CAN Network Traffic Analysis Tool (v 1.0)

> This tool is a tool for submitting to the "2017 CISC Data Challenge (http://challenge.cisc.or.kr)". It is a tool for determining the type of attack for a given CAN traffic data and visualizing it in real time.

이 도구는 "2017 CISC 데이터 챌린지 (http://challenge.cisc.or.kr)"에 제출하기 위한 도구입니다. 주어진 CAN 트래픽 데이터에 대한 공격 유형을 결정하고, 이를 실시간으로 시각화 합니다.

Getting Started

> The tool works in the local environment and displays the analysis in real time based on the dataset contents when the user uploads the dataset.

이 도구는 로컬 환경에서 작동하며 사용자가 데이터 셋을 업로드 할 때, 데이터 셋 내용을 기반으로 실시간으로 분석을 표시합니다.

Prerequisites

> It requires Python 2.7x version by default and requires installation for modules such as Flask, a lightweight web framework. Processing of visualization is done using HTML5, CSS3, javascript, Ajax, jQuery, etc. All separate installation files that require this are listed below.

기본적으로 Python 2.7x 버전을 요구하며, 경량화 웹 프레임워크인 Flask 와 같은 module 에 대한 설치를 필요로 합니다. 시각화에 대한 처리는 HTML5, CSS3, javascript, Ajax, jQuery 등을 활용하여 진행이 되며, 이이 필요한 모든 별도의 설치 파일은 아래에 기재되어 있습니다.

> The required Python library can be installed using pip, and libraries such as javascript including jQuery are configured to be referenced via links such as cdn and google. If you need file and environment configuration separately, you will need to download the libraries.

필요한 Python library는 pip 를 이용해 설치할 수 있으며, jQuery를 포함한 javascript 등의 library는 cdn, google 등의 링크를 통해 참조하도록 구성하였습니다. 별도로 파일 및 환경 구성이 필요할 경우 해당 라이브러리들에 대한 다운로드가 필요합니다.

Software requirements

- Please install the library below with pip.

```
pip install flask
pip install flask_cors
```

Development environment

- **OS**
  - *macOS Sierra. 10.12.5 version*
  - *Ubuntu 16.04 LTS x64*
- **Language**
  - *Python 2.7x, javascript, jQuery, Ajax, Bootstrap, HTML5, CSS3*
- **Editor**
  - *Sublime Text3, Vim*
- **ETC**
  - *Firefox Quantum, Chrome*

Src list

```
./CAN_traffic_detection_visualization_tool
├── Flask_Client
│   ├── app.py
│   ├── detect
│   │   ├── __init__.py
│   │   ├── __init__.pyc
│   │   ├── refactor.py
│   │   └── refactor.pyc
│   ├── static
│   │   └── js
│   │       ├── circleDraw.js
│   │       └── highcharts.js
│   └── templates
│       ├── base.html
│       ├── index.html
│       └── view.html
└── Flask_Server
    ├── app.py
    ├── can
    │   ├── __init__.py
    │   ├── __init__.pyc
    │   ├── can.py
    │   └── can.pyc
    └── templates

8 directories, 15 files
```

Installing the source

```
git clone "this repository URL"

  or

Download Zip
```

Compile & Running

> This tool is a tool written in Python, no separate compilation is required. If you just installed the python library with pip, listed above, you can run it.

본 도구는 Python 으로 제작된 도구로써, 별도의 컴파일 작업이 필요하지 않습니다. 상단에 기재된, pip 를 이용한 python library 만 설치된다면 바로 실행할 수 있습니다.

> The items shown below are those for which the Debug option works, and the Debug option does not work for the version being deployed.

아래 표시된 항목은 Debug 옵션이 동작하는 화면이며, 배포되는 버전은 Debug 옵션이 동작하지 않습니다.

- Run Flask-client app.py

```
 jsh05042@Macs-MacBook-Pro  ~/Desktop/Data Analysis Challenge/CAN_traffic_detection_visualization_tool
 cd ./Flask_Client
 jsh05042@Macs-MacBook-Pro  ~/Desktop/Data Analysis Challenge/CAN_traffic_detection_visualization_tool/Flask_Client
 ls
app.py    detect    static    templates
 jsh05042@Macs-MacBook-Pro  ~/Desktop/Data Analysis Challenge/CAN_traffic_detection_visualization_tool/Flask_Client
 python ./app.py
 * Running on http://0.0.0.0:5096/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 834-410-484
```

- Run Flask-server app.py

```
 jsh05042@Macs-MacBook-Pro  ~/Desktop/Data Analysis Challenge/CAN_traffic_detection_visualization_tool
 cd ./Flask_Server
 jsh05042@Macs-MacBook-Pro  ~/Desktop/Data Analysis Challenge/CAN_traffic_detection_visualization_tool/Flask_Server
 ls
app.py    can       templates
 jsh05042@Macs-MacBook-Pro  ~/Desktop/Data Analysis Challenge/CAN_traffic_detection_visualization_tool/Flask_Server
 python ./app.py
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 834-410-484
```

Testing environment

- **Recommended specification**
  - **At least 8 GB of RAM**
    (Cause process large amounts of data using Python, memory management is not considered properly, resulting in a significant memory footprint.)

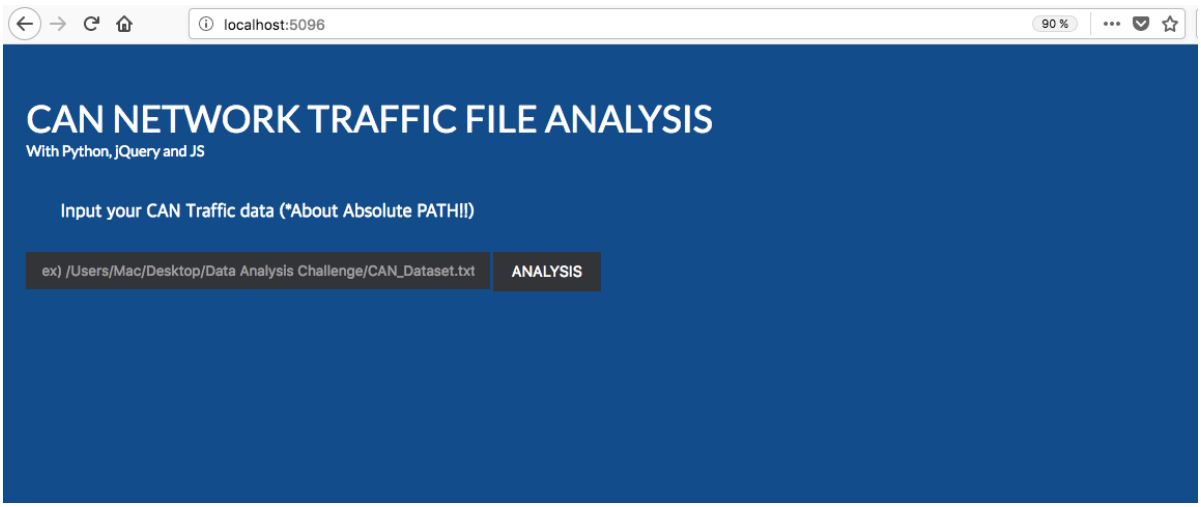
(Python 을 이용해 대용량 데이터를 처리하는데 있어서, 메모리 관리를 적절히 고려하지 않았기 때문에 상당히 많은 메모리 점유율을 가지게 됩니다.)

- **Firefox Quantum**
  (It's optimized for Chrome and Firefox, but you can see it's a little smoother on Firefox Quantum than on Chrome.)

  (Chrome 과 Firefox 에 최적화 되어 있지만, Chrome 보다 Firefox Quantum 버전에서 조금 더 원활하게 보여지는 것을 확인할 수 있습니다.)

- **Please follow the procedure below.**
  - With Flask-server and Flask-client running, follow the procedure below.
    (Flask-server, Flask-client 를 모두 실행한 상태에서, 아래의 절차대로 실행해주세요.)

1. Access to http://localhost:5096. And input your CAN traffic dataset file's absolute path. (like example)

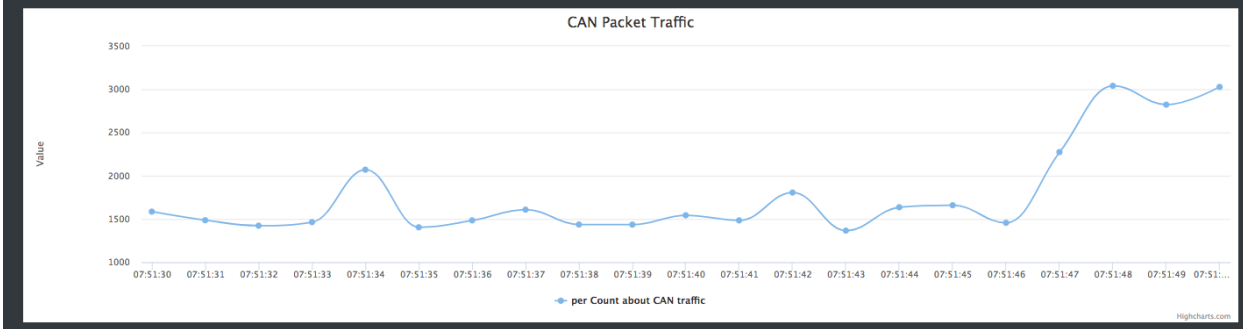


2. Click the "ANALYSIS" button

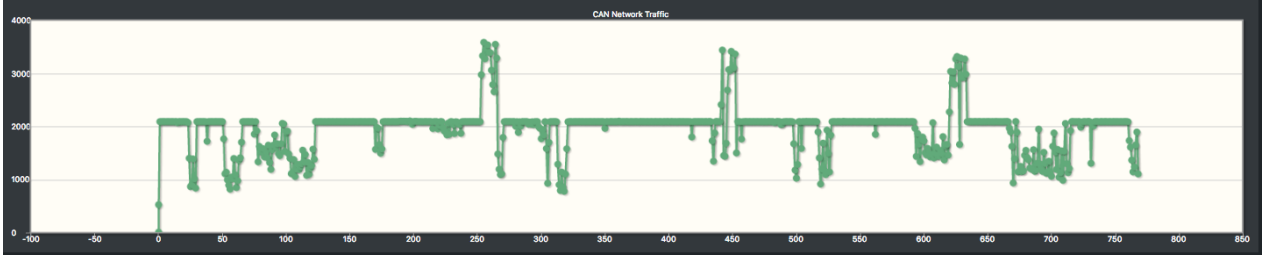
3. Now you can then see the analysis results and visualization information for the dataset as shown below.

localhost:5096/view

[During 1 Minute packet traffic] CAN Packet traffic real time



[About whole packet traffic] CAN Packet traffic real time



localhost:5096/view

[Packet traffic list] CAN Packet traffic real time

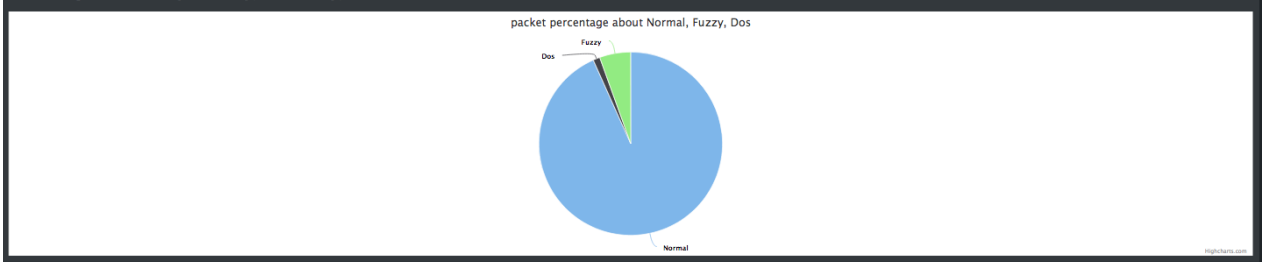
Copy CSV Excel PDF Print Search:

No.	TimeStamp	ID	RTR	DLC	Offset	Detect
1483845	1479110041.999913	0316	000	8	45 21 72 0a 21 1e 00 7f	Normal
1483844	1479110041.999686	0350	000	8	0f 2b 44 8b 94 00 00 7f	Normal
1483843	1479110041.998005	02b0	000	5	1a 00 00 07 95	Normal
1483842	1479110041.997823	0165	000	8	0b f8 7f 00 00 00 05 89	Normal
1483841	1479110041.997581	0440	000	8	ff f0 00 00 ff e0 09 00	Normal
1483840	1479110041.997337	0370	000	8	ff 20 00 80 ff 00 00 ec	Normal
1483839	1479110041.997105	043f	000	8	00 40 60 ff 61 dc 09 00	Normal
1483838	1479110041.996717	0164	000	8	00 08 00 00 00 00 01 09	Normal
1483837	1479110041.996480	04b1	000	8	04 03 00 03 00 00 00 0a	Normal
1483836	1479110041.995027	00a1	000	8	80 87 80 80 2a 00 00 00	Normal

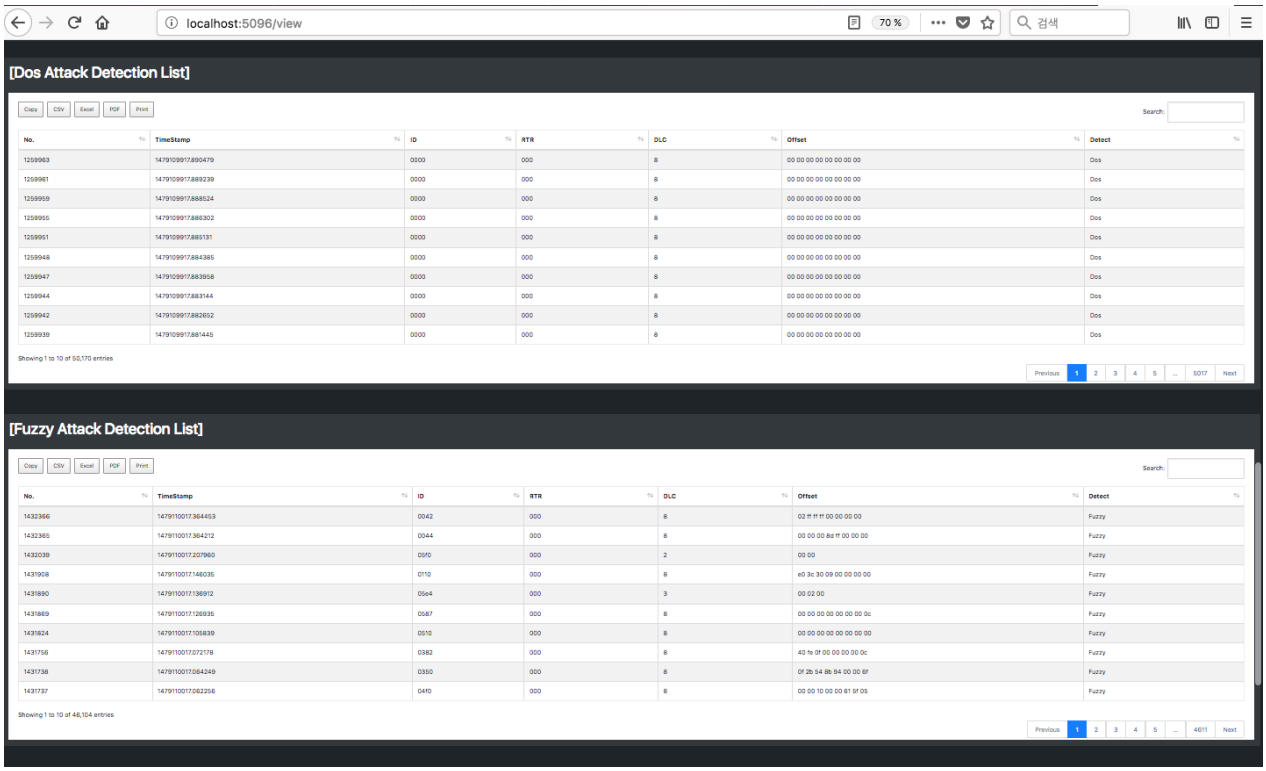
Showing 1 to 10 of 16,718 entries

Previous 1 2 3 4 5 ... 1672 Next

[Percentage of Packets by Attack Type] Attack type count real time



localhost:5096/view#



Finished..

> If you have any questions, or if there are any areas that need to be corrected, please contact us at "jsh05042@gmail.com".