Laboratory: 5

DDoS Attack Detection

Introduction

This report includes detection of DDoS attacks using Traffic Volume, CUSUM, wavelet and entropy methods. Here, hours of traffic has been captured and the capture data is preprocessed and intermediate files has been created for performing the detections.

Different python scripts are being used for the Traffic Volume, CUSUM, wavelet and entropy detection.

Setup and Process

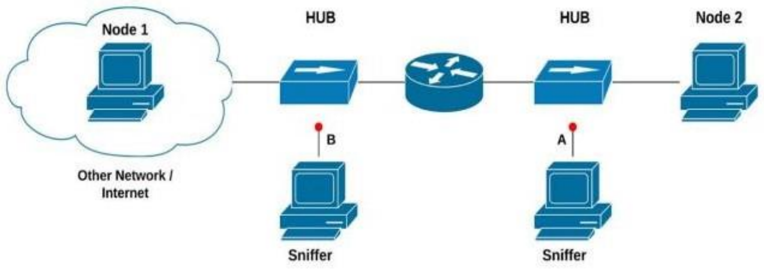


Figure1: Network set-up

Here, as shown above, node 1 (192.168.10.10) is machine from lab in clemson and node 2 (192.168.20.13) is machine from lab in charleston. All the machines are connected over the HUB and the red dots are the sniffing points (192.168.10.9). One Command & Control machine (192.168.10.111) is also connected over the sub-net that is used to observe and connect to the bots.

Now, coming to our first type of detection: Traffic Volume detection using packet count thresholding. We login to CnC machine and plot the packets count time series for the intermediate file "timeseries.txt" using the following command:

```
./plot.py -d timeseries.txt 1
```

We can see the resultant graph below (figure: 2), we can notice how the y-axis represents "rx_packet", meaning that the y-axis represents the number of packets.

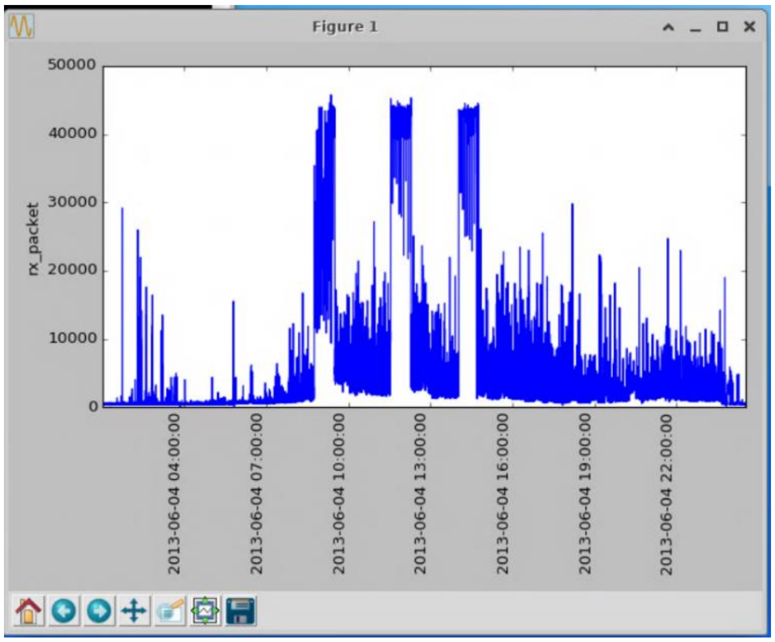


Figure 2: packets count time series

Now, looking at the traffic we can see the 3 spikes that represent the attack. Now we set a good threshold value accordingly at 30000 such that the attack is detected if the packet count crosses the threshold. I use the following command to set the threshold value:

./plot.py -d timeseries.txt 1 -r complete_attack_times -t 30000

We can see in the figures below (Figure: 3, Figure: 4) Every time the packet value has gone above the threshold value set by us, DDoS attack has been detected.

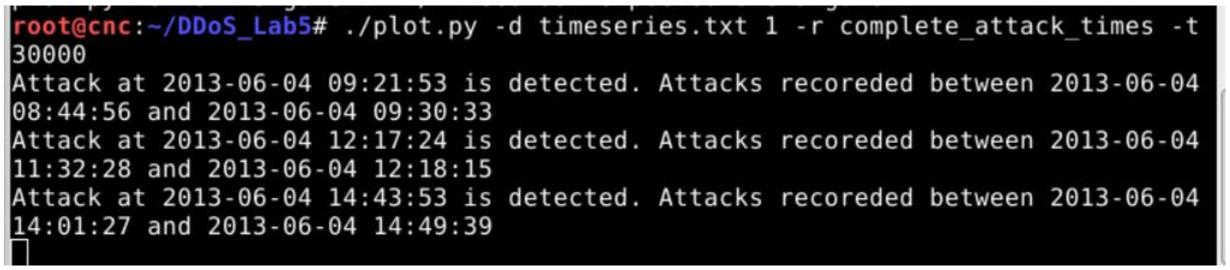

```
root@cnc:~/DDoS_Lab5# ./plot.py -d timeseries.txt 1 -r complete_attack_times -t
30000
Attack at 2013-06-04 09:21:53 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
Attack at 2013-06-04 12:17:24 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 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
```

Figure 3: Attacks detected using packet count thresholding

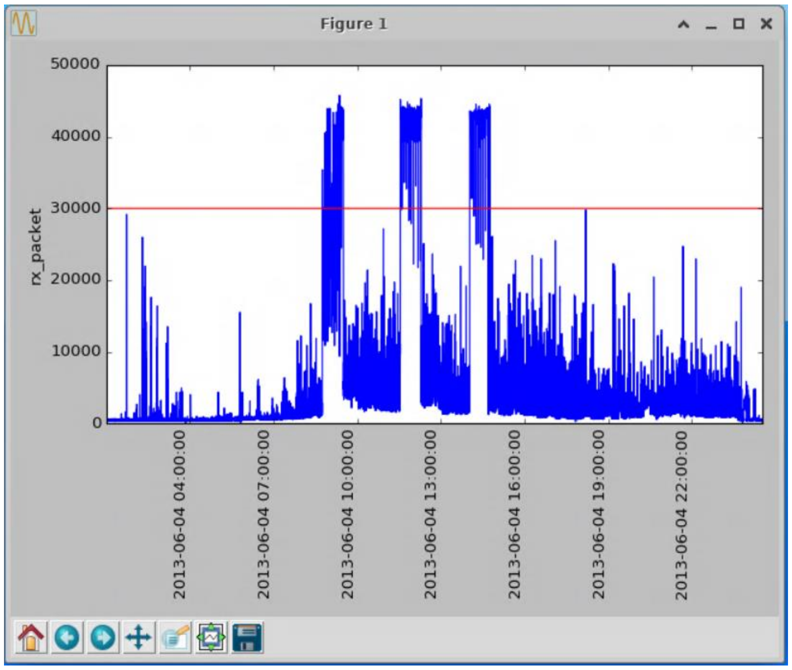


Figure 4: Threshold set at 30000

Now, we plot the ROC curve for the detection. I used the following command to plot the ROC curve:

```
./plot.py -d timeseries.txt 1 -r complete_attack_times -c
```

We can see that for this attack log, we get the perfect ROC curve (Figure 5)

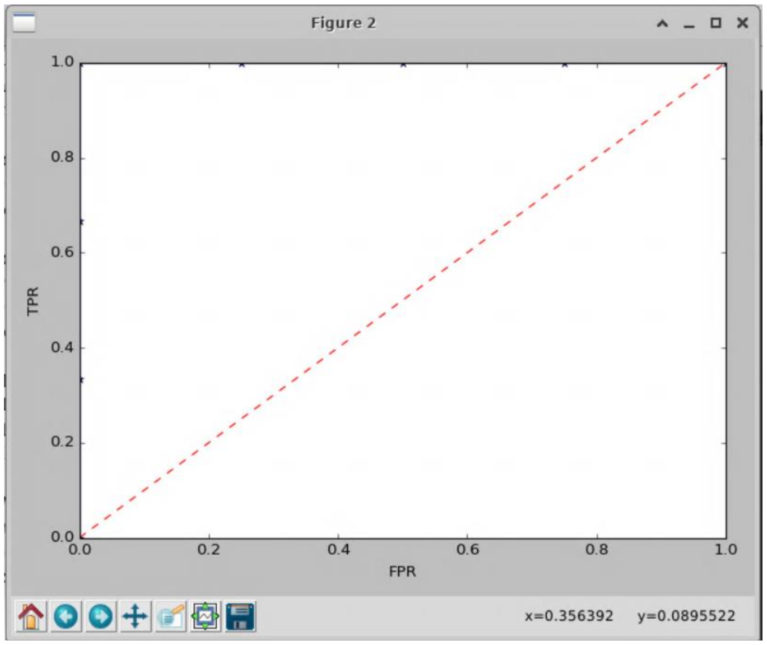


Figure 5: ROC curve

Now we move on to Traffic volume detection using Data Volume Thresholding. Here, we plot the time series for the intermediate file "timeseries.txt" with y-axis representing "rx_byte" (Figure: 6)

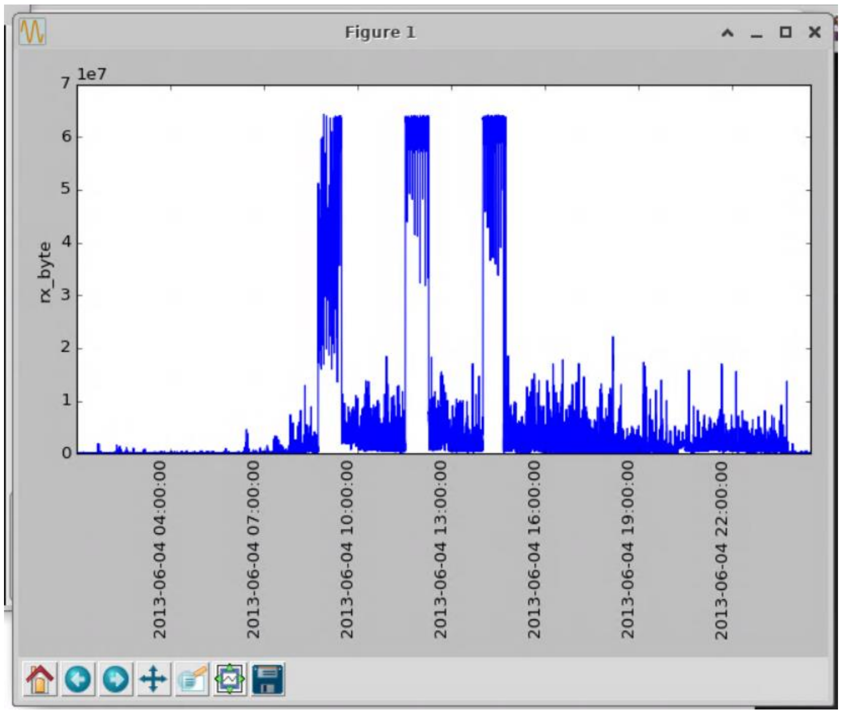
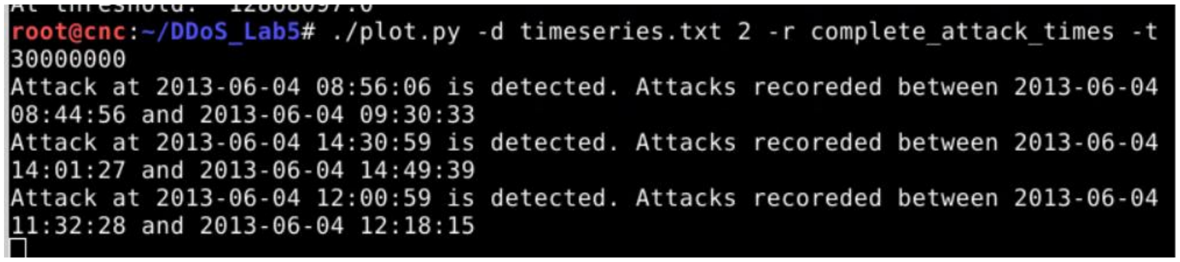


Figure 6: Data volume time series

Now we set a good threshold value accordingly at 30000000 such that the attack is detected if the packet volume crosses the threshold (Figure: 7, Figure: 8). I use the following command to set the threshold value:

./plot.py -d timeseries.txt 2 -r complete_attack_times -t 30000000



```
root@cnc:~/DDoS_Lab5# ./plot.py -d timeseries.txt 2 -r complete_attack_times -t
30000000
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
```

Figure 7: Attacks detected using packet volume thresholding

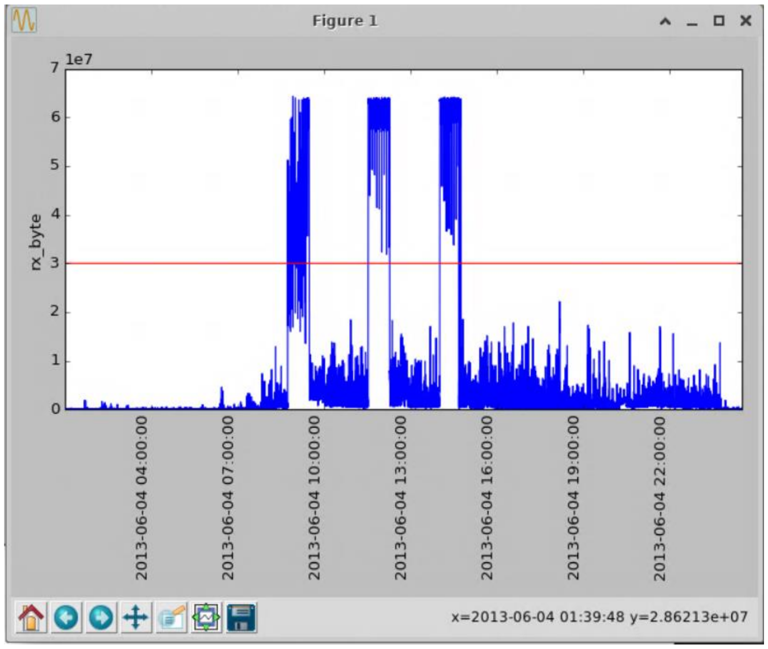


Figure 8: Threshold set at 300000000

Now, we plot the ROC curve for the detection. I used the following command to plot the ROC curve:

./plot.py -d timeseries.txt 2 -r complete_attack_times -c

We can see that for this attack log, we get the perfect ROC curve (Figure 9)

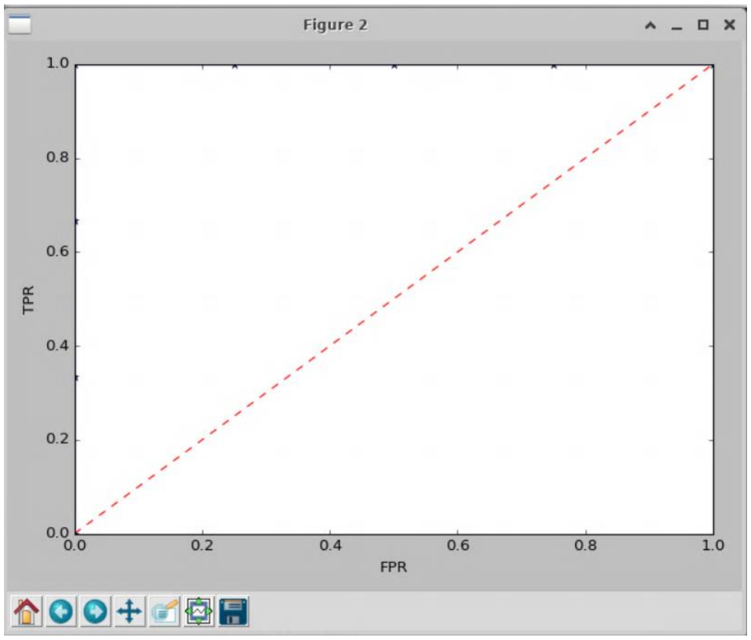


Figure 9: ROC curve

Coming to our next type of detection, CUSUM Detection, We execute *# ./vda.py -c timeseries.txt 2 -alpha 0.22 -epsilon 0.98811 -ce 0.13* command on the “rx_byte” column in the file “timeseries.txt” with the default parameters such that it generates an intermediate file “Csm_20130604_timeseries.txt” which we use for performing CUSUM analysis. We can see the time series graph of it below (Figure: 10) which is generated using *./plot.py -d “Csm_20130604_timeseries.txt 1*

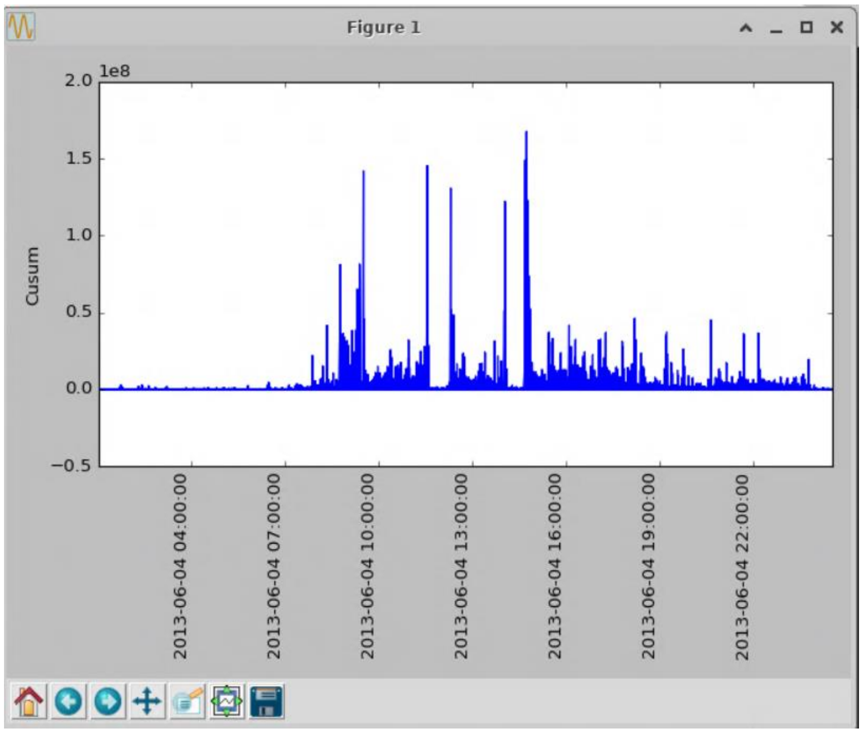
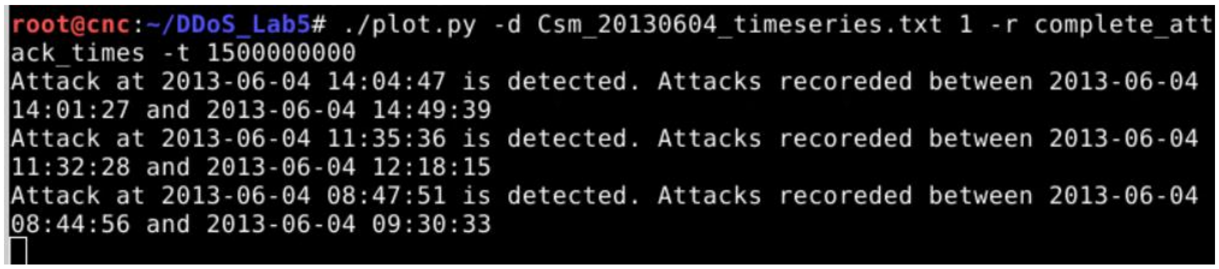


Figure 10: CUSUM time-series

Now we set a good threshold value accordingly at 1500000000 such that the attack is detected if the CUSUM crosses the threshold (Figure: 11, Figure: 12). I use the following command to set the threshold value:

./plot.py -d Csm_20130604_timeseries.txt 1 -r complete_attack_times -t 1500000000



```
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_20130604_timeseries.txt 1 -r complete_att
ack_times -t 1500000000
Attack at 2013-06-04 14:04:47 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
Attack at 2013-06-04 11:35:36 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 08:47:51 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
```

Figure 11: Attacks detected using CUSUM thresholding

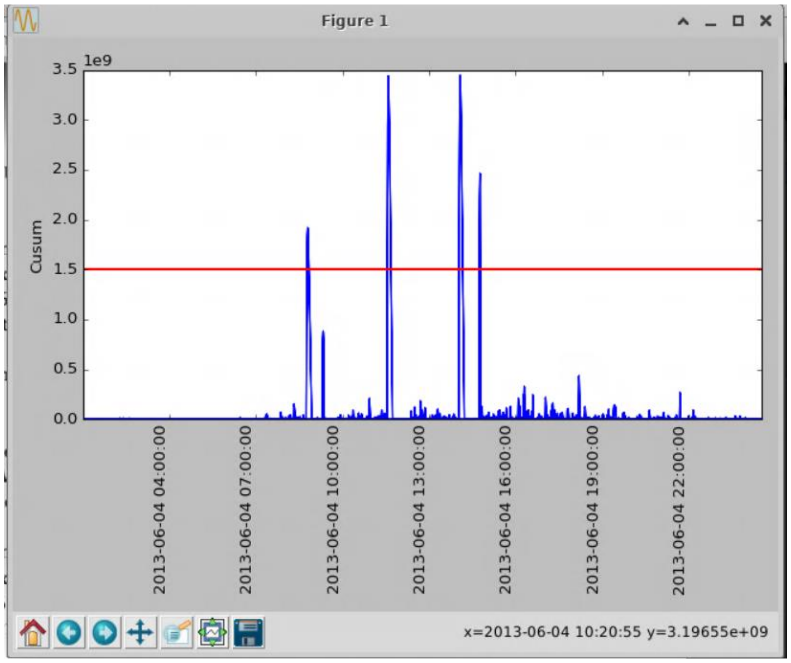


Figure 12: Threshold set at 1500000000

Now, we plot the ROC curve for the detection. I used the following command to plot the ROC curve:

./plot.py -d Csm_20130604_timeseries.txt 1 -r complete_attack_times -c

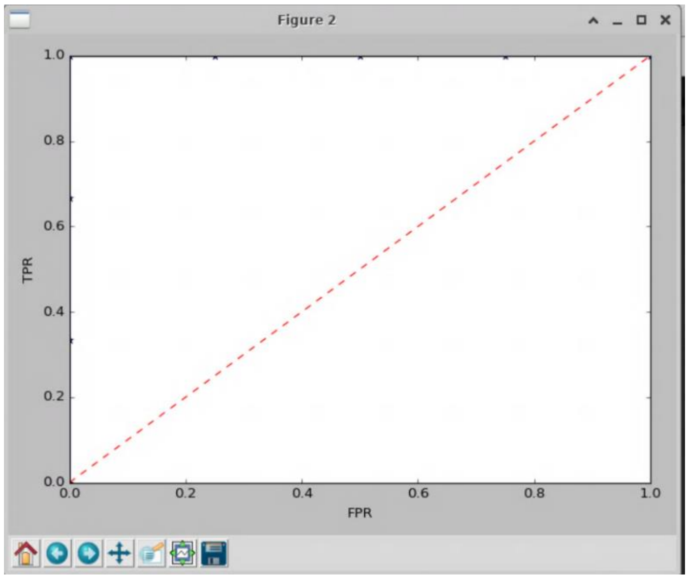


Figure 13: ROC curve

Moving on to performing Wavelet of the CUSUM detection. I executed *./vda.py -w1 timeseries.txt 2 -alpha 0.22 -epsilon 0.98811 -ce 0.13 -depth 3* for performing wavelet analysis on the “rx_byte” column in the file “timeseries.txt” to generate an intermediate file “Wvl_20130604_timeseries.txt”

On this we get the time-series graph as below (Figure: 14) using the following command:

./plot.py -d "Csm_20130604_timeseries.txt 1

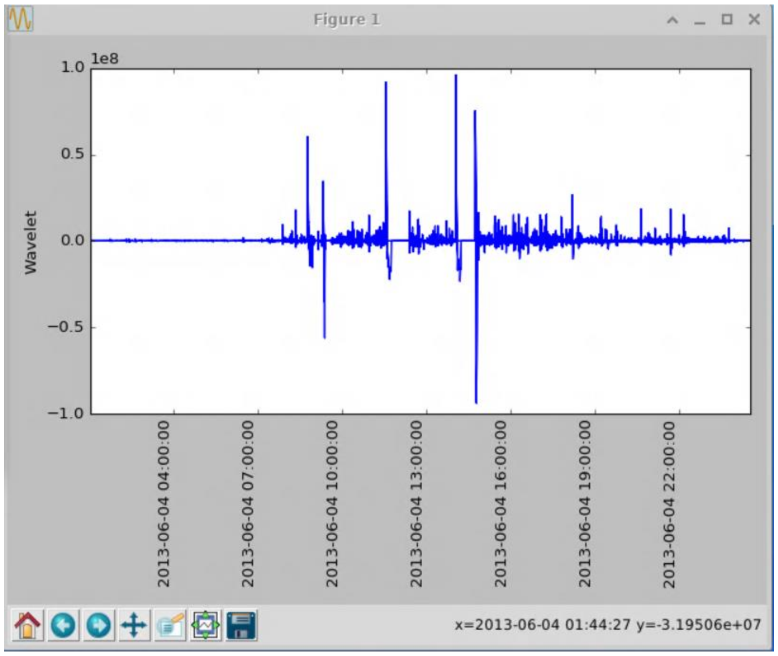
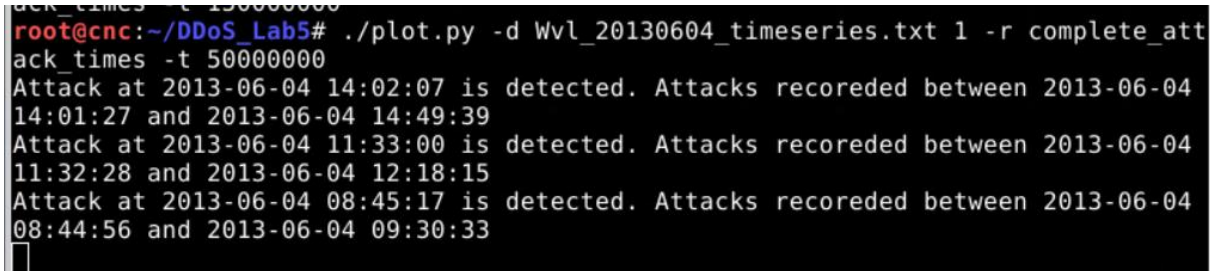


Figure 14: Wavelet time-series

Now we set a good threshold value accordingly at 50000000 such that the attack is detected if the wavelet crosses the threshold (Figure: 15, Figure: 16). I use the following command to set the threshold value:

./plot.py -d Wvl_20130604_timeseries.txt 1 -r complete_attack_times -t 1500000000

Here, the attack is detected using a high pass filter when we used the switch "-t".



```
root@cnc:~/DDoS_Lab5# ./plot.py -d Wvl_20130604_timeseries.txt 1 -r complete_att
ack_times -t 50000000
Attack at 2013-06-04 14:02:07 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
Attack at 2013-06-04 11:33:00 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 08:45:17 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
```

Figure 15: Attacks detected using low pass filter thresholding

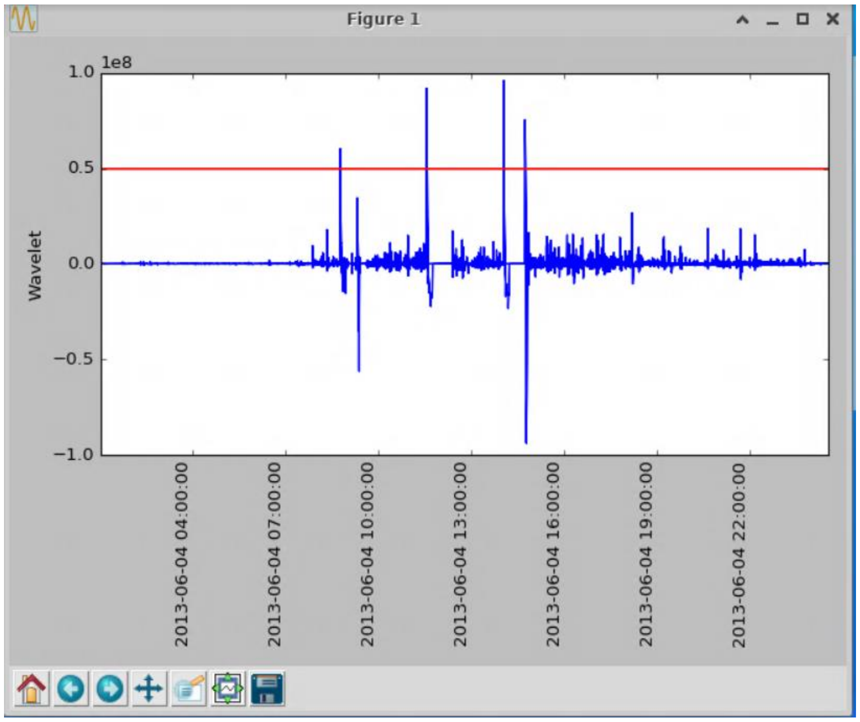
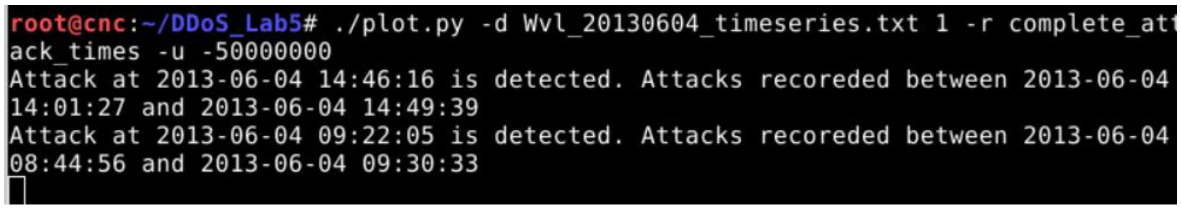


Figure 16: Threshold set at 500000000

Now, using the "-u" switch in the above command, we detect the attack with a low pass filter (Figure: 17, Figure: 18). The command I used is as follows:

./plot.py -d Wvl_20130604_timeseries.txt 1 -r complete_attack_times -u -500000000



```
root@cnc:~/DDoS_Lab5# ./plot.py -d Wvl_20130604_timeseries.txt 1 -r complete_att
ack_times -u -50000000
Attack at 2013-06-04 14:46:16 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
Attack at 2013-06-04 09:22:05 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
```

Figure 17: Attacks detected using low pass filter thresholding

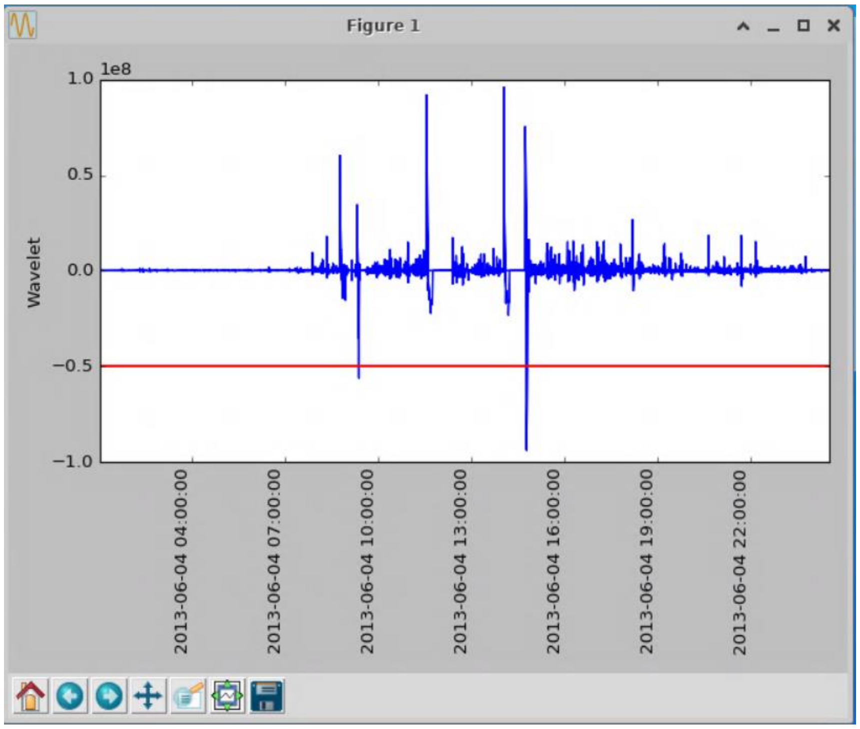


Figure 18: Threshold set at -500000000

Now, we plot the ROC curve for the detection. I used the following command to plot the ROC curve:

```
./plot.py -d Wvl_20130604_timeseries.txt 1 -r complete_attack_times -c
```

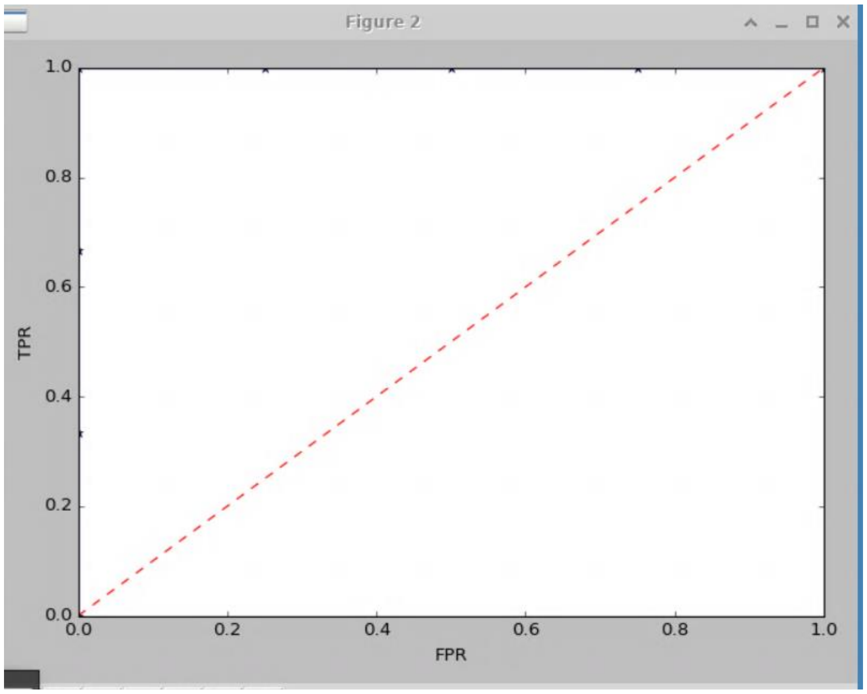


Figure 19: ROC curve

Now, we perform CUSUM of the Wavelet Detection, here we execute *./vda.py -w2 timeseries.txt 2 -alpha 0.0 -ce 0.5* on the time series data file in order to perform CUSUM analysis on the "rx_byte" column to generate an intermediate file "Csm_salem_20130604_timeseries.txt".

On this we get the time-series graph as below (Figure: 20) using the following command:
./plot.py -d Csm_salem_20130604_timeseries.txt 1

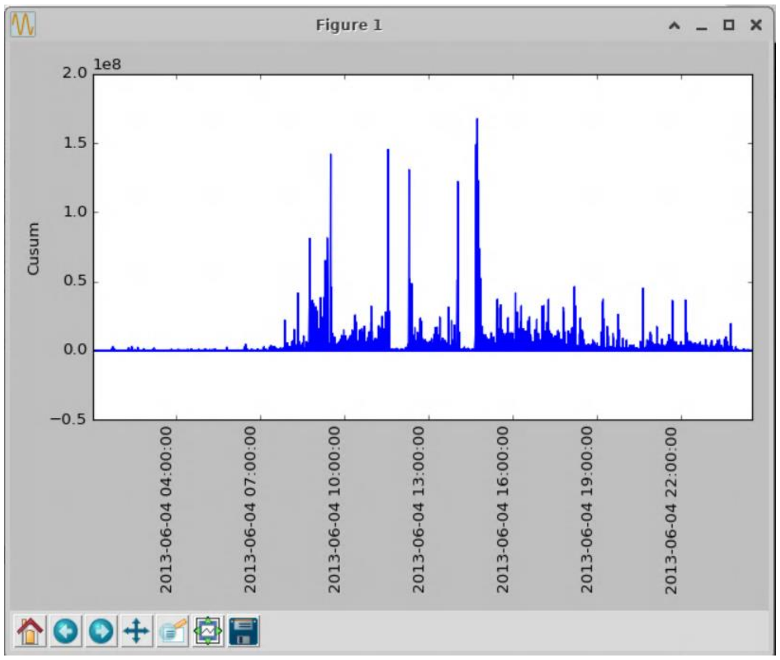
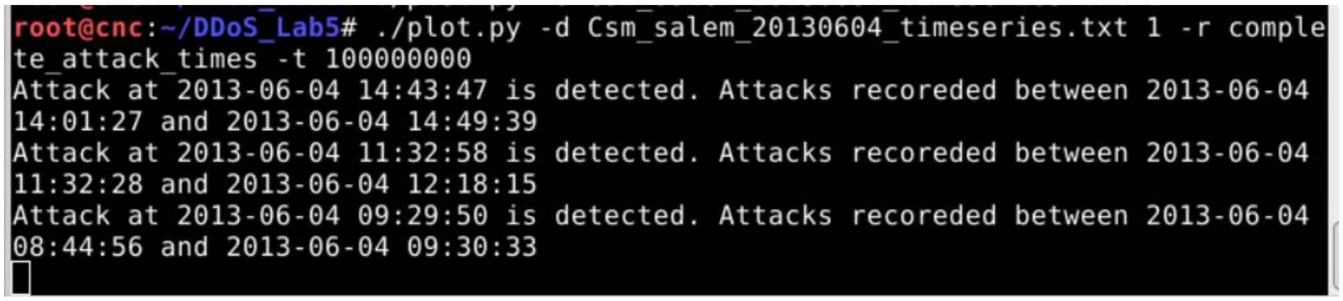


Figure 20: Wave-CUSUM time-series

Now we set a good threshold value accordingly at 100000000 such that the attack is detected if the wave-CUSUM time series crosses the threshold (Figure: 21, Figure: 22). I use the following command to set the threshold value:

./plot.py -d Csm_salem_20130604_timeseries.txt 1 -r complete_attack_times -t 100000000



```
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_salem_20130604_timeseries.txt 1 -r comple
te_attack_times -t 100000000
Attack at 2013-06-04 14:43:47 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
Attack at 2013-06-04 11:32:58 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 09:29:50 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
```

Figure 21: Attacks detected using Wave-CUSUM thresholding

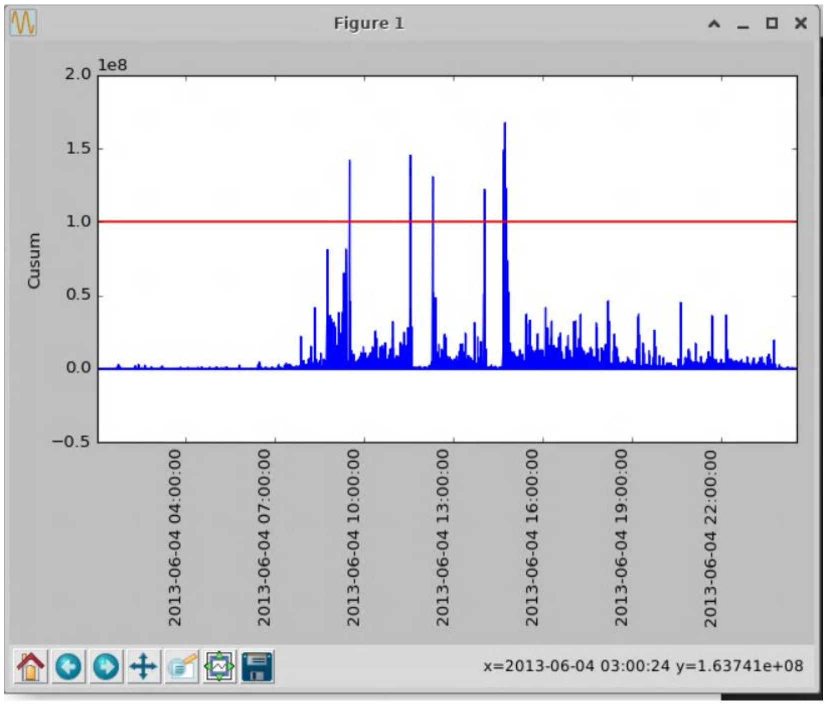


Figure 22: Threshold set at 100000000

Now, we plot the ROC curve for the detection. I used the following command to plot the ROC curve:

```
./plot.py -d Csm_salem_20130604_timeseries.txt 1 -r complete_attack_times -c
```

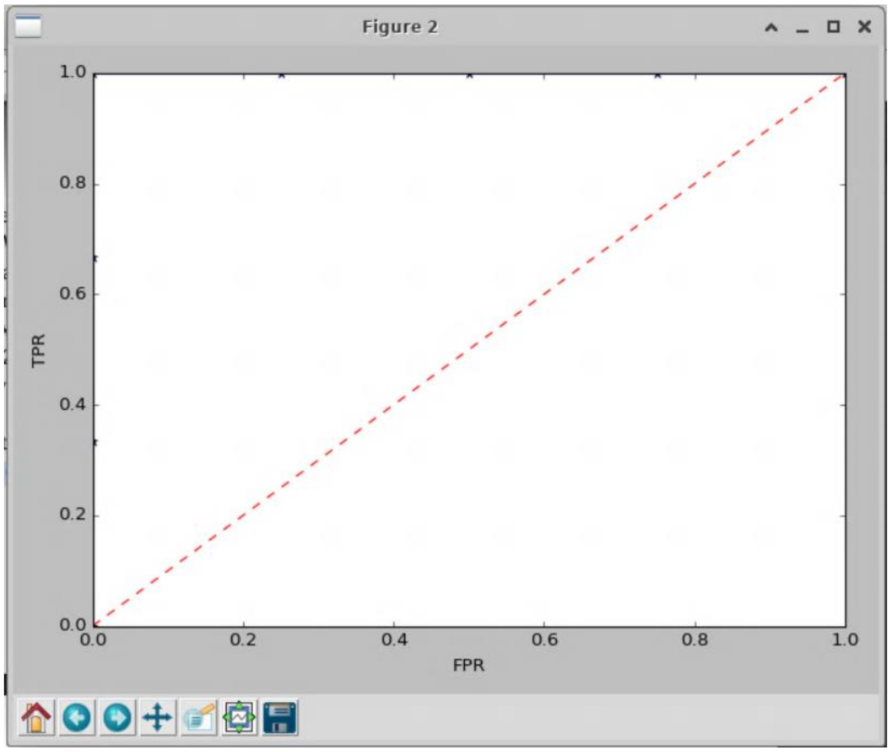
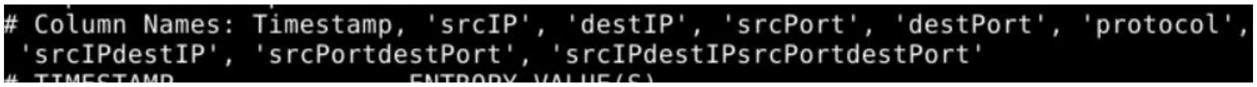


Figure 23: ROC curve

Coming to Entropy Based Detection, since the preprocessing of the intermediate file for the entropy detection takes a very long time. We use the "OutputTime0604.entr" which is already generated to carry out our detection.

The available columns by which we could plot the time series graph for this is shown below (Figure:24):



```
# Column Names: Timestamp, 'srcIP', 'destIP', 'srcPort', 'destPort', 'protocol',
 'srcIPdestIP', 'srcPortdestPort', 'srcIPdestIPsrcPortdestPort'
```

Figure 24: Available columns

First, lets do the detection with column as srcIP. Here is the time series graph for it (Figure: 25) which I got using the command *./plot.py -d outputTime0604.entr 1*

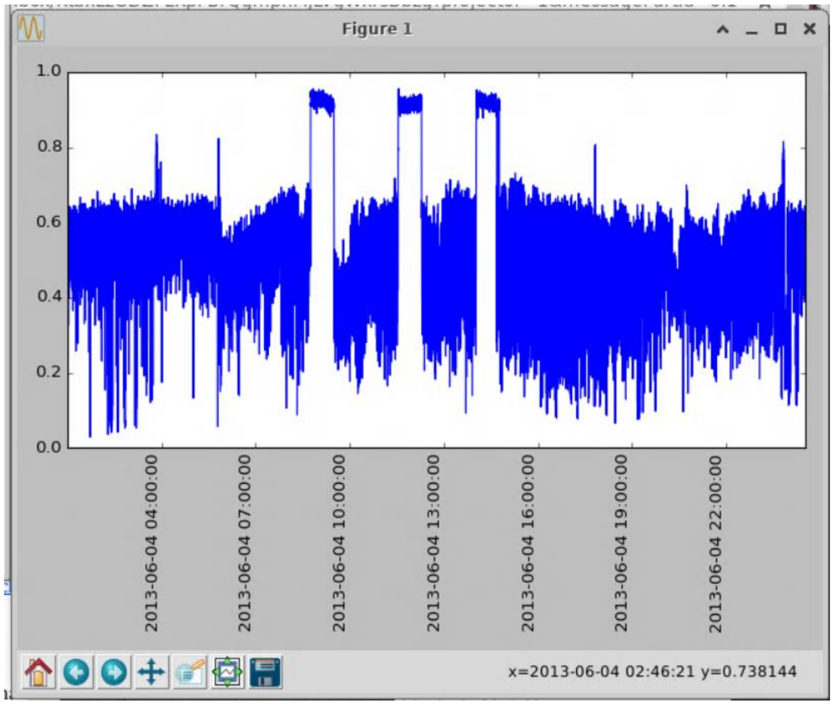
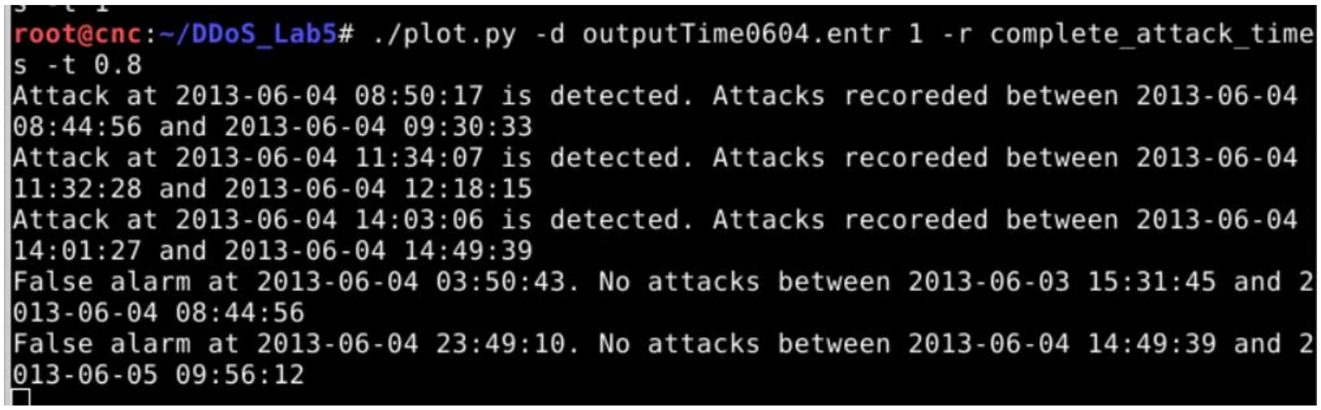


Figure 25: Entropy based time-series with "srcIP" as y-axis

Now we set a good threshold value accordingly at 0.8 such that the attack is detected if the Entropy time series crosses the threshold (Figure: 26, Figure: 27). I use the following command to set the threshold value:

./plot.py -d outputTime0604.entr 1 -r complete_attack_times -t 0.8



```
root@cnc:~/DDoS_Lab5# ./plot.py -d outputTime0604.entr 1 -r complete_attack_time
s -t 0.8
Attack at 2013-06-04 08:50:17 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
Attack at 2013-06-04 11:34:07 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 14:03:06 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
False alarm at 2013-06-04 03:50:43. No attacks between 2013-06-03 15:31:45 and 2
013-06-04 08:44:56
False alarm at 2013-06-04 23:49:10. No attacks between 2013-06-04 14:49:39 and 2
013-06-05 09:56:12
```

Figure 26: Attacks detected using Entropy thresholding

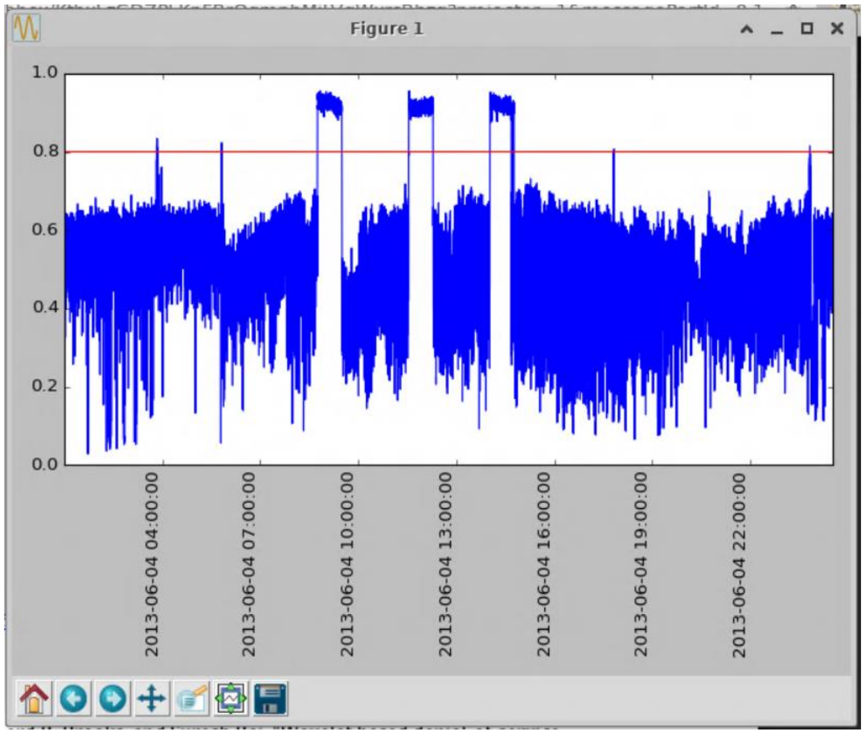


Figure 27: Threshold set at 0.8

Now, we plot the ROC curve for the detection. I used the following command to plot the ROC curve:

./plot.py -d outputTime0604.entr 1 -r complete_attack_times -c

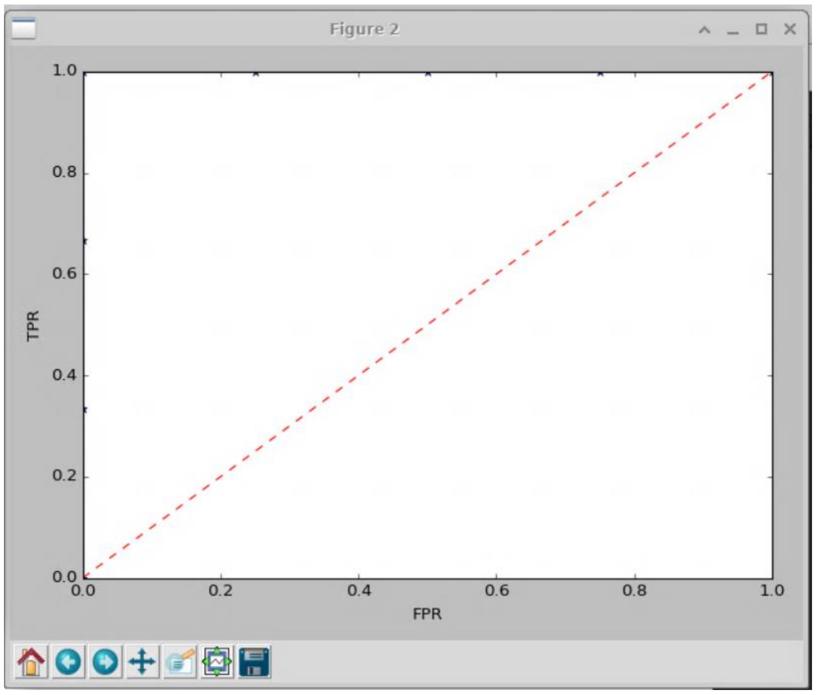


Figure 28: ROC curve

Now, let's try with a different column from "outputTime0604.entr" file. Let's take "srcIPdestIP" which can be accesses with "6" if we look back at the available columns figure.

We first plot the time series graph using *./plot.py -d outputTime0604.entr 6* (Figure: 29)

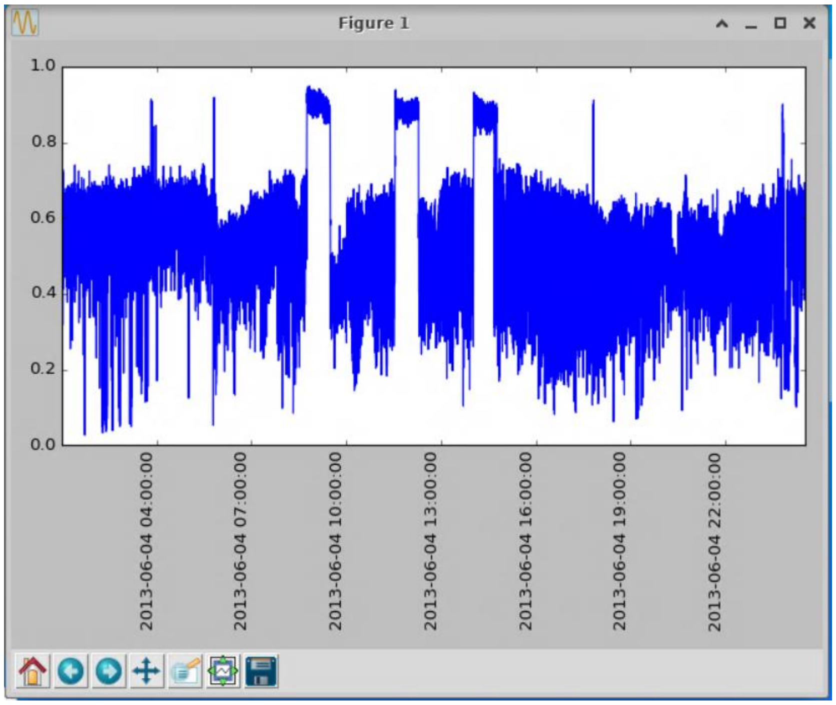
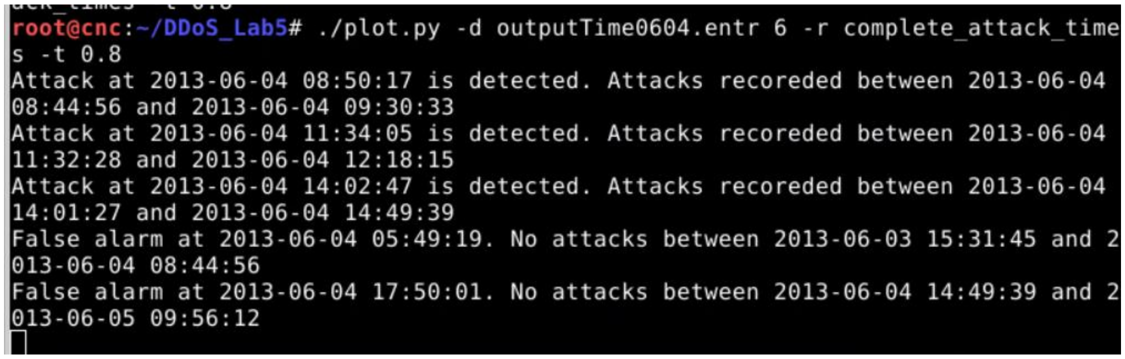


Figure 29: Entropy based time-series with "srcIPdestIP" as y-axis

Now we set a good threshold value accordingly at 0.8 such that the attack is detected if the Entropy time series crosses the threshold (Figure: 30, Figure: 31). I use the following command to set the threshold value:

./plot.py -d outputTime0604.entr 6 -r complete_attack_times -t 0.8



```
root@cnc:~/DDoS_Lab5# ./plot.py -d outputTime0604.entr 6 -r complete_attack_time
s -t 0.8
Attack at 2013-06-04 08:50:17 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
Attack at 2013-06-04 11:34:05 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 14:02:47 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
False alarm at 2013-06-04 05:49:19. No attacks between 2013-06-03 15:31:45 and 2
013-06-04 08:44:56
False alarm at 2013-06-04 17:50:01. No attacks between 2013-06-04 14:49:39 and 2
013-06-05 09:56:12
```

Figure 30: Attacks detected using Entropy thresholding

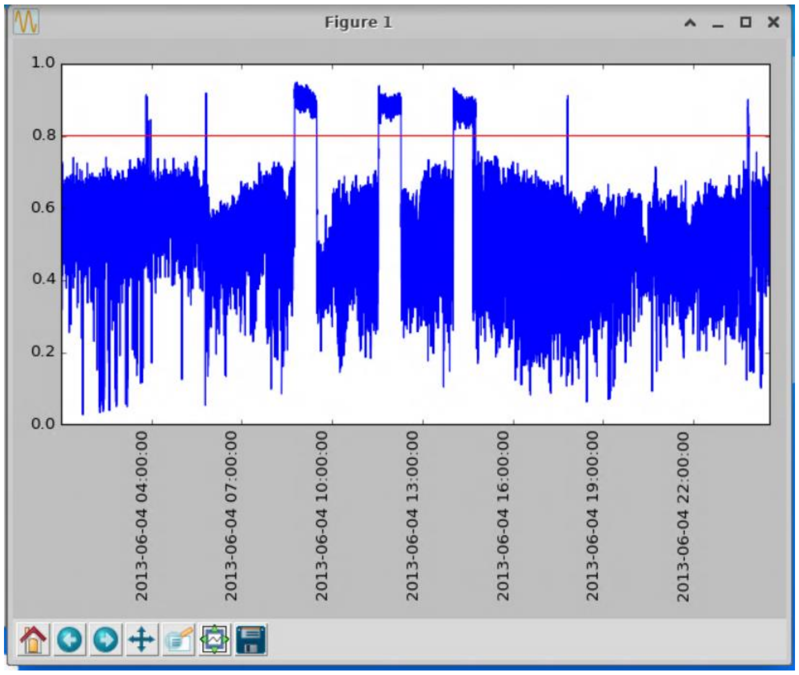


Figure 31: Threshold set at 0.8

Now, we plot the ROC curve for the detection. I used the following command to plot the ROC curve:

./plot.py -d outputTime0604.entr 6 -r complete_attack_times -c

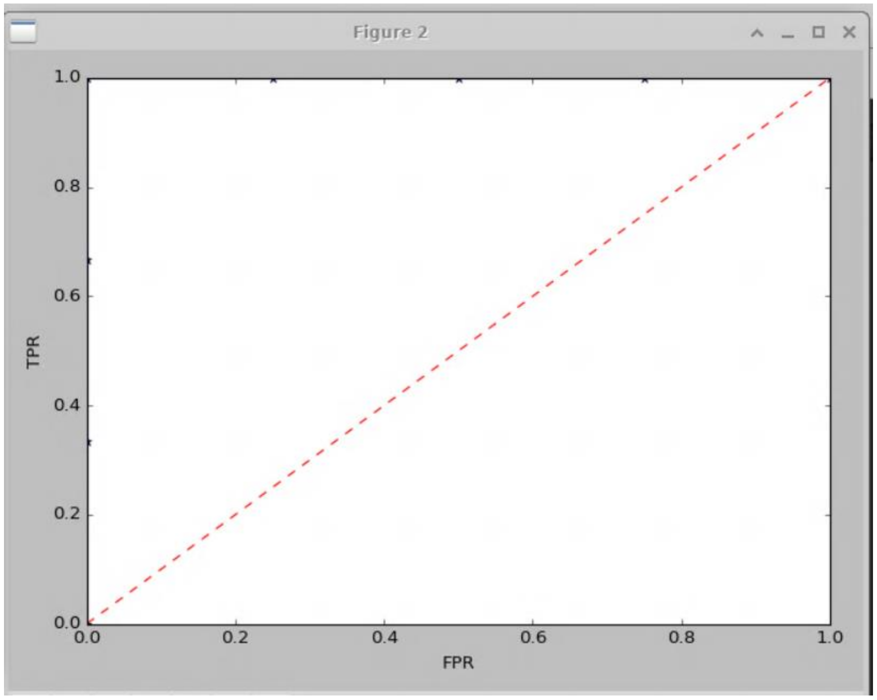


Figure 32: ROC curve

We observe that, the ROC curve for all the detection types are same and are a perfect ROC curve since the attack log used for all these detection types are same.

Questions

1. **What detection methods work well? Why**

   CUSUM and entropy methods can be considered to be efficient comparing to the other two. This is because, the detection accuracy is more and the false-positive rate for these detections are lower and also since these two detection methods monitors the network for a period of time instead of making judgment soon after detecting abnormal network condition. These detection methods observe whether the abnormal network condition persistently lasts for a certain period. Which is proven to be more effective and decrease the false-positives.

2. **How would you try to avoid being detected?**

   If I was the attacker, I would secretly produce flooding attack so as to simulate the expected normal data flow and know some packet attributes' entropy values. Once the normal entropy values range is known, I would produce the flooding attack using the attack tool with adjustable entropy values such that the values match the range. By doing this, my attack would go undetected by entropy-based detection since there is no randomization in the entropy values of the packets sent.

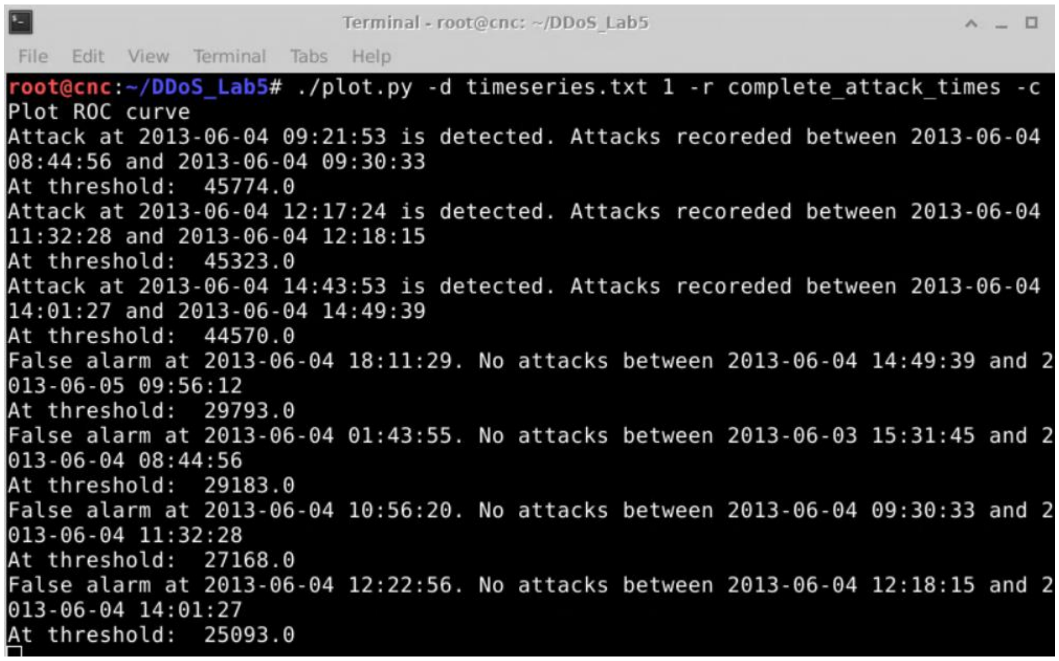
3. **Which method detects more quickly?**

   If we just take into consideration about the time taken for detecting a DDoS attack. Then, traffic volume detection using packets count thresholding or data volume thresholding detects an attack more quickly when compared to CUSUM, wavelet and entropy detection. This is because the traffic volume detection solely works on the principle of alerting when the packet count/size increases abnormally and goes over the threshold value. While the other methods monitor the network for a period of time instead of making a judgment soon after detecting abnormal network condition. They judge the network condition by observing whether the abnormal network condition persistently lasts for a certain period. Hence taking time. So, considering time as the factor, traffic volume detection method detects more quickly.

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

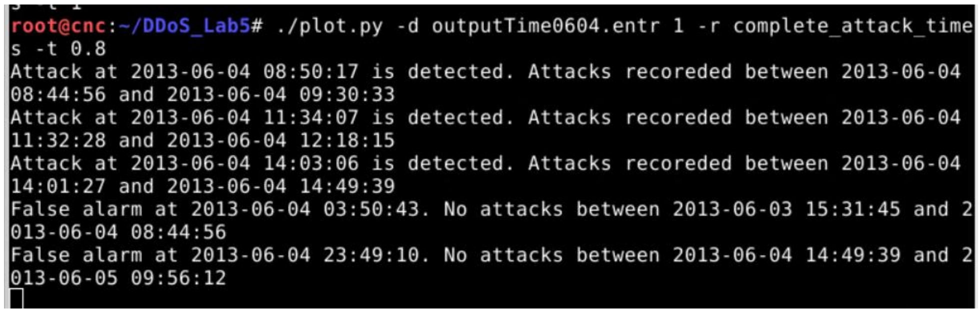
   All the detection methods used had false alarms to some extent, depending on the threshold value given, it can be said that the false alarm rate increases if the threshold value is unreasonably low. For a good value given for each detection type. Here are my ratios of true detect/false alarm for my results.

   - 3:4 for Traffic Volume Detection using packet count thresholding


Terminal - root@cnc: ~/DDoS_Lab5
File Edit View Terminal Tabs Help

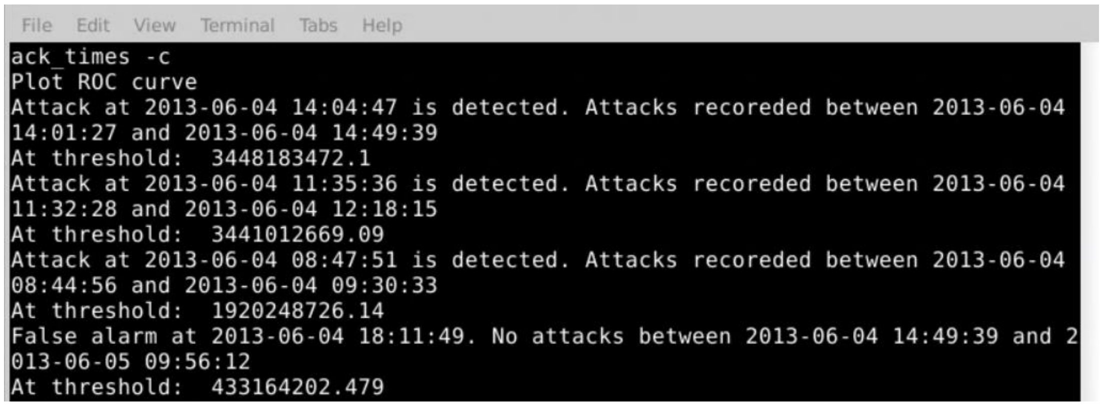
```
root@cnc:~/DDoS_Lab5# ./plot.py -d timeseries.txt 1 -r complete_attack_times -c
Plot ROC curve
Attack at 2013-06-04 09:21:53 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
At threshold:  45774.0
Attack at 2013-06-04 12:17:24 is detected. Attacks recoreded between 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
013-06-04 14:01:27
At threshold:  25093.0
```

   - 3:2 for Entropy Based Detection



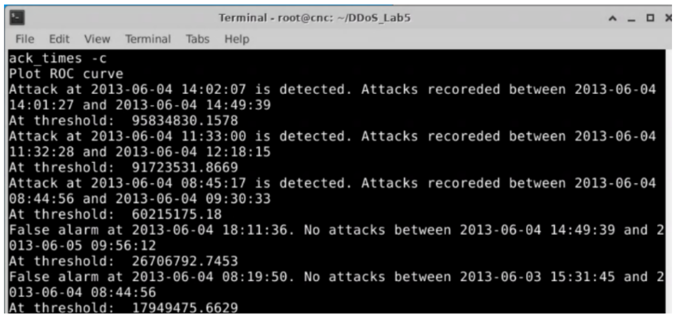
```
root@cnc:~/DDoS_Lab5# ./plot.py -d outputTime0604.entr 1 -r complete_attack_time
s -t 0.8
Attack at 2013-06-04 08:50:17 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
Attack at 2013-06-04 11:34:07 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
Attack at 2013-06-04 14:03:06 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
False alarm at 2013-06-04 03:50:43. No attacks between 2013-06-03 15:31:45 and 2
013-06-04 08:44:56
False alarm at 2013-06-04 23:49:10. No attacks between 2013-06-04 14:49:39 and 2
013-06-05 09:56:12
```

- 3:1 for CUSUM based detection



```
ack_times -c
Plot ROC curve
Attack at 2013-06-04 14:04:47 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
At threshold:  3448183472.1
Attack at 2013-06-04 11:35:36 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
At threshold:  3441012669.09
Attack at 2013-06-04 08:47:51 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
At threshold:  1920248726.14
False alarm at 2013-06-04 18:11:49. No attacks between 2013-06-04 14:49:39 and 2
013-06-05 09:56:12
At threshold:  433164202.479
```

- 3:2 for Wavelet of the CUSUM detection


Terminal - root@cnc: ~/DDoS_Lab5
File Edit View Terminal Tabs Help

```
ack_times -c
Plot ROC curve
Attack at 2013-06-04 14:02:07 is detected. Attacks recoreded between 2013-06-04
14:01:27 and 2013-06-04 14:49:39
At threshold:  95834830.1578
Attack at 2013-06-04 11:33:00 is detected. Attacks recoreded between 2013-06-04
11:32:28 and 2013-06-04 12:18:15
At threshold:  91723531.8669
Attack at 2013-06-04 08:45:17 is detected. Attacks recoreded between 2013-06-04
08:44:56 and 2013-06-04 09:30:33
At threshold:  60215175.18
False alarm at 2013-06-04 18:11:36. No attacks between 2013-06-04 14:49:39 and 2
013-06-05 09:56:12
At threshold:  26706792.7453
False alarm at 2013-06-04 08:19:50. No attacks between 2013-06-03 15:31:45 and 2
013-06-04 08:44:56
At threshold:  17949475.6629
```

References

[1]: https://link.springer.com/content/pdf/10.1007%2F978-3-540-77048-0_35.pdf