Suleman *et al.LGU (IJECI)* 2020

LGU (IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

LGU International Journal for
Electronic Crime Investigation

Research Article

Vol. 5 issue 4 Year 2021

Malware and their diverse characteristics related to detection and analysis: A literature survey

Muhammad Taseer Suleman
taseersuleman@lgu.edu.pk
Digital Forensics Research and Service Centre, Lahore Garrison University, Lahore

Abstract:

The term malware refers to a specific form of software that causes damage to the computing device through data leakage and transformation, device malfunctioning, hacking, and exploitation. A typical malware can be categorized into several categories including virus, worm, trojan, ransomware, etc. The analysis of malware helps in the categorization and behavior judgment of malware. The deep analysis of malware helps in malware detection. The current research study covers a hierarchical representation of various malware categories, different forms of detection, computing devices for malware, and a malware analysis approach. The study describes each category in detail, which helps in forming mapping of malware analysis in detail.

Keywords: malware, analysis, worms, trojans, ransomware, detection.

1. Introduction

The word Malware consists of two words: Mal means "Malicious" ware means "Software" means Malicious Software. The malware contains a set of instructions that can harm or damage your computers on the behalf of the attacker or according to the intentions of the attacker. Malware is malicious software that can harm computers or networks on a large (WAN) or small scale (LAN) [1]. The malware contains diverse types and categories. The malware consists of botnets, spyware, rootkits, backdoor, worms, viruses, spam, ransomware, scareware, and adware[2]. Due to diversity in the malware categories, these affect a variant type of devices. Nowadays, a single user can interact with several devices including a personal computer, laptop, smart phone, smart watch, etc. Several malware categories affect these devices in number of ways including hacking, data leakage, data transformation, data unavailability, device malfunctioning etc. The detection of malware is of great importance [3]. Detection can be broadly classified into central-based or peer-based. The central-based detection involves one server, to which different devices are communicated and the server scans each device for the possible matching of virus strings [4]. Moreover, peer-based detection involves any malware

detection software installed in the device. The known malware can be caught through scanning of installed anti-malware applications.

The current study focused on the malware taxonomical representation in terms of its categories, detection approaches, interaction with different devices, and malware analysis types. Each category is discussed in detail. The rest of the paper presents malware taxonomy in detail with conclusive remarks at the end.

2. Malware Taxonomy

The malware taxonomy spans largely into its different types, detection capabilities, effects on different devices most importantly the behavioral analysis through which

2.1. Malware categories

2.1.1. Botnets:

Botnets allow attackers to access the system, Botnets are usually deployed on a large scale if the computers are infected with same type of botnet that it can be accessed from one C2C server (Command and Control Server). Botnets are a special kind of malware. Botnets spread like worms. Most of the botnets are made up of **IRC (Internet Relay Chats)** bots that are connected via the channel in which they accept commands for example "DDOS" means increasing the traffic on the server, "run/exploit" to open anything in the targeted system.

- **Spyware:** The main purpose of spyware is to monitor or keep track of its targets and can steal their data. It usually collects information from the target system and sends it to the attacker for example keyloggers, sniffers, spybots, etc., [5].

- **Keyloggers:** Keyloggers are used to collect your keyboard logs, can collect your sensitive information for example bank information, passwords, confidential messages, etc.

- **Sniffers:** A sniffer is a tool in software or hardware form that can be used to track/-monitor internet traffic in real-time. It can capture all that your computer sends or receive.

- **Spybots:** Spybots can enter our computer system and can collect information about us and transfer all the collected information to a third party.

2.1.2. Rootkits:

Rootkits are designed in a way to modify the Operating System to create a backdoor. They operate in a way that they cannot be detected. These are generally combined with other malware to conceal their code. They can also take advantage of different vulnerabilities to gain remote access [6]. Rootkits can also be used to modify monitoring tools, making it difficult for them to detect. In most cases when a computer is infected with a rootkit it wiped everything on the system for example backdoors to make the code fileless or undetectable for the victim [7].

2.1.3. Backdoor:

The backdoor is the type of malicious code that can be installed inside the computer system to gain unauthorized access without user permissions and gives access to the attacker, when the backdoor is open or get executed, the attacker gets connected to the target computer with no

authentication it's like C2C Server (Command and Control Server)[8]. The attacker can easily execute commands on the target computer. It usually works in the Background, and it is difficult to detect.

2.1.4. Worms/Virus:

The main purpose of worms is to duplicate themselves to spread from one computer to another; they make multiple copies of themselves to consume more ram and to make the computer slow some things it gets hanged and unable to respond. Worms can be run by themselves for example **ILOVEYOU** worm is the famous one it comes with a .vbs extension, **Stuxnet** is responsible for causing damage to the **Nuclear Program of Iran**. The virus can also be programmed to bypass the AV detection most of them are spread by CDs, unusually downloading, attaching USB, etc.

2.1.5. Spam-Sending Malware:

In Spam-Sending Malware, malware enters the target system infects the system and uses that system to send spam mails. It also provides spam-email sending services, most of the users buy this for their marketing and business purpose.

2.1.6. Ransomware:

Ransomware enters into your computer system to encrypt all the files present in your computer and ask for a ransom to open or decrypt them. It changes the extension of the files, so it becomes impossible for the user to open that file without the key. Ransomware is usually spread with phishing emails it encourages the user to download the malicious file and then in this way it can affect the targets. Attackers ask for ransom in Bitcoin form. The Bitcoin address is visible to you. One of the popular ransomware attacks is **WannaCry** .

2.1.7. Scareware:

Scareware works in a way that frightens the user. It uses **scare** technique to trap the user to perform a specific action for example to buy something. Scareware usually has a GUI interface like and trusted antivirus or security program. It tells the user that there is some virus or bug in your system that can only be removed by their software in that way user buy their software. For example, when we are scrolling social media apps some advertisements come and display there is a virus in our mobile or laptop with the mobile or laptop model. In this way, scareware trapped their victim [9].

2.1.8. Adware:

Adware is designed to deliver advertisements automatically to the target. Adware is usually Web-Browser Based. It is very difficult for a normal user to ignore it when these are constantly popping on the target screen. It usually comes with spyware.

Fig 1. Shows the malware categories in detail.

Fig 1. Malware categories

2.2. Malware Detection Approaches:

Malware detection is mainly carried out through machine learning and artificial intelligence. The malware detection techniques can be broadly categorized into signature-based, behavior-based, heuristic-based, model-learning-based, and deep-learning-based.

2.2.1. Signature-based detection:

A signature is a set of bits that uniquely identifies the structure of a program. Signatures are commonly employed in malware identification because each program's signature is unique. The static properties of executable files are first discovered during the signature extraction process. The signature generating engine then uses the extracted features to create signatures. When a suspected sample file must be classified as malicious or benign, the file's signature is retrieved and compared to previously identified signatures. The sample file is classified as harmful or benign based on the comparison. Signature-based malware detection is the term for this method. When it comes to detecting known malware, this method is quick and effective. It, on the other hand, is unable to detect zero-day malware. Furthermore, signature-based malware detection is no longer viable because it cannot detect new malware variants, is inefficient, and requires human engagement.

2.2.2. Behavior-based Detection:

The sample program's activities are tracked in a behavior-based detection approach. The sample program is classified as malicious or benign based on the observed behaviors. The three aspects of this approach are: extracting behaviors, creating attributes, and utilizing machine learning algorithms to determine whether the studied program is dangerous or benign. System calls, API calls, or changes in the file, registry, and computer network are utilized to determine behaviors. To put it another way, the order or frequency of system calls and file-registry actions is used to influence behavior [10]. Even if the source code of software changes over time, the program's behavior does not. As a result, this method can be used to identify a variety of malicious software variations. Furthermore, this technique can detect previously undiscovered novel malware. The most significant disadvantage of behavior-based detection is that malware does not exhibit all of its true characteristics in a protected environment like virtual machines or sandboxes. Using cyber threat intelligence, machine learning, and data forensics, a new hybrid approach based on dynamic analysis has been developed [11].

2.2.3. Features-Based Detection:

Heuristic-based detection is a multi-technique detection strategy. Certain guidelines and machine learning approaches are used in this experience-based approach. To produce rules, the heuristic technique can use both strings and behavior-related data [12]we investigate the stability of a susceptible-infected-susceptible epidemic model incorporated with multiple infection stages and propagation vectors to mimic malware behavior over scale-free communication networks. In particular, we derive the basic reproductive ratio (R{0}. Signatures are made by those requirements. It's mostly used to identify various types of malware, as well as malware that has never been seen before. To begin, the system is taught using specific features. First and foremost, the system is taught using certain features. Anomalies are then discovered by testing the data. Although the success rate for

detecting new malware is high, because of optimization concerns, the rate of false positives (FP) and false negatives (FN) is also high.

2.2.4. Model-learning based detection:

Malicious and benign characteristics are retrieved and coded using linear temporal logic formulas to identify feature relationships, which are referred to as specifications, in the model checking-based detection approach. Flow linkages between behaviors that employ hiding, spreading, and injecting actions are used to extract program properties. The properties collected are compared to the previously determined parameters to classify the sample software file as malware or benign. The file is classified as malicious or benign based on the comparison. This method is impervious to stealth and packing techniques, and it can detect a part of new malware versions.

2.2.5. Deep-Learning Based detection:

Deep learning is a branch of AI that learns from examples and inherits from artificial neural networks (ANNs). Deep learning is widely utilized in sectors like image processing, self-driving cars, and voice control, but it isn't generally used in malware detection and categorization. The deep learning-based detection strategy works well and decreases the feature dimension significantly, but it is vulnerable to evasion assaults [13][14]. Furthermore, constructing hidden layers takes a long time, and adding more hidden layers only improves performance slightly. Because deep learning hasn't been widely employed in malware detection and classification, additional academic research is needed to accurately assess this method [15].

Fig. 2 depicts malware detection approaches in taxonomical form.

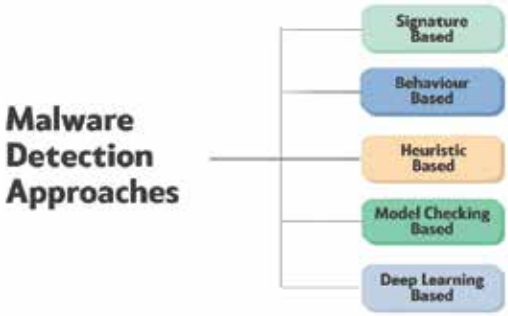

Fig. 2. Taxonomical representation of malware detection approaches

2.3. Malware Detection for various devices:

Malware Detection detects intrusions by monitoring the malware activity on different platforms and classifying it as normal or dangerous. This classification is often based on machine learning algorithms that use different types of rules to detect them instead of detecting them with signatures and patterns.

2.3.1. Malware Obfuscation Techniques:

The technique that malware researchers used to conceal their code so that it becomes difficult for the victim and AV to read or understand it this technique is called **Obfuscation Technique.**

2.3.2. Malware Detectors:

A Malware detector is a program that is used to detect malware it is largely based on rules, hashes, and signatures.

Malware Detection is Based on

- Signature-Based Detection
- Behavior-Based Detection

2.3.3. Signature-Based Detection:

The most widely used malware detection method is the detection of malware by signa-

tures. Instead of going for behavior detection methods, malware detection has largely focused on completing the signature-based analysis. In signature-based detection, all the signatures of known malware were stored when malware tries to enter the system it checks whether its signature is stored in our database or not. If the signature matches then it is confirmed that it is malware. Signature-based rely on extraordinary raw byte examples or standard articulations, referred to as marks, made to organize the harmful document [16]. Static detection of data, for example, previous signatures are used to determine whether it is malware or not. The fundamental advantage of signature-based approaches is their depth, as they cover all possible document execution contexts [17]accurate detection is challenging due to the constantly evolving nature of the malware variants that cause concept drift. Existing malware detection solutions assume that the mapping learned from historical malware features will be valid for new and future malware. The relationship between input features and the class label has been considered stationary, which doesn't hold for the ever-evolving nature of malware variants. Malware features change dynamically due to code obfuscations, mutations, and the modification made by malware authors to change the features' distribution and thus evade the detection rendering the detection model obsolete and ineffective. This study presents an Adaptive behavioral-based Incremental Batch Learning Malware Variants Detection model using concept drift detection and sequential deep learning (AIBL-MVD.

2.3.4. Mobile-Malware Detection Techniques:

Mobile Malware Detection Techniques are divided into two categories:

- Static Techniques
- Dynamic Techniques

Fig 3. Represents a mobile computing device.

Fig. 3. A mobile computing device

2.3.5. Static Technique:

The static technique mainly relies on the source code of an application to classify it accordingly without having the application being executed. These techniques are classified into one of the following classes according to the basis they rely on for analyzing source code [18].

2.3.6. Signature-Based Detection:

A program is classified as malware or a type of malware if its signature matches the existing signatures. This is a very fast method. AV can only identify the existing malware and fails against the unseen variants of malware. It also needs an immediate update of malware signatures.

2.3.7. Dynamic Techniques:

In Dynamic analysis, an application is examined during its execution, and then it is classified according to one of the following techniques. The Classification is done accord-

ing to the behavior of the detection mechanism.

2.3.8. IoT-Malware Detection Techniques:

IoT is moreover inside the midst of numerous information safety vulnerabilities and exploits. Risks of connecting networks if we tend to take below consideration the technical possibilities of devices moreover to their specific weaknesses, the benefit with those hackers will sight them, and their anticipated proliferation worldwide. And the projected global impact becomes simply evident in any elegant surroundings. Internet of Things (IoT) practice includes internet-based gadgets, internet-based cars, embedded structures, sensors, and alternative gadgets or systems that have a manner of autonomy[19]. The manner of shifting as well as aggregation understanding. IoT would possibly moreover be deployed inside aggressive ways fee the object of terrain like Security teams uses an electromagnetic field to detect malware in an IoT Device[20]. This will even work in obfuscation cases.
Fig. 4. Shows different IoT devices.

Fig 4. IoT devices and malware

The recent finding presented by security researchers from the **Research Institute of Computer Science and Random System (IRISA)** at the **Annual Computer Security Applications Conference (ACSAC)**

- Attackers use the others channel info to discover anomalies in emanations after they range from previously determined patterns and suspicious behavior in the system's normal state

- Without making any modifications in the method, this method enables detection along with the classification of rootkits that required administrative rights, ransomware, or some new malware

- The electromagnetic emanation calculated from the device is almost undetectable by the malware.

During the research Malware Researcher uses the Raspberry Pi 2B model as a target device with 1GB of Ram and 900MHZ quad-core ARM Cortex A7 processor combined with the 303 BNC preamplifier. With an accuracy of 99.82 percent and 99.61 percent, this system was able to detect three malware families.

The method involves three phases: measuring electromagnetic fields while executing 30 different malware binaries, performing benign activities to train a Convolutional Neural Network (CNN) model to classify malware samples, and instructing a Convolutional Neural Network (CNN) model to classify malware samples.

- The framework takes an executable as input and uses side-channel information to generate malware labels.

- Researchers were able to acquire useful information about the state of a monitored device by using simple neural network models.

- It works against a variety of code obfuscation/transformations, including random trash injection, hypervisor, and packaging, as well as a previously unknown transformation.

Concluding this example, we can say that IoT appliances are a lucrative target for cybercrime due to their rapid development and usage. The attack surface is much larger, making stealthy malware harder to identify. To mitigate potential security threats, researchers are required to develop malware analysis techniques.

2.3.9. Malware Detection on Computer System:

As software evolves and improves, it also develops vulnerabilities, requiring the deployment of security patches. Furthermore, Operating system security vulnerabilities are becoming increasingly significant. Older versions of operating systems receive fewer and fewer upgrades with each new release until manufacturers stop supporting them permanently. Companies that are either unable to upgrade or want to ensure compatibility with legacy software continues to employ operating systems with known vulnerabilities. As a result, the issue of legacy operating systems remains vital. Detecting malicious software, or "malware," within features extracted is widely misunderstood, and usually consists of just running an antivirus scanning application across an acquired image mounted as a volume. Malware detection is divided into four stages: analysis, categorization, detection, and eventual containment. A typical cloud computing system has been represented in Fig.5.

Fig.5. A typical cloud computing environment

Several classification strategies have been used to classify malware according to their occurrences, allowing for the recognition of a virus's type and activities, and also new types. Malware analysis entails finding malware samples using multiple classification techniques based on properties of known malware characteristics. Malware detection refers to the process of quickly detecting and detecting any instance of malware to prevent additional system damage.

- **Web Application Firewall(WAF):** The Imperva cloud PCI DSS compatible technology, which is deployed at the edge of your network, uses signature, behavior, and reputational analysis to block all malware injection attempts on your websites and online applications. Imperva Cloud WAF is available as a managed service that is supported by a security team.

- **Backdoor Protect:** On your web server, software that intercepts communication attempts using backdoor shells. The service can pinpoint the most highly obfuscated malware by monitoring these requests, even if it was installed on your web server before you signed up for

Imperva cloud protection services.

- **Login Protect:** A two-factor authentication (2FA) solution that requires no connectivity and can be deployed on any Imperva cloud-protected URL address in seconds. The service prevents cybercriminals from gaining network access and installing rootkits and backdoors on your web servers using stolen login information such as username and password.

- **Yara Rules:** The term YARA belongs to malware research and detection tool. It uses a rule-based approach to generate malware family descriptions based on textual or binary patterns. A description is just a Yara rule name, with these rules being grouped.

2.4. Types of malware analysis

Malware analysis can be broadly categorized into two categories i.e., static analysis and dynamic analysis. The static analysis refers to the analytical review of any malware through code inspection, string search, etc. However, for dynamic analysis, the malware is executed in a protected virtual environment (i.e., sandbox) for its behavior inspection. Fig. 6 represents malware analysis taxonomy.

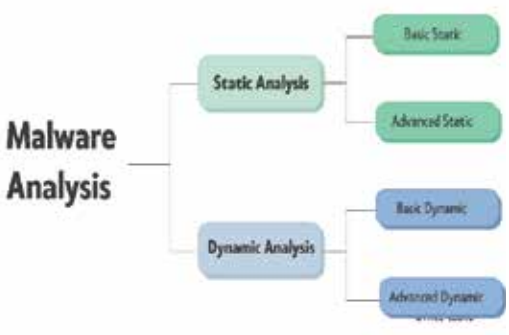

Fig. 6. Malware analysis taxonomy

2.4.1. Static analysis

Static analysis is based on the idea of examining the malicious program without executing the program. As running the malicious program can affect your system, this is a secure way to examine the malware. Static analysis shows a program or file is malicious, gives information about its functionality. Static Analysis is divided into parts: Basic static analysis and advanced static analysis. In Basic static analysis, header information, metadata such as filename, sizes are analyzed. Also, Md5 checksums and hashes are compared with the previously detected malware. Tools like BinText, PEview, MD5deep, and PEid are used to obtain this information. Basic static analysis is quick and simple, and much inaccurate for sophisticated malware. In advanced static analysis, malware code is examined in detail. One way is to reverse engineer the machine code into an assembly code using a disassembler. Various tools like IDA pro are widely used for this purpose. Assembly codes are analyzed thoroughly to discover the functionalities of malware. Also, malware headers, functions, and strings are analyzed and give much useful information about the malware. The advanced static analysis gives information about the malware‘s purpose, working, and functionality, however, the analysis requires deep expertise of assembly instructions and concepts of OS.

2.4.2. Dynamic Analysis

Dynamic analysis is the analyzing the instructions of malware by running it and behaviors of the malware is examined; finding what it does and changes on the hosted system, therefore it is also called malware behavior analysis.

Running the malware on the system is a risk, to save the system to get infected by the malware, the dynamic analysis must be done in close environments such as virtual machines and sandbox. Sandbox is an isolated virtual environment thus malware cannot affect the system or transmit on the network. Function calls, parameters, file path locations, the flow of information, registry changes, and network-related activities are examined. There can be two parts of dynamic analysis: basic dynamic analysis and advanced dynamic analysis. The basic dynamic analysis examines the malware's behavior with tools like API monitor, process explorer, process monitor, ApateDNS, Regshot, Wireshark, and virtual environments. In advanced dynamic, debugging tools are used to examine the behaviors; tools like WinDbg, OllyDbg, etc allow the analyst to run every instruction individually for examining the changes made by the malware. Changes can be registry keys, domain names, IP addresses, file path locations, and memory areas. Most of the functionalities can be found in advanced dynamic analysis. Performing analysis on debuggers requires good expertise and knowledge of assembly-level instructions and operating system concepts.

3. Conclusion

Malware is a typical software crafted to disrupt the normal operation of a computing device. Malware comes in wide categories including viruses, worms, trojans, ransomware, rootkits, etc. The effect of malware includes data stealing, device malfunctioning, data transformation, device hacking, and services unavailability. The analysis and detection of malware are extremely important due to their harmful effects on computing devices. The current study dealt with the malware categories, detection, and analysis types, and the mapping of malware with different computing devices. It has been concluded that the research highlights many aspects of malware in terms of its behavior.

4. References

[1] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," IEEE Access, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[2] R. Gupta and S. P. Agarwal, "a Comparative Study of Cyber Threats in Emerging Economies," 2017, [Online]. Available: www.InternetLiveStats.com.

[3] K. Iwamoto and K. Wasaki, "Malware classification based on extracted API sequences using static analysis," Asian Internet Engineeering Conf. AINTEC 2012, pp. 31–38, 2012, doi: 10.1145/2402599.2402604.

[4] G. Cabau, M. Buhu, and C. P. Oprisa, "Malware classification based on dynamic behavior," Proc. - 18th Int. Symp. Symb. Numer. Algorithms Sci. Comput. SYNASC 2016, pp. 315–318, 2017, doi: 10.1109/SYNASC.2016.057.

[5] J. Donahue, A. Paturi, and S. Mukkamala, "Visualization techniques for efficient malware detection," IEEE ISI 2013 - 2013 IEEE Int. Conf. Intell. Secur. Informatics Big Data, Emergent Threat. Decis. Secur. Informatics, pp. 289–291, 2013, doi: 10.1109/ISI.2013.6578845.

[6] X. Ban, C. Li, W. Hu, and W. Qu, "Malware variant detection using similarity search over content fingerprint," 26th Chinese Control Decis. Conf. CCDC 2014, pp. 5334–5339, 2014, doi: 10.1109/CCDC.2014.6852216.

[7] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," ACM Int. Conf. Proceeding Ser., 2011, doi: 10.1145/2016904.2016908.

[8] R. Komatwar and M. Kokare, "A Survey on Malware Detection and Classification," J. Appl. Secur. Res., vol. 16, no. 3, pp. 390–420, 2021, doi: 10.1080/19361610.2020.1796162.

[9] S. Sen, E. Aydogan, and A. I. Aysan, "Coevolution of Mobile Malware and Anti-Malware," IEEE Trans. Inf. Forensics Secur., vol. 13, no. 10, pp. 2563–2574, 2018, doi: 10.1109/TIFS.2018.2824250.

[10] L. Chen, C. Xia, S. Lei, and T. Wang, "Detection, Traceability, and Propagation of Mobile Malware Threats," IEEE Access, vol. 9, pp. 14576–14598, 2021, doi: 10.1109/ACCESS.2021.3049819.

[11] R. Korine and D. Hendler, "DAEMON: Dataset/Platform-Agnostic Explainable Malware Classification Using Multi-Stage Feature Mining," IEEE Access, vol. 9, pp. 78382–78399, 2021, doi: 10.1109/ACCESS.2021.3082173.

[12] A. Dadlani, M. S. Kumar, K. Kim, and K. Sohraby, "Stability and immunization analysis of a malware spread model over scale-free networks," IEEE Commun. Lett., vol. 18, no. 11, pp. 1907–1910, 2014, doi: 10.1109/LCOMM.2014.2361525.

[13] O. Aslan and A. A. Yilmaz, "A New Malware Classification Framework Based on Deep Learning Algorithms," IEEE Access, vol. 9, pp. 87936–87951, 2021, doi: 10.1109/ACCESS.2021.3089586.

[14] "Intelligent vision-based malware detection and classification using deep random forest paradigm."

[15] M. Nisa et al., "Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features," Appl. Sci., vol. 10, no. 14, 2020, doi: 10.3390/app10144966.

[16] M. Wazid, A. K. Das, J. J. P. C. Rodrigues, S. Shetty, and Y. Park, "IoMT Malware Detection Approaches: Analysis and Research Challenges," IEEE Access, vol. 7, pp. 182459–182476, 2019, doi: 10.1109/ACCESS.2019.2960412.

[17] A. A. Darem, F. A. Ghaleb, A. A. Al-Hashmi, J. H. Abawajy, S. M. Alanazi, and A. Y. Al-Rezami, "An Adaptive Behavioral-Based Incremental Batch Learning Malware Variants Detection Model Using Concept Drift Detection and Sequential Deep Learning," IEEE Access, vol. 9, pp. 97180–97196, 2021, doi: 10.1109/ACCESS.2021.3093366.

[18] G. Suarez-Tangil, J. E. Tapiador, F. Lombardi, and R. Di Pietro, "Alterdroid: Differential Fault Analysis of Obfuscated

Smartphone Malware," IEEE Trans. Mob. Comput., vol. 15, no. 4, pp. 789–802, 2016, doi: 10.1109/TMC.2015.2444847.

[19] P. Faruki et al., "Android security: A survey of issues, malware penetration, and defenses," IEEE Commun. Surv. Tutorials, vol. 17, no. 2, pp. 998–1022, 2015, doi: 10.1109/COMST.2014.2386139.

[20] J. Jeon, J. H. Park, and Y. S. Jeong, "Dynamic Analysis for IoT Malware Detection with Convolution Neural Network Model," IEEE Access, vol. 8, pp. 96899–96911, 2020, doi: 10.1109/ACCESS.2020.2995887.