electronics

Article

Comprehensive Analysis of Advanced Techniques and Vital Tools for Detecting Malware Intrusion

Vatsal Vasani [1], Amit Kumar Bairwa [1], Sandeep Joshi [1], Anton Pljonkin [2], Manjit Kaur [3] and Mohammed Amoon [4,*]

[1] Manipal University Jaipur, Jaipur-Ajmer Express Highway, Dehmi Kalan, Jaipur 303007, India; vatsal.199301139@muj.manipal.edu (V.V.); amitbairwa@gmail.com (A.K.B.); sjoshinew@yahoo.com (S.J.)
[2] Institute of Computer Technologies and Information Security, Southern Federal University, Bol'shaya Sadovaya Ulitsa, 105/42, 344006 Rostov-on-Don, Russia; pljonkin@mail.ru
[3] School of Computer Science and Artificial Intelligence, SR University, Warangal 506371, India; manjitbhinder8@gmail.com
[4] Department of Computer Science, Community College, King Saud University, P. O. Box 28095, Riyadh 11437, Saudi Arabia
* Correspondence: mamoon@ksu.edu.sa

Abstract: In this paper, we explore how incident handling procedures are currently being implemented to efficiently mitigate malicious software. Additionally, it aims to provide a contextual understanding of diverse malcodes and their operational processes. This study also compares various ways of detecting adware against a selection of anti-virus software. Moreover, this paper meticulously examines the evolution of hacking, covering the methods employed and the actors involved. A comparative analysis of three prominent malware detection tools, Google Rapid Response (GRR), Wireshark, and VirusTotal, is also conducted, aiding in informed decision-making for enhancing application security. This paper reaches its conclusion by conducting an exhaustive analysis of two case studies, offering valuable insights into a diverse range of potential leaks and virus attacks that may pose threats to various conglomerates. In essence, this article provides a comprehensive overview that spans incident handling procedures, the historical development of hacking, and the diverse spectrum of tools accessible for achieving effective malware detection.

Keywords: incident handling; malware; malware detection techniques; malware detection tools; Google Rapid Response (GRR); wireshark; VirusTotal

Citation: Vasani, V.; Bairwa, A.K.; Joshi , S.; Pljonkin A.; Kaur, M.; Amoon, M. Comprehensive Analysis of Advanced Techniques and Vital Tools for Detecting Malware Intrusion. *Electronics* **2023**, *12*, 4299. https://doi.org/10.3390/electronics12204299

Academic Editors: Cheng-Chi Lee, Tiago Cruz and Bruno Sousa

Received: 4 September 2023
Revised: 10 October 2023
Accepted: 15 October 2023
Published: 17 October 2023

1. Introduction

In today's swiftly evolving digital landscape, the proliferation of technology and interconnected systems has bestowed unmatched convenience and efficiency upon our lives [1]. Nevertheless, this surge in technological advancement has also birthed a parallel surge in cyber threats, with malware intrusion emerging as a profound concern for individuals, organizations, and governments alike [2]. Malware, designed with malicious intent to infiltrate and compromise computer systems, poses a substantial hazard to data security, privacy, and the integrity of systems [3].

As the sophistication of malware continues to burgeon, conventional methods of detection and prevention are proving inadequate in furnishing robust defense mechanisms [4]. This has compelled researchers and cybersecurity practitioners to embark on the exploration and development of advanced techniques and tools that can effectively identify, neutralize, and mitigate the impacts of malware intrusion [5]. The realm of malware detection has undergone a significant evolution, harnessing cutting-edge technologies such as artificial intelligence, machine learning, behavioral analysis, and anomaly detection to maintain an edge over cyber adversaries [6].

The insidious motives of malicious software encompass undermining user computer security and privacy; disrupting computers, servers, clients, or networks; divulging confidential data, unauthorized data, or system access; and obstructing users from accessing information. This malware's reach extends across individuals and large corporations, often referred to interchangeably as Malicious Code (MC) and Malware Executable [7]. Certain strains of malware adeptly remain concealed on a host, exploiting its resources unbeknownst to the user. In an era where protective software is not employed while surfing the web, any user becomes susceptible to encountering malicious software [8]. It is cyber-criminals, often referred to as hackers and hacktivists, who bear the responsibility for crafting such intrusive software [9].

An illustrative example is the 2011 cyber-attack by the Wtz group on Sony's PlayStation network, leading to the exposure of user credit card data and PlayStation account details [10]. Initially, the malware was designed to fulfill rudimentary objectives, rendering it relatively straightforward to identify. This class of malware is commonly termed "traditional" or "simple" malware. In contrast, contemporary iterations of malicious software pose elevated risks due to their ability to operate in kernel mode, rendering their detection more formidable [11]. This next-generation malware can potentially circumvent protective measures operating at the kernel level, including routers and antivirus software [12].

Figure 1 presents the purpose of visually illustrating the evolving landscape of malware detection. It presents data in the form of a chart or diagram and focuses on the cumulative count of detected malware instances, measured in millions, over a 13-year period. Essentially, Figure 1 provides a graphical representation of how the total number of identified malware instances has changed and developed during this specific time frame. It offers readers insights into the trends and patterns in malware detection, helping them understand how the cybersecurity landscape has evolved over the years. Traditionally, malware executed single processes without resorting to elaborate concealment tactics. However, modern malware employs a mosaic of overlapping procedures, both novel and antiquated, accompanied by a wide spectrum of obfuscation techniques to mask its true identity and persist within systems [13]. This latest generation of malicious software is capable of engineering more devastating attacks, including sustained and targeted ones, facilitated by a cocktail of malware types [14]. To discern and combat malicious software, we employ a range of malware identification tools, such as Google Rapid Response (GRR), Wireshark, VirusTotal, and more [15].

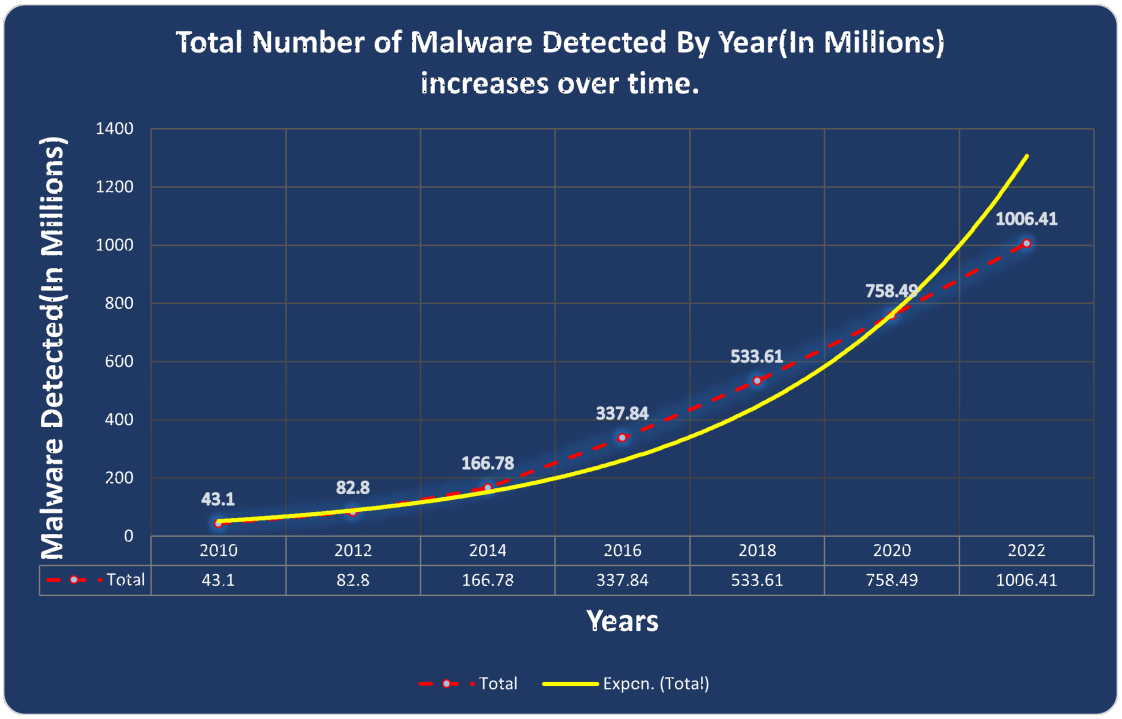

Figure 1. Total Number of Malware Detected by Year (in Millions).

1.1. Related Work

Yadav et al., 2015 [16], delved into the intricacies of the cyber kill chain methodology, examining its technical intricacies. Their analysis encompassed the various stages of this chain, underlining the significance of comprehending and scrutinizing these stages to pre-empt and detect cyber assaults. Landage et al., 2013 [17], provided a thorough survey encompassing diverse malware categories, their distinctive traits, and the gamut of detection techniques. The authors explored the challenges entailed in malware identification and charted a course for future research directions.

Idika et al., 2007 [7], provided a comprehensive survey of the array of techniques deployed for malware detection. The authors traversed signature-based, behavior-based, and hybrid approaches, evaluating their merits and limitations. They also delved into the utilization of machine learning techniques, like decision trees, neural networks, and support vector machines, for malware detection. The survey concluded by highlighting the ever-evolving nature of malware, advocating for perpetual advancement in detection techniques. Aslan et al., 2020 [18], offered an overarching panorama of malware detection methodologies, encompassing signature-based, behavior-based, and machine learning-based paradigms. The authors expounded upon the advantages and constraints of each approach, while also illuminating the avant-garde techniques in the malware detection domain. They accentuated the challenges faced by malware detection systems and furnished forward-looking directions for researchers. This comprehensive overview serves as a pivotal resource for those in the cybersecurity sphere seeking to devise effective malware detection systems.

Souppaya et al., 2013 [19], presented an all-encompassing review of malware incidents and the strategies for preventing them in desktop and laptop systems. The authors traversed various forms of malware, from viruses to trojans, advocating for incident response planning, robust security policies, and user education. The review underscored the best practices for mitigating and managing malware, encompassing the implementation of antivirus software, patch management, and regular data backups. In emphasizing a holistic approach, the authors underscored the synergy of technical and organizational measures for robust malware prevention. Talukder et al., 2020 [20], embarked on a comprehensive survey of malware detection and analysis techniques. Commencing with the rationale for effective malware detection and the concomitant challenges, the authors delved into a gamut of static and dynamic analysis methods. These encompassed signature-based detection, behavior-based identification, sandboxing, memory analysis, and the integration of machine learning for malware detection. The survey concluded by advocating for a multi-pronged approach, as it became evident that a blend of these techniques was imperative for effective malware detection and analysis.

Park et al., 2022 [21], evaluated the efficacy of an open-source Endpoint Detection and Response (EDR) system that amalgamated the Google Rapid Response and osquery for threat detection. The study juxtaposed the system's detection prowess against two other EDR counterparts, assessing its adeptness at identifying malware and suspicious activities. The authors unveiled that the proposed system outperformed its counterparts in detecting both known and unknown threats, underlining the potency of open-source EDR systems and the symbiosis of varied tools and technologies for robust endpoint security. Banerjee et al., 2010 [22], rigorously assessed Wireshark's prowess as an intrusion detection tool. The authors expounded upon Wireshark's functionalities in capturing and analyzing network traffic, advocating for its significance in the modern landscape. They spotlighted the urgency of intrusion detection, elucidated the spectrum of attacks detectable through Wireshark, and illustrated experiments that gauged its efficacy. The findings corroborated Wireshark's utility as an intrusion detection tool, underscoring the necessity of honed expertise to harness its full capabilities.

Masri et al., 2017 [23], proposed an innovative method to unearth malicious advertisements, entailing the amalgamation of outputs from multiple online malware scanners. By integrating data from tools like Virustotal, Urlvoid, and TrendMicro, the authors devised a

mechanism to identify potential threats embedded within online ads. Their methodology's potency was underscored by the empirical test on a dataset of 3000 ad URLs, yielding an accuracy of 99.4%. This approach showcased promise in the automated identification of malevolent advertisements. Souri et al., 2018 [24], created a sweeping survey of malware detection methodologies that harnessed data mining techniques. The authors dissected the limitations of conventional signature-based approaches and pivoted to data mining paradigms such as machine learning and clustering. They juxtaposed the merits and demerits of these approaches, presented a taxonomy of malware detection methods, and elucidated ongoing research challenges. This comprehensive review serves as a compass for those interested in the frontier of malware detection via data mining strategies.

1.2. Contributions

This article outlines several significant contributions in the field of cybersecurity and malware detection. These contributions are as follows:

1. **Categorization and Comparison of Malware Families:** This paper provides a comprehensive categorization of different types of malware. It also offers a detailed comparison of major malware families, likely highlighting their characteristics, propagation methods, and impact on systems. This categorization and comparison aids in understanding the diverse nature of malware and its potential threats.
2. **Intruder's Perspective and Incident Response:** This article delves into the mindset of malicious actors when attempting to compromise a system. This understanding of the intruder's perspective is used to introduce a defense mechanism known as the incident response process. Incident response involves predefined strategies and actions to be taken when a security breach occurs, helping to minimize the impact of ransomware attacks or other security incidents.
3. **Analysis of Malware Detection Systems:** This article systematically reviews various malware detection systems. This involves an analysis of current intrusion detection techniques and tools. By studying these detection systems, this paper aims to provide insights into the effectiveness, limitations, and advancements in identifying and countering malware threats.
4. **AI-Based Solutions for Malware Detection:** This article presents detailed solutions for malware detection that leverage Artificial Intelligence (AI) technologies. These AI-based solutions likely utilize machine learning algorithms to identify patterns and anomalies in system behavior, aiding in the timely detection of malware. This approach is expected to enhance the accuracy and efficiency of malware detection.
5. **Case Studies of Notable Incidents:** This paper includes case studies of specific cybersecurity incidents to illustrate real-world implications and challenges. The examples mentioned, such as the Facebook-Cambridge Analytica data breach controversy and the Cisco system intrusion by the Yanluowang Ransomware Gang, offer insights into the tactics used by threat actors and the consequences of such attacks. These case studies help contextualize the importance of effective cybersecurity measures.

2. Cybersecurity Goals

The "CIA triad" of principles, which consists of confidentiality, integrity, and availability, are the cornerstones of cybersecurity. The CIA's trinity of security infrastructure and procedures is commonly recognized as a paradigm of excellence in terms of the construction of efficient protective systems. These are employed to pinpoint areas for enhancement and protocols for examining problems and formulating feasible solutions. The CIA triad is a widely-recognized model for recognizing the three core components of successful information management: confidentiality, integrity, and availability. These three elements are essential for a business to achieve success. This distinction facilitates security personnel in examining the diverse possibilities for responding to each matter. The security posture of an organization is bolstered and more able to cope with threats when the three criteria are fulfilled.

Types of Malware

1. Polymorphic Virus: These malicious programs possess the ability to autonomously replicate and employ strategies to evade detection, such as polymorphic code that encodes and duplicates itself. Consequently, anti-virus and firewall applications encounter significant challenges when attempting to eliminate them.
2. Worms: Representing the most widely occurring type of malicious code, worms pose a considerable risk to organizations' data security. Unlike viruses, worms can replicate without any assistance from humans. Their primary objective is to propagate rapidly and cause destruction to resources or convert computers into bots.
3. Trojan Horse: Whether inflicting direct harm on the system or granting attackers access to the host, the consequences of Trojan Horse activities are destructive. These malicious entities masquerade as games, wallpapers, or other files from vending packages to clandestinely infiltrate a computer.
4. Spyware: This is primarily designed to observe and communicate host operations or pilfer data of importance to the intruder. Any information accessed by an unauthorized individual is considered stolen, including web browser data, confidential information, and promotional materials.
5. Adware: This involves using code to instantly display or download intrusive adverts, commonly encountered in the form of pop-up windows in web browsers.
6. Remote Access Tools/Trojan (RATs): These enable attackers to acquire unauthorized access and domination of the computer.
7. Rootkit: This is a set of programs utilized to assume control of either the entirety or part of a computer system. Instead of using standard system instruments, they introduce their own set of programs to eliminate any malicious activity before displaying the results.
8. Ransomware: Malicious software infiltrates a host system with the intent of deploying restrictive software or data. If the demanded payment is not generated within the delineated period, the attacker may corrupt the data, auction it off, or divulge it on the dark web.
9. Bot and Botnet: Compromised hosts provide attackers with the infrastructure to launch attacks using the resources of those computers. A system composed of multiple robotic programs is referred to as a botnet. Attackers often utilize such assaults for illicit endeavors, including sending out spam, launching denial-of-service (DoS) attacks, phishing, implanting spyware, stealing personal information, or mining cryptocurrency. This botnet is composed of computer components referred to as "zombies" or "drones" that are instructed by the bot herder or bot expert.
10. Keyloggers: These are any hardware/software system that logs all keystrokes made by a user.
11. Logic Bombs: This type of malicious software remains dormant on a host machine until a triggering event or time frame is reached. When the specified conditions are met, the system executes the pre-programmed task, usually resulting in data erasure or system malfunctions.
12. Backdoor: This is a secondary point of entry to a computing device utilized to bypass traditional system security protocols. Although often overlooked after construction, hackers may embed themselves upon gaining access to a system to acquire administrative authority. Backdoors lack a replication methodology, as they are individualized software components.
13. APTs (Advanced Persistent Threats): The primary objective of these is to gain access to and scrutinize the network to pilfer data while remaining undetected for an extended period. Organizations possessing high-value information, such as those in the military, government, finance, or corporate sectors, are usually targeted. Examples include Fancy Bears from Russia, Lazarus from North Korea, and the Periscope Group from the People's Republic of China [25–29].

Figure 2 presents a visual depiction of the distribution of cyberattacks across different sectors from 2020 to 2021. Notably, it reveals a significant disparity in attack frequencies among various industries. Specifically, the figure highlights that online stores experienced the highest number of cyberattacks during this two-year timeframe, indicating a heightened vulnerability in the e-commerce sector. On the other hand, financial services recorded the lowest incidence of attacks, suggesting a relatively stronger cybersecurity posture in this sector. This information underscores the importance of robust cybersecurity measures for online retail businesses and serves as a valuable insight for organizations seeking to allocate resources effectively to protect against cyber threats in the modern digital landscape.

The utilization of Google Trend analysis played a pivotal role in the generation of data for crafting both Figures 1 and 2. This method allowed for the examination of trends and patterns in user search queries, providing valuable insights into the dynamics of interest or popularity over a specified period. The figures, thus, present a visual representation of the Google Trend data, illustrating the fluctuating levels of interest or engagement with the subject matter under investigation. By employing this analytical approach, the research benefits from a data-driven perspective, offering a nuanced understanding of how user interest evolves, contributing to a more informed interpretation of the findings presented in the Figures 1 and 2.

Table 1 provides an insightful overview of the properties associated with various types of malware. This comprehensive table serves as a valuable resource for understanding the distinctive characteristics and behaviors exhibited by different malware strains. It covers essential aspects such as propagation methods, exploitation techniques, and impact on systems. The tabulated information facilitates comparative analysis, aiding researchers, cybersecurity professionals, and practitioners in developing effective countermeasures and strategies to mitigate the risks posed by diverse malware threats. Overall, Table 1 acts as a reference guide, offering a consolidated view of key attributes crucial for a comprehensive understanding of malware in contemporary computing environments.

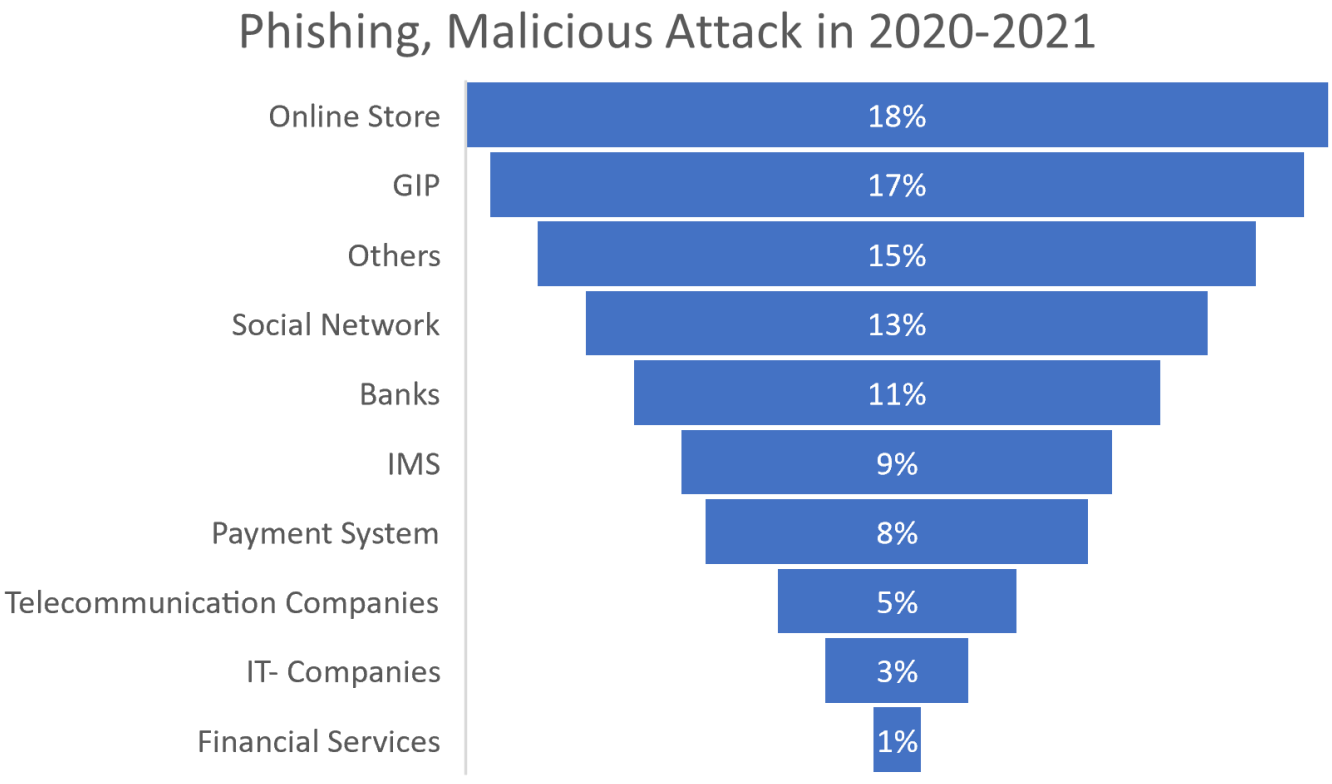

Figure 2. Total number of phishing and malware attacks based on the work categorization.

Table 1. Properties of different types of malware.

Name	Property	Examples
VIRUS [30]	Copies itself to other files; needs a host file to replicate and execute.	CIH, VIRUT, Redlof, peacomm.
WORMS [31]	Exploits the vulnerabilities that are present and can self-propagate over the network.	Code Red, Netsky, Sasser, no_virus.
LOGIC BOMB [32]	Triggers a specific code on stipulated conditions as per the logic programmed by its author.	Michelangelo
BACKDOOR [33]	It is an alternative entrance into a system. They are used to detour the existing security mechanisms built into the systems.	Xhaker, sub7, Beast, Ginwui, Rexob, Hupigon.
TROJAN [34]	A fraudulent program that hoaxes a harmless or useful program, but stores some other malware.	Torpig, Gozi, Pidief, Limbo/NetHell.
SPYWARE [35]	Software used to spy on victim's activities and steal sensitive data.	WhenUSave, PuritySCAN, SecurityToolbar, Virtumonde.
ROOTKIT [36]	Set of programs that alter the OS utilities to hide themselves.	LRK, AFX, SInAR, Rustock, Mebroot.
BOT/BOTNET [37]	Program that does the work on behalf of its handler. The handler may control millions of such bots and can use them for malicious activities.	Agobot, Slackbot, Mybot, Rbot, SdBot, poebot, IRCBot.
APT [38]	A covert attack on a system where the attacker remains undetected for a significant period and maintains unauthorized access.	GhostNet, Stuxnet, APT28, Sykipot APT.
KEYLOGGERS [39]	Any hardware-software combination that records every keystroke a user makes.	Zedlog, Simple Perl Keylogger, Symple Python Keylogger,
ADWARE [40]	Unwanted software designed to put up advertisements up on the user's screen, most often within a web browser.	Appearch, DollarRevenue, Fireball, Gator, DeskAd.
RATs [41]	Allows a hacker to take control of the computer and conduct activities such as exploring files, financial transactions, harvesting login credentials.	Slashtop, GoToMyPC, RemotePC, AnyDesk, Zoho Assist, Connectwise control.
RANSOMWARE [42]	Software designed to restrict access to a computer until a certain amount of money is paid; otherwise, the data are either deleted or leaked on the dark net.	AIDS Trojan, WannaCry, CryptoLocker, Petya, Bad Rabbit, TeslaCrypt.

Table 2 serves as a comprehensive tool for comparing various malware types across different factors. This tabulated comparison facilitates a systematic analysis, allowing researchers, cybersecurity professionals, and practitioners to discern and evaluate key attributes and characteristics of different malware strains. The factors considered in the table provide a holistic view, encompassing aspects such as propagation methods, evasion techniques, and impact on systems. As a valuable resource, Table 2 aids in making informed decisions about cybersecurity strategies, emphasizing the importance of understanding the nuanced differences among malware variants. This structured presentation of information contributes to a deeper comprehension of the diverse landscape of malware threats, ultimately assisting in the development of targeted and effective defense mechanisms.

Table 2. Comparison of Malware

Factors of Comparison	Spyware	Adware	Cookies	Trapdoor	Trojan Horse	Sniffers	Spam	Botnet	Logic Bomb	Worm	Virus
Pattern	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
Obfuscated	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Polymorphic	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Toolkit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Network	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Remote web execution	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
PC	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Removable Disks	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Internet Downloads	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Breaching Confidentiality	✓	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗
Inconveniencing to end users	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗
Denying Services	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓
Data Corruption	✗	✗	✗	✓	✗	✗	✓	✓	✓	✗	✓

3. The Cyber Kill Chain

The "cyber kill chain" paradigm is a beneficial tool for more efficiently responding to and analyzing occurrences of cybercrime. A cyber kill chain is a sequence of steps taken by an attacker to gain entry to a network and execute an attack. In other words, it paints a picture of a malicious actor's activities as they move through the various stages of a cyberattack. In this work, an extensive classification of cyberattack approaches, practices, and hardware is presented. The ultimate resolution of this study is crafted to equip a cyber security professional with an improved comprehension of the possibilities accessible to a perpetrator during an attack.

KILL CHAIN: The set of activities required to exploit the victim must be executed. Typically referred to as a form of malicious software, each distinct attack can lead to a particular sequence of events known as a kill chain.

3.1. Phases of Cyber Kill Chain

There are a total of seven different phases in a cyber kill chain, which are as follows:

1. **Reconnaissance:** Methods of investigating, acknowledging, and identifying targets.
2. **Weaponization:** Constructing an attack vector by amalgamating remote access software with a known security vulnerability, for example, Adobe PDF and MS Office files.
3. **Delivery:** Transference of weaponry to its intended recipient (via email attachments, websites, or USB drivers).
4. **Exploitation:** Upon delivery, the weapon's code is activated, exploiting systems or applications that are susceptible to attack.
5. **Installations:** Installation of a backdoor on a target's system permits continual access.
6. **Command and Control:** The external server facilitates communication between the weapons and the target network by granting "direct keyboard access" within the latter.
7. **Actions on Objectives:** The perpetrator endeavours to realize the objective of incursion, which could involve the extraction or annihilation of data or the infiltration of another objective.

Example: WannaCry malware attack by Lazarus on MS Users in May 2017, demanding ransom in Bitcoin currency [16].

3.2. Incident Response Process

The management of computer security incidents encompasses the continuous monitoring and detection of security events occurring on computers or computer networks.

The dedicated information security or incident management team is tasked with regularly surveilling these security events and deploying the necessary tools to address and mitigate them. Brief information about the three different phases of the incident response is given in Figure 3.

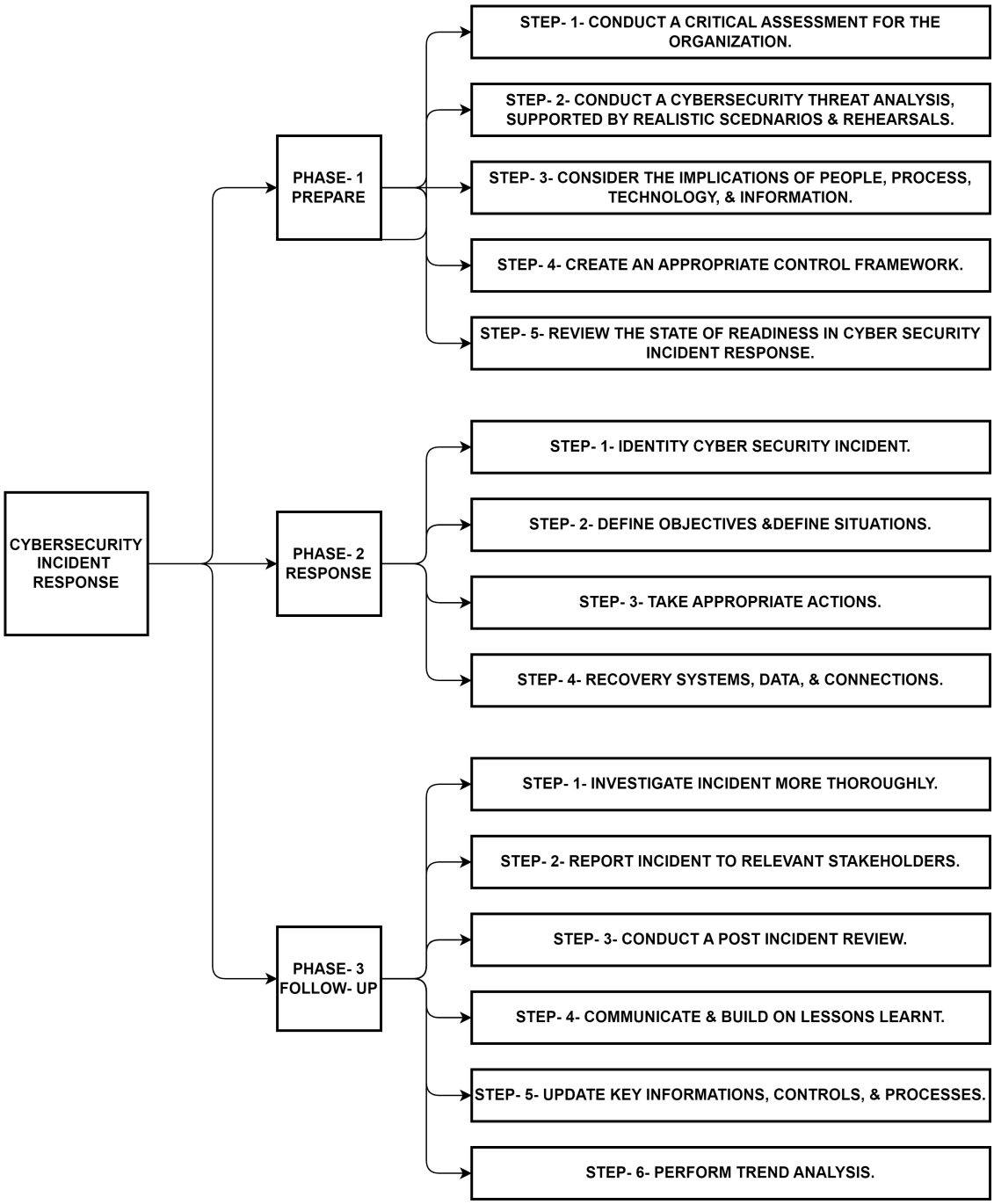

Figure 3. Incident Response Process Flow Chart.

Following are the key elements of Incident Response Process:

1. **Event:** It may be abnormal, deviating from the typical functioning of the network or the accepted practices of the organization, constituting an incident, for example, the access control system was revised and the firewall protocol was amended by a staff member of the enterprise.

2. **Incident:** The detrimental aspect of the occurrence is manifest. For example, if an individual has access to the Access Control List (ACL) and modifies it or blocks all external access to the company's servers, it could have detrimental consequences for the organization's confidentiality, integrity, and availability of its confidential data.
3. **Response Time or CERT:** Identification of the bridge, which is the source of the incident; determining the incident; determining the course of action to be taken to address the incident; and resolving the current issue. Gathering evidence and preserving its continuity of ownership to elucidate the sequence of events is undertaken by them [19,43].

4. Malware Detection

To gain a thorough understanding of the characteristics and behaviours of malicious software and to devise effective detection techniques, it is necessary to undertake rigorous research into malware. Malware analysis is the systematic investigation of malicious software to comprehend its capabilities to formulate a strategy to defend against it, thus safeguarding an organization's computer system. The three distinct approaches to malware analysis can help elucidate the functioning of malware, as well as its effect on the system; however, the tools, time, and skills required for such analysis can vary greatly. Malware can be investigated utilizing two distinct methods: static analysis and dynamic analysis. Static analysis is the initial stage of malware analysis, with dynamic analysis concluding the process. The use of reverse engineering and other analysis tools to identify adware in multiple formats is employed in the analysis of malware [18]. A flowchart of the compilation and reverse engineering process is given in Figure 4.

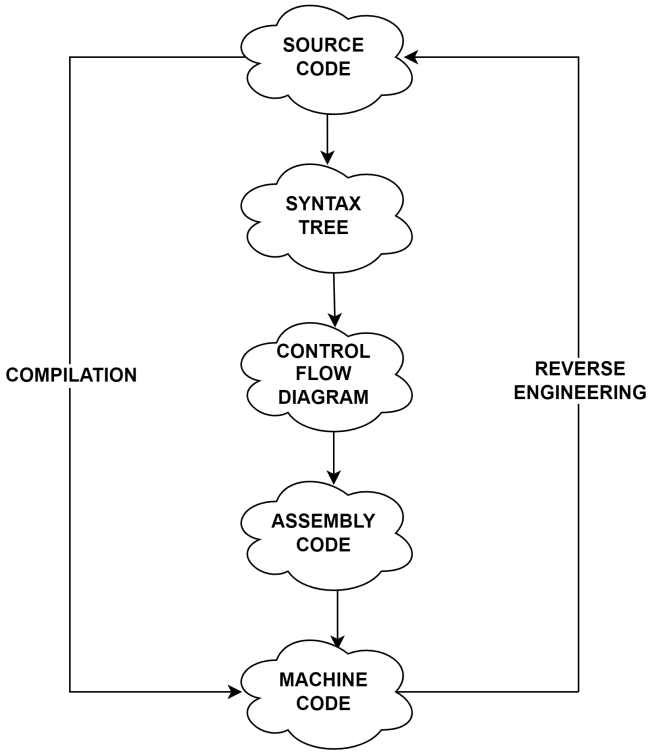

Figure 4. Reverse Engineering Process Flowchart.

4.1. Static Analysis

Static analysis, also referred to as code analysis, is the practice of analyzing an executable program without considering the underlying commands within. A static analysis can be used to determine if a file is malicious, provide insight into its value, and often generate information that can be used to construct clear network signatures. Although

static analysis is comparatively facile to introduce, it can be inadequate in cases of complex software and may fail to detect essential processes [17].

4.2. Dynamic Analysis

Dynamic analysis, sometimes referred to as behavioral analysis, involves the running of malware to observe its behaviour, gain an understanding of its operations, and generate technological indicators for signature recognition. Dynamic analysis yields technological hints such as domain names, IP addresses, file system locations, registry entries, and other documents present in the computer or network [20,44,45].

4.3. Hybrid Analysis

The initial analysis of the distinguishing characteristics of any malicious code is undertaken, and the outcomes are then amalgamated with knowledge concerning the malware's conduct (Table 3). The hybrid analysis technique excels in transcending the boundaries between static and dynamic analysis [17,20].

Table 3. Comparison between static and dynamic analysis.

Static Analysis	Dynamic Analysis
Reliable and quick.	Time consuming & vulnerable.
Efficient in analysing the multipath malware.	The multi-stage virus is hard to analyze.
Cannot analyse the complicated & polymorphic virus.	Can analyse the complicated and polymorphic virus.
Cannot detect new, unknown malware.	Can detect known as well as new malware.
High accuracy.	Low accuracy.

5. Malware Detection Techniques

The purpose of incorporating malware detection methods is to secure computer systems from the potentiality of data loss and vulnerability by detecting and deterring malicious intrusions caused by malicious software. The system is further fortified by the malware detector's endeavours to pinpoint malicious activities. While signature-based, behaviour-based, model-based, and heuristic-based detection, as well as newer techniques, such as deep learning-, cloud-, mobile device-, and Internet of Things-based methods, are effective for quickly and accurately detecting previously identified viruses, signature-based detection systems are inadequate when it comes to identifying unknown malware. Due to the absence of files to evaluate and, as a result, no VirusTotal submission, it can be difficult for security software to detect fileless malware. In contrast, due to the dearth of contemporary research and methodologies, creating an effective secure system presents a daunting task [46]. The various types of malware analysis and their detection techniques have been given in Figures 5 and 6.

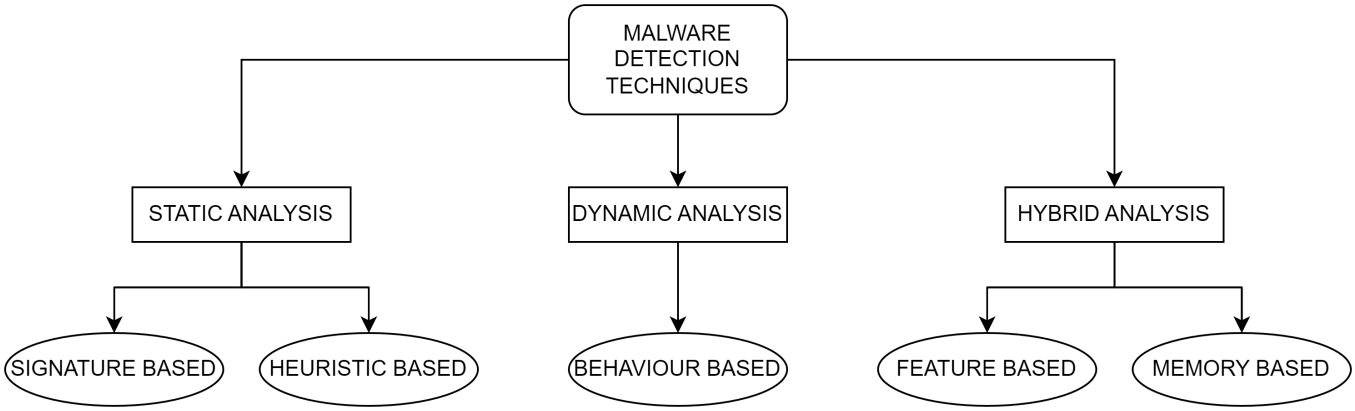

Figure 5. Malware Detection Techniques.

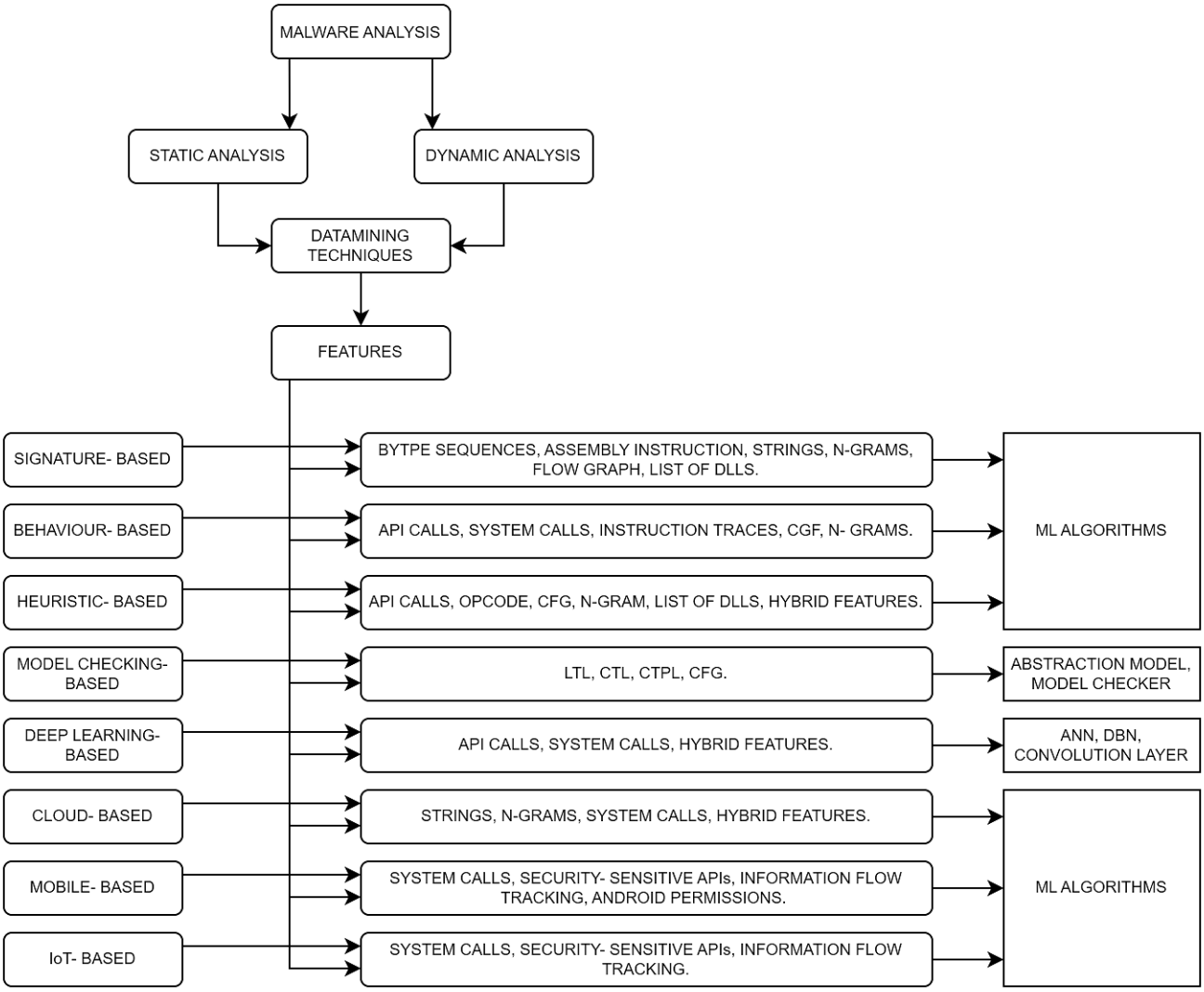

Figure 6. Categorization of Malware Analysis.

5.1. Signature-Based Malware Detection

Malware has a definable feature that is evident in its structure and can be utilized to pinpoint a single instance of an infection. When discussing the application of antiviral software in a business context, the signature-based identification approach is generally the method employed [47]. This technique is fast and accurate at finding common viruses, but it cannot catch anything novel. Malware belonging to the same family can easily evade signature-based identification by employing obfuscation techniques. Executables are the source of the first three signature recognition characteristics. After that, a signature is created by the signature creation system and added to the signature directory. This contrast helps establish whether or not the example program is malicious. Since signature-based detection algorithms can recognize previously found adware so swiftly and reliably, they have been heavily depended upon by antivirus firms. This strategy is frequently employed for the detection of adware belonging to a specific family. Unfortunately, it cannot detect ransomware of the next generation that relies on stealth and code modifications. In addition, it is susceptible to numerous FPs, and signature extraction necessitates substantial human effort [47–49]. The schema for the signature-based malware detection technique is given in Figure 7.

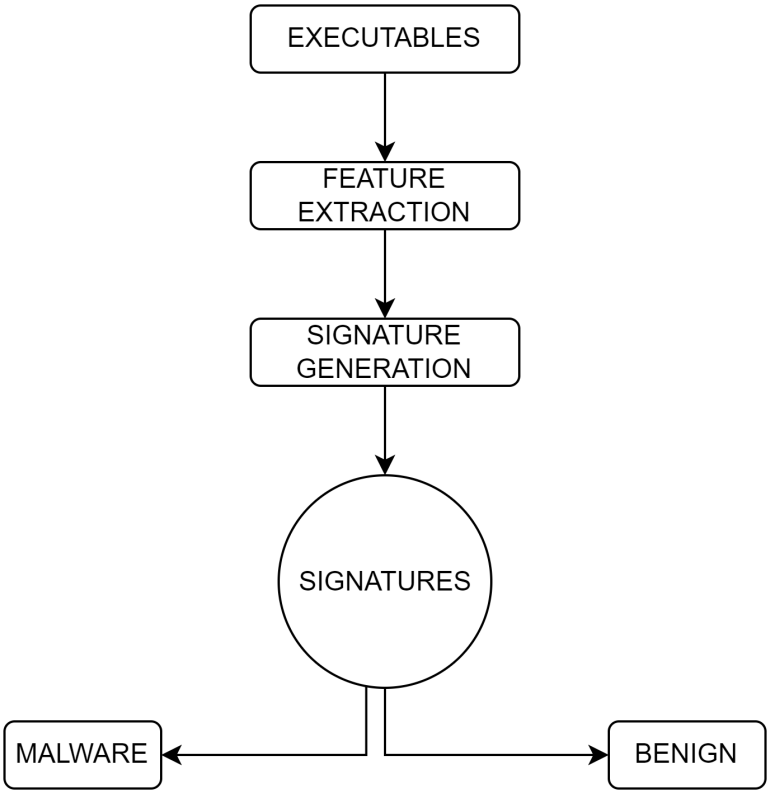

Figure 7. Signature-based Malware Detection Schema.

Assessment of Signature-based Malware Detection

The following methods should be adopted and taken into consideration while building an effective signature.

1. Identities should be as brief as possible and can include multiple virus types under a single signature.
2. A reliable system for automatically generating signatures needs to be built.
3. At some point in the signature technology, data mining and ML tactics should be used more.
4. The signature should be immune to sorting and mystification techniques [50].

5.2. Heuristic-Based Malware Detection

Heuristic analysis is a method for identifying malicious software by examining its signature characteristics. The utilization of conventional virus detection methods to detect malicious software involves the comparison of the code in a program to prior signatures of viruses that have been catalogued in a database. The signature recognition method, though still useful and employed, has seen its effectiveness increasingly limited in light of the emergence of new and contemporary threats since the early 2000s which have persisted into the present. to address this problem, a heuristic approach was created principally to identify the similarity of features between previously known and unknown malware instances. One of the few strategies employed to handle the burgeoning amount of risks presented by the continually developing cyber-criminal milieu is heuristic assessment [46,51,52]. Schema for the heuristic-based malware detection technique is given in Figure 8.

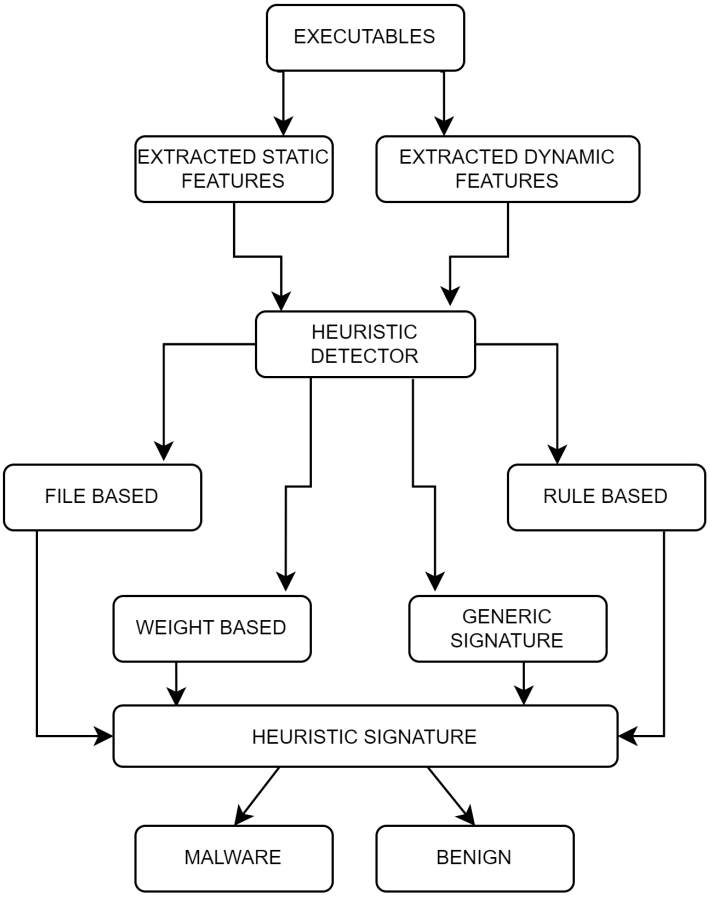

Figure 8. Heuristic-based Malware Detection Schema.

Assessment of Heuristic-Based Malware Detection

A heuristic-based model can be utilized to generate signatures, allowing rules to be constructed from texts and behaviours. This technique employs Application Programming Interface (API) calls, Configuration Files (CFG), n-grams, Opcode sequences, and composite elements to produce a signature. Despite the possibility of heuristic-based detection to detect some previously unidentified malware, it is not able to identify all emerging malware variants. The false positive rate of heuristic-based methods is particularly elevated.

5.3. Behavior-Based Malware Detection

Utilizing tracking tools, the behavior-based malware detection method conducts an analysis of the programmed software's activities to ascertain whether it can be classified as malicious. Despite variations in code, the majority of new malware can still be identified through its behavior, as this has remained constant, despite the fact that malware may not always act regularly when operating in an isolated environment (virtual machine, sandbox environment). Consequently, benevolent codes could erroneously be labeled as malevolent. The proposed technique on Windows OS has the potential to detect both unknown malware and malware that has been previously secured. The proposed technique provides a more extensive perspective on commonplace malicious software behavior, along with the specific activities that malicious software is known to execute. Behavior-based identification commences with the unveiling of behaviors through the application of one of the aforementioned methods, and a dataset is constructed by extracting the characteristics through data mining. Subsequently, ML algorithms are employed to select and label specific features of the information [48,53,54]. The schema for the behavior-based malware detection technique is given in Figure 9.

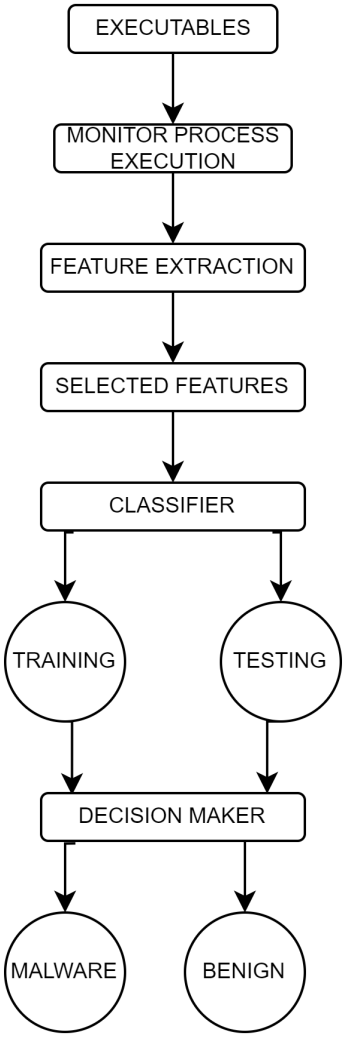

Figure 9. Behavior-based Malware Detection Schema.

Behavior-based malware detection involves a multi-step process for effective identification and classification. The detection schema typically consists of three main steps, each leveraging different techniques and methodologies.

5.3.1. Identifying Behaviors

In the first step, the focus is on identifying behaviors associated with potential malware. This involves observing and analyzing the actions and patterns exhibited by files or processes. Data mining techniques play a crucial role in this phase, helping to sift through vast datasets to identify anomalies or patterns indicative of malicious behavior.

5.3.2. Incorporating Characteristics from Behavior

Once behaviors are identified, the next step involves extracting and incorporating relevant characteristics. This step is crucial for creating a comprehensive profile of potential threats. Data mining continues to be a valuable tool in this phase, aiding in the extraction of features and attributes associated with malicious behavior. These characteristics may include patterns of system calls, file interactions, network activities, or other observable behaviors.

5.3.3. Utilizing Classification Techniques

The final step in the detection schema is the utilization of classification techniques. Machine learning, a subset of artificial intelligence, is commonly employed in this phase to

automatically categorize and differentiate between benign and malicious behavior. The system learns from the extracted characteristics and behaviors identified in the earlier steps, creating models that can make informed decisions about the nature of a given file or process.

The integration of these three steps forms a comprehensive approach to behavior-based malware detection. It not only allows for the identification of potential threats, but also enhances the system's ability to adapt and recognize emerging patterns of malicious behavior.

5.3.4. Data Mining-Based Approaches

This holistic approach to malware detection, as outlined by Chakravarty et al. in their study [49], showcases the synergy of data mining in behavior identification, the importance of incorporating diverse characteristics, and the effectiveness of machine learning in automating the classification process. As the threat landscape evolves, the adaptability and learning capabilities of behavior-based detection systems become increasingly crucial in ensuring robust cybersecurity measures.

Indeed, data mining techniques play a vital role in extracting characteristics from behaviors in the context of malware detection. The mentioned techniques, including n-gram, n-tuple, bag, graph models, etc., are commonly employed to analyze and derive meaningful features from observed behaviors. Each technique offers a unique perspective on behavior analysis, contributing to a more comprehensive understanding of potential threats. The study by Souri in 2018 [24] highlights the significance of these techniques in the field of cybersecurity.

6. AI-Based Malware Detection

6.1. Signature-Based Malware Detection

It involves the generation of unique signatures for each known malware to be compiled into malware behaviour libraries. Experts can manually identify these signatures, which may also be produced using automatic approaches and may contain a range of data, such as filenames, string values, or bytes.

The signature of unfamiliar software can be subjected to a comparison with the library of malware behaviour to ascertain whether any corresponding signatures can be identified. This detection method is the most commonplace and is utilized with a swift detection rate and a low rate of false alarms. Signature-based malware detection is rendered ineffective for newly emerging malware, just as misuse intrusion detection is. The malware library must be kept up-to-date and regularly managed. The starting point of the detection or prevention of malicious activity necessitates that a victim first reports the incident. When the initial victim is of great significance, the outcomes may be intolerable. The 2015 U.S. Office of Personnel Management vulnerability in critical infrastructure could potentially lead to several repercussions that could extend for decades.

Different machine learning techniques have been utilized for malware detection. These ML techniques opt to decompose the malware to extract pertinent information from the software, thus facilitating the models to detect malicious software. The selection of suitable input data is critical to the successful deployment of Artificial Intelligence (AI)-based malware detection.

We conduct an analysis of various exemplar ML malware detection schemes and a comparison of many alternative software information options chosen by researchers.

6.2. Network Behaviors

Wireless Multimedia System (WMS) can be utilized to maintain a constant monitoring of data and manage the condition of distant apparatuses. Wireless multimedia devices typically feature multiple sensors, which facilitate the transmission of data to adjacent nodes according to established routing protocols. The decentralised architecture of WMS facilitates the propagation of malicious software, thus posing a risk to other nodes, wireless routers, and terminals through data exchange. The acquisition of network behaviours

is imperative for the identification of WMS malware. A malware detection scheme was devised which enabled the acquisition of network behavior in WMS using the data sniffer (DroidSniffer) in conjunction with Support Vector Machines (SVM) and Backpropagation Neural Networks (BPNN) for the detection and elimination of malicious codes. The experiment yielded an infection rate of 22.17%, demonstrating that malware can be identified even at a comparatively low incidence [55].

6.3. APK and API

The Android platform is an essential factor in facilitating the expeditious growth of Internet of Things (IoT) applications. Concomitantly, the prevalence of malware in the Android operating system has increased, and the advent of virus strains incorporating highly advanced evasion tactics has been observed. An ensemble learning-based Android malware detection solution of high precision was developed by Yerima et al., 2015 [56]. The analysis of Android malware necessitates the examination of the software features obtainable from the APK. Utilizing Java-based APK analysis tools, one can extract the set of features from the existing app corpus. This study focused on extracting 65 features, comprising a range of API calls and Linux/Android command sets. The following Application Programming Interfaces (APIs) are included: SMS Manager API (utilized for sending, receiving, and reading SMS messages, etc.); Phone Manager API (for accessing device ID, subscriber ID, network operator, and SIM serial number, etc.); and the Package Management API (used to list installed packages) [56]. It has been asserted that the extraction of sensitive data streams from within the application can be an effective means of detecting malicious software. The authors developed DeepFlow, an Android malware detection tool, and implemented a technique involving the analysis of Android Application Programming Interface (API) codes with APK packages, the extraction of sensitive data streams, and the utilization of Deep Belief Networks (DBN) for classification. The results of an experiment conducted on 3,000 benign apps and 8,000 malicious apps revealed that DeepFlow achieved a noteworthy F1 score of 95.05% when appropriately configured [57].

6.4. Binary Image

A novel approach to the detection of malicious software is the analysis of the binary representation of said software. Software binary files can be reformatted into an 8-bit sequence and subsequently converted into a grayscale image, which comprises a single channel and has pixel values ranging from 0 to 255. The experiment yielded converted images which highlighted the conspicuous disparities in the structural components of benign and malicious software images. Mirai malware images often tend to have higher density in the central region of the image, providing an example of the general rule that malware images tend to be dense. The disparity between the binary images of multiple software applications has been leveraged to render the detection of malware into a classification problem via image recognition, thereby enabling a Convolutional Neural Network (CNN) to differentiate between benign and malicious software [58]. Following are the key elements of Incident Response Process.

6.5. Opcode and Graph

The utilization of opcodes as a feature in machine learning models can be a suitable and reliable approach for identifying malware. The successful amalgamation of Windows malware opcodes with machine learning techniques has been demonstrated by many researchers to effectively detect malware. The selected features (opcodes) of each sample (software) were transformed into a graph (Figure 10). The graph created by Azmoodeh et al., 2018 [59], depicted nodes as opcodes, with edges representing the affinity of each node (which was computed) in the disassembly of the respective software [59]. Graphs can be transformed into an Eigenspace, thus making it possible for Convolutional Neural Networks (CNNs) to be utilized for the classification of malicious and benign software graphs. Azmoodeh conducted an experiment in which the opcode sequences of 1078 pieces

of legitimate software and 128 pieces of malicious software were extracted. The analysis of malware using graphs converted from opcodes yielded a detection accuracy of 99.68% and a recall rate of 98.37%. The efficacy of this technique for detecting malicious software is remarkable [60]. Table 4 provides a comparison of different malware detection schemes.

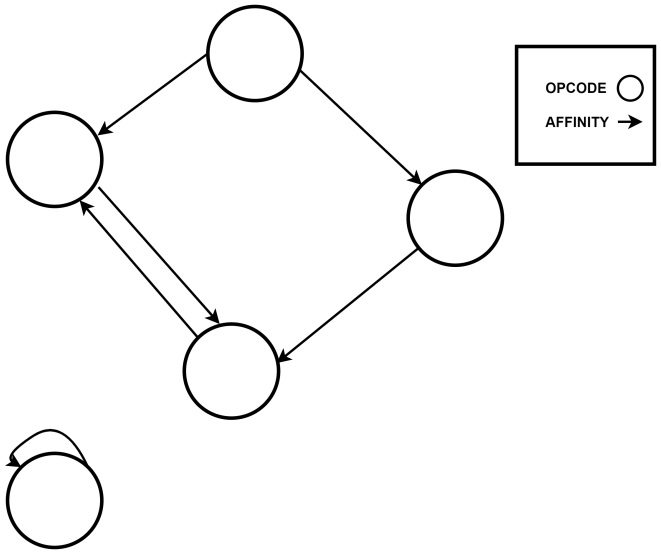

Figure 10. Graph transformed from opcode representations.

Table 4. Evaluation of Approaches for Detecting Malicious Software.

Scheme	Software Information	Malware Type	ML Methods
Signature-based Detection [61]	A signature comprising software details developed either automatically or manually.	No restrictions	No ML methods used.
Network Behaviours [55]	Network Behaviours	WMS Malware	SVM; BP Neural Network
APK and API [56,57]	APK, API	Android Malware	Ensemble Learning
Binary Image [58]	Binary Image	Telnet Attack Software	CNN
Opcode and Graph [59,60]	Opcode	Windows Malware	CNN

6.6. Active Immunity for Malware Detection

The application of adversarial machine learning to malware detection provides a novel approach to augment the existing methods of extracting software information and implementing algorithms to detect malicious code. It is imperative to be cognizant of the possibility of adversarial manipulation when utilizing machine learning algorithms. The perpetrator will take advantage of the vulnerabilities of machine learning to evade detection. If the detector is capable of anticipating the likely evasion route of the attacker, it can drastically shorten the latency of detection and mitigate the damages caused by unknown attacks, thereby achieving an "active immunity".

Chen et al., 2017 [62], referred to the dynamic between evasion attack and defense as an "arms race". To begin with, they developed an efficacious evasion model (EvnAttack) by simulating the behavior of potential attackers. to efficiently address this type of evasion attack, they proposed a malware detection learning paradigm (SecDefender) that takes into account the cost associated with the attacker's evasion attack. The efficacy of this method was demonstrated through extensive experimentation on the actual datasets of the Comodo Cloud Security Center. Wu et al., 2018 [63], noted that malignant software can circumvent machine learning detection techniques by periodically varying its structure while retaining its malicious functions [62,63].

Reinforcement learning can be utilized to continually generate simulated attackers to create new malware samples, thereby providing potential attack vectors for defend-

ers to consider. They constructed a reinforcement learning-based model to increase the likelihood of these freshly created malware to avoid ML models, and then re-trained the detection model utilizing these recently generated examples. The results of the experiments conducted showed that the detection accuracy of malware was substantially increased from 15.75% to 93.5% following retraining, significantly enhancing the detection model's capacity to detect unfamiliar attacks.

Demontis et al., 2017 [64], examined the pre-existing attack frameworks, and synthesised and organized the current attacker's objectives, information, attack strategies, and potential attack scenarios. Subsequently, they undertook a series of evasion techniques to assess the efficacy of malicious software detectors. It was demonstrated that by employing linear and nonlinear classifiers with feature weights that are uniformly distributed, the security of the system can be improved without significantly compromising the computational efficiency [61,64–67].

7. Malware Detection Tools

Malware analysis tools are utilized by researchers to facilitate the sharing of information, make predictions about potential attacks, and develop countermeasures for defense. Open-source software is often the preferred selection for this type of venture. Malware distribution has become a fast-growing industry, and analysts forecast that the widespread, convoluted issue of malware will only become further aggravated in the foreseeable future. Throughout the attack, researchers will employ open-source malware analysis tools to identify and record multiple prospective hazardous events. The increasing availability of cryptors, botnets, and zero-day vulnerabilities is a repercussion of the burgeoning number of malware trading platforms on the dark web. The sophistication of malware is escalating, making the comprehension and measurement of its presence more difficult. Cybersecurity experts must ascertain the most appropriate approach for examining a specified attack [68,69].

7.1. Google Rapid Response (GRR)

Google's security specialists designed the GRR platform, an advanced remote forensics incident response platform. Signs of malware infiltration on desktop machines are commonly discernible. For agent-to-agent communication to occur, both a server architecture and a client program running on the client computer must be present. The GRR system encompasses a Python server architecture dedicated to the monitoring of clients and the establishment of interaction with them, in addition to a Python-based client (or "agent") installed on target machines. Once the configuration of the server and proxy have been established, an individual utilizing the GRR platform can obtain the privilege of receiving server-generated transmissions. The subsequent action for the incident response team is to execute a sequence of technological operations on the main computer, comprising assessing the memory, inquiring into diverse configuration alternatives, and engaging with program selections. GRR can be utilized by researchers to expeditiously gather data from a large number of computers, as it has been designed to function in such an environment. to assist in criminal investigations and inquiries, GRR has been implemented to be both user-friendly and versatile, thus promoting the quick assessment of scenarios and enabling remote examination.

By default, all interactions between the GRR client and server are secured by employing the Advanced Encryption Standard (AES) 256 and transmitted over the Hypertext Transfer Protocol (HTTP). The GRR server comprises a front-end server, a workforce, and a user interface, with flow and search representing fundamental operations. to address matters about limited resources, the GRR server employs flows, a form of finite automata. The transmission of information between the server and the client is the most integral element of a server, as it is responsible for establishing communication. The client must take action to initiate traffic on the GRR server initially. While the server awaits a response from the client, it configures all of its constituents. Once a response has been ascertained,

the implementation of the flow state occurs, following which the relevant resource is downloaded. The GRR can be employed to address concerns about the overutilization of resources. Conducting a scan across hundreds of client machines is a process known as "scoping". The selection of computers to facilitate the various stages of the workflow is determined via the process of hunting [20,21,70].

7.2. Wireshark

It is an indisputable fact that Wireshark is ubiquitous in the analysis of networks, no matter the context. This software is compatible with a wide array of operating systems, including Windows, OS X, Linux, and UNIX, and is equipped with a comprehensive range of features. There is a regular reliance upon it by many people, which includes those in the fields of networking and security, software development, and education. It is accessible to the public without charge and distributed according to version 2 of the GNU General Public License. This example of innovative technology was engineered and implemented by protocol specialists from around the world.

Wireshark is a software program employed to visually display packet traffic on networks in real-time, rendering it comprehensible to the human user. It monitors and interprets digital information for comprehension by humans. Utilizing Wireshark's filtering and colour-coding abilities, users can analyze individual frames of network data. This is an invaluable asset for any dedicated network or system engineer, as it provides unparalleled data analysis of network information. If experiencing difficulty in resolving a network issue, one can employ the use of this gratuitous application that permits users to observe network activity in real-time. Wireshark is a useful tool for resolving common problems such as degraded connection reliability, slow data transfer speeds, and malicious network activity. Utilizing these resources, it is possible to meticulously evaluate network data, thus enabling researchers to target the fundamental cause of the problem. Management employs it to discern data exfiltration or hacking attempts against any firm, as well as to pinpoint damaged network hardware that is disseminating messages [22,71–73].

7.3. VirusTotal

VirusTotal is an economically accessible resource for assessing the security of files, passwords, and addresses. It can be accessed both via a computer application and an internet-based platform. The deployment of multiple antivirus algorithms facilitates the detection of a range of malicious entities such as viruses, worms, trojans, and other forms of adware. VirusTotal utilizes more than 70 anti-malware analyzers and URL/domain blacklisting services in its inspections, in addition to other methods for discriminating against dubious matters. VirusTotal enables the capability for any individual with a computer to transmit a file from their computing device. Apart from its primary public interface, PC uploaders, browser extensions, and an Application Programming Interface (API) are all viable avenues for the transmission of data to VirusTotal. The web UI receives the greatest consideration when examining the various publicly available registration forms. The specifications for the public, HTTP-based protocol can be formulated in any language. In addition to its primary task, VirusTotal also comprises several auxiliary tools, including the VirusTotal forum, which allows users to submit reports on files and addresses and to exchange interpretations of the outcomes amongst each other. This can be beneficial in distinguishing between potentially malicious data and false positives, which are those benign items that have erroneously been labeled as hazardous by the security system [23].

7.4. Comparison of Malware Detection Tools

In Table 5, a comparison of some well-known malware detection tools is presented. Each tool has its unique strengths and focus areas. Google Rapid Response (GRR) excels in incident response and live forensics, making it ideal for organizations prioritizing rapid detection and response to security incidents. Wireshark stands out as a powerful network protocol analyzer, suitable for those requiring in-depth analysis of network traffic.

VirusTotal serves as a convenient web-based service for file and URL analysis, making it accessible to a broad audience and effective for post-event analysis. Suricata and Snort specialize in network intrusion detection and prevention, with high-performance engines and rule-based detection, making them well-suited for network-focused security. Sophos Intercept X provides robust endpoint protection, leveraging a combination of signature-based detection, machine learning, and behavioral analysis, making it a comprehensive solution for protecting individual devices. The choice among these tools depends on specific needs, with GRR and Wireshark excelling in incident and network analysis, respectively, whereas VirusTotal and Sophos Intercept X offer accessible and comprehensive solutions for broader audiences and endpoint protection. Organizations with a network-centric focus may find Suricata and Snort particularly valuable.

Table 5. Malware Detection Tools Comparison.

Tool	Focus	Features	Limitations
GRR	Incident Response, Live Forensics	Remote Investigation, Scalable, Automation	Learning Curve, Configuration
Wireshark	Network Protocol Analysis	Packet-level Analysis, Wide Protocol Support	Networking Expertise, Limited Malware Coverage
VirusTotal	Web-based Malware Analysis	Aggregate Results, Web Interface, Community-driven	Relies on Signatures, Limited to Files and URLs
Suricata	Network IDS/IPS	High Performance, Multi-threaded, Rule-based	Configuration Expertise, Network-focused
Snort	Network IDS/IPS	Rule-based Detection, Regular Updates	Regular Rule Updates, Network-focused
Sophos Intercept X	Endpoint Protection	Real-time Protection, Exploit Prevention, Centralized Management	Endpoint-specific, Limited Network Coverage

8. Case Studies

8.1. Case Study 1: Facebook Cambridge Analytica Data Breach Scandal

Cambridge Analytica, a London, United Kingdom-based data analytics, marketing, and consulting firm, is under suspicion for the alleged illegal collection of Facebook data and the purported utilization of said data in the selection of numerous political campaigns. U.S. Senator Ted Cruz's campaign and, to a lesser extent, Donald Trump's campaign, as well as the Leave-EU Brexit movement, which ultimately led to the United Kingdom's withdrawal from the European Union, are illustrative examples of this. In 2018, the Cambridge Analytica data controversy, which involved Facebook, was a major source of embarrassment due to the firm's inappropriate acquisition and utilization of private information from millions of Facebook users' accounts for political action groups. It can be argued that the significant decrease (17%) in Facebook's valuation and the subsequent call for more stringent laws to govern technology companies' handling of personal data stemmed from this "turning point" in the public's perception of confidential information.

8.1.1. Background Information

Within a brief period, Kogan was successful in obtaining data from approximately 87 million Facebook profiles, representing approximately 23.8 percent of all Facebook members in the United States. The Trump campaign team encountered difficulty in regards to leveraging the data to stimulate voters with political messaging. Kogan's research was intended for academic purposes; however, his provision of access to the resultant data to Cambridge Analytica was in contravention of Facebook's regulations. Mark Zuckerberg, the Chief Executive Officer of Facebook, defended the offence by postulating that it was not a data breach, but rather a violation of the agreement between Facebook and its users, since no passwords were pilfered nor any systems were damaged. The United States Federal Trade Commission commenced its investigations shortly thereafter [74].

8.1.2. Facebook Data Breach

In December 2015, Harry Davies of the Times brought to light that Cambridge Analytica (CA) had procured confidential information without consent. Harry asserted that, while working for the United States Senator Ted Cruz, CA had illicitly accessed the data of millions of Facebook users. Facebook stated that it was the united report and was unable to provide any supplementary details. In March of 2018, the issue was only revealed after a former CA employee, Christopher Wylie, was exposed as an instigator. Christopher was referred to pseudonymously in Cadwalladr's 2017 exposé as "The Big British Brexit Theft". Certain individuals were dubious about the authenticity of the narrative, resulting in distrustful responses from outlets such as The New York Times. The simultaneous publication of the articles in March of 2018 set off a heated response that rapidly caused a devaluation of Facebook's market capitalization by $100 billion. Mark Zuckerberg, Chief Executive Officer of Facebook, has been requested to provide clarifications by senators from the United States and Great Britain. In response to the widespread public outcry, Mark Zuckerberg consented to testify before the United States Congress [75].

8.1.3. Summary of the Case

The Strategic Communication Laboratories Group, the parent organization of CA, was a private British corporation focused on research in the fields of behavior and strategic communication. Aleksandr Kogan, a university researcher, was mandated with the task of creating a software application entitled "This is your digital life", and he was additionally tasked with building a survey that focused on the usage patterns of the individuals he acquired from Facebook's social media platform, to utilize the acquired data for electoral/political objectives without the user's consent. The utilization of data mining and data analysis by SCL to acquire data from its users provoked indignation in the United States and other countries. Based on the research, data would be carefully crafted to important target groups to affect behaviour by the aim of SCL's client, which would threaten the relationship based on trust between Facebook and its users.

8.1.4. Legal Implications

Subsequently, the Facebook CEO was interrogated, resulting in a seventeen percent (17%) decrease in the company's stock price. A graphical representation of how Facebook stocks tanked after the data breach report, shaving billions off company's market value, has been shown in Figure 11. Moreover, he was requested to implement strict regulations for the maintenance of user information. The users were apprised of the revocation of the access they had earlier authorized to several applications, which had been reviewed in the settings, and that trial assessments of the intrusion investigation protocol had been conducted. Facebook intends to introduce an application that, upon installation, will require users to delete all of their browsing data. CA has faced numerous allegations in the past which are baseless, yet despite the company's endeavours to become more transparent, it has still garnered censure for practices that, while being legally permissible, are generally accepted as normative aspects of internet advertising in both the public and private sectors.

To examine the purported disparities, CA contracted an external auditor, Julian Malins, to conduct an investigation. The business's investigation demonstrated that the allegations were unfounded. Despite CA's continuous assurance that its personnel acted ethically and legally, there has been a significant deterioration in its customer and supplier base in response to the coverage it received in mainstream media sources. Following this, a resolution was reached in May 2018 that the business could not be sustained, thus leaving CA with no feasible solutions for the government to take it over.

The enforcement of the General Data Protection Regulation (GDPR) in May 2018 enabled the formation of standardized processes for ensuring the security of personal data across the European Union. All entities that accumulate personal information on inhabitants residing within the European Union are subject to impact globally. All activities that an organization may carry out with personal data, from resolving grievances to

archiving data to utilizing such data and then disposing of it, are encompassed under the expression "processing".

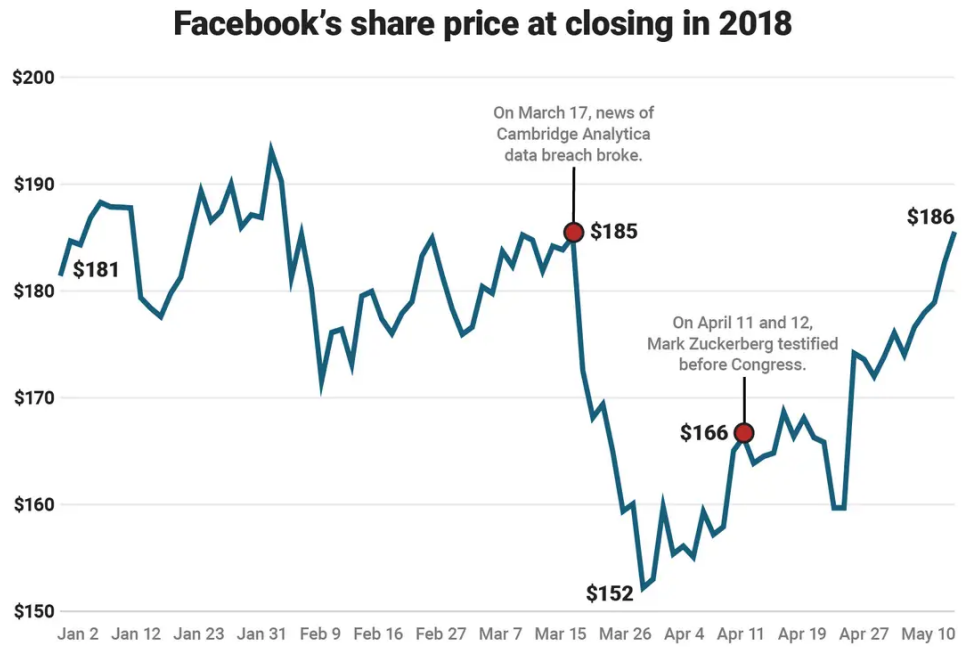

Figure 11. Facebook Shares' Price Drop.

Despite having numerous similarities to pre-existent EU data protection laws, the GDPR is distinguished by its greater comprehensiveness, augmented minimum standards, and more stringent penalties. Two illustrative examples are that this enhances individuals' capacity to access and alter their data, as well as instituting more stringent restrictions on the utilization of confidential data. Non-compliance with GDPR may incur substantial fines of up to 4% of an individual's gross income in the event of multiple offences or infringements. Regarding policy alterations, the information can only be accessed by individuals external to the organization, including coders. A petition for access results in an intensification of data regulations and a more thorough assessment of the request using an investigative tool [76].

8.1.5. Conclusions

The platform user must be aware of the types of apps and confidential data to which they have access, regardless of how often a particular program is changed or updated. Surveillance is essential to safeguard sensitive information and to be cognizant of potential repercussions stemming from a data breach. Examples of activities that could be analyzed include blocking access to illicit software and conducting regular evaluations of settings. This is exemplified by the case of CA. The government must institute a stringent regulatory framework that effectively curtails the activities of CA-based companies, thus curtailing the indiscriminate international utilization of social media user data. It is not possible to guarantee that a government entity will not exploit its access to contemporary technology. It will probably occur at this juncture.

9. Case Study 2: Cisco System Breach by Yanluowang Ransomware Gang

Cisco Systems, the largest provider of networking solutions in the world, experienced a security breach. In an announcement released on Wednesday, 10 August 2022, Cisco Systems divulged that a cyber intrusion had taken place at the company on 24 May 2022. The networking equipment manufacturer divulged its discoveries, announcing that the cyber criminals had attained entry to an employee's Google account through the synchronization of the account's passwords with Cisco's web browser.

The company revealed that the Yanluowang ransomware group had used a Google account that had been hacked by one of their employees to gain access to their system, subsequently posting a list of files they had obtained on their breach notification website.

Due to the security breach in the Google account, the aggressors were able to gain access to the corporation's virtual private network. The individual in question had linked their Cisco login information to the Chrome web browser, where they had stored them previously. Consequently, assailants could utilize this information to harmonize their Google accounts. On the 10th of August, the Yanluowang ransomware collective released documents that were taken during the security breach, tacitly implying culpability for the infraction. The malicious actor indicated that a total of 2.75 gigabytes of data, which encompassed 3100 individual files, had been pilfered. A profusion of these records encompasses non-disclosure agreements, data disgorgements, and technical representations.

9.1. Yanluowang: The 10 Kings of Hell

The UNC 2447 organization, more commonly known as the Yanlowang gang, bears responsibility for the attack on Cisco. Contrary to popular belief, it is difficult to ascribe most security threats, such as malware networks, to a particular nation. It is inappropriate to ascribe any Chinese association to Yanluowang solely based on the fact that some of his works have been related to Chinese concepts or products. Despite any potential connections to China that the creator of the ransomware may possess, it does not necessarily follow that the group is motivated by anything other than monetary reward. Cisco has provided evidence in the form of UNC2447, the initial access broker hypothesized to be the perpetrator in the actual infiltration, which appears to suggest a link to Russia. It is conjectured that Lapsus$, which has been tied to both UNC2447 and Yanluowang, is situated in Brazil, making any military action, supported by a sovereign state, more intricate. The current consensus is that the August 2021 release of Yanluowang was the result of the illicit ransomware-as-a-service operations of the criminal organizations Five Hands and Thieflock. Upon the realization of the Symantec Security Hunter Team that Yanluowang was engaging in targeting American institutions in 2021, numerous parallels were noticed between the tools, methods, and procedures employed by Yanluowang and those used by Thieflock. It is possible that an individual who has previously been associated with Thieflock could be culpable for Yanluowang.

9.2. How Attackers Bypassed MFA

Cisco asserts that cyber criminals have adopted a range of tactics to bypass the multi-factor authentication measures of the VPN client. A diversity of elements can be attributed to the weariness experienced by users of a multi-factor authentication system, such as vishing (or speech hacking). When a perpetrator continually inundates a recipient device with push notifications, the user ultimately succumbs to MFA exhaustion, ceasing to employ the authentication measure completely. Cisco Talos analysts identified that employees of the enterprise had been the victims of a successful Multi-factor Authentication (MFA) hacking attempt, thus permitting unauthorized access to the company's VPN software. Upon gaining access, they carried out several verifications of recently installed multi-factor authentication devices that were then connected to the enterprise's virtual private network. Subsequently, the assailant advanced quickly to a managerial role. Subsequently, they could access multiple platforms. This precipitated the activation of the Cisco Security Incident Response Team to intercede and reduce the magnitude of the harm. The investigation revealed that the malicious organization utilized aggressive cybersecurity procedures and digital access. These instruments included:

1. Team Viewer
2. LogMein

A study published by Cisco elucidated the outcomes of enforcing the alteration of passwords across all corporate networks. The company has generated two distinct signatures for the Clam Antivirus security solution to prevent any potential security breaches.

9.3. Post-Gaining-Access

Once accessing the system, the perpetrators engaged in activities to sustain the system's operation, impede potential forensic investigations, and gain additional privileges. They commenced to list by employing the normal Windows software to locate the user and group configurations, address, and other relevant factors of the system.

Following the infiltration of password databases, the perpetrator was observed to be utilizing computer identities to attain heightened authorization and effectuate lateral movement. The perpetrator created a distinct executive user account named "z" using the in-built "net.exe" command and added it to the local administrator groups.

This identity was employed to try to scan the directory services landscape and acquire extra passwords with the aid of tools such as adfind and secrets dump. In addition to compromising the SAM database, the perpetrator further eradicated the registration details from the purloined device. The intruder employed the MiniDump process on certain servers to exfiltrate LSASS (Local Security Authority Subsystem Service).

9.4. Clearing Tracks

Once the breach was detected, the local administrator account was removed and the "wevtutil.exe" program was employed to purge the event logs. The hosts' security configurations were modified to permit the Remote Desktop Protocol (RDP). Software such as TeamViewer and LogMein, which facilitate remote access, were also implemented and utilized. The perpetrator utilized Windows password circumvention techniques to maintain ongoing administrative access to all the machines in the network. It was common for them to employ PSEXESVC.exe to make direct modifications to the registry settings. By exploiting the accessibility features of Windows, the attacker employed "narrator.exe" and "sethc.exe" as targeted entry points. This strategic maneuver facilitated the launching of a root-level command prompt, ultimately granting the attacker full authority over the targeted machines.

It is noteworthy that, according to the Cisco Talos assessment, the unauthorized actors had entered the keys as specified, yet had not taken any additional steps on the system. Consequently, a storage procedure could be retained for potential utilization once their access had been withdrawn.

9.5. Aftermath

A cyber-criminal collective requested remuneration in exchange for the non-disclosure of pilfered Cisco data. In this instance, the trepidation of divulging the purloined information to the public was rendered unnecessary due to Cisco having no utility for it. Due to the confidential nature of the data, companies are often required to make payments to ensure its secrecy. If a business entity accedes to extortion demands, the purloined information will likely be made accessible for purchase on the dark web.

9.6. Summary

Cisco emphasizes that there was no evidence of malware being utilized during the attack. The Cisco Security Incident Response Team (CSIRT) ascertained that there have been no detrimental outcomes to the business, products and services, confidential client and personnel information, intellectual property, or supply chain activities of Cisco due to this incident. On the 10th of August, the perpetrators uploaded an overview of the hacked data to the hidden web. The company's frankness and transparency in disclosing the loss and consequent password alteration should be commended. Kaspersky has exhibited a pronounced fascination with the collective and the deleterious malware that is ransomware. In April, a vulnerability in the RSA-1024 encryption protocol implemented by the Yanluowang software was exploited, which allowed for the decryption of the running data. Thus, if a victim has one or two files that have not been encrypted, the Kaspersky Rannoh ransomware recovery utility, which is available without cost, should be capable of restoring access to these files.

9.7. Future Directions

9.7.1. Behavior-Based Detection: Advancements in Behavioral Analysis

Behavior-based detection involves analyzing the behavior of software or processes to identify abnormal patterns that may indicate malware activity. This approach goes beyond traditional signature-based detection, enabling the identification of previously unknown and zero-day malware.

- **Refining Behavioral Analysis Techniques:** Delving deeper into refining behavioral analysis techniques using machine learning and artificial intelligence to better capture subtle deviations in system behavior.
- **Reducing False Positives:** Efforts to reduce false positives and improve the accuracy of behavior-based detection are crucial to minimizing operational disruptions and efficiently managing security resources.

9.7.2. Contextual Analysis: Enhancing Detection Accuracy through Context

Contextual analysis considers the broader context surrounding potential security threats, such as user behavior, network activity, and environmental factors.

- **Leveraging Big Data Analytics:** Involving big data analytics and threat intelligence feeds to contextualize detected events, enabling a more comprehensive understanding of the threat landscape.
- **Improving Cybersecurity Posture:** Enhancing the overall cybersecurity posture by providing more accurate and relevant alerts for effective incident response.

9.7.3. Cloud-Based Detection: Advancing Protection for Distributed Systems

With the proliferation of cloud computing and distributed systems, malware threats have expanded their reach beyond traditional network boundaries.

- **Scalability and Performance Challenges:** Addressing challenges related to scalability, performance, and multi-tenancy in cloud-based detection systems.
- **Collaboration with Cloud Service Providers:** Collaborating on innovative techniques that leverage cloud resources for efficient and resilient malware detection and response.

9.7.4. IoT and OT Security: Specialized Solutions for Emerging Frontiers

The growing adoption of Internet of Things (IoT) and Operational Technology (OT) systems has opened new attack surfaces for cyber-criminals.

- **Resource-constrained Environments:** Developing lightweight, real-time, and context-aware detection methods suitable for the resource-constrained nature of IoT and OT devices.
- **Effective Malicious Activity Identification:** Identifying malicious activities in IoT and OT environments while securing critical infrastructures and connected devices.

9.7.5. Threat Intelligence Integration: Harnessing Collective Knowledge for Proactive Defense

Threat intelligence feeds provide valuable insights into the latest malware threats and attack techniques.

- **Automated Mechanisms:** Exploring automated mechanisms for processing and correlating threat intelligence data to keep intrusion detection systems up-to-date with the evolving threat landscape.
- **Adaptation of Defenses:** Enhancing the ability to adapt defenses proactively based on the collective knowledge provided by threat intelligence.

9.7.6. Automated Response: Swift Containment and Mitigation of Malware Intrusions

Intrusion detection systems can be augmented with automated response mechanisms to enable rapid containment and mitigation of malware intrusions.

- **Range of Automated Responses:** Developing automated response actions ranging from isolating affected systems to deploying countermeasures against specific malware strains.
- **Safety and Reliability:** Ensuring the safety and reliability of automated responses through careful consideration of potential risks and the use of machine learning for intelligent response strategies.

9.7.7. Adversarial Machine Learning: Safeguarding Intrusion Detection Systems

As intrusion detection systems become more sophisticated, cyber-criminals may attempt to subvert them using adversarial attacks.

- **Defending Against Adversarial Attacks:** Focusing on developing robust techniques to defend against adversarial machine learning attacks targeting intrusion detection systems.
- **Enhancing Resilience:** Exploring adversarial training, ensemble methods, and anomaly detection approaches to bolster the resilience of intrusion detection systems against such attacks.

9.7.8. Blockchain Security: Leveraging Distributed Ledgers for Enhanced Detection and Response

The potential of blockchain technology in enhancing malware detection and incident response is an intriguing area of exploration.

- **Decentralized Threat Intelligence Platforms:** Exploring opportunities for creating decentralized threat intelligence platforms using blockchain's distributed and immutable nature.
- **Addressing Challenges:** Giving careful attention to scalability, privacy, and performance challenges when integrating blockchain into intrusion detection systems.

10. Conclusions

Within the domain of computing devices, the intrusion of malicious software presents a significant threat, adept at clandestinely extracting sensitive information and, on specific occasions, compromising or incapacitating critical security protocols. This review paper meticulously explores the diverse array of techniques employed by cybersecurity professionals to mitigate the risks associated with such malicious software.

Specifically, this article provides a detailed exposition of various approaches to examining malware, encompassing static analysis, dynamic analysis, and blended methodologies. Moreover, it offers a comprehensive evaluation scrutinizing the strengths and weaknesses of established techniques for identifying and combating malware.

Drawing from their analysis, this paper advocates for the adoption of cutting-edge methods, such as data mining and machine learning, to address the limitations inherent in existing approaches. It acknowledges that despite the proliferation of methodologies utilizing various malware detection techniques, no single approach can comprehensively detect all forms of sophisticated, contemporary malware. Though traditional signature-based and heuristic-based detection techniques remain highly effective against well-known viruses, they falter when confronted with uncertain and intricate malware behaviors. In these scenarios, model verification and cloud-based methods emerge as notably promising alternatives.

Advancements in deep learning technologies, the ubiquity of mobile devices, and the expansion of the Internet of Things have empowered systems to recognize both known and novel viruses. Nevertheless, a stark reality persists: not all malware strains can be reliably detected through these advanced methods. This underscores the enduring challenge of devising a proficient system for recognizing adware, illustrating the vast expanse that awaits exploration and investigation through novel studies and approaches.

Additionally, this research encompasses a meticulous examination of prior methodologies employed in virus research and identification. It dedicates substantial attention to the complete spectrum of virus detection techniques, including memory forensics, network analysis, scanners/sandboxes, reverse engineering, troubleshooting, and website analysis

tools. Unlike most earlier studies that focus on a limited subset of these techniques, this paper strives to provide a comprehensive understanding of domain-specific analysis.

Author Contributions: Methodology, V.V.; Software, V.V.; Validation, S.J.; Formal analysis, A.K.B.; Investigation, V.V. and A.P.; Resources, A.K.B.; Data curation, A.K.B. and A.P.; Writing—original draft, S.J., M.K. and M.A.; Writing—review & editing, M.K. and M.A.; Supervision, A.K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Researchers Supporting Project Number (RSPD2023R968), King Saud University Riyadh, Saudi Arabia.

Data Availability Statement: Data sharing not applicable to this article, as no datasets were generated or analysed during the current study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schmnidt, S.; Alpcan, T.; Albayrak, S.; Basar, T.; Mueller, A. A Malware Detector Placement Game for 954 Intrusion Detection. In Proceedings of the 2nd International Workshop on Critical Information Infrastructures Security, Malaga, Spain, 3–5 October 2007; Lopez, J., Hammerli, B., Eds.; Volume 5141, p. 311.
2. Lazarov, A.D. Mathematical Modelling of Malware Intrusion in Computer Networks. *Cybern. Inf. Technol.* **2022**, *22*, 29–47. [CrossRef]
3. Chen, C.M.; Cheng, S.T.; Zeng, R.Y. A proactive approach to intrusion detection and malware collection. *Secur. Commun. Netw.* **2013**, *6*, 844–853. [CrossRef]
4. Ross, A.; Morgan, D. Malware mitigation using host intrusion prevention in the enterprise. In Proceedings of the International Conference on Security and Management, Las Vegas, NV, USA, 21–24 June 2004; Arabnia, H., Aissi, S., Mun, Y., Eds.; pp. 46–52.
5. Kidmose, E.; Stevanovic, M.; Pedersen, J.M. Correlating intrusion detection alerts on bot malware infections using neural network. In Proceedings of the 2016 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), London, UK, 13–14 June 2016.
6. Ali, R.; Ali, A.; Iqbal, F.; Hussain, M.; Ullah, F. Deep Learning Methods for Malware and Intrusion Detection: A Systematic Literature Review. *Secur. Commun. Netw.* **2022**, *2022*, 2959222. [CrossRef]
7. Idika, N.; Mathur, A.P. A survey of malware detection techniques. *Purdue Univ.* **2007**, *48*, 32–46.
8. Kuppusamy, K.; Murugan, S. Preventing Unknown Malware Attack by using Intelligence intrusion Multi detection prevention Systems. *Int. J. Comput. Sci. Netw. Secur.* **2009**, *9*, 299–307.
9. Jones, A.; Straub, J. Using deep learning to detect network intrusions and malware in autonomous robots. In Proceedings of the Conference on Cyber Sensing, Anaheim, CA, USA, 11 April 2017; Volume 10185. [CrossRef]
10. Alazab, A.; Hobbs, M.; Abawajy, J.; Khraisat, A. Developing an Intelligent Intrusion Detection and Prevention System against Web Application Malware. In Proceedings of the 1st International Conference on Advances in Security of Information and Communication Networks (SecNet 2013), Cairo, Egypt, 3–5 September 2013; Awad, A., Hassanien, A., Baba, K., Eds.; Volume 381, p. 177.
11. Golovko, V.; Bezobrazov, S.; Kachurka, P.; Vaitsekhovich, L. Neural Network and Artificial Immune Systems for Malware and Network Intrusion Detection. In *Advances in Machine Learning II: Dedicated To The Memory of Professor Ryszard S. Michalski*; Koronacki, J., Ras, Z., Wierzchon, S., Kacprzyk, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 263, pp. 485–513.
12. Marchetti, M.; Messori, M.; Colajanni, M. Peer-to-Peer Architecture for Collaborative Intrusion and Malware Detection on a Large Scale. In Proceedings of the 12th Information Security Conference (ISC 2009), Pisa, Italy, 7–9 September 2009; Samarati, P., Yung, M., Martinelli, F., Ardagna, C., Eds.; Volume 5735, pp. 475–490.
13. Bisht, P.S.; Mishra, P.; Chauhan, P.; Joshi, R.C. HyperGuard: On designing out-VM malware analysis approach to detect intrusions from hypervisor in cloud environment. *Int. J. Grid Util. Comput.* **2023**, *14*, 356–367. [CrossRef]
14. Lee, J.K.; Moon, S.Y.; Park, J.H. HB-DIPM: Human Behavior Analysis-Based Malware Detection and Intrusion Prevention Model in the Future Internet. *J. Inf. Process. Syst.* **2016**, *12*, 489–501. [CrossRef]
15. Melvin, A.A.R.; Kathrine, G.J.W.; Ilango, S.S.; Vimal, S.; Rho, S.; Xiong, N.N.; Nam, Y. Dynamic malware attack dataset leveraging virtual machine monitor audit data for the detection of intrusions in cloud. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4287. [CrossRef]
16. Yadav, T.; Rao, A.M. Technical aspects of cyber kill chain. In *Proceedings of the International Symposium on Security in Computing and Communication*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 438–452.
17. Landage, J.; Wankhade, M. Malware and malware detection techniques: A survey. *Int. J. Eng. Res.* **2013**, *2*, 61–68.
18. Aslan, Ö.A.; Samet, R. A comprehensive review on malware detection approaches. *IEEE Access* **2020**, *8*, 6249–6271. [CrossRef]
19. Souppaya, M.; Scarfone, K. Guide to malware incident prevention and handling for desktops and laptops. *NIST Spec. Publ.* **2013**, *800*, 83.
20. Talukder, S. Tools and techniques for malware detection and analysis. *arXiv* **2020**, arXiv:2002.06819.

21. Park, S.H.; Yun, S.W.; Jeon, S.E.; Park, N.E.; Shim, H.Y.; Lee, Y.R.; Lee, S.J.; Park, T.R.; Shin, N.Y.; Kang, M.J.; et al. Performance Evaluation of Open-Source Endpoint Detection and Response Combining Google Rapid Response and Osquery for Threat Detection. *IEEE Access* **2022**, *10*, 20259–20269. [CrossRef]
22. Banerjee, U.; Vashishtha, A.; Saxena, M. Evaluation of the Capabilities of WireShark as a tool for Intrusion Detection. *Int. J. Comput. Appl.* **2010**, *6*, 1–5. [CrossRef]
23. Masri, R.; Aldwairi, M. Automated malicious advertisement detection using virustotal, urlvoid, and trendmicro. In Proceedings of the 2017 8th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 4–6 April 2017; pp. 336–341.
24. Souri, A.; Hosseini, R. A state-of-the-art survey of malware detection approaches using data mining techniques. *Hum.-Centric Comput. Inf. Sci.* **2018**, *8*, 1–22. [CrossRef]
25. Joloudari, J.H.; Haderbadi, M.; Mashmool, A.; GhasemiGol, M.; Band, S.S.; Mosavi, A. Early detection of the advanced persistent threat attack using performance analysis of deep learning. *IEEE Access* **2020**, *8*, 186125–186137. [CrossRef]
26. Auty, M. Anatomy of an advanced persistent threat. *Netw. Secur.* **2015**, *2015*, 13–16. [CrossRef]
27. Talukder, S.; Talukder, Z. A survey on malware detection and analysis tools. *Int. J. Netw. Secur. Its Appl. (IJNSA)* **2020**, *12*. [CrossRef]
28. Vinod, P.; Jaipur, R.; Laxmi, V.; Gaur, M. Survey on malware detection methods. *Hackers 2009* **2009**, 74–79.
29. Saeed, I.A.; Selamat, A.; Abuagoub, A.M. A survey on malware and malware detection systems. *Int. J. Comput. Appl.* **2013**, *67*.
30. Chen, T.M.; Robert, J.M. The evolution of viruses and worms. *Stat. Methods Comput. Secur.* **2004**, *1*.
31. Schultz, E.E. Where have the worms and viruses gone?—New trends in malware. *Comput. Fraud. Secur.* **2006**, *2006*, 4–8. [CrossRef]
32. Zeidanloo, H.R.; Tabatabaei, F.; Amoli, P.V.; Tajpour, A. All About Malwares (Malicious Codes). In Proceedings of the Security and Management, Las Vegas, NV, USA, 12–15 July 2010; pp. 342–348.
33. Gao, Y.; Doan, B.G.; Zhang, Z.; Ma, S.; Zhang, J.; Fu, A.; Nepal, S.; Kim, H. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv* **2020**, arXiv:2007.10760.
34. Bhunia, S.; Hsiao, M.S.; Banga, M.; Narasimhan, S. Hardware Trojan attacks: Threat analysis and countermeasures. *Proc. IEEE* **2014**, *102*, 1229–1247. [CrossRef]
35. Thompson, R. Why spyware poses multiple threats to security. *Commun. ACM* **2005**, *48*, 41–43. [CrossRef]
36. Kim, S.; Park, J.; Lee, K.; You, I.; Yim, K. A Brief Survey on Rootkit Techniques in Malicious Codes. *J. Internet Serv. Inf. Secur.* **2012**, *2*, 134–147.
37. Kaur, N.; Singh, M. Botnet and botnet detection techniques in cyber realm. In Proceedings of the 2016 international conference on inventive computation technologies (ICICT), Coimbatore, India, 26–27 August 2016; Volume 3; pp. 1–7.
38. Singh, S.; Sharma, P.K.; Moon, S.Y.; Moon, D.; Park, J.H. A comprehensive study on APT attacks and countermeasures for future networks and communications: Challenges and solutions. *J. Supercomput.* **2019**, *75*, 4543–4574. [CrossRef]
39. Wazid, M.; Katal, A.; Goudar, R.; Singh, D.; Tyagi, A.; Sharma, R.; Bhakuni, P. A framework for detection and prevention of novel keylogger spyware attacks. In Proceedings of the 2013 7th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, Tamil Nadu, India, 4–5 January 2013; pp. 433–438.
40. Ahvanooey, M.T.; Li, Q.; Rabbani, M.; Rajput, A.R. A survey on smartphones security: software vulnerabilities, malware, and attacks. *arXiv* **2020**, arXiv:2001.09406.
41. Valeros, V.; Garcia, S. Growth and commoditization of remote access trojans. In Proceedings of the 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Genoa, Italy, 7–11 September 2020; pp. 454–462.
42. Brewer, R. Ransomware attacks: Detection, prevention and cure. *Netw. Secur.* **2016**, *2016*, 5–9. [CrossRef]
43. Bada, M.; Creese, S.; Goldsmith, M.; Mitchell, C.; Phillips, E. Computer Security Incident Response Teams (CSIRTs): An Overview. *Glob. Cyber Secur. Capacit. Cent.* **2014**.
44. Egele, M.; Scholte, T.; Kirda, E.; Kruegel, C. A survey on automated dynamic malware-analysis techniques and tools. *ACM Comput. Surv. (CSUR)* **2008**, *44*, 1–42. [CrossRef]
45. Bhatia, T.; Kaushal, R. Malware detection in android based on dynamic analysis. In Proceedings of the 2017 International Conference on Cyber Security And Protection of Digital Services (Cyber Security), Wales, UK, 19–20 June 2017; pp. 1–6.
46. Tahir, R. A study on malware and malware detection techniques. *Int. J. Educ. Manag. Eng.* **2018**, *8*, 20. [CrossRef]
47. Damodaran, A.; Troia, F.D.; Visaggio, C.A.; Austin, T.H.; Stamp, M. A comparison of static, dynamic, and hybrid analysis for malware detection. *J. Comput. Virol. Hacking Tech.* **2017**, *13*, 1–12. [CrossRef]
48. Goyal, M.; Kumar, R. The Pipeline Process of Signature-based and Behavior-based Malware Detection. In Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 30–31 October 2020; pp. 497–502.
49. Chakravarty, A.K.; Raj, A.; Paul, S.; Apoorva, S. A study of signature-based and behaviour-based malware detection approaches. *Int. J. Adv. Res. Ideas Innov. Technol.* **2019**, *5*, 1509–1511.
50. Singh, J.; Singh, J. A survey on machine learning-based malware detection in executable files. *J. Syst. Archit.* **2021**, *112*, 101861. [CrossRef]
51. Treadwell, S.; Zhou, M. A heuristic approach for detection of obfuscated malware. In Proceedings of the 2009 IEEE International Conference on Intelligence and Security Informatics, Richardson, TX, USA, 8–11 June 2009; pp. 291–299.

52. Rehman, Z.U.; Khan, S.N.; Muhammad, K.; Lee, J.W.; Lv, Z.; Baik, S.W.; Shah, P.A.; Awan, K.; Mehmood, I. Machine learning-assisted signature and heuristic-based detection of malwares in Android devices. *Comput. Electr. Eng.* **2018**, *69*, 828–841. [CrossRef]
53. Chew, C.J.; Kumar, V. Behaviour based ransomware detection. In Proceedings of the 34th International Conference on Computers and Their Applications, CATA 2019, Honolulu, HI, USA, 18–20 March 2019.
54. Lajevardi, A.M.; Parsa, S.; Amiri, M.J. On the vulnerability of behaviour-based malware detection methods. *Softw. Eng. Simul.* **2015**, *2*, 01–05.
55. Zhou, W.; Yu, B. A cloud-assisted malware detection and suppression framework for wireless multimedia system in IoT based on dynamic differential game. *China Commun.* **2018**, *15*, 209–223. [CrossRef]
56. Yerima, S.Y.; Sezer, S.; Muttik, I. High accuracy android malware detection using ensemble learning. *IET Inf. Secur.* **2015**, *9*, 313–320. [CrossRef]
57. Zhu, D.; Jin, H.; Yang, Y.; Wu, D.; Chen, W. DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; pp. 438–443.
58. Su, J.; Vasconcellos, D.V.; Prasad, S.; Sgandurra, D.; Feng, Y.; Sakurai, K. Lightweight classification of IoT malware based on image recognition. In Proceedings of the 2018 IEEE 42Nd annual computer software and applications conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; Volume 2, pp. 664–669.
59. Azmoodeh, A.; Dehghantanha, A.; Choo, K.K.R. Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning. *IEEE Trans. Sustain. Comput.* **2018**, *4*, 88–95. [CrossRef]
60. Chung, F.R. *Spectral Graph Theory*; American Mathematical Society: Providence, RI, USA, 1997; Volume 92.
61. Faruk, M.J.H.; Shahriar, H.; Valero, M.; Barsha, F.L.; Sobhan, S.; Khan, M.A.; Whitman, M.; Cuzzocrea, A.; Lo, D.; Rahman, A.; et al. Malware Detection and Prevention Using Artificial Intelligence Techniques. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 5369–5377.
62. Chen, L.; Ye, Y.; Bourlai, T. Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 11–13 September 2017; pp. 99–106.
63. Wu, C.; Shi, J.; Yang, Y.; Li, W. Enhancing machine learning based malware detection model by reinforcement learning. In Proceedings of the 8th International Conference on Communication and Network Security, Qingdao, China, 2–4 November 2018; pp. 74–78.
64. Demontis, A.; Melis, M.; Biggio, B.; Maiorca, D.; Arp, D.; Rieck, K.; Corona, I.; Giacinto, G.; Roli, F. Yes, machine learning can be more secure! A case study on android malware detection. *IEEE Trans. Dependable Secur. Comput.* **2017**, *16*, 711–724. [CrossRef]
65. Wu, H.; Han, H.; Wang, X.; Sun, S. Research on artificial intelligence enhancing internet of things security: A survey. *IEEE Access* **2020**, *8*, 153826–153848. [CrossRef]
66. Li, J.H. Cyber security meets artificial intelligence: A survey. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 1462–1474. [CrossRef]
67. Mohapatra, N.; Satapathy, B.; Mohapatra, B.; Mohanta, B.K. Malware Detection using Artificial Intelligence. In Proceedings of the 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 3–5 October 2022; pp. 1–6.
68. Meyer, M.; Auth, G.; Schinner, A. *A Method for Evaluating and Selecting Software Tools for Remote Forensics*; Gesellschaft für Informatik: Bonn, Germany, 2021.
69. Rabadi, D.; Teo, S.G. Advanced windows methods on malware detection and classification. In Proceedings of the Annual Computer Security Applications Conference, Austin, TX, USA, 7–11 December 2020; pp. 54–68.
70. Rasheed, H.; Hadi, A.; Khader, M. Threat hunting using grr rapid response. In Proceedings of the 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 11–13 October 2017; pp. 155–160.
71. Goyal, P.; Goyal, A. Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark. In Proceedings of the 2017 9th International Conference on Computational Intelligence and Communication Networks (CICN), Girne, Cyprus, 16–17 September 2017; pp. 77–81.
72. Sandhya, S.; Purkayastha, S.; Joshua, E.; Deep, A. Assessment of website security by penetration testing using Wireshark. In Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 January 2017; pp. 1–4.
73. Iqbal, H.; Naaz, S. Wireshark as a tool for detection of various LAN attacks. *Int. J. Comput. Sci. Eng.* **2019**, *7*, 833–837. [CrossRef]
74. Tuttle, H. Facebook scandal raises data privacy concerns. *Risk Manag.* **2018**, *65*, 6–9.
75. Hu, M. Cambridge Analytica's black box. *Big Data Soc.* **2020**, *7*, 2053951720938091. [CrossRef]
76. Venturini, T.; Rogers, R. "API-based research" or how can digital sociology and journalism studies learn from the Facebook and Cambridge Analytica data breach. *Digit. J.* **2019**, *7*, 532–540. [CrossRef]