Malmenator

Network Based Intelligent Malware Detection

Final Year Project

Project Plan

David B. Han
Piyush Jha
Supervisor: Dr. Dirk Schnieders
University of Hong Kong
September 29, 2019

Contents

List of Figures

List of Tables

Chapter 1

Background

This chapter offers an overview of the technology, current literature, and market conditions of various aspects of the Malmenator project and the broader cybersecurity industry. Reading this chapter provides important insights into the cybersecurity field and is a foundation for understanding the remainder of the report.

1.1 Introduction

Cybersecurity plays a role in protecting the data, integrity, and operations of computing assets that are part of an individual or organization's network. As we migrate towards an increasingly digital economy, cybersecurity is similarly increasingly important for establishing trust in digital platforms and enabling both people and businesses to seamlessly shift towards a technologically transformed world. This section introduces the reader to threats facing cybersecurity today as well as providing an overview of cybersecurity research and ethics.

1.1.1 What is Malware

Malware is a combination of the words *mal*icious soft*ware*. At its essence, malware is any software that can damage, infiltrate, or interrupt a computer system to meet the needs of a malicious attacker. As more and more industries undergo a digital transformation, there is increasing economic and political incentive to create and distribute malware. Today, much of the focus of the cybersecurity industry is on preventing this from happening. Anti-malware efforts can be broadly categorized into two forms: (1) preventing malware from infecting a system and (2) detecting and eliminating malware from an already infected system. This project focuses mainly on the latter form.

1.1.2 Types of Malware

Malware exists in many forms, and there are a number of different ways to classify malware depending on its behavior or intent. One simple classification of malware consists of file infectors vs standalone malware. File infectors are malware that embed themselves into otherwise normal files such as pdf documents or application installers, whereas standalone malware tries to hide itself in a remote folder or hidden directory.

A more robust form of malware classification consists of classifying malware by both its underlying behavior and its overarching goals. The details of the classification are in table 1.1 and are used throughout the remainder of this report. Note that this list is not exhaustive and that some malware can be classified as more than one type. It is also important to note the difference between *exploits* and *attacks*. Cyberattacks such as distributed denial of service (DDoS) attacks revolve around overloading a system to take it down, which is different from exploitative malware which take advantage of system vulnerabilities, such as backdoors. Malmenator focuses on the latter.

1.1.3 Operating System and Architecture Types

Creators of malware often aim to have have their malware infect the largest number of devices possible. Since malware needs to take advantage of specific system vulnerabilities, malware written for one operating system (OS) or platform usually does not work on another. Hence, most malware is written for personal computers (PCs) running Windows or mobile devices running Android as they have a majority of the market share. As of August 2019, 78% of desktop computers run Windows

Table 1.1: Malware Classifications

Type	Description
Viruses	Executable code that is embedded in an executable file such as an installer. Viruses spread when files are shared between machines, and once a virus is active on a computer it can spread to other files. Viruses are usually first triggered by opening an infected file.
Worms	Worms differ from viruses in that they can run on their own. They are able to quickly replicate themselves and embed themselves through different files. They also actively look for pathways between computer systems on a network such as a common file storage location.
Trojan Horse	Similar to a virus, except it embeds itself to non-executable files such as image or movie files rather than executable ones.
Logic Bomb	Logic bomb codes remain in hibernation and are inactive until a trigger event occurs such as a reaching a certain date or waiting till a set amount of time elapses. Logic bombs implement malicious code that harms the hardware components of a computer such as a hard drive or power supply by sending them into overdrive until they fail.
Ransomware	Ransomware encrypts and locks down a computer system rendering it unusable by the victim. To unlock the device, the victim needs to pay a ransom to the attackers to release the encryption.
Backdoors	A backdoor gives an attacker remote access to a system.
Rootkits	A rootkit is a specific type of backdoor that allows an unauthorized user to gain root/admin privileges on a system.
Spyware	Malware to steal private information from a computer and send it to an attacker.
Keyloggers	Keyloggers are a sub-type of spyware that records a user's keystrokes in order to obtain passwords and other sensitive information.

OS and 76% of mobile devices run some flavour of Android [1]. Malware for other OSes such as Mac OS X and iPhone's iOS also exist, but they affect a smaller number of users and are generally not as robust or well developed.

1.1.4 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 [2]. In 2019, global spending on cybersecurity initiative is expected to exceed $100 billion USD [3]. 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 [4]. 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 [5]. 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 1.1 depicting the various economic impacts of cyberattacks and their respective difficulties in quantifying cost [6]. 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.

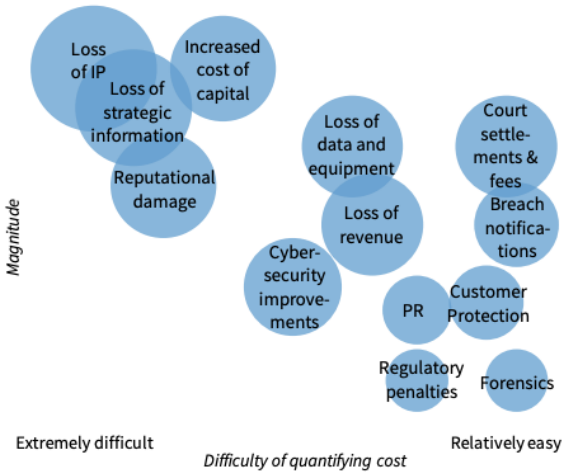

Figure 1.1: Estimating financial losses to cyberattacks

1.1.5 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 in table 1.2, but the key takeaways are that white hat hacking is done with good intentions, while black hat is not. Most, if not all, academic research in the field is a white hat effort, this paper included.

Table 1.2: Types of Malware Researchers

Type	**Description**
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.

1.1.6 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 recent published academic research in malware detection have huge challenges in practical implementation such as utilizing deep learning approaches for malware detection. These approaches and their shortcomings are discussed later in this chapter.

1.2 Current Approaches to Anti-Malware

Anti-malware efforts are growing rapidly in order to respond to the quickly shifting landscape of malware attacks. This section summarizes how top companies and researchers approach and create anti-malware solutions.

1.2.1 Major Players in the Industry

Cybersecurity and anti-malware is a field dominated by private corporations and governments. These corporations consist of cybersecurity specialists such as McAfee, Norton, Symantic, Crowdstrike, and Palo Alto Networks as well as large technology companies with a strong cybersecurity division such as Microsoft, IBM, and Cisco. They offer a variety of cybersecurity solutions that include security training for employees, firewalls, cloud-based detection system, and endpoint detection systems.

1.2.2 Static vs Dynamic Analysis

There are a number of techniques for classifying files as malicious or benign, but most methodologies fall under two large umbrellas: static analysis and dynamic analysis.

Static analysis involves looking at the contents of each file without executing it. Static approaches include checking file types and sizes, verifying checksum and hash values, and matching signatures. Strange filenames can also be used to identify the existence of a malware. For example, certain families of malware are known to generate secondary files of a given name, size, and type. The existence of these files can be used as an indicator for the existence of a malware. Checksums and hashes are used to verify the integrity of a file to ensure that it has not been tampered with during a transfer. Hashes of a specific file can also be used to rapidly check against online databases without opening or executing a file. Lastly, signatures are sequences of bits that are known to appear in malicious files but not in harmless ones. Signatures can be matched against databases to check for similar malware. However, the classic signature approach has not been used independently at scale for nearly two decades now [8].

Dynamic analysis tracks the behaviour of a file by executing it in a controlled environment or sandbox and looking at features such as memory locations it accesses or the network packets it sends out. Other features include file path locations, registry keys, the creation and reading of other files in the system, and domain names and IP addresses of any communication and any attempt to download additional malware files [9].

1.2.3 Firewalls

A firewall is a program or device that filters network traffic coming through the internet into a private network or computer system. Some firewalls can block entire internet domains, while other implementations of firewalls analyze network traffic for anomalies and take appropriate action, such as throttling traffic in the event of a denial of service attack. Comprehensive network analysis can

also be used to search for exploitative behavior caused by malware such as backdoors by scanning for abnormal traffic and port usage. There are three main methods by which firewalls control network traffic:

1. Packet Filtering: Network packets are discarded based on a set of filters. Packets that are flagged by the filters are discarded and do not make it through the firewall.

2. Proxy Service: Some firewalls can act as proxy servers to get data from the internet on behalf of a computer. The data retrieved from the internet is then inspected before it is passed to the computer.

3. Stateful Inspection: A newer method of network analysis that compares certain components of the packet against a trusted database. Information travelling into the system is also compared to information travelling out of it to ensure that only relevant traffic gets through the wall.

Malmenator's network scanning will focus on stateful inspection and packet filtering utilities.

1.2.4 Uses of Machine Learning

Due to the rapidly evolving nature of malware, many rules-based systems quickly become obsolete. Machine learning is quickly rising to fill this gap and is developing to be an integral component of malware detection algorithms. According to Kaspersky Lab, a leading cybersecurity firm, there are four key objectives that should be met when utilizing machine learning for practical malware detection in order for it to have practical usage: (1) the model needs to be interpretable, (2) the model needs to have a low false positive rate, (3) the model should be adaptable to counteractions, and (4) the model should be trained on a large representative dataset [10]. Independent research in an academic environment further confirms these core objectives for the usage of machine learning in malware detection [11].

There have been significant strides in applications of machine learning in sorting and classifying known malware into a set of different types. In 2017, a new method was proposed using control statement shingling to achieve an accuracy of 99.21% accuracy in classifying 10,260 malware files into different families by extracting features from disassembled binaries and using random forest algorithm [12]. However, it appears to be a much harder task is to determine whether any given file is malicious or benign as many of the methods that show promising results fall short in one of the four key areas listed above.

A large number of methods for classifying benign and malicious files published in academic research utilize deep learning and lack interpretability. One recent approach has been to convert binary files into greyscale images and then run convolutional neural networks (CNNs) on the images, effectively turning a malware classification problem into an image classification problem. [13]. This approach was able to achieve 94% accuracy on malware designed for internet of things (IoT) devices, but is a black box that cannot be understood by cybersecurity experts. The relationship between

a pictorial representation of code and its functionality is unknown. Additionally, the model has a fairly high false positive rate where 5.33% of benign samples are flagged as malicious, rendering it impractical for commercial implementation.

Other research in the space propose solving some of these challenges with a more modular approach as opposed to the monolithic approach most models take today [11]. Due to the existence of numerous variations of malware, the research recommends training separate detection models for different types of malware such as trojans, backdoors, worms, etc. instead of a single model to detect malware files as a whole.

A white paper published by Kaspersky Lab's, a leading cybersecurity firm, gives a high level overview of another approach for how they effectively combine different machine learning methods with rule-based classification of malware [10]. They first divide up the feature space into regions. In regions where all samples are either clean or infected, they use a straightforward rules-based classification system. In boundary regions where there are both harmless and malicious files, ensemble machine learning techniques are used and optimized for each region. The Malmenator project aims to build upon this approach.

1.3 Other Relevant Technologies

This section covers other technologies that will be utilized throughout the Malmenator project.

1.3.1 Snort

Snort is an open source network intrusion detection system (IDS) that can be installed on a variety of OS distributions to monitor network traffic to and from the system. It has open source libraries containing rules to check for abnormal traffic behavior including malicious attacks and other undesirable behavior.

1.3.2 Raspberry Pi

Raspberry Pi is a low cost computer that is highly customizable and can be utilized for a wide variety of purposes ranging from hosting a Minecraft server to replacing a desktop computer.

1.3.3 Linux Kali and Metasploit

Special tools are necessary for the creation and deployment of malware even when one is doing so for a white hat agenda. Linux Kali is a Debian-based Linux distribution that specializes in providing tools for advanced penetration testing and ethical hacking. Metasploit is an opensource framework and penetration testing platform for finding, exploiting, and validating vulnerabilities. Metasploit is natively available on Linux Kali and they work in unison as a powerful tool to create custom malware.

1.3.4 ELK Stack

ELK stands for Elasticsearch, Logstash, and Kibana. They are three opensource projects that are used for a huge number of data analytics projects across the globe. Elasticsearch is a search and analytics engine. Logstash is a serverside data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch.

Chapter 2

Objectives

This chapter offers an overview of the project aims and details what deliverables the project will have at the time of final submission.

2.1 Motivation

Cybersecurity is becoming increasingly important as technology transports various aspects of our lives. This is especially apparent in large corporations such as banks where a single breach can cost millions in damages and fines as well as a loss of reputation that can impact business for years to come. Today, enormous amounts of money are being spent on anti-malware solutions that are costly both in terms of money (licensing/subscription fees) and resources (long scanning time/high CPU requirements).

This project aims to explore and test novel ideas to better detect malware and reduce costs at the same time by analyzing network traffic for anomalies and eliminating the need to constantly run antivirus software on individual machines.

2.2 Deliverables

This section details the four aspects of deliverables of the Malmenator project. Each aspect contains core features as well as additional features that may be worth exploring given the time constraints.

2.2.1 Network Analysis

The project aims at using Network Analysis to sniff network packets primarily network traffic going in and out of a cluster of PCs on the same network for identifying malicious network traffic and potential malware inside the network of PCs.

A successful deliverable for network analysis would include a functioning version of Snort running on the Raspberry Pi with the ability to raise alerts when a sample backdoor malware program is ran on any PC on the network.

Additionally, a good to have deliverable would be our ability to write our own Snort rules to identify other types of malware on the network and contribute to the Snort open source community rules.

2.2.2 Malware Detection

The project approaches malware detection as a classification problem to identify the probability that a given file carries a malware or not.

A successful deliverable for malware detection would include a Tensorflow based classification model trained on features of a file obtained from a combination of static and dynamic file analysis to classify a file as malicious or not with a considerable accuracy.

Moreover, a good to have feature would be to look into combining machine learning along with a signature based malware identification system for a strong accuracy and better real world adaptability of the system

2.2.3 File System Scan Optimisation

The project aims at eliminating redundant scanning of the file system in the PC over and again by traditional anti-virus solutions, thereby optimising the file scanning procedures.

A successful deliverable would include an implementation that can identify the files that are modified after an infection on the network is identified and the scan is run only on the directory tree that consists of such modified files in the system.

Furthermore, an advanced version of this feature that is worth exploring would be a file system risk analysis so that each drive or directory on a PC can be categorised into different risk levels and the scans can be limited to only the most risky directories.

2.2.4 Web Interface

The project includes a way to easily visualise the network traffic, key metrics regarding every PC on the network, potentially risky PCs on the network and statistics about the files scanned and malware identified.

A successful deliverable includes the following:

1. An Elasticsearch database running on Microsoft Azure or Raspberry Pi according to feasibility which will be identified after testing.
2. Logstash installed on Raspberry Pi to send the data regarding network analysis to Elasticsearch.
3. Filebeat installed on each PC in the network to receive key metrics regarding every PC such as OS stats, process info and many other parameters from system logs of the PC.
4. A Kibana based analytics dashboard to visualise all the collected data in Elasticsearch in real time.

Moreover, another good to have feature would be an interface which would allow taking actions such as controlling the running processes and removing any suspicious files if needed on any PC in the network.

Chapter 3

Methodology

This chapter explores the necessary steps to complete the project. The project consists of two phases, which will be completed in the first and second semester respectively. The first phase will focus on completing the network scanning hardware and software as well as a basic web interface for reporting. The second phase will focus on the malware file classifier and complete the web interface to unify functionality of both phases in a single platform.

3.1 Network Scanner

This section describes the process of building a hardware device to monitor network traffic. This will be completed in phase one of our project.

3.1.1 Raspberry Pi Hardware

A Raspberry Pi 4 Model B as introduced in section 1.3.2 will be utilized to implement the network scanning software. The Raspberry Pi 4 has tremendous upgrades from it's predecessor, most notably the huge jump in processing power from 1 GB to 4 GB of RAM. Furthermore, due customizable nature and compatibility with open source distribution of Linux, a Raspberry Pi 4 can be set up to become the centerpiece hardware for our network scanner.

3.1.2 Hardware Setup

There are two main OS options for running on our Raspberry Pi, namely the Raspbian OS or the Kali Linux distribution. Raspbian has the advantage of being an OS that was tailor made for the Raspberry Pi, while Kali Linux includes a suite of penetration testing tools on installation. Further research will be done on which OS is more suitable for turning our hardware into a "plug and play" network scanner.

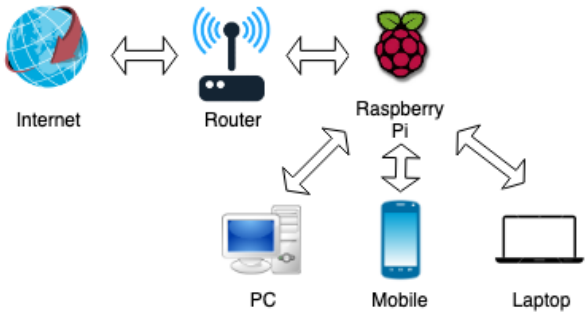

Figure 3.1: Network setup with the Malmenator network scanner

The Raspberry Pi hardware will be utilized as depicted in figure 3.1 where all network traffic is sent through the network scanner rather than directly through the router. This has the benefit of allowing network traffic of all devices on the network to be monitored and analyzed and also allows for easy setup and usage.

3.1.3 Utilizing Snort

Snort, introduced in section 1.3.1, will be installed on the Raspberry Pi based network scanner. Installing Snort on the network monitor should be fairly straightforward once the Raspberry Pi is set up as there is plenty of documentation online on how to do so. The network data generated on the Raspberry Pi will be sent to a server on the cloud for further data analytics and reporting. In addition, the project will involve creating our own custom rules on Snort to filter for any abnormal internet traffic or port usages. These rules will run on the local network scanner, and any rule-based alerts will be sent to the cloud. To create rules, it is necessary to first look at their open source library of malware detection rules to learn more about the current approaches. Then we will run our sample malware and monitor network behavior to see how we can improve the current filters.

It may prove to be a challenge to configure snort on the network scanner since the network scanner is only acting as an intermediary to route traffic rather than an internet end point. Furthermore, port scanning functionalities may be limited as we are running snort on the Raspberry Pi and not on the individual computers in the network.

In the event the port scanning functionalities are limited, it is also possible to use additional packages such as Filebeat to log computer usage data.

3.2 Malware Dataset

This section includes various methodologies for gathering sample malware to test the efficacy of our system. This portion of the project will be a continuous and ongoing process throughout phase one and two. The malware dataset will be limited in scope to backdoor malware at the beginning unless sufficient progress is made whereby we can progress to other forms of malware. We limit the scope of malware to backdoors because the behavior and variety of malware is too extensive as described in section 1.1.2. Backdoors by nature also require an active connection to a malicious command center, making them prone to detection by network scanning. Additionally, many backdoors are written in such a way to avoid detection by IDSs like Snort [14]. Thus, by making strides in backdoor detection, Malmenator can have a concrete contribution in the open source cybersecurity field.

3.2.1 Creation of Backdoor Malware

The best way to know with certainty the functions of a malware is to create it. Using Linux Kali and the Metasploit framework discussed in section 1.3.3, we will create a backdoor executable and attach it as a payload to a normal software installer. Once this malware is up and running, more advanced techniques and customizations can be added onto it.

3.2.2 Online Samples

Online libraries of malware that have been caught in the wild via honeypot traps - systems set up for the express intent of capturing and studying malware in the wild - are available for research purposes upon request. However, these malware are very much alive and may have unintentional effects that are difficult to account for and control. These samples of live malware will not be used until later stages of testing.

3.3 Malware File Classifier

This section details the different approaches we aim to combine in creating our malware file classifier that are originally described in section 1.2.2. This portion of the project will be completed in phase two.

3.3.1 Static Analysis

From a preliminary basis, we will take standard approaches to static analysis of files with a focus on file signatures using a rules-based system. Static analysis will primarily be used to filter out files we are certain are clean.

3.3.2 Dynamic Analysis

To obtain features for dynamic analysis, we will run files inside of a controlled sandbox such as Cuckoo sandbox whereby we can get information on many different aspects of the file via a convenient API. Sandboxes are essentially virtual machines that mimic a real OS environment where code can be executed in order to observe their behavior in a safe manner. We can then set rules based on these aspects as well as utilize them in our machine learning model.

3.3.3 Machine Learning

The machine learning model will follow the approach outlined in Kaspersky's Machine Learning in Malware Detection white paper and will focus on being interpretable and having a low false positive rate [10]. We can divide up the feature space by clustering. In homogeneous spaces where all samples are either clean or malware, we can directly use rules. In border region, we can use a blend of machine learning techniques that are optimized for each regions. The exact models used and where the border regions lie will depend on future exploration of the data.

3.3.4 File System Scan Optimisation

To have a malware file classifier be usable, it must run in a reasonable time frame. To do so, there must be constraints on the number of files it processes. Once unusual network traffic is detected

from a machine, we can then identify which files have been executed or modified since that time and scan all relevant files. This can be done by looking at Windows log file history.

3.4 Web Interface

This section provides an overview of the functionalities and implementation of our web interface. This portion of the project will be continuously refined throughout the course of our project.

3.4.1 Web Architecture

We will have a unified web portal hosted on Microsoft Azure's cloud services powered by ELK stack, introduced in section 1.3.4. Traffic data passing through the network scanner will be sent to the Elasticsearch database hosted on cloud for data analytics. A high level depiction of the web architecture can be found in figure 3.2. The dashboard will be powered by Kibana, which will

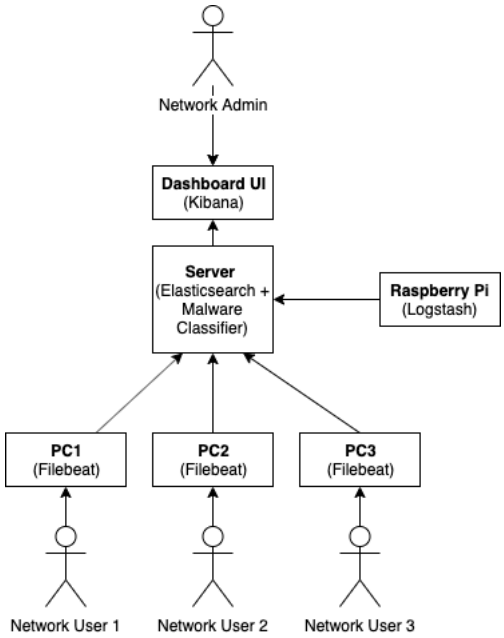

Figure 3.2: Malmenator web architecture

take data from Elasticsearch hosted on the cloud server. The Raspberry Pi network scanner will feed information into the elastic search instance via Logstash. We may also implement Filebeat on individual PCs to report their information to the server for monitoring and to further strengthen the risk profiling functionality. The malware classification model will also be running on the server and can be accessible via api.

3.4.2 Dashboard

The web interface will include a dashboard for users to view information regarding their network traffic in real time using the ELK stack. The exact information revealed on the dashboard has not yet been finalized, but will include alerts, IP addresses that are accessed, and locations of network traffic origins from the Snort logs. In addition to general network analytics uploaded purely from the Snort logs, there will also be reporting for information on an individual PC level for which PCs are at risk using data from Filebeat.

3.5 Summary

A depiction of the overall Malmenator architecture can be found in figure 3.3. The red lines denote

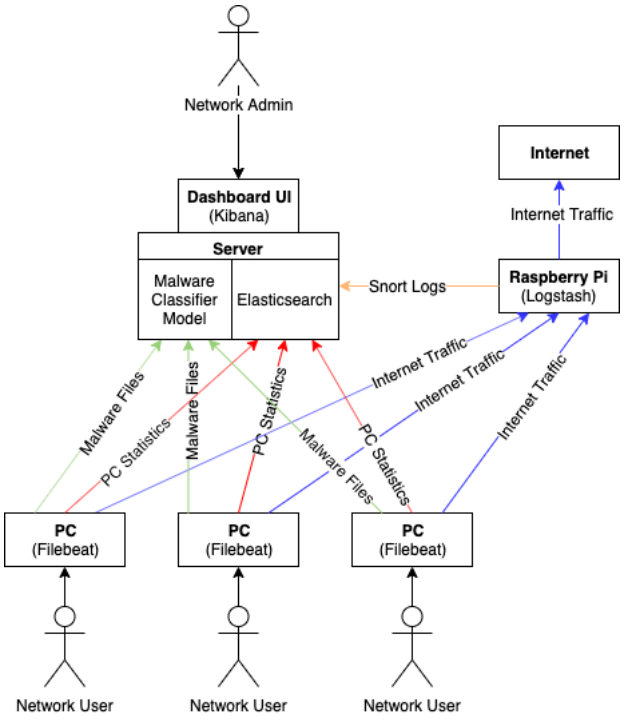

Figure 3.3: Overview of Malmenator architecture

FileBeat data, green lines denote malware files sent to the malware classifier model on cloud, orange lines represent snort log data, and the blue lines denote internet traffic generated by the end users. Network users will connect to the internet via the Raspberry Pi network scanner/router. The Raspberry Pi will be running Snort and sending data to Elastic on the cloud. Meanwhile, individual network user's PC's will be running an instance of Filebeat to send usage statistics to Elastic. The individual user will also be able to send files to our malware classifier for scanning. All of this can be

managed by the network admin through a comprehensive dashboard implemented on Kibana. This setup allows us to keep our project modular and structured.

3.6 Timetable

Our project schedule can be found in table 3.1. We aim to complete phase one by the end of December and complete phase two by the end of March.

Table 3.1: Planned project timetable

Date	**Description**	**Deliverable**
Aug 22	Initial meeting with supervisor	
Sep 20	Second meeting with supervisor	
Sep 29	Project proposal and website	Project plan and website
Oct 2	Sample backdoor file	Backdoor putty.exe
Oct 31	Snort rules to detect backdoor	Snort backdoor detection
Nov 30	Snort Setup on a Raspberry Pi	Network sniffing hardware
Dec 31	Web UI hosted on the cloud	Malmenator.com
Jan 20-24	First presentation	
Feb 2	Preliminary implementation and interim report	Interim report
Feb 28	Malware classification model	Fully trained classifier
Mar 30	Unified network scanner, classifier, and web UI	Functional product
April 1	Project in production	
Apr 19	Finalized implementation and final report	Final Report
Apr 20-24	Final presentation	
May 5	Project exhibition	
June 3	Project competition	

Bibliography

[1] StatCounter Global Stats, "Mobile operating system market share world wide," Sep 2019.

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

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

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

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

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

[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] Kaspersky, "Antivirus fundamentals: Viruses, signatures, disinfection," Oct 2016.

[9] A. Mujumdar, G. Masiwal, and B. B. Meshram, "Analysis of signature-based and behavior-based anti-malware approaches," *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, vol. 2, p. 20372039, Jun 13AD.

[10] Kaspersky Lab, "Machine learning in malware detection," tech. rep., Kaspersky, 2019.

[11] S. Saad, W. Briguglio, and H. Elmiligi, "The curious case of machine learning in malware detection," *Proceedings of the 5th International Conference on Information Systems Security and Privacy*, Feb 2019.

[12] M. Hassen, M. M. Carvalho, and P. K. Chan, "Malware classification using static analysis based features," *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov 2017.

[13] J. D. Su, V. D. Vasconcellos, S. D. Prasad, S. D. Daniele, Y. D. Feng, and K. D. Sakurai, "Lightweight classification of iot malware based on image recognition," *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Feb 2018.

[14] D. Chiu, S.-H. Weng, and J. Chiu, "Backdoor use in targeted attacks," tech. rep., Trend Micro, 2014.