[Gold Standard Traffic Replayer](https://github.com/Gitarth/AuStdTrRe)

# Design Document 3.0

## Project Introduction:

We are developing a network traffic replayer that allows the user to imitate network traffic by using a packet capture file from the client and generating network traffic based on it. The traffic replayer can be used to test normal operations of a network and detect intruders by comparing with normal operations.

## Team Member and Responsibilities:

*Hitarth Patel:* Design, development, and deployment; Quality Assurance; Documentation

*Ethan Hendrix:* Design, development, and deployment; Testing; Documentation

## Sequence Diagram:

## Class Diagram:

## 

## 

## Design Overview:

The traffic replayer will be a command-line application. It will take user input in the form of a command with options (sudo python3 austre.py -na NETWORK\_ADDRESS -ip HOST\_IP -f PCAP\_FILE1 PCAP\_FILEX). We have decided to make this a multi-threaded tool, so that the user can have multiple types of pcaps running on the network concurrently. We realised that to make the most use out of this tool making it multithreaded was not optional. Another thing that seemed important was the network address with the netmask because this is what we used to scan the network and find the available hosts.

The traffic replay is implemented in Python. An austre class will call other classes to generate the required packets. We will have the user provide a packet capture file that will be used to generate similar network traffic flow. A file parser object will open the pcap file, parse the file line-by-line, and create an array of useful data(src,dst,etc) for the generator to use. These layers will then be altered/randomized(e.g. Remapping source and destination IP addresses) to craft new packets. The user can have multiple pcap files passed in to generate different types of traffic, or they can include one pcap file that will include all of the traffic they wish.

The user will start the python application from the command line, afterwards they will enter the necessary command and options to begin generating packets. Packets will the be generated based on the user’s input file.

## Functional Requirements:

* Get pcap files
* Parse pcap files
* Generate packets based on the pcap files
* Generate a pcap log
* Simulate on localhost and GENI network

## Resources:

* [Python3 Documentation](https://docs.python.org/3/)
* [Scapy](http://scapy.readthedocs.io/en/latest/index.html)
* Built-in python libraries: Subprocess, Threads, Ipaddress