2021

A Honeypot based malware analysis tool for SACCOs in Kenya.

Mwendwa, Keith Mwesigwa
Faculty of Information Technology
Strathmore University

A Honeypot Based Malware Analysis Tool For SACCOs in Kenya

Keith Mwesigwa Mwendwa

Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Information Technology at Strathmore University

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

May, 2021

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the proposal contains no material previously published or written by another person except where due reference is made in the proposal itself.

Keith Mwesigwa Mwendwa

September 10, 2021

Approval

The thesis of Keith Mwesigwa Mwendwa was reviewed and approved by the following:

Dr. Humphrey Njogu

Lecturer, School of Computing and Engineering Sciences,

Strathmore University

Dr. Julius Butime

Dean, School of Computing and Engineering Sciences,

Strathmore University

Dr. Bernard Shibwabo,

Director of Graduate Studies

Strathmore University

Abstract

Kenya had her first established Savings and Credit Co-operative (SACCO) society in 1908 and to date, the SACCO societies have grown into a Billion-dollar industry. SACCOs contribute 5.72% to Kenya's Gross Domestic Product (GDP) and are significantly changing the lives of Kenyans in almost all sectors of the economy. Like other sectors, SACCOs are facing growing cyber threats that have potential to affect their performance. The report by Serianu of 2018 indicates that SACCOs have poor visibility on enterprise cybersecurity and thus are poorly prepared to anticipate risk, detect vulnerabilities, respond to incidents and contain threats. Further, SACCOs have low budget allocations and inadequate skilled staff to advise in prevention and protection against threats. Because of this, SACCOs across the globe lose hundreds of millions of dollars annually. The Serianu Cyber Security Report of 2018, indicates that the global cost of cybercrime was at 600 billion dollars in 2015, which had risen by $100 billion from the previous year. The report indicated the SACCOs were the most affected, while the affected organizations lost money, experienced downtimes and reputation damage. It is observed that many SACCOs in Kenya are slowly putting up measures to prevent, detect, and remediate cyber-attacks with minimal resources. This study intends to help SACCOs have a paradigm shift in how to detect and respond to malware by developing a prototype. The literature review brought to light the different applications of honeypot solutions, but the solution is not common within the SACCO industry. The prototype, a honeypot that was used for malware analysis in order to determine breach scenarios and common cyberattacks showed outstanding performance when run for a few days, in capturing malware, and helping in their analysis. The proposed solution enables SACCOs to better mitigate and possibly reverse Cyber-attacks on their infrastructure due to the information they get from analysing malware. Development of the prototype was based on Rapid Application Development methodology to build a robust malware analysis tool on Honeypots and was tested for reliability where it showed an outstanding accuracy level as all attack traffic was captured and logged. While from the first 24 hours of uptime, in 100 captured attacks, the prototype was able to give Md5 hashes of 11 malwares, the prototype captured the IP addresses associated with the rest of the attacks which can be blacklisted by a SACCO employing this tool.

Keywords: SACCOs; Malware analysis; Honeypots; Cyberattacks

TABLE OF CONTENT

List of Figures

List of Tables

Abbreviations and acronyms

CBK – Central Bank of Kenya

CIF - Collective Intelligence Framework

CHN - Community Honeynet Network

ELK – Elasticsearch, Logtash and Kabana

FTP – File Transfer Protocol

HTTP – HyperText Transfer Protocol

ICT - Information and Communications Technology

IDS – Intrusion Detection Systems

IP – Internet Protocol

IPS – Intrusion Detection Systems

Kes. – Kenya Shillings

LAN – Local Area Network

Malware – Malicious Software

Md5 - Message digest Algorithm

MSSQL – Microsoft SQL Server

SACCOs – Savings and Credit Cooperative Societies

SASRA - SACCO Societies Regulatory Authority

SMB – Server Message Block

SSH – Secure Shell

TFTP – Trivial File Transfer Protocol

VoIP – Voice over Internet Protocol

VPN - Virtual Private Network

Definition of Terms

Docker - is a software container platform used by users such as developer, researcher, enterprise production in many purposes to build environment and automation in order to deploy their applications (Docker Documentation, 2017).

Malware – Is a general term used to mean malicious software or code (Ali and Kumar, 2017)

Admin – In this research document, Admin is used to refer to a Systems Administrator, IT Officer or any human resource working in Information Security

Acknowledgement

I thank God for the opportunity of advancing my studies at Strathmore University, I am grateful for the swift and continuous support I received from my supervisor Dr. Njogu, the Thesis course co-ordinator Dr. Omwenga and classmates who helped through brainstorming sessions and discussions that helped tremendously during the entire period of my studies.

Dedication

I dedicate this to my loving family for their tireless support and understanding during my study period and in a special way, my daughter for giving me the motivation I needed.

CHAPTER ONE: INTRODUCTION

1.1 Background of the Study

Kenya had her first established Savings and Credit Co-operative (SACCO) society in 1908 (Gatuga et al 2014) and to date, the SACCO societies have grown into a Billion-dollar industry, contributing 5.72% to Kenya's Gross Domestic Product while changing lives of Kenyans in almost all sectors of the economy (Serianu, 2018). SACCOs, one of the most important and often visible type of cooperatives in Kenya are distinguished by their unique character trait from other types of cooperatives in their object and purpose for which they are incorporated, which is to transact the business of mobilization of savings and advancement of credit facilities to their members (SASRA, 2018).

According to a SASRA report(2018), in 2018, there saw an increase in growth of total membership in SACCOs increasing by 16.5% to reach 4.2 million members, up from 3.6 million the previous year. The report indicates that the total assets grew by 11.97% to Kes. 495 billion up from Kes. 442 billion the previous year. The total deposits grew by 11.99% to Kes. 341.9 Billion up from Kes. 305 billion, while gross loans grew by 13% to Kes. 374 billion up from Kes. 331 billion the previous year. Considering the above figures, and knowing that SACCOs do not invest heavily on cybersecurity, meaning less security measures are put in place to guard against potential attacks, it leaves them exposed and risk losing gains gathered over many years (Serianu, 2018).

According to the Kenya Financial Sector Stability report (2018), payment systems in Kenya have undergone changes, including innovations in financial technologies (FinTechs) that support electronic-based payment systems. These innovations have accelerated financial inclusion, reduced cost of transaction and handling of cash in the economy. The same report further states that the Central Bank of Kenya has actively supported these innovations, thus promoting efficiency in business operations, cost reductions, enhanced security, and wider payment channels. However, these have come with cybersecurity threats given their interconnectedness and heavy reliance on information technology. Thus, cyber security risks remain a major threat to all payment systems given that payment service providers operate in an interconnected and interdependent environment where the consequences of a cyber-attack on one can cascade to numerous others. As a result of

these trends, there has been an increase in cyber-attacks, specifically targeting the SACCO industry (Serianu 2018). According to the same report, the top targeted attacks on SACCOs have been Email phishing attacks, Ransomware and Malware attacks among others.

SACCOs are a lucrative target for attackers and stand to lose the most with the recent increase in cyber threats due to limited visibility on their enterprise cyber Security posture. About 83% of SACCOs manage cyber Security in-house, with majority relying on a perimeter firewall and an antivirus, while a good number using default security settings. Having mentioned that, it becomes apparent that SACCOs have the paramount need to have cyber preparedness, with goals to develop preventive measures targeted at the dynamic vulnerabilities and cyber threats businesses (Gomez et al, 2016).

According to a survey done on SACCO Cyber Security by Serianu (2018), there was an increase in cyber-attacks targeting the SACCO industry, and the top six were: Database breaches, Abuse of privilege access, keyloggers, Ransomware, Email phishing attacks, and critical data manipulation. The attackers are leveraging on database manipulation attacks, abusing privileged user accounts, capturing keystrokes and other sensitive information such as passwords. These can be delivered intentionally or accidentally through different kinds of malwares which could be Viruses, worms, Trojan, rootkits or even botnets. These kind of malware have the following features, self-replicating, can attach its code into legitimate program and become active each time this program runs, they can use the network to replicate, keep track of user's activity without their knowledge, disguise as a legitimate program, give access to a remote hijacker or malware to control user system, turn computer to a zombie (Deka et al 2016). These attacks have led to the loss of sensitive data and business intelligence within the SACCOs, downtimes, (distributed) denial of service, impersonation of users to perform fraudulent activities etc, which may lead to decline in market share, consumer trust, leakage of critical information, financial losses and other damages.

Phases of a malware attack as explained in Lockheed Martin's Cyber Kill Chain are (lockheedmartin.com, 2021) :

i) Reconnaissance - Which is the phase where harvesting is done for email address, conference information, IP address etc.

ii) Weaponization - Where coupling of exploit with backdoor to deliverable payload.

iii) Delivery - This the phase where delivery of weaponized bundle to the victim is done through various means such as the victim's email, web, USB, etc.

iv) Exploitation - it is in this phase where a vulnerability is exploited to execute code on a victim's network or system.

v) Installation - After a vulnerability is exploited successfully, malware is installed

vi) Command and Control of victim computer for remote manipulation of victim/asset.

vii) Action on objectives, which could be exfiltration or destruction of data, or intrusion of another target.

The lockheedmartin cyberkill phase show show the different stages of a malware attack which may mean that capturing an attack or malware at any of the phases in real-time may be challenging using techniques being used in SACCOs such as firewalls, antivirus and other antimalware without the presence of an Admin or without his/her knowledge (Borkar et al, 2017). An IDS solution technically, is a software or even a device that works by monitoring computer systems, then studies the traffic for any patterns of intrusion that is not authorized surrounding matters Authentication, Confidentiality, Integrity and Availability, (Chandre et al, 2018). An IPS such as a firewall, is also a software or device that can also be seen as an IDS, only that it has the capabilities of stopping possible intrusions or attempts of attack, (Chandre et al, 2018).

IPS/IDS systems are important because of their role, which is detection, prevention and reporting of any cyber breaches/attacks on an enterprise's network (Kılıç et al, 2019). As opposed to Honeypots which openly attracts an attacker to a safe environment, IPS/IDS operate in the actual/live environment, and reporting at times may be after damage has been done and there is little to salvage. Another downside to IPS/IDS is that while existing IPS/IDS may miss an attack/malware, there is a large weakness brought about by attacks using evasion techniques that are designed to bypass IPS/IDS systems to deliver exploits, attacks or malware to victims without being detected, (Jingping et al, 2019).

Honeypots are computer security mechanisms that attempt to detect, mislead or otherwise interfere with attempts to exploit a system, while providing means to thoroughly analyse security events in a modular fashion (Vadaviya, 2019).

According to Fraunholz et al, (2017), honeypots are an advanced concept in network security, and what makes them advanced is that they help an institution learn more about attempts of intrusions; they help bring out information such as date and time stamp of an attack, IP address of the attacker and the operating system they are using, wordlist being used, exploit detail and other commands after infiltration. While all this is taking place, the attacker is being kept busy in what they think is an actual system, while an institution such as a SACCO could be beefing their cybersecurity systems in preparation for a possible attack from an informed point of view.

This study implemented a Honeypot prototype that helped divert would-be attackers from the SACCOs critical systems, to the honeypot since most attacks are based on known common vulnerability exposures. This enables SACCOs learn tactics, techniques and procedures that intruders use to infiltrate them, and possible use this information to analyse and understand malware types that get to their network and would therefore be able to protect themselves.

1.2 Problem Statement

SACCOs in Kenya are riddled with cyberattacks according to Serianu's Sacco CyberSecurity Report (2019). Despite the warning on the increase in cyberattacks most of these SACCOs are not cybersecurity ready, raising doubts on their ability to keep customer data safe (Muraguri et al, 2019). Generally, SACCOs have poor visibility on enterprise cybersecurity posture, making them poorly prepared in anticipating risk, detecting vulnerabilities, responding to incidents, and containing threat (Serianu, 2018). Most SACCOs lack secure infrastructure and have limited capabilities for malware analysis. Understanding these malware attacks will put SACCOs on a vantage position in terms of threat intelligence and more importantly, they will be in a better position in allocating the right resource(s) to the areas of most need.

It is noted that malware detection is a problem, and furthermore, a computer infected by malware many a times is likely to be used as a "springboard" of malicious cyberattacks (Saikawa and Klyuev, 2019). The duo further state that a honeypot based solution act as a decoy system in the face of attack, and offers a tool for analyzing the malware and the attack methods employed, this makes the solution even more useful in the SACCO environment considering it is almost impossible to handle the vast number of malware attacks by human engineers (Sihwail et al, 2018). The honeypot prototype proves useful in such a situation as it attracts malicious traffic making it

easy to read and learn important details of malware in the SACCO environment as compared to existing tools for malware analysis which require expertise expertise and a lot of time (Aslan and Samet, 2017).

1.3 Objectives:

1.3.1 General Objective

The aim of this study is to develop a honeypot based malware analysis tool to detect and analyse malware targeting SACCOs.

1.3.2 Specific Objectives

i. To identify the common malware targeting SACCOs
ii. To review the existing malware detection and analysis techniques and tools
iii. To design, develop and test a Honeypot based prototype for malware analysis in SACCOs
iv. To validate the effectiveness of the honeypot based prototype

1.4 Research questions

i. What malware are prevalent within SACCOs?
ii. What are the existing techniques and tools used to detect and analyse malware as a response to cyber-attacks in SACCOs?
iii. How can a honeypot prototype be designed, developed and tested, to analyse malware for SACCOs?
iv. How can the effectiveness of the proposed prototype be validated ?

1.5 Justification of the study

Each SACCO owes their members and themselves protection revolving around the Confidentiality, Integrity and Availability (CIA) triad on the valuable data they hold and assets and moneys which run into billions. This research intends to help SACCOs achieve just that with available resources that will not need for them to have bigger budgets.

The prototype has proven to work as expected, meaning that SACCOs stand to benefit significantly from this study. What we were interested in, is the capturing of attacks which provides information

on the attack, helping SACCOs better understand attacks that may occur thus helping them in their preparedness of cyber-attacks, this can further allow SACCOs to learn new attacks thus creating new solutions including policies in cybersecurity revolving around honeypots and malware on the enterprise level. As a result, SACCOs save on matters finances considering they have little cybersecurity budgets, in addition, Honeypots are generally accessed by malicious traffic only, meaning there is no overload of information on traffic, no extra storage for data is required, and any computer can be made to work as a honeypot meaning no extra unnecessary budget would be needed to create such a system.

This research therefore contributes to the knowledgebase in this area of study and further opens a lee way into research areas on SACCOs and cybersecurity vis-a-vis Honeypots for malware analysis and other types of honeypots such as dynamic honeypots and other solutions or areas pertinent to SACCOs and the Financial sector cyber-preparedness.

1.6 Scope and Limitation

This research was limited to use of low-interaction honey-pot to capture attack payloads and malware. The researcher was also limited to working with virtual machines set up in the cloud due to the sensitivity of having such a prototype in the SACCO at this stage of the study. The researcher used Virtual Machines that acted as honeypots, by creating and using Community honeypots to work as a multisensor honeypot, and has imbedded on it tools such as a Community Honeynet Network server which has dashboard that helps project logs in a more presentable manner. The researcher also employed other open source tools for further malware analysis such as VirusTotal that worked as intelligence framework which helped offer further information on malware captured. The information gathered here is what SACCOs may use to harden their firewalls, intrusion detection systems, antivirus to better their cyber-attack preparedness.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter delves deeper in previous work surrounding honeypots for malware analysis vis-à-vis SACCOs cyber security posture. This chapter therefore intends to discuss existing malware detection tools and attacks, analysis of the malware and development of a honeypot prototype that may be adopted in the fight of cyberattacks within the SACCO industry through analysis of malware.

2.2 SACCOs in Kenya

SACCOs are one of the most important and often visible type of cooperatives in Kenya and are distinguished by their unique character trait from other types of cooperatives in their object and purpose for which they are incorporated, which is to transact the business of mobilization of savings and advancement of credit facilities to their members (SASRA, 2018). According to Serianu (2018), in 2018 the SACCO industry grew their asset base to KES. 495 Billion, a growth of 11.97% from the previous year, and the growth in membership as well which has been the case for years. The SACCO society in Kenya had approximate 3.6 million members as per the SASRA Sacco Supervision Report (2017), and what was noted was that the IT Administrators double up as the human resource in charge of cyber security. This leads to a gap in the required skillsets in managing time and according the right resources to cyber security matters (Serianu, 2018). Thus the need for this study that intends to bring about a simple prototype implementable in the SACCO environment, that would help in the capturing and most importantly analysis of malware, that would go a long way in guiding these institutions in matters cyber-security, considering this, there was a paramount need to have the industry regulated, to protect the general public interests.

Thus, through an Act of Parliament, on June 18, 2010, The Sacco Societies Act was gazetted which initially regulated the Deposit Taking Saccos only (Sacco Societies Act, 2010). In accordance with the Sacco Societies (Non-Deposit Taking Business) Regulations (2020), The Authority has issued a notice on the commencement of the process of supervising and regulating SACCOs that undertake non-deposit-taking SACCO business, this regulations took effect on 1st January 2021 and thus SACCOs had until 30th June 2021 to comply.

2.3 Common malware targeting SACCOs

According to a Serianu report (2018) based on a survey done on SACCOs, attacks are mostly on clickjacking malware, ransomware, trojans, viruses, brute-forcing attacks and other malware that may edit file systems.

Malware attacks today have been developed in such a way that they can target Financial systems such as SACCOs and are expected to increase. Common Malware as depicted by Wazid et al., (2019), include and not limited to:

Keyloggers: Malicious program that records everything one types which can be used to steal credentials of a user and other sensitive information of an organization, keyloggers may be seen as a type of spyware.

Spyware: are essentially programs that track crucial information from a system. These can be used for espionage etc. Spyware spies on users by covertly reading and collecting information such as keystrokes, login information, location, etc without the knowledge of the user. This information can then be sold to third parties such as cyber criminals (Kaspersky, n.d).

Virus: These are infectious programs that attach to other programs or software then reproduce self when the host executes. They end up destroying data, files, partitions and other software including Operating Systems in the infected system. Viruses are spread through downloading of infected files from the internet, email attachments, shared drives, or physical removable media like USB sticks (Kaspersky, n.d).

Worm: Is a program that replicates self and end up competing for the same resources on a system with other applications. Unlike the computer virus, worms do not require human intervention to replicate, infect and/or spread. Upon breaching a host, they infect their point of entry, thereafter propagating throughout the host and the network the host is connected to. According to a Kaspersky (n.d), worms were known to originally "eat" system resources thus reducing the host's performance, nowadays, with their capabilities to exploit vulnerabilities such missed operating system patches, weak security configurations, they may contain "payloads" crafted to exfiltrate data or even delete files after executing.

Trojan: These are programs coded with the purpose of getting access to a system as a friendly harmless file, only to give access to other malicious programs that end up being malicious.

Rootkit: This is a type of software that hides its presence or the presence of another program (e.g a reverse shell code) by making the use of operating system features such as application programming interface redirection. These are designed to allow access and/or control to third parties to a computer. This access is common in granting ICT Professionals to enable them troubleshoot computer and network issues remotely, however, they may be used maliciously to allow attackers take full control of a host where they can covetly steal information or install more malware (Kaspersky, n.d).

Banking malware has become popular and ever more prevalent mechanism to monetise malware development (Black and Opacki, 2016). Attackers and malicious software with auto-propagating capabilities are constantly scanning the internet for vulnerable targets such as Servers/computers that are exploitable through a particular type of attack (Papadimitriou et al., 2014). This becomes possible as an attacker scans running services to determine weaknesses to launch their attack vector. Malware attacks on SACCOs has become even easier as even script-kiddies or "hacker-wanna-be" can have a shot by simply using available open source tools such as Metasploit framework and visiting Websites like Exploit-DB that archives thousands of vulnerabilities that have a proof of concept code that can be run targeting a discovered vulnerability on a system for malicious intent. Considering the cybersecurity posture of SACCOs aforementioned, poor policies, weak infrastructure, having a gap in cyber security professionals, it may be relatively easy to target the SACCOs.

Cyber-attacks are not only external attacks from third parties, they can also be from within, as employees leak sensitive data, easily fall for phishing attacks or being compromised to act on behalf of an outsider. By understanding motives of a hacker, SACCOs will have better footing in understanding threats thus their mitigation.

2.4 Malware detection and analysis techniques

Traditional defences typically use signature based techniques to detect malware and prevent their activity, these tools could be malware detection and antimalware solutions such as firewalls, Antivirus among others (Talukder, 2020). The signatures or patterns get preconfigured, and say the antimalware or firewall monitor the traffic searching for patterns that could be a malware executing or attack going on. Another prevention and detection technique is by adopting an anomaly based detection technique which provides protection from unknown attacks (Sawant,

2018). In this case, the network settings is set for only trustworthy activity then the network gets regulated using network patterns.

According to Arneja and Sachdev (2015), firewalls are mostly external to a computer and protect several hosts at the same time, by intercepting all traffic and thus prevents unwanted traffic from getting inside an institutions network. This is so, as Firewalls are placed at the point of entry usually between a private network and the internet such that all packets incoming and outgoing pass through it, and mapped to a decision according its configuration policy which are a sequence of rules that will define the packet as legitimate or illegitimate (Liu and Gouda, 2009).

Antivirus is inside the network and works by detecting malware that is already inside the network, and performs necessary action such quarantining or deleting the malware, the antivirus is usually the last line of defense, at the end user computer or mobile device such as a tablet (Shahegh et al, 2017). However, as antivirus softwares become more sophisticated and powerful, malware are also being designed in ways that will evade detection by antivirus, be able to conceal self longer within the host for survivability, and this is achieved by packing or compressing malware code,avoiding execution when they are being monitored, and more aggressive malware will directly disrupt the functionality of the antivirus. (Hsu et al, 2012).

According to Borkar and Kumari (2017), it is a challenge to recognize an attack in real time using only techniques such as IPS and other IDS, without the presence of an Admin; these attacks can be from external intruders or internal. Antivirus on the other hand are the most commonly used software for providing computer security, however penetration testers and attackers will always find ways using special tools that will encrypt malware to bypass such protection mechanisms (Ali and Hameed, 2019).

With these conventional techniques, there is deficiencies in security logging and analysis that allow attackers to hide their location, default configurations for network infrastructure devices, use of default passwords on devices, failure by Organisations to scan vulnerabilities and proactively address discovered flaws, according to a survey on SACCOs conducted by Serianu(2018). And this is where the Honeypot prototype comes in as it is an intentionally vulnerable system that would stand out for the attacker to attack and/or help capture attacks that have successfully passed through the existing IPS/IDS.

2.5 Honeypot technique for malware analysis

Generally, honeypots consist of data that appears to be a legitimate part to a site, which may contain information or valuable resources to an attacker (Vadaviya et al, 2019). These honeypots are again of different types, depending on the level of interaction, and purpose of interaction.

Since traditional defence methods use signature based techniques for malware identification, it also becomes hard for them to detect previously unknown malicious executables (Talukder, 2020). The researcher through use of questionnaire, found out that generally, what is mostly used are antivirus which may not necessarily be the professional version, firewalls which come free with ISP router device, and many a times these are on default settings, other equipment on the network that may be used for hardening security such as switches and some routers offered by ISPs are not managed meaning they are on default settings as well, posing further cyber-risk to an enterprise.

While other SACCOs may have professional security solutions such as the antivirus and firewalls, at times they generate quite a huge amount of traffic in logs that may be hard to go through, unless a Cyberattack or an adverse event has taken place and discovered.

2.6 Honeypots classifications

There are quite a number of honeypot variations, pegged on the level of interaction and the purpose. The interaction is of three types of levels: (Vadaviya et al, 2019).

2.6.1 Low-Interaction

This is where one or any simple services that record every communications attempts targeting a specific service on the network, e.g web or SSH. These are basically simple daemons that give passive methods of monitoring attempted attacks. These kind of honeypots are often safe to run due to their level of interaction but are not preferred in complex and more interactive environments such as in the SACCO industry.

2.6.2 Medium-Interaction

These kind of honeypots attempt to provide more compelling information to would be attackers as they emulate a collection of software, whilst protecting the operating system of the host. The emulation of software responds as the actual software would respond and this may be complicated in terms of security as there should be no same security issues. These kind of honeypot, due to its nature, increases risk of corruption of the system as it has multiple attack surface points for would-

be attackers or malicious users. This is the kind of Honeypot that this research intends to design and employ and test on reliability using secure development lifecycle, whilst borrowing from open-source tools readily available.

2.6.3 High-Interaction

This is whereby the entire operating system used by the host, with live instances of programs running is offered to the attacker, so as to offer large interaction with actual instances and not just an emulation. While this level of Honeypot may offer the best opportunity for learning an attacker's moves and behaviour after privilege escalation, it is the riskiest as it is quite possible for an attacker to take full control and even jump to other systems in the network.

2.7 Malware Detection and Analysis Tools

According to Aslan and Samet (2017), there are several tools used to detect and analyze malware, some of them include Process Explorer, PEiD, UPX, IDA, BinText among others. These mostly use techniques such as pattern matching and malware signature, and as a point of concern, some of these tools will need much human intervention in terms of manpower to detect malware, while others such as antivirus software may not need intervention by humans but are rather reactive.

Table 2.1 describes briefly more tools used for detection and analysis of malware (Aslan and Samet, 2017).

Name of Tool	**Description**
PEiD	Tool for detecting the packed, obfuscated malware that are in PE format.
PE Explorer	Tool to display the structure and content of the PE. Additionally, it can be used as an unpacker for packed files.
BinText	Tool that is capable of searching and displaying the character strings from a binary file.
UPX	Known as Ultimate Packer for Executables. Is a tool for packing common PE, which is used to compress malware samples, as it is difficult to detect and analyze packed malware without unpacking them.
Resource Hacker	Tool for viewing, modifying, adding, and extracting resources from PE.
IDA Pro	Interactive disassembler professional, which is widely used by malware analysts, reverse engineers, and vulnerability analysts.

Process Explorer	Tool is similar to task manager, which shows running processes in detail.
Process Monitor	Tool to display file activities, registry such as reads, writes and changes, other processes and network activities
Regshot	Tool to compare registry changes taken by two registry snapshots, to analyze the system for system changes.
Wireshark	Tool intercepts and logs network traffic.
WinDbg	Debugger that can debug in user-mode, kernel-mode, x86 and x64 malware.
OllyDbg	It is an x86 debugger that is widely used for binary code analysis when source code is not available.
Burp Suite	Software platform of tools for security testing of web applications. It can be used as a man-in-the-middle to modify http/https requests sent to a remote server by a malware

Table 2. 1: Some malware detection tools

The tools in table 2.1 work mainly by analysing malware either by static analysis or dynamic analysis (Sethia and Jeyasekar, 2019). Sethia and Jeyasekar further expound by adding on what these analyses methods are: Static analysis of malware is the technique of determining the nature of malware without executing it using tools such as IDA etc, while dynamic analysis involves determining the strings in the malware's binary, imported API, etc; the malware is detonated and its behavior studied to learn its functionality using tools such as the PE Explorer, Wireshark etc. This leads to one of the major challenges such as the need to have a myriad of tools to handle the malware from the point they are detected (if they get detected), and also that one has to have great understanding of malware to perform analysis when dealing with them (Sethia and Jeyasekar, 2019). At the same time, administrators might forget to update the firewall rules/policies, while IDS systems that use anomaly detection have been noted to have high false-positive ratio (Yeh and Yang, 2008).

2.8. Conceptual framework

A conceptual framework is a structure that the researcher uses to best explain the natural progression of the phenomenon being studied (Camp, 2001). There being different types of conceptual frameworks, the visual representation of this study is as figure 2.1:

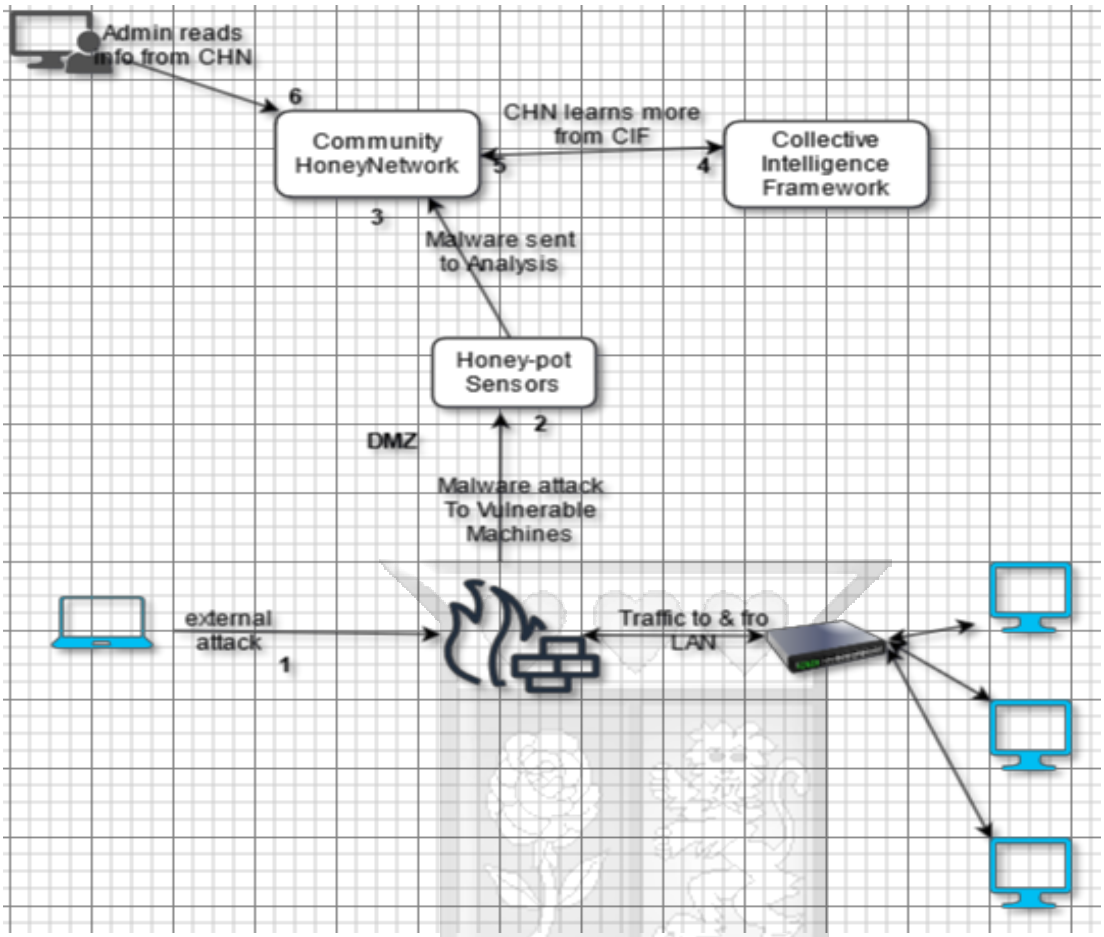


Figure 2. 1: Conceptual diagram

The Admin reads logs/alerts from the Community HoneyNet (CHN) at the other end; step 6, which is fed information from the Honeypot sensors and/or Collective Intelligence Framework (CIF) which are steps 3 and/or 4.

The Honeypot Sensors are customized with containers that are deliberately vulnerable to capture attacks.

When an attacker/malware in step 1 attempts to get into a SACCO's network, if they are not stopped by the firewall, they are likely to proceed to the honeypot prototype in step 2, which has the obvious vulnerable systems.

The attack will be captured by the Honeypot sensors, that information and any malware will be analyzed and the analysis will be presented by the CHN in step 3, in step 4 and 5, the CIF module will share any intelligence with the output received by the Admin via the CHN module via a Web interface, where the admin will be able to login in step 6 to study the analysis results..

i) If placed outside the firewall or facing the internet, they will attract many attacks which may not be as helpful to a SACCO, as there are crawlers in the internet that continuously run looking for open ports.

ii) When the honeypot sensors are inside a SACCO's network, it means they will be able to capture attacks/malwares that have escaped or bypassed the firewall or malware that is emanating from within.

iii) Such a novel design will also be critical in capturing internal attacks.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

According to Walliman and Walliman (2011), Research involves activities that guide in a systematic way in discovering things one did not have prior knowledge of. Methodology on the other hand is a philosophical framework on which the research is done (Brown, 2006). This chapter gives a description of the research design, software methodology adopted, data collection and sampling population, analysis of the data collected, and ethical considerations. A systematic

literature review process was adopted in answering objective 1 and 2, whilst Rapid Application Development methodology was used to answer research objective 3 and 4.

3.2 Systematic Literature Review Methodology

According to Godwin (2016), Systematic literature review is a methodology that provides ways to critically appraise and make sense of large bodies of literature to inform a study. According to Petticrew and Roberts (2008), Systematic reviews are also used to highlight gaps and/or concepts that are accepted as true with little evidence. The systematic literature review methodology was used by the study to guide the researcher in preparing objectives 1 and 2.

Borrego et al (2014), bring out 4 steps involved when doing a systematic literature review:

i) Determining an inclusion and exclusion criteria

ii) Searching and cataloging data sources

iii) Critiquing and appraising

iv) Synthesizing

3.2.1 Inclusion and exclusion criteria

According to Meline (2006), this criteria should be applied liberally, as this set limits to what research material such as books, journal papers, conference proceedings etc that will be included in the literature review. Therefore, as the terms suggests, inclusion criteria are the aspects that determine what to be included as essential to a study, while exclusion criteria determine what is to be excluded from the study, before gathering sources so as to reduce any potential biases (Godwin, 2016). One of the inclusion and exclusion criteria this study employed was limiting research artifacts to those that are from 2016 to date, and occasionally citing older research artifacts when they were cited by study that falls within the set criteria.

3.2.2 Finding and cataloging sources

This study used keywords such as SACCOs, Malware analysis, Honeypots and Cyberattacks to search databases and extract all sources relevant to this research. This helped this research find every research that would guide in answering the research questions. Cataloguing of sources means that the search results that were gathered were exported into a format that is usable and cleaned to remove possible duplications of entries.

3.2.3 Critiquing and Appraisal

This employs the inclusion and exclusion criteria to critique sources of data collected alongside their descriptions of the studies and measuring the quality of the study as well. This helped the researcher avoid predatory journals and possibly rogue publishers.

3.2.4 Synthesizing of included artifacts

This step means pooling together findings from all research artifacts gathered to provide outcomes of what the current body of literature has found vis-a-vis gaps that remain in our current understanding (Godwin, 2016). The researcher was able to get divergent views of the same topic which was used in converging other studies that helped the researcher cover objective 1 and 2, objectively and exhaustively.

3.3 Rapid Application Development Methodology

For objectives 3 and 4, this research applied a Rapid Application Development methodology to develop the honeypot prototype for malware analysis in the SACCO industry, and validating the prototype. This is so as to ensure security is by design, by enshrining security in every stage of the system development for a better end-product that's secure and reliable. This is so, since traditional development life cycles do not take into account security concerns in particular (Matulevicius, 2017).

Therefore, Rapid Application Development (RAD) approach, which comprises a set of activities that was performed to develop and deliver a secure honeypot for malware analysis prototype, as an unnoticed software security lapse during the early phases of the life cycle is inherited by the later stages (Daud, 2010). This approach helped as classic software development models adopt a reactive approach since tasks related to security are often relegated to the final phases of software life cycle, (Nunez et al, 2020).

RAD is development of designed cycles which provide results much faster and of higher quality as compared to traditional lifecycle models (Coleman and Verbruggen, 1998). According to Busro and Rahim (2029), RAD consists of various combined structured techniques to accelerate development of systems and have four stages: Requirements which is also the Planning phase, Design phase, Construction/Development phase and the Cut-over phase.

A brief characteristic of the stages are as shown in figure 3.1:

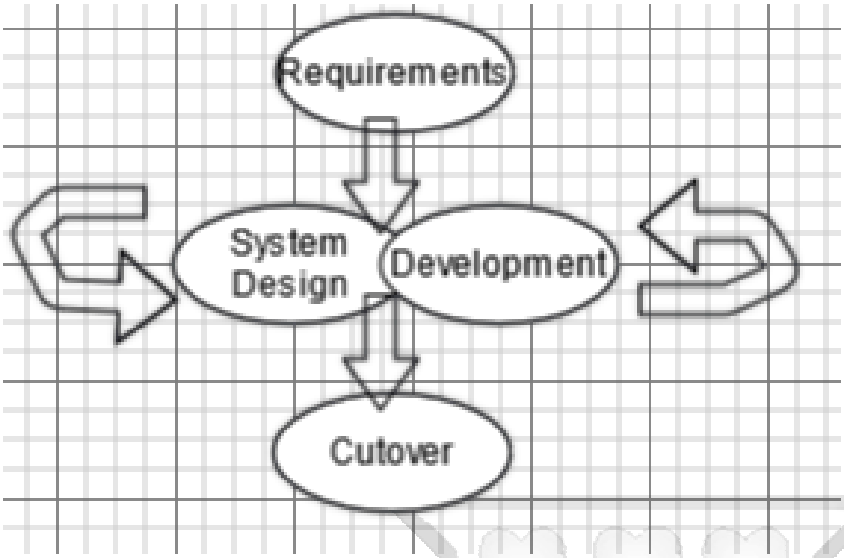


Figure 3. 1: Rapid Application Development

3.3.1 Requirement Phase

The researcher collected data through use of questionnaires so as to determine the business needs, the expectations of the prototype which eventually guided the scope for better planning of the prototype. For instance, SACCOs were generally noted not to employ use of in-depth approach to security, did not have honeypots, did not have well put systems that were automated in terms of giving alerts on malware attacks on their environment further cementing the need for such a novel system as this which is relatively new in the industry.

3.3.2 Design Phase

This stage establishes design requirements and analyses attack surface. Various iterations were used in the development of the prototype as would be when designing software that is customizable, for the purpose of being able to test every stage so as to ensure they meet expectations. Use case and Data flow diagrams were employed to depict how actors interacted with the honeypot prototype for malware analysis. Diagrams herein were drawn using diagrams.net which is an open source platform that allows drawing via web browsers.

3.3.3 Development Phase

In this phase, the researcher built the application by implementing results from the requirements and design phases into the applications and modules through writing of code, integration of

modules and testing which was required for the honeypot base tool for malware analysis which was the output after completion, having ensured that user requirements were met. The resources used in this phase was a laptop and several virtual machines that acted as the Honeypot sensors, and a Community Honeynet Network server. The honeypot components were dockered so it could support several honeypots and tools on the network with a small footprint while ensuring all the honeypots are restrained within their own environments. The researcher employed digitalocean.com as a cloud service provider for their robust Platform and Infrastructure as a Service.

3.3.4 Cutover Phase

In this phase, the honeypot based malware analysis prototype was checked to confirm all the necessary modules were present and running as expected and could attract and capture malware, this is so as to avoid access to the enterprise infrastructure by intruders. The idea is to expose services from the honeypot as much as possible, whilst hiding the live system (Gjermundrod & Dionysiou, 2015). The honeypot prototype was let run and attacks/malware captured. The researcher also used commonly known tools such as Metasploit used in Kali Linux OS to test and this helped prove the efficacy of the honeypot prototype.

This included examining the prototype's output, and performance against quality defined in the requirements phases. This was done so that the researcher had ample time for remediation, with focus on ensuring security before release.

3.4 Research Quality

3.4.1 Reliability

The Reliability for this research was through the administering of questionnaires that was shared online. Well administered questionnaires as a tool for research are known to ensure quality of the research through their findings.

3.4.2 Validity

The respondents were able to review their filled questionnaires to check for errors and give more comments. With the filled questionnaires, the researcher was able to confirm that analysis is evidence based.

3.5 Ethical Considerations

For this study, the researcher got approval and thus consent from the respondents as matters cyber security are highly confidential, and would pose potential threat if in the wrong hands. The online questionnaires had disclaimers and reminders that confidentiality will be kept at all times and that it is for study purposes only. There is acknowledgement of authors and all work that has been cited thus far.

CHAPTER 4: SYSTEM DESIGN AND ARCHITECTURE

4.1 Introduction

This chapter expresses in a detailed manner the Honeypot used for malware analysis that will boost a SACCO's cyber security preparedness, its general setup and the architecture employed to make the prototype ready in an enterprise network, and the services offered by this Honeypot prototype given in detail. The chapter will also discuss function and non functional requirements needed by the system, the architecture and design of the prototype that will include diagrams as well.

4.2 System Requirements

Briefly this includes a critical examination of the problem or gap that this study intended to bridge by guiding in the gathering of information of systems that do exist while determining requirements that makes the Honeypot based malware analysis tool better, by enhancing the SACCOs' cyber security posture.

The Honeypot prototype gets to be described in terms of its features, services it offers and limitations that may be inherent. These are well described in the two sections below; Functional requirements and the Non-functional requirements.

4.2.1 functional requirements

These are requirements that state the basic facilities that should be offered by the system

i. The prototype should be deployable on the LAN/seperate VLAN within SACCO Network
ii. The system should not be etectable as a honeypot or deception system on the network.
iii. The system should have a GUI that displays attacks; live or past..
iv. The system should be able to give reports on details of attacks taking place or those that have been captured by the honeypot prototype.

4.2.3 Non-functional analysis/Requirements

These are those requirements that dictate the quality constraints that the honeypot prototype must satisfy.

i. Availability: The proposed system should be up at all times
ii. Security: Compromise on the Honeypot prototype should not spill to the live system or other parts of the network/LAN.

iii. Capacity: The system should be able to withstand multiple attacks from different fronts concurrently; bearing in mind it is hosting a myriad of honeypots in an "all-in-one" framework.

iv. Performance: Due to the nature of the Honeypot, it can be quite demanding as a result of giving live updates, reporting, while keeping malware busy while analysis is happening.

4.3 System Architecture

The Honeypot placement was inside the network so as to capture attacks that have passed existing firewall and antimalware systems. Such a setup also helps learn and analyse attacks that are within the SACCO's network.

The outline in figure 4.1 represents the blueprint of the Honeypot System.

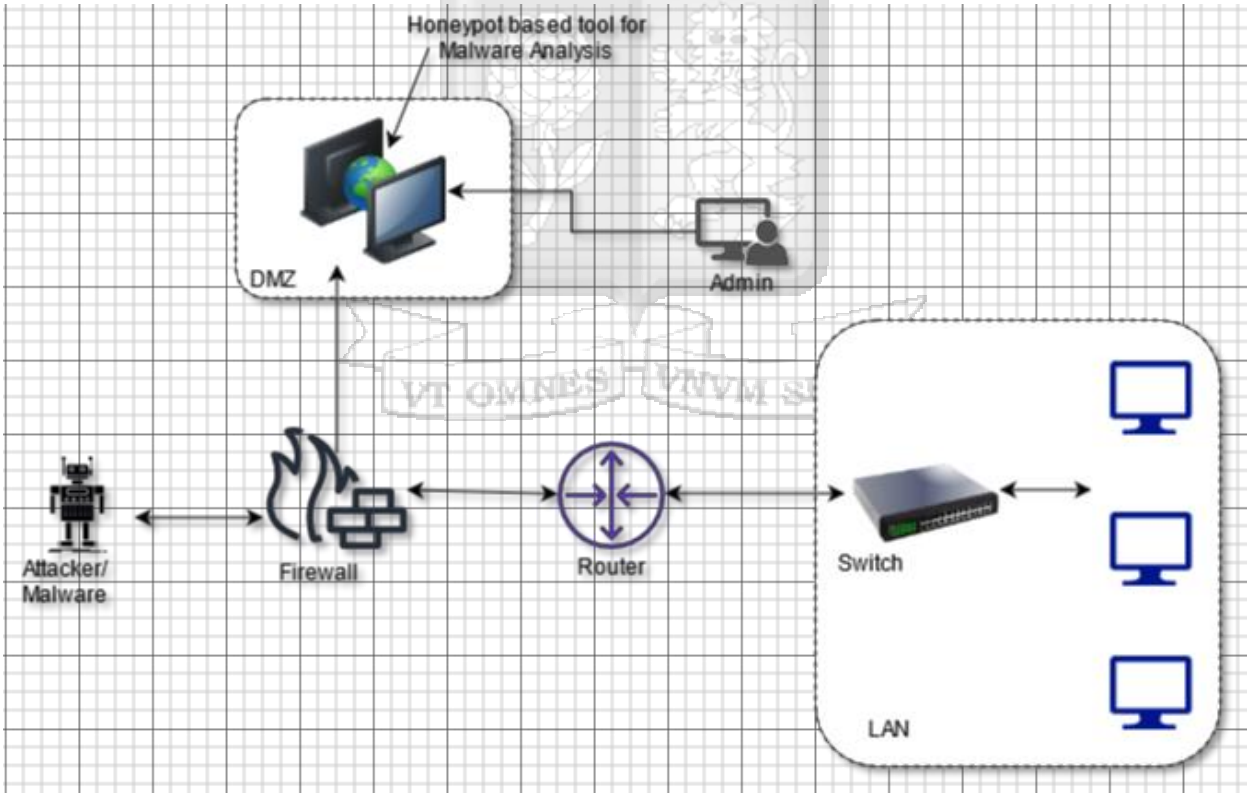


Figure 4. 1: The Honeypot based tool inside a Network

The designed honeypot was largely based on a framework that borrows from the community honeynet, whilst it is customized to suite the SACCO community. This prototype offers a multi-honeypot architecture which have been dockerized. An example of a honeypots employed is: Dionaea, other honeypots can always be added if need be.

Furtherance to this, The honeypot design uses CHN Server that displays via a web console attacks taking place in real-time. This was connected to a Collective Intelligence framework that is feeding it (CHN) with intelligence or rather analysis of malware encountered.

This setup was done in a way such that it runs in the DMZ area of the network for the particular SACCO where it is deployed.

4.3.1 Inputs

The honeypot prototype design was based on containerization which essentially is operating system level virtualization, as containers package applications and their dependencies together in an isolated virtual environment. The researcher used the Dionaea framework which is a low interaction honeypot that logs connection details in JSON format. The Dionaea honeypot framework, was initially developed under The Honeynet Project's 2009 Google Summer of Code (GSoC) as a low interaction honeypot (Sethia and Jeyasekar, 2019). The Honeypot prototype consists of the modules as seen in figure 5.1. The CHN Server is the notification module which is the point where the admin interacts with the system to learn of the analysis of malware captured. The CHN is fed from information gathered through the honeypot sensors which processes traffic, whilst the CIF was used to enrich the CHN with further intelligence on malware.

4.3.2 Key processes

The Honeypot has a component that grants the intruder/attacker access to the machine without them suspecting, this is done through an entry point that may mimic any resource in an enterprise addresses that have data made public in any way so that the attackers can find this information and use such vulnerabilities to send malicious applications or code (Polyakov and Lapin, 2018).

The honeypot prototypes simulates vulnerable services such as FTP, SMB, MS SQL, HTTP, SSH etc such that when an attacker exploits one of the services, her activities are captured by the honeypot sensor, which feeds the CHN Server for presentation. This was achieved through the following:

The Management console

Which helped the Admin determine parameters they needed information on, via a web browser such as information on attacks and payloads etc.

IP address binding module

This module is used to help bind the honeypot sensor to a particular IP such that the honeypot can have multiple templates bound to the same IP.

IP address tracing module

This helps the honeypot prototype get accurately information for the IP source countries.

The CHN server was fed information from the Honeypot VM and this information where need be was shared with the intelligence framework which used the platforms VirusTotal.

4.4 System Design Tools

System design is meant to capture elements of the honeypot prototype and in particular, the architecture, modules and components and data that goes through this system. The honeypot prototype used a virtualized system in order to overcome the limitation that is a complex setup for honeypots (Bove, 2018).

4.4.1 Flow chart Diagram

According to Veena et al, (2019), honeypot technologies are used by organisations to assist in arresting viruses, malware, or attackers, and act as an alarm system, which discover attempts to attack a network. The figure 4.2 depicts the order of events in the prototype; an attack/payload comes from a malicious actor, this is to the firewall, which has default settings in some cases, the firewall might be absent, and in events of an internal attack, the firewall may not see this. If traffic is found to contain malware, a Firewall might arrest it or miss it. If it fails to see the malware, it proceeds to the honeypot, which will capture the logs, analyse the logs for malware and then this is shared to the CHN which alerts the Admin of any malicious payloads and other information that may have been extracted such as IPs and source countries, etc. The Admin can also use an Intelligence Framework such as virus total to gather more intelligence on the malware.

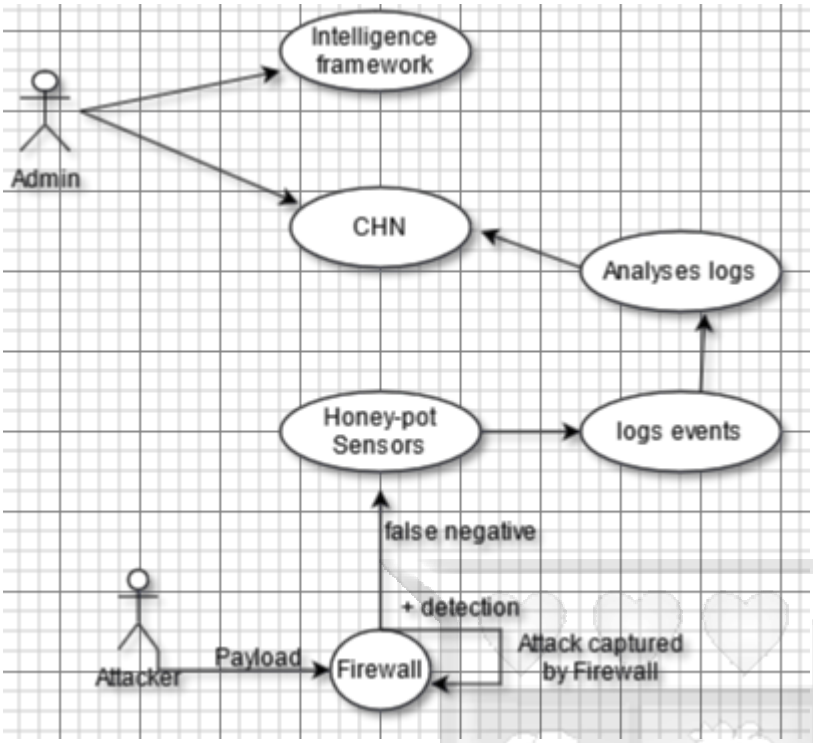


Figure 4. 2: Flow of events

4.4.2 Use Case Diagram

In Unified Modelling Language (UML) the use case diagrams are used to show interactions between the actors and the system. This prototype had 2 primary actors and a 3rd secondary actor. The 1st primary actor: the attacker who attacks the system, and the second primary actor: Admin who reads logs and analysis of malware, and the secondary actor is the intelligence framework which is used to give further intelligence on malware captured.

The attacker was considered a primary actor as she is responsible for feeding the system with malware albeit without their knowledge, while the Admin maintains the prototype as they gather more intelligence on the malware from the Intelligence framework.

Upon an attack and/or detection of malware on the honeypot prototype, the malware is gathered, and its features extracted. The Admin then login in the system, reads reports, consults the Intelligence framework and is responsible for maintaining the prototype as mentioned. malware.

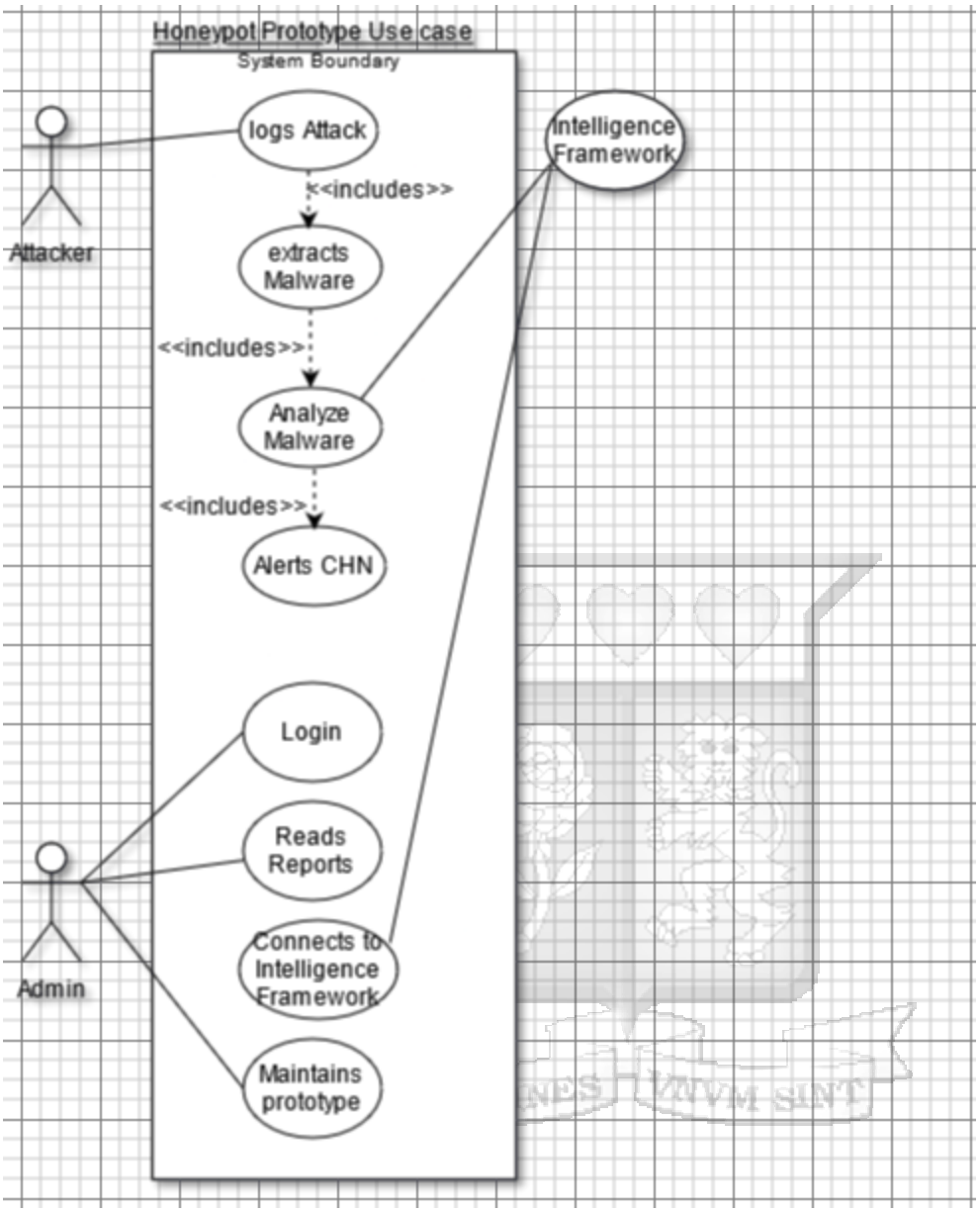


Figure 4. 3: Use case diagram

4.4.3 Sequence Diagram

The key purpose of sequence diagrams is that they show interaction between objects, and the sequence in which they occur. This show the activity flow in the Honeypot prototype to bring a better understanding as depicted in figure 4.5

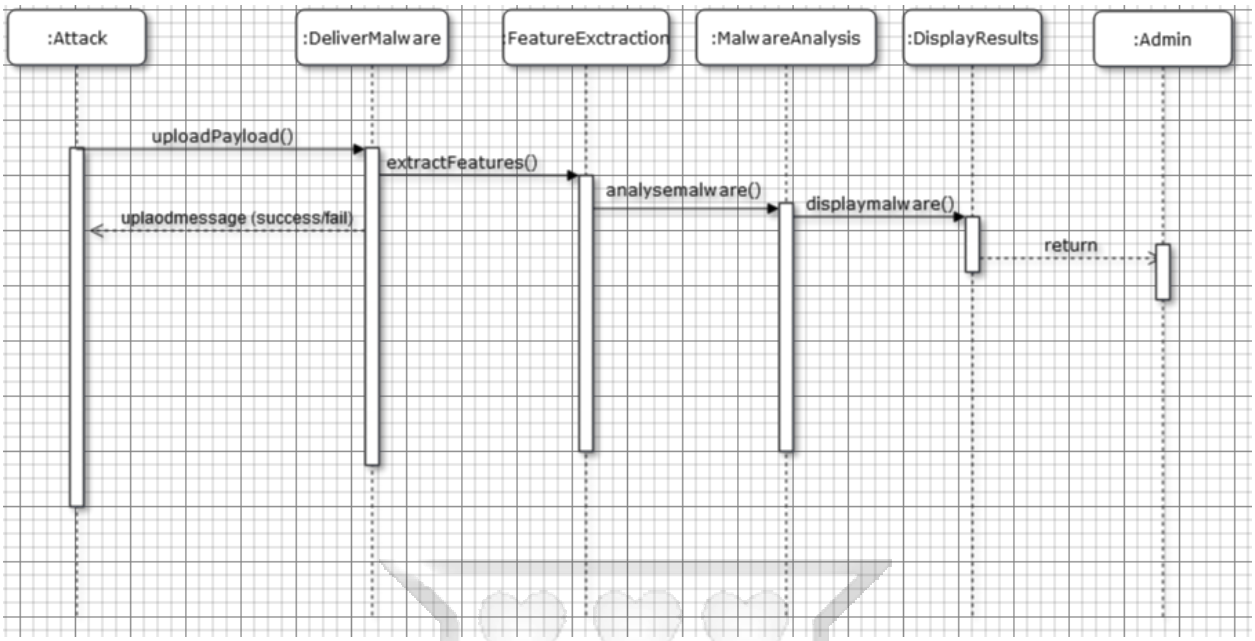


Figure 4. 4: Sequence diagram

4.5 User Interface

The Admin who is a primary user of the honeypot prototype accesses the system via a web console using an IP and secure login to the CHN server. Web login to allow accessibility across other available platforms had to be enabled in the CHN Ubuntu server command line interface/terminal.

4.6 Storage

The honeypot prototype was created on a system that had several Virtual Machines that had storage in the form of virtual hard disks. These stored the logs and allowed the prototype to run for sometime. The advantage with cloud infrastructure is the flexibility they offer in terms of either destroying/creating or quickly adding up more storage space for the prototype on the go.

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Introduction

This chapter focuses primarily on setting up and implementing the honeypot prototype, building up from the previous chapter, while giving more information on steps involved in the environment setup, and finally the prototype is tested and results are presented.

5.2 System Implementation

This honeypot prototype runs on standard Debian operating system, as is the preferred OS for running the Honeypots and the other interrelated modules. For purposes of this research, the prototype is setup in the cloud due to financial constraints but mostly for the flexibility cloud services bring, in terms of access and the portability they bring.

Server machines were created in the cloud that hosted the different services and containers. Once well installed and deployed, the honeypot prototype is able to run without interaction of the user. The honeypot deployed further used CHN to collect and dissect log files collected, the CIF also referred to as the Intelligence Framework in this research is also used to collect, merge and analyse logs and VirusTotal was primarily used as the source of intelligence for malware captured by the Honeypot prototype.

The components of the honeypot, hardware and software components used in this research work for the prototype to be setup and deployed are discussed as below:

5.2.1 Hardware Environment

The researcher used his own computer which had the specifications listed below. This are minimum specifications that had to be met so as to support the virtualized environment that may be used in a SACCO:

i) 8GB RAM
ii) AMD Ryzen 5, 2.0GHz (8 CPUs)
iii) 0.5 TB Solid-state drive storage
iv) Virtualization hardware

5.2.2 Software Environment

For the Honeypot prototype to be designed to reach an operational stage the below detail were present: The Admin had i to iii, and an attacker had iii and iv.

i) Windows 10 Professional x64
ii) Firefox 89.0.2, 64 bit, however any web browser could be used.
ii) Hyper-V Manager for Virtual Machines

iii) Kali Linux 2020 x64

iv) Metasploit framework tools

5.2.3 Cloud Environment

Due to financial constraints the research used an approach that employs Cloud infrastructure, which means paying for only what is being used, where one could build (and destroy) as many servers and or modules as they need. This also supported the iterative development and continuous improvement and makes the project quite portable as can be accessed from whichever location. Since the prototype could not be built in a live environment, the research relied heavily on the CSP's connectivity apparatus to cover for the networking equipment that would be present in a SACCO such Routers and switches.

The building blocks included dockerized Ubuntu 20.04 servers, which were used for the honeypot sensor and the community honeynet network server.

The computer system was a dockerized container hosted in the cloud. Two virtual machines were created one for the Honeypot sensor and another for the Community Honeynet Network. Each machine comprised of the below specifications.

i) 25 GB Hard disk

ii) 1 AMD vCPU

iii) 1 GB RAM

Using Digital Ocean as the Cloud Service Provider (CSP), the researcher created an account to login, and the diagrams below shows how to create the Servers referred to as *Droplets* by the CSP. The CSP offers virtually a myriad of cloud services such as Amazon Web Servvices or Microsoft's Azure offers, and it was the prerogative of the researcher to determine the services required for the research, considering cost implications as well

5.3 System modules

The honeypot sensors acted as the detection engine of the prototype, which share the information with the community honeynet network, and this is displayed as output which notified the admin of what was happening.

The researcher was interested mostly with the traffic that was potential malware. On getting the malware features the researcher used VirusTotal engine as the intelligence framework to get in-depth advise of the malware.

5.3.1 Creating the Honeypot Sensors

Through the detection engine of the prototype the researcher was keen to capture malware and their Md5 hash function which facilitated further analysis of the malware.

The researcher used dockerized containers that supports hosting multiple honeypot types which would act as the sensors, for the purpose of this study, as a proof of concept, the researcher used Dionaea honeypot as its main aim is to attract and capture malware that exploit vulnerabilities that is uncovered by services which are presented on a network (Bhagat and Arora,n.d). The Intelligence Framework added insights with confidence on what each attacking IP is known for, from which the Admin can choose what to monitor. The Admin logged in the CHN server via a web console to learn of ongoing attacks.

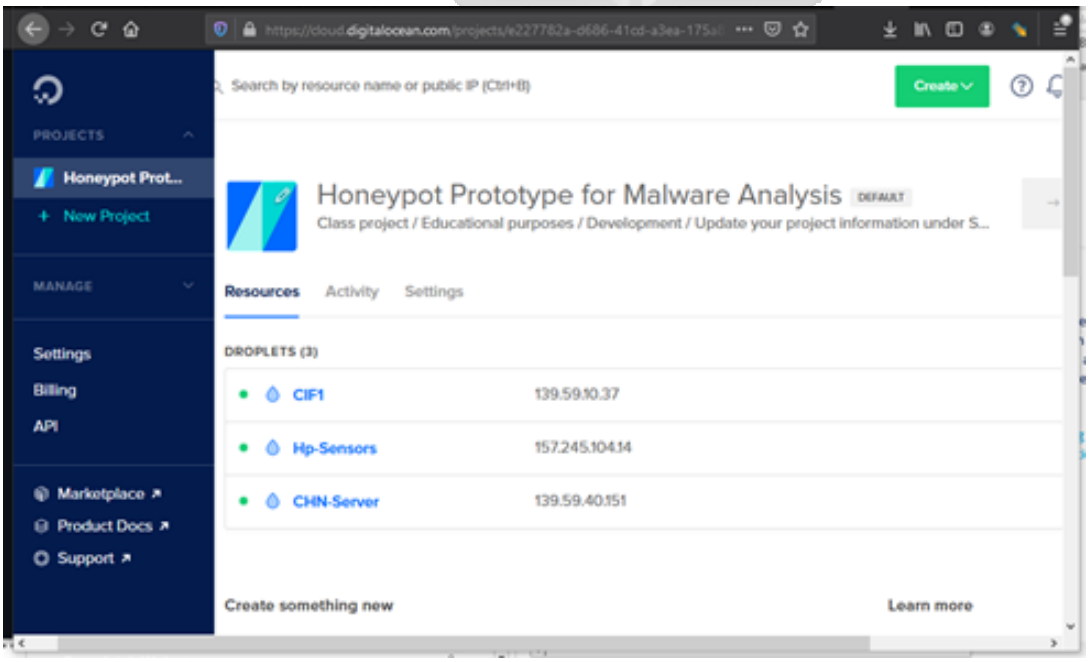


Figure 5. 1: Virtual machines created on cloud

Figure 5.1 shows the login page on the Digital Ocean Cloud platform, displaying created virtual machines, while the initial login in to the servers for configuration that are running on Ubuntu is via command line is seen in figure 5.2.

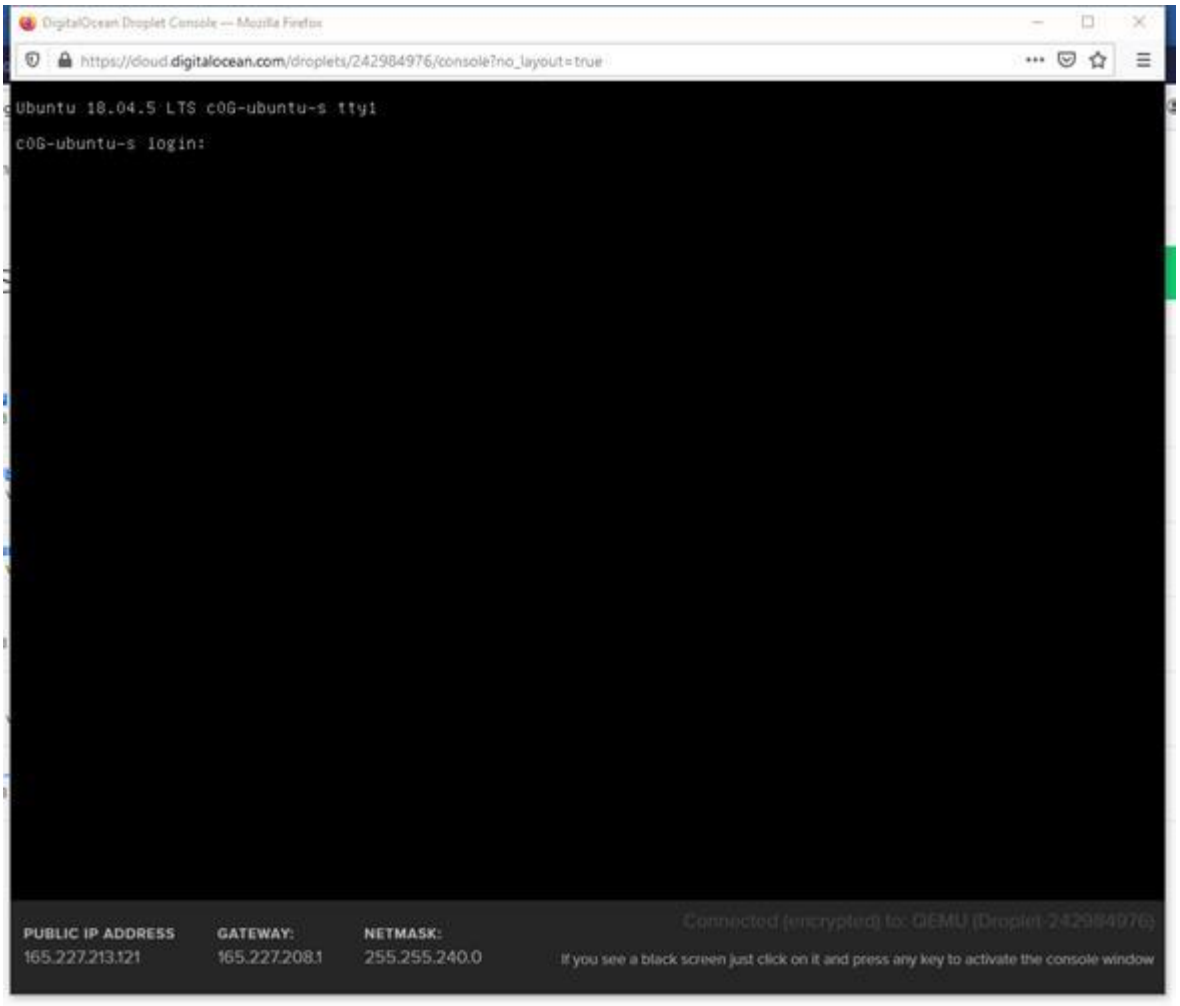


Figure 5. 2: Console login to honeypot server

On logon, the research first updated and upgraded the Ubuntu 18, so that it is up to date. This is seen in figure 5.3.

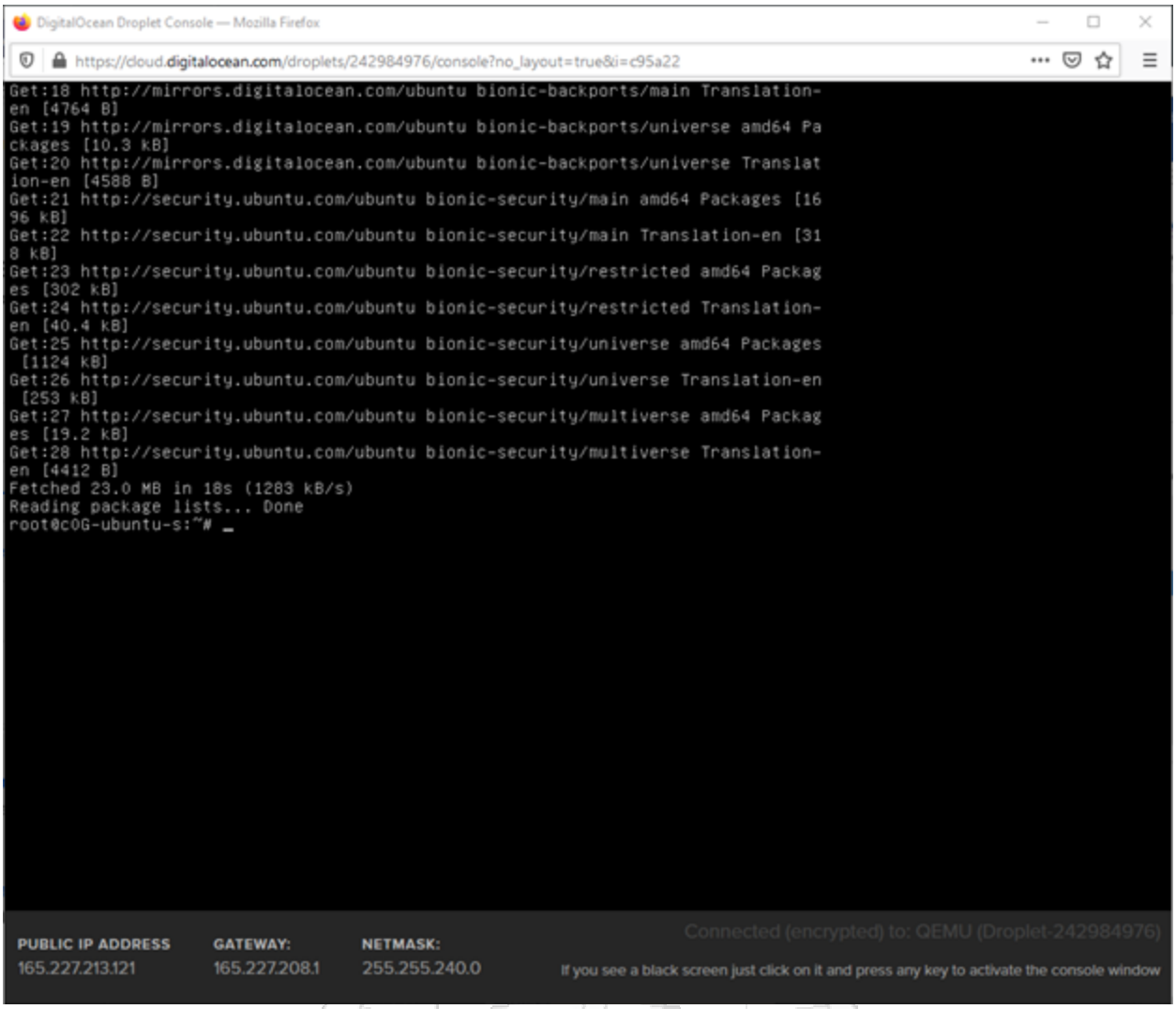


Figure 5. 3: Updating server

This was followed by downloading and installing docker as the researcher used dockerized containers that supports hosting multiple honeypot types which acted as the sensors for the honeypot prototype system for the SACCO community.

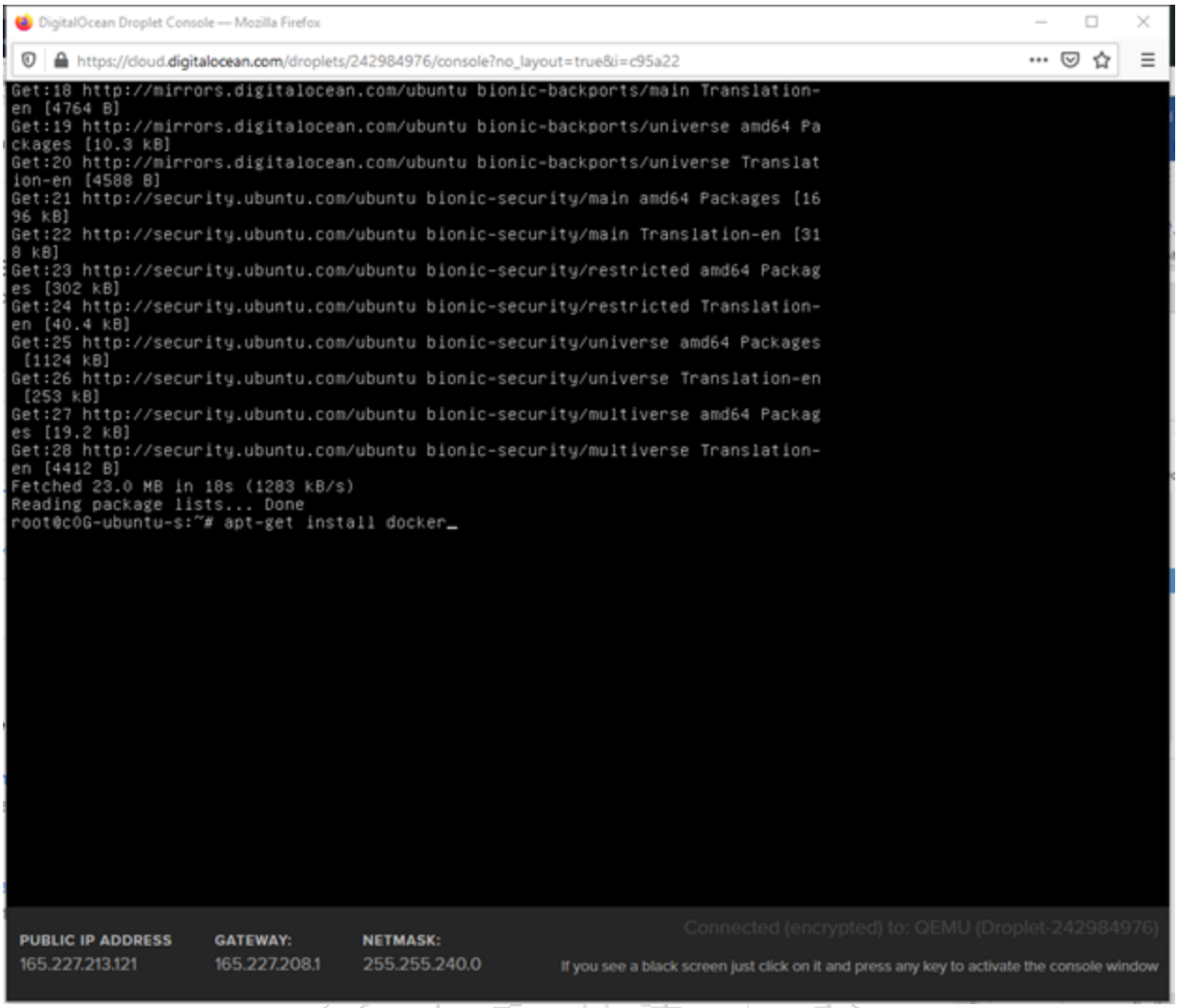


Figure 5. 4: Installing docker and docker-compose

5.3.2 Deploying the Sensors:

The honeypot servers have to be made vulnerable to would-be attackers and this is by setting up honeypot sensors that would make the system attractive to attackers and other threat agents. The sensor is mapped with the IP address to the CHN server. This is shown in figure 5.5 and figure 5.6 where the honeypot directory/files are created and configured in the CHN server.

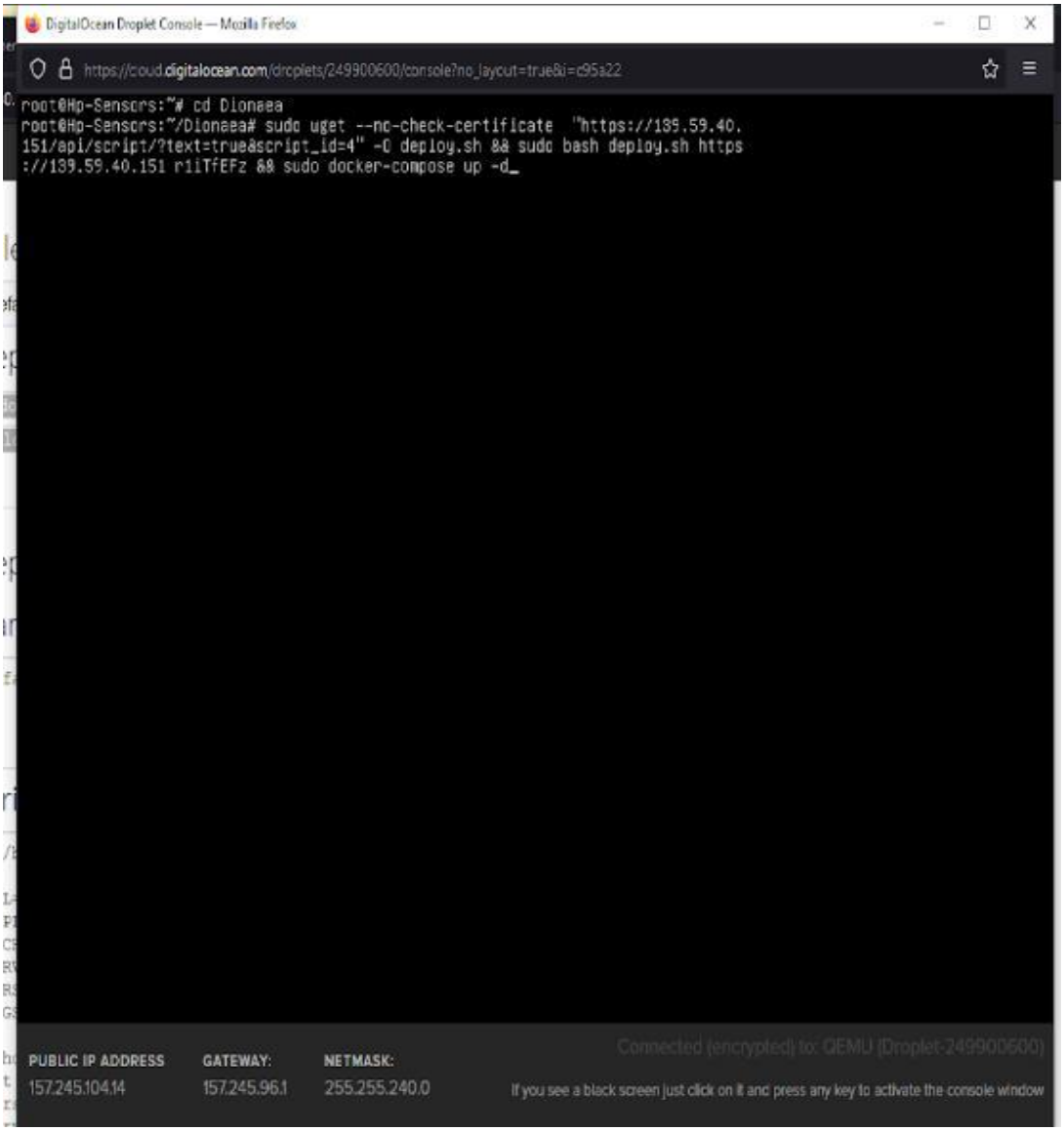


Figure 5. 5: Installing the Dionaea sensor

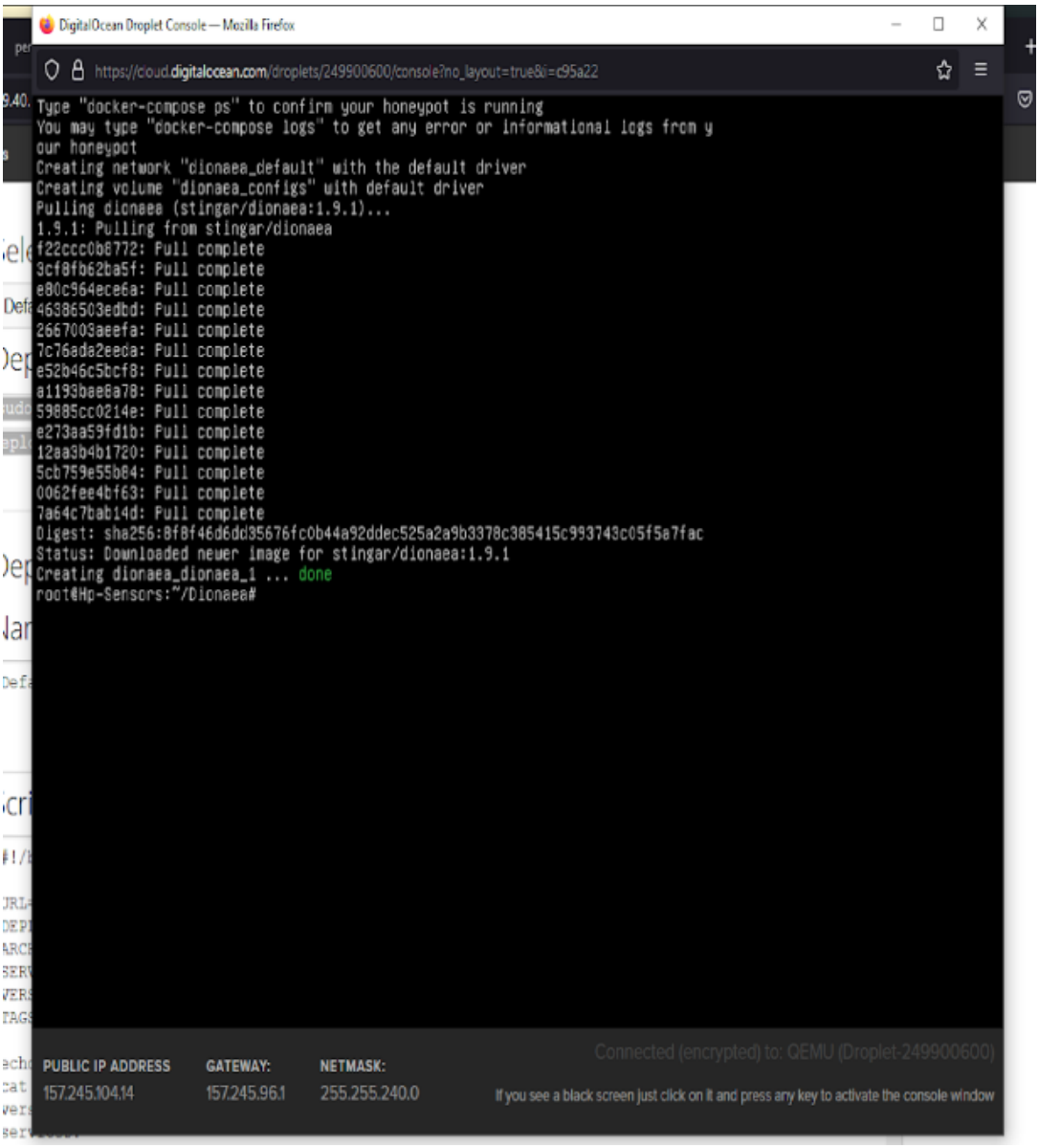


Figure 5. 6: Finalizing Sensor installation.

Once the sensor has been setup and configured, the researcher confirms it is up by typing the command *docker-compose ps*, and figure 5.7 confirms the honeypot prototype to be up.

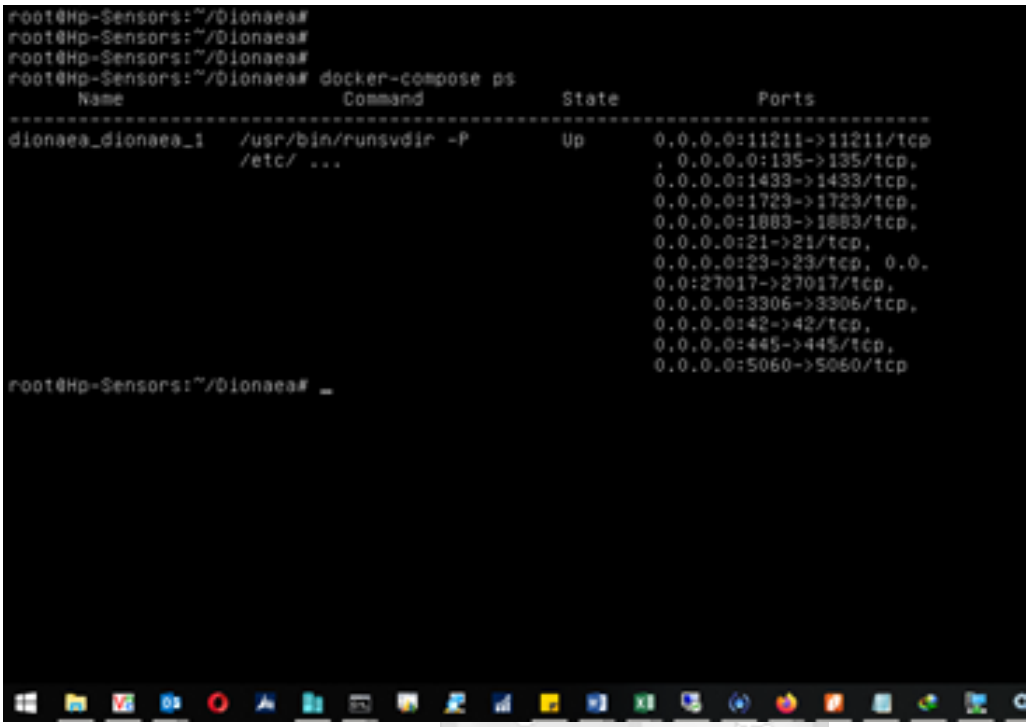


Figure 5.7: Checking to confirm honeypot sensor is up

5.3.3 CHN Server

The CHN server which has identical specifications to the Honeypot sensors, will go through the same process of OS updating and and installing docker and docker compose. Once the sensor(s) is up, the researcher enabled web login to the Community Honeynet Network Server, which gives a GUI that would enable Admins of all levels of proficiency to easily study and move around in the system. Command used to get the web browser login credentials: *grep SUPERUSER /opt/chnserver/config/sysconfig/chnserver.env*

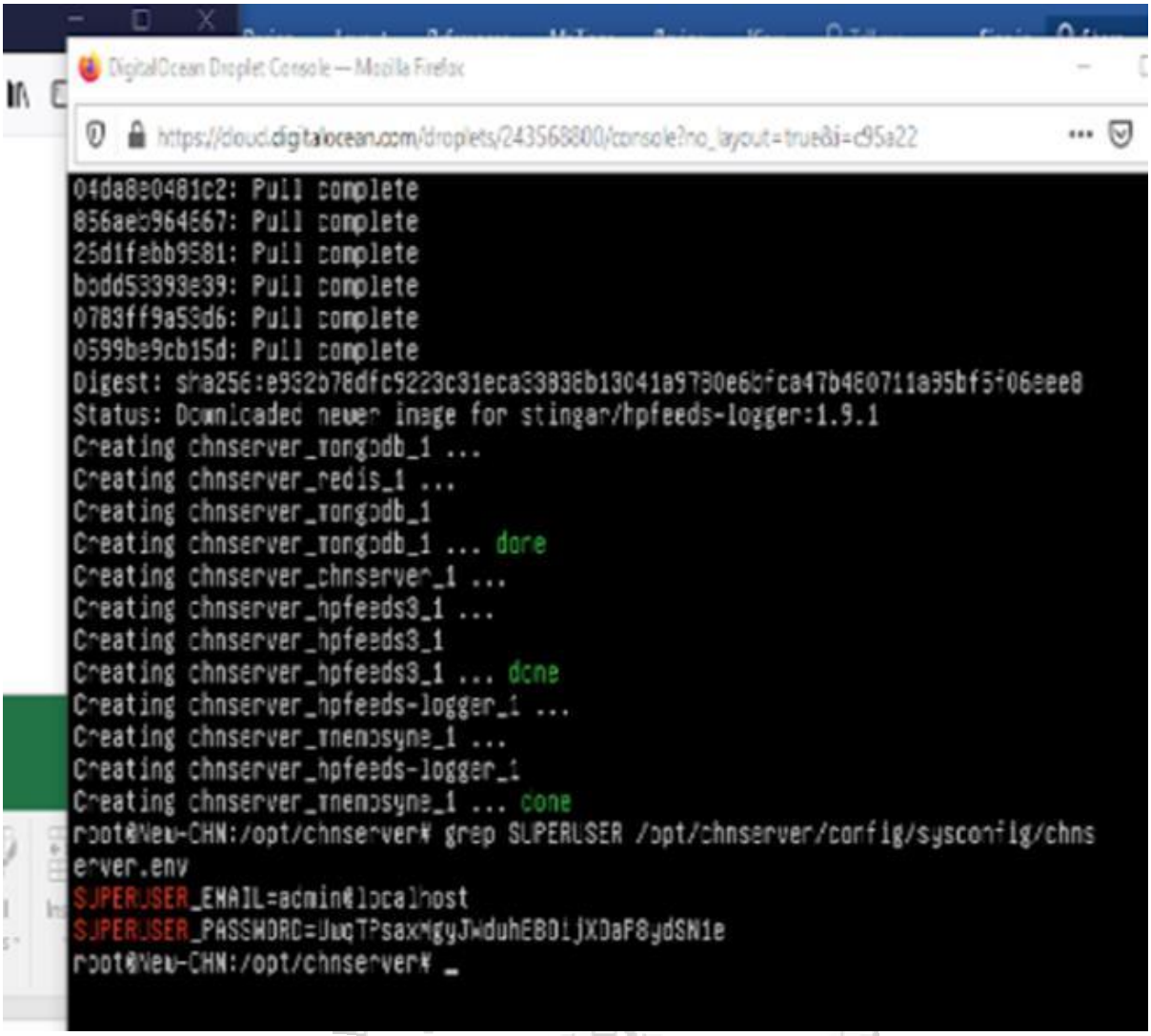


Figure 5. 7: Getting the login credentials for the CHN Server

Once login credentials are received as seen in Figure 5.7, the Admin is able to login using a GUI by typing the CHN server IP address on the web browser and proceed as shown in figure 5.8. The Admin can change the username and password after login via the web browser, or use the credentials issued.

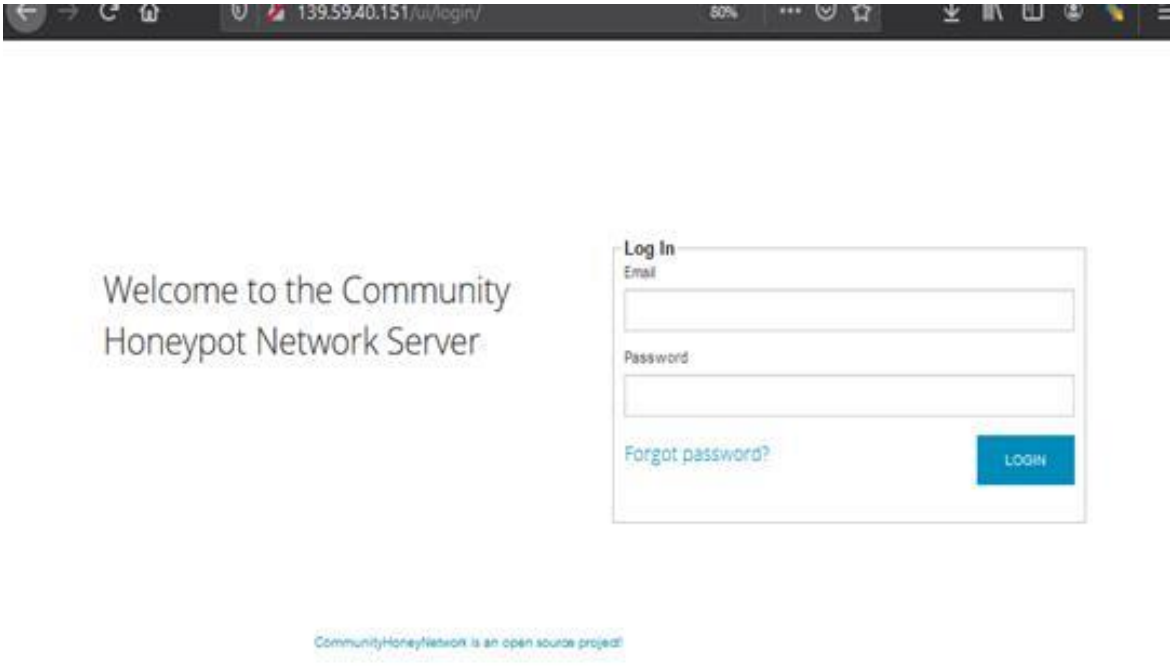


Figure 5. 8: Admin login page

5.3.4 Intelligence Framework

The Intelligence Framework added insights with confidence on what each attacking IP and malware where captured is known for.

Figure 5. 9: Admin login page

5.4 System Testing

The system was tested and seen to capture attacks, giving information such as source IPs and number of attacks and the ports attacked and Md5 hashes that made it possible for the intelligence framework to give further analysis of malware.

The honeypot prototype was made vulnerable by having open ports such as SSH which is commonly used in SACCOs to perform operations such as transferring data or doing other operations on remote computers. The honeypot has been running since April 2021.

The system was seen to work as demonstrated in figure 5.10 where it gives alerts of source IPs and countries that the attacks originated from.

Figure 5. 10: Attack information captured

The system was confirmed to be accurate in the capture of attacks, those that had malware and those that did not have. The honeypot prototype gave an analysis of attacks/malware by giving further information such as source and destination IPs involved, and the md5 hashes of files that had malware.

A consultant was involved in attacking/sending malware to the prototype using the IP 105.163.40.177, and all 4 attempts were logged as seen in figure 5.10. As much as payloads from the IP was flagged as malware as seen in figure 5.12, the IP was not flagged as malicious as it has not involved in hacking before as seen in figure 5.13.

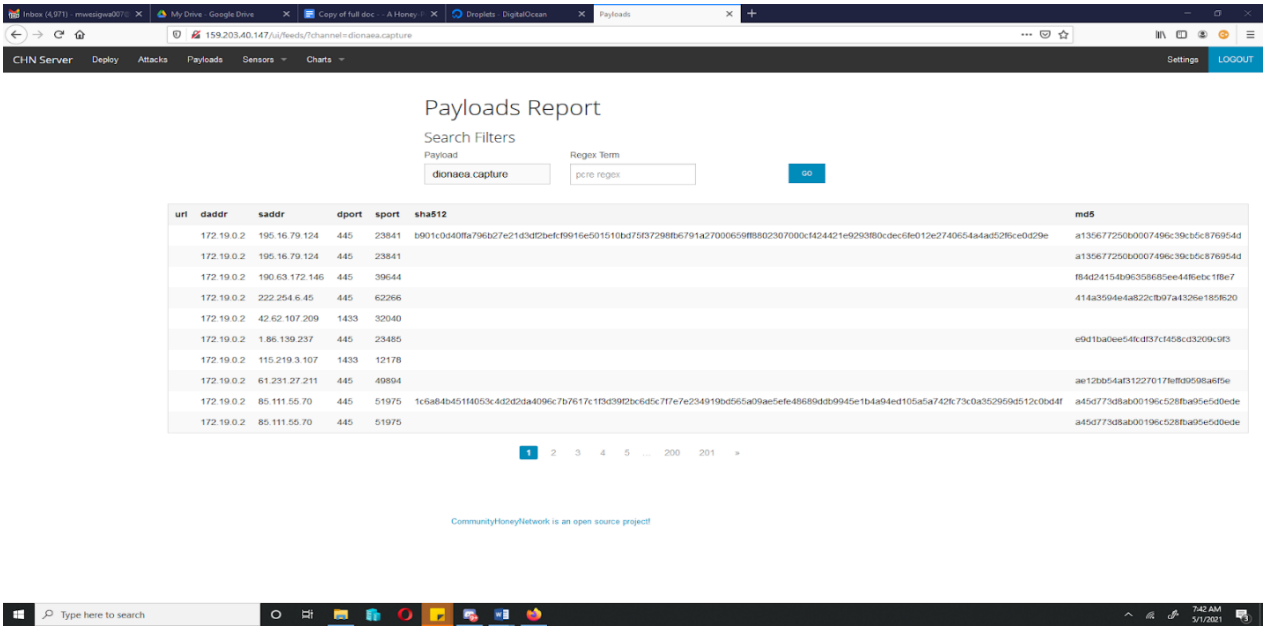


Figure 5. 11: Analysis of malware

Under the payloads tab, one can see the malware used which is already analysed and got the hash and md5 extracted. This is what gets run in VirusTotal to give further analysis of the malware as seen in Figure 5.12.

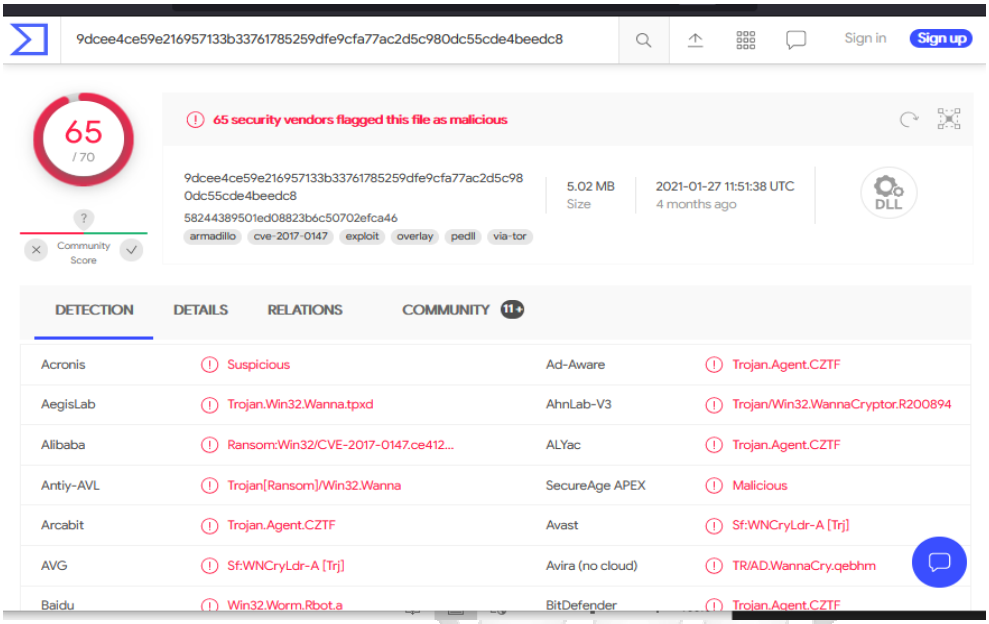


Figure 5. 12: Intelligence on malware captured

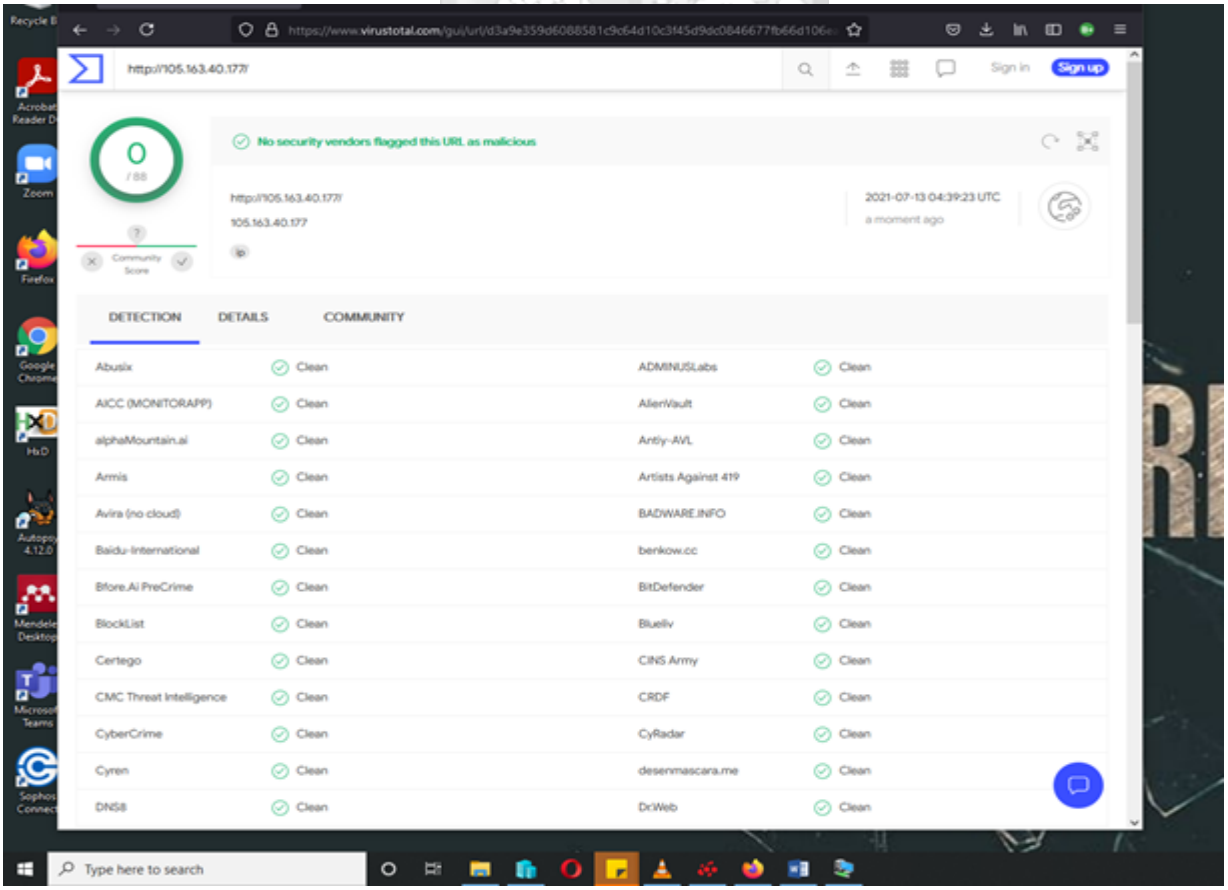


Figure 5. 13: Analysis results of IP

Further testing of the system was done to check if the prototype system meets the set requirements, both functional and non-functional.

5.5.1 Functional Testing

The system underwent a functional test to ascertain that all modules worked as expected, according to the system requirements. This testing was done on several use cases on the prototype to check if the results give the expected performance.

Test Case: Login/logout

Test case	Testing application Login/logout operation
Description	The Admin logs in successfully using a username and password
Use case	Login/logout
Results	Access to Dashboard is granted to Admin after successful login/logout
Pass/fail	**pass**

Table 5. 1: Login/logout test case

Test case	Check Honeypots presence
Description	Admin can access the different Honeypot sensors and see logs
Utilized use case	Check Honeypots
Results	Admin confirms Honeypots are up
Pass/fail	**pass**

Table 5. 2: Check Honeypots test case

Test case	Read Statistics from Honeypot sensors
Description	System is able to display malware/attack statistics on screen
Utilized use case	View statistics
results	User can view attacks and/or malware captured
Pass/fail	**pass**

Table 5. 3: Read statistics from Honeypot sensors

Test case	Extract Information for Malware Analysis
Description	Prototype can give actionable malware information for analysis
Utilized use case	View Malware information for Analysis
results	User can get actionable Malware Information
Pass/Fail	pass

Table 5. 4: Extract Malware information test case

Test case	Analyze Malware and display results
Description	System can advise Admin on Malware detail and type
Utilized use case	View malware analyses results
results	User is given exact information of Malware type for further action.
Pass/Fail	pass

Table 5. 5: Analyse Malware and display results

5.5.2 Non-functional testing

The developed prototype was further tested to check on the non-functional requirements and was found to meet the requirements. These requirements were on the availability of the prototype where it was noted to be up and running at all times, the prototype was separated from the live systems/network for security, and password protected, it was noted to be able to withstand multiple attacks and attack types at the same time, all this was while giving live data on the attacks/malware attacks. The hardware specifications were also performing as expected with glitches.

5.6 System Validation

Due to the nature of the study, and the strictness of financial institutions vis-a-vis cybersecurity, the prototype was not launched in a SACCO environment, but on the cloud for the proof of concept. Honeypots are designed to only attract malicious traffic that are attacks/malware thus this study assumes that all attacks targeting the SACCO where the honeypot is deployed, will be captured and on record, including 0-day attacks . From figure 5.11 which shows 30 attacks in the first 24 hours is a validation that the system works as it is meant to. Further, as demonstrated, a consultant helped conduct 4 attacks and all were captured, and logged giving an accuracy rate of 100% for the targeted attacks. However, within an active SACCO environment, attacks may be expected to be much less than that.

CHAPTER 6: DISCUSSION

6.1 Introduction

The researcher conducted this study with the aim of developing a honeypot prototype that would play a key role in the SACCO community in understanding their cybersecurity posture through getting to know the kind of attacks and malware that are targeted to them. The researcher learnt that this approach which not only adds a layer of security to SACCO's cyberdefense, it also goes a long way in helping them learn how to better secure themselves from common attacks/malware and even know how to invest better when it comes to cybersecurity. After data was collected from the SACCO's and doing an in depth research on literature related to this study, the expected outcome was achieved. The researcher used this chapter to give findings and interpretations of the study, which will be used to give conclusions and recommendations in the last chapter of the study.

6.2.1 Common malware targeting SACCOs

The researcher conducted extensive research on malware that are prevalent in SACCOs, and from findings, viruses, Trojan were common, however those that responded could not distinguish which type of virus or malware affected them. This prototype goes a long way in helping bridge this gap as the SACCO players will be able to become aware of the malware before they reach the live systems, and they will further get more information or intelligence on the malware from analysis that is done by the honeypot prototype.

6.2.2 Review of existing malware detection and analysis techniques and tools

Further, research was conducted to get to understand what the SACCOs are doing or had in place for malware detection and analysis in terms of techniques and tools available. What came out prominently is that SACCOs have at most two layers of security; just the antivirus, or an antivirus and a firewall, which many a times may not be as sufficient for financial institutions such as SACCOs. While the known attacker used tools such as the metasploit framework to do the attack on the honeypot system, it was logged and captures, this means that the analyses information would help the SACCO blacklist the IP involved with the attack, furthermore, the SACCO will be aware of the attacks before they get to the live environment meaning, an in-depth approach to cyber security using such a prototype is the way to go.

6.2.3 The design, development and testing of the Honeypot based prototype for malware analysis in SACCOs

The ideal placement of this prototype was within an actual SACCO setting as was the initial plan. Due to challenges such as the sensitivity of financial institutions vis-a-vis cybersecurity, possible risks, and authorization from SACCO Management/Boards, the study opted for an implementation strategy that was cloud based. The researcher purchased IaaS from the CSP Digital Oceans, created virtual machines and a firewall which was later turned off. These machines are what were developed and configured to act as the different components of the honeypot prototype. This made it simpler as opposed to using multiple physical machines, made it accessible from wherever the researcher was, without having to always seek permission or using a VPN to connect to a SACCO network. The honeypot prototype was tested by the number of attacks it captured and the analyses that it was able to do. Browser based platforms such as draw.io was used to design the DFDs, flow charts and sequence diagrams. Customization done is that this was not just an ordinary standalone honeypot prototype, as it employed use of a malware honeypot connected to a CommunityHoneyNetwork server and an intelligence framework, which further changes completely a SACCO's network infrastructure layout in their fight against cyberattacks and other malware.

6.2.4 Validation of the effectiveness of the honeypot based prototype

A total of 7897 attacks were captured by the honeypot prototype, with over 3000 payloads extracted from these attacks, of the payloads extracted IPs and MD5 hashes were extracted that were used by the intelligence framework to furnish the research with more info on the malware. To further validate this, a consultant did targeted attacks and all attempts were captured and payloads analysed and found to contain malware.

Therefore, these results show that the prototype achieved high accuracy rates even when under pressure of multiple attacks. The prototype performed well in capturing attacks and analysis of the malware through the feature extraction.

6.3 Conclusions

This research found that such a prototype goes a long way in capturing malwares, and by giving their analysis, that gives a SACCO the intelligence and work as an early warning system, thus they would be well prepared and equipped in the face of malware attacks.

Further to this, the researcher found out that of all the SACCOs that responded to the questionnaire or got interviewed, none had employed use of a Honeypot in their network or and this vindicated the basis for this research. While it is uncommon in the industry to include Honeypots in their infrastructure, the research brings in this novel idea of honeypots, considering they have been around for a while now.

According to Agnaou et al, (2017), it is evident that due to the fact that there is exponential increase in ICT infrastructure such as digital technologies, networks and other systems becoming even more vulnerable as this makes them exposed to more attacks, resulting to damages such as widespread breach to confidentiality, integrity and availability, this research proves an effective and primordial solution that gives insight in analysing techniques attackers deploy, learning the attackers' techniques and intentions using the attack types or malware deployed.

Benefits of the Honeypot Prototype System

1. A layer of security is added in a SACCOs infrastructure.
2. The prototype does not require huge budget allocations.
3. The prototype is relatively easy to implement.
4. The proposed prototype is easy to reconfigure and customize with different honeypot frameworks.
5. It is easy to access and read the logs even for non-technical resource persons.
6. Such a solution gives SACCOs a head-start in preventing specific attacks that may seem common. This means institutions cut on costs by acquisition of what is necessary in regard to cyber-security solutions.

Limitations of the developed Honeypot Prototype System

This study encountered some challenges which limited the researcher from getting the optimum or ideal results, nevertheless the minimum threshold was achieved. The first limitation was with the sample space, just a few SACCOs only in Nairobi, considering we have thousands across the

country, this was further more limiting as interviews and respondents were again from a handful of individuals from the SACCOs.

The other limitation was that with the nature of this research, even after guaranteeing confidentiality of the information gathered, IT Resource persons tend to be quite paranoid sharing such information that the researcher was requesting; and thus most of the target audience shunned away from participating, even after having a one on one dialogue.

CHAPTER 7: CONCLUSION AND RECOMMENDATIONS

7.0 Introduction

This chapter discusses what has been the conclusions in summary, and the recommendations of the research.

7.1 Conclusions

The general objective of this study was to develop a honeypot based malware analysis tool to detect and analyse malware targeting SACCOs. This was paramount in enabling SACCOs to enrich their existing cybersecurity infrastructure and provide an in-depth security posture. From review of literature related to this study and summary of responses gathered, it was evident that SACCO's suffer adversely from cyberattacks and malware that are embedded in these attacks which is attributed to factors that range from inability to get and read real-time logs on ongoing events/attacks, using default settings, little knowledge in cybersecurity matters, financial constraints that prevent acquisition of the right cybersecurity solutions among other possible challenges.

Results got from the respondents to the questionnaire and interviews indicate that such a solution is not currently being used in the SACCOs due to issues like perceived complexity of setting up the honeypot based tool within the the SACCO environment, lack of cybersecurity budgetary allocations impedes ICT departments and institutions from acquiring the right solutions to harden their cybersecurity posture, lack of training and capacity building may lead to the use of default configuration and the lack of technological know-how in reading logs and even identifying attacks in time, and even understanding the kind of malware that has been used which would go a long way in helping institutions stop them, reverse their effects, and making sure such an attack/malware does not happen again in the future.

Honeypot based tools for malware analysis are not always easily available for institutions that intend to use them, and this was brought about by the fact that, of all the ICT resource from SACCOs interviewed say they have not implemented it, and majority do not know how to or the type of honeypots that are available.

Therefore, this study concludes, from findings that the challenge SACCOs face in combating cyberattacks, and to improve this, the honeypot based prototype is a needed, and once adopted, may lead to a change in the cybersecurity landscape of finance institutions, especially SACCOs, adding a layer in their network infrastructure that has previously been non-existent, giving them an in-depth approach to cyber security in a safer and simpler way with the potential of saving them millions of shillings lost due to cyberattacks that bring malware in their payloads.

7.2 Recommendations

1. The prototype used just one honeypot framework, the Dionaea which is commonly used for capturing malware. More honeypot frameworks can be added to this solution so that it could capture a myriad of attacks or rather it could have more simulated vulnerabilities that would entice an attacker.
2. There needs to be more training on cybersecurity especially for top management and SACCO Boards. This will go a long way in giving insight that ICT Departments are not a cost centre, but the driver of businesses, ICT does not only support business, but is a part of the business. This will lead to organizations embracing cybersecurity efforts and everyone would be a player and left just to their ICT departments.
3. Since tools used herein for the honeypot based prototype are mostly open source, such can be made mandatory by the regulatory bodies such as SASRA, which will help SACCOs and the Regulator become more cognizant of malware attacks that are prevalent in the industry via a common information sharing platform; meaning a malware attack in one SACCO may not be able to get to another or other SACCOs.

7.4 Challenges encountered

It was a challenge to get more copies of malware for analysis as at times the modules could run out of space or other technical issues that would force the researcher to setup the system again. The issue of cost was also a challenge as the more modules and the longer prototype ran, the more it had financial implications on the researcher.

7.3 Future Work

The honeypot based tool for malware analysis prototype will always have room for improvement, especially with the ever dynamic cybersecurity threat-scape, ever growing need and consumption of technology. Some areas for exploration in the future may include and are not limited to the following:

1. More institutions can adopt honeypot based tools in their networks and have a sharing platform which would support access to malware attacks information and even SACCOs that do not have expertise to have such systems in the institutions can learn, and better prepare themselves in the face of adversity vis-a-vis cybersecurity attacks.
2. Honeypot based tools can be a key component integrated in SIEM tools, thus can be used with other systems like firewalls etc to further make the entire network more secure.
3. Further research can be done to give the honeypot based tools the scalability that is needed in the face of advanced threats such as the emergence of botnets, and other cyberattacks such as threat actors that have payloads designed for (a) specific organization(s).

References

A. Borkar, A. Donode and A. Kumari. (2017). *A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS).* International Conference on Inventive Computing and Informatics (ICICI), 2017, pp. 949-953, doi: 10.1109/ICICI.2017.8365277.

A. Godwin. (2016). *Visualizing systematic literature reviews to identify new areas of research.* IEEE Frontiers in Education Conference (FIE), 2016, pp. 1-8, doi: 10.1109/FIE.2016.7757690.

A. Sawant. *(*2018). *A Comparative Study of Different Intrusion Prevention Systems*. Fourth International Conference on Computing Communication Control and Automation *(ICCUBEA)*, Pune, India, 2018, pp. 1-5, doi: 10.1109/ICCUBEA.2018.8697500.

Adrian Pauna. (2012). *Improved Self Adaptive Honeypots Capable of Detecting Rootkit Malware*. 2012 9th International Conference on Communications (COMM).

B. G. Mawudor, M. Kim and M. Park. (2015). *Continuous monitoring methods as a mechanism for detection and mitigation of growing threats in banking security system.* 4th International Conference on Interactive Digital Media (ICIDM), Bandung, 2015, pp. 1-5, doi: 10.1109/IDM.2015.7516317.

Baskerville, Richard & Wang, Pengcheng. (2018). A THEORY OF DECEPTIVE CYBERSECURITY.

Bodmer, S., Kilger, M., Carpenter, G., & Jones, J. (2012). *Reverse deception: organized cyber threat counter-exploitation.* New York: McGraw Hill Professional.

Brown R. B. (2006). *Doing Your Dissertation in Business and Management: The Reality of Research and Writing.* Sage Publications.

Buller, D. B., & Burgoon, J. K. (1996). *Interpersonal Deception Theory*. Communication Theory, 6(3), 203-242.

C. R. Kothari, (2004). *Research Methodology.*

Camp, W. G. (2001). *Formulating and Evaluating Theoretical Frameworks for Career and Technical Education Research*. Journal of Vocational Educational Research, 26 (1), 27-39.

CommunityHoneyNetwork. (n.d). https://communityhoneynetwork.readthedocs.io/en/stable/

D. Deka, N. Sarma and N. J. Panicker. (2016*). Malware detection vectors and analysis techniques: A brief survey*. International Conference on Accessibility to Digital World (ICADW), Guwahati, 2016, pp. 81-85, doi: 10.1109/ICADW.2016.7942517.

D. Fraunholz, M. Zimmermann and H. D. Schotten, (2017). *An adaptive honeypot configuration, deployment and maintenance strategy*. 19th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea (South), 2017, pp. 53-57, doi: 10.23919/ICACT.2017.7890056.

D. K. Rahmatullah, S. M. Nasution and F. Azmi. 2016. *Implementation of low interaction web server honeypot using cubieboard*. International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, 2016, pp. 127-131, doi: 10.1109/ICCEREC.2016.7814970.

Dhruvi Vadaviya, Mahesh Panchal, Dr. Abdul Jhummarwala, Dr. M. B. Potdar. (2019). *Malware Detection Using Honeypot and Malware Prevention.* International Journal of Computer Engineering and Technology 10(6), 2019, pp. 1-9. http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=10&IType=6

Docker Documentation. (2017). *Docker Documentation*, [online] Retrieved from: https://docs.docker.com/.

F. Hsu, M. Wu, C. Tso, C. Hsu and C. Chen. (2012). *Antivirus Software Shield Against Antivirus Terminators*. IEEE Transactions on Information Forensics and Security, vol. 7, no. 5, pp. 1439-1447, Oct. 2012, doi: 10.1109/TIFS.2012.2206028.

Financial Sector Regulators. (2019). *The Kenya Financial Sector Stability Report* 2018, Issue No.10.

Gatuga, Kimotho, Kiptoo. (2014). *History and Organization of Cooperative Development and marketing sub-sector in Kenya*, Ministry of Industrialization and Enterprise Development.

Github, T3chn0m4g3. (2020). https://github.com/telekom-security/tpotce. Retrieved from github.com

Gjermundrod, H., & Dionysiou, I. (2015). *CloudHoneyCY - An Integrated Honeypot Framework for Cloud Infrastructures*. IEEE/ACM 8th International Conference on Utility and Cloud Computing.

Gomez et al. (2016). *Business Modeling facilitated Cyber Preparedness.* The Business and Management Review, Volume 7 Number 4May 2016.

Goode & Hatt. (1952), *Methods in Social Research.*

IBM Services. (2018). *What is data breach?* Retrieved from: https://www.ibm.com/services/business-continuity/data-breach

J. C. S. Núñez, A. C. Lindo and P. G. Rodríguez. (2020). *A Preventive Secure Software Development Model for a Software Factory: A Case Study*, in IEEE Access, vol. 8, pp. 77653-77665, 2020, doi: 10.1109/ACCESS.2020.2989113.

J. Jingping, C. Kehua, C. Jia, Z. Dengwen and M. Wei, (2019). *Detection and Recognition of Atomic Evasions Against Network Intrusion Detection/Prevention Systems*. In IEEE Access, vol. 7, pp. 87816-87826, 2019, doi: 10.1109/ACCESS.2019.2925639.

Kaspersky. (n.d). Malware & Computer Virus Facts & FAQs. Retrieved from: https://usa.kaspersky.com/resource-center/threats/computer-viruses-and-malware-facts-and-faqs

Kaur S.P. (2013). *Variables in Research*. In IJRRMS, Vol. 3, No.4,2013

KUSCCO. (2020). Kenya Union of Savings & Credit Cooperatives Ltd. https://www.kuscco.com/index.php/our-partners

Lockheed Martin Corporation. (2021). *https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html*

M. Borrego, M. J. Foster and J. E. Froyd, "Systematic Literature Reviews in Engineering Education and Other Developing Interdisciplinary Fields", J. Eng. Educ., vol. 103, no. 1, pp. 45-76, 2014.

M. Petticrew and H. Roberts, Systematic reviews in the social sciences: A practical guide, John Wiley & Sons, 2008.

M. T. Qassrawi and Hongli Zhang. (2010). *Client honeypots: Approaches and challenges*. 4th International Conference on New Trends in Information Science and Service Science, 2010, pp. 19-25.

M. Wazid, S. Zeadally and A. K. Das. (2019). *Mobile Banking: Evolution and Threats: Malware Threats and Security Solutions*. in IEEE Consumer Electronics Magazine, vol. 8, no. 2, pp. 56-60, March 2019, doi: 10.1109/MCE.2018.2881291.

Malik Imran Daud. (2010). *Secure software development model A guide for secure software life cycle.*

Manish Gupta, Raj Sharman. (2009). *Social and Organizational Liabilities in Information Security.*

Microsoft. (2020). *Microsoft | Security Engineering*. Retrieved from What are the Microsoft SDL practices?: https://www.microsoft.com/en-us/securityengineering/sdl/practices

Muraguri, N. N., Mwalili, T. & Mose, T. (2019). *Factors influencing cybersecurity readiness in deposit taking savings and credit cooperatives: A case study of Nairobi County*. International Academic Journal of Information Systems and Technology, 2(1), 157-182.

Neeraj Bhagat and Bhavana Arora. (n.d). *Malware analysis on networks using Honeypots: A Review.* International Journal of Latest Trends in Engineering and Technology, e-ISSN:2278-621X, Vol. 10, Issue 1, pp.366-370, 2018. 15.

New York Business Law Journal. (2017). NYSBA *NY Business Law Journal, Summer 2017, Vol. 21, No. 1.*

Ö. Aslan and R. Samet. (2017). *Investigation of Possibilities to Detect Malware Using Existing Tools, 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), 2017, pp. 1277-1284, doi: 10.1109/AICCSA.2017.24.*

P. Black and J. Opacki. (2016). *Anti-analysis trends in banking malware.* 11th International Conference on Malicious and Unwanted Software (MALWARE), Fajardo, 2016, pp. 1-7, doi: 10.1109/MALWARE.2016.7888738.

P. D. Ali and T. G. Kumar. (2017). *Malware capturing and detection in dionaea honeypot.* 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), *pp. 1-5, 2017.*

P. N. Bahrami, A. Dehghantanha, T. Dargahi, R. M. Parizi, K. R. Choo, and H. H. S. Javadi. (2019). *Cyber Kill Chain-Based Taxonomy of Advanced Persistent Threat Actors: Analogy of Tactics, Techniques, and Procedures*, J Inf Process Syst, Vol.15, No.4, pp.865~889.

P. R. Chandre, P. N. Mahalle and G. R. Shinde, (2018). *Machine Learning Based Novel Approach for Intrusion Detection and Prevention System: A Tool Based Verification*. IEEE Global Conference on Wireless Computing and Networking (GCWCN), 2018, pp. 135-140, doi: 10.1109/GCWCN.2018.8668618.

P. Shahegh, T. Dietz, M. Cukier, A. Algaith, A. Brozik and I. Gashi. (2017). A*VAMAT: AntiVirus and malware analysis tool.* 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), 2017, pp. 1-4, doi: 10.1109/NCA.2017.8171379.

Ponemon LLC. (2018). *Cost of a Data Breach Study: Global Overview.*

Pritha Bhandari. (2020). *A Step-by-Step Guide to Data Collection.* Retrieved from: https://www.scribbr.com/methodology/data-collection/

Rami Sihwail, Khairuddin Omar, K. A. Z. Ariffin, (2018). A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis.

Reid. (2018). *Cybersecurity Starts at the Top: Why Top Management Must Set the Tone for Data Security*. Retrieved from: https://www.onserve.ca/cybersecurity-starts-at- the- top-why-top management-must-set-athe-tone-for-data-security/

S. Kumar, R. Sehgal and J. S. Bhatia. (2012). *Hybrid honeypot framework for malware collection and analysis*, 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS), Chennai, India, 2012, pp. 1-5, doi: 10.1109/ICIInfS.2012.6304786.

S. Musman, M. Tanner, A. Temin, E. Elsaesser and L. Loren. (2011). *A systems engineering approach for crown jewels estimation and mission assurance decision making*, 2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Paris, 2011, pp. 210-216, doi: 10.1109/CICYBS.2011.5949403.

Sacco Societies (Non-Deposit Taking Business) Regulations. (2020). Retrieved from https://sasra.go.ke/

SASRA. (2010). *The Sacco Societies (Deposit-Taking SACCO Business) Regulations (L.N. No. 95 of 2010).* Retrieved from: https://sasra.go.ke/index.php?option=com_phocadownload&view=category&id=15&Itemid=118

SASRA. (2018). *Annual Supervisory Report 2018*

Serianu. (2018). *SACCO Cyber Security Report; Demystifying Cybersecurity for SACCOs*

Serianu. (2019). *SACCO Cyber Security Report; Digital Transformation and Cyber Risk within SACCOs.*

Sethia & Jeyasekar. (2019). *Malware Capturing and Analysis using Dionaea Honeypot, 2019 International Carnahan Conference on Security Technology (ICCST)*

Sherif, J. S., & Ayers, R. (2003). *Intrusion detection: methods and systems, Part II*. Information management & computer security, 11(5), 222-229.

T. Meline, "Selecting studies for systematic review: Inclusion and exclusion criteria", Contemp. Issues Commun. Sci. Disord., vol. 33, no. 21–27, 2006.

V. Sethia and A. Jeyasekar, "Malware Capturing and Analysis using Dionaea Honeypot," 2019 International Carnahan Conference on Security Technology (ICCST), 2019, pp. 1-4, doi: 10.1109/CCST.2019.8888409.

V. V. Polyakov and S. A. Lapin. (2018). *Architecture of the Honeypot System for Studying Targeted Attacks*, XIV International Scientific-Technical Conference on Actual Problems of

Electronics Instrument Engineering (APEIE), Novosibirsk, 2018, pp. 202-205, doi: 10.1109/APEIE.2018.8545323.

Vadaviya, D., Panchal, M., Jhummarwala, A., & Potdar, M. B. (2019). *Malware Detection using Honeypot and Malware Prevention*, International Journal of Computer Engineering and Technology (IJCET), Volume 10, Issue 06, pp1-9.

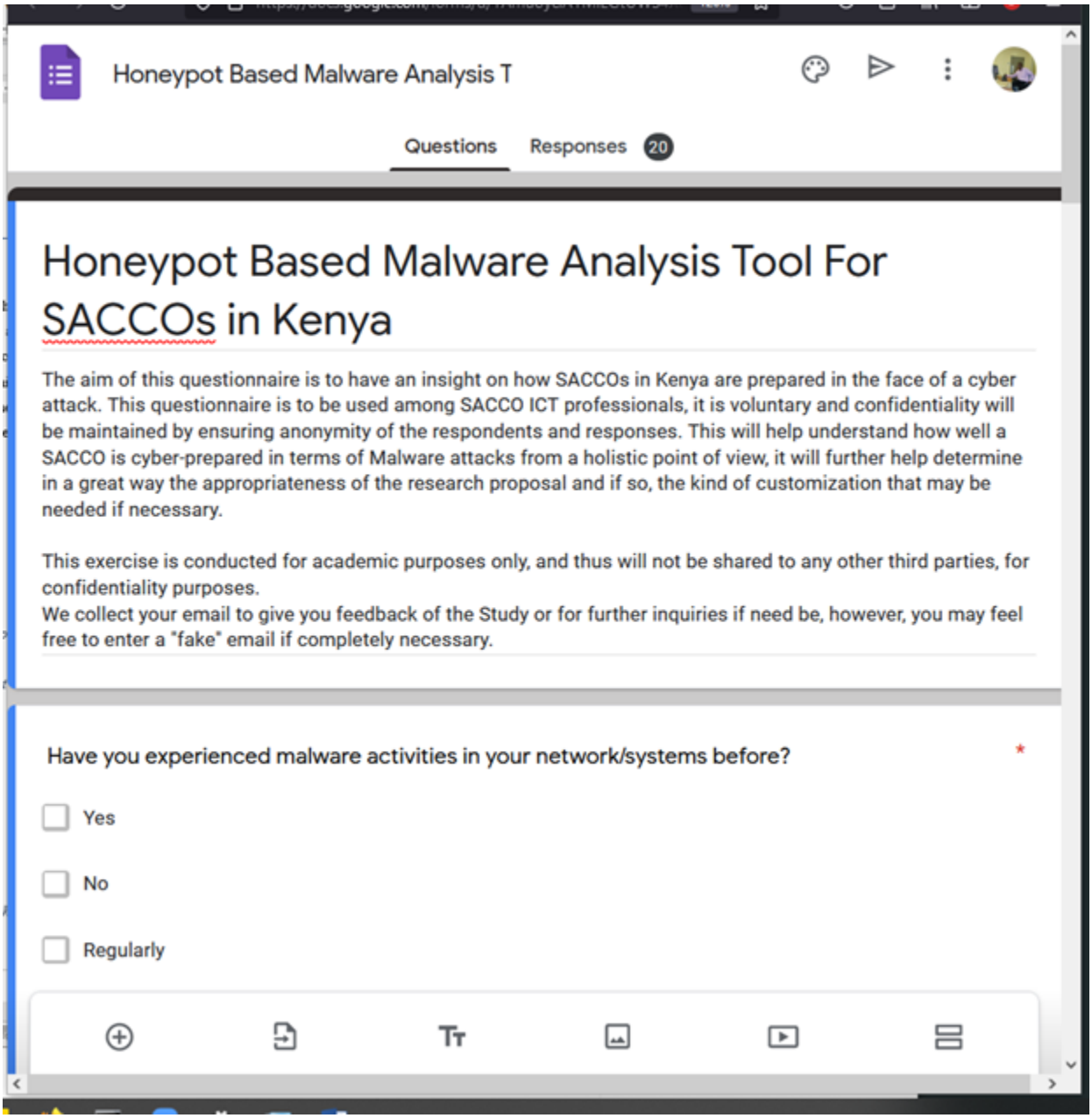
Walliman, N. S. & Walliman N. (2011). *"Research methods: the basics"*.

Weiler, N. (2002). *Honeypots for distributed denial-of-service attacks*, Paper presented at the Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002.

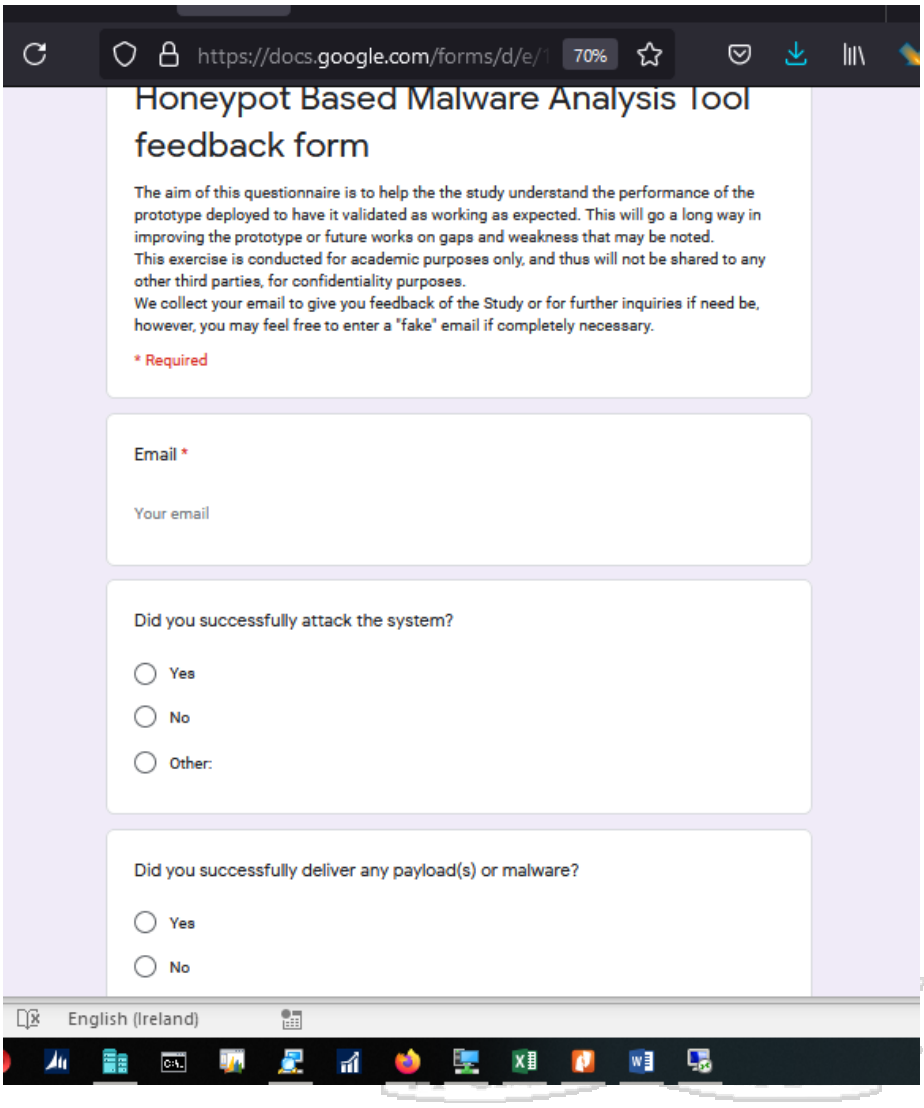
Y. Ali and A. Hameed. (2019). *Cloud Crypter for bypassing Antivirus*, 15th International Conference on Emerging Technologies (ICET), 2019, pp. 1-6, doi: 10.1109/ICET48972.2019.8994615.

Appendices

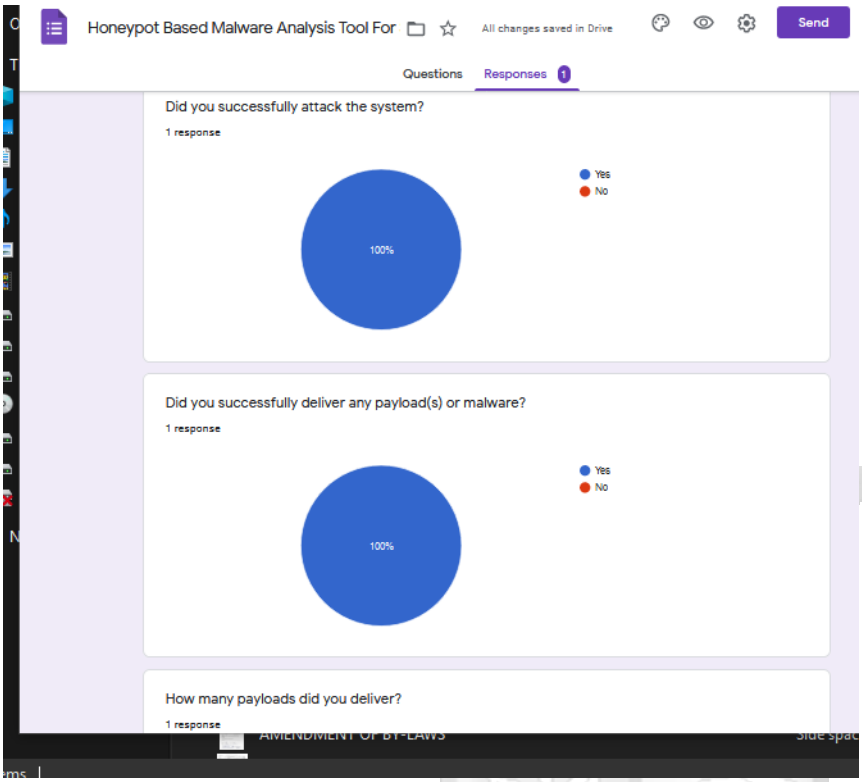
Appendix A: Gathering of requirements questionnaire



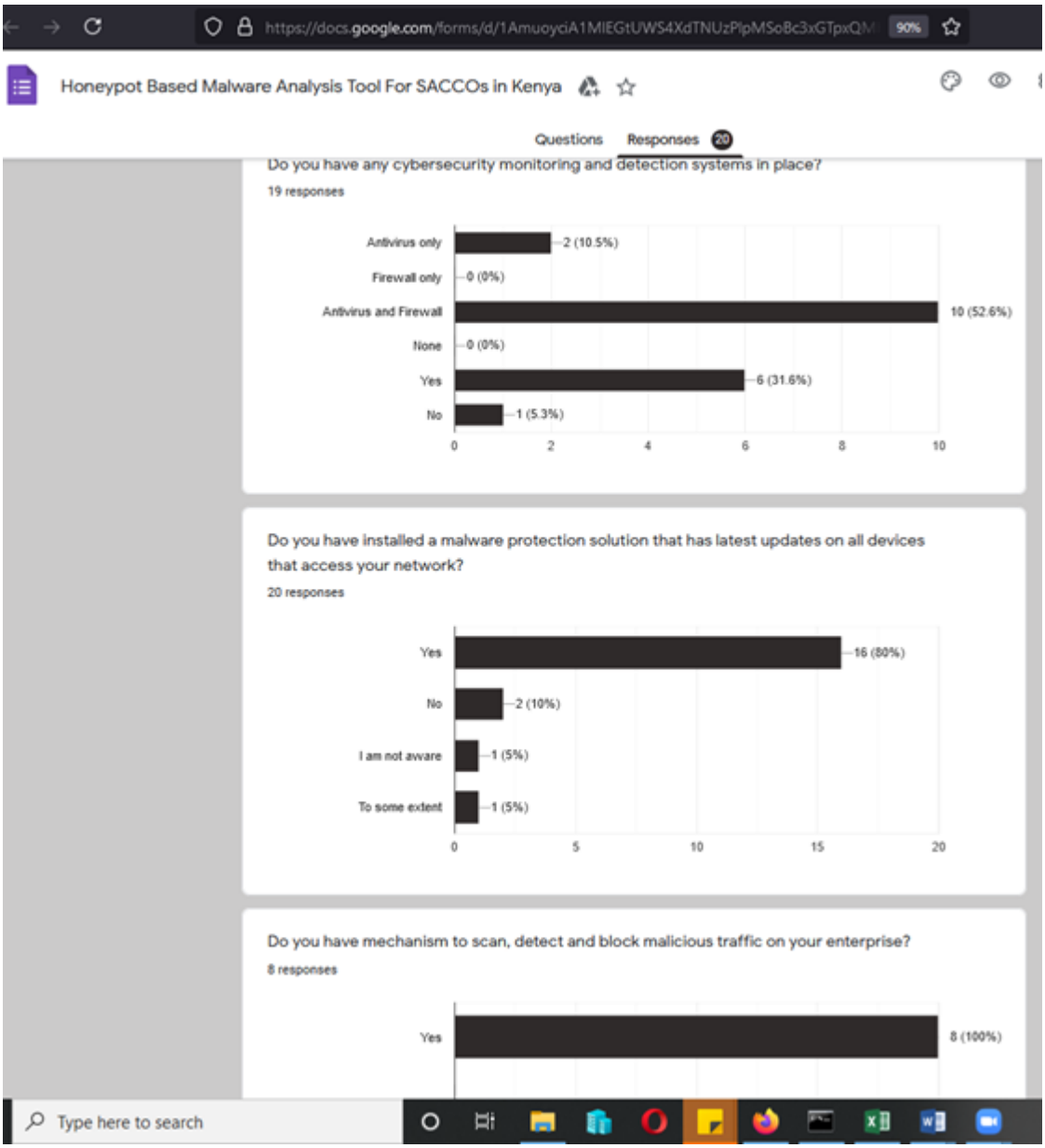
Appendix B: Feedback gathering of questionnaire



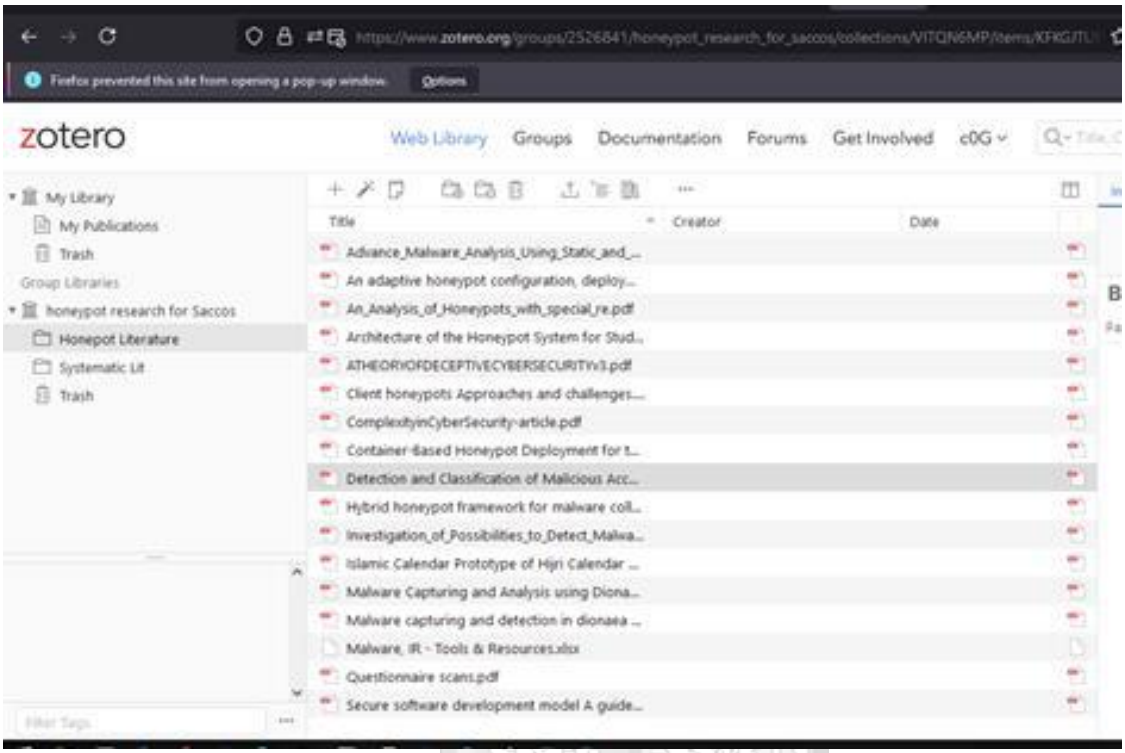
Appendix C: Sample response from feedback form



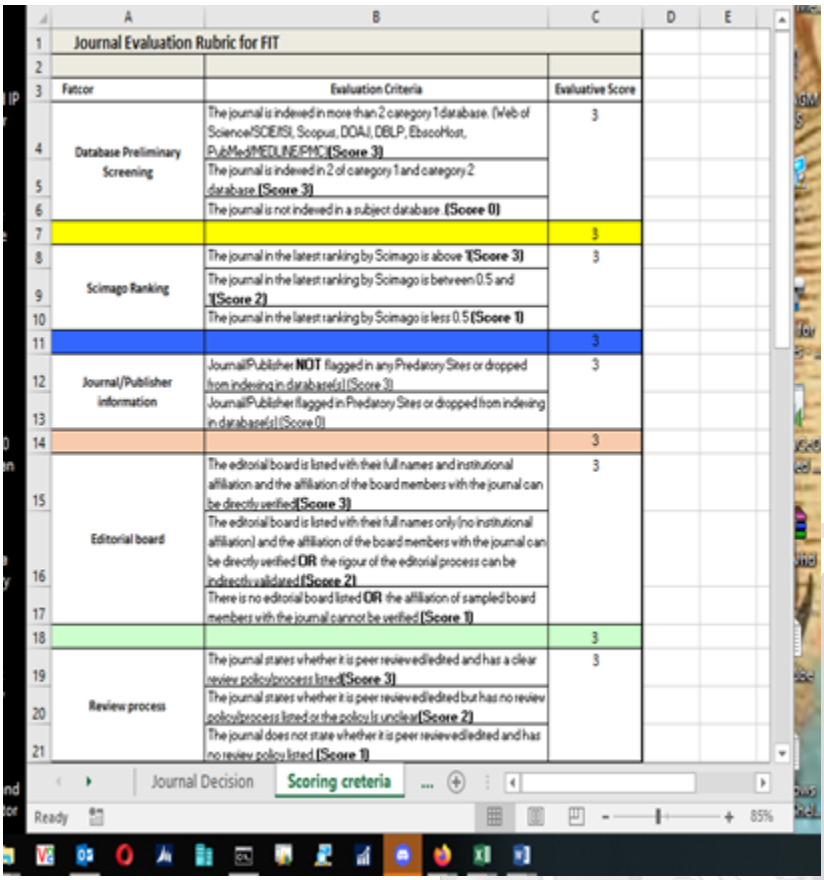
Appendix D: Sample Responses



Appendix E: Cataloguing of sources

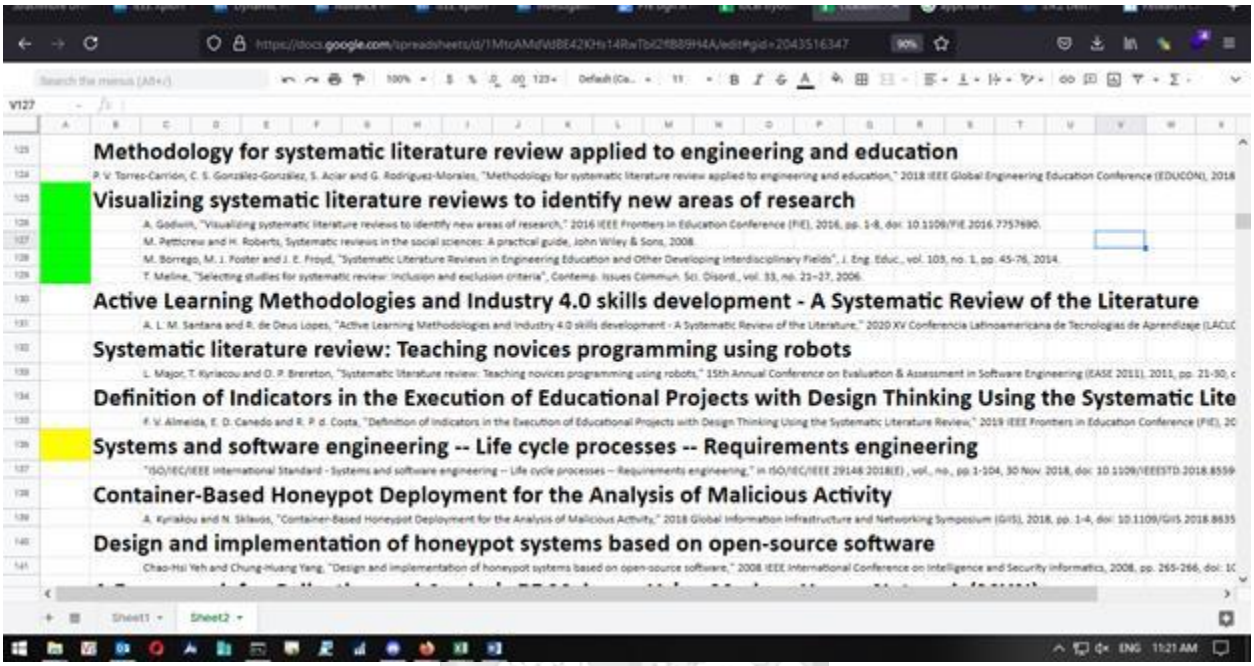


Appendix F: Tool used for Critiquing and Appraisal of literature



Journal Evaluation Rubric for FIT		
Fatcor	Evaluation Criteria	Evaluative Score
Database Preliminary Screening	The journal is indexed in more than 2 category 1 database. (Web of Science/SCIE/ISI, Scopus, DOAJ, DBLP, EbscoHost, PubMed/MEDLINE/PMC) **(Score 3)**	3
	The journal is indexed in 2 of category 1 and category 2 database **(Score 3)**	
	The journal is not indexed in a subject database **(Score 0)**	
		3
Scimago Ranking	The journal in the latest ranking by Scimago is above 1**(Score 3)**	3
	The journal in the latest ranking by Scimago is between 0.5 and 1**(Score 2)**	
	The journal in the latest ranking by Scimago is less 0.5 **(Score 1)**	
		3
Journal/Publisher information	Journal/Publisher **NOT** flagged in any Predatory Sites or dropped from indexing in database(s) (Score 3)	3
	Journal/Publisher flagged in Predatory Sites or dropped from indexing in database(s) (Score 0)	
		3
Editorial board	The editorial board is listed with their full names and institutional affiliation and the affiliation of the board members with the journal can be directly verified**(Score 3)**	3
	The editorial board is listed with their full names only (no institutional affiliation) and the affiliation of the board members with the journal can be directly verified **OR** the rigour of the editorial process can be indirectly validated **(Score 2)**	
	There is no editorial board listed **OR** the affiliation of sampled board members with the journal cannot be verified **(Score 1)**	
		3
Review process	The journal states whether it is peer reviewed/edited and has a clear review policy/process listed**(Score 3)**	3
	The journal states whether it is peer reviewed/edited but has no review policy/process listed or the policy is unclear**(Score 2)**	
	The journal does not state whether it is peer reviewed/edited and has no review policy listed **(Score 1)**	

Appendix G: Sample of pooled findings from researched artefacts



Appendix H: SU-IERC Approval

Strathmore UNIVERSITY

7th April 2021

Mr Mwesigwa Keith
keith.mwesigwa@strathmore.edu

Dear Mr Mwesigwa,

RE: A honeypot-based malware analysis tool for Saccos in Kenya 92220

This is to inform you that SU-IERC has reviewed and **approved** your above **SU-master's** research proposal. Your application reference number is **SU-IERC0944/20.** The approval period is **7th April 2021 to 6th April 2022.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and also obtain other clearances needed

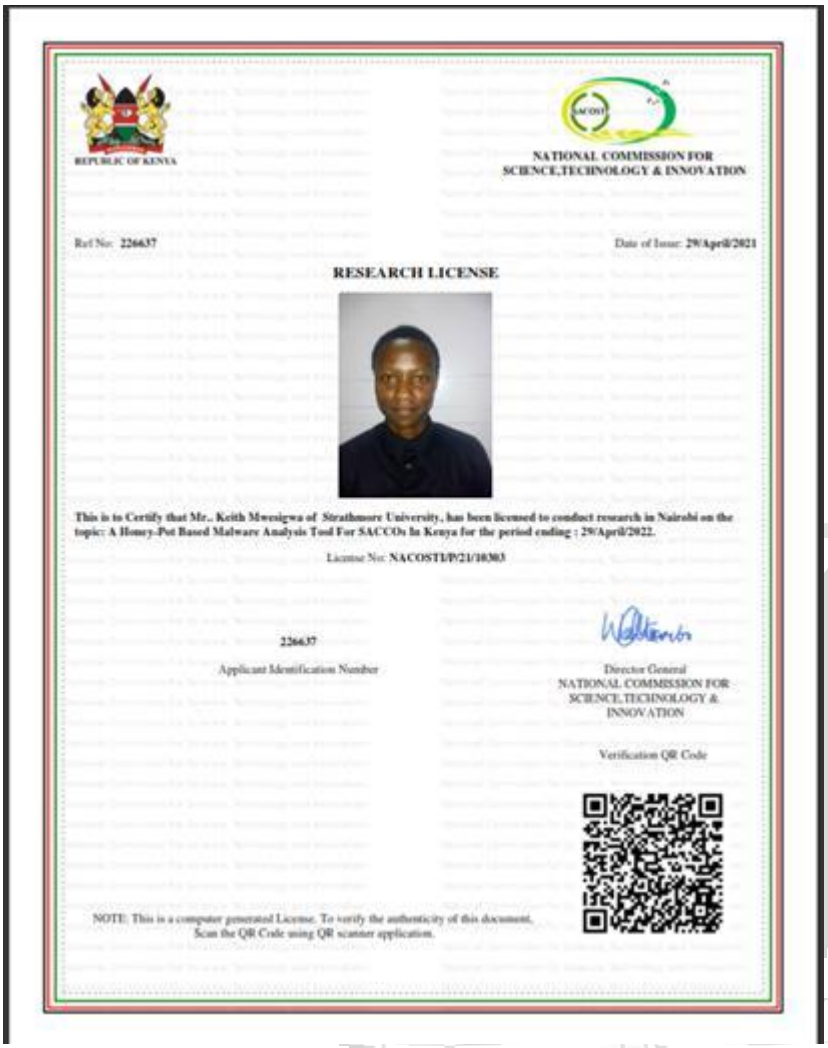
Yours sincerely,

for: Dr Virginia Gichuru,
Secretary; SU-IERC

STRATHMORE UNIVERSITY INSTITUTIONAL ETHICS REVIEW COMMITTEE (SU-IERC)
07 Apr 2021
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA

Cc: Prof Fred Were,
Chairperson; SU-IERC

Appendix I: NACOSTI Research License



REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **226637**

Date of Issue: **29/April/2021**

RESEARCH LICENSE

This is to Certify that Mr., Keith Mwesigwa of Strathmore University, has been licensed to conduct research in Nairobi on the topic: A Honey-Pot Based Malware Analysis Tool For SACCOs In Kenya for the period ending : 29/April/2022.

License No: **NACOSTI/P/21/10303**

226637

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.