Detecting Malicious Applications from the cloud by using user feedback method

Prof. Chetan J. Shelke
Head,Dept of IT
P.R.Patil College of Engineering
Amravati,
Chetanshelke7@gmail.com

Dr. P. P. Karde
Head.,Dept of CSE
Government Polytechnic Amravati,
p_karde@rediffmail.com

Dr. V. M. Thakre
Head,Dept of CSE
Sant Gadge Baba Amravati University
Amravati,
vilthakre@yahoo.com

Abstract— As in recent period of computers and internets, mobiles devices, Smartphone's plays a vital role in human day to day activities. Also now a days Smartphone's & tablets are becoming very popular especially android based Smartphone's are gaining much more popularity as compared to Apple's iOS. These Smartphone's having lot of applications and features based on only internet but these new emerging features of these devices give opportunity to new malwares & threats. Android is comparatively new OS hence its makes very hard to detect and prevent these viruses and malwares attacks by using some basic traditional mechanisms. So security of these Smartphone's is now becoming very popular issue of researchers. The lack of standard security mechanism in Android applications is very useful to hackers. So to overcome these various pitfalls we use cloud services as a security weapon for providing decent security system for Android applications.

Keywords- *Android OS, Smartphone's, Malwares, Cloud Services, Applications Security.*

*****

I. INTRODUCTION

Recently the use of Smartphone's based on Android OS has increased rapidly hence providing better security policies is becoming most important area of research. As Smartphone's devices are being rapidly utilized by enterprises, and various government agencies also in military services, security plays an important role, because many users uses these devices to hold their valuable sensitive data, attackers may use this sensitive information with wrong intent. Mobile viruses can cause many types of damages like, private data leakage, remote listening etc. also they can congest the servers by sending many unwanted messages and spam's and reduces the efficiency of communication network. Hence in order to control these malware attacks in Smartphone's some crucial steps must be taken to provide some efficient mechanism for controlling the growth and productions of these viruses.

Anti-virus research is recently ongoing process for identifying and analyzing new and unknown malware for extracting possible detection scheme that can be used within some anti-virus software. There exits some virus and malware detector software that can scan and block viruses, Trojans that are infecting Android applications. Most malwares is being detected by scanning in signature database. For generating the reports and special signatures the infected application need to be analyzed and carefully observed so that we can collect some meaningful pattern about the specific malware. One approach to transfers the limited functionality of Smartphone's, is to off-load workload into the cloud. Taking advantage of the cloud is a very good approach, since a service in the cloud can be changed as needed, but modifications to the Smartphone's devices are very difficult. There are various applications like CloneDroid [1] which introduces the idea of offloading parts of programs into a cloud for speeding-up and saving power. Also ParanoidAndroid [2] offers a system in which the device is replicated into the cloud, and various security applications are applied on the replicas that had been created.

II. PROBLEM DEFINATION

One big benefit of shifting the security functionality into the mobile network 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 mobile network, 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 mobile network (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. But the shifting of the security functionality into the mobile network could also be problematic, if not all parts of the phone can be replicated into the mobile network. Previous work has shown that it is possible to run smartphone applications in the mobile network (e.g. [10]).

Smartphones typically possess only a limited amount of battery capacity and processing power. Once the security functionality is applied outside the physical device, these limitations can be circumvented.

Emulating the smartphone in the mobile network comes with one problem: every different smartphone needs its own emulator to be perfectly emulated. Since the phones typically differin certain aspects (e.g., operating system, available sensors, etc.), no emulator can simulate more than one smartphone perfectly. To support multiple devices, many different emulators need to be implemented. Another approach would be to implement a more generic emulator which can support more than one device. This is the case for Android devices where one emulator can represent many different devices, with the downside that these devices are not emulated accurately.

Since the purpose of the security system is the detection of malicious applications, such applications do not pose a security threat per se. A malicious application can only influence the security system if it tries to manipulate the system itself, for instance, by modifying the generated images or user interactions. To prevent this, these files must be stored in a secure way, and the integrity of the files and the security system's components on the smartphone and inside the mobile network have to be guaranteed. The integrity can be guaranteed by implementing cryptographic principles (e.g., by using file checksums, or by signing the components using public key cryptography [3,6]). If the integrity is not guaranteed, the security service's announcements become useless, since a malicious application can hide itself from detection (e.g., by not copying itself into the images), or trick the security service into reporting other applications as malicious, even if they are not. Since the components of the security service for different smartphone users do not influence each other, a compromised smartphone can not disrupt the whole security system.

At present there exists several systems for detecting and avoiding malwares and malwares available in android applications but apart from this there are some shortcomings presents in these system which are

1. Existing malware detection mechanism are very time consuming process as each time consuming process as each time a new application downloaded have to be checked and analyzed for behavior patterns and datasets are recorded and stored.
2. Extremely worst battery performance as Antimalware application and other processes has to be run in background.
3. There are several approaches of detecting malwares. Anomaly detection is one of them in which various behavior patterns are observed and stored in various datasets. The main drawback of this is that, one have to maintain large datasets and as no. of datasets increases then there also exists problem of inconsistency and redundancy.
4. Existing system Crowroid which detects Trojans like malwares on Android using analyzing no. of system calls each time is executed.

III. IMPLEMENTATION

According to the limitations and shortcoming stated in the literature survey it has been found that the in most of the work the security mechanism deploy on smartphone itself rather on mobile network.

This proposed work will introduces an architecture for a security system, which will detects malicious applications for smartphones. The detection of these malicious applications is not done on the smartphone itself, but in a different environment, for instance, a mobile network, where the detection algorithm is applied to a virtual smartphone. The whole system containing all parts is called the Security System, the part in the mobile network is called Security Service. There are various reasons for the domain change from the smartphone to the mobile network: Applying the security functionality on a virtual device and not on the real smartphone offers the possibility to run analysis that would be problematic otherwise. For a normal smartphone, which is managed by an customer, it might be infeasible to install a custom firmware, or modify parts of the operating system for certain checks, which has to be done e.g., for taint checking . The virtual device does not have these barriers, it can be changed arbitrarily for every malware detection concept as needed. The virtualisation of the smartphone also has the benefit of posing no risk to real hardware. If the virtual device gets damaged, e.g. by a malicious application, it can just be recreated without any cost. Another benefit of such a security system is the centralization of the malware detection. In addition, they are able to accurately decide what to check and monitor the results. This process can even be done on a daily basis.

Android has more than 100 permissions, and broadcasts, thus it gives us a probable number of combinations of permissions and broadcasts which can be saved as malware definitions and later used for comparison with application signatures, which would help us to separate the malicious applications from the good ones.

Once the applications are filtered then malicious applications will be reported on to the mobile network so that everyone using these secure application can be informed about the malware applications which are installing or already installed on their devices. This technique would work for applications which are being installed on our devices or are being installed on our devices. Once detected the filtered applications are removed from our device and our phone is free from malwares. In this approach we can also use a Smart Agent, which would be a dummy malware, and try to access sensitive data on our phone, then trace its own behavior while accessing the data. Once the behavior is traced, then this behavior can be compared with other programs to track if they are also accessing data in a similar manner, thereby marking them as malwarees and reporting them over the mobile network. Thus making our system more robust and prone to malwarees and malwares.

In research work first of all applications will checked and analyzed for various patterns and signatures.

- For new application (without log on mobile network)

For any new application without log on mobile network it will first check for malwares without installing it for abnormal patterns and signatures.

- Proposed System Flow

The downloaded application from market then it will be checked for malwares using the user feedback available on mobile network if feedback is positive then system will informs the user that application may contain malware, if it is negative, a user can install the application. But if feedback about an application is not present on mobile network then it will check for malware by scanning in signature database if the signatures not matched then it will inform the user that feedback as well as malicious signature are not present then they may install the application on there own risk.

The following methods would be used in our project for development of the security provider application on Android,

1. User Feedback Methods: Android leverages a vast amount of users which are actively using applications and facing issues with these applications. We would develop a feedback model where the users would be able to report malicious applications on
2. mobile network servers and this would define the score of these applications. If the number of reports for a given application are above a certain level, then we would mark the application as a malware application. In future if any other user tries to download and install the same application then security provider would actively scan the application and recommend to the user that this application has a potential security threat.

2. Application offline scanning: Each android application is made up of the following components,

a. Activities: The number of screens the application has

b. Services: The number of background processes by these applications

c. Broadcast receivers: The number of event receivers for this application

d. Permissions: The number of components this application has been granted access

The user is shown the permissions before the application is installed, and if the user feels that the application has an unwanted behaviour, the application installation can be cancelled by the user. Most malware affected applications take advantage of the user's negligence and ask for permissions which are not even needed by these applications. Example, a game might ask for permissions to access the messages, call logs and the internet, even though it's normal functionality does not depend upon these parameters. Thus, the user might install the application and it might send all the device's messages and call logs to an unknown server online, which is like spying on the user.

To avoid it, this work develop an offline application scanner program which would scan the application signatures and show the level of maliciousness for the given application, there by the user decides either to keep the application or to remove it.

3. Online application signature check with assistive user feedback: In this method, online signature database will be developed, and updating it on the user's phone as soon as new entries are added to the database. The signature updation would be done in two ways,

a. By manually updating the online signatures

b. By auto updating the online signatures

In manual updation, we would be manually adding malware signatures and giving the user's regular timely updates so that the user's phone can be secured.In auto updation mode, the user feedback about the application would be used and the reported application's signature would be saved and pattern recognition techniques like Hidden Markov Model or Gaussian Mixture model would be used in order to check for matching signature patterns and then these patterns would be updated on the user's mobile so that the user can be protected from applications which are reported by other user's with the help of signature analysis. And also represent a social malware detection.

IV. CONCLUSION

Thus this paper introduced a security mechanism for Smartphone’s, which uses user feedback report generated by user, to check whether the application is malicious or not. We uses apriori algorithm for malware detection. As Smartphone’s are very much prone to viruses and malwares hence we implements new approach of using cloud as a security weapon for providing security. Also we provided user feedback analysis as a solution to the problems of malwares in Android applications

REFERENCES

[1] M. Becher, F. Freiling, and B. Leider. “On the effort to create smartphone worms in windows mobile”, *Information Assurance and Security Workshop, 2007. IAW ’07. IEEESMC*, pages 199–206, 20-22 June 2007.

[2] J. Burns. “Developing secure mobile applications for android - an introduction to making secure android applications”,https://www.isecpartners.com/files/iSEC_Securing_ Android_Apps.pdf, 2008. [Online; accessed 05-Sep-2013].

[3] W. Enck, M. Ongtang, and P. McDaniel. “Understanding android security*”, IEEE Security and Privacy*, 7(1):50–57, 2009. [Online; accessed 02-Sep-2013].

[4] C. Mulliner. Advanced attacks against pocketpc phones. 2006. [Online; accessed 04-Sep-2013].

[5] C. Mulliner. Exploiting symbian: “Symbian exploitation and shellcode development”, http://mulliner.org/symbian/feed/CollinMulliner_Exploiting_ Symbian_BlackHat_ Japan_2008.pdf, 2008. Talk on BlackHat Japan 2008, visited 15.6.2009.

[6] M. Becher, F. C. Freiling, J. Hoffmann, T. Holz, S. Uellenbeck, and C.Wolf, “Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices” ,in *Proceedings IEEE Security and Privacy*, May 2011. [Online; accessed 04-Sep-2013].

[7] C. R. Mulliner, “Security of Smart Phones,” Master’s thesis, University of California, Santa Barbara, 2006. [Online; accessed 04-Sep-2013].

[8] G. Lawton. “Is it finally time to worry about mobile malware?”, *Computer*, 41(5):12–14, 2008.

[9] Google Mobile Blog, “An Update on Android Market Security”, March 2011. [Online]. Available: http://googlemobile.blogspot.com/ 2011/03/update-on-android-market-security.html

[10] D. K. Goldhammer, D. A. Wiegand, D. Becker, and M. Schmid. Goldmedia mobile life report 2012, “mobile life in the 21st century, status quo andoutlook”,http://www.b\itkom.org/60376.aspx?url=081009_bitkom_goldmedia_mobile_life_ 2012(1).pdf. [Online; accessed 01-May-2010].

[11] J. Six, “Android architecture.” In Application Security for the Android Platform, Sebastopol, CA, O’Rielly Media 2011, pp 13–24

[12] M. Becher, F. Freiling, and B. Leider. “On the effort to create smartphone worms in windows mobile”, In *Information Assurance and Security Workshop, 2007. IAW ’07. IEEESMC*, pages 199–206, 20-22 June 2007.

[13] J. Bergeron, M. Debbabi, J. Desharnais, M. M. Erhioui, Y. Lavoie, and N. Tawbi. “Static detection of malicious code in executable programs”, In *Proceedings of the Symposium on Requirements Engineering for Information Security (SREIS’01)*, 2001.