**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**

*IJCSMC, Vol. 4, Issue. 4, April 2015, pg.829 – 834*

**RESEARCH ARTICLE**

# DroidCheck: Android Malware Detection by Behavioral Techniques and Honeypot

**Pallavi Kaushik[1], Amit Jain[2]**

[1]PG Student in CSE Department at Panchkula Engineering College, Mouli, Haryana, India

[2]Head of Department CSE at Panchkula Engineering College, Mouli, Haryana, India

[1] kaushik.pallavi7@gmail.com; [2] amit014@gmail.com

***Abstract*— Android, the name is quite enough to show its dominance in the mobile computing world. Android is now the market leader among all its competitors. As, it is the largest shareholder in the market it has become bull's eye for the attackers. Security is one of the major concerns for android users today. It has become the most viable target of security threats. With the increase in power and features of android applications, the vulnerability for malware attacks has increased.**

***Malware can be detected in two ways either statically or dynamically. Most anti-malware applications use static analysis for detection but they can be easily obfuscated, also they require regular updates. Static analysis use techniques like signature verification which is good for known malware but fails in case of unknown malware. Dynamic analysis means behavioural analysis of malicious application taking into consideration certain parameters like asking for permissions while installation, monitoring system calls, observing network traffic, extracting information from android manifest file. The technique illustrated in this paper not only detects known malware but it is also useful in case of unknown malware. Besides this, it lures more and more malware to attack by using the tool Honeypot. Honeypot helps in maintaining repository of maximum types of malware present in the cyber world. So, by this proposed method we cannot only detect malware but also we can collect unknown and infectious malware as well.***

***Keywords*— Behavioural detection, Honeypots, Malicious applications, Malware, Manifest files, Security, Static detection**

## I. INTRODUCTION

Android has now become leading edge. It has established itself as the primary OS among other competitive OS. As per IDC (International Data centre) worldwide smartphone shipment market share by OS from 2010 to 2014 for android was 81% (Fig.1). Among all other OS android is leading in its market share. Google currently claims more than 80,000 third party applications for their respective platform.

 

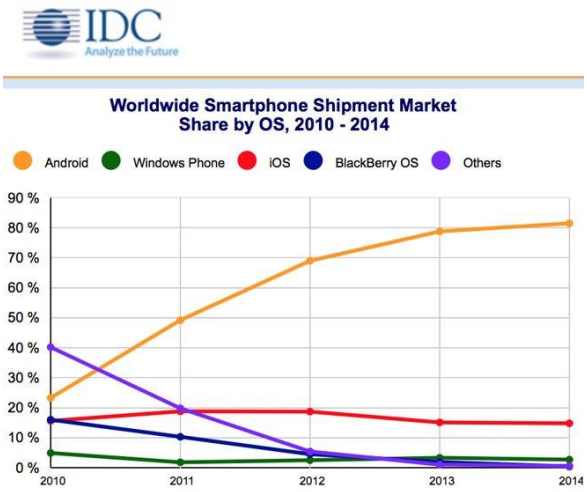


Figure 1: Showing android share among its competitors.

As per www.statista.com[1] cumulative number of applications downloaded from Google PlayStore as of July 2013 was about 50 billion (Fig 2).

According to Gartner in 2013 android has 78.4% market share as compare to 2012 when it was 66.4%. With so many applications in different android markets, android provides wealth of functionality to its users. This popularity attracts malware authors too. Allowing anyone to develop and publish application into the android market presents fortuity for attackers to easily deliver malicious applications.

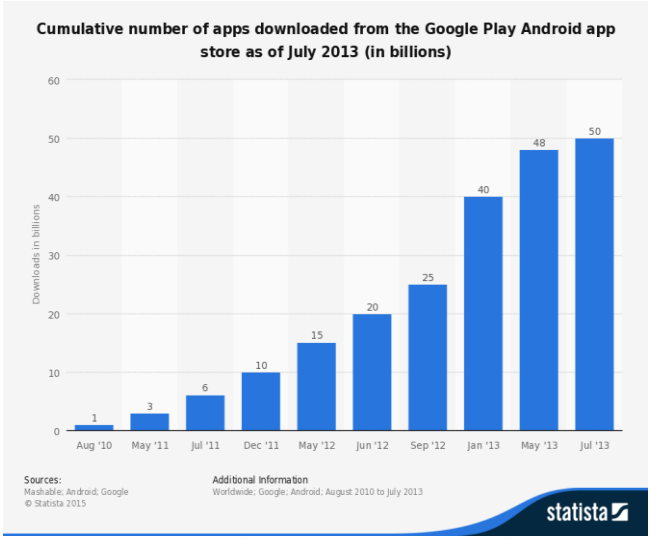


Figure 2: Statistics of applications downloaded as of July 2013

Android application distribution is based on a centralized market where developers can upload and sell their applications. Authorized market is Google PlayStore but now many unauthorized markets are available which contribute in the distribution of malware, as applications are not scanned in these markets. A number of applications have been modified and malware have been binded, packed and spread through these unofficial repositories. John Oberheide made a proof of concept application as an Angry Bird bonus to show the weakness of security of Android market place. So far two approaches have been proposed for the analysis and detection of malware. i) Static Analysis ii) Dynamic Analysis (Behavioural Analysis). Static Analysis is used by antivirus components based on binaries inspection looking at suspicious patterns. Earlier this technique was quite successful but now malware authors have developed various obfuscation techniques effective against static analysis.

Dynamic analysis 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. The rest of this paper is organized as follows: Section 2 proposes the related work in the field of malware detection. Section 3 consists of the methodology used for malware detection and collection. Section 4 will cover the future references. Section 5 will conclude whole paper. Section 6 shows the references and bibliography

## II. RELATED WORK

Earlier first malware(Cabir)[2] was detected in 2004 for Symbian mobile system. Since then there was lot of work done in the field of malware detection. Initial techniques used for malware detection were those of Power Consumption and Signature verification. After that much of the research work was done in that area. 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 behavioural techniques that detect any malware application based upon its behaviour. This means how application uses its resources or other parameters and behave accordingly. Much of the research work is done in this area and many techniques have been developed. Crowdroid [3] 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. MADAM has been tested on real malware found in the wild and uses a global-monitoring approach that is able to detect malware contained in unknown applications. Finally, surveys were also done on security solutions for mobile devices[4][5]. Honeypots are new in the field of security and mobile honeypot concept is also very new. First mobile honeypot was HoneyDroid[6]. After that many surveys were done on honeypots as well[7].

## III.METHODOLOGY

The technique used in 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. Once the application will be installed all events of that application will be monitored. Events will be monitored firstly by asking permissions from the user whether to install this application or not, then static techniques (Signature Verification) and dynamic techniques (by reading manifest file, monitoring system calls and network traffic) will be used. Honeypot will maintain a log file which will record details (parameters) of an application whose behavior will be seen malicious. If in between it is found that an application is trying to access privileged services it will be aborted at the same moment and user will be notified plus logs of that application along with the application details will be sent on our server. On server side to cross-verify if the application is malicious or not, it will be scanned on www.virustotal.com[8] and if the application is infected it will be added by server in its repository of malicious applications.

So, firstly user will download an application from internet(no matter if it is from Google Play or from any other unauthorized android market) then following steps will occur:

### *A. Permission*

Permission will be asked from the user. If user grants permission application will be installed otherwise it will be aborted. An android application can access limited system resources. To access sensitive API's the application must declare required permissions in AndroidManifest.xml file. This file contains application related properties. Sensitive API's which should not be given permission for each and every application are those of camera function, loading data(GPS), Bluetooth and telephony functions, SMS/MMS functions, Network data connections and details like IMEI number(Fig. 3).User installs the application by accepting the required permissions. Once application is installed then during runtime system will no longer notify the user when these sensitive API's are being accessed again. Declining these permissions will abort the application installation.

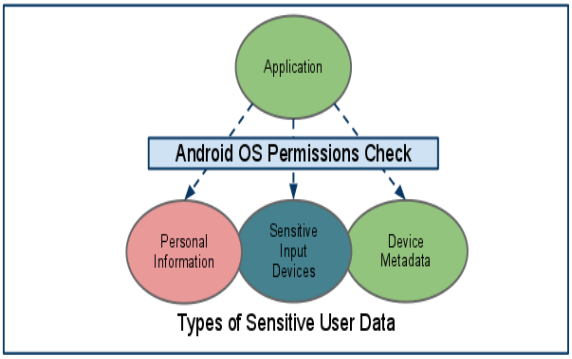


Figure 3: Sensitive API's

Now, next will be to do static analysis which will be done by verifying signatures of an application.

 

*B. Static Analysis*

This analysis require signature verification. Signatures are unique fingerprints of a file and signature verification means searching a known pattern. In this step, we have already maintained a definition file which stores many features of a known malware. These features are type of a file, hash, string signatures and series of bytes in a file. When we scan an application these features will be extracted from the application and will be matched with those specified features in the definition file. So, when any new application will be scanned, it will be classified into finer and finer categories and then will be matched against small set of signatures. E.g. after only few memory reads and processor cycles we can define type of application JAR file. Then we will match it with the features of JAR file if they are same then the static analysis will be aborted otherwise we keep on going with other features.

To prevent any false alarm we have to check if the code of suspected application is either the malicious code or a checksum of it(checksum is a method used to determine if data has been changed or not, it involves summing of all the bits in a file). Usually, signatures of a file are generated with hash.

In this step if signatures of an application matches with any of the known malware then that application will be aborted and log file will be populated. Then that log file will be sent to our server for cross verification. If signatures do not match then we will move on to behavioural analysis.

*C. Dynamic(Behavioural) Analysis*

Dynamic analysis means checking the application if it is benign or malicious while it is executing. In this analysis we will check behavioural characteristics of an application. Behavioural characteristics include checking of following parameters:

- monitoring system calls.
- monitoring network traffic.
- extracting features from manifest file.

We will check above characteristics step by step , if at any step it is concluded that an application is malicious then it will be logged into the log file and sent to the server.

*1) Monitoring System Calls:* System call executes a specific function that controls a device or executes a privileged instruction. By monitoring system calls in a device we can be ensured that any malicious application is not using privileged function in the background. We will maintain a script of privileged system calls. These privileged system calls will be for sending messages(SMS Manager), system call that detects our gps location, system call that access our contact list or images, system call that read files from the SD card, system call that check our network connection again and again. If any of the application accesses or tries to access those system calls mentioned in the script either in background or foreground, it will be notified to the user and again that application has to take permission of user.

sTrace tool will be used to monitor single application and it will be much easier to find abnormal behaviour. If sTrace is installed set uid to root then only invoking user will be able to attach to and trace processes owned by any user. However, it is a bit time consuming process because for this we have to make USB connection of our phone with that of adbshell.

*2) Monitoring Network Traffic:* To check if downloaded application is not making excessive use of internet or to monitor its activities and to analyse its packets we will use a tool Shark For Root. This tool can be easily downloaded from Google PlayStore. This is a traffic sniffer and works on 3G and Wi-Fi. This is a small version of wireshark tool for android phones. This application will create a *.pcap and we can read *.pcap having this file in PC with wireshark or we can use Shark Reader, this program allows us to read *.pcap directly in our android phone. Shark application only works with rooted devices. So, for using this we have to root our device. By unlocking the bootloader we can gain root access to our device.

While monitoring network traffic if any application is sending maximum packets , it will be notified to the user and permission will be asked from the user to abort that application. If the application is aborted again it will be logged in the log file and that will be sent to server.

*3) Extracting Features from Manifest File*: Manifest file for an android application is a resource file which contains all details about the application[9]. This file contain information about application package, including components of the application such as activities, services, broadcast receivers, content providers etc. This file describes the functionality and requirements of our application to android. It is a key file that works as a bridge between the android developer and android platform. Every android application must have AndroidManifest.xml file and it is located inside the root directory. This file is also responsible to protect application by providing permissions. It also declares android api that application is going to use. Format of manifest file will be same for both benign application and malicious application.

 

But we will extract following features from the manifest file to differentiate among benign application and malicious application:

- permissions used by application.
- number of redefined permissions.
- process name.
- intent filter (priority, category, actions).

From above characteristics if we find that the application is malicious again information will be entered into the log file and sent to server. Now next step will be of deploying honeypot.

*D. Honeypot*

Honeypot is a special kind of software with the only purpose of being probed and attacked. Main intention behind deploying honeypot is to learn about nature and characteristics of different kinds of attacks. It also helps in gaining about the threat intelligence. Real life examples of honeypot deployed on PC are honeyd and the Honeywall CD-ROM. Uptil now various studies have been made on smart phone honeypots which include HoneyDroid : creating a smartphone honeypot and Nomadic Honeypots : A novel concept for smartphone honeypots[10]. Honeypots can be classified according to the level of interaction of the attacker they allow. They can be classified and identified as:

i. High interaction honeypots
ii. Low interaction honeypots

Honeypot with a different level of interaction provide different quality and quantity of information.

Main benefit of deploying honeypot can be achieved if that honeypot is visible to the attacker, so we have to ensure its visibility. In case of DroidCheck, our honeypot perform two functions:

i. Monitoring events or applications
ii. Generating logs and sending them to the server along with details of malicious application

Events are monitored by various static techniques and dynamic techniques mentioned in the above sections. Whenever any of these event monitoring techniques come across any malicious application they will notify it to the honeypot, honeypot will maintain a log file and populate the log file with the necessary features of that application(type of application, why it is aborted)sent by those techniques. Honeypot will only maintain logs of those files which are malicious and sends them to the server along with the malicious application details.

In the case of DroidCheck, honeypot is not virtualized as in case of HoneyDroid because of event monitoring and as soon system feels that an application is malicious it will abort that application and will maintain log and send it to server. So, in that case honeypot is safe and will not be compromised.

*E. Handheld server*

This server will be any other android device. Its main purpose is to collect the logs and application details sent by the honeypot and cross verify them. This server will cross verify if an application is malicious or not on www.virustotal.com. Server will maintain a database of malicious applications with it and once decided that the application sent by honeypot's log file is malicious, server will modify its database and add that file to its database In this way we can detect new malware families and update our database of new malwares.

## **IV.** CONCLUSION

In this paper 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.

## **V.** FUTURE WORK

As a future work, I will try to enhance this idea of DroidCheck into a light weight approach as this model consumes many resources. Light weight approach will include various other techniques which can easily detect both known and unknown malware using less system resources. These techniques will use various machine learning concepts and artificial intelligence concepts which we can deploy directly on our smartphone and utilize minimal resources.

## References

[1] Statista Website. Available: http://www.statista.com/statistics/281106/number-of-android-app-downloads-from-google-play
[2] Cabir, Smartphone Malware. Available: http://www.f-secure.com/v-descs/cabir.shtml
[3] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behaviour based malware detection system for android*," in Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, ser. SPSM* '11. New York, NY, USA: ACM, 2011, pp. 15–26

 

[4] Tiwari Mohini, Srivastava Ashish Kumar and Gupta Nitesh" Review on Android and Smartphone Security " in *Research Journal of Computer and Information Technology Sciences* Vol. 1(6), 12-19, November 2013
[5] Mahinthan Chandramohan and Hee Beng Kuan Tan "Detection of Mobile Malware in the Wild "
[6] Collin Mulliner, Steffen Liebergeld, and Matthias Lange" HoneyDroid - Creating a Smartphone Honeypot "
[7] Dr.Hanaa Mohsin Ahmed, Dr. Nidaa Flaih Hassan, PhD Student Assmaa A. Fahad " A Survey on SmartPhone Honeypot " in *International Journal of Computers & Technology*, vol. 11, No. 4, Oct 2013
[8] Hispasec Sistemas. Virustotal malware intelligence service. Available: http://bit.ly/mytpXt
[9] Ryo Sato1, Daiki Chiba and Shigeki Goto " Detecting Android Malware by Analyzing Manifest Files " *in Proceedings of the Asia-Pacific Advanced Network* 2013 v. 36, p. 23-31
[10] Steffen Liebergeld1, Matthias Lange1, and Collin Mulliner2" Nomadic Honeypots: A Novel Concept for Smartphone Honeypots

 