A Behavior based Malware Detection Approach for Android Smartphone

Dr. P. C. Senthil Mahesh[1], K. Muthumanickam[2]

[1]Professor, [2]Associate Professor

[1, 2,] Department of Computer Science and Engineering

[1]Annamacharya Institute of Technology and Sciences, Rajampet, Andra Pradesh, India

[2]Arunai Engineering College, Tiruvannamalai, Tamilnadu, India

Email: [1]pcsenthimahesh@gmail.com, [2]kmuthoo@yahoo.com

Abstract

Today, ubiquitous devices set two important sides for the business. Although one side brings lot of potential in terms of reaching end-users, the other side mounts up huge number of malware. Particularly, Android mobile phones now symbolize an idyllic option for malware developers. This paper proposes a behavior based approach for detecting and preventing malicious activities of the processes inside the applications to be installed in Android devices. In order to appraise our system, we have taken 20 legitimate applications and 20 applications are malicious samples. The manual analysis and practical results show that our system is more efficient in malware detection and interruption in Android smart phones with high detection rate with minimal false positive.

Keywords: Android, Behavior, Mobile Malware, Security.

1. Introduction

Today, mobile phone technology is turning out to be the fast growing technology which becomes an essential communication tool for personal and business growth. Different mobile phones use different Operating System (OS). Android is the most popular OS and very efficient when compared to other OSes. According to [1], the world wide mobile phone sale to end-users has achieved over 1.5 billion at the end of 4Q 2017. Android OS is based on embedded Linux which is open source software available for free. Android platform provides easy way for programming interface. Malware writers not only targeting traditional computers, but they can also ensure that their creations run in mobile phone as well. McAfee Labs mobile threat report has found a 16 million mobile malware in the 3Q of 2017[2]. Thus, it becomes a major issue for preventing Android platforms from malware attacks. Behavior monitoring and interception techniques are widely used in security tools for desktop computers. So far, some achievement related to the behavior monitoring and interceptions in Android platform have been proposed. All of them are based on monitoring and intercepting system calls in kernel level. But monitoring malware activities at the kernel level is less encouraged. The most important reasons are listed as follows. First, it is not efficient. From user-mode we can enter into kernel-mode through system call interface provided by the OS.

JAVA Application Programming Interface (API) to be invoked by the Android application will resolve into many system calls and monitoring and intercepting system calls in kernel level will affect all other running processes. Secondly, it is not applicable for real Android devices. Because, device cannot use loadable kernel module and developer cannot run the code without recompiling the kernel. All these reasons make monitoring and intercepting system calls in kernel level more difficult.

In this paper, we proposed a method to detect malicious activities in Android mobile phone by monitoring the behavior of applications to be installed. As our system does not rely on malware signature, it can able to detect unknown malware.

2. Related Works

The two basic approaches used to detect mobile vulnerabilities are signature-based and behavior-based approach. The former approach can be easily mitigated by obfuscation technique. For example, Android SDK [3] has integrated with ProGuard which is a framework for code obfuscation. The latter approach [4-5] monitor software program installation to determine whether the execution exhibits malicious behavior. The malicious activity of an executable can be effectively identified by crump the data movement intervening system calls [6-7]. Today, Antivirus software developers have been dealt with thousands of potentially dangerous malicious samples. Gathering and analyzing such huge number of new malware samples everyday becomes part of daily work which is mainly need to identify unique signature of them. M.Egele et al. [8] presented a survey on dynamic malware analysis techniques which are mainly concerned to discover possible malicious software. Additionally, techniques that require human assistance or intervention to identify unknown malicious behavior were also discussed. Today almost all mobile phone users use web-enabled Android OS to download and install third party services without running any security measures. This actually poses a serious threat to mobile phone users. However, the existing protections available for Android OS permit a user to access an application without any restriction. Article [9] presented Apex, a policy enforcement framework for Android which allows a user to selectively grant permissions for certain applications and also manipulate constraints based on the usage of resources. They also described an extended package installer which permits the user to setup constraints through an easy-to-use interface. But the user does not have a choice to protect the privacy of their location if they wish to use the application for which the exact location is offered by the application itself.

A single Android mobile vulnerability can attack and compromise many devices at once which then become a botnet. This botnet can later used for many illegal activities such as stealing sensitive information, sending premium SMS, and sending spasm. Today, there is no effective mechanism available to completely prevent mobile vulnerabilities. [10] proposed a method to detect Android malware by analyzing manifest files. However this approach suffers from two limitations. Initially, the unique signature of a malware is obtained using Signature-based method. Although this methods is effective against known malware, it is inadequate for detecting unknown malware. Finally,IP-address-based blacklists. This method cannot detect unknown malware, because the blacklists are generated from known malicious activities. Malware writers target mobile platform and applications (apps). Basically malware writers deliver malicious software either as a fake application or part of legitimate apps which makes detection more difficult. In [11], the authors have developed a system called, RiskRanker, a proactive method to discover Android malware which reside as part of untrusted or third-party apps. RiskRanker discovers a malware if any apps exhibits malicious activity.

Schmidt et al, [12] presented a technique based on evaluating the security of Android smart phones with a focus on Linux part. The results are not limited to Android but also applicable to Linux based smart phones such as Open Moko Neo Free Runner. The proposed method checks security functionalities by analyzing Android framework and Linux-kernel.

However the use of ClamAV relevant files approximate size of 28 MB exceeds the system image of size 21 MB. In [13], a behavior-based system named, CROWDROID, was proposed to hijack system calls to collect information of event generated by Android applications and then creates an output file. Then the output file was uploaded to remote server for further analysis to detect malware. Their framework has been demonstrated by analyzing the data collected in the central server using two different types of inexplicably created malware data-sets for test purpose and those from real malware found in the wild. The main drawback of their system is the use of an application that simulates user interaction which will never be as a real user.

Article [14] evaluated the stat-of-art commercial mobile Anti-malware products for Android and verify how they react against common obfuscation techniques. And also developed DroidChameleon, a systematic framework with various transformation techniques. Their implementation results show that the tool is resistant against common malware techniques. The authors also suggested some ideas for improving the current state of mobile malware detection techniques. Mobile devices have become popular in our lives since they offer almost the same functionality similar to personal computers. However, Android-based mobile devices had appeared lately and they were now the timely target for malware writers. Android-based smart phone users can download free apps from the Internet. But, the downloadable apps were not certified apps so that they can contain malicious executables. Aung et al., [15] proposed a framework which was based on machine learning approach for detecting malware apps in Android smart phone.

3. Proposed Approach

Though many techniques have been proposed in the past, this scourge of malware persists despite the existence of various types of security software, Anti-Virus (AV) software being the best example. Although AV software decreases the threat of malware, it has some limitations. However, AV software typically uses static characteristics of different malwares such as distrustful format of instructions in the binary to detect threats. Unfortunately, it is quite easy for advanced malware writers to create different variants of the same that are functionally equivalent, both automatically and manually, thus overwhelm static analysis easily. The proposed work can be used in security protection tools to employ a dynamic defense solution. This method exercised by security tools to detect and identify unknown malware by updating malware database. With the rapid growth of Android malware, the system and techniques that it uses will have an extremely great prospect of application. This approach potentially solves both problems. First, by executing AV protection second, we posit that dynamic analysis makes detection of new, undiscovered malware variants easier. It automatically updates new malware to the database and provides protection from newer malwares with increased in accuracy.

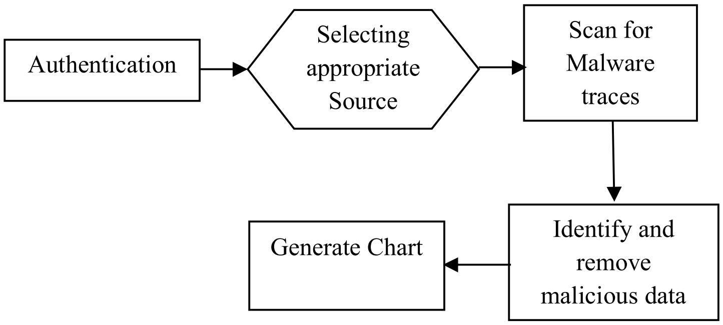
Authentication
Selecting appropriate Source
Scan for Malware traces
Identify and remove malicious data
Generate Chart

Figure 1. Architecture of the proposed model

Implementation is the stage of the project where the execution of a plan is done. Therefore it is considered as the most critical stage in achieving a successful new proposed system and satisfying the user, giving confidence that the new system will work effectively.

3.1. Authentication

It is the first module in the application in which a user has to register for getting access through the application. The application demands the user to fill the User Name, Email Id, Password and Confirmation Password fields. After pressing the submit option in registration form the user can enter with the registered details to pass through next step in the application.

3.2. Selecting appropriate source

After a successful login, the user will be shown a list with various sources containing check box. Now, the user can select any one of the source from the list and choose confirm selection option from the option menu that will be displayed when a option button is clicked in the smartphone.

3.3. Scanning for malicious data

When the confirm option is pressed, the application will start scanning the selected source. After a few second, a form will be listed with a list of malicious data available in that source

3.4. Selecting appropriate source

In this module, the detected malicious data is removed from the source. At the same time, the effect of the malicious data in the source is recorded and its percentage has been recorded for the chart generation.

3.5. Selecting appropriate source

After the malicious data list is displayed, now choose generate chart from the option menu. Now, the malware pie chart is generated which describes both malware attack zone and safe zone. The safe zone will be indicated by green color and malware attack zone will be indicated in different colors according to the type of malware present in the source. The percentage will describe the source condition in the present time.

4. Experimental result

We have run our system in Pentium IV processor, with memory size of 1 GB and Mobile (FROYO) device which runs Android OS 2.2 with Android SDK 3.5. The system can also run through emulator. Additionally, the computer runs Windows XP OS. We have downloaded and installed 20 legitimate applications form official websites as normal apps, and 20 malicious samples were downloaded from ContagioMobile [16]. The simulated results are given in Table 1.

Table 1. Test results

Type	Malware Samples	Detected	TPR (%)	FPR (%)
Malicious	20	20	100	0
Benign	20	1	0	3.1

After testing the only forged positive application, our system has reported that it was a Antivirus tool (Lookout). As false positive rate is predictable, we can eliminate it by adding more trusted applications.

9. Conclusion

By the experiment result, we have found that this project provides better performance than the existing systems. The existing system provides protection from the malwares that are present in the malware database. It needs to be updated regularly. Updation is possible only when the developer updates and releases the new version. Thus it’s not possible to detect new malwares. To overcome this, we are providing a dynamic defence mechanism to detect a newly originating malwares. After detecting the new malware the application will compare it with the database record. If the information about the new malware is not present then the information about that malwares is recorded in the database automatically for the future detection. Thus, the detection of newer malware detection will be at higher accuracy, which leads to high performance of device and at the same time providing security for the user data. The sources in the smart phones are very much vulnerable to attacks. Thus providing security for those sources is must to protect from mobile malware attacks. Providing a dynamic defense will secure our device from malwares and increase the efficiency of the device.

References:

[1] Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017. Available. https://www.gartner.com/newsroom/id/3859963.

[2] Kaspersky Security Bulletin: Threat Predictions for 2018.

[3] Android SDK.http://developer.android.com/sdk/index.html

[4] M.Christodorescu, S.Jha, and C.Kruegel, “Mining Specifications of malicious behavior”. Proceedings of the First India software Engineering Conference, Dubrovnik, Croatia **(2008)** September 5-14.

[5] M.Fredrikson, S.Jha, and M.Christodorescu, R.Sailer, and X.Yan, “Synthesizing near-optimal malware specifications from suspicious behavior”, Proceedings of the 8th International Conference on Malicious and Unwanted Software: "The Americas", Fajardo, PR, USA (MALWARE), (**2014**) January 45-60.

[6] C.Willems, T.Holz, and F.Freiling, “Toward automated dynamic malware analysis using CWSandbox”, IEEE Security & Privacy, vol. 5, no. 2, (**2007**), pp. 32- 39.

[7] U.Bayer, I.Habibi, D.Balzarotti, E.Kirda, and C.Kruegel, “A view on current malware behaviors”, Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more, Boston, MA, (**2009**), pp. 8-8.

[8] M.Egele, T.Scholte, E.Kirda,C.Kruegel, “A survey on automated dynamic malware-analysis techniques and tools”, ACM Computing Surveys (CSUR), vol. 44, no. 2, (**2012**), Article. 6.

[9] M.Nauman,S.Khan,X.Zhang, “Apex: extending Android permission model and enforcement with user-defined runtime constraints”, Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS '10), Beijing, China, (**2010**) April 328-332.

[10] Ryo Sato, Daiki Chiba and Shigeki Goto, “Detecting Android Malware by Analyzing Manifest Files”, Proceedings of the Asia-Pacific Advanced Network, vol.36, (**2013**), pp. 23-31.

[11] M.Grace,Y.Zhou,Q.Zhang,S.Zou and X.Jiang, “RiskRanker: scalable and accurate zero-day android malware detection”, Proceedings of the 10th international conference on Mobile systems, applications, and services(MobiSys '12), Low Wood Bay, Lake District, UK , (**2012**) June 281-294.

[12] A.schmidt,H.Schmidt,J.Clausen,K.Yüksel,O.Kiraz,S.Camtepe,S.Albayrak, “Enhancing Security of Linux-based Android Devices”, Proceedings of the 15th International Linux Kongress, Berlin, Germany.

[13] I.Burguera,U.Zurutuza,S.Nadjm-Tehrani, “Crowdroid: behavior-based malware detection system for Android”, Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices(SPSM '11), Chicago, Illinois, USA, (**2011**) October 15-26.

[14] V.Rastogi,Y.Chen,X.Jiang,“DroidChameleon: evaluating Android anti-malware against transformation attacks”, Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security(ASIA CCS '13), Hangzhou, China , (**2013**) May 329-334.

[15] Z.Aung, W.Zaw, “Permission-Based Android Malware Detection”, International Journal of Scientific & Technology Research, vol. 2, no. 3, (**2013**), pp. 228-234.

[16] ContagioMobile Blog. Available at http://contagiominidump.blogspot.com