Appendix A

MSC CYBER SECURITY ENGINEERING

CAN I TRUST IT?: OPEN SOURCE INTELLIGENCE IN DEFENCE AGAINST PHISHING

O. Fatodu

SOUTHAMPTON SOLENT UNIVERSITY

SCHOOL OF MEDIA ART AND TECHNOLOGY

SEPTEMBER 2022

SOUTHAMPTON SOLENT UNIVERSITY
SCHOOL OF MEDIA ARTS AND TECHNOLOGY

MSc Cyber Security Engineering

Academic Year 2021-2022

O. Fatodu

Can I Trust It?: Open Source Intelligence in Defence Against Phishing

Supervisor: Dr Andy Farnell December 2022

This report is submitted in fulfillment of the requirement of Southampton Solent University for the degree of MSc Cyber Security Engineering

Acknowledgment

Foremost, I would like to express my sincere gratitude to my supervisor Dr Andy Farnell for the continuous support for my MSc study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my MSc programme.

I would also thank my friends and course mate Dr Simon, Joseph Obadina, Seun, Edward Kelechi, Adtiya, Francis and Osaz for the stimulating discussions, sleepless nights and for all the fun we had in the past one year.

Lastly, I would like to thank my wife Adenike Oladimeji and children Oluwafirepemi and David Oladimeji for their support, encouragement and tolerance throughout the period of this study. I am so grateful and cannot love you less.

Abstract

This study aimed at developing an open intelligence source which has spam detection and filtering functionality against phishing activities via email. The proposed system was designed with the aim of catering for the limitations observed in existing spam detection software. In line with the general objective of the study, the alternative system to be designed and developed was named ‘Can-I-Trust-It’. The artifact was designed using PHP, as well as, html, CSS, jQuery tools and MySQL tool. The spam filter is designed to identify emails which hackers use to send unwanted or dangerous content to potential victims of phishing attacks. It is built to use different filtering methods to identify the content of emails of their senders and then flag the email as spam. It also examines an email for explicit content that could contain malicious links. It is built with different spam filtering algorithms using blacklists of keywords, headers, languages, domains etc. which enables a scalable knowledge base system that can be fully customized according to user preferences. The system is also suitable for analyzing offline bulk email correspondence checks and forensic investigations within an organization.

Table of Contents

List of Figures

Acronyms

AI	-	Artificial Intelligence
CSS	-	Cascading Style Sheets
CSV	-	Comma-Separated Values
F2T2EA	-	Find, Fix, Track, Target, Engage, Assess
HTML	-	Hypertext Markup Language
IBM	-	International Business Machines Corporation
ID	–	Identity
IDE	-	Integrated Development Environment
IMAP	-	Internet Message Access Protocol
IP	-	Internet Protocol
IT	-	Information Technology
KNN	-	K-Nearest Neighbour
MDA	-	Mail Delivery Agent
MTA	-	Message Transfer Agent
MUA	-	Mail User Agent
MVC	-	Model-View-Controller
OPSEC	-	Operations Security
PDF	-	Portable Document Format
PHP tools	-	Personal Home Page Tools
POP3	-	Post Office Protocol
RDBMS	-	Relational Database Management System
SMTP	-	Simple Mail Transfer Protocol
SQL	-	Structured Query Language
SVM	-	Support Vector Machine
URI	-	Uniform Resource Identifier
URL	-	Uniform Resource Locators
VSC	-	Visual Studio Code

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Web activities form an integral part of everyday life and have attained a state of indispensability among members of the society. Advancements in education, entertainment, businesses, leisure etc. are being tied to web-based activities. This implies that, as technology advances, the use of networks is emerging in every aspect across the society. This dependence on the internet poses a risk of attacks on the various web related networks that exist individually and collectively. Thus, the security and safety of computer network has begun to take center stage in the current world of IT (Otoum & Nayak, 2021). Malicious attacks and hacks of computer based networks and wireless systems have become a major concern for corporate organizations with web-based presence. This is because the conventional security packages including antivirus software and firewalls have become insufficient to maintain the integrity, reliability and security of networks (Goel & Mehtre 2015). Firewalls and antivirus programs cannot protect computer networks as they are only able to identify specific attacks that emanate from outside the network. Thus, scholars and researchers have begun to increase their interest in alternative methods of web-based securities. This has led to a proliferation of other methods which can serve as additional security measures to complement existing security measures being adopted to protect a network. A spam filtering system for detecting phishing attempts is one of such novel technologies that have the potential to guarantee and secure a network from third party hacks or intrusions.

The concept of phishing may be viewed from different perspectives in terms of its meaning and may also be identified as brand-spoofing, carding, pharming, fraud attack, semantic attack etc.; however, the basic description of phishing still remains the same. Phishing simply

describes the use of deceitful activities to scam victims into revealing personal information and details which can be used to compromise specific domains of the victims (Kenkre, Pai & Colaco, 2015). Phishing at the web level therefore focuses on gaining access to web-based domains of individuals or groups. At the web-based level phishing could be described as a fraudulent process in which a look-alike web page is developed with similarities of an existing web page to deceive users into revealing personal details and/or access codes to sensitive information such as financial details. This is often achieved by using devious means to ensure that the user clicks links to such look-alike web pages and follow the emanating prompts (Malek, Trivedi & Shah, 2020). The diagram depicted as figure 1 below highlights the process flow and phases of a basic phishing attack.

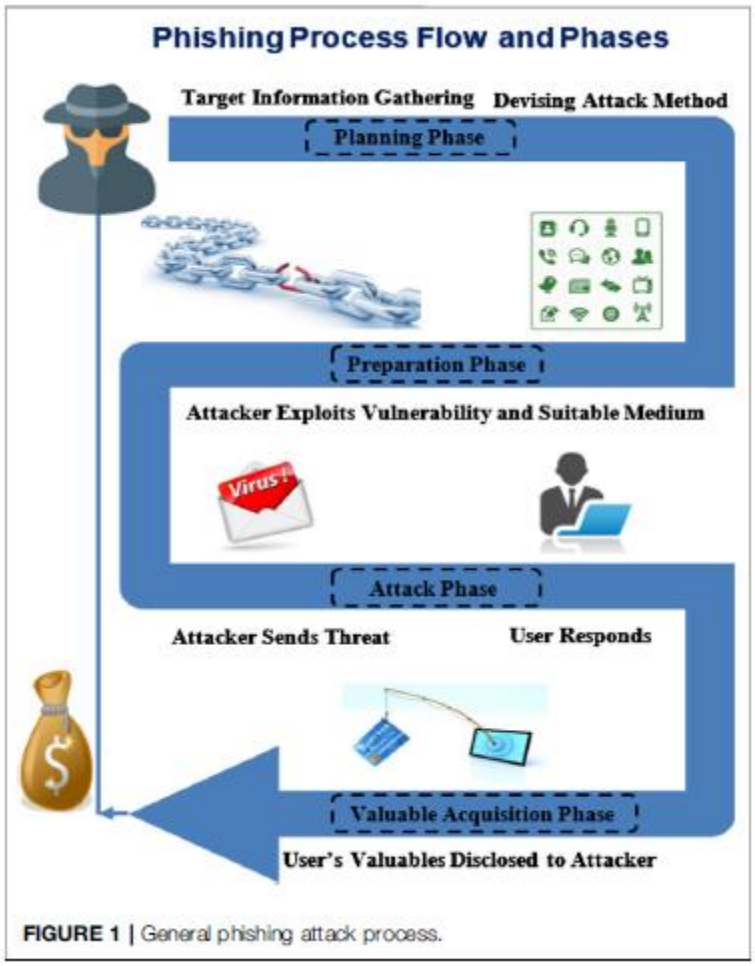

Figure 1: General Phishing Attack Process

Figure 1 shows a flow process of a basic phishing attack in 4 phases. These phases are detailed in the proposed phishing anatomy. However, as shown in Figure 1, the process of basic phishing attacks is initiated by collecting data and personal details of the victim. Then, as the first step in the planning phase, phishers decide which attack method to use in the attack. The second phase is the preparatory phase, where phishers begin looking for vulnerabilities that could catch the victim. In the third phase, phishers carry out the attack and wait for a response from the victim. The attacker could then gain access to the victim's finances during the valuable acquisition phase, which is the final step in the phishing process. Using the above phishing process as an example, an attacker could pretend to be from the victim's bank and send fraudulent emails to Internet users asking them to verify their banking account details for updating, otherwise, their account may be blocked. Users can consider this email to be legitimate because the perpetrator uses the same graphic elements, brands, and colors as legitimate banks. The personal details are then sent directly to the phisher, who uses it for a variety of malicious purposes such as withdrawing money, extortion, or further fraudulent attempts.

A common limitation that all phishing detection technologies may have been their inability to provide 100% accuracy in detection. This is because; there is every possibility of false positives and negatives being recorded as anomalies (Sharma, et al., 2019). A false positive is simply a case in which a normal event is detected by the system as being an anomaly. This may be due to a new signature embedded in the normal event which the system identifies. A false negative occurs when the system identifies an anomaly as a normal event. It is however common for developers of warning systems for phishing to focus on reducing false negatives irrespective of a potential increase of false positives. A lot of effort has been put into the production of universal datasets in form of keywords, urls, ip addresses, and email addresses (Verma et al., 2020), still yet there exists a significant dearth in datasets that should have theoretical basis for

the modeling of normal behavior. Approaches based on the identification of anomalies are efficient in identifying both known and unknown attacks; and are sometimes able to identify attacks that appear normal. It is therefore left to the end user to examine and analyze the detected behavior by system in order to establish if it is an attempt at a phishing attack (Rai, Devi & Guleria, 2016).

On the other hand, approaches based on signature systems identify attacks with known signatures thereby enhancing the ability of the system to identify specific signature attacks. This approach provides a more reliable solution for specific attacks which are common threats to web domains of the user (Thakkar & Lohiya, 2020). Therefore, the development of filter systems which are specific to web based phishing signatures are more likely to yield a 100% success rate for identifying such attacks. This provides a basis for the use signature databases in developing early warning systems which helps to notify users of potential web-based attacks at the point of entry (Aldweesh, Derhab & Emam, 2021). A signature based early warning systems should have the capability of creating real-time notifications of its activities and important events it identifies and stores. These notifications could be in form of e-mails, flash messages or symbolized icons that notifies end users of these events. These notifications therefore serve as prompts for the end user to access the system and analyze the event. The event storage process of signature based early warning systems is the generation of a detailed report of these events. The report could be generated as an initial summary, which can then be expanded for more details of the events. The signature based early warning system should also have settings that can be enabled and modified by the user to ensure that user preferences and machine learning capabilities can be obtained.

1.2 Statement of Problem

Within the context of modern threats, the Internet of things and computer related networks are beginning to gain scholarly attention in relation to security issues. The focus on security of the

cyberspace describes the use of cyber-related tools and processes aimed at protecting cyber networks, computer systems, computer software, and data repositories from external attacks, unauthorized access, tampering, or destruction. All of these forms of intrusion have the sole goal of compromising the integrity of the system being attacked (Baig et al., 2017). Effectively identifying attacks in a distributed, resource-constrained, and continually evolving space can be a daunting task. Faced with such challenges, humans may be unable to make the right decisions concerning the type of design to be used.

In addition, the profiles of attacks, in terms of purpose and capabilities, have changed significantly. Some decades ago, cyber offenders who attack cyber systems were usually thought of as socially deficient and isolated young persons (Moustafa et al., 2018) who are spurred by a few motives, such as curiosity, illegal thrills, and peer approval. Irrespective of the talents possessed by some of these intruders, their financial prowess and resource to launch an innovative attack was not readily available. But today, advanced attacks and motives are evolving and being sponsored by multinational corporations and wealthy individuals. For example, the adoption of Advanced Persistent Threats (Yulianto et al., 2019) with some advanced approaches such as socially engineered zero-day exploits have become a current trend. This allows the intruders to bypass the activities of security tools embedded in the system and initiate a continuous presence in the compromised environment while controlling and collecting data over time. A Study by Rosenthal, (2022) reported that employees of different organisations in the world received about 14 malicious email annually. Also, ESET (2022) reported an increase of about 7.3% on email-related phishing attacks between May and August of 2021. According to IBM (2022) research published in 2021, phishing assaults increased by 2 percentage points between 2019 and 2020, in part, because of COVID-19 and supply chain uncertainties. According to Cisco, (2022) analysis on cyber threat patterns for 2021, 86% of firms had at least one employee who has clicked a phishing link. According to the company's

data, phishing is thought to be responsible for almost 90% of data breaches. Verizon, (2022) also reported that 96% of phishing attacks are delivered via email.

Furthermore, the evolution of cyber intruders into well networked and resourced groups strengthens their motivations into extending their cyber activities into politics and socio-economic fields, with the aim of gaining unfair advantage by attacking high-profile corporations and government agencies. The major categories of cyber-attack motives include information harvesting, disruption and financial gains. The risks in operating cyber systems are thus becoming more diverse and complex which increases the difficulties and challenges of developing countermeasures to mitigate the continuous attacks on cyber systems (Thakkar & Lohiya, 2020). Reports suggest that the forms of vulnerabilities which were identified ten years ago have tripled in number in current times, with suggestions that about 45 new vulnerabilities are discovered on a daily basis (Verma et al., 2020). Thus, in a bid to provide more comprehensive analyses of these threat and vulnerabilities, there is need for experts in cyber security to develop interventions that address securing the network, server and more importantly, the application. Some of the common cyber security tools such as encryptions, firewalls and anti-viruses are not sufficient to guarantee high level security due to the evolving nature of these vulnerabilities; thus the introduction of alternative warning systems for potential attacks are becoming more popular.

1.3 Objectives

This study intends to develop and implement open-source intelligence for detecting web-based phishing intrusions via emails which could be used as a personal resource or organizational resource for detecting possible phishing attacks. As a build up to the development and implementation process, the study would also

1. Develop a dataset of signatures in form of keywords, urls, email addresses and ip addresses identified as features of email phishing attacks

2. Develop a web based open-source intelligence for filtering and detecting web-based phishing intrusions via emails

3. Evaluate the functionality of the designed open intelligence source for filtering and detecting web-based phishing intrusions via emails

1.4 Significance of the Study

The development of an open intelligence source which has a functionality of an early warning system against phishing activities via email is always a welcome innovation. The researcher acknowledges that several forms of spam detection systems have been developed and deployed across the globe; however current practices encourage the use of multiple security protocols in guaranteeing the integrity of web-based system; therefore, there is a growing need of an abundance of choices and alternatives for cyber security applications in order to cater for regular changes and inclusion. Moreover, the evolving nature of the *modus operandi* of cyber attackers also calls for constant injection of new cyber security applications in the society. It is therefore believed that the outcomes of this study would be relevant for corporate bodies who wish to adopt the system into their existing security protocol. The outcomes of the study would also serve as a pivot for replication of other systems for the purpose of enhancing its limitations or upgrading its functionality.

CHAPTER TWO

LITERATURE REVIEW

This chapter contains a review of relevant literature to the study. The review spans across content that relates directly or indirectly to phishing activities at email levels, with the aim of gaining an in-depth understanding of how phishing activities work and how they can be mitigated. Outcomes of previous empirical studies in line with the study objectives are also reviewed.

2.1 Information Security

This study focuses on the issue of information security. Over the last years there has been a surge in interest and public awareness of concepts relating to information security. In understanding security, it is often practical to identify the attacker and activities at this level. The concept of kill chain as a cyber-defense strategy is highlighted as a borrowed concept from military activities (Hutchins, Cloppert, & Amin, 2011). The kill chain which was used as a military tactics was epitomized by the cumbersome acronym F2T2EA (Find, Fix, Track, Target, Engage, Assess). The tactics highlights the ability to identify and destroy time-sensitive targets using a series of steps (Tirpak, 2000). As used for cyber security, the kill chain involves the identification of high-severity cyber-attack, starting with reconnaissance of a target and ending with hackers remotely interacting with the target's computer. The different steps in the kill chain for cyber defense have their specific indicators and activities to implement a combination of preventive and remedial measures. The steps are illustrated in figure 2.1.

Features and activities related to phishing attacks can be identified in Step 3 (Delivery) of the cyber defense kill chain. At this delivery stage, the attacker lays down a trap for an identified victim. It is the action of the victim that determines if the chain will continue or not. In the event that the victim does not fall for the trap which has been set, then the next stages of the chain cannot materialize. Potential victims who are knowledgeable about information security

and potential pitfalls are more likely not to fall for the phishing attack. There are also other automated measures that may be applied to block the phishing email from getting to its targeted destination, such that the intended recipients are protected from such attacks (Kumaraguru, Sheng, Acquisti, Cranor, & Hong, 2010). Some concepts related to information security are highlighted in the following sections

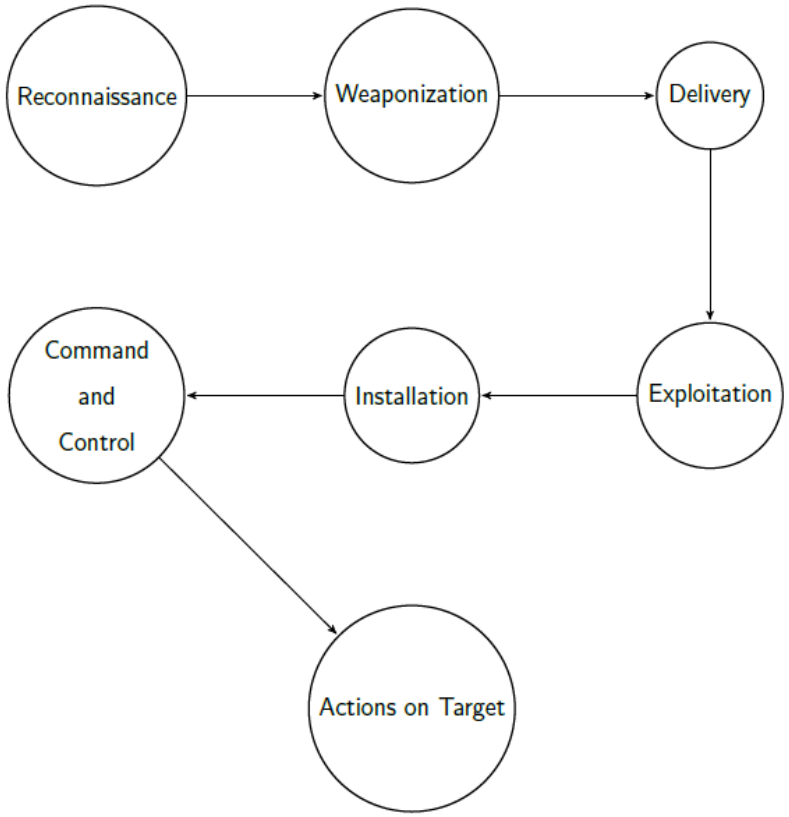

Figure 2.1: Cyber Kill Chain (Hutchins, Cloppert, & Amin, 2011)

2.1.1 Social Engineering

Social engineering involves the use of social interactions to obtain information needed to access or breach cyber securities and interfaces. In their book about social engineering, Mitnick and Simon (2002) adopted the use of case studies to analyze the various types of scam and cons involved in social engineering. One of the chapters in the book is dedicated to phishing attacks which involves the use of of automated tools to develop and send phishing emails as a form of phishing attack to compromise the integrity of seemingly secure sites. TrustedSec's (2013)

SpearPhisher is a tool which is often utilized by professional in the field of information security to perform penetration testing in line with their professional duties. Metasploit, is another popular crowd sourced exploit framework developed by Rapid7 (2012), which can be used ti facilitate the development of emails with phishing traits. The common goal of these tools is to create email templates with various spots to attach malicious content that can be used for phishing activities. This tool generates all the emails needed with minimal effort. The proliferation of these tools has lowered the skill requirement barrier that would have limited the proliferation of phishing attacks. Almost anyone with intentions to carry out phishing activities can be aided by these tools

2.1.2 Software Exploits

The software in computers may harbour errors, also known as "bugs" within the lexis of the computer world. Very complex software may harbor more bugs than simpler software (Mayer, 2012). Such bugs, when distributed to public domains, create vulnerabilities that threaten not only the specific software in question, but also the system on which the software runs, and even the whole network which contains the system. An exploit is specially designed software that exploits a single or a variety of vulnerabilities in order to control the systems in a computer in malicious ways (Koziol et al., 2004). These vulnerabilities expose the system to different levels of exploits such as information leaks or full manipulation of the system (Tsipenyuk, Chess, & McGraw, 2005). This vulnerability can lead to a crash of the software or make the system unresponsive to commands. They could also lead to a total denial of service if the entire system is compromised. However, one of the more serious vulnerabilities that exist (Microsoft Security TechCenter, 2012) is one that allows an attacker to create a kind of ‘weird machine state’ (Dullian, 2011) if it can be accessed by hackers from remote networks (Open Web Application Security Project, 2013).

2.1.3 Exploit Kit

Exploit kits can be described as a form of tool that phisher use to create phishing attacks (Segura, 2015). These exploit kits are designed to target the vulnerable spots within browsing webs or other web-based components. Some of the common exploit kits include Neutrino, Nuclear, Magnitude, and Angler (Chen & Li, 2015; Howard, 2012). Phishers set up web servers with public accessibility to internet domains. The phishing email, containing a malicious link, is then sent to these domains using the phishing kit, with the hope that potential victims will react to the ink by clicking on it; which then exposes the vulnerabilities of the system to exploited. This is an effective method of sending malicious PDF attachments (Stock, Livshits, & Zorn, 2015). These kits may include OPSEC features to protect a phisher's infrastructure investment and increase the number of successful attacks. For example, malicious links to exploit kit servers may contain unique identifiers. The server will attempt the exploit the first time it receives a connection request with this ID. However, more attempts to use former confirmed identities only serve harmless content and do not attempt to exploit. In this way, if the link is made available for scrutiny by a security expert for further analysis, it is less likely that the expert will be able to manipulate the server into demonstrating the malicious nature of the link.

2.2 Phishing

The concept of phishing as opined by Andress (2019) describes a form of social engineering where the personal information or details of an individual is compromised and accessed by an attacker via the use of malware in system environments which deceives the individual into giving up these details. This deception in often persuasive in nature such as directing the individual to click on malware links or entering personal information on cloned websites which look like the original websites. The use of cloned or fake websites is a common phishing

technique, such that unsuspecting victims can be directed to fake websites of existing banks or organizations. These websites would bear great similarities to the original websites, but upon careful inspection, differences and anomalies can be spotted to distinguish them from the original websites. Phishing attacks are often targeted at multiple victims in the hope that a few of them would fall for the deceptive element embedded in the attack. The success of phishing attacks can vary depending on the type of phishing used and the knowledge levels of information security which may guide the actions of potential victims. However, studies have shown that spear phishing often accrues higher success rates (Andress, 2019).

Spear phishing is a more targeted attack because it involves phishing attacks on a specific organization or individual (Hadnagy & Fincher, 2015). In spear phishing, the attacker may have some knowledge about weaknesses that can be exploited among the targets. Often times, the attacker spends time and resources in studying the targets via social engineering techniques and other research-based activities. Having obtained relevant information that provides a gateway for gaining entry into weaknesses of the target, the attack is deployed and aimed at deceiving the target into responding to the attack in ways that compromise the security of their personal information or details (Hadnagy & Fincher, 2015). For instance, sending a mail which seems to emanate from the victim's bank may prompt an immediate response from the victim. Such mails can be infused with contents that specifically identify the victim such as their names, date of birth and place of birth which enhances the victim's trust in responding to the mail. Spear phishing is one of the most difficult for potential victims to detect due to the personalization effects that are used in the attack. In the event an attacker can use sufficient truthful details about the target in a phishing email and give the target a reasonable reason to believe that the source is legitimate and credible, the chances of an attacker's success are greatly increased (Andress, 2019).

There are several types of phishing attacks where attackers can have a large number of targets or only a few clearly selected targets when performing a spear phishing attack. However, before attackers can actually launch phishing campaigns, they must first set up an infrastructure to host and serve their phishing payloads or websites. Figure 2.2 depicts the high-level stages of a typical phishing attack as described by Oest et al (2020) in their research paper, where they analyze the lifecycle and effectiveness of attacks. phishing attack.

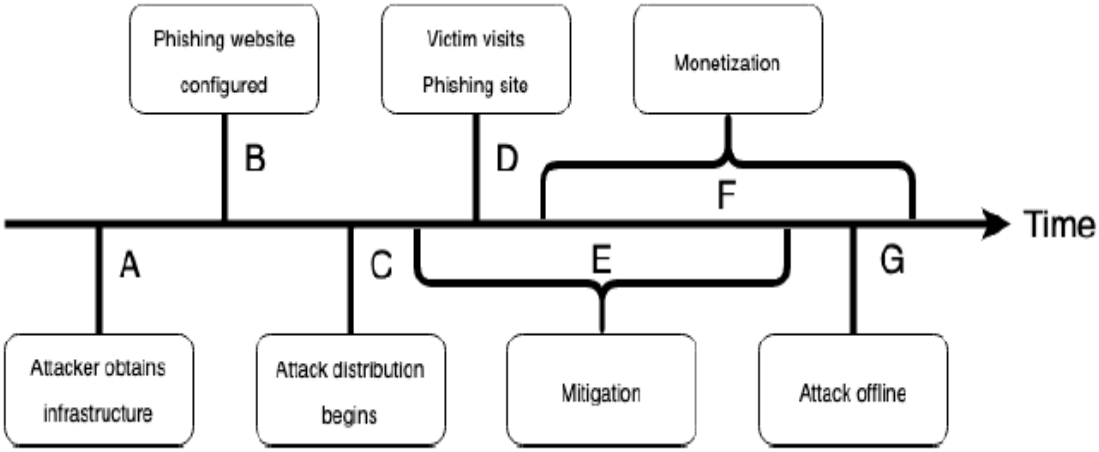

Figure 2.2: End-to-end life cycle of a phishing attack (Oest et al., 2020)

As depicted in Figure 3, the infrastructure is obtained by the researcher (A) then utilizes a phishing tool kit to set up the site targeted for phishing attacks which is often hosted on this infrastructure (B), which is then used to harvest credentials or deliver malware for download. After the website is up and running, the attackers begin to distribute it to their victims (C), usually via email, after which the victim starts visiting the website (D). Recognizing that a phishing campaign is underway, organizations can begin to contain attacks, depending on the organization's ability to target employees, such as through user reports (E). In the best scenario, if the user has not yet visited the phishing site, a mitigation action will be taken before (D) to prevent any future traffic from the victim. However, if unsuccessful, the attacker will be given a time frame to initiate an attack monetization (F) through a stolen data or network intrusion. Phishing sites can be deleted or deleted by the attacker himself (G). However, once an attacker

retrieves data that affects the organization, whether in form of credentials or easy accessibility to the network, monetization will continue even if the original infrastructure goes offline (Oest et al., 2020).

2.3 Email and Spam Filtering

Upon sending an email, it is directed into the message system and moves from server to server until it eventually gets to the mailbox of the recipient. There are several protocols that the email depends on; some of which include the Simple Mail Transfer Protocol (SMTP), the Post Office Protocol (POP3) and the Internet Message Access Protocol (IMAP). The SMTP is responsible for transmitting details while the POP3 and IMAP are responsible for the receipt of messages. The process of sending and receiving messages takes place between a Message Transfer Agent (MTA) and a Mail User Agent (MUA). The appropriate route for a mail is determined by the MTA upon receiving such mails from the MUA of a sender (Katalis et al., 2007). The MTA of the recipient then processes the delivery of the mail-to-Mail Delivery Agent (MDA) which are usually servers with POP/IMAP protocols. Reading and writing of mails by the user is made possible through the use of e-mail clients such as Mozilla, Microsoft etc.

There are specific places across the clients and servers in which spam filters can be incorporated. For instance, such spam filters are commonly deployed at the email server level of several organizations and internet service providers. Gateways and mail routers are some of the preferred locations to deploy spam filters. When deployed at the client level, proxies and plug-ins are suitable places to install such spam filters (Irwinn & Friedman, 2008). There are cases when spam filters may be deployed at both the client level and the email server level to beef up the security level. The evolution of spam filters has been highlighted by several authors in the literature, and the trends are tailored towards improvements of earlier spam filters (Almeida & Yamakami, 2012; Guzella & Caminhas, 2009; Carpinter & Hunt, 2006). The

dynamism in spam mails often increases the difficulties of spam filters to achieve 100% efficiency. The specific characteristics and signatures which are identified by spam filters can be modified to produce false positives or otherwise. Spammers therefore seek novel strategies to beat the spam filters such as the use of word obfuscation, image formats to represent text etc. This dynamism in the nature of spams fuels the need for continuous research and development of spam filters.

2.4 Machine Learning

The concept of machine learning is significant in the discourse of filtering techniques for detecting spam and phishing mails. This is because; it introduces the application of artificial intelligence in the process to boost the capabilities of the filtering systems to enhance their functionality through automated learning with minimal explicit programming (Aloaydin, 2020). Machine learning simply involves incorporating algorithms that enables the system to identify new patterns of related signatures from previous patterns of signatures. The machine process begins with a labeled data which is introduced into the system as a dataset for training. This training dataset may include textual contents, images etc. which can be used to recognize trends of signatures in incoming emails and make informed evaluations of the security status of such emails. Basically, the idea behind machine learning is to enable to system to recognize, adapt and record changes in signatures without the aid of human intervention. There are three forms of machine learning modules which can be used in filtering techniques (See Figure 4).

In recent pasts, the attention on email communication has been geared towards enhancing the process and user experience. This involves ensuring that email communication systems are devoid of security risks, and the presence of such insecure elements can be identified and mitigated. The use of spam filtering techniques is one of the ways to secure the email communication process. The literature has been flooded with various forms of machine

learning perspectives used to enhance the spam filtering techniques in identifying and processing emails with high security risks; however, gaps in literature have also been identified which necessitates more scholarly interest in the machine learning and filtering techniques (Sanz, et al 2008; Pitchaimani, 2020). Some of the more common approaches that have been adopted in machine learning studies include K-Nearest Neighbour (KNN) where K is the unknown variable which has been identified, support vector machine (SVM) algorithm, Naïve Bayes and Random Forest.

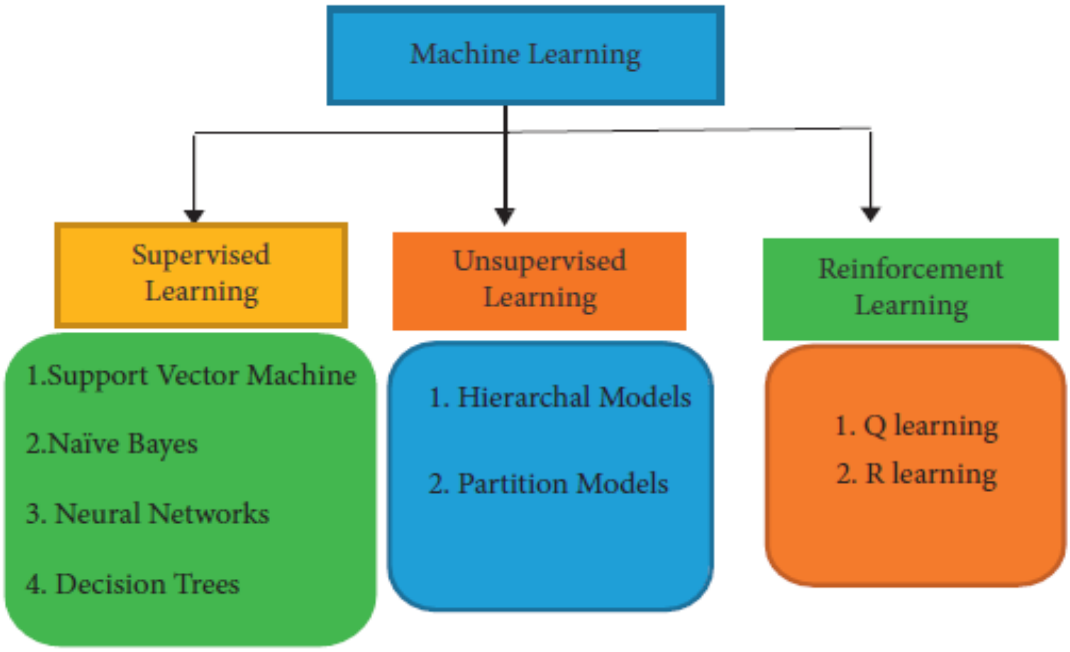

Figure2.3: Types of machine learning

The supervised machine learning algorithms include models that apply the labeling of data as the basis of training (Bhuiyan et al., 2018). In this case, data are labeled and introduced into the system along with algorithms that enable the system utilize the labeled data to form other related signatures which can be identified in the future. This suggests that the use of this model is initiated with an existing dataset for training which can be used to extrapolate more patterns

from the existing dataset as a predictive tool for detecting new signature patterns in the future (Singh et al., 2016). The robustness of the training dataset and algorithm used would determine the level of efficiency in the system's ability to accurately identify emails with high security risks. Furthermore, an error modification is also embedded in the process whereby algorithms to enable the system compare its output to specified expectations and make adjustments accordingly. The merits of supervised learning models have been recorded in the literature across fields of advert popularity, classification of spam, face recognition and classification of objects.

In contrast to the supervised learning model, the unsupervised machine learning are used in the absence of an initial training dataset (Diale et al. 2019). The algorithms built into unsupervised machine learning models are required to explore ways in which the system can identify hidden signatures through inference of a feature within an unlabeled dataset (Ghahramani, 2003). In this case, the machine does not have a basis upon which its output can be compared; however, it relies on its capability to make inferences about the security status of an email. As these outputs are produced and validated by the user, the system is able to generate and store clusters of such data in its repository for easier and faster identification in the future. This process enables the machine to adopt a system where it trains itself based on user validation of outputs. This machine earning model has been applied in various fields to solve problems associated with human spontaneity and novel capacities in which previous data may not exist.

The use of reinforcement learning describes a model learning model in which decisions are made in sequences. The goal of learning is achieved in a complex and uncertain environment that is devoid of training datasets its environment. It takes suitable actions to make or get the maximum reward in a given situation (Lison, 2015). The system therefore adopts a game-like situation in which trials and error decisions are made as a means to identify the outcome with the best possible outcome. The trial and error outcomes are based on a reward or penalty

outcome, such that rewards are attached to outcomes that are significant. Thus, the systems work on its ability to maximize the sequence of rewards it gets from its trial and error process. The system is incorporated with algorithms that specify the reward and penalty policy (i.e., the rules guiding which outcome is significant), but no training dataset is provided. As such, the system has to rely on random trials to produce several outcomes on its own in a bid to identify which ones are significant, and which ones are not significant based on the reward policy. This continuous process reinforces the system's ability to determine sequences that are significant over time as it learns from the outcomes of its trial and error process (Smadi, et al. 2018).

2.5 Review of Related Studies

Ludl et al. (2007) conducted a study in which the potency of phishing was examined and relevant mitigating techniques were sought. They identified two common systems for mitigating phishing attacks and sought to test their efficiency. The systems were integrated into two major web browsers (Firefox and Internet Explorer) and earmarked a 3-week testing period for the anti-phishing solutions by running about 10,000 URLs which had been blacklisted as being fake by Google and Microsoft. Furthermore, they explored how the use of page attributes such as links, suspicious urls, forms, and input fields to identify phishing activities were captured by the anti-phishing solutions by analyzing the phishing pages identified by the solutions. The importance of the attributes identified as phishing markers in each of the pages were discussed in relation to creating awareness of users about security information that can aid them to make informed decisions about their actions (and inactions) on websites and pages to avoid being victims of phishing scams.

Oest et al. (2020) were interested in examining the end-to-end lifecycle of phishing attacks done on a large scale. This was achieved by isolating and identifying gaps in their measures and evaluations. The authors developed a specific framework provided a platform in which

victim tracking to suspected pages of phishing could be passively measured while ensuring that thousands of accounts were proactively protected in the process. The test period spanned for over a year, during which about 4.8 million individuals visiting phishing pages were monitored and recorded. The activities of these individuals on the phishing pages; from their first connection to the network, through the distribution of email, visitor tracking routines, detection in ecosystem and engagement in account; were analyzed as a basis for understanding user behavior in internet web environments. Results suggested that it took an average of about 21 hours for the phishing cycle to be completed from the inception of the victim's actions. Among the individuals, about 7.42% of them were identified as potential victims because they actually provided their personal details and credentials which would lead to fraudulent transactions. Furthermore, about 89.13% of the individuals benefited from anti-phishing campaigns as they applied their knowledge of security information to avoid being scammed. The results highlighted the success of anti-phishing campaigns while identifying patterns of user responses to sophisticated phishing attacks.

Siaddati et al (2017) conducted a study in which the reaction of about 19 thousand employees from an organization to real-world phishing campaigns was evaluated by deploying over 115,000 phishing emails as a test among them. The objective of the study was to examine the impact of such anti-phishing campaigns by providing insight into the individuals' understanding of the campaigns, while deducing vague or incomprehensive aspects of the campaigns. The authors were also interested in using the information obtained to improve on the shortcomings and flaws in the phishing campaigns. Based on their findings, it was observed that the targets of most of the real-world anti-phishing campaigns were on more persuasive phishing emails, with fewer efforts directed at less persuasive ones. The notion that more persuasive phishing emails are more dangerous than less persuasive ones is an illusion as both attacks culminate in defrauding the victims.

Alghamdi (2017) carried out a study in which its objective was to ascertain how effective education and training about phishing could be used in identifying and mitigating threats of phishing attacks. The study was based on evaluating the ability of users to know phishing attributes which were embedded in email, SMS messages, phone call or social networks. Against the backdrop of the study objectives, a structured questionnaire was used to obtain relevant data from prospective participants. The questionnaire was designed for use as a pre and posttest tool in an experimental process. A treatment group and control group were used for the study. The treatment applied in the experiment was a phishing education and training content provided in a classroom setting. Both groups were given the questionnaire at pretest stages; the treatment group was given the treatment while the control group was not given the treatment. Both groups were given the questionnaire at the posttest stage. Results obtained showed that there was no significant difference in posttest outcomes between the control and treatment groups, which suggests that the phishing education/training content was not effective. Factors that contributed to the insignificant difference were addressed and noted in order to make the treatment more effective.

Walrave et al, (2018) examined how routine activities on the internet by users make them victims of motivated offenders through phishing activities. This was achieved via the use of an integrative lifestyle exposure model which identified routine activities as ‘risky’ or ‘not risky’ for phishing attacks. The research methods included data collection from a representative sample of 723 participants across a variety of contextual and behavioral variables that were deemed relevant and consistent with victim susceptibility to online scams. Results from the analysis showed that phishing susceptibility was significantly associated with online shopping behaviour, impulsivity and digital copying behaviour. The results suggest that the kind of activities indulged by online shoppers and internet users may make them susceptible to

phishing. Such risky behaviour should therefore be incorporated in anti-phishing training and awareness program for online users.

CHAPTER THREE

PILOT STUDY AND FORMULATION OF HYPOTHESES

3.0 Introduction

Phishing has been identified as a current challenge plaguing the society. One of the most common methods of phishing is through e-mailing (Verizon, 2022). This is because the use of emails has become a significant part of daily activities at both personal and organizational levels. It is therefore practically impossible to ignore all forms of e-mail communication and more tasking in determining which of these emails are legitimate or potential scams (Verizon, 2022). The objective of this study is to build a software which can be used to conduct forensic analyses on emails from a selected hub, with the aim of detecting mails which have traits of phishing activities.

3.1 Preliminary System Investigation

The preliminary system investigation phase is concerned with investigating and evaluating the efficiency of other existing software with similar functions (Cahyana, 2018). This information was used to understand the shortcomings of other existing software with the aim of improving on any identified limitations and improve on the requirements in the development of alternative software. Information during the system investigation was obtained via interview methods and a review of literature. The interview method entailed conducting interview sessions with users of various spam detecting software for emails in corporate settings. A structured interview guide was developed and used for the interview sessions in order establish an objective and homogenous process. The major limitation of existing spam detecting software being relied upon as identified from the participants' responses included their inability to identify and categorize outgoing emails as spam and their limited machine learning capabilities. Further review of literature on related studies provided the researcher with ample information on the

various methodologies and models that have been adopted in developing similar software and applications (Cahyana, 2018).

For instance, Fahad (2015) developed a similar using Naïve Bayesian classifier for identifying spam and legitimate emails. The system was integrated with 149 signatures as spam identifiers which included features which have been commonly associated with previous spam messages. It was recommended that replicated studies should increase the signature base to accommodate the rapid evolution of novel signatures. Kumar and Sonowal (2020) conducted a survey which focused on the process of knowledge discovery for systems that detect spam emails. The machine learning systems which they highlighted included multilayer perceptron neural network support vector machine, Naïve Bayes and the non-machine learning systems highlighted included Blacklist and Whitelist, Mail Header Checking systems, and Signature systems. Saleh et al. (2019) adopted survey methods to highlight intelligent spam email detection. The authors placed higher preference on the use of frameworks with multi-algorithms than single algorithms. In their conclusion, it was opined that majority of the empirical studies on the identification of phishing and spam emails depended on clustering systems and word-based categorization system. Blanzieri and Bryl (2008) highlighted some of the ethical issues and considerations in spam email analyses. In their study, an in-depth review of learning-based spam filtering was provided across different domains. The review also included an exploration of various classification techniques for spam and phishing emails. They justified the potency of Naïve Bayes classifier for machine learning algorithms due to its precision, speed and simplicity

3.2 The Alternative System

Based on the information obtained from the system investigation phase, an alternative system was proposed with the aim of catering for the limitations observed in existing spam detection software. in line with the general objective of the study, the alternative system to be designed

and developed will be called ‘Can-I-Trust-It’. This name is derived from the uncertainties bordering on phishing scams that users experience when receiving and responding to emails. The alternative software is therefore a go-to solution for such users with the aim of increasing their level of trust in the category of mails received. The improvements to be made in the alternative system included improved user friendliness, faster processing rates, detection of both incoming and outgoing spam mails, and capability update dataset signatures. The software would be designed to support specific types of neural networks by providing a number of different coding languages for each type. It will also be specifically designed to be compatible with the semantic-web and graph databases.

These would be achieved by adopting a spam filtering model in which contents and sources of emails would be analyzed and filtered according to a signature-based repository. Content filtering describes the identification of email contents in form of text, images or language usage within an email in order to detect traits of phishing attacks or spam. Based on the common objective of many phishing emails which is focused on persuading the user to take certain actions online that lead to a compromise, some words, word usage and images are often used as contents. Words that may provoke human emotions of greed, fear, desire such as *‘a minute’s delay will cost you this limited offer….’.* The use of such words in multiples of in specific sequences may be the trigger for flagging and filtering such content. The use of inappropriate language may also be a marker for spam or phishing. For instance, sexually provocative languages may be used to stimulate the erotic desires of the user as a persuasive element to initiate action. For instance, sexual stimulating language like ‘*meet local girls in your area for free’* may be enough to motivate users into clicking certain links. The use of languages which are alien to a particular space may also trigger the content filtering process. For instance, a legitimate email sent in Chinese language to an American based email address may be flagged as spam and filtered accordingly. This suggests that content filtering may also yield false

positives based on the content signatures which it has been programmed to associate with spam or phishing mails.

.

Aside the filtering of email content via text, images or language, some spam filters are designed to identify email addresses, internet protocol addresses, online domains, URLs etc. which have been identified and blacklisted as spam or phishing sources. The blacklist containing such markers is continually updated as more successful and unsuccessful phishing or spam activities are discovered. Therefore, new domain being operated by phishers and spammers due to the blacklisting of their previous domains are still susceptible to being blacklisted in the nearest future when the blacklist is updated. Several organizations have adopted the use of blacklist filtering to ward off their competition from poaching their employees through emails. This is achieved by including a URL or domain that is associated with the addresses of their completion in their blacklist. Aside the use of blacklist filters; spam and phishing activities can also be detected and filtered based on the header or mail subject. For instance, the header or mail subject could suggest that the email is a copy of a generalized email sent to a targeted audience or recipient group. Such a mail would be flagged as spam or phishing attacks and subsequently filtered accordingly in order to stop its being accessed by the intended recipient.

3.3 System Development Tools

System development describes the activities involved in designing, testing and implementing a system or software application. In developing a system, specific tools are required to actualize the objectives. The system development tools may therefore include the integrated development environment in which the system would be built, the programming languages to be used, the forms of toolkits to be incorporated in the system, the program codes etc.

3.3.1 Integrated Development Environment (IDE)

An integrated development environment is a useful environment for software development. It is equipped with tools which can assist software developers. This makes the work of the software developers easier, and it could save time as well. Integrated development environments are designed to maximize programmer productivity by providing tight-knit components with similar user interfaces. IDEs present a single program in which all development is done. This program typically provides many features for authoring, modifying, compiling, deploying and debugging software. For the project, the IDE of choice is Visual Studio Code (VSC). The Visual Studio Code is a free, open-source, cross-platform integrated development environment. It integrates the latest versions of the IntelliSense and debugging capabilities of Microsoft Visual Studio. It is also used to develop web applications using HTML, CSS and JavaScript. After this, it uses the Beautiful Soup library together with Scrapy to examine and analyze contents of emails with phishing traits. The library is written in Python making it possible to use it for writing a program. The Scrapy library extracts valuable URLs that are suspicious-looking like keyword domains, text patterns, and links which they consider as indicators of phishing.

3.3.2 Programming Languages

PHP scripting language will be used for the server domain while MySQL will be used for the database system. PHP is a scripting language, and can be used to create web pages written in HTML. PHP runs on the server (the system from which the page comes), and is a full-fledged programming language (Prokofyeva and Boltunova, 2017). MySQL is an Oracle-backed open-source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including LINUX, UNIX and WINDOWS. Although it can be used in a wide range of applications, MySQL is most often

associated with web applications and online publishing. This is the database used for the application.

3.3.3 Filtering Toolkit

A filtering toolkit will be embedded in the system to identify signature based anomalies in the incoming and outgoing emails. The filtering toolkit would run on algorithms that have been built to filter contents and sources of emails based on the signature repository. The filtering toolkit would also enhance the capability of the signature repository to be updated and/modified based on the context of usage.

3.4 Feasibility Study

A feasibility study would be carried out to evaluate the potentials for success or failure of the project as well as to identify the risks involved in executing the project. A preliminary study will be carried out to determine whether the proposed application is viable economically, technically and operationally.

3.4.1 Economic Feasibility

An economic feasibility analysis takes into consideration the process of evaluating the worth and cost of venturing into an activity (McConnell, 1998). It adopts a cost-benefit analysis which takes into account the financial and operational costs of developing and operating a planned venture. The economic feasibility analysis therefore helps in understanding the potential risks inherent and making adjustments to reduce such risks to the barest minimum. A wide array of economic cost factors is typically analyzed to assess the cost effectiveness of the venture. Table below 3.5 shows an illustration of the types of cost implication of hardware and software requirements of the proposed alternative system.

Table 3.1: Cost Estimate of Proposed Application

PROJECT DESCRIPTION	COST ESTIMATE
Hardware	
Smartphone	
Laptop (High level visual graphic card)	
Operational	
Maintenance and Version Upgrade	
FAQ Maintenance fee	
Software Development	
Backend development	
User interface design	
Software Development Kit	
Playstore	
Graphic design	
Mobile app development	
Total	

3.4.2 Technical Feasibility

A technical feasibility analysis would also be conducted. Technical feasibility assesses the likelihood of the system being developed with the techniques, knowledge and skills that are at the disposal of the developer (Shanguo, 2016). This assessment would outline a flow chart of the different stages of the system development, as well as the technical resource and competence required and available for each stage. The availability of alternative techniques for each of the stages is also assessed. By so doing, the developer is assured that there would be no technical hitches during the development process. Technical knowledge to be applied may include programming language, scripting language, database management system etc.

3.4.3 Operational Feasibility

Operational feasibility assesses the extent to which the proposed system is able to solve the existing problems, how user friendly the proposed system is and how acceptable the proposed system is by the users. Operational feasibility analysis ensures that upgrades to limitations of previous systems are satisfactorily met (Shanguo, 2016). Therefore, the user experience must be taken into consideration during the operational feasibility analysis by understanding current trends and structures that have high ratings of societal acceptability and use. Operational feasibility also considers the level of support that the proposed system would receive from target organizations. It is therefore important to know how committed management of target organizations are towards the adoption of the proposed system.

3.5 System Description and Functionality

The figure below describes the hypothesized system description and functionality

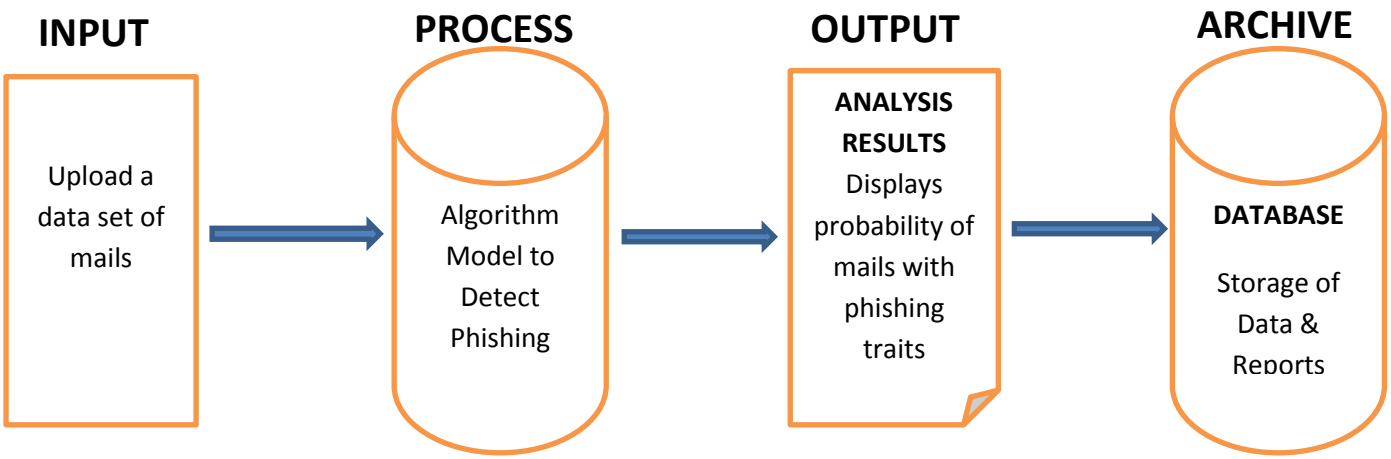

Figure 3.1: Flow Chart showing system description and functionality

Based on the flowchart presented in Figure 3.1, the artifact would have four functional phases. The input phase would have features that enable the attachment and upload of a file or dataset of varying formats such as CSV. The dataset would be a compilation of mails showing the different attributes of each mail in different fields. For instance, attributes to be displayed may

include sender address, receiver address, content of mail, time of mail sent, origin/source of mail etc. This phase will be built using html + CSS tools.

The process phase would have the functionality of analyzing the uploaded dataset based on the spam filtering algorithm which is built into. This phase would be built using PHP or Python tools.

The output phase would have fields that display the results of the analyses on the data set. Presentation of results would be in descriptive formats via texts, percentages or charts which show the probability of each mail within the dataset having features of phishing. This phase would be built using html, CSS and JQuery tools.

The final phase is an archive section for storing data and reports for future purposes or further analysis. This phase would be built using the MySQL tool.

CHAPTER FOUR

CAN I TRUST IT: OPEN-SOURCE INTELLIGENCE

Can i trust it is a spam filter designed to identify emails which hackers use to send unwanted or dangerous content to potential victims of phishing attacks. It is built to use different filtering methods to identify the content of emails of their senders and then flag the email as spam. It also examines an email for explicit content that could contain malicious links. It is built with different spam filtering algorithms using blacklists of keywords, headers, languages, domains etc. which enables a scalable knowledge base system that can be fully customized according to user preferences. The system is also suitable for analyzing offline bulk email correspondence checks and forensic investigations within an organization.

4.1 Description and Functionality

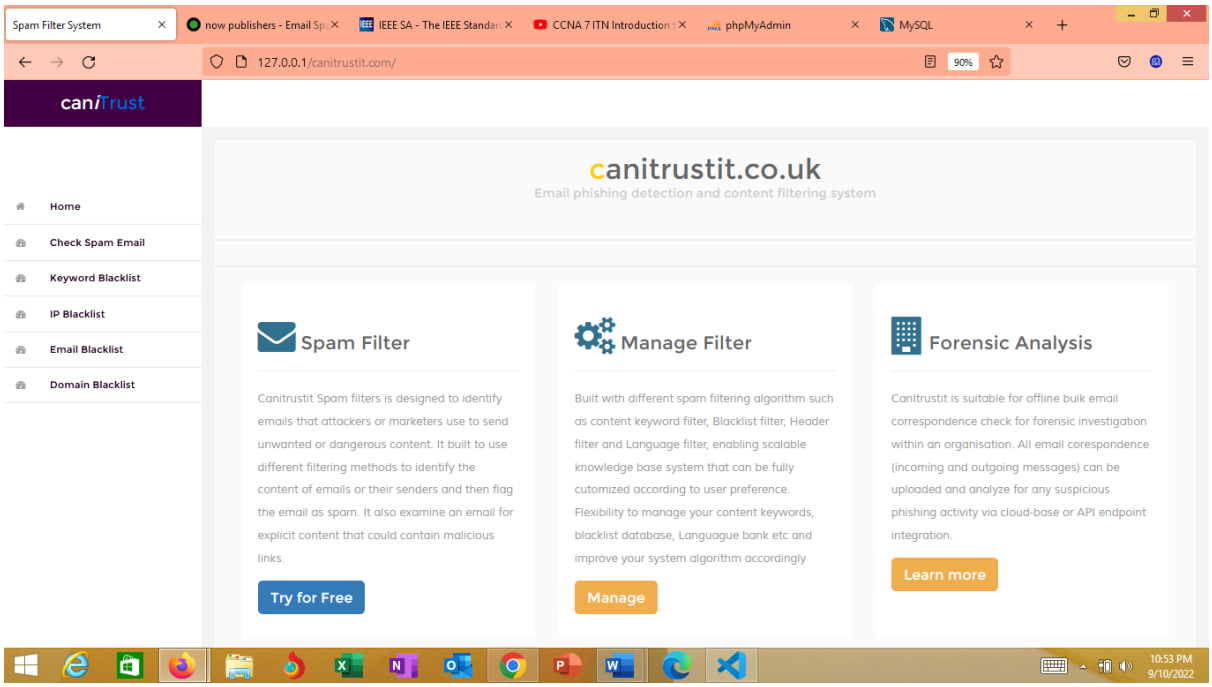

Figure 4.1: The Home Page

The figure above is a screenshot of the home page. The home page is designed with relative simplicity in aesthetic features and user friendliness. The web address bar is displayed at the top of the page while other tabs displayed by the left side include home tab, spam filter tab, keyword blacklist tab, IP blacklist tab, Email blacklist tab, Domain blacklist tab. The main

features displayed in the center of the homepage are links to spam filter, manage filter and forensic analysis. The spam filter link directs the user to the interphase in which emails can be analyzed for phishing activities (see Figure 4.2). Within this page, there are fields into which the user inserts the email content, sender's address, IP address, and email subject. Having made all these inputs, the user then clicks the 'analyze it' button below. The email is then analyzed for phishing feature based on the built-in algorithms. The results of the analysis are then displayed on the right-hand side of the screen. The results are flagged as 'high risk spam', '. low risk spam', 'High Risk Spam, not harmful', 'Not harmful' and 'not spam' depending on the categorization that is detected in the analysis

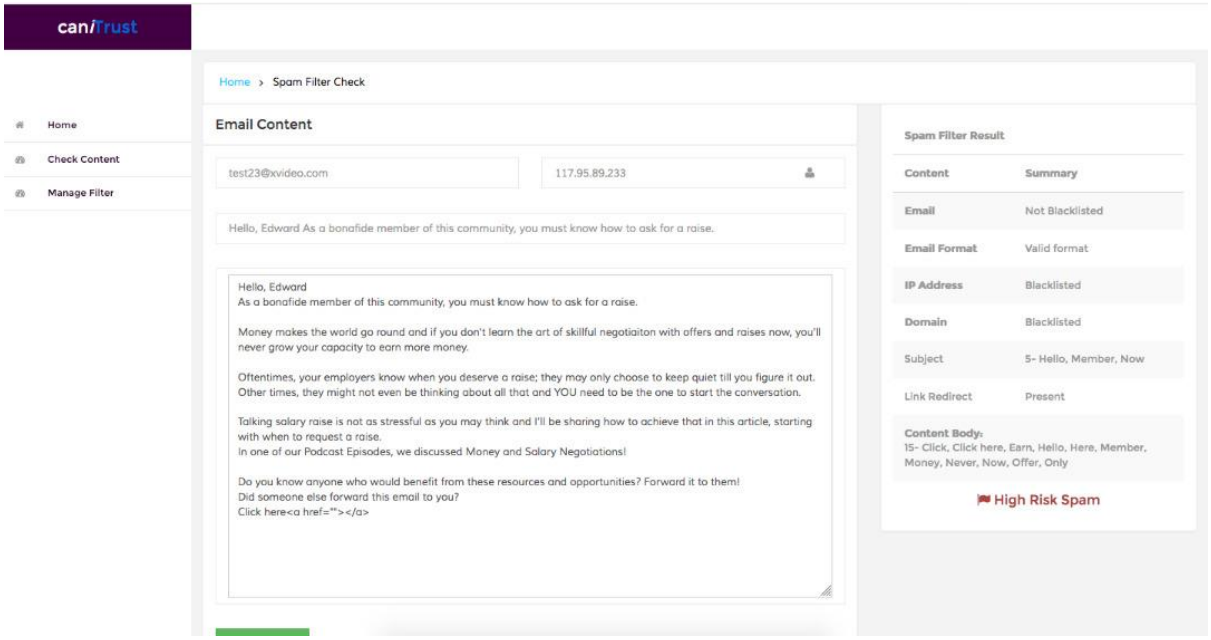

Figure 4.2: Spam Filter Check

The 'Manage Filter' Tab on the home screen provides a link to operations for managing the blacklist database for keywords, emails, domains and IP addresses (see Figure 4.3). There are two major fields in this page. The field by the left is used for entering blacklisted signatures (keywords, emails, domains and IP addresses) into the blacklist database while the field towards the right shows that list of blacklisted signatures available in the database. The user is

therefore able to modify and update the contents of the blacklist database based on user preferences and usage contexts. The Forensic Analysis tab on the home page provides a link to an interphase where bulk email files or folders can be analyzed. The user is therefore able to upload bulk emails saved in CSV or excel format. The results of each of the emails are then displayed and flagged as 'high risk spam', '. low risk spam', 'High Risk Spam, not harmful', 'Not harmful' and 'not spam'.

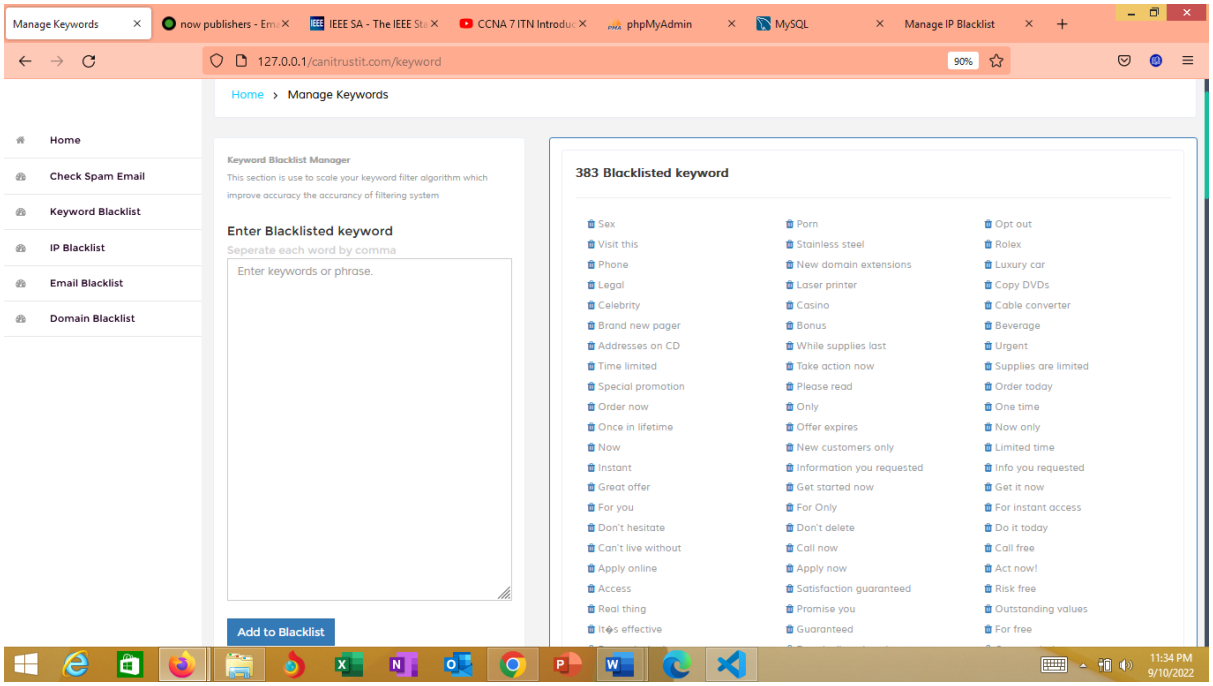

Figure 4.3: Managing the Blacklist Database

4.2 Source Codes and Functionality

4.2.1 The App Library

The App Library handles basic software operation and database connection and management using the following source code

```
<?php
class app {
                function __construct(){
                if(!$_SESSION){session_start();}
                $con=$this->dbconnect();
```

```
	}
	//connection to database
	function dbconnect(){
		( file_exists("Connections/pconnect.php") ) ? include("Connections/pconnect.php") :
include("../Connections/pconnect.php");
		$this->db = $mysqli;
		return $this->db;
	}

//restrict access to pages where user did not login
function check_for_valid_login(){
	if(!isset($_SESSION['loginUserId'])){
	session_destroy();
	@header('location: ./application-login');
	exit;
	}
}

//logout
function logout(){
	unset($_SESSION['application_no']);
	@session_destroy();
	header("location: ./?p=/dologout/");
}
//fetch site configuration setting
function configurationSettings(){
	$data = $this->rowset("SELECT * FROM configuration LIMIT 1");
	return $data;
}

function mvc($string)
{
	$string = strtolower($string);
	$string=str_replace(' ','-',$string);
	$string=str_replace(' ','',$string);
	$string=str_replace("'",'-',$string);
	$string=str_replace("(",'',$string);
	$string=str_replace(")",'',$string);
	$string=str_replace("/",'',$string);
	$string=str_replace('"','',$string);
	return $string ;
}
//TRANSACTION SQL
function begin(){
$con = $this->dbconnect();
```

```
 mysqli_query($con, "BEGIN");
 }
function commit(){
$con = $this->dbconnect();
mysqli_query($con,"COMMIT");
}
function rollback(){
$con = $this->dbconnect();
mysqli_query($con, "ROLLBACK");
}

function query($query){
	$con = $this->dbconnect();
	$msg = array();
	if(!empty($query)){
		$this->begin();
		$execute=mysqli_query($con, $query)or die("QUERY ERROR:[ ".mysqli_error($con)."
]");
		if($execute){
			$this->commit();
			$msg['affected_row']=mysqli_affected_rows($con);
			$msg['status']='success' /* success */ ;
		}
		else{
				$this->rollback();
				$msg['status']='failed' /* fail */; }
	}

	return $msg;
}
//run multiple query in one connetion
public function multi_query($query){
	$con = $this->dbconnect();
	$msg = array();
	if(!empty($query)){
		$this->begin();
		$execute=mysqli_multi_query($con, $query)or die("MULTI QUERY ERROR: [
".mysqli_error($con)." ]");
		if($execute){
			$this->commit();
			$msg['affected_row']=mysqli_affected_rows($con);
			$msg['status']='success' /* success */ ;
		}
		else{
				$this->rollback();
```

```
                                        $msg['status']='failed' /* fail */; }
        }

        return $msg;
}

 //return data set
function dataset($sql){
        $con = $this->dbconnect();
                $this->sql=$sql;
                if(!empty($this->sql)){
                        $query=mysqli_query($con, $this->sql)or die('DATASET:
'.mysqli_error($con));
                                if($query){
                                while($row=mysqli_fetch_array($query)){$r[]=$row;}
                                }

                }
                        return $r;
 }

 //return row set
function rowset($sql){
  $con = $this->dbconnect();
                $this->sql=$sql;
                if(!empty($this->sql)){
                $query=mysqli_query($con, $this->sql)or die('ROWSET ERROR: [
'.mysqli_error($con).']');
                $row=mysqli_fetch_array($query);
                }
                        return $row;
 }
//count row
function rowcount($statement){
 $con = $this->dbconnect();
                        $this->statement=$statement;
                if(!empty($this->statement)){
                        $this->execute=mysqli_query($con,$this->statement)or die('ROW COUNT
ERROR: [ '.$statement. mysqli_error($con).' ]');
                        return mysqli_num_rows($this->execute);
                }
 }
//count row
function rowcount2($statement){
 $con = $this->dbconnect();
```

```
                        $this->statement=$statement;
                if(!empty($this->statement)){
                        $this->execute=mysqli_query($con,$this->statement)or die('ROW COUNT2
ERROR: [ '.$statement. mysqli_error($con).' ]');
                        $rows = mysqli_fetch_array($this->execute);
                        return $rows[0];
                }
 }
//check a variable value is not enpty
function isEmpty($variable, $statusMsg){
                if(empty($variable)){$this->msg=$statusMsg;}
                return $this->msg;
 }
 //check a variable value numeric
function isNotNumeric($variable, $statusMsg){
                if(!is_numeric($variable)){$this->msg=$statusMsg;}
                return $this->msg;
 }
 //get file extention
function fileExtention($file_name){
        $ext = explode('.'.$file_name);#get the dot(.) position from the filename which is return in
number
        $ext = '.'.end($ext); # use the dot position to get letter after the dot
        return $ext;
        }
function file_mime_type($fileName){
                switch(file_extention($fileName)){
                        case ".jpg"; $mm_type = "image/jpg"; break;
                        case ".pjpeg"; $mm_type = "image/pjpeg"; break;
                        case ".png"; $mm_type = "image/png"; break;
                        case ".gif"; $mm_type = "image/gif"; break;
                        case ".xls"; $mm_type = "image/jpg"; break;
                        default; $mm_type="";
                        }
                        return $mm_type;
}
        //clean html input shorter method name
function clean($theValue, $theType, $theDefinedValue = "", $theNotDefinedValue = "")
{
        $con = $this->dbconnect();
  $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
  $theValue = function_exists("mysqli_real_escape_string") ? mysqli_real_escape_string($con,
$theValue) : mysqli_escape_string($con,$theValue);

  switch ($theType) {
```

```
   case "text":
     $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
     break;
   case "long":
   case "int":
     $theValue = ($theValue != "") ? intval($theValue) : "NULL";
     break;
   case "double":
     $theValue = ($theValue != "") ? "'" . doubleval($theValue) . "'" : "NULL";
     break;
   case "date":
     $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
     break;
   case "defined":
     $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
     break;
  }
  return $theValue;
}
}#end class
 $objectApp = new app();
 $config = $objectApp->configurationSettings(); //get global data for application setting
 ?>
```

4.2.2 This Spam filter Library

This spam filter library contains the filtering algorithms which categorize the emails being analyzed into ‘high risk spam’, ‘moderate risk spam’ and ‘low risk spam’ based on the results obtained from the identified signatures from the blacklist database. The following source code was used.

```
<?php
class spamfilter extends app {

#propety if the spamfilter class
public $email;
public $domain;
public $ip;
public $email_subject;
public $email_body;

//check if the email if blacklisted and return score of 15
function checkEmailBlacklist(){
        $email = $this->clean('%'.$this->email.'%',"text");
        $score = 0;
```

```
		$stmt ="SELECT count(email) FROM blacklist_email WHERE email LIKE". $email;
		$count = $this->rowcount2($stmt);
		if($count>0){ $score=15;}
		return $score;
}
//check if the IP is blacklisted and return score of 15
function checkIpBlacklist(){
		$ip= $this->clean($this->ip,"text");
		$score = 0;
		$count = $this->rowcount2("SELECT count(ip) FROM blacklist_ip WHERE ip=".$ip);
		if($count>0){ $score=15;}
		return $score;
}
//check if the email domain is blacklisted and return score of 15
function checkDomainBlacklist(){
		$email = explode('@', $this->email);
		$domain = array_pop($email);
		$domain1 = 'http://www.' . $domain;
		$domain2 = 'www.' . $domain;
		$score = 0;
		$count = $this->rowcount2("SELECT count(*) FROM blacklist_domain WHERE (domain LIKE
'%$domain%') OR (domain LIKE '%$domain%') OR (domain LIKE '%$domain%')");
		if($count>0){ $score=15;}
		return $score;
}
//check if email address is a valid email format give score of 10 for invalid
function checkEmailFormat(){
		$email = trim($this->email); #remove white spaces
		$score = 0; #initial score is zero for valid format
		if(!filter_var($email, FILTER_VALIDATE_EMAIL)){
			$score=10;
		}
		return $score;
}
/*
 *	check for script injection or URL redirection in body of email message
 *	and give a score of 20 if redirection or link is present
*/
function check_email_body_for_script_injection(){
		$email_body = $this->email_body;
		$score = 0;
		preg_match('/(http|ftp|mailto|www|https|a href|href)/', $email_body, $matches);
//var_dump($matches);
		$count_result_found= count($matches);
		if($count_result_found >0){ $score=20;}
		return $score;
}
//check email content for words matching the spamword in the database keyword blacklist;
//and in subject of email give a score of 5 for email subject and a score of 15 for email message
body;
function checkSpamWords($content, $content_type){
```

```
		$score = 0;
		$result=array();
		$stmt = "SELECT spamword FROM blacklist_word ORDER BY spamword ASC";
		$data = $this->dataset($stmt);
		for($i=0;$i<count($data);$i++){
			$spamWord=$data[$i]['spamword'] ;
			if (stripos($content, $spamWord) !== false){
			$matches[]=$spamWord;
			}
		}
		if(count($matches) > 0 )
		{
			$score = ($content_type=='body') ? 15 : 5 ;
			$result['spamword']= implode(', ', $matches);
		}
		$result['score'] = $score;
		$result['wordcount'] = count($matches);

		return $result;
}
/*
 * check email body and subject for upper case letter
*/
function isUpperCaseLetter(){
}
//summarise result of spamfilter check for harm or spam
function spamFilterResult(){
		$urlResult=$this->check_email_body_for_script_injection();
		$domainResult=$this->checkDomainBlacklist();
   $emailFormat=$this->checkEmailFormat();
		$emailBlacklistResult= $this->checkEmailBlacklist();
		$ipResult=$this->checkIpBlacklist();
		//arrays are return from this two method
		$getSpamWordInBody = $this->checkSpamWords($this->email_body, 'body');
		$getSpamWordInSubject = $this->checkSpamWords($this->email_subject, 'subject');
		$spamWordInBody = $getSpamWordInBody['score'];
		$spamWordInSubject = $getSpamWordInSubject['score'];
		$result = $urlResult+$domainResult+$emailFormat+$emailBlacklistResult + $ipResult +
$spamWordInBody+$spamWordInSubject;
		$result = ($result > 0)? (100/$result) : 0;
		$result = number_format($result,2);
		return $result;
}
function conclusion($score){
			//$score=(float)$score;
		if($score >=0 && $score <=1.99){$lg='<h4 class="text-danger"><i class="fa fa-flag text-
danger"></i> High Risk Spam</h4>';}
		elseif($score >=2.0 && $score <=3.99){$lg='<h4 class="text-warning"><i class="fa fa-flag
text-warning"></i> Low Risk Spam</h4>';}
		elseif($score >=4.0 && $score<=5.99){$lg='<h4 class="text-success"><i class="fa fa-
check"></i> Spam, Not harmful</h4>';}
```

```
        elseif($score >=6.0 && $score <=7.99){$lg='<h4 class="text-success"><i class="fa fa-
check"></i> Not Harmful</h4>';}
        elseif($score >=8.0 && $score<=10){$lg='<h4 class="text-success"><i class="fa fa-
check"></i> Not Spam</h4>'; }
        else{$lg="Unknown Content";}
        return $lg;
}
} # end my class curly brases
```

4.2.3 The MVC LIBRARY

The MVC library is a self-developed mini library/framework for handling page navigation via hyperlinks. This library is dependent on the .htaccess file for URI parameter handling. The following source code was developed

```
<?php
class mvc extends app{
        private $url;
        function __construct()
        {
                $this->url = isset($_SERVER['PATH_INFO']) ? explode('/',
ltrim($_SERVER['PATH_INFO'],'/')) : '/';
        }
        function controller(){
    //The first element should be a controller
    $requestedpage = $this->url[0];
    // If a second part is added in the URI,
    // it should be a method
    $requestedAction = isset($url[1])? $url[1] :'';
        return  $requestedpage ;
        }
        function controllerParameter() {
                 // The remain parts of the url are considered as
                // arguments of the method
                $requestedParams = array_slice($this->url, 2);
        }
        //route
        function route(){
                         $requestedpage = $this->controller();
                        if($requestedpage=='/')
                        {
                          //home page
                          $route='home';
                        }
                        else
                        {
                                //other webpage
                                $route=$requestedpage;
                        }
```

```
		return $route;
	}
	function views(){
			$dir = is_dir('./include') ? './include/': '../include/' ;
			$html_Model = $dir.$this->route().'.html';
			$php_Model = $dir.$this->route().'.php';
			if(file_exists($html_Model))
			{
				//check if page is html exist
				$include = include($html_Model);
			}
			elseif(file_exists($php_Model))
			{
				//check if page is php and exist
				$include = include($php_Model);
			}
			else
			{
				//show 404 page if no php or html page exist
				header("location: ./?404");
				 //$include = include($dir.'/404.html');
			}
			return $include;
	}
?>
```

4.2.4 Index Page

This is the app page inclusion framework usually known as the "index page". Each page is injected in into the index page according to the URI parameter. The page injection process is control by the MVC library. The following source code was used.

```
<?php
//ini_set('error_reporting', E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED); //disable all type
of error reporting
	/*************************************************************************
	include all library here
	*************************************************************************
*/
	include('class/class-app.php');
  include ('class/class-mvc.php');
	include('class/class-user.php');
	include('class/class-spamfilter.php');
	/*************************************************************
	Instantiate all class object here
	*************************************************************/
	$mvc = new mvc();//initiate mvc object here

	//logout action
	if($_GET['p']=='/logout/'){
```

```
        $objectApp->logout();
        }
        /*****************************************************
        page include method call here
        *****************************************************/

    //load webpages
   $mvc->views();

        //$page = include($objectApp->pageLoad($_GET['p']) );
         ?>
```

4.2.5 The Home Page

The home page was written in php and html (hypertext markup language) for ui/ux design. This page is known as the inclusion page, into which injectable content are inserted into the index page to make a complete web page

```
<?php
$objectApp = new app();
$pageTitle = "Spam Filter System";
$pageKeyword = "";
$header = include('header-html.php');
$nav = include('include/sidenav.php');
?>

<div id="page-wrapper" class="gray-bg dashbard-1">

        <div class="content-main">
        <div class="col-md-12">
                <div align="center" class="blank-page" style="background-
color:rgba(255,258,257,0.6);">
                <h1 style="color: #FFCC00; :10px;">c<span
style="color:#666666">anitrustit.co.uk</span></h1>
                <span style="color: #ccc; font-size:1.0em">Email phising detection and content
filtering system</span>
                <p style="margin-top:30px;"></p>
                </div>
        </div>

        <div class="col-md-12">
        <div class="blank-page" style="background-color:rgba(255,258,257,0.6);"></div>
        </div>

        <div class="col-md-12">
                <div class="blank-page" style="background-color:rgba(255,258,257,0.6);
padding:20px; display:block; overflow: auto;">

        <?php
```

```
$menu = array(
"
<h3><i class='fa fa-envelope fa-2x text-info'></i> Spam Filter</h3><hr>
<p style='font-weight:100'>Canitrustit Spam filters is designed to identify emails that attackers or marketers use to send unwanted or dangerous content. It built to use different filtering methods to identify the content of emails or their senders and then flag the email as spam. It also examine an email for explicit content that could contain malicious links.</p>
<br><a href='./spamfilter' class='btn-lg btn-primary'>Try for Free</a>",

"<h3><i class='fa fa-cogs fa-2x text-info'></i> Manage Filter</h3><hr>
<p>Built with different spam filtering algorithm such as content keyword filter, Blacklist filter, Header filter and Language filter, enabling scalable knowledge base system that can be fully cutomized according to user preference. Flexibility to manage your content keywords, blacklist database, Languague bank etc and improve your system algorithm accordingly</p>
<br><a href='./keyword' class='btn-lg btn-warning'>Manage</a>
",

"<h3><i class='fa fa-building fa-2x text-info'></i> Forensic Analysis</h3><hr>
<p>CanItrustit is suitable for offline bulk email correspondence check for forensic investigation within an organisation. All email corespondence (incoming and outgoing messages) can be uploaded and analyze for any suspicious phishing activity via cloud-base or API endpoint integration. </p>
<br><a href='./login' class='btn-lg btn-warning'>Learn more</a>
"
);

//var_dump($menu);
 foreach($menu as $menu){
 ?>

<div class="col-md-4 col-xs-12 col-sm-12">
<div align="" class="login-bottoms" style="height:450px;background-color: #fff; rgba(206,124,70,0.9); color:#666666; padding:20px;">
 <br><?php echo $menu ; ?>
 </div>
</div>

<?php } ?>

</div>
</div>
</div>
    </div>
<?php $footer = include('footer-html.php');?>
```

CHAPTER FIVE

SYSTEM EVALUATION

5.1 Evaluation Process and Results

To evaluate the functionality and effectiveness of the system, a quasi-experimental design was used (Kampenes et al., 2009). Quasi-Experimental research design is a method in which study units are assigned to experimental groups non-randomly (Laitenberger and Rombach, 2003). This involved the use of simulated vignettes. The evaluation process is highlighted as follows. Ten individuals were purposively and conveniently selected to participate in the evaluation process. The participants were divided into two groups; Group A and Group B. The members of each group were tasked with writing an email ad to a designated email address.

- Members of Group A were given a text card containing a list of phrases which should be used in composing their ads. Their ads must consist of five or more of the phrases from the list to be used in both the email subject and email content.

- Members of Group B were not given any text cards. They just needed to compose email ads based on their subjective creativity. They were however told to make it look as genuine as they possibly could.

Unknown to the participants, the phrases in the text cards had all been input as blacklisted keywords in the 'can-i-trust-it' system. It was expected that email ads received from Group A will be categorized as having more significant level of spam while email ads received from Group B will be categorized as having less significant level of spam.

Results obtained after the analysis of the emails showed that 80% of emails obtained from Group A Members were categorized as 'low risk spam' while the remaining 20% were categorized as 'high risk spam'. The majority of the mails from Group A were categorized as 'low span risk' because the IP address and sender domains were not flagged as 'blacklisted'

while the email subject and contents contained the use of blacklisted phrases. The 20% with high risk spam had significantly more usage of the blacklisted phrases. On the other hand, results obtained from the analysis of mails from Group members showed that 60% of them were categorized as ‘spam, not harmful’, while 40% were categorized as ‘Not Spam’. The initial batch categorized as ‘spam, not harmful’ showed that the system detected sparse usage of some of the keywords in the blacklist, but the pattern of usage did not seem malicious.

5.2 Evaluation Report

Based on the results obtained a spam filter prototype has been implemented. Outcomes from the evaluation process show that the ‘can-i-trust-it’ system is an effective spam filter which may be used to detect phishing activities. The potency of the system is however dependent on the user’s dexterity in identifying keywords and other signatures that would be included in the system’s blacklist database. Thus, the system can be used for context specific spam detection or general spam detection. Thus, to increase the potency of the system over time, the user has to carry out regular updates of the blacklist database. The system has the ability to filter spam based on filter rules managed by the administrator through the filter management interface. Filtering is based on matching keywords or phrases. Keywords and phrases are often identified and collected from common spam emails.

Filter rules appear in the form of "if condition, then action". A filter rule's criteria section contains keywords or phrases that characterize spam and the overall priority of the filter rule. The action section of a filter rule specifies the categories to mark emails for when the keywords or phrases in the criteria section match. . A filter rule is said to be triggered by an email if the email matches a keyword or phrase in the filter rule's condition domain. Domains that the system checks for in email include sender's IP address, sender's email address, subject, context, attachment name, and attachment content. If an email triggers multiple filter rules, the highest

priority filter rule determines which category the email will be flagged for. If all triggered filter rules have the same priority, the peak score of each triggered filter rule determines the labeled classification of the email.

CHAPTER SIX

CONCLUSION AND FUTURE DIRECTIONS

6.1 Summary

The discourse on spam is gaining much more attention due to the increasing prevalence of spam related challenges in email domains and other text-messaging platforms. This has led to a proliferation of various spam detection and filtering measures to mitigate and address various spam related challenges in the society. In this dissertation, an examination of how to more effectively use the semantic information conveyed by the email body content to distinguish between spam and non-spam emails was the main focus. Traditional approaches to spam detection based on text mining are largely limited to spam detection based on lexical and general semantic information. However, in our approach, the analysis of email content is at two semantic levels to create domain-specific spam classifiers that enable more effective spam detection. This approach radically exploits the information contained in the text by distinguishing semantic context within different domains (categories) targeted by spammers.

6.2 Contribution

Therefore, this research mainly resulted in his two major contributions. The first contribution is the development of a system that has the ability to function based on context specific domains when analyzing the emails. For instance the user is able to utilize the system in fields of health, education, finance, etc.) to give a conceptual view of spam for each domain separately, and to provide domain-specific spam detection. A set of generated semantic features expressed in the form of logical rules. These semantic features provide an accurate description of each spam domain, enabling better detection. The researcher shows that the designed system provides an efficient representation of the internal semantic structure of email content, enabling more accurate and interpretable spam filter results compared to existing methods. The second

contribution of this system in relation to existing prototypes is an improvement in detection techniques by considering a hybrid approach that combines manually configured and auto-generated rules. Auto-generated rules efficiently capture basic semantics, while manually specified rules enable semantic tuning by incorporating domain-specific knowledge from experts and end-users. The combination of manual and automatic rule types provides richer semantic capabilities and enhances spam detection.

6.3 System Limitations

As with all system designs, the spam detection and filtering system designed in this study is not without its limitations. For instance, the fact that the system is not a ready-made spam filter may reduce its popularity. Many users are currently conversant with packaged spam filters which are ready-made and incorporated in their mailing service. However, the spam filter designed in this study is a standalone system which is designed for user preferences to be configured before use. Therefore it may not serve as a ready-made option for immediate use. Moreover, the potency of the system in spam detection and filtering is based on the subjective evaluation of the user. This is because the categorization of spam is based on the user's operationalization and definition of spam content. Furthermore, the absence of artificial intelligence (AI) algorithms to enhance the learning capabilities of the system poses a limitation to users who are in search of alternative artificial intelligence (AI) spam filtering systems. Thus system updates would be dependent on the manual input of user, as well as the subjectivity in regularity and quality of such manual inputs. The absence of this feature may therefore affect the marketability of the system in an industry that is currently driven by trends in artificial intelligence and robotics.

6.4 Marketing Implications

Having designed this open source intelligence for spam filtering, its viability as a marketable product needs to be considered. The system is considered useful in a society that is challenged by spam and phishing attacks on a daily basis; however the system represents just one of several similar systems that abound in the market and are available for users. Therefore, the system has to be able to withstand the competitive nature of the market for its value to significant. it is however expected that with the vast disparity in user preferences and other factors that determine the purchasing behaviour of customers, the 'can-i-trust-it' system should appeal to a market segment of the society if effective marketing strategies and considerations are made. A few of the marketing issues are highlighted below;

- **Acceptability**

New products flood the market on a daily basis and customers need to have a certain level of acceptability of these products before they can be viable in the market. In relation to the spam detection and filtering system designed in this study, the issue of enhancing its acceptability needs to be addressed. In doing so, factors to be considered includes its user friendliness, its aesthetic value, its affordability (in terms of cost), its availability, its effectiveness over other alternatives, its unique features, etc., In designing the system the ease of use, aesthetic appearance and unique feature of customizability were considered. However, as market product expert consultations have to made on its economic value and cost in relation to market trends.

- **Trust**

Based on the fact that the spam filter designed in this study is a stand-alone system, its usage would be viewed as a third party in the messaging service. This is because, using the system may involve another entity having access to a mail content ordinarily meant for the sender and receiver. This scenario can raise trust issues, especially when mail contents are supposed to be confidential. For instance, mail correspondence between lawyers and their clients are

confidential in nature; however the use of a third party spam filter may be viewed as compromising the confidential nature of such correspondence. This is a major consideration to be taken into account in the marketability of the system. There is need for measures to be put in place to raise the stakes of trustworthiness and trust towards the third party presence in the utilization of the system.

- **Reputation**

Cyber criminals are developing new techniques from time to time in order to maneuver around solutions developed against their criminal activities by finding vulnerabilities existing in the solutions and rendering ineffective in the new exploit applications. The lack of assurance of security over time has reduced the reputation of solutions developed against cybercrimes. This can-i-trust-it solution is yet to build its reputation and therefore relies on its usage by individuals and organisations to build it.

6.5 Directions for Future Studies

In future works, other studies may intend to enhance this study's approach in several dimensions. Sentiment analysis (SA) is used in the extraction and analyses of knowledge from provided data or feedbacks using machine learning natural language processing (NLP). Future solutions in mitigating phisphing using AI's sentiment analysis approach of deep learning-based approach which would assist in providing better solutions to evaluating the quality of service and product than other traditional technologies (AlBadani et al., 2022), and so, more research and technologies related to the SA should be carried out and developed.

The use of semantic features can be explored. For instance, the polarity of the message may be used to identify the tone, emotion and opinion in the mail contents. Such polarity in mails may be identified via sentiment analysis which involves labeling the message contents as positive, negative or neither. Another dimension may be to create and online and efficient updating

procedure for the semantic features and domain classifiers. This can be achieved by considering the right feedback algorithms, which then enables you to get a dynamic system that refines over time and adapts to the user's needs. Additionally, the semantic rules may need to be updated to process the data developed. This process is described as concept drift and can be addressed in a number of ways, including using online rule-based algorithms and evolutionary rule-learning approaches. The latter allows new data to fit existing knowledge. Finally, for processing large rule sets after the rule combining step, it may be interesting to devise an evaluation step that uses rule quality measures to sort and select the most important rules.

References

AlBadani, B., Shi, R., & Dong, J. (2022). A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM. *Applied System Innovation, 5*(1), 13. https://doi.org/10.3390/asi5010013

Aldweesh, A., Derhab, A. & Emam, A.Z. (2021) "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," Knowledge-Based Systems, 189, 105-124

Baig, M.M., Awais, M.M. & El-Alfy, E.S. (2017) multiclass cascade of Artificial neural network for network intrusion detection. J. Intell. Fuzzy Syst. 32, 2875–2883.

Blanzieri, E & Bryl, A. (2008) "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence Review*, 29(1), 63–92.

Bul'ajoui, W, James, A.E & Pannu, M. (2015). Improving network intrusion detection system performance through quality of service configuration and parallel technology. Journal of Computer System Science, 81(6), 981-999

Cahyana, R. (2018). A Preliminary Investigation of Information System using Ishikawa diagram and sectoral statistics. *IOP Conference Series: Materials Science and Engineering*, *434*, 12050. https://doi.org/10.1088/1757-899x/434/1/012050

Chen, J. C., & Li, B. (2015). Evolution of exploit kits: Exploring past trends and current improvements (Research Paper). Irving, Texas: Trend Micro. Online. Retrieved from

Cisco. (2022, September 19). *2021 Cybersecurity Threat Trends: Phishing, Crypto Top the List*. Cisco Umbrella. https://umbrella.cisco.com/info/2021-cyber-security-threat-trends-phishing-crypto-top-the-list

Dullian, T. (2011). Exploitation and state machines: Programming the \weird machine", revisited [Presentation]. Online. Miami Beach. Retrieved from http://titanium.immunityinc.com/infiltrate/archives/Fundamentals%5fof%5fexploitation%5frevisited.pdf

ESET. (2022). *Threat Report T2 2021 Foreword*. https://www.welivesecurity.com/wp-content/uploads/2021/09/eset_threat_report_t22021.pdf

Fahad, S. (2015). Developing a spam Email Detector. *International Journal of Innovative Research in Computer and Communication Engineering, 5*(2), 16-21.

Ghavamzadeh, M., Mannor, S., Pineau, J. and Tamar, A. (2015). Bayesian Reinforcement Learning: A Survey. *Foundations and Trends in Machine Learning, 8*, 359-483.

Goel, J.N. & *Mehtre*, B.M. (2015). Vulnerability Assessment &. Penetration Testing as a Cyber Defence Technology. Procedia, 57, 710 – 715

Howard, F. (2012). Exploring the Blackhole exploit kit (Technical Paper). Abingdon: SophosLabs, UK. Retrieved from https://sophosnews.files.wordpress.com/2012/03/blackhole%5fpaper%5fmar2012.pdf

https://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-evolution-of-exploit-kits.pdf

Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011, March). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Proceedings of the 6th International Conference on i-Warfare and Security, 1

IBM. (2022). *X-Force Threat Intelligence Index*. https://www.ibm.com/downloads/cas/M1X3B7QG

Karbalayghareh, A., Qian, X. and Dougherty, E. R. (2018). Optimal Bayesian Transfer Learning. *IEEE Transactions On Signal Processing*, 66, 3724-3739.

Kendall, A. and Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In 31st Conference on Neural Information Processing Systems. NIPS 2017.

Kenkre PS, *Pai* A, *Colaco* L. (2015) Real time intrusion detection and prevention system. Information & Management., 52(1),123-134.

Koziol, J., Litch_eld, D., Aitel, D., Anley, C., Eren, S., Mehta, N., & Hassell, R. (2004). The shellcoder's handbook. Indianapolis: Wiley.

Kumar, N & Sonowal, S. (2020) "Email spam detection using machine learning algorithms," in *Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 108–113, Coimbatore, India, 2020.

Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L. F., & Hong, J. (2010). Teaching Johnny not to fall for phish. ACM Transactions on Internet Technology, 10 (2), 1{31.

Liu, H., & Lang, B. (2019) "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey". Applied

Malek, Z.S.. Trivedi, B. & Shah, A. (2020), "User behavior Pattern-Signature based Intrusion Detection," In 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4) (pp. 549-552).

Mayer, D. (2012, November 11). ratio of bugs per line of code [Blog Post]. Online. Retrieved 2016-06-17, from http://www.mayerdan.com/ruby/2012/11/11/ bugs-per-line-of-code-ratio/

McConnell, S. (1998). Feasibility Studies. *IEEE Software*, *15*(3), 120, 119. https://doi.org/10.1109/52.676989

Microsoft Security TechCenter. (2012, May). Security bulletin severity rating system. Online. Retrieved 2016-06-17, from https://technet.microsoft.com/en-US/security/gg309177.aspx

Mitnick, K. D., & Simon, W. L. (2002). The art of deception: Controlling the human element of security. Indianapolis: Wiley Publishing.

Moustafa, N., Turnball, B. & Kwang, K. (2018) An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. IEEE Internet Things J., 6, 4815–4830.

Open Web Application Security Project. (2013, December 31). Code injection [Wiki Page]. Online. Retrieved from https://www.owasp.org/index.php/Code%5fInjection

Otoum, Y. & Nayak, A. (2021). "AS-IDS: Anomaly and Signature Based IDS for the Internet of Things," Journal of Network and Systems Management, 29(3), 1-26, 2021.

Prokofyeva, N., & Boltunova, V. (2017). Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. *Procedia Computer Science*, *104*, 51–56. https://doi.org/10.1016/j.procs.2017.01.059

Rai, K., Devi, M.S. & Guleria, A. (2016) "Decision tree based algorithm for intrusion detection," International Journal of Advanced Networking and Applications, 7(4), 28-39

Rapid7. (2012, December 19). Social engineering security and phishing with Metasploit. Retrieved from http://www.rapid7.com/resources/videos/ phishing-with-metasploit.jsp

Rosenthal, M. (2022, August 25). *Must-Know Phishing Statistics: Updated 2020*. Tessian; Tessian. https://www.tessian.com/blog/phishing-statistics-2020/

Sahara Reporters, New York (2021) https://saharareporters.com/2021/11/15/nigerian-government-warns-new-iran-based-hacking-group-targeting-telecoms-companies

Sahara Reporters, New York (2022) http://saharareporters.com/2022/01/10/exclusive-hacker-breaks-nimc-server-steals-over-three-million-national-identity-numbers

Saleh, A.J., Karim, A., & Shanmugam, B. (2019), "An intelligent spam detection model based on artificial immune system," *Information*, 10(6), 209-215

Segura, J. (2015, January 21). Exploit kits: A fast growing threat. Retrieved from https://blog.malwarebytes.com/101/2015/01/exploit-kits-a-fast-growing-threat/

Shanguo, Z. (2016). The Software Engineering Analysis in Computer Science and Technology. *International Conference on Education, Management, Computer and Society*. https://www.atlantis-press.com/article/25848893.pdf

Sharma, V., You, I., Yim, K., Chen, R. & Cho, J.H. (2019) "BRIoT: Behavior rule specification-based misbehavior detection for IoT-embedded cyber-physical systems," IEEE Access, 7, 118556-118580.

Sonule, A.R, Kalla, M., Jain, A. & Chouhan, D. (2020) "UNSW-NB15 Dataset and Machine Learning Based Intrusion Detection Systems," International Journal of Engineering and Advanced Technology, 9, 2638-2648

Stock, B., Livshits, B., & Zorn, B. (2015). KIZZLE: A signature compiler for exploit kits (Tehnical Report). Redmond: Microsoft. Online. Retrieved from http://research.microsoft.com/pubs/240495/tr.pdf

Thakkar, A & Lohiya, R (2020) "A review of the advancement in intrusion detection datasets," Procedia Computer Science, 167, 636-645

Tirpak, J. A. (2000, July). Find, _x, track, target, engage, assess. The Air Force Magazine. Retrieved 2016-06-16, from http://www.airforcemag.com/MagazineArchive/pages/2000/july%202000/0700find.aspx

TrustedSec. (2013, September 11). Introducing SpearPhisher a simple phishing email generation tool. Retrieved from https://www.trustedsec.com/september-2013/introducing-spearphisher-simple-phishing-email-generation-tool/

Tsipenyuk, K., Chess, B., & McGraw, G. (2005). Seven pernicious kingdoms: A taxonomy of software security errors. Security & Privacy, IEEE, 3 (6), 81-84.

Verizon. (2022). *Data Breach Investigations Report*. Verizon Enterprise. https://enterprise.verizon.com/en-gb/resources/reports/dbir/

Verma, J., Bhandari, A. & Singh, G. (2020) “Review of existing data sets for network intrusion detection system,” Advances in Mathematics: Scientific Journal, 9(6), 3849-3854

Yulianto, A., Sukarno, P. & Suwastika, N.A. (2019) Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset. J. Phys. Conf. Ser., 1192, 012-018.