Survey on Permission Based Android Malware Detection Techniques

[1]M.Mohana, [2]Dr.S.M.Jagatheesan
[1]Research Scholar, [2]Associate Professor
[1]Gobi Arts & Science College (Autonomous),
[2]Gobi Arts & Science College (Autonomous)

***Abstract* - Smart phone has become very popular nowadays; due to is portability and high performance, smart phone became a must device for persons using information and communication technologies. Most of the smart phone users are connected to internet for various usages. Even after the advanced technology used in the smart-phones still the users are remaining unprotected from malware attacks. Moreover, many kinds of Android applications require to many permissions than which they need to provide user's services. Android Operating Systems are most commonly used systems in the smart phones. Many applications are available in android play store and it is very difficult to distinguish and to discriminate between benign and malicious applications. To analyze the malwares, static and dynamic techniques are used. Static analysis has advantage of being undetectable, as malware cannot modify its behavior during run time. Despite number of detection's and analysis techniques are in place, high detection accuracy of new malwares is still a critical issue. In this literature survey paper, we aim to briefly discuss the exceptional strategies utilized in Android Malware Detection Techniques.**

***keywords* - Smart phone, Malware, Permissions, Malware attacks, Malware Detection**

I. INTRODUCTION

Android is currently the most used smart-phone devices platform in the world. The market of smart-phone will grow four times faster than mobile phone market. Unfortunately, the popularity of Android device also spurs interests from cyber-criminals who create malicious application that can steal sensitive information and compromise from mobile users. Unlike other competing smart-phone device platforms, such as iOS, Android allows users to install applications from unverified sources such as third-party app stores and file-sharing websites. The demand of smart-phones will increase until the customers will replace their old mobile phones with smart-phones. These devices are used to assist users in surf the Internet, receive and send emails, SMSs, and MMSs with other devices by activating various applications, thus making these devices potential attack targets [1]. A smart-phone can be partially or fully unusable because of malware which may cause unwanted billing, stealing of private information. Cellular networks, Internet connections (via Wi-Fi, GPRS or 4G network), USB and other peripherals are the most likely to be attacked in smart phones. A recent report indicates that a new malicious application for Android is introduced approximately every 10s. Malicious application is created to perform different types of attacks in the form of Trojans, Worms, Exploits and Viruses. Various malware detection tools have been developed, including signature based and behavior approaches. Considering the large amount of new malicious apps, we need a effective detection system that can operate efficiently to identify these apps. Google also identifies 24 permissions out of the total of more than 300 permissions as "dangerous" [2]. Android developer attaches permission in application which is stored in Androidmanifest.xml file for their specific accessing their camera, call and location resources. When installing android application user needs to accept the permission for successful installation. Android use permission for desired performance and communicate with other application that possible permission re-delegation attack means privileged task can malicious activity without application permission uses[3]. Permission combination is also become a malicious activity. e.g., Application requesting the permission ACCESS_COARSE_LOCATION and INTERNET permission can exhibit your location to other via internet connectivity; Application requesting READ_CONTACTS and INTERNET permissions can expose your contacts information when internet connection is available [4]. This paper review on various permission based malware technique to detection of malware.

II. MALWARE CLASSIFICATION

Malware is software that is inserted into the mobile device without user knowledge. It can harm the smart-phone by compromising mobile functions, stealing data or evading access controls. The following list presents the common categories of malware:

***Virus*:** A malicious Application that duplicates itself by injecting its code into other programs. Virus can spread from one program to another and from one device to another [5].

Worms: Are malicious programs that replicate themselves in a device and destroy the files and data on it. Worms might also encrypt files or send junk of bulk e-mails. Unlike viruses, worms carry themselves in their own containers [6].

Trojan horse: While acting as legitimate programs, Trojans perform unknown and unwanted activities [5]. Trojans allow attackers to gain access to the effective computer and extract user confidential information like password and banking details.

Spyware: Spyware is software that continuously spies on the users activities. It is used to gather information about the users like WebPages regularly visited and credit card number without their knowledge then sends that information back to the attackers [7].

Rootkit: Rootkit is a collection of malicious software that is programmed to access a computer system and allow other types of malware to get into the system [8].

Ransomware: Ransomware is harmful software that allows the hacker to lock the computer and restrict the victim access to the vital information. Ransomware encrypts the important data on the infected computer or network then asks for payment to lift the restriction [9].

Adware: Advertising-supported software is a type of malware that continuously brings advertisements to the computer. Usually adware is bundled with free downloaded software and applications like free playing games [10].

Botnet: A malware that remotely controls a group of devices like PCs, smart phones and Internet of Things (IOT) devices are infected and controlled by a cybercriminal. Botnet is typically used for spam emails campaigns or denial of service attacks. Users are often unaware that their systems are infected by a botnet malware [11]

III. MALWARE ANALYSIS TECHNIQUES

Malware Analysis refers to the process by which the purpose of functionality of the given malware samples are analysed and determined. There are two major types of malware analysis, Static and Dynamic.

Static Analysis

Static Analysis also called static code analysis is a process of software debugging without executing the code or program. In other words, it analyses the malware without analyse the code or executing the program. The techniques and tools instantaneously discover whether a file is malicious intent or not. Then the information on its functionality and other technical indicators help create its signatures.

Dynamic Analysis

The dynamic analysis technique runs malware to examine its behavior, learn its functionality and recognize technical indicators. When all these details are obtained, they are used in the detection signatures. The technical indicators exposed may comprise of IP addresses, domain names, file path locations, additional files, and registry keys, found on the network or computer.

Table 1 Comparison of Static and Dynamic Techniques

S.NO	STATIC	DYNAMIC
1	Fast & safe	Time Consuming & vulnerable
2	Good in analyzing the Multipath malware (Global view)	Difficult to analyze the multipath malware
3	Can't detect new, unknown malware	Detect known as well as unknown malware
4	Low level of false positive (accuracy is high)	High level of false positive (accuracy is low)

IV. MALWARE DETECTION APPROACH

Malware detection is the process of scanning the device and files to detect malware. It is effective at detecting malware because it involves multiple technique and approaches. The good thing is malware detection and removal takes less than 50 seconds only. Generally, malware detection technique can be categorized into three type's Signature-based Anomaly-based and Specification-based detection.

Signature-Based Detection

Signature based detection uses virus codes to identifying the malware. It is also called Misuse detection. Malware carries a special code that is used to identify it. When a code reaches the computer, the malware scanner collects the code and sends it to a cloud-based database. The database has a huge collection of virus codes. If the file code is found in the list, the database returns with a result that the file is malware.

Heuristic-Based Detection

It also called behavior or anomaly- based detection. The main motive is to analyze the behavior of known or unknown malwares. Behavioral parameter includes various factors such as source or destination address of malware, types of attachments, and other countable statistical features. It usually occurs in two phase: Training (learning) phase and detection (monitoring) phase. During the training the behavior of the system is observed in the absence of attack and machine learning technique is used to create a profile of such normal behavior. In detection phase, this profile is compared against the current behavior, and deviations are flagged as potential attacks. A key advantage of anomaly based detection is its potential to detect zero-day attacks. Zero-day attack is attacks that previously unknown to the malware detector.

Specification-Based Detection

Specification-based detection is a derivative of anomaly based detection that tries to beat the typical high false alarm rate associated with the anomaly-based detection. Specification-based detection relies on program specification that describes the intended behavior of security-critical program. It monitors executions program involve and detecting deviation of their behavior from the specification, rather than detecting the occurrence of specific attack patterns. This technique is similar to anomaly detection where they detect the attacks as vary from normal. The difference is that instead of relying on machine learning

techniques, it will be based on manually developed specifications that capture legitimate system behavior. It can be used to monitor network components or network services that are relevant to security, Domain Name Service, Network File Sharing and routers.

V. RELATED WORKS

In Android system, permissions requested by the app plays a major role in governing the access rights. By default, application has no permission to access the user data and affect the system security. During installation, user must allow the application to access all the resources requested by the apps. Developers must mention the permissions requested for the resources in the AndroidManifest.xml file. But all declared permissions are not necessarily the required permissions for that specific application.

Saracino *et al.* [12] has proposed "*MADAM: Effective and Efficient Behavior-based Android Malware Detection and Prevention*". Their paper presents a novel staggered and behavior based, malware finder for Android gadgets called MADAM (Multi-Level Anomaly Detector for Android Malware). Specifically, to distinguish application misbehaviors, MADAM screens the gadget activities, its cooperation with the client and the running applications, by recovering five gatherings of highlights at four distinct dimensions of deliberation, specifically the kernel level, application-level, client level and package level. For a few gatherings of highlights, MADAM applies an anomaly based methodology, for different gatherings, it actualizes a signature based methodology that considers standards of conduct that we have gotten from known malware misbehaviors.

Fan *et al.* [13] has proposed "*DAPASA: Detecting Android Piggybacked Apps through Sensitive Sub graph Analysis*". Their work recognizes Android piggybacked applications by using the discernable invocation examples of delicate APIs between the rider and carrier. Sensitive APIs are administered by consents for applications to get to sensitive data or to perform delicate undertakings. To additionally comprehend the recognizable invocation designs, two assumptions are set up dependent on an observational investigation of piggybacked applications.

Yerima *et al.* [14] has proposed "*DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection*". Their paper displays and examines a novel classifier fusion approach that uses a multilevel architecture to expand the prescient intensity of machine learning calculations. The system, called DroidFusion, is intended to prompt a classification Model for Android malware recognition by training various base classifiers at the lower level. A set of positioning based calculations are then used to infer combination plans at the larger amount, one of which is chosen to assemble the last model. The structure is able of utilizing not just traditional singular learning calculations like Decision Trees or Naive Bayes, yet in ensemble learning calculations like Random Forest, Random Subspace, Boosting and so forth for enhanced grouping exactness trained on a training set using a stratified N-fold cross- validation technique to assess their relative prescient correctness. The results are used by four distinctive positioning based calculations that characterize certainly criteria for the choice and ensuing mix of a subset of the relevant base classifiers. The results of the positioning, calculations are consolidated in sets with the end goal to discover the most grounded combine, which is in this manner used to construct the last DroidFusion demonstrate.

Li *et al.* [15] has proposed "*Significant Permission Identification for Machine Learning Based Android Malware Detection*". Their paper presents SIGPID, a methodology that extracts significant permissions from applications, and utilizes the extricated data to successfully recognize malware utilizing supervised learning algorithms. The goal of SIGPID is to recognize malware productively and precisely. This methodology investigates permissions and at that point recognizes just the ones that are noteworthy in recognizing malignant and benign applications. In particular, a multilevel data pruning approach including permission ranking with negative rate, permission mining with association rules and support based permission ranking to extract significant permissions strategically is proposed. At that point, machine learning based characterization calculations are utilized to arrange distinctive kinds of malware and benign applications.

Tong *et al.* [16] have presented "*A Hybrid Approach of Mobile Malware Detection in Android*". Their paper proposed a novel hybrid approach for mobile malware detection by embracing both dynamic and static analysis. Accumulation of execution data of sample malware and benign applications utilizing a net link technology to create patterns of system calls identified with document and system get to is done. Moreover, a malicious pattern set and an ordinary pattern set is developed by looking at the examples of malware and benign applications with one another. For identifying an obscure application, a dynamic technique to gather its system calling information is utilized. At that point, they are contrasted and both the malicious and ordinary pattern sets disconnected with the end goal to pass judgment on the unknown application.

Arp *et al.* [17] has proposed "*DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket*" performs a broad static analysis, gathering as many features from an application's code and manifest as possible. These features are organized in set of strings (such as permissions, API calls and network addresses) and embedded in a joint vector space. This method considers linear Support Vector Machines (SVM) for this task. Drebin identifies malware efficiently but it can't accessible obfuscation or dynamic execution code when retrieve.

Zhu *et al.* [18] Designed permlyzer a general purpose basic static and run time behavior framework. Permlyzer provide fine-grained information what permission actually used in run time environment. Permlyzer use call stack based analysis means when application's activity/service is started, application profiler log all the function invocation API method in an activity. Call stack use search tree based algorithm to improve the analysis. Using call stack method finds which permission is actually used in application.

Felt *et al.* [19] has proposed Stowaway tool that detect over privileged permission same as permlyzer but stowaway find permission static analysis and can't analyze complex reflective calls, the java reflection and obfuscate application. Permlyzer uses 51 malware families and over 110,000 applications and find the application is benign or malicious apps. It is helpful for developer to analyze application before publishing to the market.

Aswini *et al.* [20] has proposed static analysis of android malware file by Data mining using less misclassification and determine Bi-Normal Separation (BNS) and Mutual Information (MI) feature selection method on permissions from manifest file. It uses .apk file to permission extraction input to Androguard mining permission for detection malicious application with 209 malware samples- 105 samples for training set, 104 samples for test set and 227 benign samples-114 samples for training set and 113samples are in test set. This method proposed only initial classification of malware and benign detection.

Chouhan *et al.* [21] has presented "*A Preface on Android Malware: Taxonomy, Techniques and Tools*". Their paper describes the various approaches are presented to detect malware at two stages, I. Before execution i.e. Static approaches II. At the time of execution i.e. Dynamic approaches. Detailed description of techniques and tools used for malware detection in Android is given. Their paper acts as a base to understand the taxonomy of malwares in Android.

Sanz *et al.* [22] has proposed "*PUMA: Permission usage to detect malware in android"* for detection of malicious apps by analyzing the requested permissions for application. They used permission tags such as <uses-permission> and <uses-features> present in AndroidManifest.xml file to analyse the malicious behavior of apps and applied different classifier algorithms on dataset of 357 benign apps and 249 malicious apps. The solution provides high detection rate but results generated have high False Positives Rate (FPR) also it is not adequate for efficient detection of malware it still requires more information related to other features and dynamic analysis.

Tang *et al.* [23] has proposed a Security Distance Model for mitigation of Android malware. Security Distance Model is based on the concepts that not a single permission is enough for an application to threaten the security of Android devices. For example an application requesting permission of READ_PHONE_STATE can access the phone number and IMEI but it cannot move data out of the device.

Enck *et al.* [24] has developed *KIRIN*, a tool that provides light weight certification at run time. It defines the security rules and simply compares the requested permissions of app with its security rules and certifies the app as malware if it fails to pass all the security rules. The installation of application is aborted if the app is attributed as malware list. Authors have tested 311 applications downloaded from official Android market store and found that 5 applications failed to pass the specified rules. Proposed solution is light weight as it only analyses the Menifest.xml file. The limitation of *KIRIN* includes that it may also declare some legitimate applications as malware because the information provided for application certification is not adequate for detection of malware.

VI. CONCLUSION

Malwares are spreading around the world and affecting not only the end users but also large organizations and service providers. Android operating system (OS) seems to have attracted the most attention from malicious source code writer due to its popularity. Earlier, Signature based detection techniques were used to detect unknown malwares. But it was insufficient because these techniques were unable to detect unknown malwares (0-day attack). To analyze the malwares, static and dynamic techniques are used. Static analysis has advantage of being undetectable, as malware unable modify its behavior during analysis. Despite number of detection and analysis techniques are in place, high detection accuracy of new malwares is still a critical issue. This survey paper highlights the existing detection and existing analysis methods used for the android malicious codes. The available Android Malware Detection approach has not been able to provide exact accuracy. Most of approaches are based on permission-set only which was insufficient to detect new Android malware list. Few approaches consider few code properties but they were not able to provide good accuracy.

REFERENCES

[1] M. P. D. Sawle and A.B. Gadicha , *Analysis of Malware Detection Techniques in Android*, IJCSMC, (2014) 176-182.

[2] G.Android, "*Requestiong permissions.*"2017.[Online].Available:https://developer.android.com/guide/opics/ permissions / requestiong.html

[3] A. P. Felt, J. Helen. Wang and A. Moshchuk, "*Permission Re-Delegation: Attacks and Defenses*", Usenix Security, 2011

[4] S. Liang and X. Du,"*Permission combination based scheme for android mobile malware detection*", IEEE, 2014

[5] M. Karresand, "*Separating Trojan horses, viruses, and worms – A proposed taxonomy of software weapons,*" in IEEE Systems, Man and Cybernetics Society Information Assurance Workshop, 2003, pp. 127–134.

[6] X. Wang, W. Yu, A. Champion, X. Fu, and D. Xuan, "*Detecting worms via mining dynamic program execution,*" in Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks, SecureComm, 2007,pp. 412–421.

[7] Y. Ye, T. Li, D. Adjeroh, and S. S. Iyengar, "*A Survey on Malware Detection Using Data Mining Techniques,*" ACM Comput. Surv., vol. 50, no. 3, pp. 1–40, 2017.

[8] A. Zaki and B. Humphrey, "*Unveiling the kernel: Rootkit discovery using selective automated kernel memory differencing,*" Virus Bull., no. September, pp. 239–256, 2014.

[9] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "*CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data*," in Proceedings - International Conference on Distributed Computing Systems, 2016, vol. 2016–Augus, pp. 303–312.
[10] G. A. N. Mohamed and N. B. Ithnin, "*Survey on Representation Techniques for Malware Detection System*," Am. J. Appl. Sci., vol. 14, no. 11, pp. 1049–1069, 2017.
[11] M. Chowdhury and A. Rahman, "*Malware Analysis and Detection Using Data Mining and Machine Learning Classification*," in International Conference on Applications and Techniques in Cyber Security and Intelligence, 2018, vol. 580, pp. 266–274.
[12] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, "*Madam: Effective and efficient behavior-based android malware detection and prevention"*. IEEE Transactions on Dependable and Secure Computing, 15(1), 2018, pp.83-97.
[13] M. Fan, J. Liu, W. Wang, H. Li, Z. Tian and T. Liu. *DAPASA: Detecting Android Piggybacked Apps*. IEEE Transactions on Information Forensics and Security. Volume 12, Issue 8.2017.
[14] S.Y. Yerima, and S. Sezer, *DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection*. IEEE Transactions on Cybernetics.2018.
[15] J. Li., L. Sun, Q. Yan, Z. Li, Srisa-an, W. and Ye, H. "*Significant Permission Identification for Machine Learning Based Android Malware Detection*". IEEE Transactions on Industrial Informatics, 2018.
[16] F. Tong and Z. Yan, *A hybrid approach of mobile malware detection in Android*. Journal of Parallel and Distributed Computing, 103, 2017, pp.22-31.
[17] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, " *DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket*", Internet Security, 2014
[18] W. Xu, F. Zhang, and S. Zhu, "*Permlyzer: Analyzing Permission Usage in Android Applications*", IEEE, 2013
[19] A. P. Felt, E. Chin, S. Hanna, D. Song and D. Wagner, "*Android Permissions Demystified*", IEEE, 2011
[20] A. M. Aswini and P. Vinod, "*Droid Permission Miner: Mining Prominent Permissions for Android Malware Analysis*", IEEE, 2014
[21] R.R Chouhan, and A.K.Shah, *A Preface on Android Malware: Taxonomy, Techniques and Tools*. International Journal on Recent and Innovation Trends in Computing and Communication, 5(6),2017, pp.1111-1117.
[22] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. G. Bringas, and G. Álvarez, "*PUMA: Permission usage to detect malware in android*," Adv. Intell. Syst. Comput., vol. 189 AISC, pp. 289–298, 2013.
[23] W. Tang, G. Jin, J. He, and X. Jiang, "*Extending android security enforcement with a security distance model*," 2011 Int. Conf. Internet Technol. Appl. iTAP 2011 - Proc., 2011.
[24] W. Enck, M. Ongtang, and P. McDaniel, "*On lightweight mobile phone application certification,*" Proc. 16th ACM Conf. Comput. Commun. Secur. - CCS '09, pp. 235–245, 2009.