DDoS Detection Application

Architecture/Design Document

Table of Contents

[1. Introduction 3](#_Toc1456372)

[2. Design Goals 3](#_Toc1456373)

[3. System Behavior 4](#_Toc1456374)

[4. Logical View 4](#_Toc1456375)

[4.1 High-Level Design (Architecture) 4](#_Toc1456376)

[4.2 Mid-Level Design 5](#_Toc1456377)

[4.3 Detailed Class Design 7](#_Toc1456378)

[5. Use Case View 8](#_Toc1456379)

[5.1 Basic Flow 8](#_Toc1456380)

[5.2 Alternative Flow 8](#_Toc1456381)

[5.2.1 File Not Recognized 8](#_Toc1456382)

[5.2.2 Geolocation URL does not open 9](#_Toc1456383)

[5.3 Post Conditions 9](#_Toc1456384)

[5.4 Special Requirements 9](#_Toc1456385)

Change History

**Version:** 1.0

**Modifier:** Brian Vail

**Date:** 01/22/2019

**Description of Change:** Initial rough draft. Contains design goals, system behavior, logical view, and use case view.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Introduction**

This document describes the architecture and design for a DDoS Detection Application developed as a graduate level project at Florida Atlantic University. The DDoS Detection Application is an executable file that reads unsolicited internet traffic and identifies a type of cyber-attack known as a distributed denial of service attack. The application infers the source IP address of the cyber-attack, the type of DDoS attack and the rate of the attack. Then the application geo-locates the source IP address to determine the city, country, and Internet Service Provider of the source of the attack. The DDoS Detection Application is designed for users with a technical background in computer science and cyber security.

The purpose of this document is to describe the architecture and design of the DDoS Detection Application as applicable to all major stakeholders. For this application, the stakeholders are:

* Users – they expect the architecture to provide system functionality and non-functional quality requirements such as usability, processing speed, readability, etc.
* Developers – they expect architecture that reduces complexity, is fully labeled, and will reduce future development effort
* Advisor – the advisor is responsible for assigning application requirements. He or she expects the application to be divided into functional components with well-defined interfaces.

The architecture and design of a software application is complex and stakeholders have individual interests. For this reason, there is no one diagram or model that can illustrate a system’s architecture and design. Therefore, this application is presented from the following perspectives:

1. Logical View – describes the functionality the application provides to end-users. Includes major components of the application, their attributes and operations.
2. Use Case View – describes a user’s interaction with the application, including multiple scenarios the user may encounter, and the types of user’s who will interact with the application. The application should have all the necessary components to execute the various scenarios of the use case.

# **Design Goals**

The value of a design depends on the needs and requirements of stakeholders. For example, depending on the circumstances, an easily maintainable design may take priority over a design with efficient runtime, or vice versa. Therefore, the design of this application will be judged according to how well it satisfies the stated priorities.

The priorities for the design are as follows:

* The design should minimize complexity.
* The application should be divided into functional components with well-defined interfaces.
* The application should have a reasonable runtime, taking into account that it will analyze hundreds of millions of data points in a single use.

# **System Behavior**

The architecture description presented here details the expected system behavior. This acts as a primer for the architecture description that follows.

“Attack Type = TCP

Source IP = 92.42.38.238

Country = Russia…”

**Welcome**

“Enter file name: ”

“Reading file…”

“Packets analyzed…”

“Searching geolocation…”

**Figure 1 System Behavior**

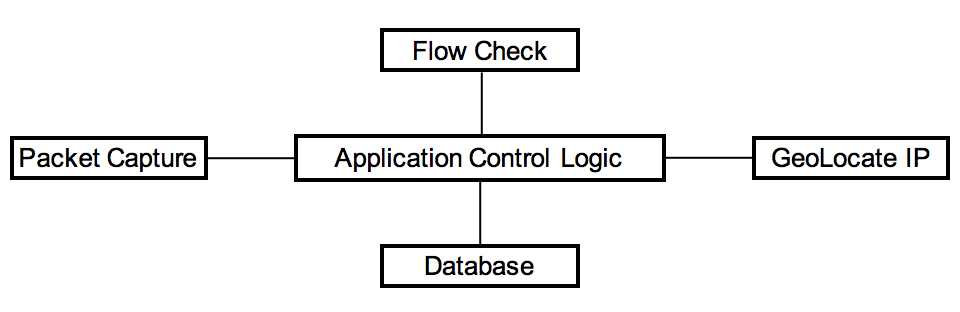
# **Logical View**

The logical view describes the functionality the application provides to end-users. This includes major components of the application, their attributes and operations.

In this section the modules of the application are first expressed in terms of high level architecture and progressively refined into more detailed components and eventually classes with specific attributes and operations.

## High-Level Design (Architecture)

The high-level view or architecture consists of 5 major components:



**Figure 2 System Architecture**

* The **Packet Capture** opens a “pcap” file that contains unsolicited internet traffic. It then reads the unsolicited internet traffic, parses IP information, and labels suspected DDoS attacks.
* The **Database** is a central repository for internet traffic related to a DDoS attack.
* The **Flow Check** scans data from the packet capture component and adds unsolicited internet traffic that qualifies as DDoS attacks into the database.
* Given an IP address from the database, **GeoLocate IP** opens a URL to determine the city, country, ISP, latitude, and longitude of the IP address.
* The **Application Control Logic** is the main driver of the application. It reacts to user input and presents results to the user.

## Mid-Level Design



**Figure 3 Mid-Level Design Components and their Relationships**

Figure 4 shows the dynamic behavior of mid-level components.



**Figure 4 Sequence Diagram**

## Detailed Class Design

The DDoS Detection Application is a small program and only contains a single class. The rest of the components are simple functions that do not require a class structure. They will be listed here with the **Database** class for program clarity.



**Figure 5 Class Diagram**

# **Use Case View**

## Basic Flow

1. System prompts user to enter the name of the file containing unsolicited internet traffic.
2. User enters file name.
3. The system opens and reads file.
4. The system analyzes the unsolicited internet traffic data and labels DDoS attacks.
5. The system searches geolocation data for each labeled DDoS attack.
6. The system displays geolocation information for each labeled DDoS attack.

## Alternative Flow

### File Not Recognized

At step 3 of the Basic Flow, if the file is not recognized or does not open the system informs the user the file was not recognized and the program exits.

### Geolocation URL does not open

At step 5 of the Basic Flow, if the URL used to identify geolocation information does not open the system informs the user there is no connection to the geolocation database.

## Post Conditions

Upon successful completion, the user will view all unsolicited internet traffic from their file that has been confirmed as a DDoS attack. This includes type of attack, source IP, number of packets in attack, length of attack, rate of attack, country of origin, and city of origin.

## Special Requirements

The system can handle only 1 file at a time. The size of an acceptable file is dependent on the amount of RAM of the user’s machine. This is relevant as files containing internet traffic may contain hundreds of millions of data points, leading to the risk of a system crash.