A state-of-the-art analysis of android malware detection methods

Jebin Bose S
Research Scholar/ Department of
Computer Science and Engineering
Noorul Islam Centre for Higher
Education
Kumaracoil
jebinboses@gmail.com

Kalaiselvi. R
Associate Professor/ Department of
Computer Science and Engineering
Noorul Islam Centre for Higher
Education
Kumaracoil
kalaiselvir32@gmail.com

Abstract— **Smartphones are constantly changing in today's world, and as a result, security has become a major concern. Security is a vital aspect of human life, and in a world where security is lacking, it becomes a concern for mobile users' safety. Malware is one of the most serious security risks to smartphones. Mobile malware attacks are becoming more sophisticated and widespread. Malware authors consider the open-source Android platform to be their preferred target as it came to lead the market. State-of-the-art mobile malware detection solutions in the literature use a variety of metrics and models, making cross-comparison difficult. In this paper various existing methods are compared and a significant effort is made to briefly address android malwares, various methods for detecting android malwares and to give a clear image of the progress of the android platform and various malware detection classifiers.**

Keywords— Android, Machine Learning, Malware Detection, Security, Smart phone.

I. Introduction

Malware is intended to penetrate or destroy a computer device without the permission of the owner. Malware is a term that encompasses many of these forms of computer threats. Incoming files and separate malware are two types of malwares, according to a basic definition. Worms, backbone, trojans, rootkits, spyware, adware, and other types of malwares can be distinguished based on their particular actions: worms, backbone, trojans, rootkits, spyware, adware, and so on. Malware detection using regular, signed methods is becoming increasingly difficult. To avoid detection by anti-virus software, malware applications often have several polymorphic layers or use automated automatic updates for a new version in a short period of time.

In the opposite case, the attacker's intelligence can be classified as complete information, incomplete information, or zero information. The attacker has a full knowledge of feature space and a trained model, including a type of isolation, in a complete information attack. In a limited information attack, the attacker knows the location of the feature and the separator, but not the partition details of the editor. In that case, the attacker will select the surrogate database according to the same basic distribution. In the case of email spam detection, the attacker may collect a file for features by performing some network attacks. In a zero-awareness attack, the attacker has little or no control information on the type of differentiation and model parameters used by the detector. This is an attack called the black box attack.

Attack tax depending on the impact of the attack, security breaches and specifications. The impact of the attack can be tested or the cause depending on the attack made during the test and the training period. Test attacks modify test samples to avoid detection while causal attacks control data during training creating the wrong distinction. Security breach attacks can be a discovery attack, a private attack or an attack on integrity. The discovery of the attack led to a rejection of the service which resulted in the official sample being divided as cruel, thus refusing. The attack on privacy, too, brings back privacy and empathy data from the system. In the case of an honest attack, the enemy's intent is to infiltrate a vicious circle sample as official.

Research on malware detection on android is enriched with many concepts and strategies to combat the spread of malware. The first way to detect malware for android is using static system analysis. There are several suggested ways to statistically check programs and decrypt their code. Android malware detection during operation was the second method. Therefore, there is a need to study the background of developments on the android platform, malware detection and the latest malware detection variants on the android platform. With this detailed introduction of malwares, Section 2 outlines the development of android platform, malware detection and existing android malware detection methods, Section 3 analyzes various malware detection classifiers followed by conclusion in Section 4.

II. ANDROID MALWARE DETECTION – SYSTEMATIC BACKGROUND

A. Development of Android Platform

Abikoye et al [1] explained the need for systemic performance analysis has been significantly improved because the Android platform is widely used for embedded systems, including intelligent mobile devices. Alam et al [6] have exploded the Apple iPhone deployment and release of the Google Android operating system, mobile application use and deployment. These applications are much easier to develop than previous versions, but they also have the same complexities and problems. Cahya, R et al. [10] have developed teaching aids for language courses on mobile devices. Mobile systems are increasingly common in today's world. Students may benefit greatly from language textbooks that provide mobile learning aids.

Xu Jiang et al [2] presented that an android device application is displayed to control the anti-locking system in the laboratory. The controls are taken into account to avoid locking the wheel. The Android OS architecture is provided. Dirisina et al. [2016] described the tremendous development with the increasing use of manual mobile devices. The methodologies of program implementation have also changed and new techniques have evolved. Kocakoyun et al. [14] developed a mobile application for undergraduate courses. Not only was the application developed and applied, but it was also used for eight weeks. In other words, rather than theory, the results are based on real-world experience. Students who worked on the creation and implementation of the NEU-CEIT Android application were polled on their opinions of the mobile learning environment, education, and shared structure of the app.

Alam et al. [6] presented the development of a robust mobile sensor for smartphone-controlled devices on microfluidic chips. In contrast to the classical spectrometer classics, the device's footprint is very compact and therefore a mobile unit. The mobile compressor device is made up of an embedded microcontroller, an optical sensor and a number of cellular scales that measure a wireless transceiver, produced by polymers from disposable microfluidic chips down.

Ekanayake et al.[2018] examines the android operating system structure and looks at the functioning of the operating system with new functions constantly added. Google developed an OS for mobile phones based on the Linux kernel. Android was called. Mallikarjuna et al.[15] explained the development of mobile healthcare on pervasive devices with several challenges; common people can benefit from the development of healthcare services in Android. Data acquisition, resource availability, security, and privacy are used to develop a mobile healthcare app for the cloud environment. Cahya et al.[10] described the educational sector technological developments on android-based learning.

B. Android Malware Detection

Several research papers analyze different Android malware detection technologies, some of which are novel, whilst others provide a new perspective on past research. The methods used vary from identification using host-based frameworks to static analysis of executable to function and behavior extraction. Each paper is scrutinized in detail, with the main characteristics of each technique emphasized and contrasted. The difficulties that these technologies face, as well as the potential prospects for Android malware detection, will be addressed.

Mutto et al. [17] explains the popularity of android apps is growing with malware for android. Malware authors are employing new technologies to build malicious Android apps that significantly limit the ability of traditional malware detectors to detect unknown malicious apps. Features gleaned from consistent and efficient Android application review can be combined with mechanical tutorials to detect Android malware. Riasat et al. [2017] described the widespread adoption of smartphones and the rapid growth of the contextually-sensitive nature of smartphone devices has now resulted in the revival of mobile app services and increased concerns about smartphone malware. Bhatia et al. [9] presented a dynamic analytical approach to android applications, which can be classified as malicious or not.

Odusami et al.[18] has conducted a survey of malware detection techniques to identify gaps and to help improve and effectively measure unknown android malware. Smartphones change every day in today's world and security becomes a major problem with this evolution. Surendran et al.[2018] discussed the current malware detection mechanisms and their drawbacks on smart devices in this study.

III. ANDROID MALWARE DETECTION METHODS

Researchers have done a lot of work in the area of detecting Android malware. This section looks at the different methods that have been used in the literature. The Figure 1 shows the detection process of malware detection.

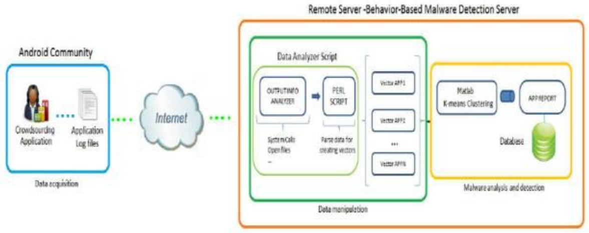

Fig. 1. Detection process of Malware Detection

A. Static Analysis

The majority of applications are not analyzed while in use. Since anonymous applications may access different signatures via obfuscation and encryption, static methods for detecting anonymous malware are ineffective. Combining attacker information as a function and separating unauthorized apps into related groups improves Android malware detection. By incorporating good analytical results, the model's typical distortion is reduced. Foot printing techniques, integrated static framework, logic properties- and dynamic operating time, allowed malware detection, behavioral similarity used to detect malware for Android. Permission to detect malicious software machine learning dividers in behavior patterns to distinguish applications that do not think of unsafe behavior due to the combination of permissions they need.

A description of the methods of standing by signing the foundation, too permission removed. A signature-based approach, for example, is used to generate and store impressions of recognized malware families in a database. When an anonymous app is compared to another app and the parallel score exceeds a certain threshold, the app is known as malware. Although signature-based approaches do not generate false positives, they are incapable of detecting new malware. The permission-based approach used both the server and client sides to verify the validity of applications.

Application status behaviour is classified as normal or malicious on the server side Sihwail et al.[20]; Hamid et al.[12].

B. Dynamic Analysis

Powerful analysis will detect the application's output during the sprint, and it's often done in the sandbox. On Android, the boot device sequence necessitates in-depth identification of deep malware patterns. Based on device logs, this approach examines the complex output of applications. Each app's system logs are used to build a database that categories the system as risky or normal.

C. Machine Learning Based

To train and test computers, the machine-based approach for detecting malware employs powerful analytics to extract features of software functions. To identify android malfunctions, powerful analysis and a strong stance, machine learning, and local and remote control are used. Malware detection has a high level of accuracy due to its storage and power consumption capabilities. To recognize illegal programmes based on app call series, return to neural network broadcasts. In illegal implementations, the static markov chain is a diagram of the sequence of device calls, and the chances of switching from one driving system to another are different from traditional systems. Install the malware detection reading for Android. Since no feature selection is needed, this method allows malware detection for zero days and thus provides reliability and durability in the access code. Malware on Android phones is identified by in-depth reading.

Two types of Deep Neural Networks are used in this in-depth learning method for malware detection. A trained Recurrent Neural Network is used to delete features, and a versatile neural network is used to separate images. Or, due to the use of a small database, a successful strategic outcome was achieved but never completely enforced. To detect malware for Android, a Hidden Markov Model (HMM) with a combination of structural entropy is used. To obtain negative codes, the mathematical pattern analysis algorithm used the length of the visual sequence, the number of visual clues, the view sequence, the matrix transformation matrix, and the first matrix status distribution matrix. Traditional machine learning mechanisms, such as back propagation neural networks, are not very deep and can only train with a limited amount of data, resulting in lower detection accuracy. A common method called depth learning using several layers of the neural network is able to train itself with a large database that has improved the accuracy of detection Damodaran et al.[11]; Sugunan et al. [21]; Raghuraman et al. [19]; Ashok kumar et al. [8]. Figure 2 shows the classification of smartphone malware detection techniques.

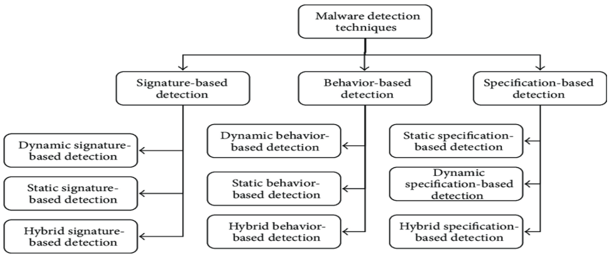
Malware detection techniques
Signature-based detection
Behavior-based detection
Specification-based detection
Dynamic signature-based detection
Static signature-based detection
Hybrid signature-based detection
Dynamic behavior-based detection
Static behavior-based detection
Hybrid behavior-based detection
Static specification-based detection
Dynamic specification-based detection
Hybrid specification-based detection

Fig. 2. Classification of Smartphone Malware Detection Techniques.

TABLE I. METHODS COMPARISON ANALYSIS

S.No	Method	Approaches	Pros	Cons
1	Static methods	Signature , Permission, Application Programming Interface, Filter, Contrasting permission pattern	Respond quickly to malware threats on Android. The false positive rate is zero, and the computational overhead is minimal. The degree of accuracy is very great. Low deployment costs. To detect malware, examine manifest files. Error is kept to a minimum.	False positives are created. Defend yourself against the most recent malware. Binder and device calls are intercepted by it. External dependencies are included. For detecting adware samples, this program is insufficient. The opposing pattern has a large interval between it.
2	Dynamic methods	Boot sequence, Honey pot, System calls log, Anomaly behavior monitoring	Malware with the intent of hijacking a user's session is detected. High efficacy at a reasonable cost. In a separate environment, run Android apps.	False negative rate is high. Root vulnerabilities and script malware are not detectable. Low efficiency and high energy consumption.
3	Machine learning methods	Back propagation Neural network, Deep belief network, Convolutional neural network, Recurrent neural network, Hybrid, Hidden markov model	Instantaneous attacks must be captured. Low false positive detection. A malicious programme can be detected with a 96–97 percent accuracy rate. Effectively detect unknown malware.	Because of the reliance on the server for communication, performance is hampered. There is a need for a lot of feature space. Malicious code that uses unrealistic dynamic analysis can evade detection. Experimentation dataset is insufficient. There are insufficient behavioural characteristics.

TABLE II. COMPARISON OF EXISTING WORKS

S.No	Title	Advantage	Disadvantage
1	DAMBA: Detecting Android Malware by ORGB Analysis W.	Efficient in terms of space and time. High flexible and simple.	Authenticity and integrity not achieved.

	Zhang, H. Wang, H. Heand P. Liu , IEEE Transactions on Reliability, March 2020		Hash chaining fails when some events are dropped.
2	Enhancing Malware Detection with Static Analysis using Machine Learning. Hamid and Fatema [12].	Highly efficient. Resilient to malicious data modification attack.	Fails to achieve the claimed security properties. Computation overhead.
3	Malware detection in android based on dynamic analysis ,Bhatia, Taniya, Kaushal & Rishabh [9]	It achieves desirable security and efficiency. Remote data integrity auditing is efficient.	It is a time-consuming process. High computation and communication overheads.
4	Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis Sihwail, Rami, Omar, et al (2018)	Efficient in terms of space and time. High flexible and simple.	Authenticity and integrity not achieved. Hash chaining fails when some events are dropped.
5	Static and Dynamic Analysis for Android Malware Detection Sugunan, KrishnaKumar et al [21].	High efficient. Resilient to malicious data modification attack.	Fails to achieve the claimed security properties. Computation overhead.

IV. MALWARE DETECTION CLASSIFIERS REVIEW

Mijwil et al. [16] introduced a new generation of Android malware families with the emergence of technological escape skills that make finding common methods more difficult. This article suggests and explores how to differentiate segmentally based on machine learning to detect malware for Android. App stores like Google Play run apps in various categories. Each category has its own different features. In terms of their static and dynamic features, applications within a specific category are similar. Benign apps in a particular category often share the same set of features. In contrast, malicious applications often have unpleasant side effects in their category. Ali et al. [17] Proposed categories of machine learning classification that improve the performance of class models of malicious applications that will be found in a particular category. Strict machine learning tests have shown that classification has very high performance compared to non-phase-based categories.

Anonymous applications cannot be detected using a traditional method that detects malware based on signature. Wen et al. [2017] they have proposed a lightweight reading machine that can detect malware on Android devices. In the article, the separation model will then be built using a vector support engine (SVM). Kedziora et al. [13] addressed malware detection problems in reverse engineering java code for android mobile applications. Mijwil et al. [16] used monitoring machine learning strategies to detect malware on the Android OS and its P: Random Forest (RF); Sustainable vector machine (SVM), Naïve Bayes (NB) and Decision Tree (ID3).

Abikoye et al. [1] provided a comprehensive review of engine learning strategies and their Android malware detection programs in content. In this paper, their strategies and programs for detecting malware of Android as found in modern books were completely reviewed. Agrawal et al. [3] aims to improve the acquisition of Android Malware through Classifiers Learning Machines. Analyzes are performed in this paper on various ways to detect malware for Android Machine Learning.

CONCLUSION

Because of the rapid development of Android malware, there are more threats for Android users. Ongoing research seeks to solve the limitations of previous malware detection methods. Since malware is becoming more complex and advanced, traditional methods such as signature-based and machine learning-based detection are no longer able to detect it in a timely and accurate manner. The review's results reveal the benefits and drawbacks of previous strategies. This paper offers a systematic and thorough overview of the Android platform and malware detection techniques used for Android malicious device analysis, classification, and recognition.

REFERENCES

[1] Abikoye, Oluwakemi & Gyunka, Benjamin & Oluwatobi, Akande. (2020). Android Malware Detection through Machine Learning Techniques: A Review. International Journal of Online and Biomedical Engineering (iJOE). 16. 14. 10.3991/ijoe.v16i02.11549

[2] Xu Jiang et al (2020) Android Malware Detection Using Fine-Grained Features, Hindawi Scientific Programming Volume 2020, Article ID 5190138.

[3] Agrawal, Prerna & Trivedi, Bhushan. (2021). Machine Learning Classifiers for Android Malware Detection. 10.1007/978-981-15-5616-6_22.

[4] Hongpeng Bai et al (2020) FAMD: A Fast Multifeature Android Malware Detection Framework, Design, and Implementation, ieee access.

[5] Ming Fan et al (2018) Android Malware Familial Classification and Representative Sample Selection via Frequent Subgraph Analysis, IEEE transactions on information forensics and security, vol. 13, no. 8, August 2018.

[6] Alam, Muhd & Kumar, Jayanth & Whyte, Daniel & Doeven, Egan & Kouzani, Abbas. (2018). A portable sensor for cell optical density measurement in microfluidic chips. Measurement and Control. 51. 002029401878344. 10.1177/0020294018783440.

[7] Ali, Huda & Oh, Tae & Fokoue, Ernest & Stackpole, Bill. (2016). Android Malware Detection Using Category-Based Machine Learning Classifiers. 54-59. 10.1145/2978192.2978218.

[8] K.Ashok Kumar,S.K.B.Sangeetha,S. Gayathri,K. Kamala, S. DivyaKeerthi(2021). IoT-enabled infrastructure privacy preservation in big data.European Journal of Molecular & Clinical Medicine, 2021, Volume 8, Issue 2, Pages 724-731

[9] Bhatia, Taniya & Kaushal, Rishabh. (2017). Malware detection in android based on dynamic analysis. 1-6. 10.1109/CyberSecPODS.2017.8074847

[10] Cahya, R & Suprapto, E & Lusiana, R. (2020). Development of Mobile Learning Media Based Android to Support Students Understanding. Journal of Physics: Conference Series. 1464. 012010. 10.1088/1742-6596/1464/1/012010

[11] Damodaran, Anusha & Di Troia, Fabio & Visaggio, Corrado Aaron & Austin, Thomas & Stamp, Mark. (2017). A comparison of static, dynamic, and hybrid analysis for malware detection. Journal of Computer Virology and Hacking Techniques. 13. 10.1007/s11416-015-0261-z

[12] Hamid, Fatema. (2019). Enhancing Malware Detection with Static Analysis using Machine Learning. International Journal for Research in Applied Science and Engineering Technology. 7. 38-42. 10.22214/ijraset.2019.6010

[13] Kedziora, Michal & Gawin, Paulina & Szczepanik, Michał & Jozwiak, Ireneusz. (2018). Android Malware Detection Using Machine Learning And Reverse Engineering. 95-107. 10.5121/csit.2018.81709

[14] Kocakoyun, Şenay & Bicen, Huseyin. (2017). Development and evaluation of educational android applications. Cypriot Journal of Educational Sciences. 12. 58. 10.18844/cjes.v12i2.1938

[15] Mallikarjuna, Basetty & D., Arunkumar. (2018). Mobile Healthcare Application Development on Android OS in Cloud Computing. SSRN Electronic Journal. 10.2139/ssrn.3169035

[16] Mijwil, Maad. (2020). Malware Detection in Android OS Using Machine Learning Techniques. 3. 5-9

[17] Muttoo, Sunil & Badhani, Shikha. (2017). Android malware detection: state of the art. International Journal of Information Technology. 9. 111-117. 10.1007/s41870-017-0010-2

[18] Odusami, Modupe & Abayomi-Alli, Olusola & Misra, Sanjay & Shobayo, Olamilekan & Damasevicius, Robertas & Maskeliunas, Rytis. (2018). Android Malware Detection: A Survey. 10.1007/978-3-030-01535-0_19

[19] Raghuraman, Chandni & Suresh, Sandhya & Shivshankar, Suraj & Chapaneri, Radhika. (2020). Static and Dynamic Malware Analysis Using Machine Learning. 10.1007/978-981-15-0029-9_62

[20] Sihwail, Rami & Omar, Khairuddin & Zainol Ariffin, Khairul Akram. (2018). A Survey on Malware Analysis Techniques: Static, Dynamic, Hybrid and Memory Analysis. 8. 1662. 10.18517/ijaseit.8.4-2.6827

[21] Sugunan, Krishna & Kumar, T. & Dhanya, K.. (2018). Static and Dynamic Analysis for Android Malware Detection. 10.1007/978-981-10-7200-0_13