International Journal of Engineering Research and General Science Volume 4, Issue 1, January-February, 2016
ISSN 2091-2730

Malware Detection System for Android Mobile Applications

Mr. Akash J. Wadate, ME 2nd year, CSE department, G.H.Raisoni COE&M, Amravati, Maharashtra, India,

akashwadate007@gmail.com

Prof. N. R. Chopde,Assistant Professor,CSE Dept.,G.H .Raisoni COE&M,Amravati,Maharashtra,India

Nitin.chopde@raisoni.net

Prof.D.R.Datar, Assistant Professor,CSE Dept.,G.H. Raisoni COE&M,Amravati,Maharashtra,India

Dinesh.datar@raisoni.net

Abstract- With day to day increase in the number of mobile applications there is an analogous increment in the mobile threats. For such kinds of threats to mobile devices there should be some security mechanism to be implemented. In this proposed system in order to improve the security of the mobile apps one methodology is proposed which will evaluate the mobile applications security based on the cloud computing platform and data mining. Here also a prototype system named Malware detection system is presented to identify the mobile app's virulence or benignancy. Compared with traditional method, such as permission pattern based method, Malware combines the dynamic and static analysis methods to comprehensively evaluate an Android app. In the implementation, Android Security Evaluation Framework (ASEF) and Static Android Analysis Framework (SAAF) are adopted , the two representative dynamic and static analysis methods to evaluate the Android apps and estimate the total time needed to evaluate all the apps stored in one mobile app market.

As mobile app market serves as the main line of defense against mobile malwares, the evaluation results show that it is practical to use cloud computing platform and data mining to verify all stored apps routinely to filter out malware apps from mobile app markets.

Keywords- Android app, mobile malware detection, cloud computing, data mining.

1. INTRODUCTION

MOBILE THREATS

These years witness an explosive increase in mobile apps. According to Mary Meeker's report [1] on Mobile Internet trends, more and more PC client software's are migrating to the mobile device. According to Gartner's statistical prediction [1], the amount of total downloads of mobile apps in 2013 will be about 81 billion. Among these, there are about 800 000 Android apps in Google Play market, and the total download is about 48 billion as of May 2013[3]. In contrast with Apple AppStore, there are different sources for Android apps download, such as wandoujia, AppChina, Baidu mobile assistant, etc. While these markets give a good supply and bring more convenience for Android users, they will also bring mobile threats as different market places have different malware detection utilities and methods. Some sophisticated malwares can escape from detection and spread even via such Android markets.

For some years, Smartphone's are growing in popularity rapidly; more and more people and companies have the possibility to use these devices. Nowadays there are many different devices which can be considered Smartphone's, and there are also many different operating systems (OS). For instance, Apple's iOS, many different Smartphone's using Google Android as OS, Blackberry OS Smartphone's, Windows Mobile Phones, and some others[2]. These devices also have become much more powerful than traditional mobile phones. Having a powerful processor with dedicated graphic chip and several hundred MB RAM is not uncommon anymore (e.g. HTC Sensation or Apple Iphone 4). But security mechanisms on these devices are not as sophisticated as they are on traditional computer systems. Companies are eager to secure their computer systems with proper security mechanisms (anti-virus, firewall, up-to-date operating system, etc.), but these techniques are not yet utilized to their full potential on Smartphone's. One reason for this is that only few such software is available, but an even bigger obstacle is that many existing security mechanisms require severe changes to the Smartphone's operating system. Another reason why such software is not used widely is the limited battery capacity of the devices and the fact that available security applications like virus scanning consume a considerable amount of this capacity.

One big benefit of shifting the security functionality into the cloud is the almost indefinite processing power and “battery” capacity. This makes it possible to run very resource intense security services that would not be feasible on the phone. If the phone is replicated in the cloud, this also allows the developer of a security service to extend this service without changes on the phone. The security service can examine the phone not only from inside its system (similar to an application on the phone), but it can also monitor the replica itself which runs the cloud (e.g. look at the connections the replicated phone attempts to make). This can further improve the chances of finding malicious software and open up possibilities that would not be feasible on the device itself. For example, detecting a root kit could be impossible on the phone itself, but a security service which only scans the replica’s files without executing the replica, might be able to detect the root kit[2]. But the shifting of the security functionality into the cloud could also be problematic, if not all parts of the phone can be replicated into the cloud.

2. LITERATURE REVIEW

Security analysis of Android apps is a hot topic. More and more researchers use static analysis and dynamic behavior analysis, and even integrate it with machine learning techniques to identify malware.

Barrera et al.[3] made an analysis on permission based security models and its applications to Android through a novel methodology which applies Self- Organizing Map (SOM) algorithm preserving proximity relationships to present a simplified, relational view of a greatly complex dataset. The SOM algorithm provides a 2-dimensional visualization of the high dimensional data, and the analysis behind SOM can identify correlation between permissions. They discover insights on how the developers use the allowed permission model in developing and underlining the permission model’s strengths as well as its shortcomings through their methodology. Based on their results, they proposed some enhancements to the Android permission model.

Enck et al. [4] (TaintDroid) built a tool that warns users about applications that request blacklisted sets of permissions. They took both dangerous functionality and vulnerabilities into consideration and applied a wide range of analysis techniques. They designed and implemented a Dalvik decompiler, ded, which can recover application’s Java source code only using its installation image. Besides, they analyzed 21 million LOC retrieved from the top 1100 free applications in the Android market using automated tests and manual inspection. Their results show the wide misuse of privacy sensitive information, the evidence of telephone misuse, wide including of ad libraries in Android application, and the failing to securely use Android APIs of many developers.

Dai, Fei, Guo [5] has proposed a system in which a mobile malware behavior analysis method based on behavior classification and self-learning data mining is proposed to detect unknown or metamorphic mobile malware. The network behavior of mobile malware is analyzed according to the behavior characteristic and divided into different categories. An improved Naïve Bayesian anomalous network behavior analysis method based on behavior classification is proposed to detect the different types of network behavior of mobile malware. An incremental self-learning method is used to adjust the proposed behavior-classification based Naïve Bayesian Classifiers to adapt the variable network behavior of mobile malware.

Wenhui hu [6] has proposed a system in which they identified three library-centric threats in the real-world Android application markets: (i) the library modification threat, (ii) the masquerading threat and (iii) the aggressive library threat. He proposed Duet, a library integrity verification tool for Android applications at application stores. This is non-trivial because the Android application build process merges library code and application specific logic into a single binary file. Their approach uses reverse-engineering to achieve integrity verification. They implemented a full working prototype of Duet.

Asaf shabtai [7] proposed a system where they applied Machine Learning (ML) techniques on static features that are extracted from Android’s application files for the classification of the files. Features are extracted from Android’s Java byte-code (i.e., .dex files) and other file types such as XML-files. Their evaluation focused on classifying two types of Android applications: tools and games. Successful differentiation between games and tools is expected to provide positive indication about the ability of such methods to learn and model Android benign applications and potentially detect malware files. The results of an evaluation, performed using a test collection comprising 2,285 Android .apk files, indicate that features, extracted statically from .apk files, coupled with ML classification algorithms can provide good indication about the nature of an Android application without running the application, and may assist in detecting malicious applications. This method can be used for rapid examination of Android .apks and informing of suspicious applications.

Aubrey-Derrick Schmidt [8] contributed twofold. First, they performed static analysis on the executables to extract their function calls in Android environment using the command readelf. Function call lists are compared with malware executables for classifying them with PART, Prism and Nearest Neighbor Algorithms. Second, they presented a collaborative malware detection approach to extend these results. In their work, they employed collaboration for security approach to extend Malware detection results. Therefore, a set of entities is enabled to work on a common task without predefined roles in a hierarchical manner. The collaborative scheme is used to interact with other mobile devices in order to exchange detection data and system information. It can be considered as an operation mode whenever a mobile device is relying on the remote server but cannot access it.

Tianyang Li [9] proposed an offline phishing detection system named LARX (acronym for *L*arge-scale *A*nti-phishing by *R*etrospective data-e*X*ploration). LARX uses network traffic data archived at a vantage point and analyzes these data for phishing attacks. All of LARX’s phishing filtering operations use cloud computing platforms and work in parallel. As an offline solution for phishing attack detection, LARX can be effectively scaled up to analyze a large volume of trace data when enough computing power and storage capacity are provided. In this project, they proposed LARX, acronym for *L*arge-scale *A*nti-phishing by *R*etrospective data-e*X*ploration, an offline phishing attack forensics collection and analysis system. First, they used traffic archiving in a vantage point to collect network trace data. Secondly, they leveraged cloud computing technology to analyze the experimental data in a way similar to the"divide and conquer" scheme. They used two existing cloud platforms, Amazon Web Services and Eucalyptus. A physical server is also used for comparison. All of LARX's phishing filtering operations are based on a cloud computing platform and work in parallel. Finally, as an offline solution, they concluded that LARX can be effectively scaled up to analyze a large volume of network trace data for phishing attack detection.

3. PROPOSED WORK

It is a system to check whether an Android app is virulence or benignancy based on some customized tools in cloud platform. Proposed system is an automatize system which can be used to analyze Android apps. The main Objectives are as follows:

i. In this work, a methodology is proposed to evaluate the security of Android mobile apps based on cloud computing platform.

ii. In this work ASEF and SAAF are adopted, the two representative dynamic analysis method and static analysis methods, to evaluate the Android apps and estimate the total time needed to evaluate all the apps stored in a mobile app market.

4. CONCLUSION

This work strongly focuses on the issue of malware analysis of mobile application. By this work the user will be able to analyze and test the presence of malware in the apk before its installation. The user can also verify the apk after installation by using some tools. By this the user will be protected from malware attacks.

REFERENCES

1. Jianlin Xu, Yifan Yu, Zhen Chen_, Bin Cao, Wenyu Dong, Yu Guo, and Junwei Cao, MobSafe: Cloud Computing Based Forensic Analysis for Massive Mobile Applications Using Data Mining, TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 10/10, pp418-427 ,Volume 18, Number 4, August 2013.
2. Dennis Titze, 'Cloud based Security services for smartphones',Master's Thesis informatics, May 15, 2012, pp-1-85.
3. D. Barrera, H. G. Kayacik, P. C. van Oorschot, and A. Somayaji, "A methodology for empirical analysis of permission-based security models and its application to Android, in Proc.", 17th ACM Conference on Computer and Communications Security, Chicago, USA, 2010, pp. 73-84.
4. W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, "A study of android application security", in USENIX Security Symposium, San Francisco, USA, 2011,pp. 1-6.
5. Dai-Fei Guo1, Ai-Fen Sui1, Yi-Jie Shi, "Behavior Classification based Self-learning Mobile Malware Detection", JOURNAL OF COMPUTERS, VOL. 9, NO. 4, APRIL 2012, pp. 851-857.
6. Wenhui Hu, Damien Octeau, and Patrick McDaniel, "Duet: Library Integrity Verification for Android Applications", in Proc. 2nd ACM conference on Data and Application Security and Privacy, San Antonio , TX, USA, February, 2012, pp. 317-326.
7. Asaf Shabtai Yuval Fledel, Automated Static Code Analysis for Classifying Android Applications Using Machine Learning, IEEE 2010 International Conference on Computational Intelligence and Security, Nanning, China, December 2010, pp.321-332.
8. A. D. Schmidt, R. Bye, H. G. Schmidt, J. Clausen, O. Kiraz, K. A. Yuksel, S. A. Camtepe, and S. Albayrak, "Static analysis of executables for collaborative malware detection on Android in Communications", ICC'09, IEEE International Conference on, Dresden, Germany, 2009, pp. 1-7.
9. T. Li, F. Han, S. Ding, and Z. Chen, LARX: Large-scale Anti-phishing by Retrospective Data-Exploring Based on a Cloud Computing Platform, in Proc. 20th International Conference on. IEEE. Computer Communications and Networks (ICCCN), Maui, Hawaii, USA, 2011, pp. 1-5.