*International Journal of Latest Trends in Engineering and Technology*
*Vol.(11)Issue(1), pp.044-047*
*DOI: http://dx.doi.org/10.21172/1.111.08*
*e-ISSN:2278-621X*

# SURVEY ON DETECTION OF MALWARE IN ANDROID

D.Rubiya Sweetlin[1], Dr. A.S.Radhamani[2]

**Abstract - Today, Malware is spread all around the world and which infects not only every single user but also big organizations. Android is very popular nowadays which can be used by many user to share their information, and hence several malicious writer aims to attack the android operating system due to its popularity. The detection methods can be classified into two types: Static and dynamic approach. In Static approach, Signature Based Method uses the signature which is a sequence of code to identify the malicious software. But there are some limitations that it cannot detect new malware and limited signature database. In Dynamic approach, it can detect new malware arrived in device. In this paper, a brief survey of all the malware detection approach that has been used in the field of Android was provided with the merits and demerits.**
**Keywords - Android Malware, Machine Learning, Malware Detection Techniques**

## 1. INTRODUCTION

Android is an adaptable working system in perspective of open source programming and arranged basically for touchscreen PDAs, for instance, mobile phones and tablets. Android is the open source that it increases risk and increment the issues identified with the malicious applications. Android client can download numerous free android applications from the application store. Now and again these downloads may contains malware programs that can take the client private data.

Malware is any bit of programming which is planned to harm your framework or system. Malware is not quite the same as the vast majority of them can spread itself in the system, stay imperceptible, cause changes/harm to the infected system or network, persistence. They can cut down the machine's execution to knees and can cause a destruction of the network. Malware has evolved with the technology and has taken full advantage of new technological developments. A program that classified as malware if it does the following activities: allows an unauthorized person to take control over the system, injects the code to another program, and connects to suspicious servers, send confidential information to other system without user's permission.

Malware come in different structures like trojans, back-passages, worm,botnets, spywares, ransomwares, and riskwares. Various approaches must be concocted to battle them based on their nature. The two primary methodologies are static analysis furthermore, dynamic analysis. The malware recognition applications utilize these systems in their attempting to alarm the clients if the application indicates malicious behavior. This paper is proposed to discuss about the android malware detection techniques along with the advantages and limitations. The paper is organized as literature survey, summary, conclusion and the future work.

## 2. LITERATURE SURVEY

In this section, various android malware detection techniques that are used in the various papers are discussed.

AndroSimilar: robust statistical feature signature for android malware detection [1] utilizes a syntactic foot-printing mechanism that identifiessimilar regions by statistical strategies utilizing known malwares to find the obscure variations of existing malwares. It classifies an app as malware or benign based on variable length signatures which are further compared with signatures in the AndroSimilar database. The paper reports an accuracy of 60%. This paper is based on the signature based detection. It centers around ordering a code as malware based on the signature by comparing it and existing malware family signature. It is exceptionally effective however neglects to distinguish the obscure malwares because of restricted size of signature database.

Droid Analytics: A signature based analytic system to collect, extract, analyze and associate android malware [2] method extracts and analyses the apps in the op-code level. First a signature can be generated at the method level. In Later, signature methods, application signatures along with traces can obtain from the API calls are used to generate signatures at the class level which can be used for the further analysis. In this paper, totally 150,368 android applications were collected as a sample. In that from 102 families 2,494 malware samples were determined. Among those, there were 342 zero-day malware samples from six different malware families. This paper uses three level signature of app to detect the malware. It can detect zero day repackaged malware. But this method relatively higher rates of false positives.

Stowaway: Android permissions demystified [3] compiled android apps had to be tested for over-privilege using this tool. This toolcan easily determine the API callthat was made by the app and identifies the permissions that are required for each API call using a permission map. Stowaway was applied to a set of applications nearly 940 and it was found that about 33%

[1] Department of Computer Science and Engineering, VV College of Engineering, Tisaiyanvilai, Tamil Nadu, India
[2] Department of Computer Science and Engineering, VV College of Engineering, Tisaiyanvilai, Tamil Nadu, India

were over-privileged. The cause this developer error may be because of a lack of information about reliable permission. This paper uses the static analysis and permission maps to detect over-privileges in the applications. This method generates maximum set of permissions which are needed for an application. But this type of method can't able to handle the complex reflective calls.

Tang, Wei: Extending android security enforcement with a security distance model [4] proposed a security distance model which utilizes the idea that it needs more than one per-mission to challenge the Android devices security. Permissions can be classified into four categories by authors, namely safe, normal, dangerous and severe security distance.The idea of this method was to create awareness among users about the dangers that are come by giving permissions blindly. This method identifies the malware at install time and which is highly scalable. But noteasy to classify if the threat point is between 50 and 100.

Enck developed KIRIN: On lightweight mobile phone application certification [5], proposed scheme called a light weight tool that provides certification at installation time. An app is categorised as malware only when it is unable to pass all of its security tests. These tests are a comparison of the security rules and the permissions requested by the app. 10 apps were found to request for dangerous permissions out of the 311 apps tested spanning 16 categories. This method gives a collection of rules that are compared with the collection of requested permissions and if rules are not matched with the predefined rules it is declared malicious. In this method, legitimate apps certification at maybe declared install time.

Droid Mat: Android malware detection through manifest and API calls tracing [6] proposed a behavior based detection which works by extracting information from the androidmanifest.xml file (that is manifest files). It further analyses the app using permissions and API call tracing mechanism. To improve the working of the classifier, K- means classifier is used along with K-nearest neighbor algorithm. This method takes lesser time and cost saving as it does not require samples simulation and manual efforts. And light weight. But it is inadequate for detecting adware samples.

DroidAPIMiner: Mining API-level features for robust malware detection in android [7]: DroidRanger joins consent based behavioral impression and a separating plan contingent upon heuristics to recognize the nearness of malware. Programming interface level data is utilized to separate amongst pernicious and benevolent applications. Through various examination steps, basic APIs which happen in malignant applications at a higher recurrence are distinguished. This apparatus has the accompanying restrictions: Malware creators can utilize reflections to muddle hazardous API calls. It may give a poor outcome if malware creators incorporate more generous API calls into their application. 99% exactness with false positive rates of 2.2% was accounted for. This technique utilizes an assortment of classifiers for better outcomes. Yet, at some point Malware creators can trick the application utilizing jumbling.

Static analysis of executable for collaborative malware detection on android [8]: This strategy depends on a lightweight and static technique to identify malwares on android frameworks. This instrument devours fewer assets influencing it to fit for portable clients. The approach includes a capacity call examination, information extraction, preparing set creation and characterization of executable utilizing classifiers like PART, crystal and closest neighbor classifier. This technique utilizes numerous classifiers. However, this technique is powerless to assaults.

DREBIN: Effective and explainable detection of android malware in your pocket [9]: proposed a light weighted method which allows us to identify the malicious apps on a smartphone and does a broad static analysis. The features which are enclosed in a joint vector space allowing the ordinary patterns that indicates the malware to be identified automatically and are also used for explaining the decisions. DREBIN tracks down 94% of the malicious cases with a few false alarms outperforming the other approach of malware detection. This method is suitable for analyzing the downloaded applications as the checking time is only about 10s. In order to alleviate the absence of dynamic analysis, it provided features that enabled us to spot the execution of hidden code. The quality of results produced by DREBIN relies on the accessibility of malicious and benign apps. Use of machine learning brings in the prospect of poisoning and mimicry attacks like renaming of activities and components during the learning stage. This method is mainly suitable for checking downloaded applications. But the use of machine learning brings about mimicry and poisoning attacks.

**3. SUMMARY**

Table - 1 shows the list of malware detection techniques.

| SI No | Topics | Description | Advantages | Limitations |
|---|---|---|---|---|
| 1 | Andro Similar | Uses a signature based approach to detect unknown variants of existing malware | Useful to detect malware generated by obfuscation and Repackaging | Limited signature databases. |
| 2 | Droid Analytics | Generates 3 level signatures of apps to detect malware | Analyses app at op-code level. More robust than cryptographic hash methods. Can detect zero day repackaged malware. | Relatively higher rates of false positives. |

| | | | | |
|---|---|---|---|---|
| 3 | Stowaway | Uses static analysis and permission maps to detect over privileges in apps | Generates maximum set of permissions needed for an application. | Unable to handle complex reflective calls. |
| 4 | Security Distance Model | Uses the idea that more than one permission is required to threaten the security of android devices. | Identifies the malware at install time.<br>Highly scalable. | Not easy to classify if the threat point is between 50 and 100. |
| 5 | KIRIN | Gives a set of rules which is compared with the set of requested permissions and is declared malicious if the rules are not satisfied. | Light weight certification at install time. | Legitimate apps maybe declared as malware. |
| 6 | Droid Mat | Extracts information from manifest file and analyses the behavior of application. | Lesser time and cost saving as it does not require dynamic simulation and manual efforts.<br>Light weight. | Inadequate for detecting adware samples. |
| 7 | Droid API Miner | Combines permission based behavioral footprint and a heuristic based filtering scheme. | Uses a variety of classifiers for better results. | Malware authors can fool the app using obfuscation. |
| 8 | Static Analysis of Executable for Collaborative Malware Detection on Android | Uses collaboration for security approach to extent malware detection results. | Light Weight.<br>Uses many classifiers. | Higher rate of false positives.<br>Susceptible to attacks. |
| 9 | DREBIN | Detects Malware on the smart phone itself | Light Weight app.<br>Features are embedded in a joint vector space.<br>Suitable for checking downloaded applications. | Lacks dynamic analysis.<br>Use of machine learning brings about mimicry and poisoning attacks. |

## 4. CONCLUSION AND SCOPE OF FUTURE WORK

In this paper the various kinds of approaches used to detect malware in android are discussed. The available Android malware detection approaches has not possessed the capacity to give better accuracy. A large portion of methodologies depend on authorization set just which was inadequate to distinguish new Android malware. It has been observed that static approaches are generally faster, provide higher accuracy rates and is effective against existing malware attacks whereas dynamic approaches perform well and can detect malware variants and uses a real time analysis. We can conclude that no single approach is enough to make a system secure.

## 5. REFERENCES


[1] Parvez Faruki, Ammar Bharmal, Manoj Singh Gaur, Vijay Laxmi, and Vijay Ganmoor. Androsimilar: robust statistical feature signature for android malware detection. Proceedings of the 6th International Conference on Security of Information and Networks, pages 152–159, 2013.

[2] Min Zheng, John CS Lui, and Mingshen Sun. Droid analytics: a signature based analytic system to collect, extract, analyze and associate android malware. Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on, pages 163–171, 2013.

[3] Adrienne Porter Felt, Steve Hanna, Erika Chin, Dawn Song, and David Wagner. Android permissions demystified. Proceedings of the 18th ACM conference on Computer and communications security, pages 627–638, 2011.

[4] Wei Tang, Jiaming He, Guang Jin, and Xianliang Jiang. Extending android security enforcement with a security distance model. Internet Technology and Applications (iTAP), 2011 International Conference on, pages 1–4, 2011.

[5] William Enck, Patrick McDaniel, and Machigar Ongtang. On lightweight mobile phone application certification. Proceedings of the 16th ACM conference on Computer and communications security, pages 235–245, 2009.

[6] Dong-Jie Wu, Te-En Wei, Ching-Hao Mao, Kuo-Ping Wu, and Hahn- Ming Lee. Droidmat: Android malware detection through manifest and api calls tracing. Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on, pages 62–69, 2012.

[7] Yousra Aafer, Heng Yin, and Wenliang Du. Droidapiminer: Mining api-level features for robust malware detection in android. International Conference on Security and Privacy in Communication Systems, pages 86–103, 2013.

[8] A-D Schmidt, H-G Schmidt, Rainer Bye, Jan Clausen, Kamer A Yuksel, Osman Kiraz, Seyit Ahmet Camtepe, and Sahin Albayrak. Static analysis of executables for collaborative malware detection on android. Communications, 2009. ICC'09. IEEE International Conference on, pages 1–5, 2009.

[9] Daniel Arp, Malte Hubner, Michael Spreitzenbarth, Konrad Rieck, CERT Siemens, and Hugo Gascon. Drebin: Effective and explainable detection of android malware in your pocket. NDSS, 2014.

[10] Asaf Shabtai, Yuval Elovici, Uri Kanonov, Yael Weiss, and Chanan Glezer. andromaly: a behavioral malware detection framework for android devices. Journal of Intelligent Information Systems, 38(1):161– 190, 2012.

[11] Min Zhao, Tao Zhang, Fangbin Ge, and Zhijian Yuan. Antimaldroid: An efficient svm-based malware detection framework for android. International Conference on Information Computing and Applications, pages 158–166, 2011.

[12] William Enck, Seungyeop Han, Peter Gilbert, Vasant Tendulkar, Landon P Cox, Byung-Gon Chun, Jaeyeon Jung, Anmol N Sheth, and Patrick McDaniel. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM Transactions on Computer Systems (TOCS), 32(2):5, 2014.

[13] Lok-Kwong Yan and Heng Yin. Droidscope: Seamlessly reconstructing the os and dalvik semantic views for dynamic android malware analysis. USENIX security symposium, pages 569–584, 2012.

[14] Yu Feng, Isil Dillig, Saswat Anand, and Alex Aiken. Apposcopy: Semantics-based detection of android malware through static analysis. Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, pages 576–587, 2014.

[15] Annamalai Narayanan, Lihui Chen, Mahinthan Chandramohan, and Yang Liu. Context-aware, adaptive and scalable android malware detection through online learning (extended version). arXiv preprint arXiv:1706.00947, 2017.

[16] Thomas Bl¨asing, Aubrey-Derrick Schmidt, Leonid Batyuk, Sahin Albayrak, and Seyit Ahmet Camtepe. An android application sandbox system for suspicious software detection. Malicious and unwanted software (MALWARE), 2010 5th international conference on, pages 55– 62, 2010.