Stretch Project

Blue Sentinel personal IDS

Moller.r

# Analysis

## Problem definition

I intend to develop a desktop [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface)-based [Intrusion detection system (IDS)](https://en.wikipedia.org/wiki/Intrusion_detection_system) which provides features to detect malicious content on the host [(HIDS)](https://en.wikipedia.org/wiki/Intrusion_detection_system#Host_intrusion_detection_systems) as well as on the network [(NIDS)](https://en.wikipedia.org/wiki/Intrusion_detection_system#Network_intrusion_detection_systems).

This system will focus more on the detection of malicious content rather than prevention but will provide notifications for users and possible actions to deal with malware most typically to delete/remove the found malware.

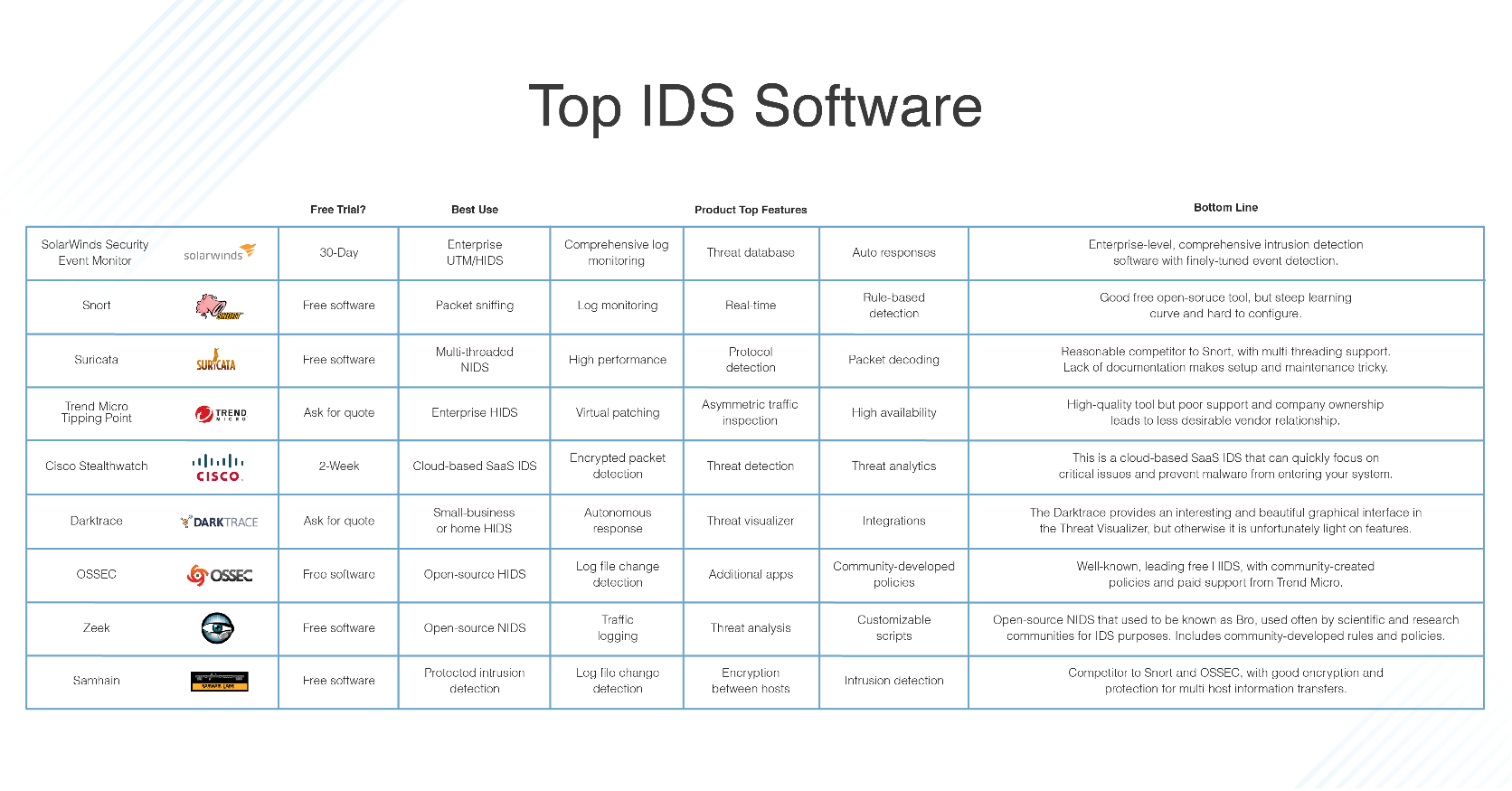
In order to detect malicious traffic, I will be using these three main approaches:

* [Signature-based detection](https://en.wikipedia.org/wiki/Intrusion_detection_system#Signature-based)
* [Policy-based detection](https://en.wikipedia.org/wiki/Intrusion_detection_system#Detection_methods)
* [Anomaly-based detection](https://en.wikipedia.org/wiki/Intrusion_detection_system#Anomaly-based)

*References:*

* [*https://en.wikipedia.org/wiki/Intrusion\_detection\_system*](https://en.wikipedia.org/wiki/Intrusion_detection_system)
* *CompTIA, Network+, N10-007 – Page. 456*

## Research

Current observations

https://www.dnsstuff.com/wp-content/uploads/2019/10/top-ids-software.png

## Requirements Specification / Features I intend to implement

|  |  |  |
| --- | --- | --- |
| FEATURE | DESCRIPTION | PRIORITY |
| Minimalistic and responsive UI | The GUI for the application will have a simple minimalistic design with a consistent design scheme.  The GUI will also be responsive and not hang during operations. | HIGH |
| Network intrusion detection system | Network scan which will inspect packets against a set of rules, alerting user if they are broken. | HIGH |
| Host intrusion detection system | File system scan for malicious files by checking hashes of files on system against a malicious has database, | HIGH |
| Notifications/alert system | A mechanism for alerting user if any rules are broken or malicious files are found. | HIGH |
| Report generation | Formatting results in an exportable medium for user convenience. | MEDIUM |

# Design

## Top down modular design – system diagram

## Technologies

Programming language

For the development of this project I intend to use Python for the backend and frontend.

The reason for doing so is that Python is lightweight has a variety of built-in libraries which will make the development of this project faster and easier.

Furthermore, our application is not reliant on speed or uses a whole lot of resources and so a lower level language with more control over memory management is not necessary.

Operating System

I will specifically develop this program for Windows, I am doing this due to time constraints as I would have to put more resources into testing and implementing features differently if it were to be cross-platform.

OOP approach

In order to create my program, I will be using an object orientated approach. Object

orientated programming involves creating solutions using objects that interact with each

other. In order to create objects a template known as a class is used. Classes have attributes

(variables associated with an object) and methods (functions that the object carries out). The

act of creating an object using a class is known as instantiation.

I chose to use an object orientated approach over a procedural approach due to 3 main

features: Inheritance, Encapsulation and Polymorphism. Inheritance allows derived classes to

inherit the attributes and methods of super classes as well as having their own attributes and

methods; furthermore, instances of the same class can share variables between them.

Encapsulation prevents attributes of a class from being manipulated by external methods and

classes, therefore upholding the integrity of the system; attributes can only be altered using

methods of their respective class. Finally, OOP allows objects to be process differently

depending on their data type or class through the feature of polymorphism.

Agile methodology

During the development phase of my project, I have decided to switch from the waterfall

methodology to an agile approach. Using an agile approach in the development phase will

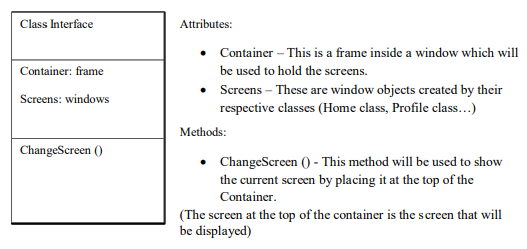
allow me to perform multiple iterations of testing and evaluating on a piece of code/

prototype in the hopes of improving my program to follow and meet the requirements that

were stated in the analysis section which will allow me to make any modifications/improvements before I move on to the next prototype.

## Interface

The interface module will act as a control unit which will link all the individual screens together. The interface module is essential as it will allow each screen and their respective functions to interact with other screens (Share variables and methods, change screens). The interface will consist of 3 screens: Home, FileScan, NetScan. Using OOP the interface module can be modelled as a class.



## Home

The home screen will be the first screen displayed to the user, and will consist of a navigation bar, sidebar, host-based security features and net-based security features.

Proposed home screen mock-up:

Sidebar/status bar

Navbar

Net-based features

Host-based features

## File Scan

The file scan screen will allow the user to scan the host system for malicious content based on signature IDs in the form of hash comparison.

Scan Results

Scan configuration

Sidebar/status bar

Navbar

## Net Scan

The net scan screen will allow the user to scan their network based on a set of network rules. When these rules are broken an alert is displayed to the user.

Sidebar/status bar

Navbar

Scan configuration

Rules

Alerts

## Core functionality breakdown / simple pseudocode

File scan

* Allow user to configure scan path/type
* Use threading to avoid application hanging
* Iterate through each file in path (including sub directories)
  + Get md5 hash of file
  + Compare hash with malicious hash database
  + If match alert user
* Display results to user
* Give user actions to deal with results

Net scan

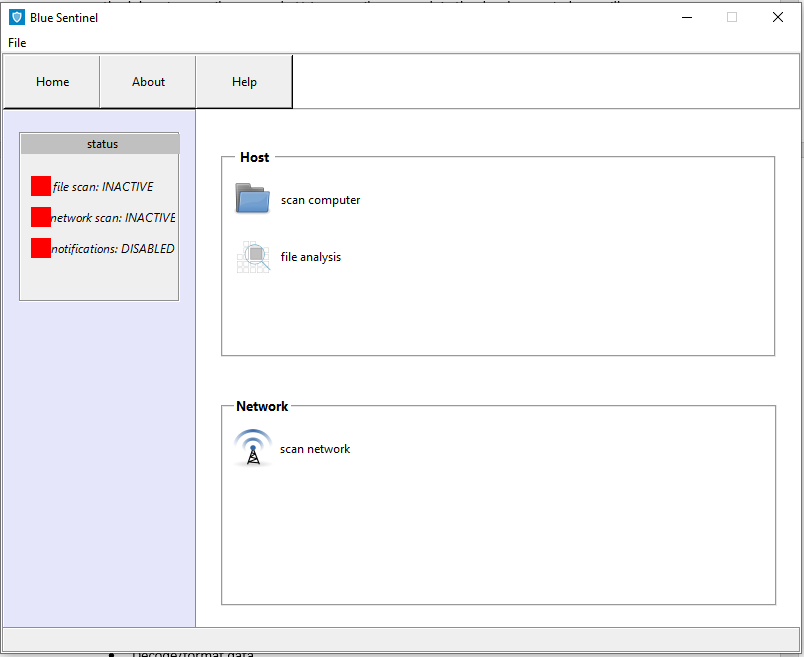
* Capture ethernet frame packets and decode/format correctly
* Decode/format IPv4 packets
* Decode/format TCP segment
* Decode/format UDP segment
* Decode/format data
* Covert IP and MAC addresses to appropriate format
* Check packets against rule set
* If rules broken alert user

# Development

GUI

* I will be using Python’s built in Tkinter library to build the GUI as not only does it give me more control but is cross-platform and provided by default with Python installation.

Home



Use of frames to divide and organise widgets

features represented as buttons

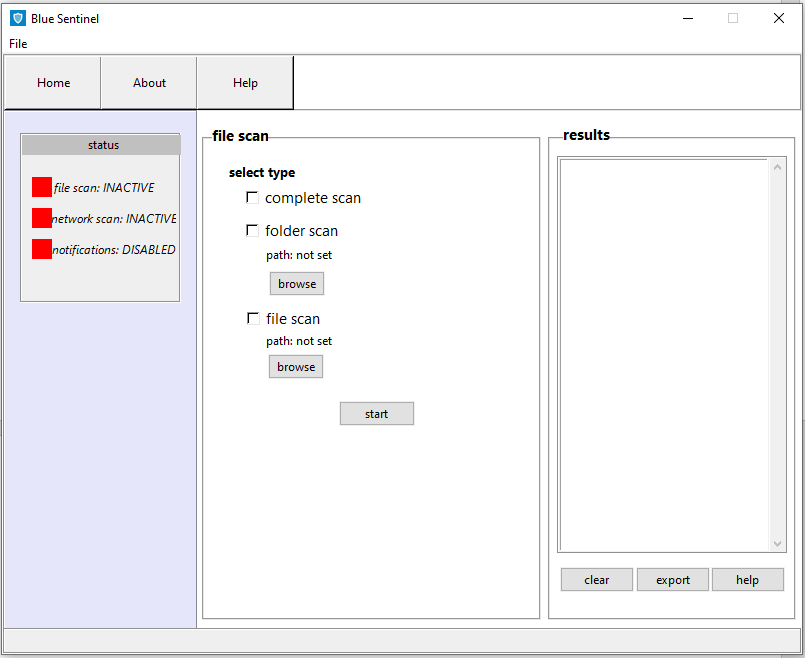
Status bar

Sidebar

Nav bar which allows user to navigate around the APP easily

## File Scan

Scan config using check buttons



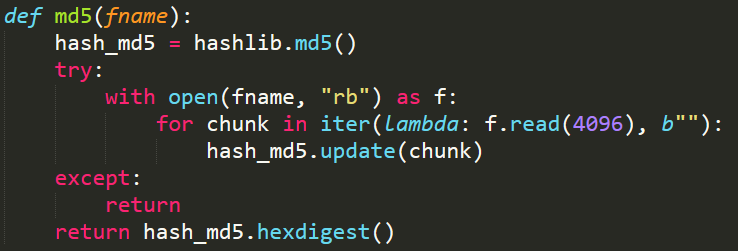
Button which starts the file scan but also does validation e.g.

* Is path empty
* Is an option selected (only 1 check button can be activated at a time)

Listbox which is populated with scan results

File scan functions overview

* Calculate md5 hash



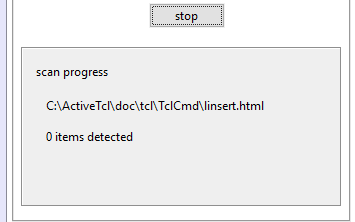
Reads files in binary mode and in chucks of 4096 bytes incase file size is too large.

* Threading

I make use of threading by executing the file scan code in a thread to prevent the app from hanging allowing the user to continue using the app whilst the scanning is ongoing.

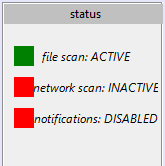


* Scan progress

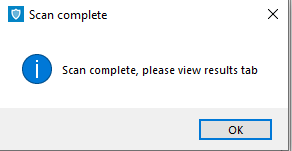


Dynamic display which shows scan progress

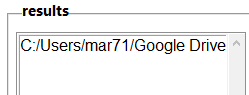
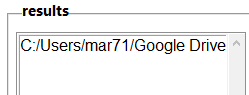
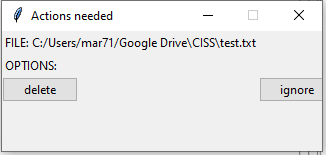
* Updates status bar when scan is active/inactive



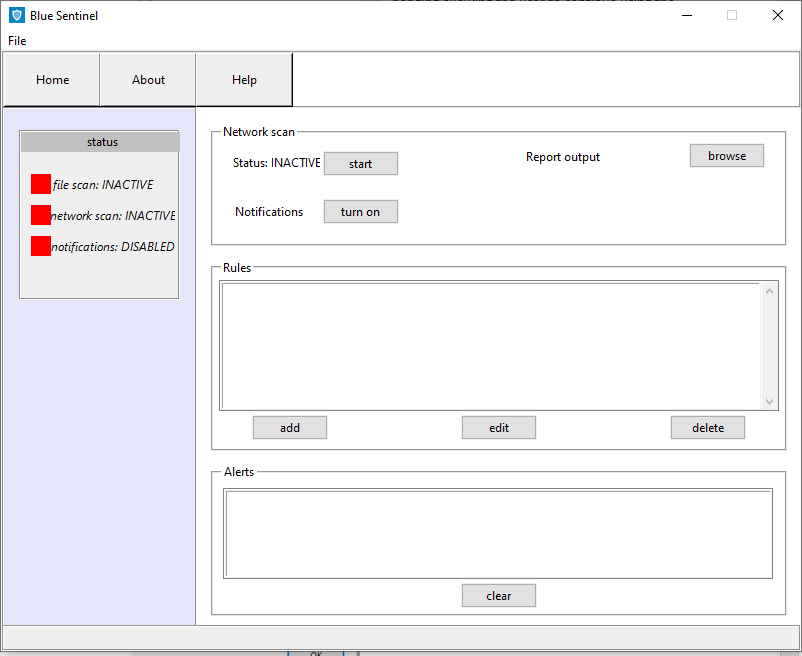
* Alerts



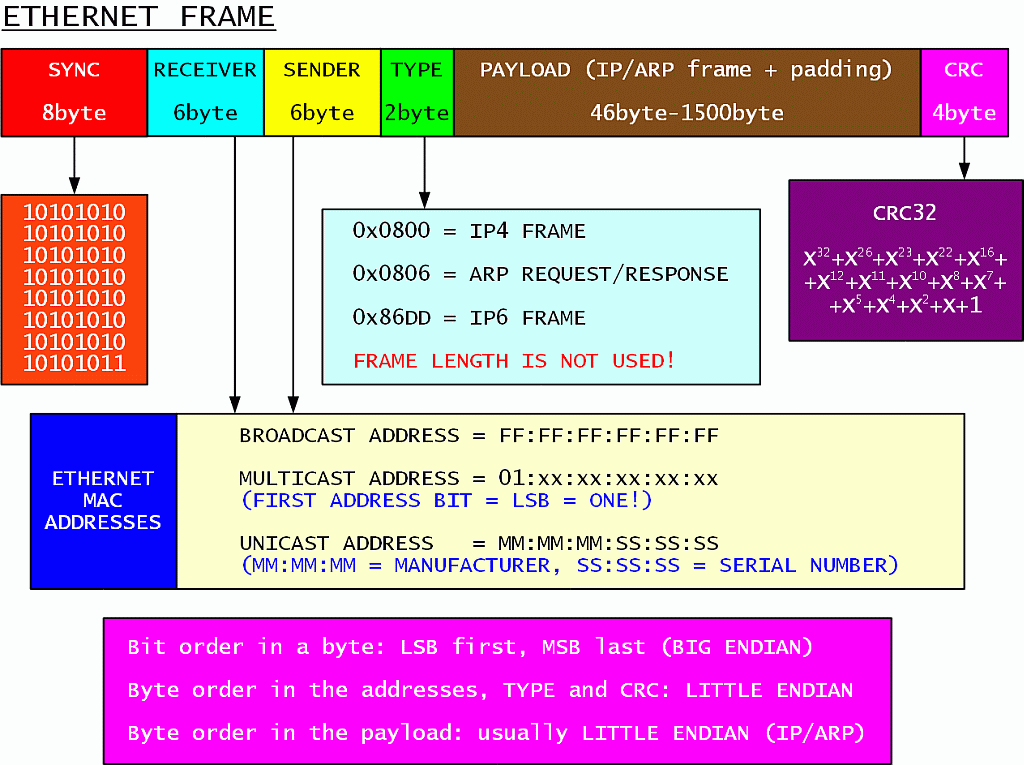
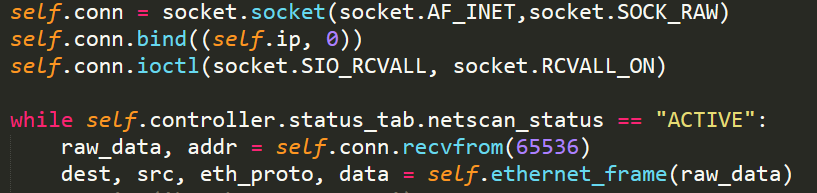
* Results



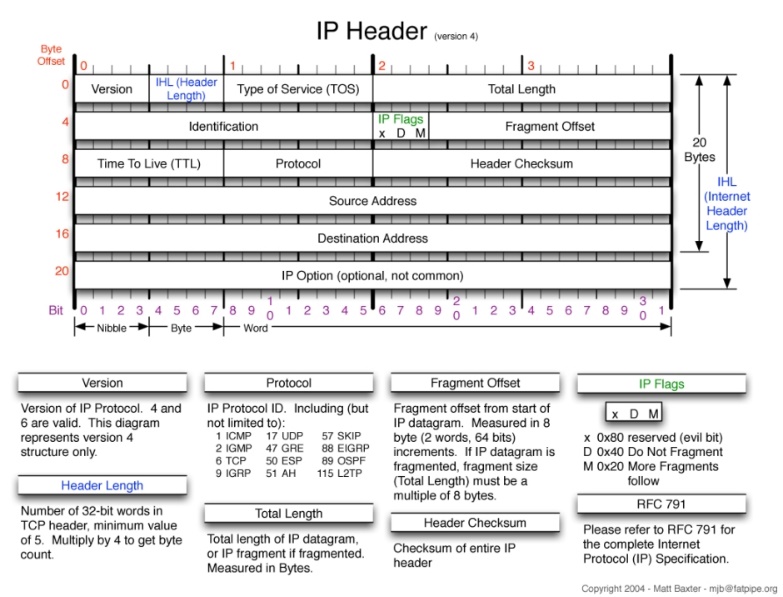
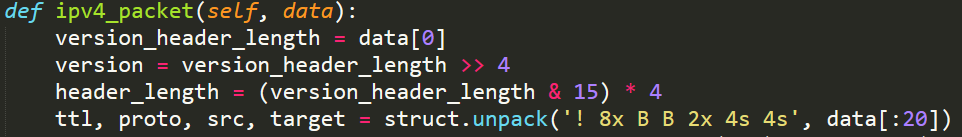
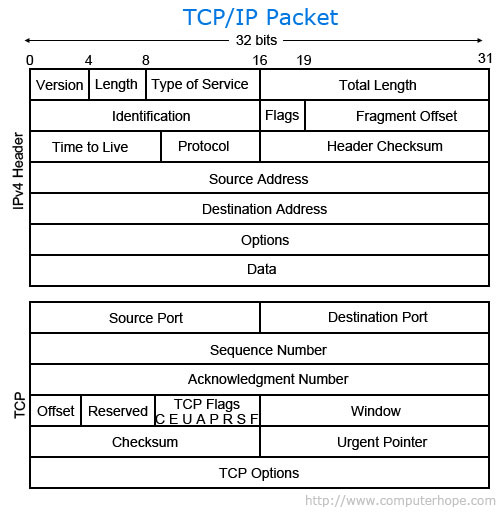
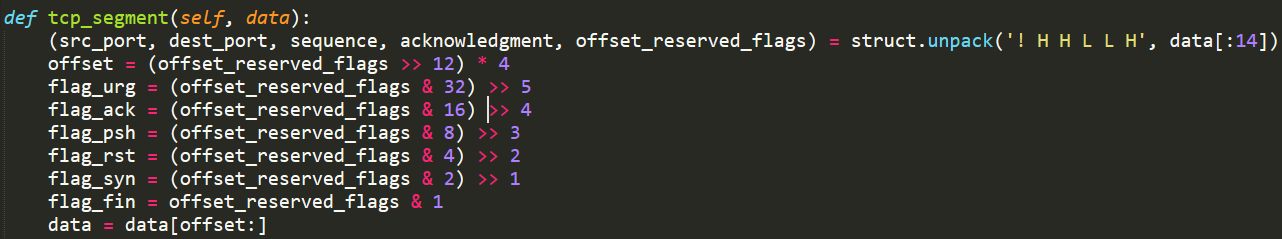
## Net scan



Net scan functions overview

* Decoding and formatting packets and protocols (Ethernet frame, IPv4, TCP, UDP, ICMP)

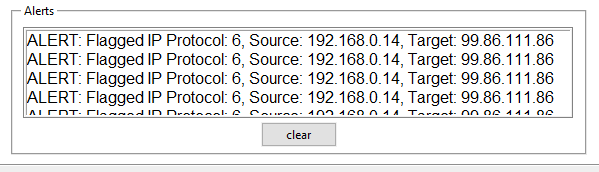




* Rule checking



* Alerts



## Project structure

I have followed the recommended/good practice way of structuring a large scale Python application for deployment as documented in detail on <https://docs.python-guide.org/writing/structure/>.

## Git

The development of this project was done on a private GIT Repo which I am happy to share upon request.

# APP USAGE

# Stretch Project

DIY Intrusion Detection System (HIDS + NIDS)

USAGE:

- Clone repo

- Install project directory as a package (run in direcotry of project folder)

python -m pip install -e .

- Install required packages from requirements.txt (Although all are default with Python installation)

pip install -r requirements.txt

- Change IP in net\_scan.py to your Host IP

- Run main.py

python main.py