Behavioural Analysis of Android Malware and Detection

Lokesh Vaishanav[1], Shanu Chauhan[1], Sneha Kumari[1], Mahipal Singh Sankhla[2], Dr. Rajeev Kumar[3]

[1]*Students of Bachelor of technology in Computer Science, School of Computer Science Engineering, Galgotias University, Greater Noida.*
[2]*Student of M.Sc. Forensic Science,* [3]*Assistant Professor, Division of Forensic Science, School of Basic and Applied Sciences, Galgotias University, Greater Noida.*

Abstract

Malware is a malevolent software used to disturb installed application, collect information, or gain access to a computer system or mobile device that is chiefly built to attack various operating systems and their application. Since last decade, we are using smartphones excessively and due to its open source nature and widespread popularity it is becoming the main aim of cyber terrorist. The advancement of security threat and privacy leakage are becoming more vulnerable without users attention. Since effective mechanism to identify malicious application for blocking their entry into android market place is hindered that's why the attackers are becoming more and more powerful in the market. In this paper we are going to analyse the behaviour of various android malwares and threats. At last we are going to discuss various malware detection techniques

Key-words: *Android, Malware, Attack, Threats.*

INTRODUCTION

Smart phones and tablets are the most popular and widely used personal electronics devices today and roughly 70% of these devices run Android **[1].** With the reduction of smart phone's cost and the rapid development of the application software based on smart phones, the smart phone becomes more and more popular in people's life in recent years. Due to the characteristic of openness, the share of Android smartphone is continually increasing in smartphone market. According to IDC, the market share of Android smartphone is up to 81 percent till the third quarter of 2013, which far exceeds all the other competitors. Android smartphone has brought great convenience to people's life, at the same time, it causes problems of security. CNCERT has detected 702, 861 mobile internet malicious sample programs in 2013, among which 99.5 percent aims at Android platform [2]. Android is the popular platform for smart-phone based malware authors as any third-party vendor can create applications for android phones and deploy it on android market. Sometimes, even trusted applications are able to leak user's location and phone's identity and share it on server without its consent. Due to this growing skill-set of cyber-criminals who device their algorithms for breaching privacy, embarrassing service-provider and bring inconvenience to the users. So, it requires special care to secure these networked devices from malwares with the help of anti-developed techniques and algorithms for detection. This paper focuses on describing mobile-based threats and its counter detection techniques [3].To make matters worse, there are other, third-party markets that are available for downloading Android applications that can easily purvey malicious applications [4, 5]. Another challenge is that not all applications reveal their malicious behavior when they are installed or even run on the device. Instead, the malicious behavior can be triggered based on different conditions. Forinstance, a group of malware can stay dormant until they are triggered by an event [6]. Some events are independent of user interactions with the application (i.e. existence of network, etc), yet some others are based on user input [7,8]. However, most software testing is performed towards checking the quality of software or apps but recently security testing has gained popularity and has proved its importance [9]. However, testing mobile application is not a straight forward task due to variety of inputs and heterogeneity of the technologies [10].

Android System Structure

Android operating system comprise of different software layers as shown in Figure 1:

Linux kernel: Is the Bottom layer of android operating system which provides the basic system functionality such as process management, memory management, device management and device drivers which make our task easier while interfacing the android with other devices.

Libraries : Libraries are Java libraries build specific for android operating system. It provides the

different libraries useful for well-functioning of android operating system

Application Framework: This layer directly interacts with the Android applications and also manages the basic functions of android device such as location management, voice call management and resource management [12]

Android Runtime : it includes set of core Java libraries. Application programmers use Java programming language for developing apps. It includes the Dalvik Virtual machine and Core Java libraries [13].

Dalvik Virtual Machine: A virtual machine runs as if it was an independent device having its own operating system. It permits numerous instances of virtual machine to be executed simultaneously providing isolation, security, memory management and threading support [14]. Each application runs as its own process in a virtual machine such that no other application is dependent on it and in case of application crash, it would not affect any other application running on the device [15]. These features together are called the sandbox.

Core Java Libraries: Most of the functionality defined in the Java SE libraries, including tasks such as string handling, networking and file manipulation is provided by these libraries [13].

Applications: This is the final destination for the developer journey where his application installed and run here. This layer communicate with the underlying layers through different form of Inter Process Communication (IPC) Endpoints[12].

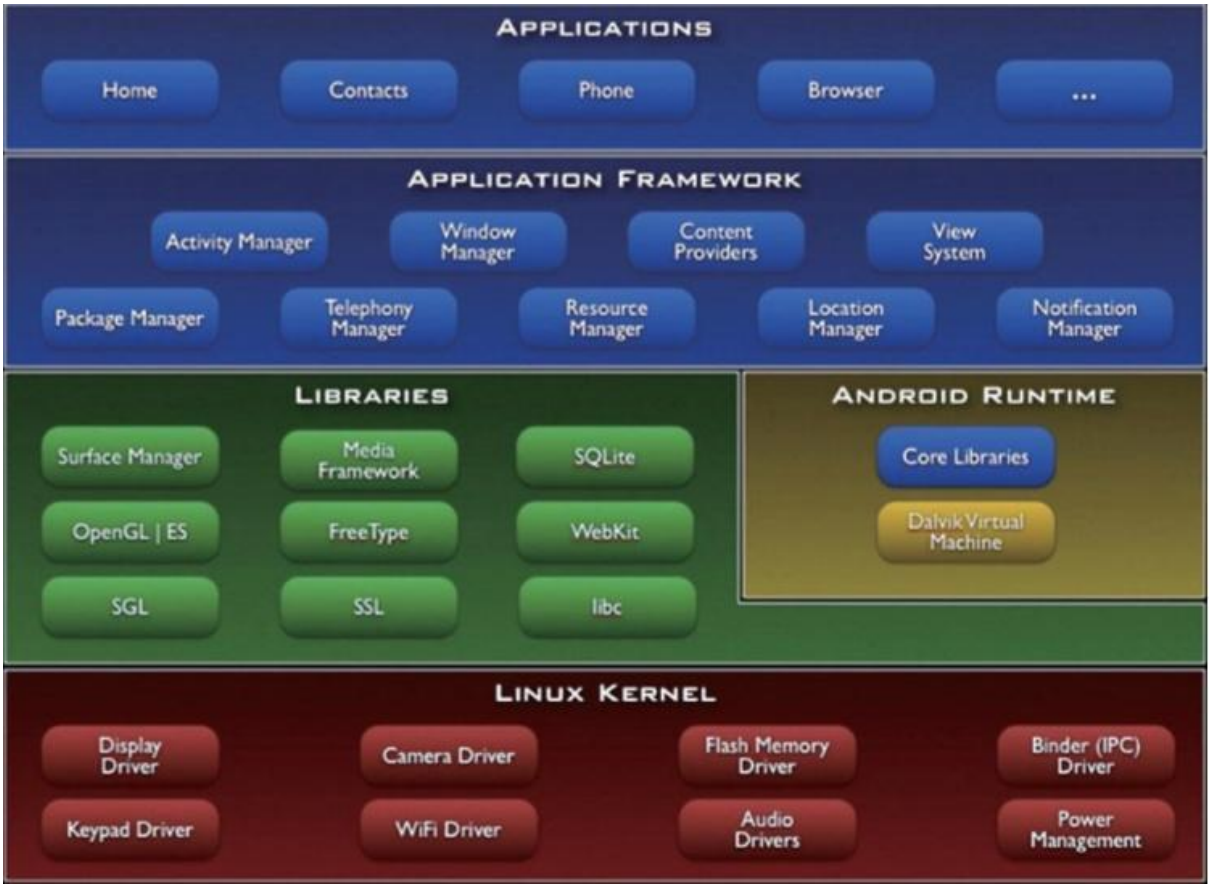

Figure 1:- Android System Structure [11].

What is Mobile Malware?

Generally malware is a software designed to damage a computer system without the owner's knowledge or consent. Malware is short for malicious software and includes viruses, Trojan horses, spy-ware and or any other unwanted or malicious software. Malware is a crime and the most common path criminal's use is through the Internet. Malware is being produced at an increasingly alarming rate. Unfortunately, it is well known that computers already attacked by a malware so we are always looking forward to update our anti-virus application, but what about the malwares which attack our mobile devices? As we will discuss and implement our proposed spyware application in the next section. We will see by practical experiment that mobile malicious applications on the rise, and the intercepted information from our cell phones are

horrible. For instance, we could accidentally download a malicious application that hijacks our identity, personal photos, contacts, SMS,emails and our private information and sends them to a remote server. Or, we could download a dangerous application that sends SMSs or dialing services numbers from our phone, charging us with an expensive money on our mobile bill. Other malicious programs can potentially malfunction our cell phone basic functionality[12].

Spyware and Adware

Spyware secretly gather confidential information about the mobile users and relays this data to third party. In some cases these may be advertisers or marketing data firms, which is why spyware is sometimes referred to as "adware". Spyware uses the victim's mobile connection to relay personal information such as contacts, location, message habits, and browser history and user preferences or downloads. Spyware also gathers devices information such as OS versions, product ID, International Mobile Equipment Identity number, and International Mobile Subscriber Identity number which can be used for future attacks [16].

Trojans and Viruses

Mobile Trojans infect user devices by attaching themselves to seemingly harmless or legitimate programs, Trojans are installed with the app and then carry out malicious actions. Trojans are closely related to mobile viruses. Malicious parties can use mobile viruses to root the device and gain access to files and flash memory [16].

Phishing Apps

Just like desktop computing, attackers are creating mobile phishing apps that look like legitimate services but may steal sensitive information and credentials to perform financial fraud. One such recent example was a fake security app for Facebook, which claimed to secure user's Facebook account but infact stole user's information for identity theft [16].

Bot Processes

Mobile malware is getting more sophisticated with programs that operate in the background on the user devices, concealing themselves and lying in wait for certain behaviors like an online banking session to strike. Hidden processes can execute completely invisible to the user, run executable or contact botmasters for new instructions [16].

Mobile Malware Symptoms:

These types of mobile malware differ greatly in how they spread and infect devices; they all can produce similar symptoms. Signs of malware infection can include munwanted behaviors and degradation of device performance. Mobile malware can reduce battery life or processing power, hijack the browser, send unauthorized SMS message and freeze the device entirely [16].

ANDROID MALWARE DETECTION

There are mainly two approaches to analyze the Android malwares: Static and Dynamic Approach. We have further categorized the antimalware using static and dynamic approaches. Figure-2 shows the taxonomy of existing antimalware techniques based on our study [17].

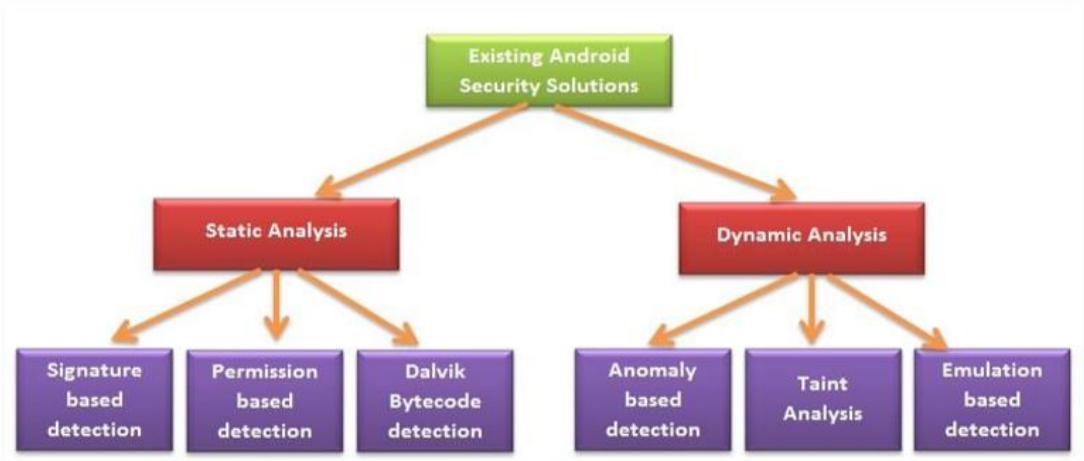

Figure-2 Taxonomy of Existing Android Antimalwares [17].

Static Approach

Static approach is a way to check functionalities and maliciousness of an application by disassembling and analyzing its source code, without executing the application. It is useful for finding malicious behaviors that may not operate until the particular condition occurs [17].

1. **Signature Based Approach:** Signature based malware detection methods are commonly used by commercial antimalware products. This method extracts the semantic patterns and creates a unique signature [18]. [25]. A program is classified as a malware if its signature matches with existing malware families‟ signatures. ProposedAndroSimilar, a robust statistical signature method to detect the unknown variants of existing malwares that are usually generated by using repackaging and code obfuscation techniques [19]. [26]. It generates the variable length signature for the application under test and compares it with the signatures in AndroSimilar malware database and identify the app as malware and benign on the basis of similarity percentage. Authors tested the AndroSimilar against 1260 apps among which 6779 apps were Google Play apps and 545 apps were from third party app store.DroidAnalytics[20]. [27] is a signature based analytic system which extract and analyze the apps at op-code level. It not only generates the signature but also associate the malware with existing malwares after identifying the malicious content. It generates 3 level signatures. First it generates signature at method level by API call tracing then combining all the signatures of methods in a class it generates the class level signatures and at third level it generates the application signature by combining the signatures of the classes in the application. Authors have used DroidAnalytics to detect 2,494 malware samples from 102 malware families and 342 repackaged malwares from other six malware families [17].

2. **Permission Based Analysis:** In Android system, permissions requested by the app plays a vital role in governing the access rights. By default, apps have no permission to access the user‟ data and effect the system security. During installation, user must allow the app to access all the resources requested by the app. 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 [17].

3. **DalvikBytecode Analysis:** In Android, Dalvik is a register-based VM. Android apps are developed in java language, compiled in java bytecode and then translated to dalvik byte code. Bytecode analysis helps us to analyze the app behavior. Control and data flow analysis detect the dangerous functionalities performed by malicious apps [17].

Dynamic Approach

Dynamic analysis examines the application during execution. It may miss some of the code sections that are not executed but it can easily identify the malicious behaviors that are not detected by static analysis methods. Although static analysis methods are faster to malware detection but they fail against the code obfuscation and encryption malwares. In Egele provided a detailed overview of different dynamic analysis methods used for discrimination between malware and benign apps. Dynamic analysis approach is effective against polymorphic and metamorphic code obfuscation techniques employed by the malwares but it requires more resources [21, 22].

1. **Anomaly Based Detection:**proposed CrowDroid to detect the behavior of applications dynamically. Details of system calls invoked by the app are collected by the Strace tool and then crowdsourcing app, which is installed on the device, creates a log file and sends it to remote server[23, 24]. Log file may include the following information: Device information, apps installed on device and system calls. 2-mean clustering algorithm is applied at server side to classify the application as malware or benign. Results are stored at server database. The solution provides deep analysis and thus require large amount of resources. The solution requires client app to be installed on the user‟ device and may classify the legitimate app as malware if it invoke more system calls, proposed Andromly , a behavior based Android malware detection system. In order to classify the application as benign or malware it continuously monitor the different features and patterns that indicate the device state such as battery level, CPU consumption etc. while it is running and then apply the machine learning algorithms to discriminate between malicious and Benign apps. the solution can detect continuous attacks and can notify the user about these attacks [25].AntiMalDroid, a malware detection framework using SVM algorithm is proposed by Zhao, can identify the malicious apps and their variants during execution. First it monitors the behavior of applications and their characteristics then it categorize these characteristics as normal and malicious behavior. Then it puts the two types of characteristics into learning module and generates the signatures for the behavior characteristics, produced by learning module. Then it store the signature in database and compare it with the already existing malware and benign app signatures. It classify the app as benign if the signature matches with already existing benign app‟ signatures. The solution can extend the signature database dynamically and

can provide high detection rate. But it consumes more time while detection process [26].

2. **Taint Analysis:**proposed TaintDroid which provides system-wide information flow tracking for Android. It can simultaneously track multiple sources of sensitive data such as camera, GPS and microphone etc. and identify the data leakagein third party developer apps. It labels the sensitive data and keeps track of that data and app when tainted data leaves moves from the device. It provides efficient tracking of sensitive information but it do not perform control flow tracking. Also it cannot track information that leaves deice and returns in network reply[27].

3. **Emulation Based Detection:** present Android dynamic analysis platform DroidScope, based on Virtual Machine Introspection. As the antimalware detect the presence of malwares because both of them reside in the same execution environment so the malwares also can detect the presence of antimalware. DroidScope monitors the whole operating system by staying out of the execution environment and thus have more privileges than the malware programs. It also monitors the Dalvik semantics thus the privilege escalation attacks on kernel can also be detected. It is built upon QEMU[28].Were detected with this technique.Proposed Android Application Sandbox (AASandbox) which detect the suspicious applications by performing both static and dynamic analysis on them. It first extracts the .dex file into human readable form and then performs static analysis on application. Then it analyzes the low level interactions with system by execution of application in isolated sandbox environment. Actions of application are limited to sandbox due to security policy and do not affect the data on device. It uses Money tool to dynamically analyze the application behavior which randomly generates the user events like touches, clicks and gestures etc. it cannot detect the new malware types [29, 30].

Discussion and future scope

We have observed that due to its openness android platforms are prone to maximum cyber threats (android malwares), also it is being observed that with passing time android users are increasing so as their chances of becoming victim is also growing and hence we are left with huge data which could be used to analyse the behaviour of various threats so that we can improve the performance of android applications. Hence we need heuristic detection technique so that future could be saved, all we are left with huge data generated every seconds in the population of billions. In this paper we have discussed about numerous detection and solution techniques which includes Static and Dynamic approaches of detection and solutions. The most appropriate way of understanding malwares is by studying the working of an already existing malware. The next step is to try and alter a normally functioning application and expertise it to behave as a malware. Other than machine learning classifier various techniques of data mining and artificial intelligence can be used to differentiatebetween gentle and malicious applications. Hence advancement in the field of artificial intelligence and machine learning must be made to make a perfect cyber threat detector which can learn about latest android malware by (learning algorithms).

Conclusion

This paper is all about android malwares and their detection techniques, we have pricesly discussed about various types of malware available and static and dynamic approach of their detection. Due to its open nature the range of the Android malicious software is far greater than other platforms. These malicious applicationslurk the user data privacy, money and device integrity, and are difficult to detect since they deceptively behave as genuine apps bringing no harm, so it is much needed to improve security problems of android platform for security of cyber world. These detection techniques aims to increase integrity, confidentiality, and availability of the system by improving the isolation of data, protecting access to data, justifying the threat of service interruption to data or application. At last since,the advancement in mobile technology is rapidly increasing it is necessary thatother technical and conceptual solutions must be formulated which are more resourceful in thwarting the efforts of the malware developers to access private and sensitive data and exploit the susceptibilities of the mobile operating system.

REFERENCES

1. Kaspersky Lab and INTERPOL Survey Reports, “Mobile cyber threats.”
2. “CNCERT/CC.CNCERT/CC Annual Report”, (2013), http://www.cert.org.cn/publish/main/upload/File/2013 Annual Report.pdf.2014:53-56.
3. LoviDua*et al.* “REVIEW ON MOBILE THREATS AND DETECTION TECHNIQUES”, International Journal of Distributed and Parallel Systems (IJDPS) Vol.5, No.4, July 2014.
4. Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. (2012, Febuary) Hey, you, get off of my market:detecting malicious apps in official and alternative android markets. [Online]. Available: http: //www.internetsociety.org/sites/default/files/07 5.pdf.
5. W. Zhou, Y. Zhou, X. Jiang, and P. Ning, “Detecting repackaged smartphone applications in third-party android marketplaces,” in Proceedings of the second ACM conference on Data and Application Security and Privacy, ser. CODASPY ’12. New York, NY, USA: ACM, 2012, pp. 317–326. [Online]. Available: http://doi.acm.org/10. 1145/2133601.2133640.
6. S.-H. Seo, K. Yim, and I. You, “Mobile malware threats and defenses for homeland security,” in Multidisciplinary

Research and Practice for Information Systems, ser. Lecture Notes in Computer Science, G. Quirchmayr, J. Basl, I. You, L. Xu, and E. Weippl, Eds. Springer Berlin Heidelberg, 2012, vol. 7465, pp. 516–524. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32498-7 39.

7. C. Zheng, S. Zhu, S. Dai, G. Gu, X. Gong, X. Han, and W. Zou, "Smartdroid: an automatic system for revealing ui-based trigger conditions in android applications," in Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices, ser. SPSM '12. New York, NY, USA: ACM, 2012, pp. 93–104. [Online]. Available: http://doi.acm.org/10.1145/2381934.2381950.
8. M. Szydlowski, M. Egele, C. Kruegel, and G. Vigna, "Challenges for Dynamic Analysis of iOS Applications," in Proceedings of the Workshop on Open Research Problems in Network Security (iNetSec), Luzerne, Switzerland, June 2011.
9. C. Wysopal, L. Nelson, D. D. Zovi, and E. Dustin, The art of software security testing:identifying software security flaws. Symantec Press, 2006.
10. D. Amalfitano, A. R. Fasolino, and P. Tramontana, "A gui crawling-based technique for android mobile application testing," pp. 252–261, 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/ epic03/wrapper.htm?arnumber=5954416.
11. Fan Yuhui*et al.* "The Analysis of Android Malware Behaviors",International Journal of Security and Its Applications Vol. 9, No. 3 (2015), pp. 335-346 http://dx.doi.org/10.14257/ijsia.2015.9.3.26 .
12. Mustafa Hassan Saad, "Android Spyware Disease and Medication", 2015 Second International Conference on Information Security and Cyber Forensics (InfoSec)**IEEE *Xplore*:** 21 March 2016.
13. HimanshuShewale *et al.* "ANALYSIS OF ANDROID VULNERABILITIES AND MODERN EXPLOITATION TECHNIQUES",ICTACT JOURNAL ON COMMUNICATION TECHNOLOGY, MARCH 2014, VOLUME: 05, ISSUE: 01ISSN: 2229-6948(ONLINE) DOI: 10.21917/ijct.2014.0122.
14. Android Architecture, Available at: http://www.androidapp-market.com/android-architecture.html.
15. Ruben Jonathan Garcia Vargas, Ramon Galeana Huerta, Eleazar Aguirre Anaya and Alba Felix Moreno Hernandez, "Security Controls for Android", Proceedings of Fourth International Conference on Computational Aspects of Social Networks, pp: 212-216, 2012.
16. Dinesh, Aswathy, and Ming Chow. "AN ANALYSIS OF MOBILE MALWARE AND DETECTION TECHNIQUES".
17. Saba Arshad*et al.* "Android Malware Detection & Protection: A Survey",(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 2, 2016.
18. A. Aiken, "Apposcopy : Semantics-Based Detection of Android Malware Through Static Analysis," Fse 2014, pp. 576–587, 2014.
19. "Android.Fakedefender.B | Symantec." [Online]. Available: https://www.symantec.com/security_response/writeup.jsp?docid=2013091013-3953-99. [Accessed: 15-Dec-2015].
20. M. Zheng, M. Sun, and J. C. S. Lui, "DroidAnalytics : A Signature Based Analytic System to Collect , Extract , Analyze and Associate Android Malware," 2013.
21. M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," ACM Comput. Surv., vol. 44, no. 2, pp. 1–42, 2012.
22. I. You and K. Yim, "Malware obfuscation techniques: A brief survey," Proc. - 2010 Int. Conf. Broadband, Wirel. Comput. Commun. Appl. BWCCA 2010, pp. 297–300, 2010
23. I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: BehaviorBased Malware Detection System for Android," Proc. 1st ACM Work. Secur. Priv. smartphones Mob. devices - SPSM '11, p. 15, 2011.
24. "strace download | SourceForge.net." [Online]. Available: http://sourceforge.net/projects/strace/. [Accessed: 22-Dec-2015].
25. A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, "„Andromaly": a behavioral malware detection framework for android devices," J. Intell. Inf. Syst., vol. 38, no. 1, pp. 161–190, 2012. [
26. M. Zhao, F. Ge, T. Zhang, and Z. Yuan, "AntiMalDroid: An efficient SVM-based malware detection framework for android," Commun. Comput. Inf. Sci., vol. 243 CCIS, pp. 158–166, 2011.
27. W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," Osdi '10, vol. 49, pp. 1– 6, 2010.
28. L. Yan and H. Yin, "Droidscope: seamlessly reconstructing the os and dalvik semantic views for dynamic android malware analysis," Proc. 21st USENIX Secur. Symp., p. 29, 2012.
29. F. Wu, H. Narang, and D. Clarke, "An Overview of Mobile Malware and Solutions," J. Comput. Commun., vol. 2, no. 2, pp. 8–17, 2014.
30. T. Bläsing, L. Batyuk, A. D. Schmidt, S. A. Camtepe, and S. Albayrak, "An android application sandbox system for suspicious software detection," Proc. 5th IEEE Int. Conf. Malicious Unwanted Software, Malware 2010, pp. 55–62, 2010.