Table of Contents

ECE/CPSC 8860 Lab 6 Entropy Spoofing

This document is the DDoS attack entropy detection spoofing 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 files, and the last chapter gives the questions.

There are several data formats you need to be aware of before doing this lab. They are "pcap" files, end with ".pcap"; transcript files, end with ".trns"; histogram files, end with "hist"; and entropy files, end with ".entr". The attack record file is simply a text file. You should be familiar with some of them. In lab 5, we use "pcap2entropy.py" script to generate the entropy file from pcap files. That script actually generates temporary transcript files and histogram files but then removed those temporary files.

In Ilker's provided script, the entropy spoofing is performed based on the histogram files. That means the spoofing is not happing in real-time. The "entropy_spoofV3.py" script takes a histogram file and an attack record time file as input, and calculates the desired packets needs to be injected during the attacks, and generate a spoofed histogram file.

So the work follow for entropy spoofing would be: Perform some flood attack --> Capture the traffic and attack times --> Use pcap files to generate transcript files --> Use transcript files to generate histogram files --> Use the histogram files and attack record files to generate spoofed histogram files --> Use the spoofed histogram files to generate spoofed entropy files --> Plot the spoofed entropy files.

Whereas to generate non-spoofed entropy files, simply use the histogram files before the spoof to generate entropy files, then plot.

That looks simple isn't it.

However, every step could take several minutes to several hours to process.

So we take the time consuming steps out of the scope of grading. We keep the whole instructions in this document but you do not to perform the following steps to get full credit: the flood attack, capture traffic and attack times, and generate hist files from pcap files

The steps are not in the grading scope will be marked in the instructions as well.

In the quick guide section, we show how to plot the entropy files before and after the spoofing and spoof some preprocessed histogram files.

In the full guide, we show the complete process and roughly estimated time to perform each step. Please be aware that larger data set will take a longer time to process and the increase of time is not linear to the size of the input file. Please note that some of the code is very buggy, and it will be stopped with a spectacular crash. However, it will generate desired output before the crash. That is how research works.

The entropy spoofing script was 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

In this section, detailed examples will quickly show you how to plot the entropy files before and after spoofing. If you have time, you can also spoof the provided histogram 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 from the NAS:

Step 1.1. VM login:

- Copy Kali_DDoS_Lab6.ova to your local and import it.
- The commands in this step should be executed on the VM.

Start the Lab6 VM from VirtualBox. Login the Lab6 VM using the following credentials:
username: root
password: private123

All scripts used in this lab are stored in directory "/root/DDoS_Lab6". Enter the directory by the following command:

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

Step 2. Compare entropy files:

- The commands in this step should be executed on the VM.These are just to let you know

how to plot the entropy files before and after spoofing.First, plot the entropy file before spoofing:

```
root@cnc:~/DDoS_La6# ./plot.py -d outputTime0604.entr 1
```

Open another terminal, go to the same directory, then plot the entropy file after spoofing:

```
root@cnc:~/DDoS_La6# ./plot.py -d spoofed_1_outputTime0604.entr 1
```

Step 3. Spoof entropy lab:

The "**outputTime0604.hist**" is provided so you don't need to perform step 1 in the Full guide. You are supposed to **replace** the input file name "***autoTest.hist***" everywhere in Step 2 of the full guide with "***outputTime0604.hist***".

Full Guide

This chapter covers how to: use pcap files to generate histogram files; spoof the histogram files; and generate entropy files from histogram files.

* Step 1. Generate histogram files from pcap files:

*** You do not need to perform this step to get full credit.**

- The "outputTime0604.hist" is provided so you can use this file for the following steps.

This step takes pcap files in the autoTest folder as input and generate "autoTest.hist" file.

Before proceed, you need to have some pcap files in the "/root/DDoS_Lab6/Appendix/autoTest/" directory. Follow instructions in the previous labs to perform attacks and get pcap files.

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

All the preprocessing scripts are stored in directory "/root/DDoS_Lab6/Appendix/". Enter the directory by the following command:

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

Generate the histogram files by:

```
root@cnc:~/DDoS_Lab6/Appendix# ./pcap2hist.py
```

This step takes about 1 hour or more to finish depending on the size of the pcap files.

Step 2. Generate spoofed histogram file:

This step uses file "autoTest.hist" as the input file to generate "spoofed_1_autoTest.hist" file.

This step show how to spoof given histogram file The original histogram files and spoofed histogram files are provided as examples.

Execute the following command to spoof the given histogram file. Modify the scale value to whatever other value close to 1. This script will keep running and generate several output files and eventually crash. This script will take around 10 to 20 minutes to finish(Crash.:

```
root@cnc:~/DDoS_Lab6/Appendix# ./entropy_spoofV3.py autoTest.hist autoTest.hist.out co
mplete_attack_times 1.25 -c 1
```

You will see output similar to this:

```
Namespace(FA=None, attack_times='complete_attack_times', columns=[], ent=False, file_n
ame='autoTest.hist', margin=[0.15], output='autoTest.hist.out', scale='1.25', spoofPac
ketCount=False)
Copying autoTest.hist to temp.txt
Performing attack number 1
Average cycle: 5.95511450382
Average time: 0.000528254351145
Performing attack number 2
Average cycle: 5.81375166889
Average time: 0.000568560080107
Performing attack number 3
Average cycle: 5.98514517218
Average time: 0.000573065158677
('It took ', 339, ' seconds')
End of process!!
Copying spoofed_1_autoTest.hist to temp.txt
Performing attack number 1
Average cycle: 5.62683823529
Average time: 0.000446215992647
Performing attack number 2
Average cycle: 5.49799062291
Average time: 0.000533518084394
Performing attack number 3
Average cycle: 5.7324270557
Average time: 0.000679098474801
('It took ', 333, ' seconds')
End of process!!
```

This step takes about 15 minutes to finish(crash). Roughly 5 minutes for each sub-step. After this step finishes, you would have a new histogram file(spoofed).

Step 3. Generate entropy files from histogram files:

This step uses file "outputTime0604.hist" as input to generate the "one.entr" file and "outputTime06041.hist" as input to generate the "two.entr" file. These scripts take about 10 minutes each to generate the entropy files. So be patient!

```
root@cnc:~/DDoS_Lab6/Appendix# ./calculate_entropyV2.py outputTime0604.hist one
root@cnc:~/DDoS_Lab6/Appendix# ./calculate_entropyV2.py outputTime06041.hist two
```

The *.entr* files are the entropy files. *one.entr* is the original entropy; *two.entr* is the spoofed entropy.Then you can move the generated entropy files to the upper directory(DDoS_Lab6 dir)

```
root@cnc:~/DDoS_Lab6/Appendix# mv one.entr ..
root@cnc:~/DDoS_Lab6/Appendix# mv two.entr ..
```

After you get the "one.entr" and "two.entr" files, you can follow the instructions in the quick guide to plot these, i.e. use step 2 in the quick guide to plot these entropy files (replacing the names of the files with your entropy files)

Questions

1. Execute ./entropy_spoofV3.py script for three different values of the scale (but should be close to 1) and see if you can get better results.
2. Discuss the pros and cons of using entropy to detect DDoS attacks.