Chapter 1

Introduction

1.1 Introduction

Computer security is a significant concern in the modern digital era, where malicious software, commonly referred to as malware, poses a grave threat to computer systems and the data they store. The concept of malware is not a recent one. In 1951, John von Neumann introduced the theoretical concept of self-replicating automata. However, early instances of malware were created for benign purposes, such as for amusement or experimentation, and they did not have malicious intentions. The landscape of malware changed dramatically with the emergence of the boot sector virus in the late 1980s, marking the introduction of malicious intent in malware. Another notable development was the rise of polymorphic viruses, which could alter their appearance to evade detection.

With the advent of Windows 9x operating systems, a new category of malware known as macro viruses emerged. These viruses had the ability to spread through email attachments and infect Microsoft Office documents. During the 1990s, malware did cause some disruptions, but its overall impact remained relatively limited. However, as the internet rapidly expanded in the late 1990s, more destructive forms of malware surfaced. These sophisticated malware variants utilized email and other complex technologies to propagate rapidly, aiming to disrupt computer systems, pilfer confidential information, and inflict financial harm. It’s noteworthy that, at that time, most malware attacks were primarily targeted at MS-DOS or Windows platforms.

From 2001 to 2010, malicious software attacks grew in complexity and became more focused. The primary target for malware attackers was the Windows operating system, and email was the main avenue for spreading malware. Nevertheless, attackers began utilizing alternative channels like IRC and Instant Messenger, and they started developing more intricate and specific attack methods. In the late 2000s, malware reached new levels of sophistication, with attackers crafting malware to infiltrate programmable logic controllers (PLCs) within industrial control systems. They also exploited cross-site scripting (XSS) vulnerabilities in major social networking platforms to launch attacks. Additionally, malware authors devised novel techniques to obtain digital certificates legally, enabling malware to evade browser security and anti-malware systems. It is continuously evolving with changes in technology. Day by day, the cyber attacks are becoming more and more targeted with even better evading techniques. The existing vulnerabilities in the Windows platform are exploited by malware attacks and have even spread to mobile operating systems, especially Android, because of its mass use. In 2012, state-sponsored cyber espionage attacks grew in frequency and complexity. These attacks employed highly advanced malware to extract sensitive information from selected computers, primarily in the Middle East. One notable example was the Wiper malware, which aimed to disable computer systems within numerous Middle Eastern oil organizations. This incident heightened concerns about cyber warfare, prompting countries to invest in enhanced defense mechanisms. The year 2013 witnessed an escalation in financial cyber threats. Banking malware groups developed Trojans and backdoors to pilfer funds from online accounts or gather data necessary for financial theft. The increasing popularity of Bitcoin also attracted malware authors who devised ways to steal cryptocurrency from victims' systems. Additionally, cybercriminals harnessed victims' machines for cryptocurrency mining. The years 2014 and 2015 marked the emergence of ransomware attacks. Attackers leveraged Tor anonymization technology to conceal command servers and employed Bitcoin for transactions. Security concerns escalated as attackers expanded their expertise to non-Windows platforms like Android and Linux.

Present-era cyber threats have evolved to hazardously infect systems and platforms that were not known to be vulnerable earlier. With each passing day, the structure, aspect, and methods of cyber-attacks are emerging complexly with increased stealth and frequency of attack. From clickless threats to the emergence of personal Internet-of-Things (IoT) attacks. In late 2016, a new form of cyber-attack came into being. Mirai, an open-source botnet that infects Internet-of-Things (IoT) devices like thermostats, webcams, home security systems, and routers. The attacks on such devices show that the entire malware system is in constant adaptation.

With the wide availability of tools and technologies, existing malware programs are becoming sophisticated. New malware binaries are constantly uploaded on the Internet to create havoc.

1.2 Malware and Its Types

Any program that ”deliberately fulfills the harmful intent of an attacker” is commonly referred to as malicious software or malware [3]. Such programs are designed to gain access to computer systems and network resources to gain sensitive information, disrupt ongoing services, and normal operations without the owners' knowledge. When this happens widely, it creates devastation that permeates the entire Internet. Malware programs that have been observed in the wild come in various forms. The most common types of malware are Adware, Backdoor, Bot, Downloader, Ransomware, Rootkit, Trojan, Virus, and Worm.

1.3 Stages of a Malware Attack

Malware attacks can vary greatly in complexity and sophistication. Some attacks are simple and involve only a single stage, while others are more complex and involve multiple stages and different types of malware. In simpler attacks, a single malicious code or technique may be used to breach a system or compromise security. In more complex attacks, a series of interconnected steps may occur, each with a specific purpose in compromising the target. The common steps of a malware attack are as follows:

- Reconnaissance: Typically, this is the first step of a malware attack, wherein an attacker assembles various useful and potentially exploitable information about a chosen target network or system. It is one of the most crucial steps because, depending on the information collected in this step, the next plan of action is prepared. Information such as network configurations, IP addresses, system vulnerabilities, software versions, and domain names are usually collected in this step. In this step, the attacker doesn't actively engage in any malicious activities; it only passively listens or monitors the activities of the users/systems in the targeted network for a period of time.
- Scanning and Distribution: The reconnaissance step is immediately followed

by the scanning and distribution stages. After identifying the truly vulnerable systems, malware assesses the identified targets to carry out port scans. Port scans on the target systems are effective for discovering open network ports which in turn serve as potential entry points for the attacker. At this point, known vulnerabilities associated with any running services or software on the target system may be identified for further exploitation in the subsequent steps.

Once the scanning is over, the next task for the attacker is to mechanize a delivery method to distribute the malicious payload to the target systems. For distribution purposes, methods such as social engineering, click-bait techniques, phishing, or drive-by downloads may be used. Both these stages are significant because these are the stages through which the attacker tries to gain a foothold or initial access into the targeted system.

- Exploit: The next step in a malware attack is exploitation where the vulnerabilities identified earlier are leveraged by the attacker to gain unauthorized and illegal access to the system. For each identified vulnerability, an exploit code may be designed with malicious intents such as taking control of the target system or executing unwanted software.

- Infection: After successfully identifying and exploiting the vulnerabilities, the next step is to infect the target system by executing the malicious code. Here, the primary malicious objective of the attacker is executed. Consequences of infection may be in the form of data theft, system compromise, or other malicious activities. Another objective of this step is for the malicious code to persist on the system for a long period and even after system reboots.

- Lateral Movement: After infecting a target, the attacker tries to escalate and propagate the attack for more damaging consequences. The attacker would want the malicious software and its activities to be propagated through the connected network to infect more systems or target more highly valued victims.

- Exfiltration and Corruption: This is the final step in a malware attack. Valuable data from the compromised system are stolen or transferred to a system directly under the control of the attacker. For transferring the victim's valuable data, it is first identified (files of specific types and extensions), collected, encrypted (for maintaining confidentiality), and concealed to avoid detection. However, instead of transferring the data, the attacker may have some other motives as well, such as tampering with or corrupting the data rendering it unusable; encrypting the data and demanding ransom from the victim; destructing files in the system, making them irrecoverable; launching DoS attacks; and lastly, the most damaging consequence may be wiping off the data from the system as a whole, making it extremely difficult for recovery.

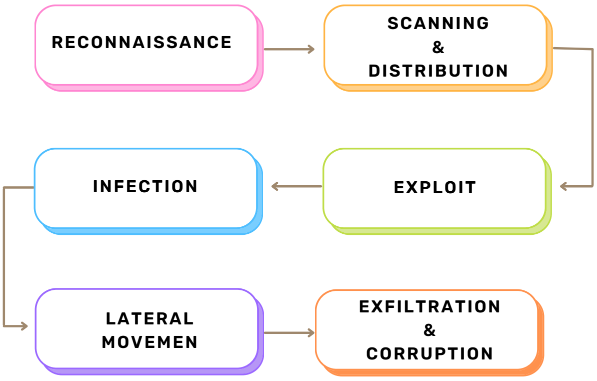

Figure 1-1: Stages of a malware attack

1.4 Defense Approaches

A detection technique aims to identify malware instances. Although many techniques exist for detecting malware, the increasing complexity of malware, often employing hidden techniques, necessitates the development of advanced methods. Generally, malware detection techniques can be broadly categorized into two types: anomaly-based detection and signature-based detection. Figure 1-2 presents a hierarchy of malware detection approaches.

1.4.1 Signature-based Detection

Signature-based detection techniques attempt to build a repository of signatures and use this repository in the detection of malware. Any software that exhibits any malicious behavior already present in the signature repository can be categorized as malicious. Whenever a new malware instance is encountered, the signature database is updated with its signature. The signature is created by looking for

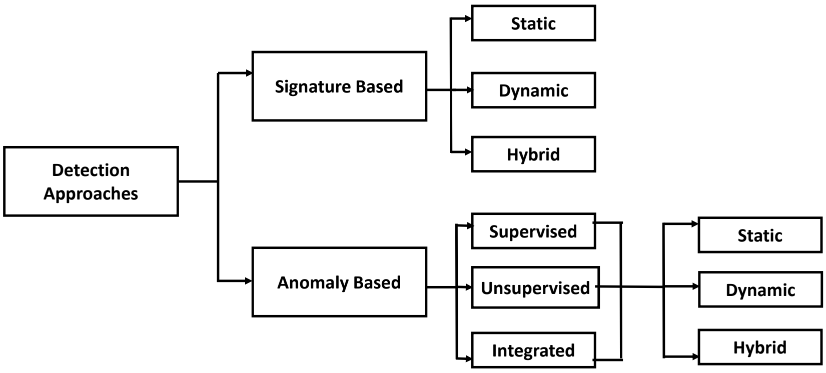

Figure 1-2: Detection Approaches

key features of the malware binary. Once the signature is created, it is stored in the signature repository. Most anti-malware scanners are based on signature-based detection. The main advantage of this technique is that it can detect known instances of malware with a low false positive rate. The major drawback is that it cannot detect zero-day malware since no corresponding signature is available in the repository.

1.4.2 Anomaly-based Detection

The term anomaly-based malware detection refers to finding exceptional behavior in malware that does not conform to the expected or normal behavior. These non-conforming patterns are often referred to as anomalies, outliers, exceptions, aberrations, surprises, peculiarities, or discordant observations in various application domains. An anomaly-based detection technique uses the knowledge of normal behavior to decide the maliciousness of a program under inspection. The technique has two phases: the training phase and the detection phase. During the training phase, a model is trained based on the normal behaviors, and a machine learning technique is used to create a profile of such normal behaviors, to be used later as a reference to distinguish between normal and malicious instances. In the detection phase, this profile is compared against the current behavior and the classifier decides if it is normal or malware.

1.4.2.1 Supervised Approach

In a supervised machine learning approach, there are two key phases: training and testing. During the training phase, a model is trained using a labeled dataset, where each instance is paired with a known class label. The model learns a target function that maps the input data to the corresponding class labels. Once the training is complete, the model's performance is evaluated in the testing phase. Here, the model is tested on a separate set of data, known as the test dataset, which contains instances it has not seen before. This phase assesses the model's accuracy and its ability to make correct predictions on new, unseen data.

1.4.2.2 Unsupervised Approach

In an unsupervised machine learning approach, the dataset used for training does not contain labeled instances. The goal is for the model to learn patterns and structures from the data on its own. The model tries to group similar data points into clusters or categories based on their inherent similarities. This type of learning is useful when the class labels are not known or when we want to discover hidden patterns within the data. Unsupervised learning techniques include clustering, anomaly detection, and dimensionality reduction.

1.4.2.3 Integrated Approach

The hybrid approach combines both supervised and unsupervised methods to build a comprehensive detection model. Initially, the model is trained on labeled data using a supervised learning technique to recognize known malware patterns. This phase leverages the strengths of supervised learning in accurately identifying known threats. After this, an unsupervised component is incorporated, such as clustering or anomaly detection, to identify novel and previously unseen threats. By combining these approaches, the hybrid model benefits from the precision of supervised learning while remaining adaptable to new and evolving malware through unsupervised techniques. This dual strategy enhances the model's ability to detect both known and emerging threats effectively.

1.5 Motivation

Malware, in its various forms, continues to pose a significant risk to individuals, organizations, and society as a whole. In an ever-changing landscape of malware threats that continue to grow in complexity, it is imperative to prioritize the protection of sensitive data, key infrastructure, and digital assets. The motivation to build robust malware defense solutions comes from our ever-increasing reliance on technology and the persistent threat landscape of the digital world. As the size and complexity of malware data keep growing, the defense against malware is becoming increasingly challenging. Nevertheless, this particular challenge serves as a driving force for us to leverage the capabilities of data-driven technologies, such as machine learning, in order to construct robust solutions for defending against malware. However, the appropriate selection of relevant features is necessary to build effective malware defense solutions. Feature selection helps us to distill vast and complex data into the most meaningful and relevant attributes for the detection of malware. The aforementioned factors have collectively served as motivation for the development of cost-effective and robust malware defense solutions.

1.6 Objectives

The objective of this work is to develop effective methods to detect a set of malware and malware-based attacks in various platforms and to evaluate the performance using our own datasets as well as existing benchmark datasets. The objectives are listed below.

- To explore various malware types and their characteristics, with the purpose of creation/analysis of new/existing datasets.

  Justification: Malware plays an important role in any cyber attack. There are many types of malware in the wild and each has unique characteristics. We intend to carry out an in-depth study of various malware types and their characteristics with the goal of creating new real-life datasets. To carry out this task, it is aimed to design a framework to generate malware feature datasets for two different platforms to support the validation of the effectiveness of malware detection methods. It is also aimed to examine the following characteristics to ensure the quality and effectiveness of our datasets.

  1. *Labels in the data*: The instances in the dataset are to be labeled accurately.
  2. *Adequate number of instances*: An adequate number of quality data instances representing each class help to improve the predictive capability of the defense solution.

3. *Adequate number of features*: An adequate number of features helps build the class profile properly as well as avoid overfitting problems.
4. *Balance between the classes*: The distribution of the number of instances against each class in the dataset should be balanced.
5. *Recency*: The constituent data should not be outdated.
6. *Lack of inconsistency*: Data in a standard dataset should be consistent. Any model using inconsistent data is never reliable and will result in wrong decision-making.
7. *Relevance*: The features present in the dataset should be relevant to the problem at hand. Irrelevant features will bring down the performance of the model.

A good number of defense mechanisms against malware have been introduced. It is aimed to carry out an extensive survey with the goal of studying malware, types, analysis mechanisms associated with the detection of malware, defense approaches to detect and mitigate malware, and their pros and cons.

- To explore the selection of an appropriate machine learning or statistical technique for the detection of malware.

  Justification: A large number of detection methods have been proposed in the recent past to hinder the growth of malware and malware-based attacks. It is intended to carry out a theoretical study to analyze the pros and cons of some of the popular machine learning-enabled defense techniques.

- To conduct an empirical study on a specific malware type and its variants to identify interesting patterns/features.

  Justification: In order to detect malware, it is necessary to identify patterns or features that distinguish a malware program from a benign program. In this work, it is intended to carry out an empirical study on a particular malware type and its variants to identify the most discriminative features or attributes of that malware type to achieve the best possible accuracy.

- To develop a robust defense mechanism using an optimal feature subset specifically tailored for a type of malware, to minimize false alarms and explore its applicability to other malware types.

  Justification: A critical aspect of a defense system is its ability to respond to network intrusion cases with a low false alarm rate. Given the vast amount of data generated daily, the defense system must efficiently process this data and provide near real-time responses with high detection accuracy. This objective aims to create a more robust system for handling malicious attacks, ensuring effective and timely responses.

- To develop an ensemble approach to identify the key Indicators of Compromise (IOCs) that significantly contribute to malware detection.

  Justification: By accurately identifying these IOCs, this approach aims to enhance the effectiveness of malware detection systems, ensuring more precise and timely identification of threats while reducing false positives.

- To develop a robust malware defense system using advanced techniques like Convolutional Neural Networks (CNN) and Graph Neural Networks (GNN) to explore various malware properties for improved detection.

  Justification: Malware has some hidden properties that can be effectively utilized using advanced techniques like Convolutional Neural Networks (CNN) and Graph Neural Networks (GNN). These methods automatically identify intricate patterns and relationships within malware data, enhancing detection accuracy and adaptability. By extracting features and capturing complex interactions, these techniques improve the ability to detect and respond to sophisticated and novel malware threats.

1.7 Contributions

This thesis work comprises the following contributions discussed in detail in the later chapters.

1. Design and development of a malware dataset generation pipeline. This work presents two malware feature datasets, TUMALWD and TUANROMD, on two different platforms to support the validation of malware detection methods. Additionally, image-based and Function Call Graph (FCG) datasets have been developed and publicly released. A web-based tool has also been created for the generation of these datasets.

2. A supervised filter-based feature selector based on the rough set theory. The method introduces a new criterion for the identification of the most relevant features.

3. A fast, yet reliable ransomware defense solution, referred to as ERAND, powered by an optimal feature selection method to discriminate the ran-

somware class as a whole, as well as the eleven variants of the ransomware family from the goodware instances.

4. An ensemble approach called FRAMC is proposed to identify the key features that significantly contribute to the detection of malware. Given the ever-evolving landscape of malicious software, developing effective detection methods is crucial. It focuses on identifying important features for malware detection to enhance the accuracy and efficiency of malware detection systems.

5. A parallel version of the k-Nearest Neighbors (KNN) algorithm, referred to as TUKNN, leverages parallel processing capabilities to enhance the speed and efficiency of KNN computations. An extensive experimental study on various proximity measures within the KNN framework results in recommendations for the most effective measures to achieve improved accuracy with TUKNN. The study also identifies the optimal range of k values specifically for malware and malware-based attack datasets to ensure the best performance. By leveraging CUDA's parallel processing capabilities, the TUKNN is significantly accelerated, enabling much faster computations.

6. Design and development of a Convolutional Neural Network (CNN) based malware defense solution. The method leverages the deep learning capabilities of CNNs to identify and classify a wide range of malware threats by learning and recognizing complex patterns within the data.

7. Design and development of a Graph Neural Network (GNN) based malware defense solution. The method leverages the advanced capabilities of GNNs to identify and classify a wide range of malware threats by analyzing the relationships and interactions within the function call graph of malware.

1.8 Organization of Thesis

The thesis is organized as follows:

1. Chapter 2 describes the related background for understanding malware and its existing defense approaches. It also discusses malware analysis techniques and various security vulnerabilities. Further, a discussion on machine learning approaches and cost-effective methods for malware detection is also included along with the various validation measures.