ISSN 2320 - 2602
Volume 5 No.7, July 2016
International Journal of Advances in Computer Science and Technology
Available Online at http://www.warse.org/IJACST/static/pdf/file/ijacst02572016.pdf

SURVEY ON ANDROID MALWARE DETECTION METHODS USING STATIC AND DYNAMIC ANALYSIS

Reyma Joy
Department of Computer Science and Engineering, VAST, Calicut University, Kerala, India,
reymajoy1@gmail.com
Ayana Ajith
Asst. Professor, Department of Computer Science and Engineering VAST, Calicut University, Kerala, India
ayana.ajith@vidyaacademy.ac.in

Abstract— Android, the name is quite enough to show its dominance in the mobile computing world.Android mobile apps are widely popular and hence malicious applications pose a threat to the security of the android platform. most mobile malware is designed to disable a mobile device, allow a malicious user remotely control the device or to steal personal information stored on the device. Most anti- malware applications use static analysis for detection. the objective of the topic is to detect android malware either statically or dynamically. A new method, DREBIN is proposed, which is a light-weight method for the detection of Android malware that enables identifying malicious acations directly on the smartphone. DREBIN is a method that is capable of identifying android malware with high accuracy and few false alarms, independent of manually crafted detection. For implementing dynamic analysis along with broad static analysis, a tool Honeypot is introduced which helps in maintaining repository of maximum types of malware present in the cyber world. A method to control the permissions is also done for the user to become more aware of the permissions that are given to the apps.
Keywords: DREBIN, static analyss, dynamic Analysis, honeypot .

1 INTRODUCTION

The world is contracting with the growth of mobile phone technology. As the number of users is increasing day by day, facilities are also increasing. Starting with simple regular handsets which were used just for making phone calls, mobiles have changed our lives and have become part of it. Now they are not used just for making calls but they have innumerable uses and can be used as a Camera , Music player, Tablet PC, T.V. , Web browser etc . And with the new technologies, new software and operating systems are required. Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android. One of the most widely used mobile OS these days is A DROID. Android is a software bunch comprising not only operating system but also middleware and key applications.Android Inc was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. Android is a powerful Operating System supporting a large number of applications in Smart Phones. These applications make life more comfortable and advanced for the users. Android applications are written in java programming language. Android is available as open source for developers to develop applic tions which can be further used for selling in android market. Some of the current features and specifications of android is shown in the figure 1.1.

This paper involves firstly deploying honeypot tool on our smartphone and then installing an application on our smartphone either from Google PlayStore or any other unofficial android market. After that permission will be asked from the user. Then static analysis can be done i.e. signature (finger prints) verification can be done. In dynamic analysis, while executing application is checked for monitoring system calls, monitoring network traffic, and extracting features from manifest file. Dynamic anal-

Figure 1.1 Features of android

ysis includes techniques that run application in a controlled environment.The main contribution of this work is the use of various behavioural techniques and honeypot which helps in collecting different samples of malware application execution traces. These traces can be used into two different groups to differentiate benign application and malicious applications.

Next is DREBIN, [1]a light weight method for detection of android malware that infers detection patterns automatically and enables identifying malware directly on the smartphone.

DREBIN performs a broad static analysis, gathering as many features from an application's code and manifest as possible. These features are organized in sets of strings (such as permissions, API calls and network addresses) and embedded in a joint vector space. As an example, an application sending premium SMS messages is cast to a specific region in the vector space associated with the corresponding permissions, intents and API calls. This geometric representation enables DREBIN to identify combinations and patterns of features indicative for malware automatically using machine learning techniques. For each detected application the respective patterns can be extracted, mapped to meaningful descriptions and then provided to the user as explanation for the detection. Aside from detection, DREBIN can thus also provide insights into identified malware samples.

2 EXISTING SYSTEM

Initial techniques used for malware detection were those of Power Consumption and Signature verification[2]. Then connectivity of smart phones with internet increased and malware market also started booming. With this, signature verification techniques become obsolete and there comes other behavioral techniques that detect any malware application based upon its behavior. Crowdroid is a framework that recognizes Trojan-like malware on Android smart phones, by analysing the number of times each system call has been issued by an application during the execution of an action that requires user interaction. MADAM: a Multi-Level Anomaly Detector for Android Malware uses 13 features to detect android malware at both kernel level and user level[3].
Several methods have been proposed that statically inspect applications and disassemble their code. For example, the method Kirin checks the permission of applications for indications of malicious activity. Similarly, Stowaway analyses API calls to detect over privileged applications and RiskRanker statically identifies applications with different security risks. Common open-source tools for static analysis are Smali and Androguard, which enable dissecting the content of applications with little effort.
The system TaintDroid and DroidScope enable the dynamically monitoring applications in a protected environment, where the first is to focus on taint analysis and the later enables introspection at different layers of the platform.

DISADVANTAGES:

- Initial techniques used for malware detection were those of Power Consumption and Signature verification.
- Not available for new malware instances.

3 CURRENT SCENARIO

The growing amount and diversity of these applications render conventional defences largely ineffective and thus Android smartphones often remain unprotected from novel malware. Android is one of the most popular platforms for smartphones today. With several hundred thousands of applications in different ma kets, it provides a wealth of functionality to its users. Unfort nately, smartphones running Android are increasingly targeted by attackers and infected with malicious software. In contrast to other platforms, Android allows for installing applications from unverified sources, such as third-party markets, which makes bundlingand distributing applications with malware easy for a tackers. According to a recent study over 55,000 malicious appl cations and 119 new malware families have been discovered in 2012 alone. It is evident that there is a need for stopping the pro liferation of malware on Android markets and smartphones.

4 PROPOSED SYSTEM

Malware in android can be detected by many methods.This paper involves firstly deploying honeypot tool on our smartphone and then installing an application on our smartphone either from Google PlayStore or any other unofficial android market. After that permission will be asked from the user. Then static analysis can be done i.e. signature (finger prints) verification can be done.
In dynamic analysis, while executing application is checked for monitoring system calls, monitoring network traffic, and extracting features from manifest file.
Next is DREBIN, a light weight method for detection of android malware that infers detection patterns automatically and enables identifying malware directly on the smartphone[4].
In DREBIN, we introduce a method combining static analysis and machine learning that is capable of identifying Android malware with high accuracy and few false alarms, independent of manually crafted detection. Patterns of features indicative for a detected malware instance can be traced back from the vector space and provide insights into the detection process. For efficiency we apply linear time analysis and learning techniques that enable detecting malware on the smartphone as well as analysing large sets of applications in reasonable time[5].

ADVANTAGES

Here we use both stastical analysis and dynamic analysis for malware detection, Can reduce power consumption and Detection of unknown malware.

4 SYSTEM DESIGN

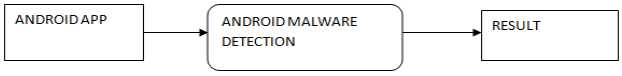

figure 3.1 level 1

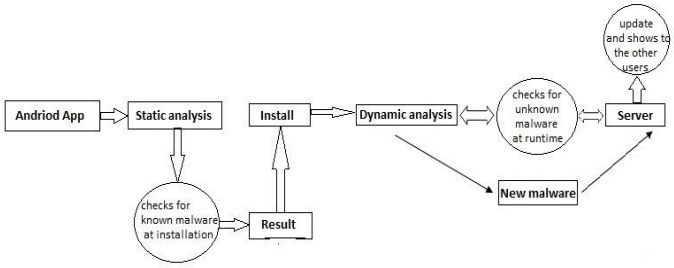

Figure 3.2 Level 2

5 MODULES

- STATIC ANALYSIS
- DYNAMIC ANALYSIS

STATIC ANALYSIS**:**

1) Check functionalities and maliciousness of an application by analyzing its source code, without executing the application.

2) Extract feature sets from manifest file.

DYNAMIC ANALYSIS**:**

1) Examine an application during runtime.

2) It may miss some parts of the code that is not executed, but it can easily reveal certain malicious behaviors too complicated to be found by static analysis.

6 CONCLUSION

Two different techniques of malware detection static analysis and dynamic analysis are combined so to obtain the optimum results. Today malware are ubiquitous so such techniques are needed which can identify both known and unknown malwares. Many techniques are used for this but most of them either work on QEMU emulator or any other virtual machine, DroidCheck directly checks on smartphone that if an application is benign or malicious. In DroidCheck not only malware are detected but new malware are lured to attack, so that we can get information about the maximum number of android families present in the market and save many smartphone users from malicious attacks.

DREBIN a lightweight method for detection of Android malware.DREBIN combines concepts from static analysis and machine learning, which enables it to better keep pace with malware development. Our evaluation demonstrates the potential of this approach, where DREBIN outperforms related approaches and identifies malicious applications with few false alarms.In practice, DREBIN provides two advantages for the security of the Android platform: First, it enables efficiently scanning large amounts of applications, such as from thirdparty markets. With an average run-time of 750 ms per application on a regular computer, it requires less than a day to analyze 100,000 unknown applications. Second, DREBIN can be applied directly on smartphones, where the analysis can be triggered when new applications are downloaded to the device. Thereby, DREBIN can protect users that install applications from untrusted sources, such as websites and third party markets.

REFERENCES

[1] DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, Konrad Rieck

[2] W. Enck, P. Gilbert, B. gon Chun, L. P. Cox,J. Jung, P. McDaniel, and A. Sheth." *Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones*". In Proc. of USENIX Symposium on Operating Systems Design and Implementation (OSDI), pages 393 – 407

[3] D.J. Wu, C.H. Mao, T.E. Wei, H.M. Lee, and K.P.Wu. Droidmat: Android malware detection through manifest and API calls tracing. In Proc. of Asia Joint Conference on Information Security (Asia JCIS),pages 6269, 2012.

[4] L.K. Yan and H. Yin. Droidscope: Seamlessly reconstructing os and dalvik semantic views for dynamic android malware analysis. In Proc. of USENIX Security Symposium, 2012.

[5] M. Zheng, P. P. Lee, and J. C. Lui. ADAM: an automatic and extensible platform to stress test android anti-virus system. In Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), pages 82101, 2012.

[6] Y. Zhou and X. Jiang. "Dissecting android malware:Characterization and evolution". In Proc. of IEEE Symposium on Security and Privacy, pages 95109, 2012.

[7] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. "Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets". In Proc. Of Network and Distributed System Security Symposium (NDSS), 2012.15

[8] 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. of ACM Conference on Computer and Communications Security (CCS), pages 7384, 2010

[9] B. Bloom. Space/time trade-offs in hash coding with allowable errors.Communication of the ACM, 13(7):422426, 1970

[10] T. Cormen, C. Leiserson ans R. Rivest. Introduction to Algorithms, MIT, Press, 1989.

[11] G. Cretu, A. Stavrou, M. Locasto, S. Stolfo, and A. Keromytis. Casting out demons: Sanitizing training data for anomaly sensors. In Proc. Of IEEE Symposium on Security and Privacy, pages 8195, 2008.

[12] N. Cristianini and J. Shawe-Taylor. "An *Introduction to Support Vector Machines*". Cambridge University Press, Cambridge, UK, 2000.

[13] C. Curtsinger, B. Livshits, B. Zorn, and C. Seifert. Zozzle: "*Fast and precisein-browser JavaScript malware detection*". In Proc. of USENIX Security Symposium,2011