CHAPTER 1

INTRODUCTION

This chapter includes the following subtopics, namely: (1) Rationale; (2) Statement of the Problem; (3) Objective and Hypotheses; (4) Scope and Delimitation; (6) Significance of the Study.

1.1 Rationale

Ransomware is one of the most significant threats to cybersecurity [5]. The threat posed by attacks with ransomware keeps increasing quickly over time due to the introduction of new versions with different signatures [5, 8, 16, 21, 29]. The threat of ransomware has been the subject of much investigation. It is acknowledged, meanwhile, that ransomware is still searching for weaknesses in the security protocols in place. Ransomware has genuinely developed into a new economic model in the field of cybercrime. With the advent of ransomware-as-a-service (RaaS) in 2015, anybody may launch an attacks with ransomware by purchasing attack services [13, 26]. Globally, attacks involving ransomware are increasing every year. One characteristic of this situation is the spread of many ransomware attack variants [7].

Ransomware is a type of malware, sometimes known as harmful software. Data that is unique and significant to individuals, businesses, government organizations, the healthcare industry, and the Internet of Things (IoT) can be encrypted by ransomware. Additionally, ransomware can hinder computer functions and deny users access to certain data or systems [5, 26, 37]. In the case of a ransomware attack, the decryption process is the only way to restore files. However, a password that only the attacker knows is needed to decode the data. To the victim's detriment, the attacker will unlock the password if the victim pays the required ransom [5, 20, 21]. Ransomware may conduct encryption using a variety of key types, such as symmetric, asymmetric, and hybrid symmetric keys. Once successfully installed on a computer, some ransomware will establish a connection with a Command and Control (C&C) server in order to transmit the key. After then, the ransomware will start encrypting the files. It is highly likely to be detected in a network as long as the ransomware is still connecting and communicating with the server [29].

Ransomware spreads through phishing emails, malicious software, unapproved downloads, and vulnerabilities [26]. Ransomware may swiftly encrypt victim data and functions in tandem with file activities. Using signatures as a security measure is insufficient. Given that many contemporary attacks by ransomware take advantage of zero-day vulnerabilities, further ransomware detection techniques are also required [7]. The built-in detection model has issues since attackers can still supply fictitious data files to avoid detection models.

Ransomware attacks can persist because attackers can evade existing detection methods [1].

The most time-sensitive issue, when detecting the detectability problems of ransomware, is how quickly to take action on them and resolve them. In terms of detection and response to ransomware, the response time refers to just how long it takes for a security system, once it has encountered such a ransomware threat in action, in the first place to discover that (if at all) and, secondly, what possible mitigation measures are possible. A rapid response is vital to mitigating the effects of an attack and stopping the wider spread of ransomware. According to Microsoft, threat detection and response refers to a set of cybersecurity processes aimed at the identification of threats to an organization's digital assets and the immediate application of measures to mitigate them. This involves continued monitoring, threat analysis, and implementation of controls to deliver adaptive threat management, especially around threats like ransomware [25].

Over the previous few years and to the present, a tremendous deal of research has been conducted to identify ransomware using various methodologies. Three types of analysis are available for ransomware behavior and characteristics, depending on the network or host: hybrid analysis, which combines static and dynamic analysis; dynamic analysis; and static analysis. Without actually running the file, static analysis determines if it contains ransomware. One of the methods used in static analysis is feature extraction from files, which involves examining the binary file to get information about its contents and structure. One problem with static analysis is that it can't find complex malware. Also, attackers can use encryption and polymorphism on purpose to get around security measures that depend on structural data from static analysis. Initially, ransomware attacks search for harmful activity using a dynamic analysis technique. To prevent harm during the analysis process, dynamic analysis is carried out in a secure environment, such as a cuckoo or sandbox. Dynamic analysis is thought to be more accurate in detecting malware than static analysis. One problem with dynamic analysis is that it can't find ransomware attacks, which change behavior to trick the system [1, 17, 26]. These attacks are also known as evasion attacks. In order to fool the classification model into believing that an attack has taken place, evasion attacks are executed by altering harmful inputs [1]. Given the limitations of both static and dynamic analysis, combining the two techniques can increase the ability to detect ransomware attacks. Hybrid analysis, which combines static and dynamic analysis, provides a more robust security mechanism [17]. Some antivirus software still often uses analysis based on signature matching in addition to static and dynamic analysis. However, signature-based analysis also has drawbacks because it is unable to identify novel variants [29].

1.2 Statement of the Problem

Usman Ahmed et al. performed both static and dynamic analysis using the Android APK host file. The detection models are anticipated tovaded by the finished product. The model, which was constructed using ensemble learning, has 99% accuracy, recall, and F measure values. Even if the results are nearly perfect, the integrated detection model is unable to immediately thwart ransomware attacks. From the perspective of dynamic analysis, the system must be infected before it can be compromised since the proposed architectural model incorporates an ensemble learning voting mechanism [1].

Ahmad et al. used network-based dynamic analysis methods because they thought host-based detection would not work as well because the system can't be fixed until the host is infected. The researchers thus developed a network-based packet categorization system that consists of two separate classifiers running concurrently. The researchers created the PCAP file records that made up the dataset by taking advantage of one of the cryptoattacks known as Locky. Because dynamic analysis demands a lot of data, datasets from the Czech Technical University's (CTU) Malware Capture Facility Project (MCFP) were added. The study found some new features that could be used to find ransomware attacks. These features include low false positive rates and high detection accuracy for each level (Random Tree at 97.92% and Bayes Net at 97.08%) [3].

A research study by Muhammad Ijas et al. found that machine learning-based static analysis has an accuracy rate of 94.64%, but dynamic analysis has an accuracy rate of 99.36%. However, the study pointed out that dynamic analysis has a number of limitations, such as the inability to conduct a comprehensive analysis because of network behavior and restricted network access [18].

The problem identified in this study is the lack of emphasis on detection in stopping ransomware after it has successfully penetrated the host system. Although dynamic analysis is more successful than static analysis in detecting ransomware, it has the disadvantage of requiring the host system to be infected before the threat can be discovered. Furthermore, ransomware is always evolving with obfuscation methods to mask its traces, allowing it to often avoid detection technologies and penetrate systems unnoticed. To solve these challenges, this study creates a response-time detection framework that can identify the existence of ransomware as soon as feasible, before it begins processing and encrypting all files on the host system. This strategy is designed to stop the attack early, minimizing the effect of ransomware. However, the proposed framework requires datasets that are appropriately aligned with its design. To address the challenge of a lack of reliable sources for this type of data, this study also develops a relevant data set to evaluate the effectiveness of the framework. This study seeks to address the identified challenges through the following research questions:

1. How can a response-time detection framework improve ransomware identification

before it starts encrypting files on the host system?

2. What are the advantages of using a response-time detection approach compared to traditional dynamic analysis in stopping ransomware attacks?

3. How can a properly aligned dataset enhance the evaluation and effectiveness of the proposed ransomware detection framework?

1.3 Objective and Hypotheses

This research aims to improve the detection and prevention rate of ransomware attacks on hosts as early as possible while maintaining or improving accuracy compared to previous research. This improvement is achieved through the implementation of a new detection architecture, namely response-time host-based detection and prevention. This research hypothesizes that the application of dynamic analysis in host-based detection can increase the effectiveness of identifying and preventing ransomware attacks at the earliest stage of execution.

The two primary components of ransomware analysis are static analysis and dynamic analysis, as previously mentioned. Modern ransomware is constantly changing and has various signatures, making signature-based approaches less effective. To identify new threats, detection techniques must be more rapid and flexible. In this research, the use of machine learning is an appropriate solution because it can increase the speed of detection and recognize more complex ransomware patterns. In addition, dynamic analysis is also used to understand ransomware activity in more depth. With this approach, machine learning can learn ransomware characteristics automatically and more accurately. Since the main goal of this research is rapid detection, the proposed framework utilizes machine learning to build a pre-trained classifier model that is able to identify ransomware efficiently.

This strategy aims to mitigate the impact of ransomware attacks before data encryption occurs. The architecture in which the trained detection model is developed is designed to continuously monitor system activity and identify attack patterns at an early stage. This study also develops a dedicated dataset using Cuckoo, in which ransomware files are gathered and their API calls are logged during execution to analyze attack patterns. This dataset is utilized for training and evaluating the detection model. Furthermore, the framework's capability to detect ransomware in its early execution phase is evaluated to determine its effectiveness.

1.4 Scope and Delimitation

This study is confined to dynamic analysis and datasets gathered over several months for both network-based and host-based detection. This work employs many machine learning models previously utilized by academics, specifically Support Vector Machine, Random

Forest, and Naive Bayes, for identification purposes. Each model is evaluated according to performance metrics, including accuracy, precision, recall, F1-score, false positive rate (FPR), and false negative rate (FNR). This research does not thoroughly address the implementation of response-time detection in host-based systems. This research primarily concentrates on the notion of quick detection using windowing approaches to enhance the efficiency and speed of ransomware identification. This research employs a windowing technique to enhance the host-based detection of ransomware in response-time. This system restricts the number of API requests to 1,000 per window, with a maximum of 10 windows in a single looping process. Because of this, the system may look at the first 25,000 to 35,000 API requests, making sure that the attack is found before it's too late. This strategy dramatically enhances the speed of ransomware detection at the onset of an attack, thereby minimizing possible harm.

This study is confined to dynamic analysis and datasets gathered over several months for host-based detection. This work employs many machine learning models previously utilized by academics, specifically Support Vector Machine, Random Forest, and Naive Bayes, for identification purposes. Each model is evaluated according to performance metrics, including accuracy, precision, recall, F1-score, false positive rate (FPR), and false negative rate (FNR). This research does not thoroughly address the implementation of response-time detection in host-based systems. This research primarily concentrates on the notion of quick detection using windowing approaches to enhance the efficiency and speed of ransomware identification.

The training dataset and observation material for the response-time detection system can only employ API calls that function on Windows. As a result, this study only looks at API calls in the Windows operating system and not in other operating systems like macOS or Linux. This might limit the generalizability of the research findings, especially when extending the proposed approach to contexts with different operating systems.

1.5 Significance of the Study

This research's innovation is in the response-time detection based on the host system. The objective is to enhance the detection rate by implementing a response-time detection strategy. Additionally, windowing techniques conduct a response-time update of the detection methodology, which halts the execution of ransomware files. This detection system may be developed and utilized in several domains, including enhancing security on the user's host system and being applied in major corporations that keep critical public data. This detection technique is crucial in safeguarding critical government data, given that ransomware frequently targets major companies. This detection technique may identify ransomware attacks, which require time to execute encryption. This response-time detection approach can save at least more than 50% of files from the onset of a ransomware attack.