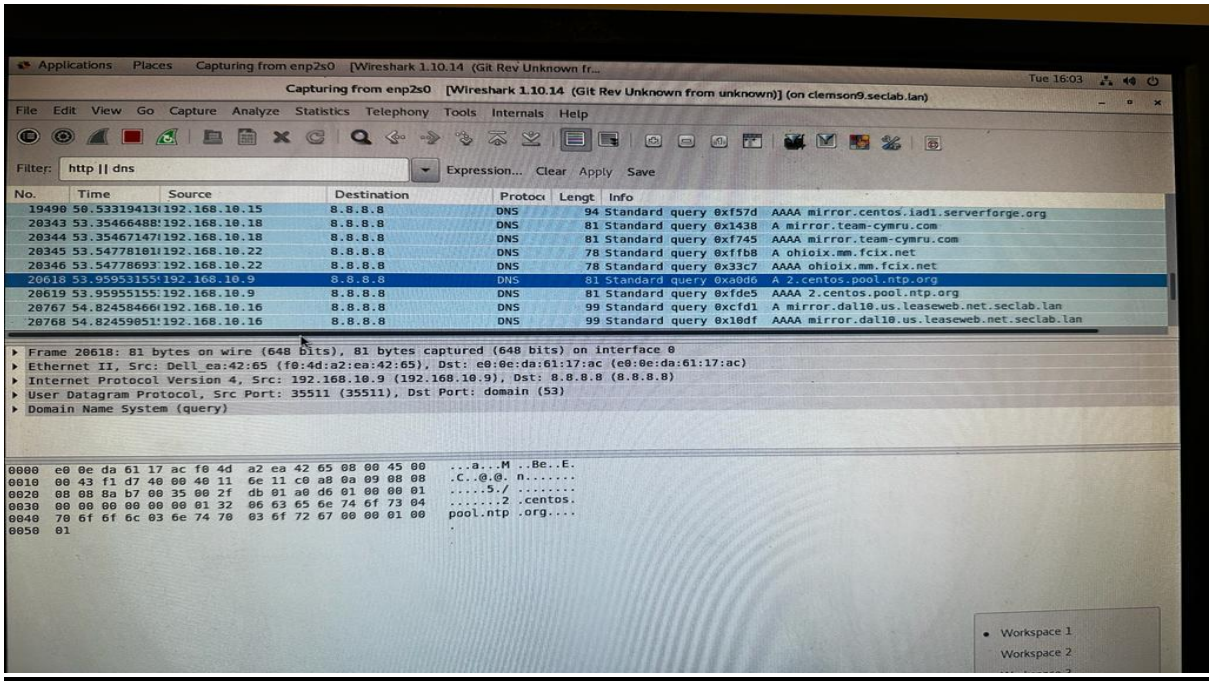
Appendices:

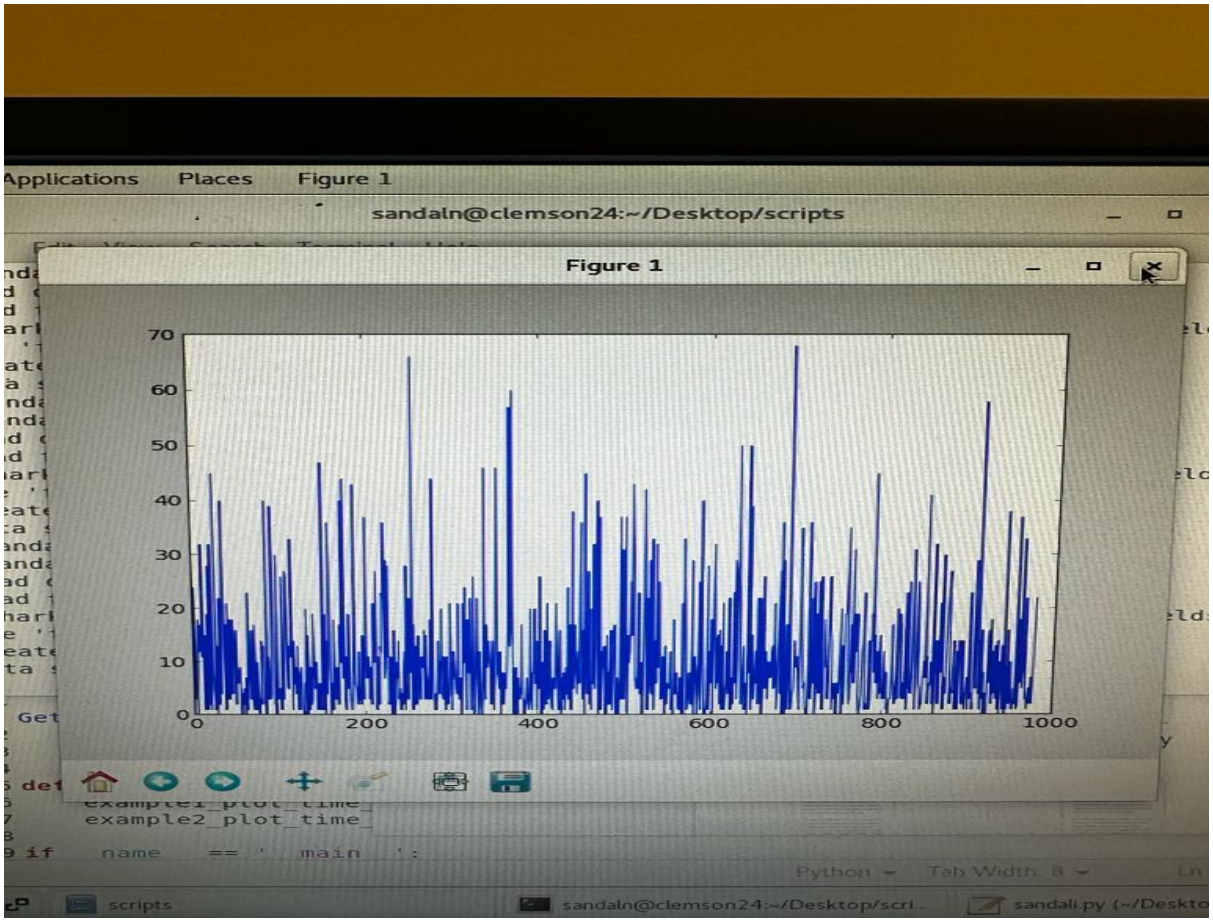

Applications Places Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown fr...
Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown from unknown)] (on clemson9.seclab.lan)
Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
http || dns
Expression... Clear Apply Save
Time Source Destination Protoc Lengt Info
22493 59.00394507 8.8.8.8 192.168.10.101 DNS 244 Standard query response 0x03b9 A 35.232.111.17 A 91.189.91.48 A 34.122.121.32 A
22501 59.09346551 192.168.10.101 185.125.190.17 HTTP 153 GET / HTTP/1.1
22503 59.18054007 185.125.190.17 192.168.10.101 HTTP 213 HTTP/1.1 204 No Content
22937 59.91950143 192.168.10.24 8.8.8.8 DNS 86 Standard query 0x51e3 A mirror.fcix.net.seclab.lan
22938 59.91950750 192.168.10.24 8.8.8.8 DNS 86 Standard query 0xfaf1 AAAA mirror.fcix.net.seclab.lan
22949 59.96258836 192.168.10.10 8.8.8.8 DNS 81 Standard query 0xf933 A 1.centos.pool.ntp.org
22950 59.96259433 192.168.10.10 8.8.8.8 DNS 81 Standard query 0x923d AAAA 1.centos.pool.ntp.org
23506 60.34046216 192.168.10.16 8.8.8.8 DNS 81 Standard query 0x5811 A mirror.team-cymru.com
23507 60.34046797 192.168.10.16 8.8.8.8 DNS 81 Standard query 0x8d18 AAAA mirror.team-cymru.com
Frame 22501: 153 bytes on wire (1224 bits), 153 bytes captured (1224 bits) on interface 0
Ethernet II, Src: e4:54:e8:d0:10:1d (e4:54:e8:d0:10:1d), Dst: e0:0e:da:61:17:ac (e0:0e:da:61:17:ac)
Internet Protocol Version 4, Src: 192.168.10.101 (192.168.10.101), Dst: 185.125.190.17 (185.125.190.17)
Transmission Control Protocol, Src Port: 36824 (36824), Dst Port: http (80), Seq: 1, Ack: 1, Len: 87
Hypertext Transfer Protocol
Workspace 1
Workspace 2

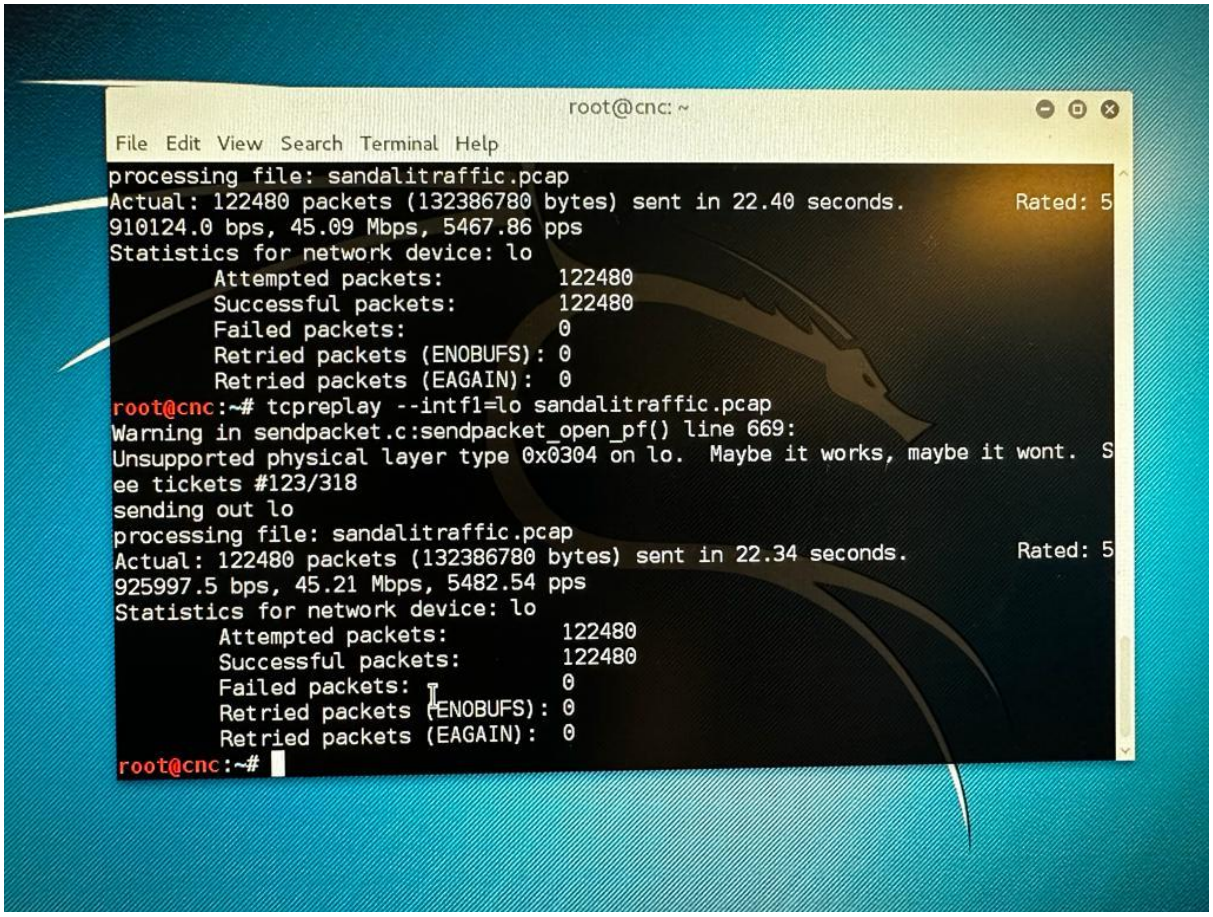

Applications Places Oracle VM VirtualBox
Kali_DDoS_AttackTools [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Fri 16:56
File Edit View Search Terminal Help
Oracle VM VirtualBox Manager
Kali_DDoS_AttackTools [Running] -
DELL

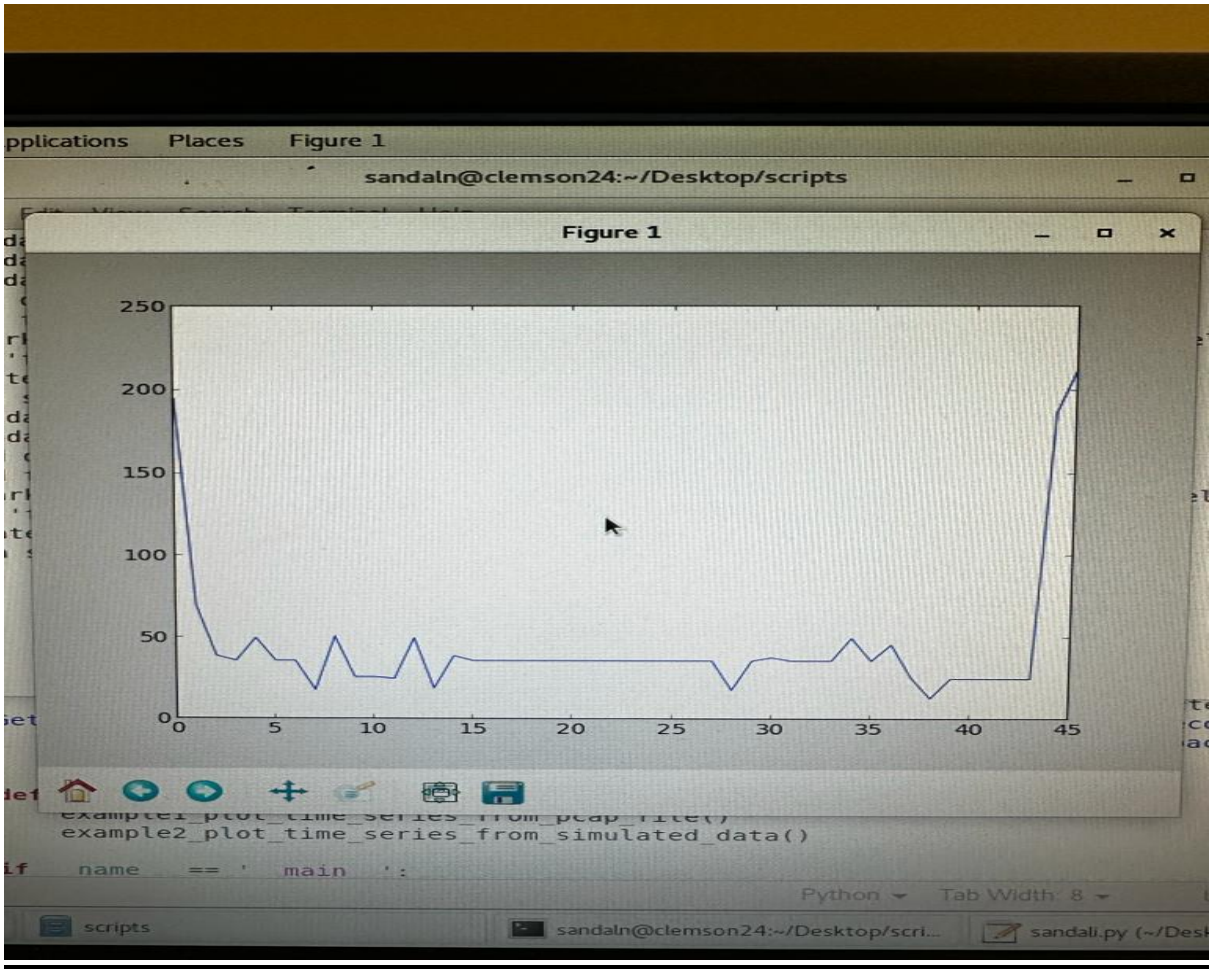

Places
Figure 1
sandaln@clemson24:~/Desktop/scripts
Figure 1
250
200
150
100
50
0
0
5
10
15
20
25
30
35
40
45
example2_plot_time_
name == ' main ':
Python
Tab Width: 8


Applications
Places
Figure 1
sandaln@clemson24:~/Desktop/scripts
Figure 1
18
16
14
12
10
8
6
0
200
400
600
800
1000
example2_plot_time_series_from_simulated_data()
name == ' main ':
Python
Tab Width: 8


Applications Places Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown fr...
Tue 16:03
Capturing from enp2s0 [Wireshark 1.10.14 (Git Rev Unknown from unknown)] (on clemson9.seclab.lan)
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: http || dns
Expression... Clear Apply Save
No. Time Source Destination Protoc Lengt Info
19490 50.53319413 192.168.10.15 8.8.8.8 DNS 94 Standard query 0xf57d AAAA mirror.centos.iad1.serverforge.org
20343 53.35466488 192.168.10.18 8.8.8.8 DNS 81 Standard query 0x1438 A mirror.team-cymru.com
20344 53.35467147 192.168.10.18 8.8.8.8 DNS 81 Standard query 0xf745 AAAA mirror.team-cymru.com
20345 53.54778101 192.168.10.22 8.8.8.8 DNS 78 Standard query 0xffb8 A ohioix.mm.fcix.net
20346 53.54778693 192.168.10.22 8.8.8.8 DNS 78 Standard query 0x33c7 AAAA ohioix.mm.fcix.net
20618 53.95953155 192.168.10.9 8.8.8.8 DNS 81 Standard query 0xa0d6 A 2.centos.pool.ntp.org
20619 53.95955155 192.168.10.9 8.8.8.8 DNS 81 Standard query 0xfde5 AAAA 2.centos.pool.ntp.org
20767 54.82458466 192.168.10.16 8.8.8.8 DNS 99 Standard query 0xcfd1 A mirror.dal10.us.leaseweb.net.seclab.lan
20768 54.82459051 192.168.10.16 8.8.8.8 DNS 99 Standard query 0x10df AAAA mirror.dal10.us.leaseweb.net.seclab.lan
Frame 20618: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
Ethernet II, Src: Dell_ea:42:65 (f0:4d:a2:ea:42:65), Dst: e0:0e:da:61:17:ac (e0:0e:da:61:17:ac)
Internet Protocol Version 4, Src: 192.168.10.9 (192.168.10.9), Dst: 8.8.8.8 (8.8.8.8)
User Datagram Protocol, Src Port: 35511 (35511), Dst Port: domain (53)
Domain Name System (query)
0000 e0 0e da 61 17 ac f0 4d a2 ea 42 65 08 00 45 00 ...a...M ..Be..E.
0010 00 43 f1 d7 40 00 40 11 6e 11 c0 a8 0a 09 08 08 .C..@.@. n.......
0020 08 08 8a b7 00 35 00 2f db 01 a0 d6 01 00 00 01 .....5./ ....
0030 00 00 00 00 00 00 01 32 06 63 65 6e 74 6f 73 04 ....2 .centos.
0040 70 6f 6f 6c 03 6e 74 70 03 6f 72 67 00 00 01 00 pool.ntp .org....
0050 01 .
Workspace 1
Workspace 2


Applications Places Figure 1
sandaln@clemson24:~/Desktop/scripts
Figure 1
70
60
50
40
30
20
10
0
0
200
400
600
800
1000
example1_plot_time
example2_plot_time
if __name__ == '__main__':
Python Tab Width: 8
scripts
sandaln@clemson24:~/Desktop/scri...
sandali.py (~/Deskto


root@cnc: ~
File Edit View Search Terminal Help
processing file: sandalitraffic.pcap
Actual: 122480 packets (132386780 bytes) sent in 22.40 seconds. Rated: 5
910124.0 bps, 45.09 Mbps, 5467.86 pps
Statistics for network device: lo
Attempted packets: 122480
Successful packets: 122480
Failed packets: 0
Retried packets (ENOBUFS): 0
Retried packets (EAGAIN): 0
root@cnc:~# tcpreplay --intf1=lo sandalitraffic.pcap
Warning in sendpacket.c:sendpacket_open_pf() line 669:
Unsupported physical layer type 0x0304 on lo. Maybe it works, maybe it wont. S
ee tickets #123/318
sending out lo
processing file: sandalitraffic.pcap
Actual: 122480 packets (132386780 bytes) sent in 22.34 seconds. Rated: 5
925997.5 bps, 45.21 Mbps, 5482.54 pps
Statistics for network device: lo
Attempted packets: 122480
Successful packets: 122480
Failed packets: 0
Retried packets (ENOBUFS): 0
Retried packets (EAGAIN): 0
root@cnc:~#


pplications
Places
Figure 1
sandaln@clemson24:~/Desktop/scripts
Figure 1
250
200
150
100
50
0
0
5
10
15
20
25
30
35
40
45
example2_plot_time_series_from_simulated_data()
if __name__ == '__main__':
Python
Tab Width: 8
scripts
sandaln@clemson24:~/Desktop/scri...
sandali.py (~/Des