Malmenator

Network Intrusion Detection Architecture

Final Year Project

Final Report

May 3, 2020

Piyush Jha (3035342691)
David B. Han (3035344211)
Supervisor: Dr. Dirk Schnieders
University of Hong Kong

github.com/piy0999/Malmenator

Abstract

Strong security measures in the digital world are becoming increasingly important as indicated by the meteoric rise of spending on cybersecurity as well as the increased losses from successful cyberattacks. One important research area in cybersecurity is network anomaly detection, which is implemented in network-based intrusion detection system (NIDS) to identify and stop malicious network activity before it causes damage. Malmenator takes takes a two pronged approach to tackle this issue through both an engineering and a research component, each covered in their own report.

This report will focus on the engineering component of the project. As a research based project this part of the report gives in depth guidance on creation of a minimum viable product which can act as the basis for further feasible industrial development in the domain of network anomaly detection. We propose working with a Raspberry Pi and cloud based prototype that doubles up as a router and a NIDS with capability to connect with the cloud to flag anomalies on the Wi-Fi network created using the Pi. The contributions of the engineering side of Malmenator proves the technical feasbility of this research and provides a preliminary pathway to implement such a system on an industrial scale.

Acknowledgements

We would like to thank our supervisor, Dr. Dirk Schnieders, for his support and advice regarding the usage of machine learning in our project. His input allowed us to appropriately define our project scope and focus our research in a direction with the most impact. Furthermore, we would like to extend a special token of gratitude to professors from the Centre of Applied English Studies at the University of Hong Kong in helping us improve our writing skills to express the work done during the project. Lastly, we would like to thank the Department of Computer Science at The University of Hong Kong for reimbursing the costs incurred for buying various equipment during the project and making the project feasible.

Contents

List of Figures

List of Tables

Abbreviations

AWS Amazon Web Services

ELK Elasticsearch Logstash Kibana

ICMP Internet Control Message Protocol
IDS intrusion detection system
IP Internet Protocol

LAN local area network

NIDPS network-based intrusion detection and prevention system
NIDS network-based intrusion detection system
NIPS network-based intrusion prevention system

TCP Transmission Control Protocol

UDP User Datagram Protocol

Wi-Fi Wireless Fidelity
WLAN wireless local area network

Chapter 1

Introduction

This chapter offers an insight into the motivation behind the projects and the key problems solved along with the contributions made to the cybersecurity research globally. It will also briefly discuss the organisation of the whole report.

1.1 Motivation

By making cybersecurity more readily accessible, we aim to increase trust in a society that is undergoing an enormous digital transformation. Over the past decade, spending on cybersecurity has been consistently increasing year over year, yet the losses due to cyberattacks only continue to grow [1]. With the scarcity of public investment in network security research and the shortage of robust and interpretable network anomaly detection methods [2], this research project aims at empowering the network security community by analyzing the efficacy of anomaly detection techniques across a variety of network environments. Additionally, the Malmenator project aims to create an effective, affordable, and portable NIDS that can be adapted to any home or small office environment through its engineering component.

This report will primarily focus on the engineering component offering detailed understanding regarding developing such a viable and portable NIDS with the use of state of the art technologies.

1.2 Problem Formulation

From an industry perspective, network security solutions for individuals are inaccessible and expensive. Due to the confidential nature of data breaches and network traffic, individual parties such as governments and corporations have very little cooperation compared to other areas of computer science research [3]. This is especially true for network cybersecurity as network traffic data often contains highly sensitive information. Currently, most network security solutions are tailored towards large enterprises that can afford the implementation costs, while most individuals and small groups leave network security to their Internet Service Providers. Furthemore, running most network security solutions cause an added overhead on each machine in the network. They are also hard to properly configure by layman individual users besides their expensive subscriptions along with little to no open source alternatives.

Problems from a research perspective are described in detail in [4].

1.3 Contributions

There are several contributions made by Malmenator to the cybersecurity research and industry. First, Malmenator contributes to the research community in its research of robust anomaly detection methodologies built on Netflow data. Next, Malmenator sheds a light on the difficulties of testing NIDSs with both simulated and real datasets and provide a more comprehensive solution to this issue. Furthermore, Malmenator introduces novel combinations of malicious network detection methodologies that have real-world and commercial implications. Lastly, Malmenator makes available an innovative and an accessible design for a NIDS implemented on top of a Raspberry Pi that functions as a secure internet access point.

This report focuses on the latter two contributions detailing the novel NIDS architecture and its minimum viable product, while the former two are covered in [4].

1.4 Report Organization

This report begins by providing a brief introduction to the cybersecurity industry as a whole. It then gives an introduction to NIDSs and the capabilities of intrusion detection softwares. Following which the report describes the ways of capturing all the network data flows. It then proceeds to the project methodology and describes the steps by which Malmenator will implement its hardware,

analyze its dataset, detect anomalies, and visualize its system. Lastly, this report concludes by dicussing the project's key achievements along with the scope of future improvements.

Please note that this report is mean to be read in conjunction with [4] in order to understand the complete scope of the research as well as the implementation component of the project. Thus, the project has adopted a twin report methodology.

Chapter 2

A Primer on Cybersecurity

This chapter offers an overview of the technology and economic conditions of various aspects of the the cybersecurity industry. Reading this chapter useful for understanding the basics of cybersecurity for the unfamiliar and may be skipped if one wishes to delve directly into details of the project. This chapter first describes the nature of cyberattacks before giving an overview on various high level aspects of cybersecurity including its economic impacts, ethics, and challenges.

2.1 What are Cyberattacks?

A cyberattack is an intentional and malicious attempt by an individual or organization to disrupt or penetrate the information system of another individual or organization. There are many forms of cyberattacks, some of common forms are described below. Note that the list below is non-exhaustive; cyberattacks are constantly evolving and adapting to defensive measures as they are deployed.

Malware: Malware is a combination of the words *mal*icious soft*ware*. Malware breaches a network through a vulnerability or exploit, and usually occurs when a user clicks a dangerous link or email attachment that installs the malware. Once activated, malware can perform any of a number of actions depending on the malware type. In the case of ransomware, the malware can block access to key files until a fee is paid. In the case of spyware, the malware can secretly obtain and transmit information from the system. In the case of a backdoor, the malware can allow hackers to gain remote access to the computer. There are a large number of malware variations, and new forms are discovered every day.

Man-in-the-middle attack: Man-in-the-middle attacks occur when attackers insert themselves into a a communication line. Thus, instead of one party communicating directly with the other party, all communication traffic is unknowingly sent through the "man-in-the-middle", which enables the attacker to monitor and steal data.

Denial-of-service attack: Denial-of-service attacks overwhelms system networks with a huge amount of traffic to exhaust their resources and bandwidth to cause the system to be unable to fulfill legitimate attacks. In the event that this attack is simultaneously launched from a number of compromised devices, the attack is known as a distributed-denial-of-service (DDoS) attack.

Zero-day exploit: Zero-day exploits occur when a network vulnerability is discovered and announced, but before an update has been released to patch the problem. Attackers target the disclosed vulnerability during the window of time before that patch is released.

Each form of cyberattack takes advantage of different system vulnerabilities and have a variety of different risks. Furthermore, each form can be further subdivided into various types and families as is evident in the malware section. In this project, we focus on forms of attack that can be detected using network traffic patterns, which includes denial-of-service attacks, man-in-the-middle attacks, and certain types of malware such as backdoors.

2.2 Economic Impacts of Malware

Cyberattacks from malware are an increasingly large problem, and companies and governments are spending more year over year in order to properly protect themselves. From 2012 to 2018, average annual cybersecurity expenditures per employee doubled from $584 USD to $1,178 [5]. In 2019, global spending on cybersecurity initiative is expected to exceed $100 billion USD [6]. These numbers are only expected to continue to rise as the financial incentives to engage in malicious attacks only continue to rise.

In 2018, a conservative estimate of financial losses caused by cyberattacks is at least $45 billion USD across millions of reported attacks such as DDoS and ransomware attacks [3]. Including the financial losses due to data breaches and the loss of more than 2 billion consumer data records, this number rises to a staggering $654 billion USD for US corporations in 2018 alone [2]. Cyberattacks to U.S. based financial services organizations in Q1 of 2019 alone cost more than $6.2 billion USD, a sharp rise from just $8 million USD in Q1 of 2018.

It must be noted that these costs are general estimates as the exact loss of value can be difficult to calculate. There is no centralized data set on the costs of cyberattacks and data breaches. Thus, many statistics in the cybersecurity industry come from surveys, which can suffer from non-representative and inaccurate reporting. Often, firms are reluctant to report negative information which may cause these statistics to bias downwards due to under-reporting. The White House published a report in 2018 containing figure 2.1 depicting the various economic impacts of cyberattacks and their respective difficulties in quantifying cost [1]. It can be seen that certain losses such as court fees and forensic costs are easy to justify, but damage to reputation and loss of IP are difficult to quantify. Overall, cybersecurity breaches impact organizations across the world in various ways and magnitudes.

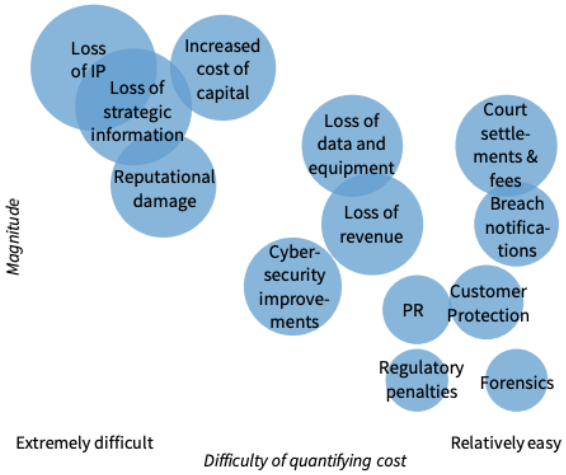

Figure 2.1: Estimating financial losses to cyberattacks

2.3 Categories of Cybersecurity Solutions

Cybersecurity efforts are growing rapidly in order to respond to the quickly shifting landscape of malware attacks. Defensive solutions can take on a variety of forms and roles including anti-virus software, identity and access management tools, intrusion detection system (IDS), and others. The uses of most of these follow directly from their naming convention. Anti-virus software are the bread and butter of the cybersecurity industry and include software such as Windows Defender, Norton Antivirus, and Avast Antivirus. These tools can scan system files and downloads to check for virus

signatures that are then matched to a trusted database. Identity management tools include solutions like 2-factor authentication where users must confirm their identity through a verification code sent to an email or mobile device. These solutions enable organizations to more stringently verify a user's online identity before granting access to sensitive information. IDSs are used to monitor network traffic for suspicious activity and issue alerts when such activity is discovered. Appropriate actions can then be taken such as blocking traffic from suspicious IP addresses or discarding undesirable packets. IDSs are at the heart of the Malmenator project.

2.4 Ethics of Malware Research

In order to defend against malware attacks, researchers first need to understand how malicious code works. Often, this requires preemptively creating malware in order to find vulnerabilities and appropriate solutions. Thus, not all malware is created with malicious intentions. People in the cybersecurity industry can be categorized into one of four categories: white hat, black hat, grey hat, and red hat. The definitions of these types of hackers are listed below:

White Hat: White hat hackers are people who create malware and attempt to break into computer systems for a good cause. These people could work for a cybersecurity firm, or could be a professional penetration testing consultant.

Black Hat: Black hat hackers create malware for malicious causes in order to extort individuals and corporations for personal incentives such as money or power. These people are the ones responsible for much of the malware that cause tremendous economic losses.

Grey Hat: Grey hat hackers do a mix of white hat and black hat activities and dabble in using their knowledge and skills for both good and bad depending on the circumstances.

Red Hat: Red hat hackers are similar to black hat hackers, except they are employed by a government to initiate attacks on foreign powers.

The key takeaway from this section is that ethics in malware research is not always straightforward. However, publicly published academic research is generally a white hat effort, this paper included.

2.5 Challenges of Malware Research

Performing substantive literature review to understand the cutting edge technology in malware research is a difficult task because there is little incentive to publicly publish anti-malware techniques. Any research that is published is also available to attackers who can choose to exploit other vulnerabilities in the system. Thus, the papers published by the top cybersecurity firms and government organizations are usually either outdated or extremely abstract without precise implementation and performance details. A cybersecurity report published by the Royal Society highlighted cybersecurity's distinct characteristics including having multidisciplinary, global, and cross-sectoral interest, which cause research to take place across academic, commercial, and government sectors that further adds to these difficulties [7]. Information sharing across these fields is not transparent, and many corporations are hesitant to share their vulnerabilities in academic or government research as that may impact their reputation and harm their business. Thus, much of the recently published academic research in malware detection have challenges in practical implementation that must be taken into account.

Chapter 3

Technical background

3.1 Overview

This section provides an introduction to the concept of NIDSs. Given that cybersecurity is an extremely niche field in Computer Science, the information in the background is critical for understanding the relevant tools and technologies that the Malmenator project is based on. This section begins with a high level overview of what NIDSs are before delving into the details of their implementation and categorization.

3.2 What is an NIDSs

NIDSs monitor network traffic in order to detect when an unauthorized intrusion is being carried out by hostile entities by providing some or all of the following:

- Monitoring the condition of routers, firewalls, and servers
- Providing system admins a way to tune, organize and understand relevant operating system audit trails and other logs that are often otherwise difficult to track or parse
- Including an extensive attack signature database against which information from the system can be matched
- Recognizing and reporting when the IDS detects that data files have been altered
- Generating an alarm and notifying that security has been breached.

NIDSs offer organizations a number of benefits, starting with the ability to identify security incidents. NIDSs can be used to help analyze the quantity and types of attacks, and organizations can use this information to change their security systems or implement more effective controls. NIDSs can also help companies identify bugs or problems with their network device configurations. These metrics can then be used to assess future risks.

NIDSs can also help the enterprise attain regulatory compliance. An IDS gives companies greater visibility across their networks, making it easier to meet security regulations. Additionally, businesses can use their NIDSs logs as part of the documentation to show they are meeting certain compliance requirements.

Although NIDSs monitor networks for potentially malicious activity, they are also prone to false alarms (false positives). Consequently, organizations need to fine-tune their NIDS products when they first install them. That means properly configuring their intrusion detection systems to recognize what normal traffic on their network looks like compared to potentially malicious activity.

3.2.1 NIDSs vs NIPSs

NIDSs are used to monitor and analyze network traffic in order to identify suspicious activity and alert system administrators in the event of an attack. A NIPS is similar to an NIDS, but differs in that an NIPS can also be configured to automatically block potential threats without the intervention of a system administrator. Historically, NIDSs were tailored to process network data more thoroughly and rapidly than NIPSs, but with the advent of increased processing power, the line between the two has become blurred. Today, most NIDSs provide configurations to allow for their capabilities to extend into the territory of NIPSs. Hence, the term network-based intrusion detection and prevention system (NIDPS) was coined. network-based intrusion detection and prevention systems (NIDPSs) are systems that combine the capabilities of NIDSs and NIPSs. Although the prevention aspect of NIDPSs are critical is dealing with certain aspects of cybersecurity, Malmenator focuses on NIDSs implementation to allow for a more focused project. Further research should be done to determine and implement the optimal reaction to different forms of cyberattacks.

3.2.2 NIDS Implementation Strategies

NIDSs are deployed at strategic points within a system network where it can best capture traffic to and from all devices on the network. This usually involves being deployed directly within or in parallel to a router, switch, or access point in a network. Figure 3.1 depicts two implementation

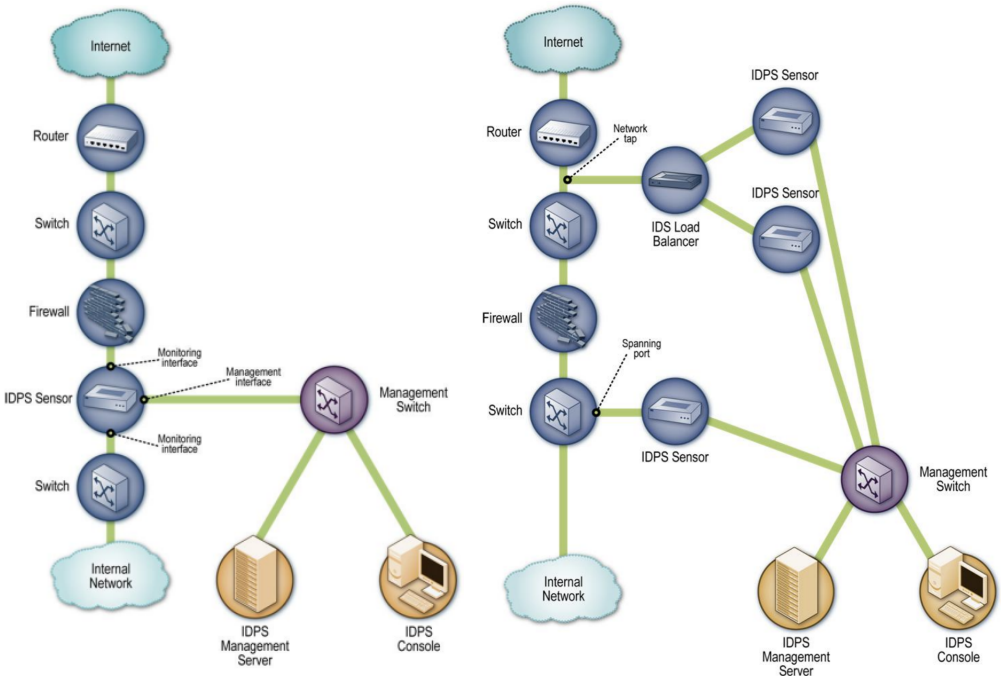

Figure 3.1: Inline and passive implementations of an NIDS

Inline (left) and passive (right) implementation of a NIDS as recommended by the National Institute of Standards and Technology, U.S. Department of Commerce [8]. In the inline implementation, all network traffic must pass through the NIDS which may throttle network speeds. In the passive implementation, a copy of all network traffic is send to the NIDS for processing, enabling network speeds to remain high.

methods of an NIDS. Inline implementation has the benefit of being able to directly respond to attacks by blocking network traffic. On the other hand, passive implementation must rely on a separate tool such as a firewall to secure traffic. Malmenator utilizes inline implementations of NIDSs since Malmenator focuses not only on detecting, but also on preventing network attacks. Note that in Figure 3.1, the NIDS is depicted as a separate hardware component. In some inline implementations, such as with Malmenator, the NIDS can be set up directly within the router.

3.2.3 Overview of NIDS Techniques

This section describes the high-level methodologies that are employed in NIDSs to detect malicious traffic, namely signature-based detection and anomaly-based detection. These methodologies can be combined in a hybrid approach to balance each other's weaknesses [9].

3.2.3.1 Signature-based Detection

Also known as misuse-based techniques, signature-based techniques refer to the detection of attacks by looking for specific patterns within traffic data. The detected patterns are known as signatures, which are then matched against a trusted database to check if they have previously appeared in any malicious attacks. Signature-based techniques are effective for detecting previously known attack types with high accuracy and without raising a large number of false alarms, but their efficacy is only as good as the signature database [10]. Signature-based techniques rarely detect novel attacks (zero-day attacks) whose signature is not already inside the database. For example, one of the most popular NIDS, Snort, conservatively captures 8.2% of zero-day attacks [11]. Overall, this technique is computationally fast, but not highly adaptable [12].

3.2.3.2 Anomaly-based Detection

To solve the shortcomings of signature-based techniques in novel attacks, anomaly-based techniques model normal network behavior and identify deviations from the norm, enabling them to detect novel attacks whose signatures may be previously unknown. However, anomaly-based detection techniques may suffer from false positives - normal activity not yet seen before may be incorrectly classified as malicious. Furthermore, anomaly-based detection is a time consuming process in both training and execution, and many implementations suffer from excessive delay during the detection process that degrades their performance [13]. Anomaly-based detection is an area of ongoing and active research, and is one of the areas of focus for the Malmenator project.

3.3 Packet-based and Flow-based Data Formats

Network traffic data is formatted in one of two ways: packet-based or flow-based. Packet-based data is captured in PCAP format and contains both metadata and payload information for each network packet. Metadata information for each packet depends on the transport protocol, and their differences are highlighted in Figure 3.2. There are a number of different protocols, but the most important ones are IP, TCP, UDP, and ICMP as they constitute the majority of internet traffic and are the core of packet-based datasets [14]. It is important to critically evaluate the impact of various transfer protocol data types on models built using packet-based data.

Flow-based data is more compact compared to packet-based data, mainly containing metadata about network connections. Flow-based data aggregates packets within similar properties in a time

time frame into one flow and discard payload information [14]. As packet-based data is more detailed and thorough, packet-based data can be converted into flow-based data but not vice versa.

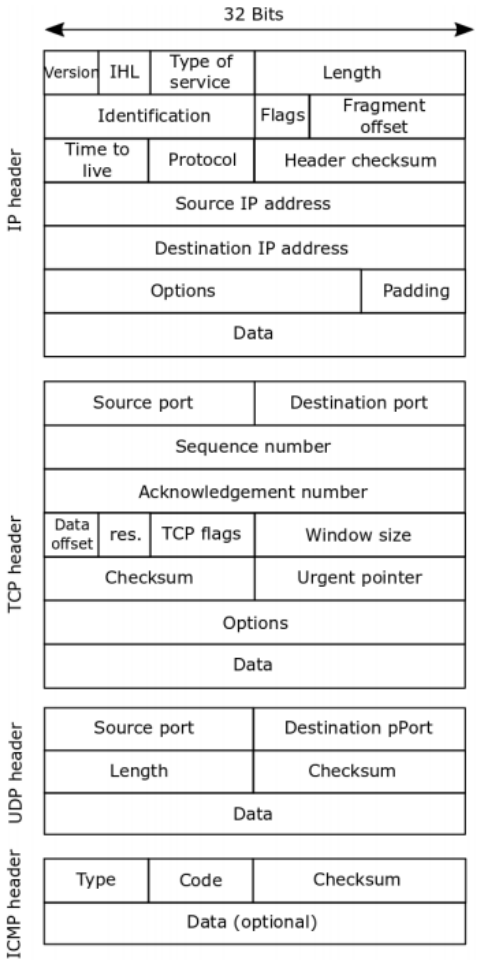

Figure 3.2: Overview of packet-based headers by protocol

Packet header formats for the IP, TCP, UDP, and ICMP transport protocols [14]. Each segment of the header is 32 bits in lengths. Note that there may be multiple data segments in a network packet. Packet data information can be used in payload analysis for anomaly detection, which is another branch of network anomaly detection.

3.3.1 Netflow

There are a number of variations of flow based network data including sflow and jflow, but the most widely used format of network flow data is called NetFlow. NetFlow was created by Cisco

and defines a flow as a set of packets that have a common combination of key-fields in the packet. This includes information such as source and destination IP addresses and port numbers, protocol type, signature byte and logical interface. A packet is sorted into a flow record if it matches the combination of key-fields listed above [15].

There are a number of different version of NetFlow, the most widely used being V4, V6, V9, and V10. At the time of writing, V10 is still relatively new the market and the most widely implemented is V9, which we will be using throughout the remainder of this project. The main differentiator between versions is the underlying methodology in which the packets are gathered and sorted - not in the flows themselves. Thus, research done on V9 should be widely applicable to NetFlow data of both newer and older versions as the high level concept behind NetFlow data remains unchanged despite underlying engineering changes [15].

Chapter 4

Previous Works

4.1 Overview

This section details the information regarding available open source tools previously created that will form a vital part in the development of Malmenator. This carries a significance from an engineering aspect of the project as these tools will be used directly in the methodology chapter 5 of the report. This section does not focus on previous research in detail as this report is centered towards the engineering application of the project. For understanding of previous research please refer to the twin report [4].

4.2 Snort

Snort is one of the most-known open source NIDPS available, and is actively maintained by Cisco Systems. It features cross platform compatibility, and has open source rule libraries to check for abnormal traffic behaviour.

4.2.1 Snort Architecture

Snort operates under either sniffer, packet logger, or intrusion detection mode. Malmenator uses intrusion detection mode since the former are purely for logging and reporting. Intrusion detection mode is based on a set of signature-based rules for packet filtering. Rules can be customized and distributed publicly - there are currently more than 20,000 rules built by the Snort community [16].

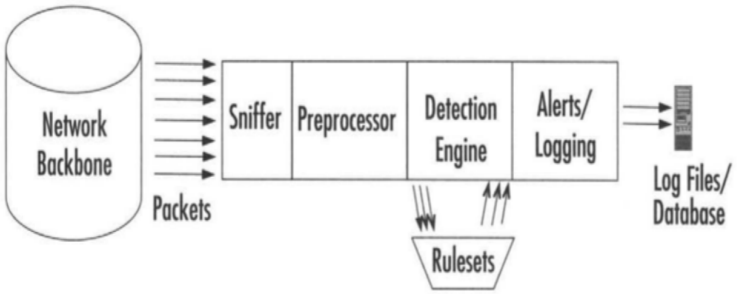

Figure 4.1: Snort processing flow

The basic process by which snort processes each incoming network traffic as described in [17]. The process flows from left to right, with the Snort sniffer taking in and processing network packets from the network backbone (usually bidirectional internet traffic) before processing them via a detection engine and customized rules.

A general flow of Snort's packet processing steps are shown in Figure 4.1. Each packet must be individually acquired and processed by Snort, which can throttle traffic if the detection engine and rules are too complex or poorly implemented. This results in either slowed network speeds or dropped network packets. It is an ongoing research topic to develop quick methods of detecting malicious network traffic without any side effects and is something to bear in mind when researching anomaly-based detection methods that rely on machine learning [9].

4.3 Network Flow Data Collection

This section describes some of the open source tools utilised in this project to facilitate effective collection and conversion of network flow data.

4.3.1 CICFlowmeter

This is an open source tool (*link*) developed by the Canadian Institute for Cybersecurity. The purpose of this tool is to calculate the bidirectional flows in an industry standardised method such that the network flows captured can be analysed more effectively and meaningful insights can be generated from such flows. This tool enriches the pcap packet based data captured using traditional methods like tcpdump or snort and converts the data into bidirectional flow generating 83 statistical features such as Duration, Number of packets, Number of bytes, Length of packets, etc and the output is generated in the form of s comma separated file [18]. The details regarding the precise

application of this tool is described in chapter 5.

4.3.2 Tcpdump

Tcpdump (*link*) is a command-line tool which helps in analyzing network packets flowing in and out of any given interface from a device. It is extremely effective in recording packet data and efficiently helps the constant logging and storage of all packets flowing in and out of the system in PCAP formatted file for the further analysis of the network flow. This forms a baseline for network traffic capture for the Malmenator project and also acts as a data input pipeline along with the CIC Flowmeter, the details for which are discussed in chapter 5.

4.3.3 ELK stack

ELK refers to Elasticsearch, Logstash and Kibana which are a set of open source solution working in conjuction with each other to facilitate an effective storage, analysis and visualisation of logs and other data formats. The 3 components are described in detail in this section.

4.3.3.1 Elasticsearch

Elasticsearch is an index based database which forms as the basis for the big data storage. It is a powerful search and analytics engine that provides quick indexing and storage for the constant bidirectional flow of network packets [19].

4.3.3.2 Logstash

Logstash is a server side data processing pipeline for ingesting data into the elasticsearch database from multiple sources or machines. Logstash as a daemon program can instantly convert the logs and alerts to the JSON format required to send and store the data into elasticsearch [19].

4.3.3.3 Kibana

Kibana is an effective visualisation tool that runs on top of the elasticsearch database. It can easily connect with database and display all the information stored across multiple indices in the elasticsearch backend. Moreover, it can be used to create dashboard with customised visualization like charts, meters, data tables, timelines, etc to effectively summarise the big amount of data stored in each index on elasticsearch. Besides, it also provides capabilities for raising alerts to the user, in our case, when an anomalous or malicious data flow is detected [19].

Chapter 5

Methodology

5.1 Overview

This chapter describes the in depth development of various components that entail the implementation of Malmenator. There are three broad parts of the implementation. First is the implementation of the NIDS on a hardware based tool. Second is development of the machine learning model for network anomaly detection which is described in the the twin report [4]. Third is the creation of the web interface for real-time network traffic evaluation and anomaly alerting system.

5.2 NIDS Architecture

This section describes the process of building a NIDS hardware device on Raspberry Pi to monitor network traffic.

5.2.1 Raspberry Pi

Raspberry Pis are low cost computers that can be customized for a variety of purposes such as web servers, personal computers, or in our case a hybrid WiFi Router and NIDS device. The recent release of the Raspberry Pi 4 Model B expanded its processing power from 1GB to 4GB of RAM, making it an ideal tool to use to build Malmenator. Raspberry Pis can run a variety of open source operating systems, but Raspbian OS's latest September 2019 distribution made it the ideal choice to use in Malmenator for stability and compatibility reasons [20].

5.2.2 Hardware Configuration

The Raspberry Pi was used as illustrated in Figure 5.1, where it acted as a NIDS integrated into a router, similar to the inline NIDS implementation described in section 3.2.2. The NIDS device was connected to the internet through a wired Ethernet connection into the HKU local area network (LAN). From there, the device was configured to act as a wireless router by broadcasting a wireless local area network (WLAN) and routing all connections through the Ethernet port by implementing the iptables library for Linux and intensively modifying a number of network configuration files. To broadcast a wireless network, an external wireless USB adapter was originally utilized before discovering that Raspberry Pi's inbuilt wireless card could similarly be utilized to accomplish the same task. Further customization was performed to ensure that the Raspberry Pi's internal internet usage was not impacted and that multiple devices could connect to the wireless network without interference. By accomplishing this setup, all network traffic sent and received from devices on the wireless network can be analyzed by the NIDS.

5.2.3 Network Traffic Capture

Tcpdump command line tools (described in section 4.3.2), was installed on the Raspberry Pi. This tool works constantly on the Pi as a daemon program to constantly capture all the network traffic flowing in and out through the Pi. This captures packets in packet based PCAP format file. This PCAP file is instantly pipelined using a bash shell script to the CIC Flowmeter program described in section 4.3.1. The CIC Flowmeter takes the PCAP file and converts it into a bidirectional flow based format which makes it meaningful and enriched for being further utilised in the Malmenator's anomaly detection model as an input. For more details regarding the working of the anomaly detection model refer to [4]. This forms the first line of defence for identifying anomalies on the network.

5.2.4 Rules Based Anomaly Detection

The Raspberry Pi hardware was also installed with Snort as described in section 4.2. Snort was configured to work as an intrusion detection system with the latest snort rules providing the ability to catch anomalies on a network level arised by an inconsistent or abnormal flow of network packets in and out of the Raspberry Pi hardware. These alerts are stored in a log file which is also transferred to the web based dashboard for reporting any issue in the network. This forms the second line of

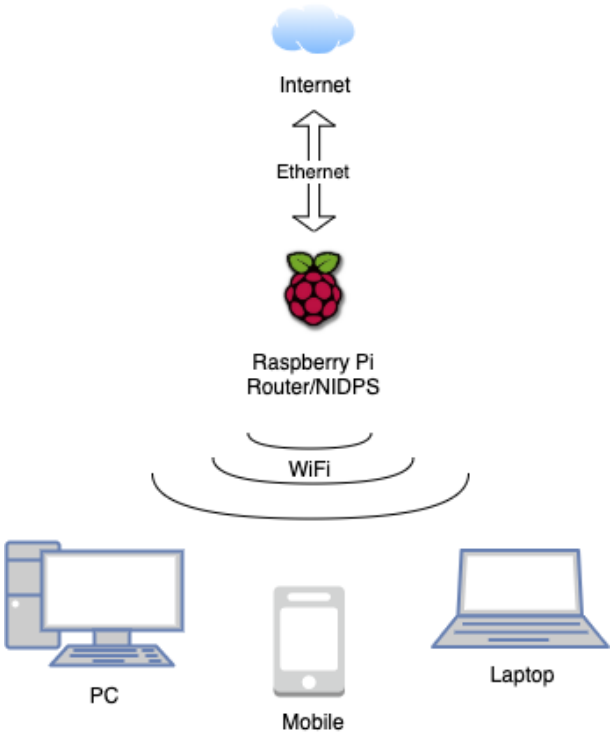

Figure 5.1: Network setup with the Malmenator network scanner

Depiction of the utilization of the Raspberry Pi with integrated NIDS features. The Raspberry Pi is directly connected to the internet through an ethernet channel and functions as a WiFi Router.

defence for network anomaly detection. More about the working of the dashboard and the web interface is described in the subsequent sections below.

5.3 Web Interface

This section gives an overview of the implementation of Malmenator's web interface for interacting with the NIDS. The web interface enables a seamless monitoring of the network traffic on the Raspberry Pi NIDS and is able to flag and notify the network administrator about any potential anomalies identified on the dashboard in almost real time.

5.3.1 Dashboard

The main functionality of the web interface is to monitor and control aspects of the Raspberry Pi NIDS. This includes features such as viewing the different devices connected to the network as well as seeing live updates and data analytics on the traffic flow. This is made possible with the use of Kibana as described in 4.3.3 where the visualisations effectively highlight any alerts received from the anomaly detection model as well as from Snort and the information regarding every bidirectional flow (network data) can be inspected swiftly making the process of identifying and eliminating potentially malicious IP seamless.

The network data analytics include network anomaly alerts along with complete bidirectional flows of ALL data in and out of the Raspberry Pi providing over 96 features about each and every flow. The 96 features include important parameters like source and destination IP along with condensed information of different protocols and flags found in the packets of each flow indexed with the timestamp of the flow from the Raspberry Pi. More details regarding the importance of bidirectional flow data for this project can be found at [4].

5.3.2 Web Architecture

The web interface is centered around the Elasticsearch Logstash Kibana (ELK) stack, three open source tools that drive thousand of data analytics projects across the world as described in 4.3.3.

The figure 5.2 details the various connections that are involved in the creation of the web interface for the Malmenator project. It starts with a bunch of users connecting to the Wi-Fi signal broadcast by the Raspberry Pi hybrid router. Note that the number of user can be more than 3 as they have been limited for depiction purposes on the diagram. The bidirectional flow data is then captured on the Pi and sent to the elasticsearch server hosted on the Amazon web services. The Raspberry Pi also has a logstash daemon running which monitors for any alerts generated using the Snort NIDS and sends those alerts to elasticsearch as well. All the flow data as well as the alerts are then visualised with customs dashboard created using Kibana which is also hosted on Amazon web services and connected to the elasticsearch database. This enables a complete solution for capturing all the network flows from all the users connected on the network and send the data to elasticsearch for visualisations using kibana.

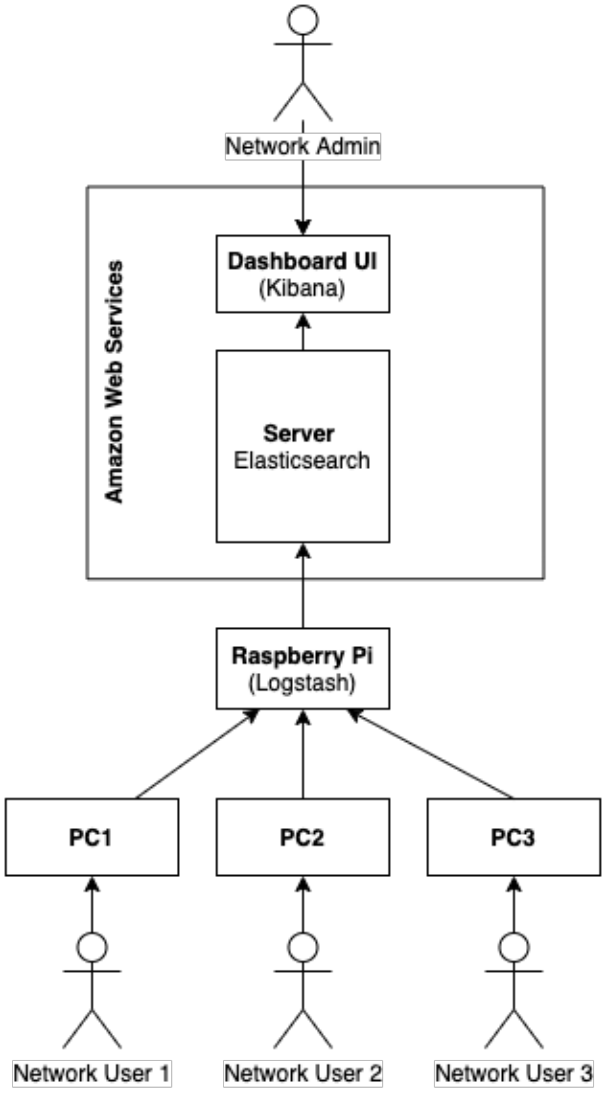

Figure 5.2: Malmenator web architecture diagram

5.3.3 System Architecture

The complete system architecture diagram is represented by the figure 5.3. Following the web architecture description in the previous subsection there are some additions in the complete Malmenator minimal viable product. As described earlier in 5.2.2 the raspberry pi hardware connects to the internet and acts as a Wi-Fi router. The blue lines represent the flow of internet traffic in the network. All the users connect to the Pi for internet access and the Pi acting as a

router in turn connects to the outside internet network, thereby ensuring that there is a single point of entry and exit for all the external network flows. Hence this setup ensures that all data in the network flows in and out of the Pi.

Following this raspberry pi runs tcpdump command line network packet logging tool as described in 4.3.2. This captures all this network packet data flowing in and out of the raspberry pi without leaving a single packet undetected. This network packet data is constantly exported to the network anomaly detection model. Moreover, the Snort NIDS running on the Pi also generates its own logs and alerts which are then pipelined into elasticsearch, as represented by the orange line, with the use of logstash as detailed in 4.3.3.

On the amazon web services, Malmenator has an instance of Elastic Compute 2 virtual machine specifically dedicated for machine learning applications containing the Network Anomaly Detection model. As soon as any network packets, the green lines in figure **??**, are received on this virtual machine the CICFlowmeter program is used to convert this packet based data into bidirectional flows as described in 4.3.1. This enriched flow based data is then fed to the network anomaly detection model which predicts and classifies each flow as anomalous or not. These results from the model along with the corresponding bidirectional flows are then sent to elasticsearch as well for centralised alerts and data capture.

Finally, as and when the data keeps flowing into elasticsearch hosted on amazon web services the Kibana dashboard is visible to the network administrator at all times which acts as an interface for monitoring the network at all times in real time. This produces a complete solution from capturing the all the network packets to analysing and predicting their maliciousness and having tools to analyse and receive alerts in real time.

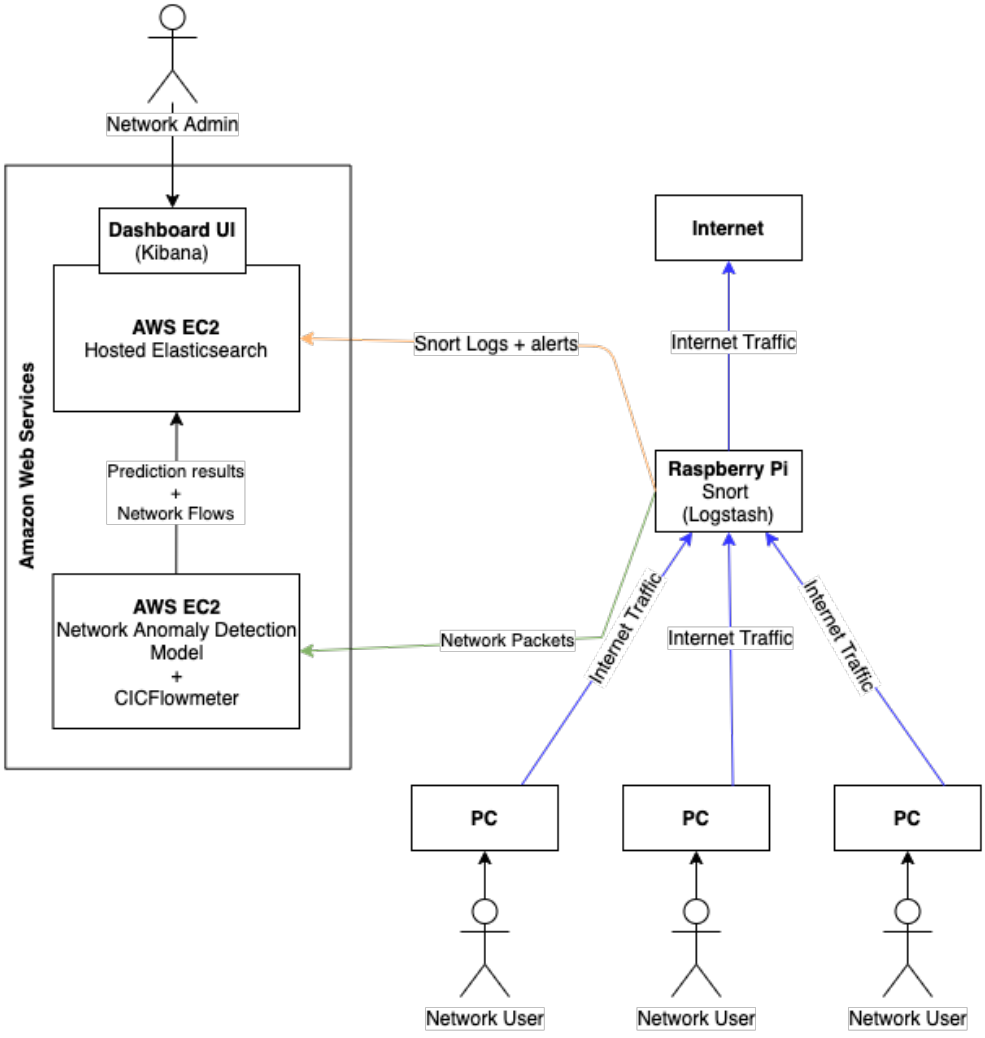

Figure 5.3: Malmenator system architecture diagram

Chapter 6

Discussion and remarks

6.1 Overview

This chapter focuses on the key achievements made by the current minimum viable product for this research based project. It will also detail the key aspects regarding Malmenator's performance and user experience for the engineering implementation. Finally, some key finding, future improvements and challenges primarily from the engineering perspective will be detailed. For more details regarding achievements and finding of the research component refer to [4].

6.2 Accomplishments

The project since its inception in August 2019 has been consistently making progress towards achieving its goals. Based on the methodology, there are 4 key components in this research-based project. These are Network Intrusion Detection hardware, Dataset Curation, Hybrid Network Anomaly Detection Model and Web Architecture. This section describes the overall progress of the project and objectively evaluates the major accomplishments done in its lifetime. Also, for the hybrid network anomaly detection model this report will only focus on accomplishments of the rules based detection and more details regarding achievements of network anomaly detection can be found in the other report [4].

The project has successfully created a hardware based network intrusion detection system (NIDS) and network packet analyser. The Raspberry Pi hardware has been successfully configured so that it can connect to ethernet and broadcast a Wifi network. Moreover, tcpdump and Snort has been

installed and configured on the Raspberry Pi hardware. This part of the project was finished in October 2019 and has been constantly in use since. Hence, the Raspberry Pi was a successful experiment for the minimum viable product for this project and it stands a great model to serve as a efficient router as well as a NIDS requiring little to no effort on the user side in plugging to the ethernet cable and start using this system. Achieving the objective of making network anomaly detection easily accessible.

Furthermore, the web architecture was established in January 2020 and was completely developed in March 2020. This has helped Malmenator ensure that the current arrangement for exporting all the alerts and network packets along with the predictions from network anomaly detection model provides a sustainable pathway to keep the big data manageable using elasticsearch. Besides, Kibana acts a great tool for constantly monitoring the network and it can effectively notify as well as visualise any anomalies or network behaviour with the help of a large variety of visualisation aids custom designed and edited for Malmenator's dashboard. Thereby, making it easy to interact with a nids and interpret network flows and network anomalies achieving another key objective for this project.

A summary of our overall progress based on the 4 key components of the project detailed in section 5 can be found in the following table.

Table 6.1: Progress Evaluation

Component Name	**Description**	**Status**
Network Scanner	Hardware and software for logging network traffic flow	Finished
Dataset Curation	Researching, extracting and analyzing public network traffic datasets for model building	Finished
Network Anomaly Detection Model	Building, training and testing the model with different machine learning methods	Finished
Web Architecture	Setup Elasticsearch, kibana and the data pipeline from network scanner to elasticsearch	Finished

6.3 Malmenator Performance

The performance of Malmenator depends on multiple connections and factors according to the design architecture found in figure 5.3. With regards to this, the first performance bottleneck can be arguably caused by the Raspberry Pi router. After significant tests the results of the connection

speeds on Pi is summarised in the table below.

Table 6.2: Raspberry Pi Network Up/Down speed

Configuration	**Raspberrry Pi**	**Traditional Router**
1 Mobile Device	Up:6 mbps Down: 11mbps	Up: 9mbps Down: 10 mbps
1 Computer	Up: 34 mbps Down: 55mbps	Up: 45 mbps Down: 60 mbps
1 Computer + 1 Mobile Device	Mobile (Up: 4 mbps Down: 8 mbps) Computer (Up: 30 mbps Down: 50 mbps)	Mobile (Up: 7 mbps Down: 9 mbps) Computer (Up: 32 mbps Down: 48 mbps)

It is evident from the table 6.2 that in terms of speed when connected to the same ISP from the identical location, there is not a bottleneck on the end of the Pi as compared to the traditional router. The download and upload speed does not differ significantly. Among all configurations speed remain nearly the same with speeds increasing for connected computers compared to mobile devices showing that the bottleneck is caused by the device and not by the Raspberry Pi. Although the performance for the traditional router overall remains better but with the limited capabilities of the Raspberry Pi the current performance can be considered as adequate and reasonable.

After constant operation and monitoring of over 30 days using the system architecture described in section 5.3.3. There were no systematic lags found in the process involving data export to elasticsearch from Raspberry Pi and AWS elastic compute virtual machine. Moreover, performance of the Kibana dashboard remains state of the art with fast querying and analysis of data even when significant storage overhead of over 20 GB was reached.

6.4 User Experience

The major components of Malmenator are server side resources and configuartions from the network anomaly detection model to Snort based NIDS, hence not too graphical. However, a key part of the user experience is formed by the Raspberry Pi hardware which is completely preconfigured and only requires the user to plug the Pi to power and attach the ethernet cable connecting it with the internet and then the Pi can act as a self sustainable router with NIDS running and network flows and alerts being analysed. In figure 6.1 the black cable is the ethernet cable in our setup and the grey cable is the USB type C power cable for the raspberry Pi.

The Kibana based dashboard forms the graphical user interface part of the project. The user

Figure 6.1: Raspberry Pi router / NIDS

can easily interact with it on a web browser without the need for any installation and configuration. It can represent a variety of data and current metrics in a single page dashboard. Some screenshots of the dashboard is provided in figures 6.2 6.3 6.4.

The figure 6.2 represents the discovery tab of the Kibana dashboard where user can see each data entry in detail, indexed by the timestamp. The user here can also filter and query the complete pool of data using Kibana Query Language (Lucene syntax) and reach to the desired datapoints for in depth analysis with extremely fast capabilities for querying.

The figure 6.3 represents the dashboard for Malmenator which can be monitored by the network administrator at all times. Primarily it carries two metric visualisations which changes color to red if any anomalous flow is detected and triggers an alert in the alert tab of Kibana. The current state is representative of the data flow captured in the last 15 minutes and also contains graphs to keep a check on any unusual spikes in inflows and outflows per 30 seconds. Also using input control

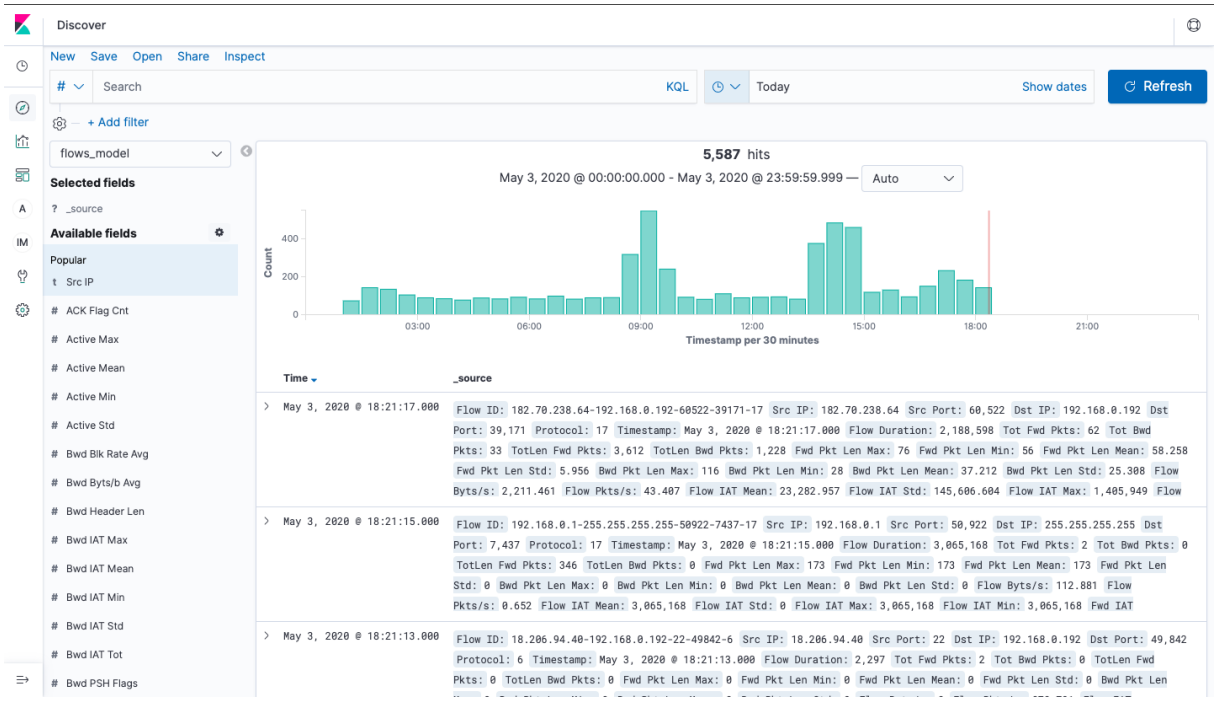

Figure 6.2: Kibana: A full day snapshot of flow data on May 3

any anomalous flow or IP can be searched from the dashboard itself and the most recent data with primary metrics like timestamp, source IP, destination IP, flow duration and flow packets per second can be seen on the summary table in the right corner.

The figure 6.4 is the continuation of the same dashboard but represents other important graphs that help the network administrator to monitor network health and check for any unusual activity easily. The pink bar graph represents the total flow duration per 30 seconds which can make it easy to detect if there is a sudden spike or loss of netflows in the network. Moreover, the smaller green chart identifies the top IPs responsible for maximal flows in a 15 minute window which can again help targeted monitoring and enable the network administrator in identifying malicious IPs if that gets undetected by the NIDS or the network anomaly detection model. Lastly, the heatmap at the bottom represents the number of total flows in 30 second windows and it will change to darker green color in a window where an exceptionally high flow is detected.

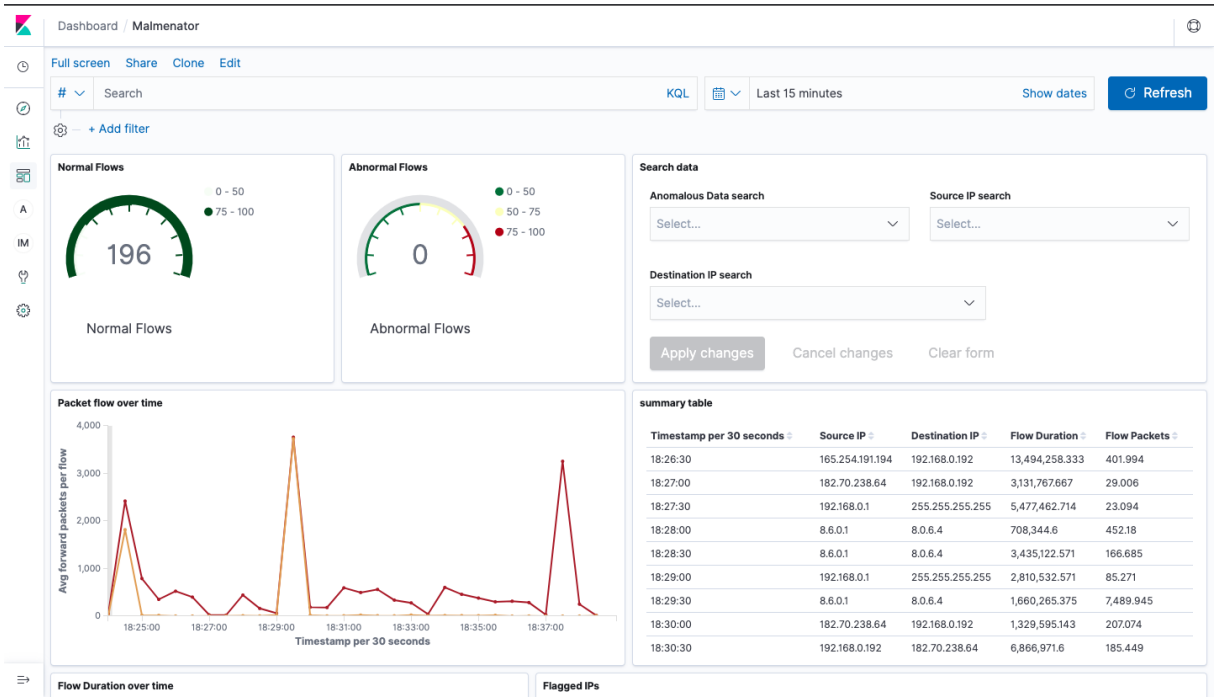

Figure 6.3: Kibana Dashboard snapshot part 1

6.5 Discussion of Findings

The Wifi network broadcasted by the Raspberry Pi has been tested by connecting several mobile devices and personal computers which were able to connect using the Pi's Wifi network. After rigorous runs and analysis over a period of 3 months, it was concluded that ALL the packets flowing in and out the Pi were being captured with the help of tcpdump. It was also found that tcpdump functions very efficiently without causing any lags on the Raspberry Pi's normal functionality.

Moreover, after more than 50 runs of Snort over durations ranging from 1 hour to 2 weeks, it was identified that on average snort was able to analyse 94% of the network packets for detecting anomalies using the NIDS rules based system. A sample summary of a 1 hour run of Snort on raspberry pi is captured in the figure 6.5 indicating a near 96% packet analysis for this run. It can also give detailed breakdown by protocols and summarise how many alerts and what kind of alerts were generated based on the applied rules.

It was found that CICFlowmeter worked seamlessly on AWS elastic compute virtual machine in conjuction with packet based data received from tcpdump program on the Pi. The CICflowmeter

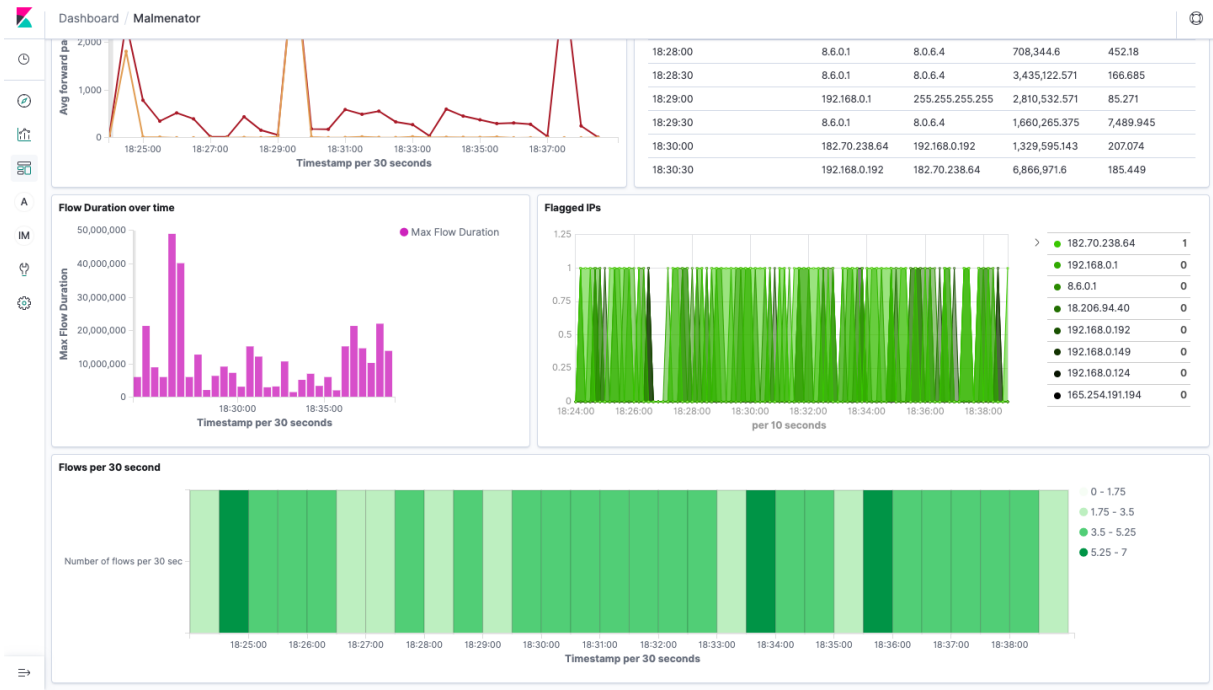

Figure 6.4: Kibana Dashboard snapshot part 2

program running on java could readily analyse and generate bidirectional flows for 100% of the traffic passed. There was negligible latency between receiving the packet data from the Pi and converting it to flows using CICFlowmeter.

The web infrastructure was found to be very resilient to the large number of data inflows and not a single flow got dropped during transfers of data from Raspberry Pi to the AWS based elasticsearch server or while automated transfer from the Pi to the AWS based elastic compute virutal machine running the network anomaly detection model.

More details regarding our findings for the anomaly detection model can be found in [4]

6.6 Future Works

As with every project there is always a scope of improvement given more time and resources are put into the development of Malmenator.

First, more experiments can be done in terms of improving the logging and monitoring on the Raspberry Pi for the Snort NIDS. In its current version if Snort is run in the intrusion detection

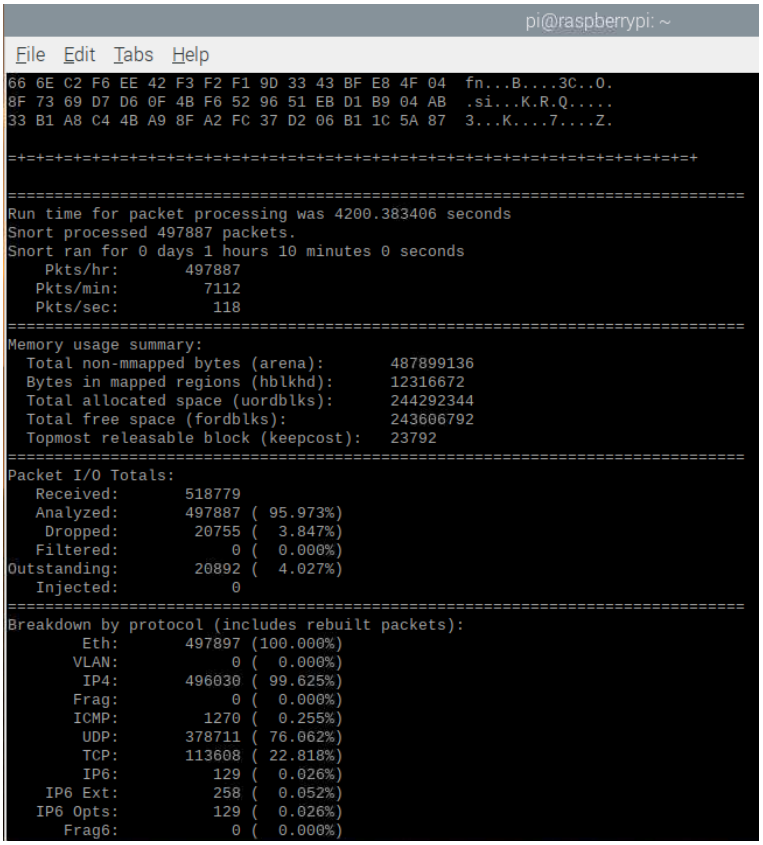

Figure 6.5: Snort sample run

mode, it is not able to simultaneously acts a packet logger which leads to only alerts being logged into the Pi and to the elasticsearch [21]. For a more comprehensive analysis with multiple points of failures in capturing the network packets, this requires multiple instances of Snort as a packet logger and as an IDS running on the Pi in order to capture the alerts as well as all the network packets which makes the resources of the Pi stretched unnecessarily . A possible solution to this can be the user of other available open source IDS such as Suricata which can possibly mitigate the problem of multiple instances of Snort being run on the single Pi.

Second, although the use of the Pi as a Wi-Fi router is viable and does not cause any major reduction in terms of download and upload speed for the user, it might not be sustainable for long term and resilient usage as the project experienced issues such as extraordinary heating of the Pi on constant and sustained operations. This makes it necessary for the project to look into alternative solutions that can be embedded into the wifi router for basic capture and transfer of data which will

be imperative on scaling up this system. Thus, the Pi makes a perfect solution for experimenting and testing as a minimum viable product but for scaling up the same architecture for actual industrial usage will require additional research.

More details regarding future improvements for the network anomaly detection model can be found at [4].

6.7 Challenges

6.7.1 Steep Learning Curve

There have been several major challenges posed in the way of the Malmenator project, the first and foremost being our team's lack of prior knowledge in the cybersecurity field. This directly resulted in us setting unrealistic expectations for our project scope which required us to reevaluate our decision and narrow our scope several times. This has resulted in us modifying our initial methodology several times to more accurately reflect feasibility and utility based on our research.

6.7.2 Scope Identification

Cybersecurity is a huge domain in itself, entailing specialisations from malware analysis to penetration testing. This made it harder to limit the scope of the project as the team started with a broad set of goals including cleaning of malware from a network as well as malware file classification. A reason for this can be attributed to the lack of prior knowledge of the team in the field of cybersecurity. The team was eventually able to identify key problems and the need for improvement as well as innovation in the niche of network security and solved this challenge by limiting the scope of research to this field by working on identifying anomalies/cyberattacks on a network level.

6.7.3 Proprietary Information

A second challenge comes from the inherent nature of the cybersecurity field. Unlike other fields such as computer vision or natural language processing, there are huge barriers in navigating and researching the field due to a lack of transparency and a general obfuscation of materials. For example, a universally used benchmark dataset does not exist due to the privacy and legal concerns over data sharing (in contrast with ImageNet for image recognition problems) as well as the ever

changing nature of cyberattacks. Furthermore, cybersecurity is a field that is prone to industry research being far ahead of academic research since academic research is accessible by malicious attackers. Lastly, experimenting with live viruses and malware pose serious safety hazards that require additional safety precautions, which further slows progress.

Conclusion

Malmenator is research based project that aims to deliver a powerful and adaptable network security tool to individuals and small organizations. By exploring powerful and comprehensive anomaly detection techniques from multiple approaches, our project thoroughly analyzes network traffic data for suspicious and unwanted activity. The NIDS hardware has been implemented on a heavily configured Raspberry Pi using a modular open source NIDS software, snort and tcpdump. Furthermore, research into the appropriate dataset for model creation has been completed in our research component, leading us to leverage a feature engineered CICIDS 2017 dataset for building and evaluating a hybrid technique for network anomaly detection.

Bibliography

[1] The White House, "The cost of malicious cyber activity to the u.s. economy," tech. rep., The Council of Economic Advisors, Feb 2018.

[2] Forgerock, "U.s. consumer data breach report 2019: Personally identifiable information targeted in breaches that impact billions of records," tech. rep., Forgerock, 2019.

[3] The Internet Society, "2018 cyber incident and breach trends report," tech. rep., The Internet Society, Jul 2019.

[4] Han, "Malmenator: Network anomaly detection," tech. rep., The University of Hong Kong, May 2020.

[5] A. Asen, W. Bohmayr, S. Deutscher, M. Gonzalez, and D. Mkrtchian, "Are you spending enough on cybersecurity?," tech. rep., Boston Consulting Group, Feb 2019.

[6] International Data Corporation, "Worldwide semiannual security spending guide," tech. rep., International Data Corporation, Mar 2019.

[7] The Royal Society, "Progress and research in cybersecurity: Supporting a resilient and trustworthy system for the uk," tech. rep., The Royal Society, Jul 2016.

[8] K. A. Scarfone and P. M. Mell, "Guide to intrusion detection and prevention systems (idps)," tech. rep., National Institute of Standards and Technology, U.S. Department of Commerce, Feb 2007.

[9] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proena, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, p. 447489, Jul 2018.

[10] J. K. K. M. H. Bhuyan, D. K. Bhattacharyya, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 303–336, Jan 2014.

[11] H. Holm, "Signature based intrusion detection for zero-day attacks: (not) a closed chapter?," *2014 47th Hawaii International Conference on System Sciences*, Jan 2014.

[12] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, p. 1624, Jan 2013.

[13] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 128, p. 3355, Feb 2019.

[14] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, p. 147167, 2019.

[15] Cisco, "Netflow version 9 flow-record format," tech. rep., Cisco, May 2011.

[16] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, vol. 239, p. 201225, Aug 2013.

[17] W. J. Yuan, J. S. Tan, P. D. Le, J. Tan, and P. Q. Dung, "Snort network intrusion detection system with load balancing approach," tech. rep., Faculty of Information Technology, Monash University, 2013.

[18] C. I. for Cybersecurity, "Network traffic flow analyzer," 2015.

[19] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide*. O'Reilly Media, Inc., 1st ed., 2015.

[20] Raspberry Pi Foundation, "Raspberry pi foundation," Sep 2019.

[21] Cisco, "Snort user manual 2.9.15," Jan 2019.