Table of Contents

ECE/CPSC 8860 Lab 5

This document is the DDoS Attack detection assignment description as well as the lab guide. This guide includes four chapters including this chapter. In this chapter, an introduction and lab setup is provided. In the second chapter, a quick start guide is provided for the lab session. The third chapter presents the guide to preprocess the pcap files, and the last chapter gives the questions.

Because some of the detection methods, e.g., wavelet detection and entropy detection, requires several hours to capture traffic and preprocess the pcap files. We preprocessed previously captured data and created the intermediate files for you to save some time. Since you already learnt how to capture traffic and perform attacks, there is no need to do it again for this lab. However, in case you are interested, the third chapter provides the guide to create the intermediate files from the pcap files.

The cusum, wavelet, and entropy detection script were implemented by Ilker during his Ph.D. study at Clemson. The code has been tested in the lab. However, if you find any problem using the code, please contact Ilker and/or Xingsi.

Quick Guide to Four Detection Methods

In this section, detailed examples will quickly show you how to perform the four types of detection using preprocessed intermediate data files.

- **Important**

- When copy any code, don't copy the following mentioned symbols at the beginning of each line.
- "$" precedes Linux commands that are typed at a regular user's shell prompt, usually your host machine
- "#" precedes Linux commands that are typed at a root shell prompt, usually in a VM
- ">>>" precedes Python or Scapy commands

Step 1. Import the VM

- The commands in this step should be executed on the VM.

Once finished, the DDoS_Lab6 VM will be available in your VirtualBox.
Start the Lab6 VM from VirtualBox. Login the Lab6 VM using the following credentials:
username: root
password: private123

Execute the following line in the VM

```
root@cnc:~# pip install PyWavelets
root@cnc:~# pip install pyshark
```

All scripts used in this lab are stored in directory "/root/DDoS_Lab5". Enter the directory by the following command:

```
root@cnc:~# cd /root/DDoS_Lab5
root@cnc:~/DDoS_Lab5#
```

Step 2. Traffic Volume Detection

- The commands in this step should be executed in the Lab6 VM.

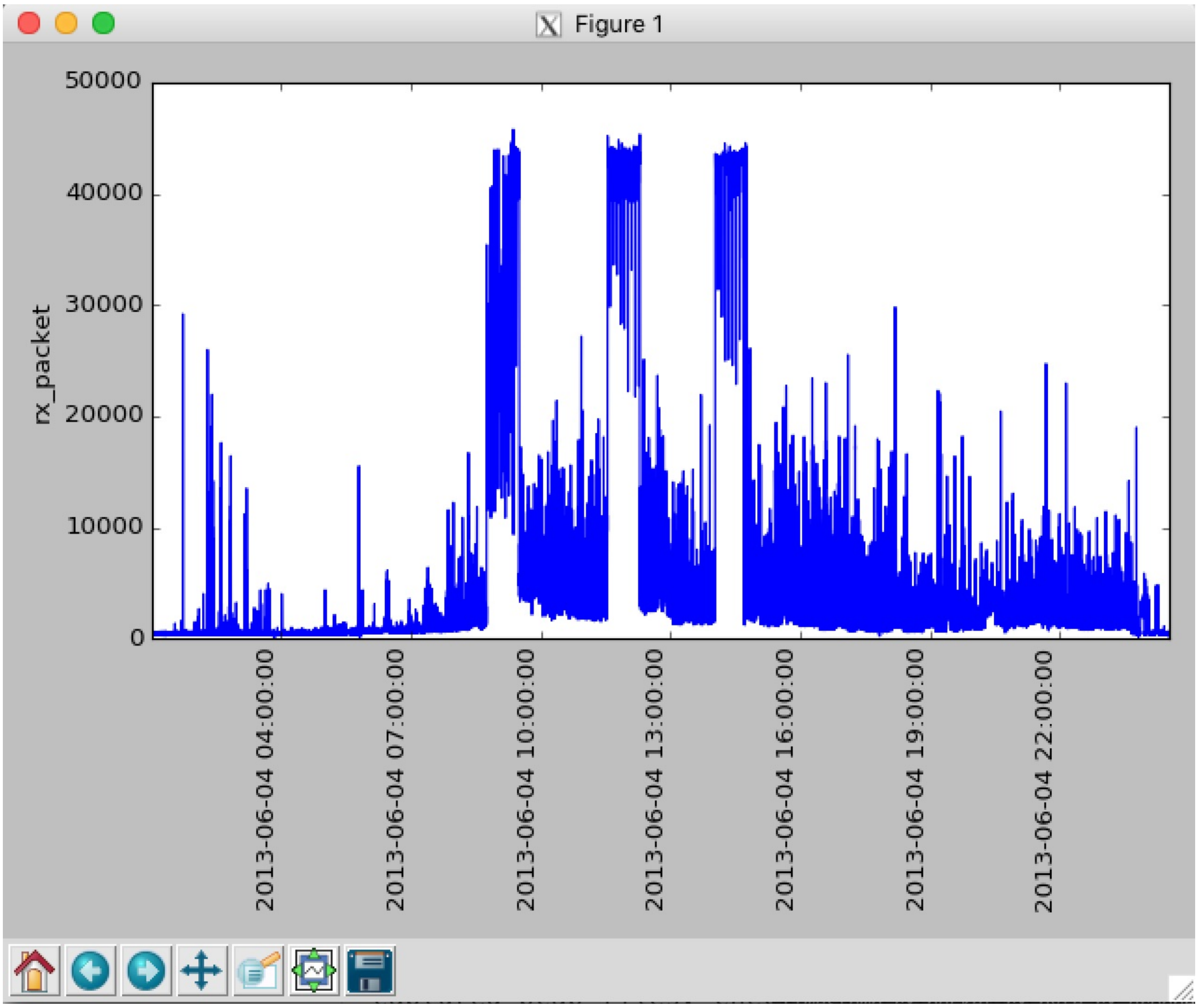
This method simply use a threshold on the traffic timeseries to decide whether there is a DDoS attack.

The file "timeseries.txt" is generated from previous captured pcap files. Check the next chapter to see how to generate this file.

Step 2.1. Packets Count Thresholding

First, plot the packets count timeseries and get a rough idea about the traffic. The following command plot the "rx_packet" column of the file "timeseries.txt"

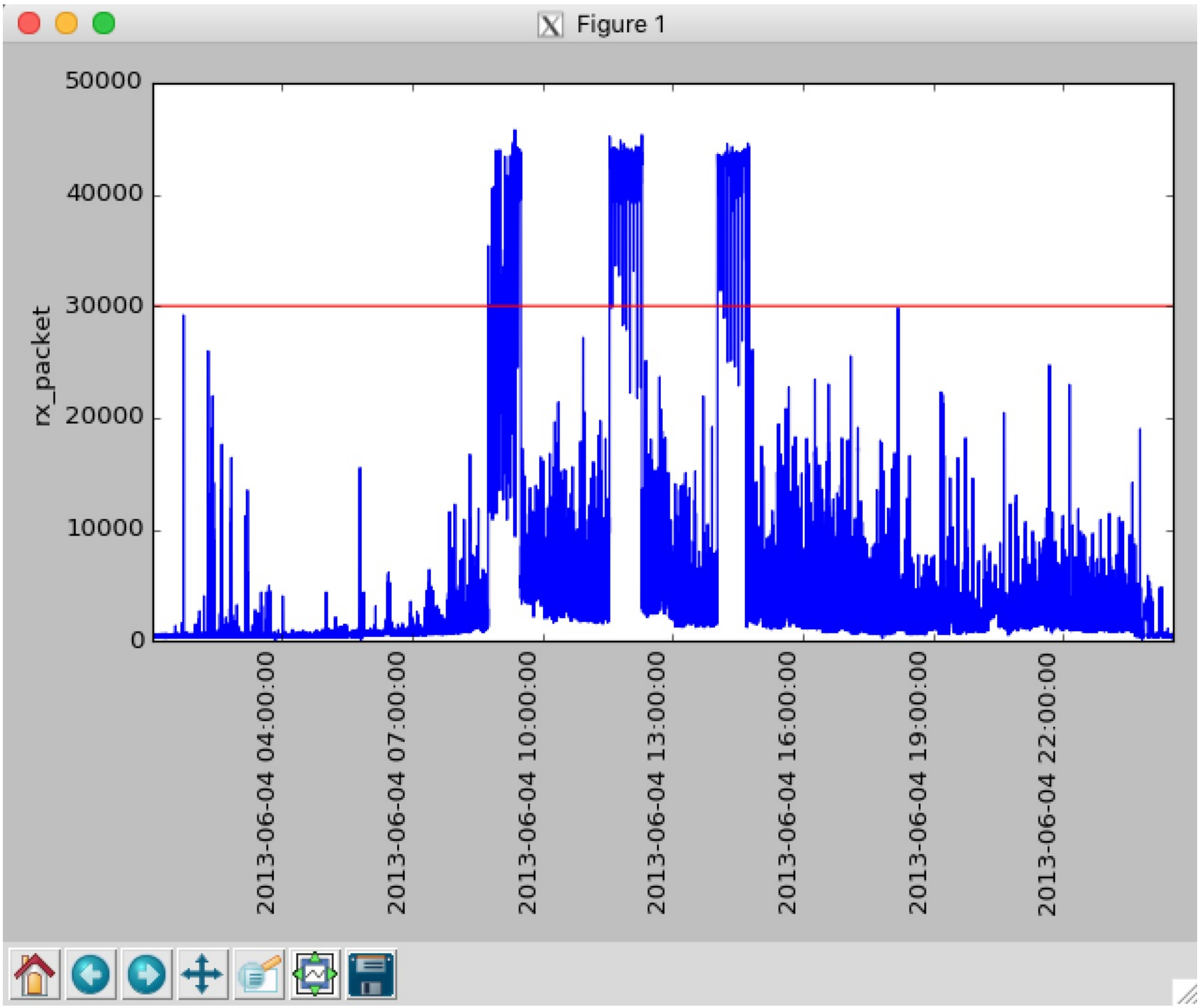
```
root@cnc:~/DDoS_Lab5# ./plot.py -d timeseries.txt 1
```



The attack time tags for this data set is logged in file "complete_attack_times". Check the next chapter to see how to generate this file.

Use option "-r" and "-t" to determine a good threshold. The following command uses 30000 as the threshold:

```
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
```

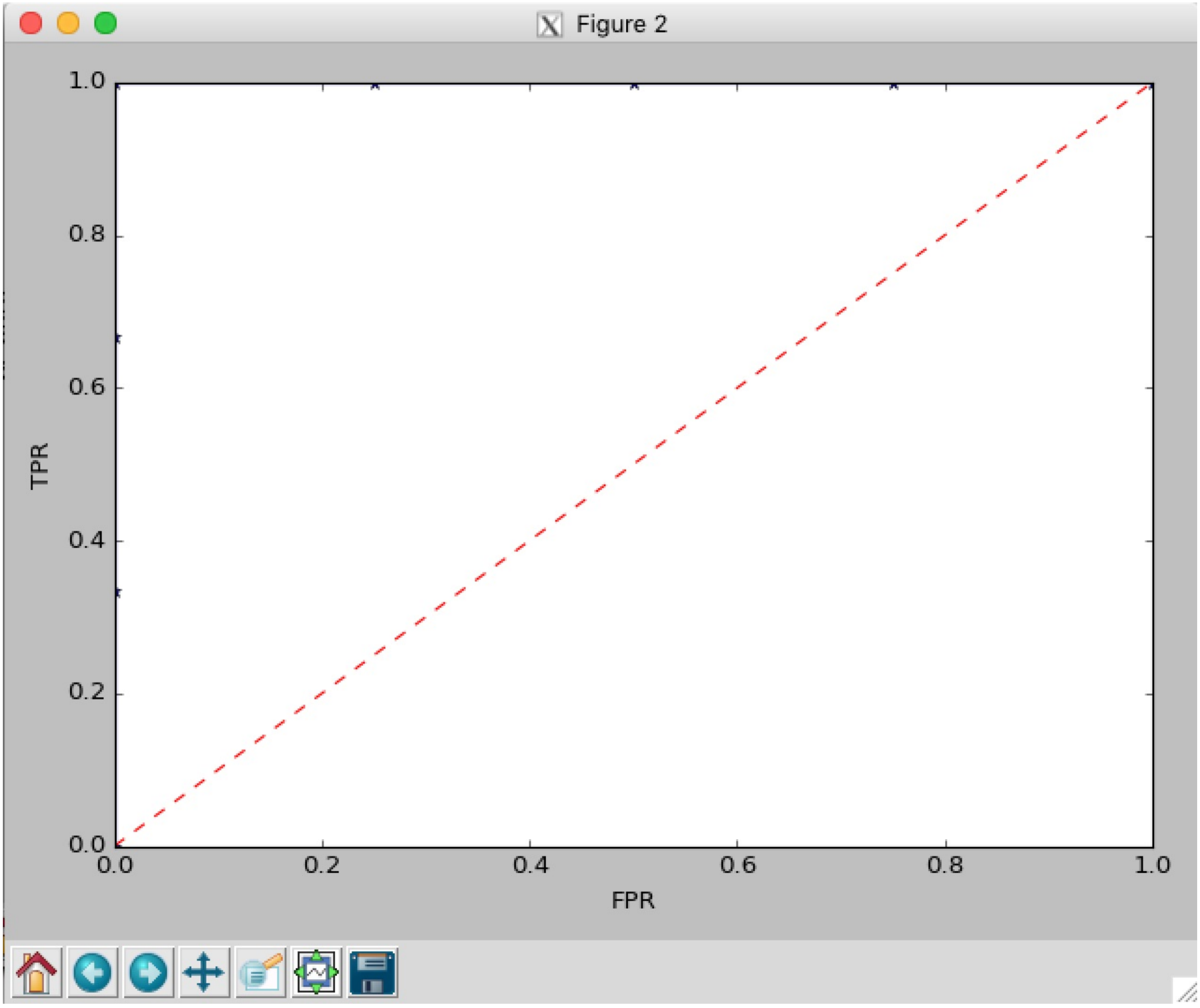


Use option "-c" to plot the ROC curve automatically

For this attack log, you can get a perfect ROC curve:

```
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 2013-06
-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 2013-06
-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 2013-06
-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 2013-06
-04 14:01:27
At threshold:  25093.0
```

The ROC curve will looks like the following:



Please note that the automated ROC function assumes that the filter is a high pass filter. In another word, any data point that is higher than a threshold is considered as "Positive". This function does not work for cases where a low pass filter or a band filter is needed.

Step 2.2. Data Volume Thresholding

Repeat all steps in Step 2.1. using the "rx_byte" column instead.

```
root@cnc:~/DDoS_Lab5# ./plot.py -d timeseries.txt 2
root@cnc:~/DDoS_Lab5# ./plot.py -d timeseries.txt 2 -r complete_attack_times -t 1000000

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

Step 3. CUSUM Detection

To perform CUSUM analysis, execute "vda.py" with "-c" option on the time series data file.

The following command perform CUSUM analysis on the "rx_byte" column in file "timeseries.txt" with the default parameters and generates an intermediate file "Csm_20130604_timeseries.txt":

```
root@cnc:~/DDoS_Lab5# ./vda.py -c timeseries.txt 2 -alpha 0.22 -epsilon 0.98811 -ce 0.13
```

Then perform the steps in step 2.1 to process the file "Csm_20130604_timeseries.txt":

```
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_20130604_timeseries.txt 1
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_20130604_timeseries.txt 1 -r complete_attack_times -t 1000000000
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_20130604_timeseries.txt 1 -r complete_attack_times -c
```

Step 4. Wavelet of the CUSUM Detection

To perform Wavelet analysis, execute "vda.py" with "-w1" option on the time series data file.

The following command perform wavelet analysis on the "rx_byte" column in file "timeseries.txt" with the default parameters and generates an intermediate file "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
```

Then perform the steps in step 2.1 to process the file "Wvl_20130604_timeseries.txt":

```
root@cnc:~/DDoS_Lab5# ./plot.py -d Wvl_20130604_timeseries.txt 1
root@cnc:~/DDoS_Lab5# ./plot.py -d Wvl_20130604_timeseries.txt 1 -r complete_attack_times -t 50000000
root@cnc:~/DDoS_Lab5# ./plot.py -d Wvl_20130604_timeseries.txt 1 -r complete_attack_times -c
```

You can try to modify the parameters when create the intermediate file to get better result.

Step 4.1. Apply a low pass filter

You can use "-u" option instead of "-t" option to detect the attack with a low pass filter. Every node below the given value is considered as "Positive":

```
root@cnc:~/DDoS_Lab5# ./plot.py -d Wvl_20130604_timeseries.txt 1 -r complete_attack_times -u -50000000
```

Step 5. CUSUM of the Wavelet Detection

To perform WAVE-CUSUM analysis, execute "vda.py" with "-w2" option on the time series data file.

The following command perform CUSUM analysis on the "rx_byte" column in file "timeseries.txt" with the default parameters and generates an intermediate file "Csm_salem_20130604_timeseries.txt":

```
root@cnc:~/DDoS_Lab5# ./vda.py -w2 timeseries.txt 2 -alpha 0.9 -ce 0.5
```

Then perform the steps in step 2.1 to process the file "Csm_salem_20130604_timeseries.txt":

```
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_salem_20130604_timeseries.txt 1
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_salem_20130604_timeseries.txt 1 -r complete_attack_times -t 50000000
root@cnc:~/DDoS_Lab5# ./plot.py -d Csm_salem_20130604_timeseries.txt 1 -r complete_attack_times -c
```

Step 6. Entropy Based Detection

The preprocessing of the intermediate file for the entropy detection takes a very long time. In this step, we preprocessed the pcap files and generated the intermediate "fileoutputTime0604.entr" for you. In your report, you could use this ".entr" file. You could also check the last chapter for how to generate ".entr" files.

There are multiple columns in the "outputTime0604.entr" file. Check the available columns by:

```
root@cnc:~/DDoS_Lab5# head outputTime0604.entr
```

Try with different columns, e.g. plot the "srcIP" column of the entropy file with:

```
root@cnc:~/DDoS_Lab5# ./plot.py -d outputTime0604.entr 1
```

Or plot the "destPort" column of the entropy file with:

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

Then you can use the "plot.py" script to check the detection rate like those in the previous steps.

Log and Preprocessing

- This chapter provides instructions to log the attack and preprocess the captured pcap files.
- The following steps are not required for this lab.

1. Attack and Log

1.1. Capture traffic on the victim VM's host machine:

Capture the attack traffic before the attack.

You should capture as long as possible(e.g. 10 hours or more). Some detection script will give you error if the time series is too short.

Execute the following command from any host machine to login to the host of the victim VM. Replace username with your username:

```
$ ssh username@192.168.10.9
```

Then capture traffic on the remote host. The mac address in the filter should be the mac address of the host machine(192.168.10.9).

Reduce the file size parameter "filesize:100000" to "filesize:10000" if the detection script crashes when process the pcap files.

```
$ tshark -f 'not ether host {MAC addr}' -s 90 -i enp3s1 -b filesize:100000 -w campus_m
ar14.pcap -a files:10 -F libpcap
```

Hit Ctrl + C to stop logging or wait until 10 files filled up.

1.2. Log attack times

Execute the following command from the DDoS_Lab5 VM to perform a DDoS attack and log the attack time:

```
root@cnc:~/DDoS_Lab5# cd /root/DDoS_Lab5/Appendix
```

Modify line 14 in file "attackNlog.sh" to suit your appetite.

Then execute this bash script.

```
root@cnc:~/DDoS_Lab5/Appendix# ./attackNlog.sh
```

The attack log is stored in file "Attacklog001.txt".

Modify line 7 in file "attackNlog.sh" to change the name if nessessary.
Please note that there might be time shift between your machine and the victim's host machine. You might have to adjust the time shift if necessary.

2. Generate the Timeseries File

Execute the following command from the DDoS_Lab5 VM to copy the pcap files back to the ~/DDoS_Lab5/Appendix/autoTest2 directory Kali_DDoS_Lab5 VM:

```
root@cnc:~/DDoS_Lab5/Appendix# scp username@192.168.10.8:~/campus_mar14*.pcap ./autoTest2
```

Generate the timeseries file by:

```
root@cnc:~/DDoS_Lab5/Appendix# ./pcap2timeseries.py
```

The output is "Time_Series.txt"

3. Generate the Entropy File

Generate the entropy file by:

```
root@cnc:~/DDoS_Lab5/Appendix# ./pcap2entropy.py
```

The output is "autoTest2.entr"

Questions:

1. What detection methods work well?
2. How would you try to avoid being detected?
3. Which method detects more quickly?
4. How often do you get false alarms from your results?

References:

[1] Glenn Carl, Richard R. Brooks, and Suresh Rai. 2006. Wavelet based Denial-of-Service detection. Comput. Secur. 25, 8 (November 2006), 600-615. [2] C. Callegari, S. Giordano, M. Pagano, T. Pepe, WAVE-CUSUM: Improving CUSUM performance in network anomaly detection by means of wavelet analysis, Computers & Security, Volume 31, Issue 5, July 2012, Pages 727-735, ISSN 0167-4048, [3] I. Ozcelik, Y. Fu and R. R. Brooks, "DoS Detection is Easier Now," 2013 Second GENI Research and Educational Experiment Workshop, Salt Lake City, UT, 2013, pp. 50-55. doi: 10.1109/GREE.2013.18