Signature Based Pattern Matching Over GPU
User Guide

Contents

1 Introduction

This project is a threat detection system over a network, using NVIDIA CUDA GPU to perform the signature matching. It is designed to deliver high throughput over a network interface, utilizing computational power of GPUs.

2 Prerequisites

1. Linux OS. [Preferably Ubuntu, Kernel 3.0+]
2. Server/System with an NVIDIA CUDA supported GPU installed. [SM Architecture 3.2+]
3. nVidia drivers, and CUDA toolkit installed.
4. Pthread and Pcap libraries available.
5. A network interface to monitor.
6. Root access to run the program.

3 Compiling the code

1. Gather all source code files in a folder.
   Note: 6 Source files include: config.cuh, kmp_gpu.cu, kmp_gpu.cuh, list.cuh, main.cu, StreamThreadPool.cuh, signatures.dat
2. Open a terminal and navigate to the source code folder root.
3. Type in the following:

   ```
   nvcc -o run main.cu kmp_gpu.cu -lpcap -lpthread
   ```

   Note: This step requires CUDA toolkit, the pcap and pthread libraries available on the system.
   This will produce an executable 'run' in the folder

4 Running the program

Note: The program requires a signature database file to run.
Continuing from compilation, let 'run' be the executable file name.

1. Open a terminal and navigate to where the executable 'run' is present.
2. Run the following commands in terminal:

```
sudo ./run
```

   On successful run, the terminal will show program stats and prompt for interface to monitor.

3. Input the Interface number to monitor, as shown in the terminal.
4. The program will monitor for threats and alert on terminal if any found.

4.1 Terminating the program

Press 'Ctrl+C' or send signal SIGINT to exit the program.

5 Configuring options for the program

The program settings can be configured. The methods are listed in order of least priority first (will be considered last).

5.1 Default Configuration

By default, the program has the following options configured:

- Algorithm: 1 [*'KMP'*]
- Streams: 4
- Threads: 512
- Blocks: 4
- Signature Patterns: '../signatures.dat'
- Log: Terminal Output
- Zero Copy: Enabled

5.2 Configuration File

To create a persisten configurations file:

1. Create a file: *config.cfg* where the program executable is located.
2. Put the options (listed below) with respectful values in the file:
    - streams=*(>0)*
    - algo=*("kmp", "rk", "ac")*
    - threads=*(1-2048)*
    - blocks=*(1-8)*
    - zerocopy=*('y'/'n')*
    - patterns=*("path_to_signatures/filename")*
    - logfile=*("path_to_logFile/filename")*

5.3 Command line arguments

Command line arguments are highest priority in settings. Will override everything else. Same options as in configurations file. Run the program as:

```
sudo ./run [options=value] ...
```

Following options are available

- –streams=*(>0)*
- –algo=*("kmp", "rk", "ac")*
- –threads=*(1-2048)*
- –blocks=*(1-8)*
- –zerocopy=*('y'/'n')*
- –patterns=*("path_to_signatures/filename")*
- –logfile=*("path_to_logFile/filename")*

6 Signatures

The signatures are provided with the package, in hexadecimal encoding. They are required to run the program.