International Journal of Pure and Applied Mathematics
Volume 116 No. 9 2017, 237-241
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

# ANDROID APPLICATION FOR DETECTING MALWARES IN SMARTPHONE APPS

K.P.Kaliyamurthie
Dean, Department of CSE,
Bharath University,Chennai-73.
kpkaliyamurthie@gmail.com

**Abstract:** Malware authors use stealth techniques dynamic execution, code obfuscation methods, rep to bypass the existing protection mechanisms provided by the Android platform and commercial anti-malware repackaging and encryption. Multiple complementary approaches can be used in tandem for effective malware detection. Android security, analysis, and malware detection techniques. Conventional signature-based and static analysis methods are vulnerable in proposed system.

**Keywords:** Malware, Android, Smartphone, Security

## 1. Introduction

With the escalating growth of communication and information systems, a new term and acronym invaded the digital world called as malware. It is a general term, which stands for malicious software and has many shapes (codes, scripts, active content and others). It has been designed to achieve some targets such as, collecting sensitive data, accessing private computer systems, even sometimes harming the systems. The malware can reach the systems in different ways and through multiple media; the most common way is the downloading process from the internet, once the malware finds its way to the systems, based on the functions of the malware the drama will begin[1,2,5]. In some cases, the malware will not totally harm the system, instead affect the performance and creates overload process; in case of spying, the malware hides itself in the system, which cannot be detected by the anti-virus software, these hidden malware send critical information about the computer to the source. Based on the above challenges[3,4], it is critical to carry out an in-depth analysis to understand the malware for better. Detection and removal chance. This paper is organized as follows: Section two has covered the recent state of the malware security and threats through results obtained from different journals. Section three discusses about the types of malware, section four presents the malware analysis techniques. Section five studies the propagation of malware in different applications and environment, and finally section six explains malware detection techniques. Many studies, surveys, experiments, brainstorming, statistical analysis and modeling methods have been done to gain deeper knowledge and valuable information about malware[6,7] , because the attackers are continually developing their abilities, attacking skills and techniques. In order to make the tracking and detection processes difficult, and to pose new challenges to inspectors, all these studies and works are not sufficient enough to cover the rapid increase in malware evolution. Based on our understanding Virus Bulletin (1988) was the first dedicated Journal to study the malware, while, now there are a lot of Journals available that are dedicated to the security issues, especially malware issues. This paper has been presented to gain understanding about the various issues related to malware. We have used much recourse to form different papers and journals, the details of the recourses that we used, will be shown in data collection part in more details [8,9].

## 2. Related work

Understanding Android Security William Enck, Machigar On tang, and Patrick McDaniel Pennsylvania State University, 2009 An enhanced installer and security framework to answer a variant of these questions—namely, “does an application break some larger phone-wide security policy?” Our tool, called Kirin,7 extracts an application’s security policy from its manifest file to determine if the requested permissions and component permission. Assignments are consistent with the stakeholders’ definition of a secure phone (stakeholders in this context range from the network provider to an enterprise to a user). Kirin uses a formalized model of the policy mechanisms described in this article to generate automated proofs of compliance using a Prolog engine running on the phone. If an application’s policy isn’t compliant [11,12], it won’t be installed. By defining security requirements in logic, which we call policy invariants, we significantly reduce the need to defer install-time decisions to the user—that is, the policy invariants capture the appropriate response.

We've successfully used Kirin to identify multiple vulnerabilities in the base applications Provided with Android and have subsequently established an ongoing relationship with Google to fix the flaws and further investigate Android's security via Kirin.

Android Application Sandbox System for Suspicious Software Detection A sandbox created for analysing Android applications applicable as cloud service. Therefore, we showed how the Android emulator can be used to run Android applications in an isolated environment. Unlike other sand boxes[13,14,15], we added a pre-check functionality that can analyse Android executables in a static manner. This can indicate usage of malicious patterns within source code. In the dynamic analysis, system calls can be traced and corresponding reports are logged. These can be used for further investigations, either performed manually or automatically.

Various Approaches in Analyzing Android Applications with its Permission-Based Security Models Ittipon Rassameeroj and Yuzuru Tanahashi, 2011.

A high-level contextual analysis based on the network visualizations, we were able to verify several aspects of permission-based security models of APKs that had been observed and addressed in past research. Also, by taking the relative frequency of the permission requests into consideration, we have discovered that APKs that belong to certain categories, such as Communication category, have more chance of causing a false positive detection when in a search of malwares[16,17]. This suggests that a deterministic analysis based on permission requests fails to provide users with credible alerts. Finally, by focusing on the similarity of the APKs' permission requests, we were able to distinguish some APKs from others based on their fundamental functionality, which had also demonstrated as a potential technique for detecting malware in a nondeterministic approach. Although in most cases the distinction between APK[20] clusters were not clear, we consider with a hierarchical refinement, or applying weight on permissions based on its frequency (i.e., common permissions and rare permissions may have more influence to the APK's functionality than other permissions), the clustering may result in a better classification of services.

Dissecting Android Malware: Characterization and Evolution Yajin Zhou Department of Computer Science North Carolina State University yajin zhou@ncsu.edu. Xuxian Jiang Department of Computer Science North Carolina State University jiang@cs.ncsu.edu,2012.

A systematic characterization of existing Android malware. The characterization is made possible with our more than one-year effort in collecting 1260. Android malware samples in 49 different families, which covers the majority of existing Android malware, ranging from its debut in August 2010 to recent ones in October 2011. By characterizing these malware samples from various aspects, our results show that (1) 86.0% of them repackage legitimate apps to include malicious payloads; (2) 36.7% contain platform-level exploits to escalate privilege; (3) 93.0% exhibit the bot-like capability. A further in depth evolution analysis of representative Android malware shows the rapid development and increased sophistication, posing significant challenges for their detection [18,19].

A Signature Based Analytic System to Collect, Extract, Analyze and Associate Android Malware Min Zheng, Mingshen Sun, John C.S. Lui Computer Science & Engineering Department The Chinese University of Hong Kong,2013 An Android malware analytic system which can automatically collect malware, generate signatures for applications, identify malicious code segment (even at the op code level), and at the same time, associate the malware under study with various malware and applications in the database. Our signature methodology provides significant advantages over traditional cryptographic hash like MD5-based signature. We show how to use Droid Analytics to quickly retrieve, associate and reveal malicious logics. Using the permission recursion technique and class association, we show how to retrieve the permissions of methods, classes and application (rather than basic package information), and associate all applications in the op code level. Using Droid- Analytics[25,26], one can easily discover repackaged applications via the similarity score. Last but not least, we have used Droid Analytics to detect 2,475 malware samples from 102 families, with 327 zero-day malware samples from six different families. We have conducted extensive experiments to demonstrate the analytic and malware detection capabilities of Droid Analytics.

## 3. Proposed methodology

Conventional signature-based and static analysis methods are vulnerable in proposed system. A hybrid Android malware analysis and detection framework. It is used to detect unknown malware [21,22]. Signature based on detection is efficiency and simplicity. It extracts the interesting syntactic or semantic patterns, features and create a unique signature matching that particular malware. Signature-based methods fail against the unseen variants of already existing and known malware.

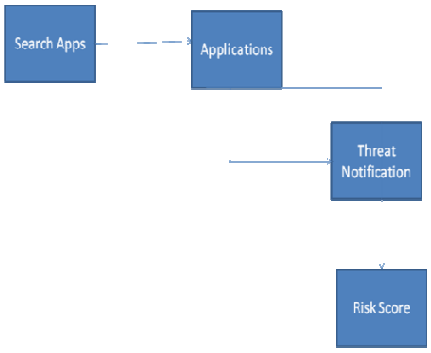


**Figure 1.** Proposed Architecture

The application allows the user to search the installed apps within the application and find out what is the application they are in need to use. This feature is common and it is being done to get all the application which is allowed to find out the risk score. The searched applications will display the amount of risk score which contained in it and will display a threat message. This feature helps the user to find the applications which are in risk and will help to user to uninstall all the applications which are in risk state. We propose a solution that leverages a method to assign a risk score to each app and display a summary of that information to users. Results from four experiments are reported in which we examine the effects of introducing summary risk information and how best to convey such information to a user. Our results show that the inclusion of risk-score information has significant positive effects in the selection process and can also lead to more curiosity about security-related information [23,24].

## 4. Experimental results

It is a technique used by software developers and writers targeting to hide the details of their products so that the reverse engineers can't find the correct code, it has been used as an advantage by the malware writers to achieve the same goal, obfuscation can be achieved by different operations and easily can make changes in the signature of malware in order to make the process of detecting the malware very difficult. The obfuscation techniques can be done in different methods, starting from inserting some (no operation) instructions and inserting (pushpop) x, which known as dead-code because nothing will be achieved and accomplished and inserting some instructions for branching unconditionally, moving to inserting process for some registers and substituting instructions, all of these methods will guide to obfuscate the code of the malware and make the process of detection difficult to malware scanners. Malware normalization can be identified as a process and mechanism to detect the obfuscated copies of malware and increasing the rate of catching the malware by the detector, the output of the normalization will be the original signature of the malware which has been obfuscated and accordingly the signature will be compared to the signatures to verify it, then it will be saved in the list of known signatures in order to decrease the time of scanning and detecting next times.

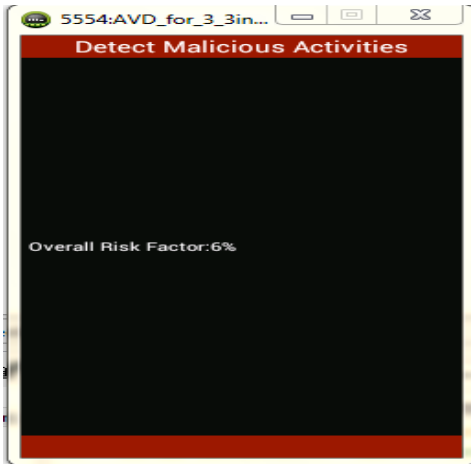


**Figure 2.** Malicious Activities

## 5. Conclusion

The problem of detecting targeted malware via behavioral analysis requires the ability to reproduce an appropriate set of conditions that will trigger the malicious behavior. Determining those triggering conditions by exhaustively searching through all possible states is a hard problem. In this paper, we have proposed a novel system for mining the behavior of apps in different user-specific contexts and usage scenarios. The objectives of this simulation are to avoid systems threats before being infected by real malware.

## References

[1] Y. Wang, K. Streff, and S. Raman, "Smartphone security challenges," IEEE Computer, vol. 45, no. 12, pp. 52–58, 2012.
[2] L. Cai and H. Chen, "Touchlogger: inferring keystrokes on touch screen from smartphone motion," in Proc. USENIX, ser. HotSec'11, Berkeley, CA, USA, 2011, pp. 9–9.
[3] E. Fernandes, B. Crispo, and M. Conti, "Fm 99.9, radio virus: Exploiting fm radio broadcasts for malware deployment," IEEE TIFS, 2013.
[4] T. Vidas and N. Christin, "Sweetening android lemon markets: Measuring and combating malware in application marketplaces," in Proc. ACM, ser. CODASPY '13. ACM, 2013, pp. 197–208.

[5] J. Oberheide and C. Miller, "Dissecting the android bouncer," SummerCon2012, New York, 2012.
[6] G. Suarez-Tangil, J. E. Tapiador, P. Peris, and A. Ribagorda, "Evolution, detection and analysis of malware for smart devices," IEEE Comms. Surveys & Tut., vol. 16, no. 2, pp. 961–987, May 2014.
[7] M. Rangwala, P. Zhang, X. Zou, and F. Li, "A taxonomy of privilege escalation attacks in android applications," Int. J. Secur. Netw., vol. 9, no. 1, pp. 40–55, Feb. 2014.
[8] S. Chakradeo, B. Reaves, P. Traynor, and W. Enck, "Mast: Triage for market-scale mobile malware analysis," in Proc. ACM, ser. WiSec '13. New York, NY, USA: ACM, 2013, pp. 13–24.
[9] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "Riskranker: scalable and accurate zero-day Android malware detection," in Proc., ser. MobiSys '12. ACM, 2012, pp. 281–294.
[10] Y. Zhou and X. Jiang, "Dissecting Android malware: Characterization and evolution," in Proc. IEEE, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 95–109.
[11]Udayakumar R., Kaliyamurthie K.P., Khanaa, Thooyamani K.P., Data mining a boon: Predictive system for university topper women in academia, World Applied Sciences Journal, v-29, i-14, pp-86-90, 2014.
[12]Kaliyamurthie K.P., Parameswari D., Udayakumar R., QOS aware privacy preserving location monitoring in wireless sensor network, Indian Journal of Science and Technology, v-6, i-SUPPL5, pp-4648-4652, 2013.
[13]Brintha Rajakumari S., Nalini C., An efficient cost model for data storage with horizontal layout in the cloud, Indian Journal of Science and Technology, v-7, i-, pp-45-46, 2014.
[14]Brintha Rajakumari S., Nalini C., An efficient data mining dataset preparation using aggregation in relational database, Indian Journal of Science and Technology, v-7, i-, pp-44-46, 2014.
[15]Khanna V., Mohanta K., Saravanan T., Recovery of link quality degradation in wireless mesh networks, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4837-4843, 2013.
[16]Khanaa V., Thooyamani K.P., Udayakumar R., A secure and efficient authentication system for distributed wireless sensor network, World Applied Sciences Journal, v-29, i-14, pp-304-308, 2014.
[17]Udayakumar R., Khanaa V., Saravanan T., Saritha G., Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction, Middle - East Journal of Scientific Research, v-16, i-12, pp-1781-1785, 2013.
[18]Khanaa V., Mohanta K., Saravanan. T., Performance analysis of FTTH using GEPON in direct and external modulation, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4848-4852, 2013.
[19]Kaliyamurthie K.P., Udayakumar R., Parameswari D., Mugunthan S.N., Highly secured online voting system over network, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4831-4836, 2013.
[20]Thooyamani K.P., Khanaa V., Udayakumar R., Efficiently measuring denial of service attacks using appropriate metrics, Middle - East Journal of Scientific Research, v-20, i-12, pp-2464-2470, 2014.
[21]R.Kalaiprasath, R.Elankavi, Dr.R.Udayakumar, Cloud Information Accountability (Cia) Framework Ensuring Accountability Of Data In Cloud And Security In End To End Process In Cloud Terminology, International Journal Of Civil Engineering And Technology (Ijciet)Volume 8, Issue 4, Pp. 376–385, April 2017.
[22]R.Elankavi, R.Kalaiprasath, Dr.R.Udayakumar, A fast clustering algorithm for high-dimensional data, International Journal Of Civil Engineering And Technology (Ijciet), Volume 8, Issue 5, Pp. 1220–1227, May 2017.
[23]R. Kalaiprasath, R. Elankavi and Dr. R. Udayakumar. Cloud. Security and Compliance - A Semantic Approach in End to End Security, International Journal Of Mechanical Engineering And Technology (Ijmet), Volume 8, Issue 5, pp-987-994, May 2017.
[24]Thooyamani K.P., Khanaa V., Udayakumar R., Virtual instrumentation based process of agriculture by automation, Middle - East Journal of Scientific Research, v-20, i-12, pp-2604-2612, 2014.
[25]Udayakumar R., Thooyamani K.P., Khanaa, Random projection based data perturbation using geometric transformation, World Applied Sciences Journal, v-29, i-14, pp-19-24, 2014.
[26]Udayakumar R., Thooyamani K.P., Khanaa, Deploying site-to-site VPN connectivity: MPLS Vs