CLEMSON
UNIVERSITY

ECE 8860: Distributed Denial of Service Attacks

Spring 2023

Laboratory 3

Name: Sandali Sanjay Nemmaniwar

Email: sandaln@g.clemson.edu

Major: Computer Science

Abstract

This report discusses a study that explains about DDoS attack detection. We will be using the provided scripts to learn the DDoS detection methods with hands on experience. Due to the fact that some of the algorithms, e.g. wavelet, requires hours of data, and it takes a lot of time to read pcap files for entropy detection. During the lab, we will mainly focus on detection analysis with provided data. The detections methods used are Traffic Volume Detection, Wavelet Detection, CUSUM Detection, Wave CUSUM detection, and Entropy based detection.

Introduction:

The study employs five different detection methods, each with its unique approach to detecting DDoS attacks. Traffic Volume Detection focuses on analysing the network traffic volume to detect any sudden spikes in the volume of traffic, which is a common indication of DDoS attacks. Wavelet Detection is another method that uses wavelet transforms to analyse network traffic and identify any anomalies or irregularities that might indicate an attack. CUSUM Detection, on the other hand, utilizes a statistical technique called CUSUM to detect any changes in the network traffic pattern that might indicate an attack. Wave CUSUM Detection combines the wavelet and CUSUM methods to detect attacks by analysing both the frequency and magnitude of network traffic. Lastly, Entropy-based Detection uses entropy analysis to detect any changes in the randomness of network traffic.

Although the scripts provided allow for hands-on experience with each of the detection methods, some algorithms, such as wavelet, require hours of data to produce results, and reading pcap files for entropy detection can be a time-consuming process. Therefore, the lab will mainly focus on detection analysis with the provided data rather than running each detection method from scratch. This approach will allow for a more in-depth analysis of each method's strengths and weaknesses without sacrificing the time required to run the algorithms.

Methodology:

The data analysis is described below:

1. ‘timeseries.txt’ is the file contains 24 hours of time series data.
2. ‘outputTime0604.entr’is the entropy file generated from the pcap files captured during the same time period.
3. ‘Complete attack times’ records the attack start/endtimes.
4. ‘plot.py’ is a python script used to plot data and compare with attack records with different thresholds. Use ‘python plot.py -h’ for help.
5. ‘vda.py’ is a python script used to perform CUSUM, Wavelet, and Entropy analysis. The input of ‘vda.py’ is the file that contains the time series data (e.g., ‘timeseries.txt’). The output file that contains data produced with selected algorithm and defined parameters. Use ‘python vda.py -h’ for help.

Data Analysis:

In the lab, graphs are an essential tool for collecting data and analyzing the results of the DDoS attack detection methods. The "Plot.py" script is utilized to generate these graphs. On the other hand, the "vda.py" script is employed to perform various analyses, such as CUSUM, Wavelet, and Entropy analyses, and to produce output files that can be plotted in line with the results. To aid in the detection of DDoS attacks at times, threshold values are also computed using "plot.py," which draws a line on the graph at the specified threshold value. These lines assist in identifying when the network traffic surpasses the threshold and thus helps to detect a potential DDoS attack. Overall, the combination of graphing and threshold values helps to simplify the process of detecting and analyzing DDoS attacks in the lab setting.

Questions:

1. What detection methods work well?

   Ans: The experiment found that the best techniques for detecting DDoS attacks were data volume thresholding, wavelet analysis, and entropy analysis based on the srcIP column. Data volume thresholding is effective in detecting high-volume attacks, wavelet analysis detects low-frequency attacks, and entropy analysis detects attacks involving spoofed IP addresses. Combining these techniques and implementing them as part of a comprehensive DDoS protection strategy can enhance network security and mitigate the impact of DDoS attacks.

2. How would you try to avoid being detected?

   Ans: There are various methods that attackers may employ to avoid detection during a DDoS attack. One such method is the use of **distributed networks**, where hackers leverage multiple compromised devices, such as botnets, to make it difficult to trace the origin of the attack. This is achieved by using different IP addresses in the attack flow.

Another technique involves using **low and slow attacks**, where attackers send small amounts of traffic over an extended period instead of overwhelming the target network with a large volume of traffic. This type of attack may be harder to identify and less likely to trigger alarms in network monitoring systems.

Attackers may also use **amplification techniques**, such as DNS reflection or NTP amplification attacks, to generate a significant amount of traffic using minimal bandwidth. By making the attack traffic seem to come from legitimate sources, it can be more challenging to identify the attack's origin.

Encryption is another method attackers may use to evade detection and analysis of their attack traffic. By encrypting the traffic, it can be more difficult for network security tools to recognize and block the attack.

Finally, attackers may use **IP address spoofing** to make it appear as if the attack traffic is originating from a trustworthy source. This can make it more challenging to detect and prevent the attack flow.

3. Which method detects more quickly?

Ans: Traffic volume detection is a simple and easy-to-implement technique for detecting DDoS attacks. It involves setting a threshold for the amount of traffic that is deemed normal for a particular network, and if traffic exceeds this threshold, it is flagged as potentially malicious. This technique does not require complex algorithms or sophisticated analysis, making it a straightforward and effective way to detect high-volume DDoS attacks. While it may not be as sophisticated as other detection methods, such as wavelet analysis or entropy-based detection, traffic volume detection is still a valuable tool in a comprehensive DDoS protection strategy. By quickly detecting abnormal traffic patterns and alerting network administrators, traffic volume detection can help prevent DDoS attacks from causing significant damage to a network.

4. How often do you get false alarms from your results?

Ans: DDoS attack detection methods vary in effectiveness and false alarms can be a problem. In the experiment, entropy analysis resulted in a high number of false alarms. This can cause unnecessary disruptions and waste resources. Careful evaluation and adjustment of threshold values can minimize false alarms. A combination of techniques can enhance detection accuracy and mitigate the impact of DDoS attacks.

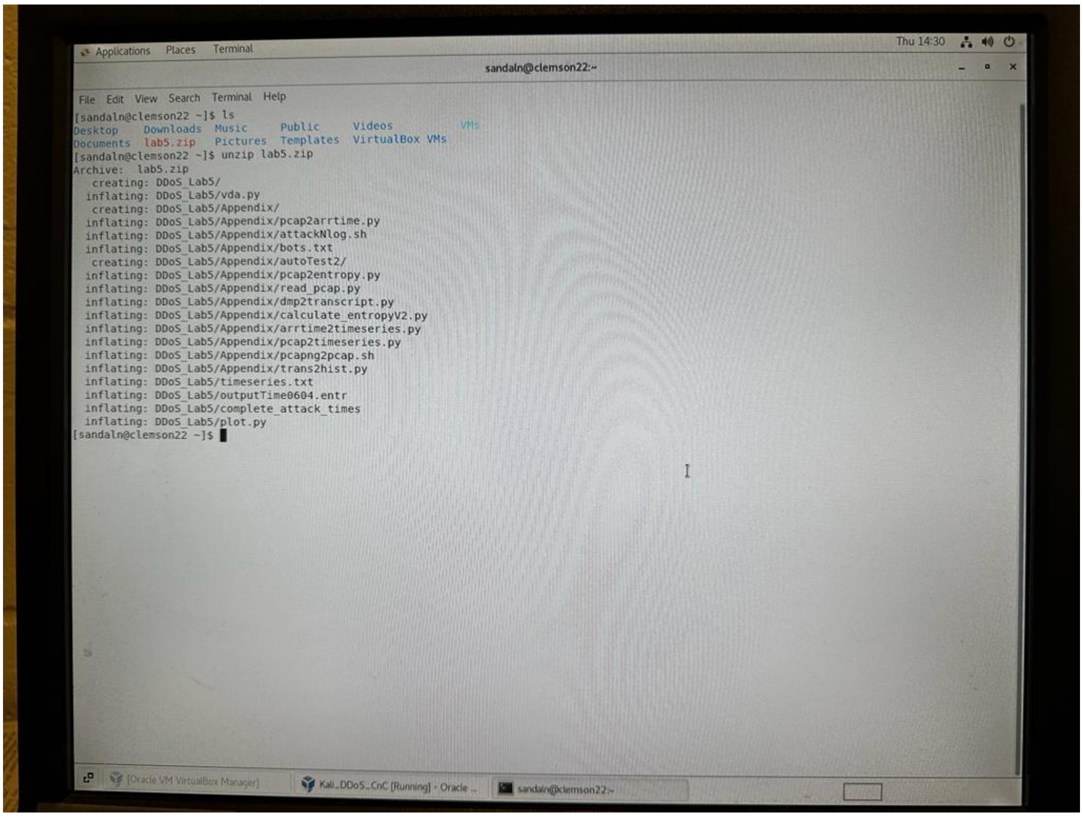
5. A DDoS attack can cost financial losses, reputation damage, customer attrition and even legal pursuits. According to Incapsula Survey: What DDoS Attacks Really Cost Businesses, the cost of a DDoS attack exceeds $20,000 per hour for some companies. How much do you think it will cost a company to analyse false alarms? When do you think DDoS monitoring makes sense?

Ans: The cost of DDoS monitoring is a crucial factor that businesses need to consider in their cybersecurity strategy. The cost is dependent on several factors, such as the size of the business, the complexity of their network infrastructure, the frequency of false alarms, and the level of expertise needed to investigate them. However, the cost of investigating false alarms associated with DDoS monitoring is typically much lower than the cost of a successful DDoS attack.

DDoS monitoring is vital for businesses that rely on internet services, particularly online service providers such as banks, social networks, and e-commerce websites. These businesses are more vulnerable to DDoS attacks due to their dependence on online traffic. Additionally, organizations in sectors such as finance, healthcare, and gaming, which are frequently targeted by DDoS attacks, should prioritize DDoS monitoring.

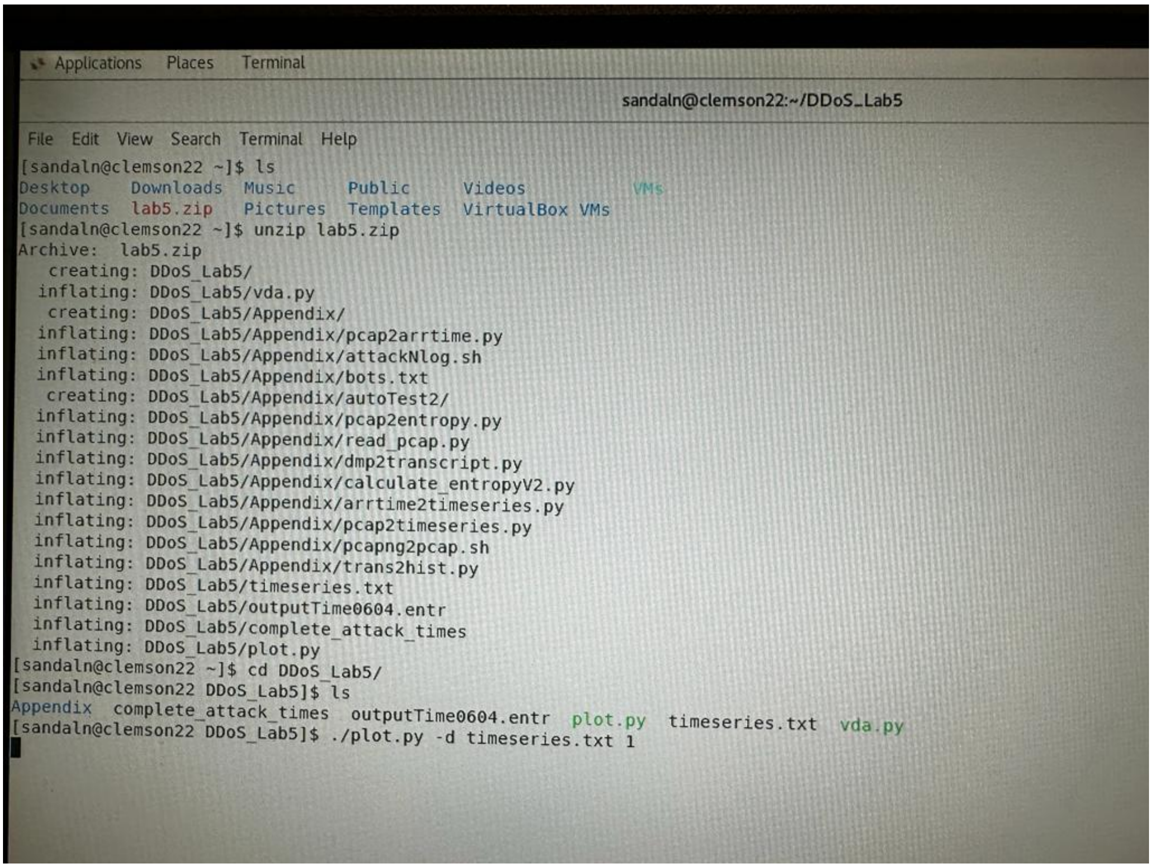
The cost of implementing DDoS monitoring is minor compared to the potential harm to a business's finances and reputation that could result from a successful DDoS attack. DDoS monitoring can help minimize downtime, quickly identify, and mitigate attacks, and protect the business's reputation and clients. Therefore, businesses should consider the cost of DDoS monitoring as a necessary investment in their cybersecurity strategy.

Result



```
[sandaln@clemson22 ~]$ ls
Desktop    Downloads  Music     Public     Videos          VMs
Documents  lab5.zip   Pictures  Templates  VirtualBox VMs
[sandaln@clemson22 ~]$ unzip lab5.zip
Archive:  lab5.zip
   creating: DDoS_Lab5/
  inflating: DDoS_Lab5/vda.py
   creating: DDoS_Lab5/Appendix/
  inflating: DDoS_Lab5/Appendix/pcap2arrtime.py
  inflating: DDoS_Lab5/Appendix/attackNlog.sh
  inflating: DDoS_Lab5/Appendix/bots.txt
   creating: DDoS_Lab5/Appendix/autoTest2/
  inflating: DDoS_Lab5/Appendix/pcap2entropy.py
  inflating: DDoS_Lab5/Appendix/read_pcap.py
  inflating: DDoS_Lab5/Appendix/dmp2transcript.py
  inflating: DDoS_Lab5/Appendix/calculate_entropyV2.py
  inflating: DDoS_Lab5/Appendix/arrtime2timeseries.py
  inflating: DDoS_Lab5/Appendix/pcap2timeseries.py
  inflating: DDoS_Lab5/Appendix/pcapng2pcap.sh
  inflating: DDoS_Lab5/Appendix/trans2hist.py
  inflating: DDoS_Lab5/timeseries.txt
  inflating: DDoS_Lab5/outputTime0604.entr
  inflating: DDoS_Lab5/complete_attack_times
  inflating: DDoS_Lab5/plot.py
[sandaln@clemson22 ~]$ ▋
```

Fig 1: unzipped the required file “lab5.zip”



```
[sandaln@clemson22 ~]$ ls
Desktop    Downloads  Music     Public     Videos         VMs
Documents  lab5.zip   Pictures  Templates  VirtualBox VMs
[sandaln@clemson22 ~]$ unzip lab5.zip
Archive:  lab5.zip
   creating: DDoS_Lab5/
  inflating: DDoS_Lab5/vda.py
   creating: DDoS_Lab5/Appendix/
  inflating: DDoS_Lab5/Appendix/pcap2arrtime.py
  inflating: DDoS_Lab5/Appendix/attackNlog.sh
  inflating: DDoS_Lab5/Appendix/bots.txt
   creating: DDoS_Lab5/Appendix/autoTest2/
  inflating: DDoS_Lab5/Appendix/pcap2entropy.py
  inflating: DDoS_Lab5/Appendix/read_pcap.py
  inflating: DDoS_Lab5/Appendix/dmp2transcript.py
  inflating: DDoS_Lab5/Appendix/calculate_entropyV2.py
  inflating: DDoS_Lab5/Appendix/arrtime2timeseries.py
  inflating: DDoS_Lab5/Appendix/pcap2timeseries.py
  inflating: DDoS_Lab5/Appendix/pcapng2pcap.sh
  inflating: DDoS_Lab5/Appendix/trans2hist.py
  inflating: DDoS_Lab5/timeseries.txt
  inflating: DDoS_Lab5/outputTime0604.entr
  inflating: DDoS_Lab5/complete_attack_times
  inflating: DDoS_Lab5/plot.py
[sandaln@clemson22 ~]$ cd DDoS_Lab5/
[sandaln@clemson22 DDoS_Lab5]$ ls
Appendix  complete_attack_times  outputTime0604.entr  plot.py  timeseries.txt  vda.py
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py -d timeseries.txt 1
```

Fig 2: Command to generate Packet count vs time plot from timeseries.txt

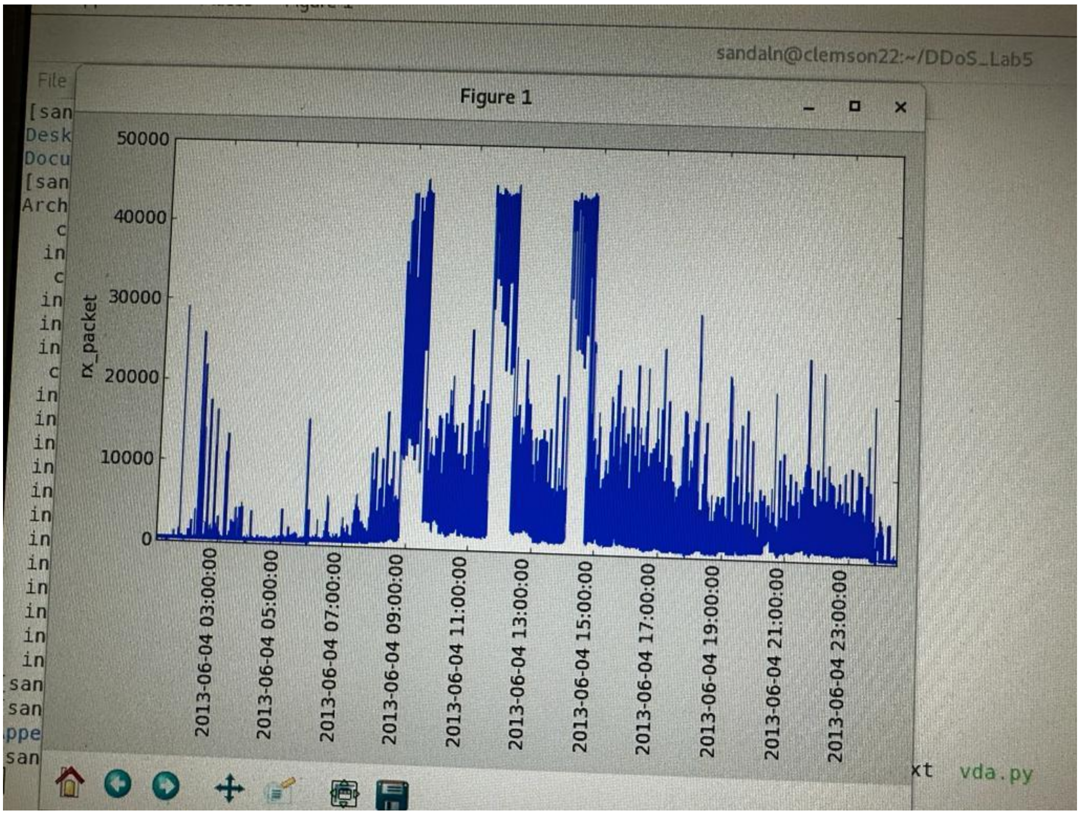
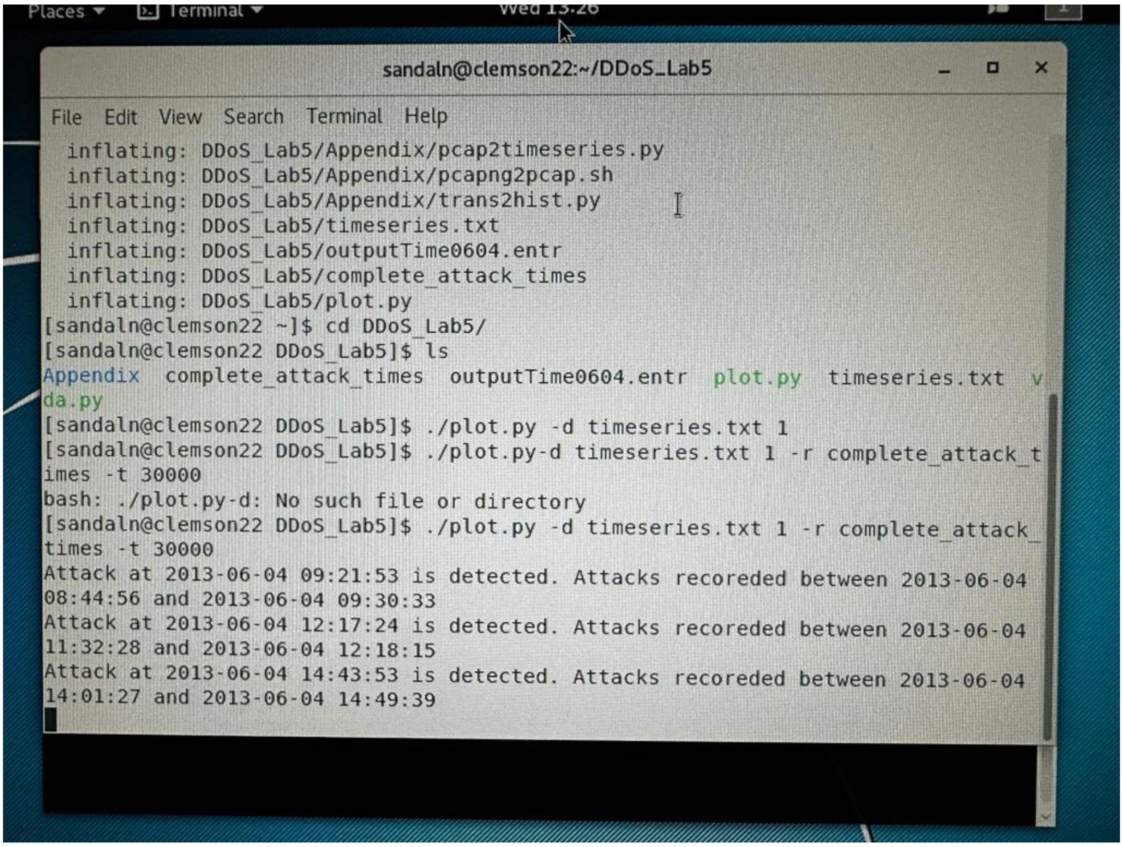


Fig 3: Packet count vs time plot from timeseries.txt



Places
Terminal
Wed 13:26
sandaln@clemson22:~/DDoS_Lab5
File Edit View Search Terminal Help

```
  inflating: DDoS_Lab5/Appendix/pcap2timeseries.py
  inflating: DDoS_Lab5/Appendix/pcapng2pcap.sh
  inflating: DDoS_Lab5/Appendix/trans2hist.py
  inflating: DDoS_Lab5/timeseries.txt
  inflating: DDoS_Lab5/outputTime0604.entr
  inflating: DDoS_Lab5/complete_attack_times
  inflating: DDoS_Lab5/plot.py
[sandaln@clemson22 ~]$ cd DDoS_Lab5/
[sandaln@clemson22 DDoS_Lab5]$ ls
Appendix  complete_attack_times  outputTime0604.entr  plot.py  timeseries.txt  v
da.py
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py -d timeseries.txt 1
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py-d timeseries.txt 1 -r complete_attack_t
imes -t 30000
bash: ./plot.py-d: No such file or directory
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py -d timeseries.txt 1 -r complete_attack_
times -t 30000
Attack at 2013-06-04 09:21:53 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
Attack at 2013-06-04 12:17:24 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 14:43:53 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
```

Fig 4: Command execution to generate packet count vs time graph with threshold 30000.

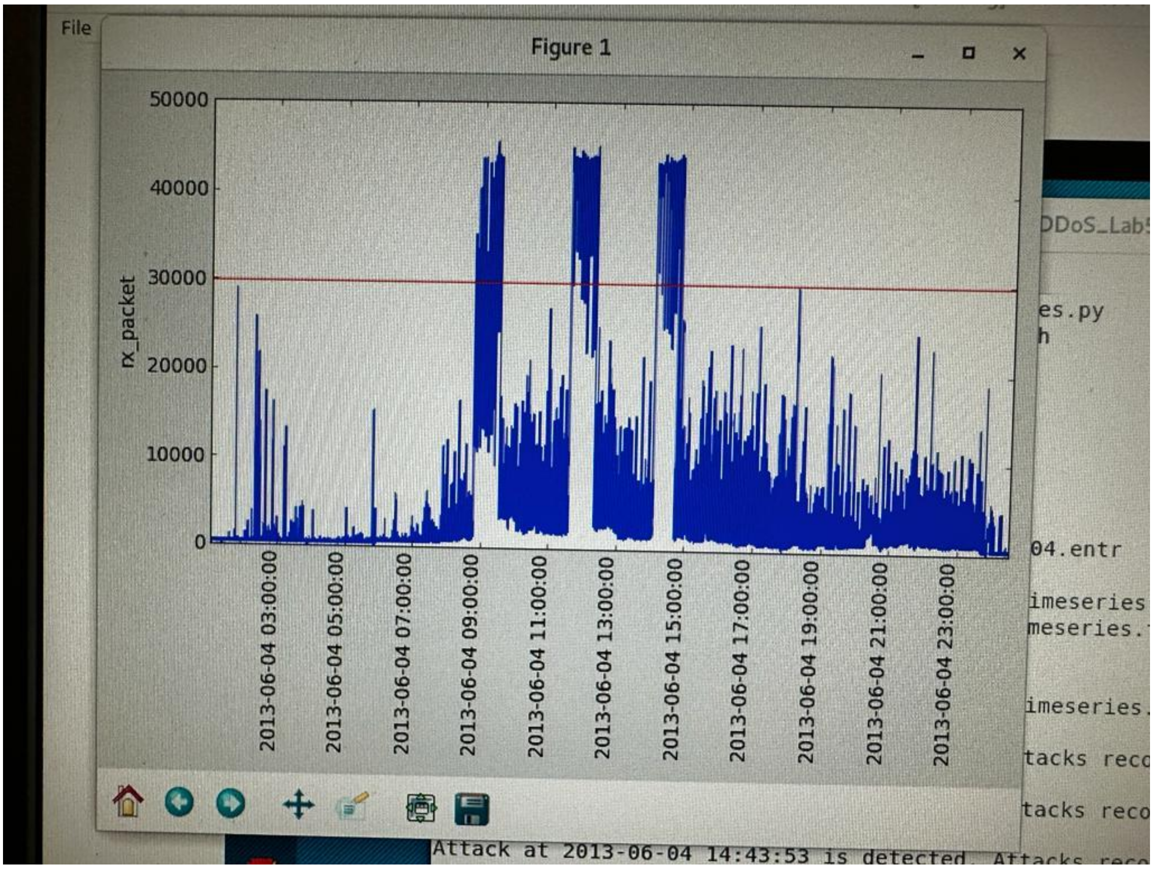
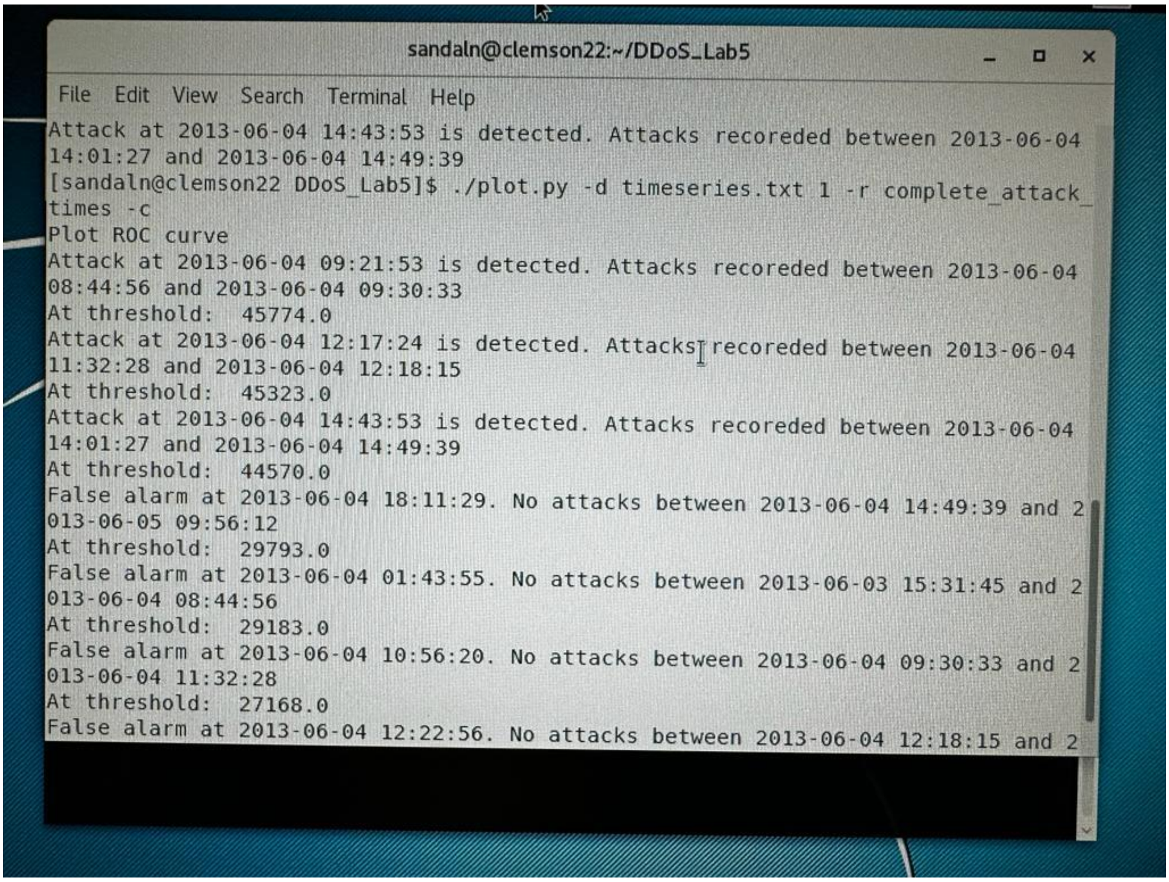


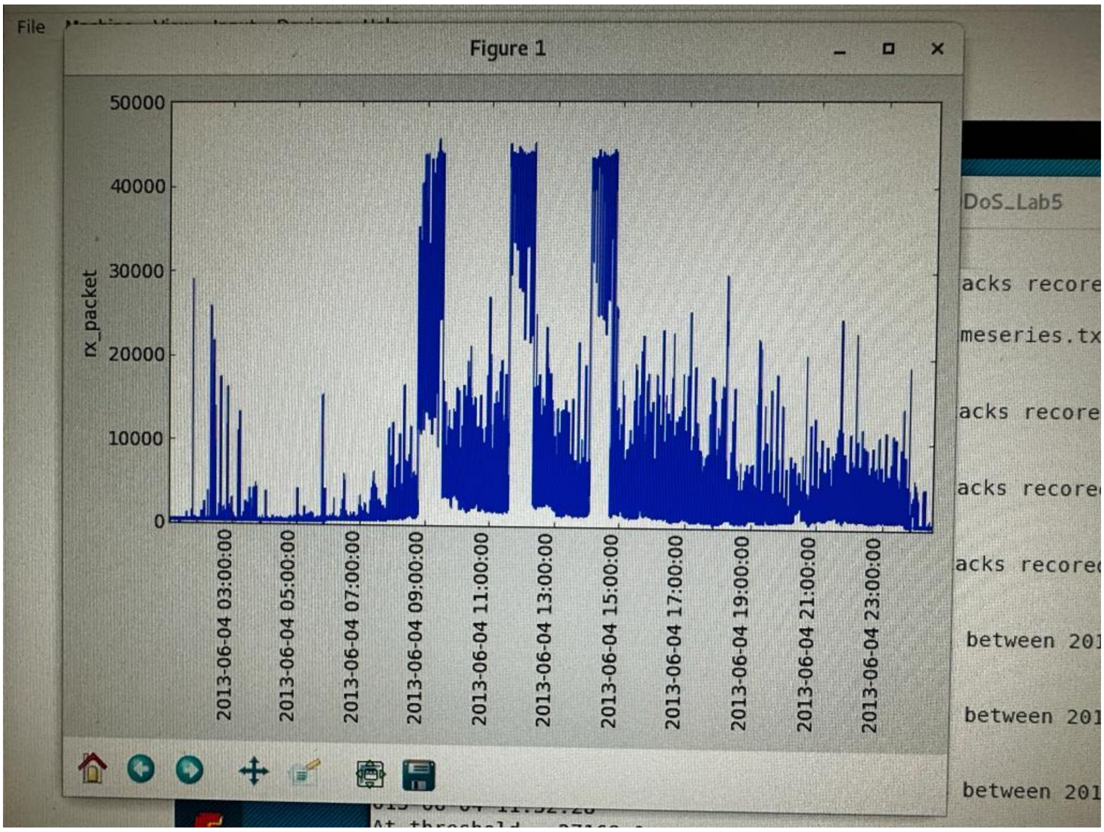
Fig 5: Packet count vs time plot from timeseries.txt with a threshold of 30000.

A straightforward technique for identifying DDoS attacks is to apply a threshold to the traffic time series, which can quickly determine whether a DDoS attack is occurring on the network before any further analysis is performed. Figure 5 illustrates the same plot as Figure 3, but with a threshold of 30000 applied. Figure 3 displays a plot of the number of packets versus time (red line). Anytime the number of packets exceeds this threshold, a DDoS attack is detected. The command used to create the plot in Figure 5 is plot.py, and the complete command is shown in Figure 4.


sandaln@clemson22:~/DDoS_Lab5
File Edit View Search Terminal Help

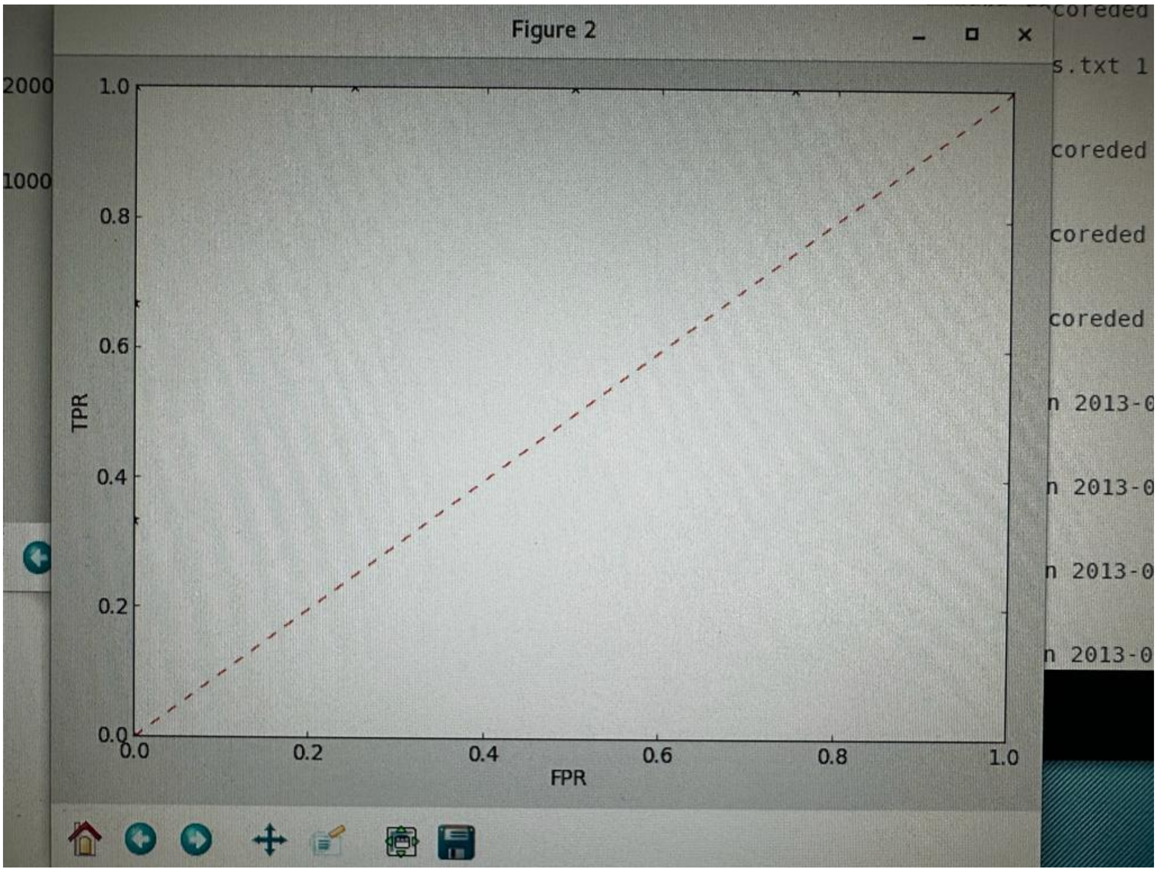
```
Attack at 2013-06-04 14:43:53 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py -d timeseries.txt 1 -r complete_attack_
times -c
Plot ROC curve
Attack at 2013-06-04 09:21:53 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
At threshold:  45774.0
Attack at 2013-06-04 12:17:24 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
At threshold:  45323.0
Attack at 2013-06-04 14:43:53 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
At threshold:  44570.0
False alarm at 2013-06-04 18:11:29. No attacks between 2013-06-04 14:49:39 and 2
013-06-05 09:56:12
At threshold:  29793.0
False alarm at 2013-06-04 01:43:55. No attacks between 2013-06-03 15:31:45 and 2
013-06-04 08:44:56
At threshold:  29183.0
False alarm at 2013-06-04 10:56:20. No attacks between 2013-06-04 09:30:33 and 2
013-06-04 11:32:28
At threshold:  27168.0
False alarm at 2013-06-04 12:22:56. No attacks between 2013-06-04 12:18:15 and 2
```

Fig 6: command for generation of ROC curve of packet count vs time graph



```
acks recore
meseries.tx

acks recore

acks recore

acks recore

 between 201

 between 201

 between 201
```

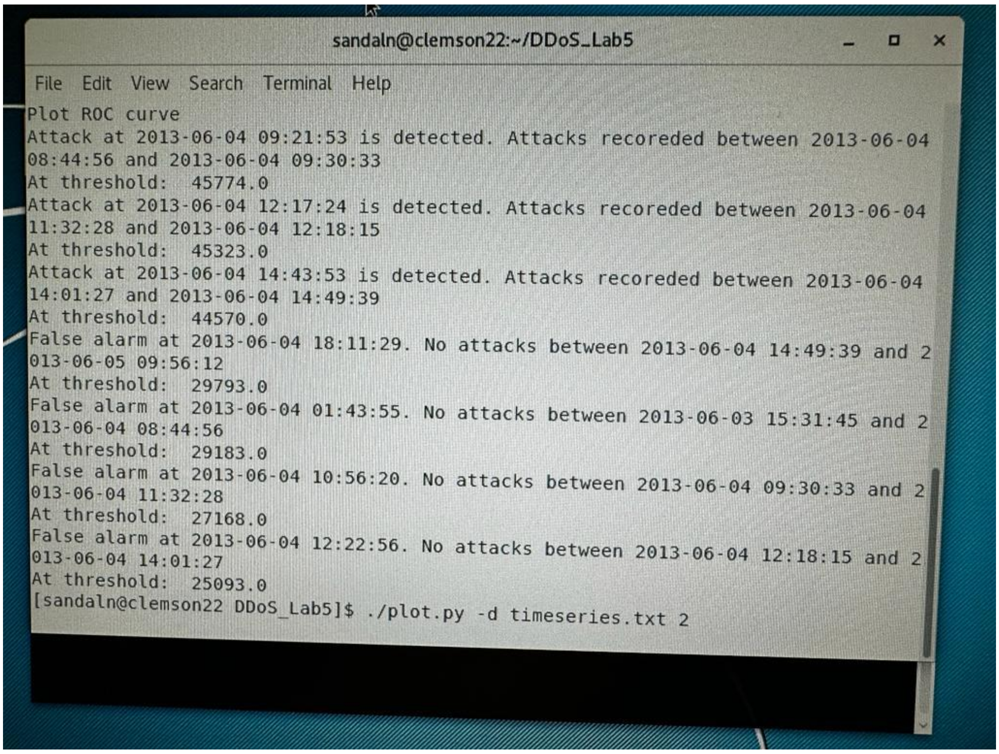
Fig 7: ROC curve of packet count vs time graph



```
coreded
s.txt 1

coreded

coreded

coreded

n 2013-0

n 2013-0

n 2013-0

n 2013-0
```

Fig 8: ROC curve of packet count vs time graph

Figure 7&8 shows a perfect ROC curve which was obtained with an automated ROC function using the command Python command line: “$python plot.py -d timeseries.txt 1 -r complete_attack_times -c”


sandaln@clemson22:~/DDoS_Lab5
File Edit View Search Terminal Help

```
Plot ROC curve
Attack at 2013-06-04 09:21:53 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
At threshold:  45774.0
Attack at 2013-06-04 12:17:24 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
At threshold:  45323.0
Attack at 2013-06-04 14:43:53 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
At threshold:  44570.0
False alarm at 2013-06-04 18:11:29. No attacks between 2013-06-04 14:49:39 and 2
013-06-05 09:56:12
At threshold:  29793.0
False alarm at 2013-06-04 01:43:55. No attacks between 2013-06-03 15:31:45 and 2
013-06-04 08:44:56
At threshold:  29183.0
False alarm at 2013-06-04 10:56:20. No attacks between 2013-06-04 09:30:33 and 2
013-06-04 11:32:28
At threshold:  27168.0
False alarm at 2013-06-04 12:22:56. No attacks between 2013-06-04 12:18:15 and 2
013-06-04 14:01:27
At threshold:  25093.0
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py -d timeseries.txt 2
```

Fig 9: command for generating bytes vs time graph from timeseries.txt

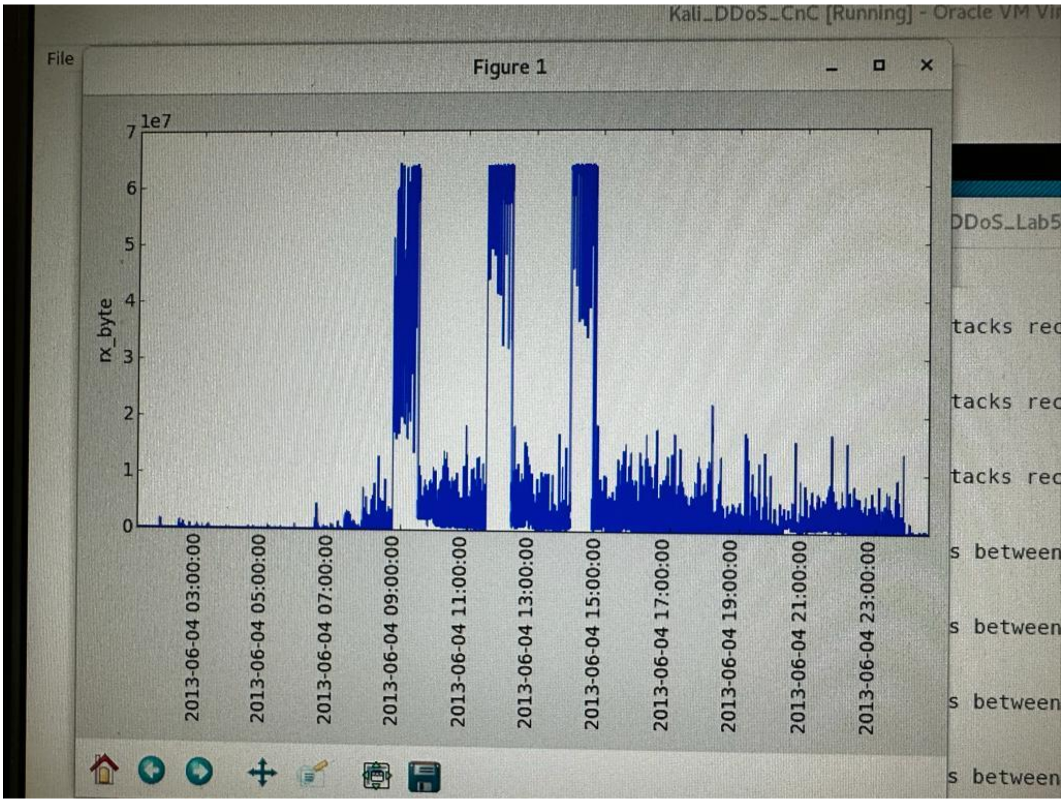
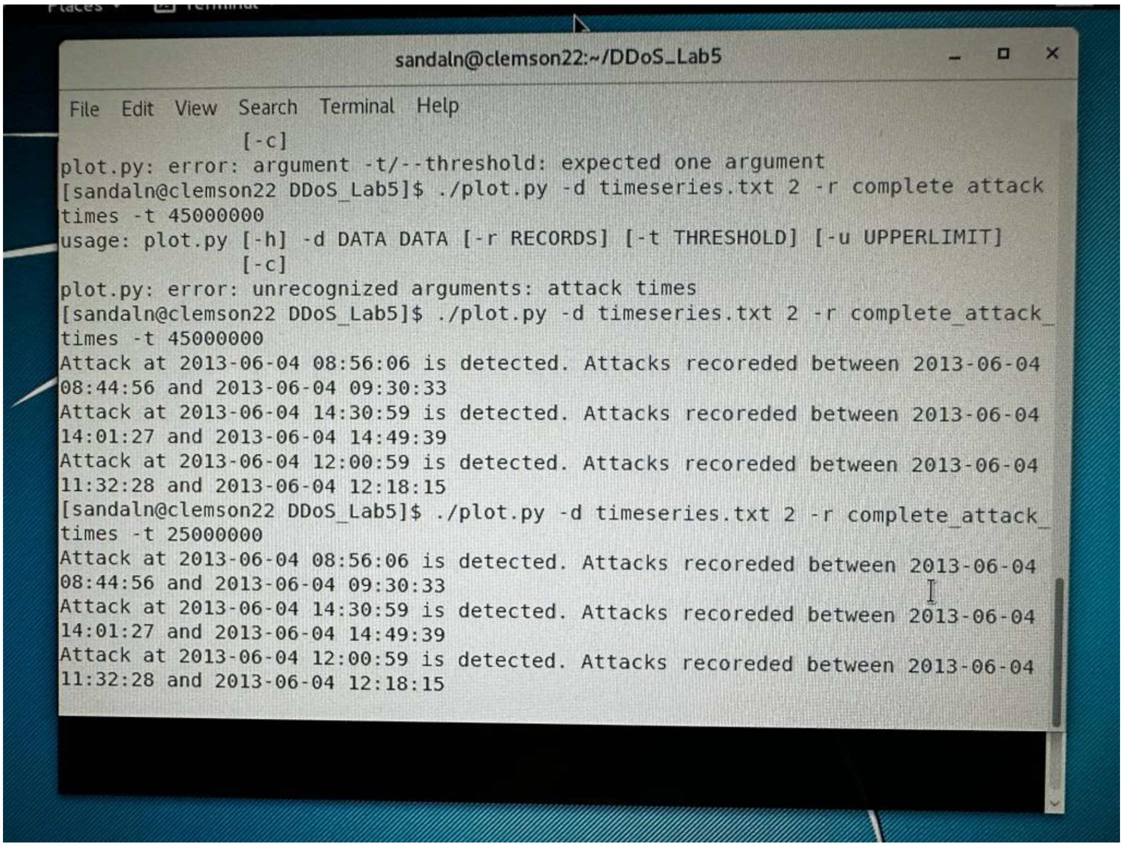


Fig 10: Received bytes vs time graph from timeseries.txt



sandaln@clemson22:~/DDoS_Lab5
File Edit View Search Terminal Help

```
               [-c]
plot.py: error: argument -t/--threshold: expected one argument
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py -d timeseries.txt 2 -r complete attack
times -t 45000000
usage: plot.py [-h] -d DATA DATA [-r RECORDS] [-t THRESHOLD] [-u UPPERLIMIT]
               [-c]
plot.py: error: unrecognized arguments: attack times
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py -d timeseries.txt 2 -r complete_attack_
times -t 45000000
Attack at 2013-06-04 08:56:06 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
Attack at 2013-06-04 14:30:59 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
Attack at 2013-06-04 12:00:59 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
[sandaln@clemson22 DDoS_Lab5]$ ./plot.py -d timeseries.txt 2 -r complete_attack_
times -t 25000000
Attack at 2013-06-04 08:56:06 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
Attack at 2013-06-04 14:30:59 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
Attack at 2013-06-04 12:00:59 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
```

Fig 11: set the threshold for 2500000 to get bytes vs time graph.

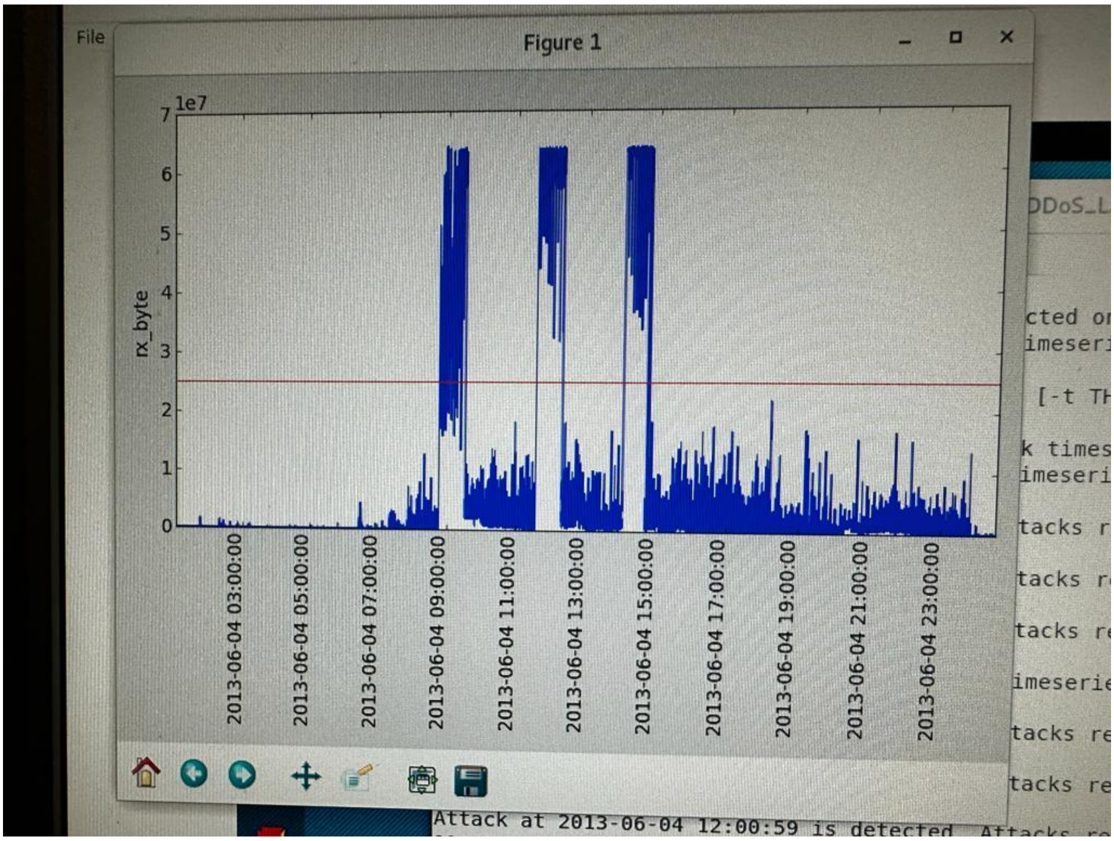
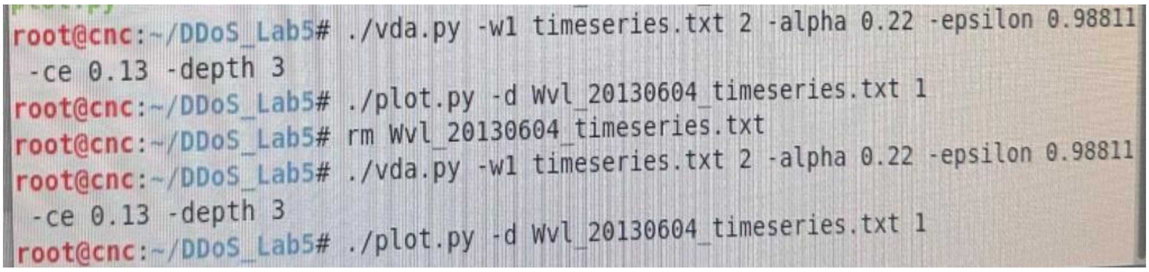


Fig 12: Determining threshold for received bytes vs time graph.



```
root@cnc:~/DDoS_Lab5# ./vda.py -w1 timeseries.txt 2 -alpha 0.22 -epsilon 0.98811
 -ce 0.13 -depth 3
root@cnc:~/DDoS_Lab5# ./plot.py -d Wvl_20130604_timeseries.txt 1
root@cnc:~/DDoS_Lab5# rm Wvl_20130604_timeseries.txt
root@cnc:~/DDoS_Lab5# ./vda.py -w1 timeseries.txt 2 -alpha 0.22 -epsilon 0.98811
 -ce 0.13 -depth 3
root@cnc:~/DDoS_Lab5# ./plot.py -d Wvl_20130604_timeseries.txt 1
```

Figure 13: generated Wvl_20130604.txt to perform Wavelet detection.

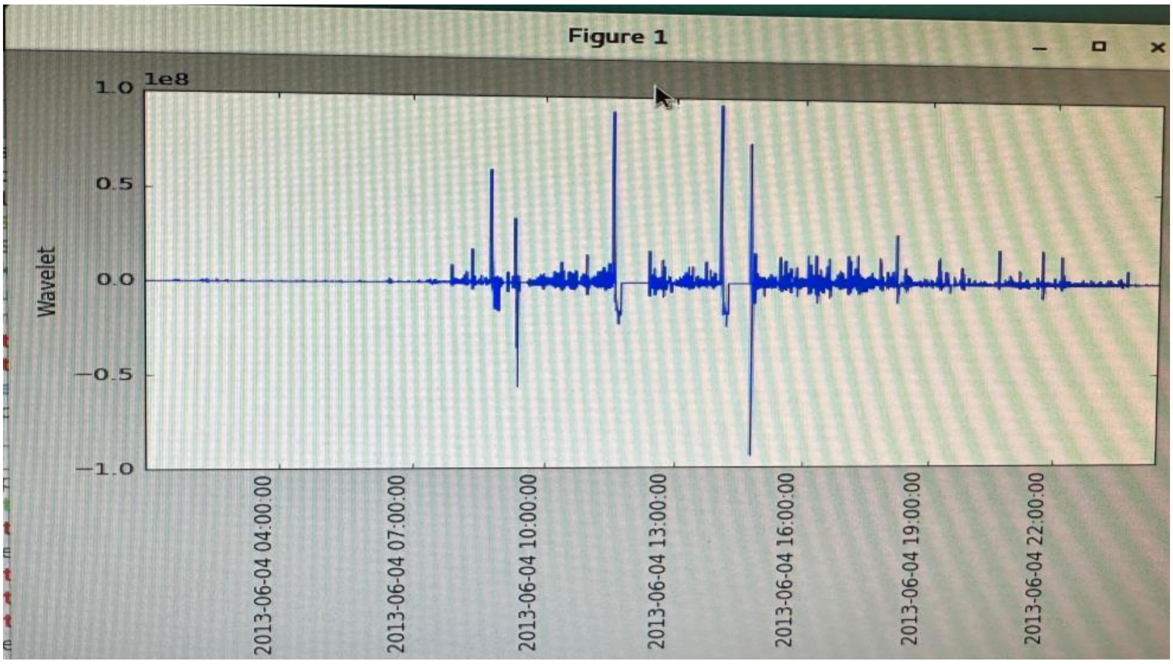


Figure 14: Wavelet transform of received bytes vs time graph.

To avoid issues with dependency on local machines, all subsequent experiments were carried out on the CnC machine, which will be further discussed later in this report. The Wavelet analysis on the bytes versus time data from the received timeseries.txt was performed using the following command.

Python command line: "$python vda.py -w1 timeseries.txt 2 -alpha 0.22 -epsilon 0.98811 -ce 0.13 -depth 3" This is seen in Figure 13. The command above created a file called Wvl_20130604_ timeseries.txt, and a plot can be made using the file (see Figure 14).

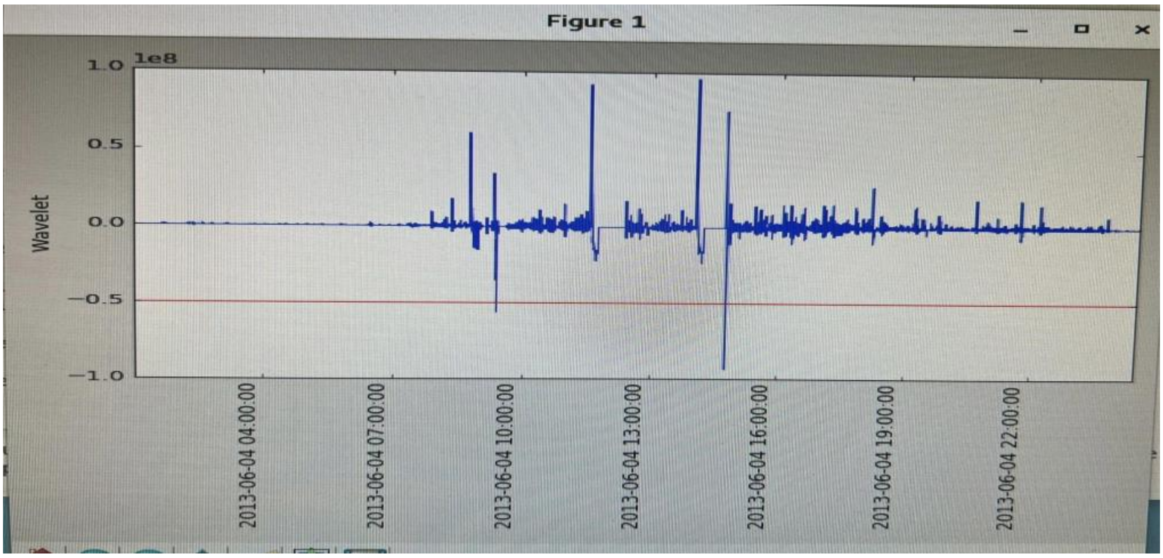


Figure 15: Wavelet transform with low pass filter of -50000000.

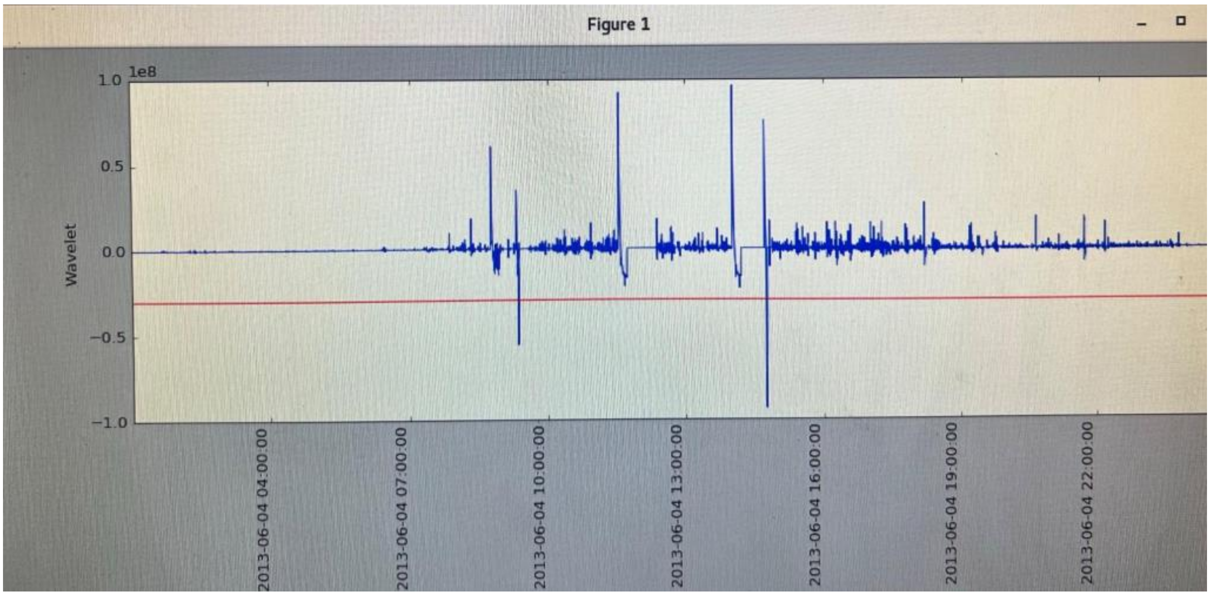
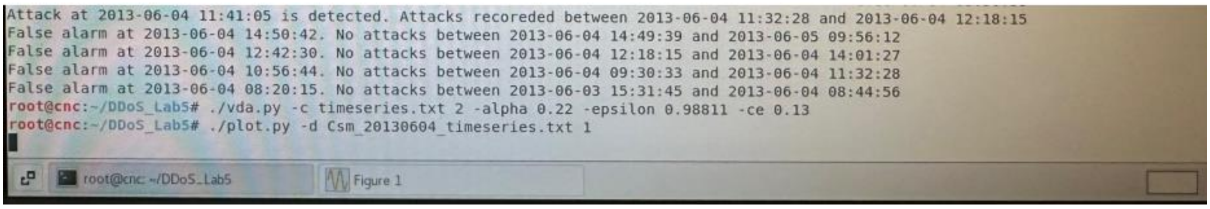


Figure 16: Wavelet transform with low pass filter of -30000000

After obtaining the Wavelet transform plot, a low pass filter was applied with a value of -50000000, as shown in Figure 15. However, it was found that a low pass filter with a value of -30000000 was more effective in detecting DDoS attacks. This implies that any node below the threshold of -30000000 is considered to have detected a DDoS attack.



```
Attack at 2013-06-04 11:41:05 is detected. Attacks recoreded between 2013-06-04 11:32:28 and 2013-06-04 12:18:15
False alarm at 2013-06-04 14:50:42. No attacks between 2013-06-04 14:49:39 and 2013-06-05 09:56:12
False alarm at 2013-06-04 12:42:30. No attacks between 2013-06-04 12:18:15 and 2013-06-04 14:01:27
False alarm at 2013-06-04 10:56:44. No attacks between 2013-06-04 09:30:33 and 2013-06-04 11:32:28
False alarm at 2013-06-04 08:20:15. No attacks between 2013-06-03 15:31:45 and 2013-06-04 08:44:56
root@cnc:~/DDoS_Lab5# ./vda.py -c timeseries.txt 2 -alpha 0.22 -epsilon 0.98811 -ce 0.13
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_20130604_timeseries.txt 1
```

Figure 17: Generated Csm_20130604_timeseries.txt to perform CUSUM analysis on received bytes column.

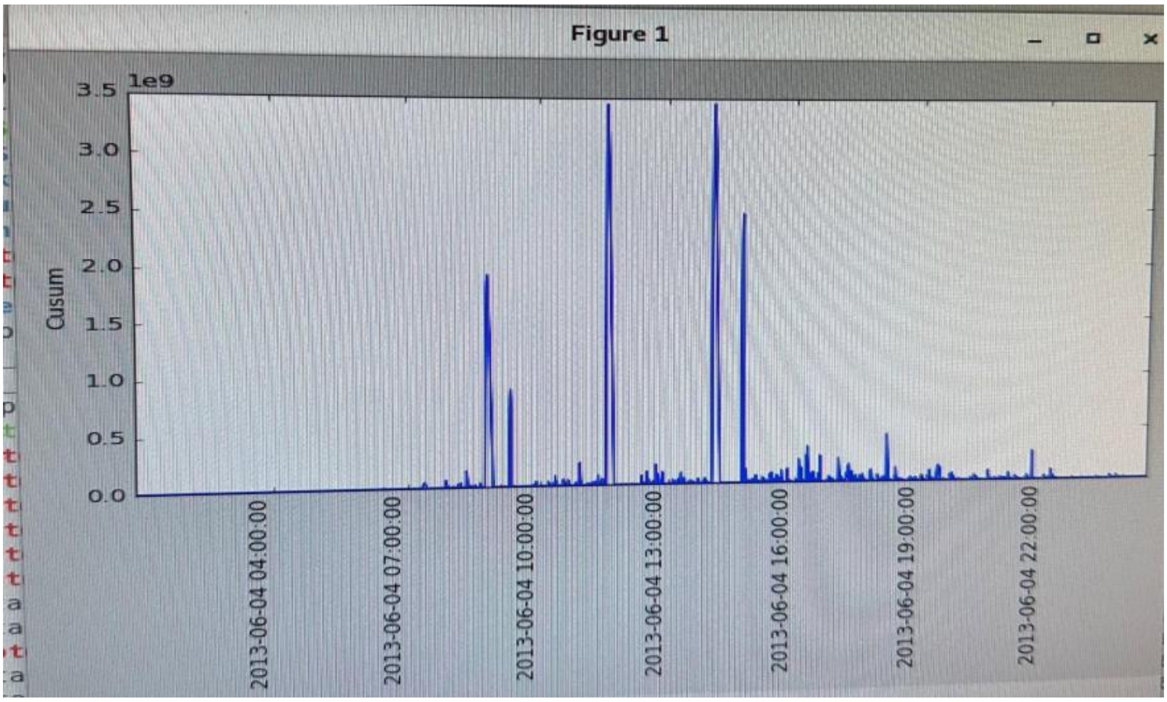
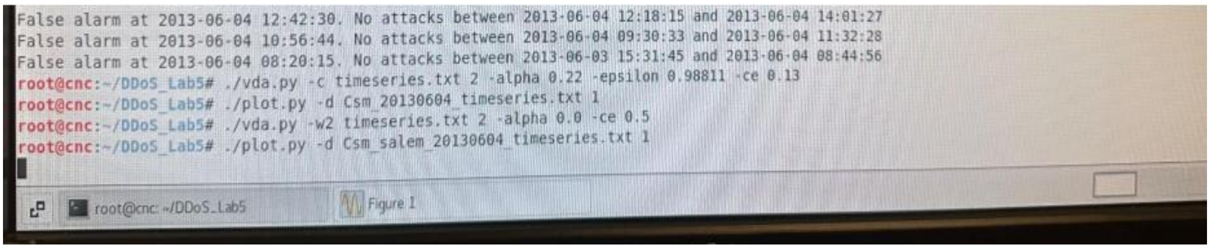


Figure 18: CUSUM analysis of received bytes vs time graph.

Initially, Wavelet transform analysis was presented as a technique for detecting DDoS attacks, followed by CUSUM analysis. To apply CUSUM analysis, the rx_byte column of the timeseries.txt file had to be analyzed. The command used and the resulting file, Csm_20130604_timeseries.txt, are both displayed in Figure 17. Figure 18 illustrates a plot generated from the resultant file, and the estimated threshold value is around 0.85.



```
False alarm at 2013-06-04 12:42:30. No attacks between 2013-06-04 12:18:15 and 2013-06-04 14:01:27
False alarm at 2013-06-04 10:56:44. No attacks between 2013-06-04 09:30:33 and 2013-06-04 11:32:28
False alarm at 2013-06-04 08:20:15. No attacks between 2013-06-03 15:31:45 and 2013-06-04 08:44:56
root@cnc:~/DDoS_Lab5# ./vda.py -c timeseries.txt 2 -alpha 0.22 -epsilon 0.98811 -ce 0.13
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_20130604_timeseries.txt 1
root@cnc:~/DDoS_Lab5# ./vda.py -w2 timeseries.txt 2 -alpha 0.0 -ce 0.5
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_salem_20130604_timeseries.txt 1
```

Figure 19: Generated Csm_salem_20130604_timeseries.txt to perform WAVE-CUSUM analysis.

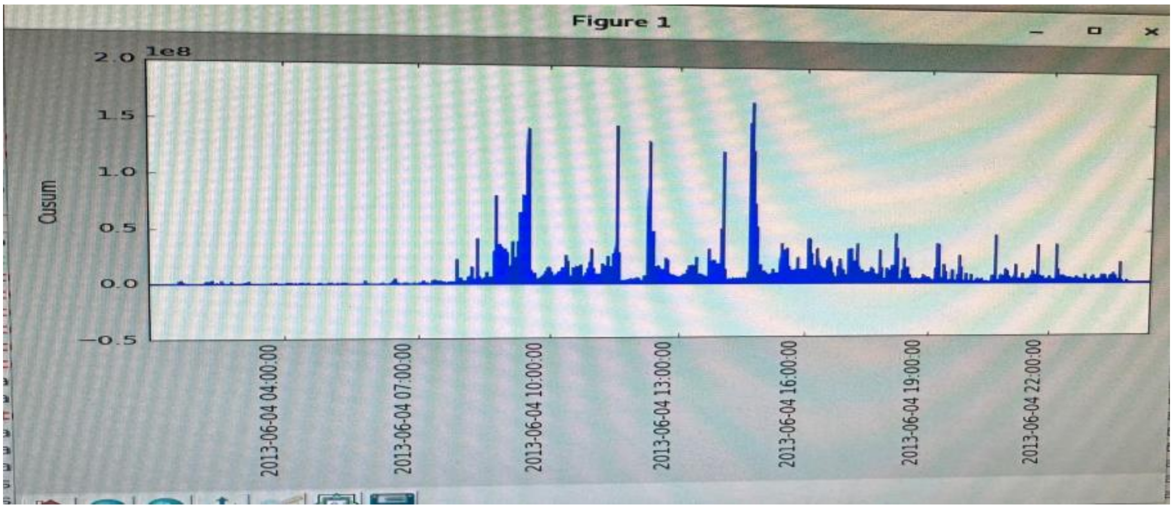
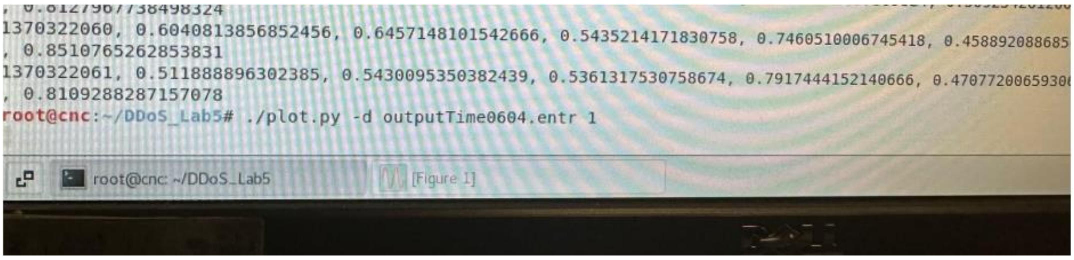


Figure 20: Plot of WAVE-CUSUM analysis. The WAVE-CUSUM detection technique comes next after CUSUM.

After conducting the WAVE-CUSUM analysis, the output file Csm_salem_20130604_timeseries.txt was generated as depicted in Figure 19, with the corresponding plot presented in Figure 20. Based on the results, the threshold value for detecting a DDoS attack falls within the range of 0.8 to 0.85.



```
, 0.8127967738498324
1370322060, 0.6040813856852456, 0.6457148101542666, 0.5435214171830758, 0.7460510006745418, 0.458892088685
, 0.8510765262853831
1370322061, 0.511888896302385, 0.5430095350382439, 0.5361317530758674, 0.7917444152140666, 0.47077200659306
, 0.8109288287157078
root@cnc:~/DDoS_Lab5# ./plot.py -d outputTime0604.entr 1
```

Figure 21: Executed the outputTime0604.entr 1 command.

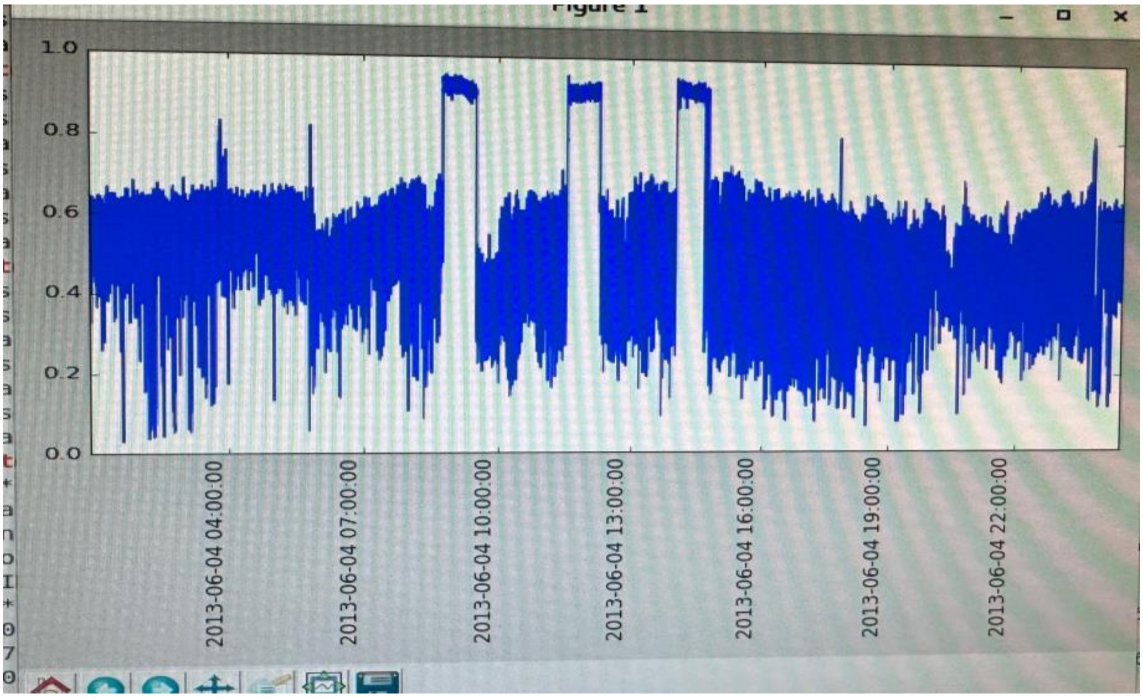
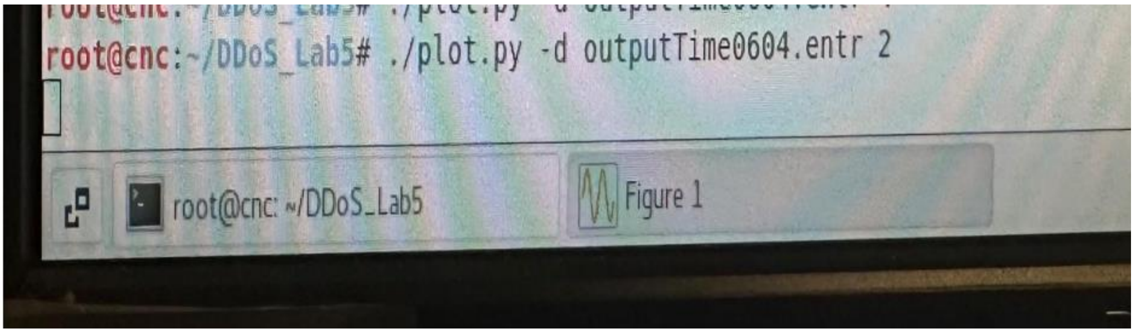


Figure 22: Plot of srcIP column from outputTime0604.entr



```
root@cnc:~/DDoS_Lab5# ./plot.py -d outputTime0604.entr 2
```

Figure 23: Executed the outputTime0604.entr 2 command.

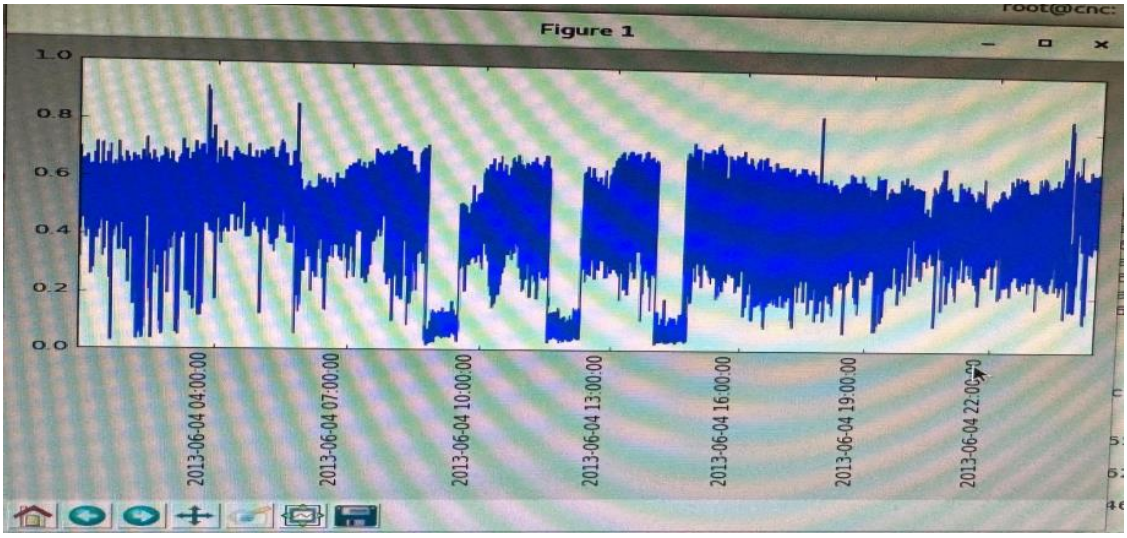
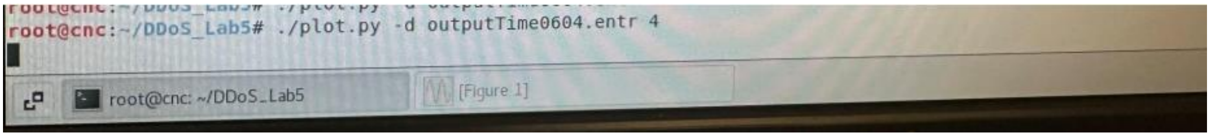


Figure 24: Plot of destIP column from outputTime0604 2.entr.



```
root@cnc:~/DDoS_Lab5# ./plot.py -d outputTime0604.entr 4
```

Figure 25: Executed the outputTime0604.entr 4 command.

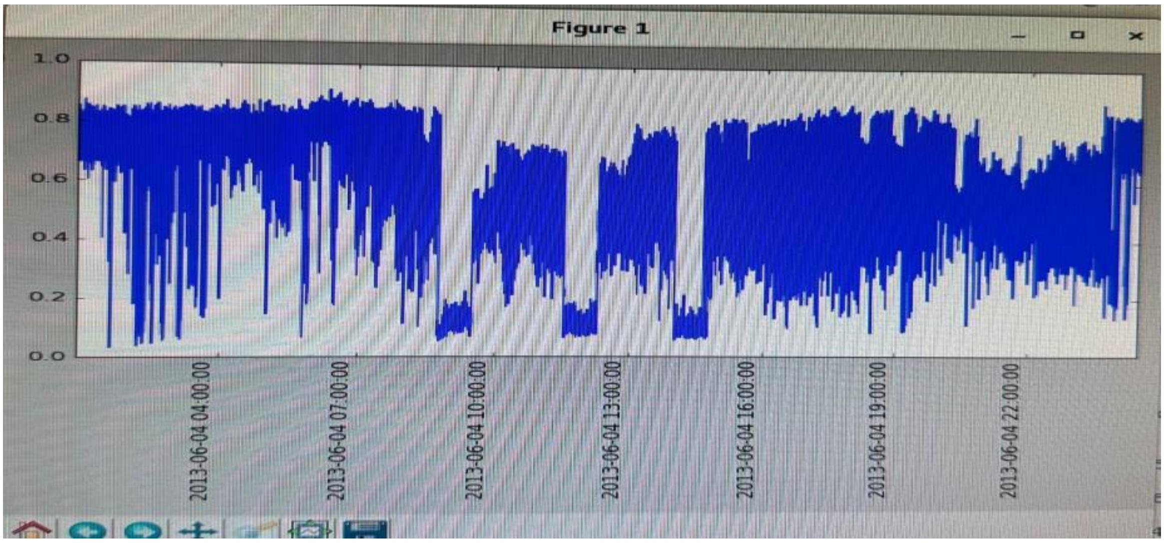


Figure 26: Plot of destPort column from outputTime0604 4.entr.

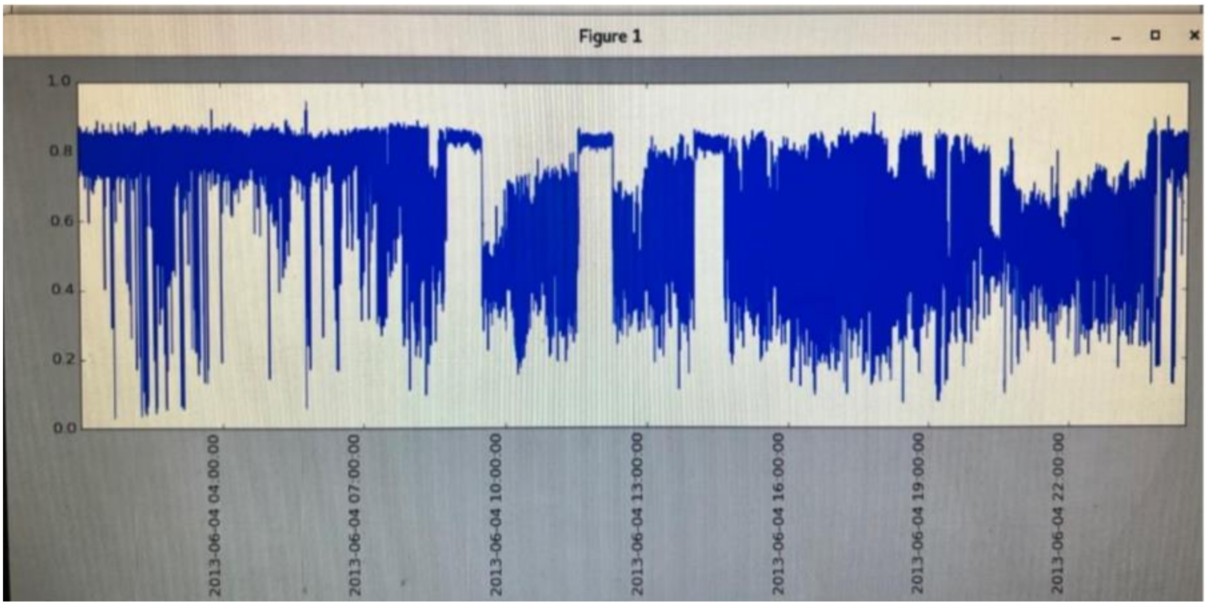
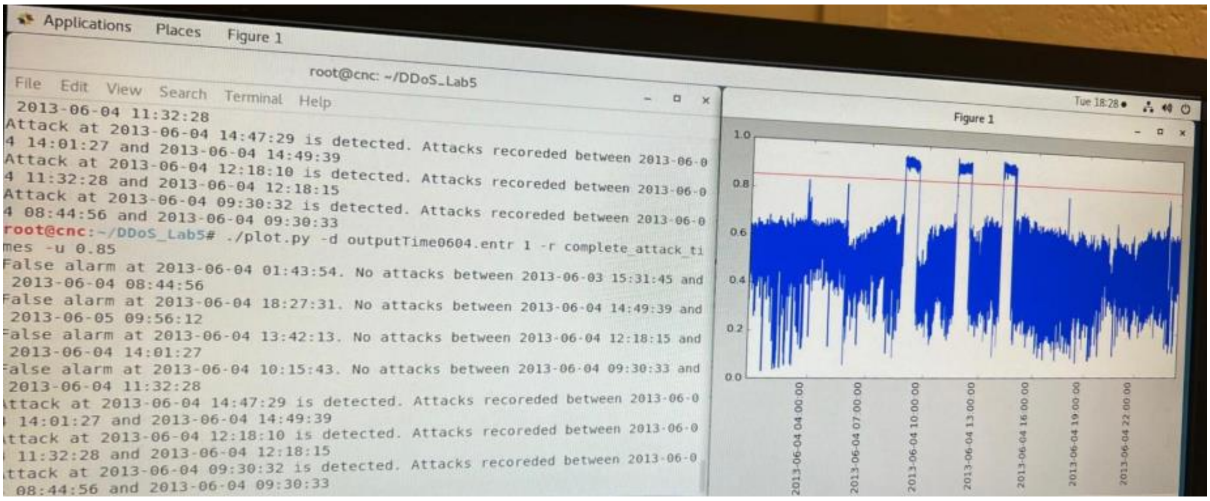


Figure 27: Plot of srcIPdestIPsrcPortdestPort column from outputTime0604.entr



```
 2013-06-04 11:32:28
Attack at 2013-06-04 14:47:29 is detected. Attacks recoreded between 2013-06-0
4 14:01:27 and 2013-06-04 14:49:39
Attack at 2013-06-04 12:18:10 is detected. Attacks recoreded between 2013-06-0
4 11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 09:30:32 is detected. Attacks recoreded between 2013-06-0
4 08:44:56 and 2013-06-04 09:30:33
root@cnc:~/DDoS_Lab5# ./plot.py -d outputTime0604.entr 1 -r complete_attack_ti
mes -u 0.85
False alarm at 2013-06-04 01:43:54. No attacks between 2013-06-03 15:31:45 and
 2013-06-04 08:44:56
False alarm at 2013-06-04 18:27:31. No attacks between 2013-06-04 14:49:39 and
 2013-06-05 09:56:12
False alarm at 2013-06-04 13:42:13. No attacks between 2013-06-04 12:18:15 and
 2013-06-04 14:01:27
False alarm at 2013-06-04 10:15:43. No attacks between 2013-06-04 09:30:33 and
 2013-06-04 11:32:28
Attack at 2013-06-04 14:47:29 is detected. Attacks recoreded between 2013-06-0
4 14:01:27 and 2013-06-04 14:49:39
Attack at 2013-06-04 12:18:10 is detected. Attacks recoreded between 2013-06-0
4 11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 09:30:32 is detected. Attacks recoreded between 2013-06-0
4 08:44:56 and 2013-06-04 09:30:33
```

Figure 28: Plot of srcIP column from outputTime0604.entr with threshold of 0.85

The final technique for DDoS attack detection is the entropy-based approach. In this regard, the pre-processed file outputTime0604.entr has been provided, and Figures 21 to 26 depict the plots of srcIP, destIP, and destPort, respectively, from each column of the entropy file. The aim of plotting these columns is to determine the most effective information for detecting DDoS attacks. Based on the analysis, Figure 28 shows the plot of the srcIP column with a threshold of 0.85, which is considered the optimal plot for DDoS attack detection.

Discussions

In the present study, various techniques were evaluated for the detection of DDoS attacks, each with its own strengths and limitations. Packet count and data volume thresholding were identified as the simplest and most straightforward techniques, yet they are vulnerable to attacks that are designed to stay just below the threshold values.

Furthermore, false positives may occur if legitimate traffic surpasses the threshold numbers. Wavelet analysis, on the other hand, was found to be a powerful method for studying non-stationary signals and detecting DDoS attacks, but it requires significant computational resources and may not be suitable for real-time detection. Additionally, the accuracy of the analysis may depend on the selected wavelet function and parameters. CUSUM analysis, a statistical technique for detecting changes in mean values of time series, was also examined, but it may result in false positives if variations in traffic patterns are due to legitimate reasons, or it may not be effective in identifying attacks that spread across multiple networks or computers.

To overcome these limitations, WAVE-CUSUM analysis, which combines the strengths of wavelet and CUSUM analysis, was proposed. However, it still requires substantial computing resources and may not be suitable for real-time detection. Entropy analysis, which measures the randomness of traffic, was also explored and found to be effective in detecting DDoS attacks that produce low-entropy traffic.

Nevertheless, false positives may occur if the traffic has low entropy by nature or if the attack is designed to mimic normal traffic patterns.

Conclusions

In conclusion, it is important to note that there is no one-size-fits-all solution to detecting DDoS attacks. Each technique has its own strengths and weaknesses, and the optimal approach will depend on the specific requirements and limitations of the network being protected. Therefore, it is essential to carefully consider the goals of the DDoS detection system, as well as the available resources and expertise, when selecting a technique or combination of techniques. Ultimately, the key to successful DDoS detection lies in continuously monitoring and analysing network traffic in real-time, as well as staying up-to-date with the latest attack strategies and mitigation techniques. By doing so, organizations can effectively defend against DDoS attacks and protect their critical infrastructure and services.

Collaboration

For the completion of this lab report, I worked collaboratively with Manasa Thatipamula, and Anjan Kumar Depuru.

References

[1] Carl, G., Brooks, R. R., & Rai, S. (2006). Wavelet based denial-of-service detection. Computers & Security, 25(8), 600-615.

[2] Callegari, C., Giordano, S., Pagano, M., Pepe, T. (2012). WAVE-CUSUM: Improving CUSUM performance in network anomaly detection by means of wavelet analysis. Computers & Security, 31(5), 727-735.

[3] Incapsula. (n.d.). Incapsula Survey: What DDoS Attacks Really Cost Businesses. Retrieved from https://lp.incapsula.com/rs/incapsulainc/images/eBook%20-%20DDoS%20Impact%20Survey.pdf.

[4] Radware. (2013, May 29). How Much Can a DDoS Attack Cost Your Business? [Blog post]. Retrieved from https://blog.radware.com/security/2013/05/how-much-can-a-ddos-attack-cost-your-business/.