ISSN:0975-9646

An Analysis and Defending of Mobile Malware in the Android Devices

P Surendra[1] K.Navin[2]

[1]Dept. of Information Security and Computer Forensics
[2]Associate Professor, Dept. of Information Technology

[1,2]SRM University,Kattankulathur, Chennai, India.

Abstract— **Android mobile phones are more popular now a days. With the use of android phones people always concern about the security and malware attacks. Here I introduce an approach for detection of malware by its behaviors. Suspicious behaviors are detected by comparing trace abstractions to reference malicious behaviors. This concept allows us to grasp common suspicious behaviors rather than malware code and then, to differentiate malware modification. I present an implementation validating to my approach. First have to analyses the applications with respect to behavior patterns. This paper analysis the mobile threats, vulnerabilities, based upon the detection principles, architectures and collected data, especially focusing on IDS-based models and tools.**

Keywords—android; security; mobile; malware

I. Introduction

Mobile devices (smart phone) become more and more global, and often needed in our day to day life. Usually they contain lots of important and sensitive information, like a list of contacts, text messages, emails and important calendar dates etc. Latest models feature a complete OS, but for many people they are just phones, so there is an under-estimation of the risk connected to phone security. This makes Phones an interesting target for malicious users and hackers. Because smartphones are becoming more diversified in providing general services (i.e., IM and music), the effect of malware could be extended to include connecting to unauthorized websites, wipe out batteries, extra charges, and bringing down network etc. The APK (dex2jar) conversion is used to analyses the flow, operation, functionality of malware using open source tools. This is useful in understanding the files, services, code and parameters which were added or modified by the malicious software. It includes analysis in two phases namely Behavioral Analysis and code Analysis.

II. Mobile Threats

The mobile threat model includes different types of threats: Malware, grayware, and spyware. We differentiate Policy between them based on their delivery method, legality, and notice to the user. This paper focuses specifically on malware, personal spyware and grayware.

Malware. Malware, short for malicious software, is software used to disrupt mobile operation, gather sensitive information, or gain access. It can appear in the form of program code, active data, and other software. Malware' is a term used to refer to a different forms of invasive software. Malware includes viruses, Worms and Trojans, etc. Malware gains access to a mobile or other devices with the intension of gaining access to device, gathering information, damaging the mobile or other devices, or disturbing the user, etc.

Spyware. As the name suggests it spies on the devices for information such as device location, email messages, user name, passwords and browsing history over a time period. With spyware, the attacker can get access to the device and installs the malicious software without the knowledge of user's. Personal spyware sends the information of victim's to the Attacker. Some examples for spyware are keyloggers.

Grayware. Some applications collect user data for the purpose of website, company promotions. Grayware supervise on users, but the most of the legitimate companies that distribute grayware do not aim to harm users. Grayware is a very broad term for all of those computer bugs that are annoying but not necessarily destructive, including adware, comedy or fun programs etc.

Trojan. Unlike virus, a Trojan is a non-self-replicating type of malware but as destructive as virus. Trojans are different types, some of them are remote access Trojan, proxy Trojan, and ftp Trojan etc.

Virus. A virus is a type of malware that replicates itself by being initiated to copying or copied into other program. Viruses are spread by email attachments or instant messaging messages.

Worm. Worm is a self-replicating virus that does not alter files but resides in active memory and replicates itself. A worm is a program that replicates itself in order to spread to other computers. Often, it uses a network to spread itself. Worm spared by utilize vulnerabilities in operating system.

*Botnet***.** Mobile bots are just like computer bots. If smartphone is not protected by an antivirus, it can get infected with a bot malware, added to a botnet and controlled by a bot master from afar. As scary as all this may sound, getting to know the problem in detail makes it easier to find a solution for it.

III. Malware Analysis

Basics of malware analysis:

Malwares are increasing in every day and it's become difficult to stop malware. There are so many antivirus are there to detect the malwares but with the rapid growing of technology the attackers take the advantage with different programming languages to develop the malicious codes. Even though the antiviruses are releasing new updates everyday some of the viruses are narrowly escaping from the antivirus. Reverse engineering is help to detect the malware behaviors from the code and it is used to check the permissions list in the manifest file.

Types of malware analysis:

a) Behavioral analysis. Behavioral analysis is used to analyses the permissions and compare the package name of the each application with the help of the malware database which is created in local host. Here we can check the permissions list of all application installed in the mobile devices. The package names are unique and not possible to rewritten once they installed into the market or saved in the web sources.

b) Code analysis. Code analysis is used to analyses the code of the programs which are embedded with malicious software. As it is very difficult to get the source code of the malware especially executable we need to analyses the binary code. There are debuggers and decompiles which are used to convert the malware to its binary form or assembly level. By analyzing the code, a reverse engineer will come to know the exact malicious code which is embedded in the actual code.

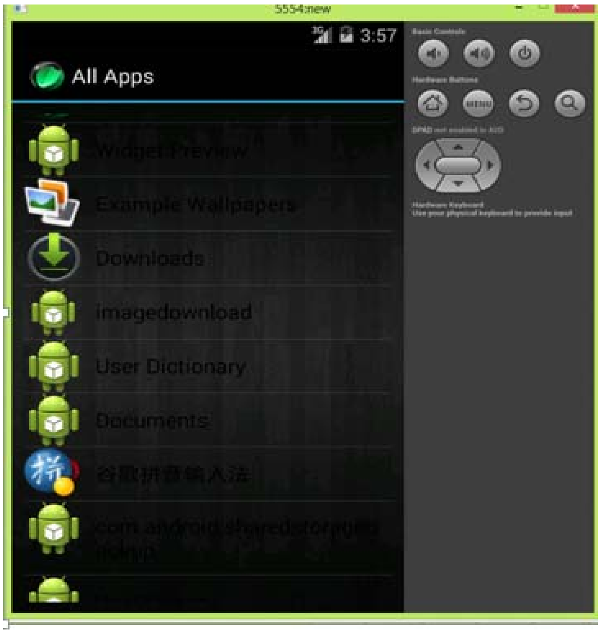
IV. Methodology

There are many methods involves in malware detection but two important methods are behavioral analysis and code analysis.

1. Behavioral Analysis

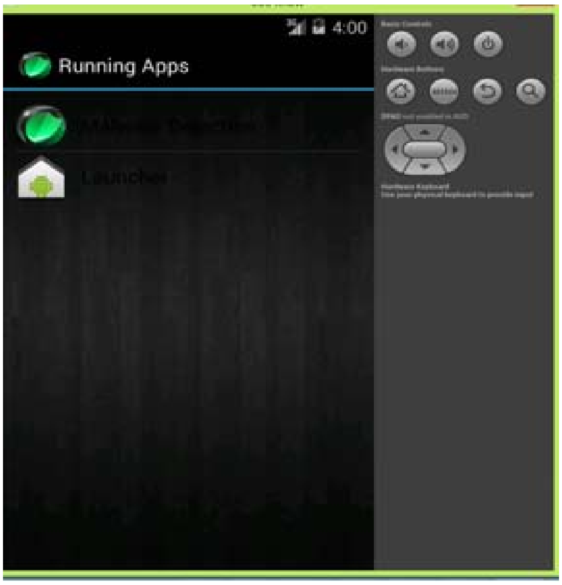
A) Getting installed apps:

Android market has a huge selection of third party developed applications, which can be downloaded or installed by users either through an app store such as Google Play store or the Amazon Application store, or by downloading and installing the application's APK file from other website. The Google play store allows the mostly trusted applications but the third party applications which are downloaded from websites are dangerous and harm to install without the verification of malware. Here in this module we are getting all the applications which are installed from the play store and other third party services. The play store apps are mostly genuine and trust worthy. The applications which are installed from the third party services need to be verified before installing into mobile device. Here if we listed all the installed applications, users can able to check the all the applications which are installed onto the device.



B) Getting running tasks:

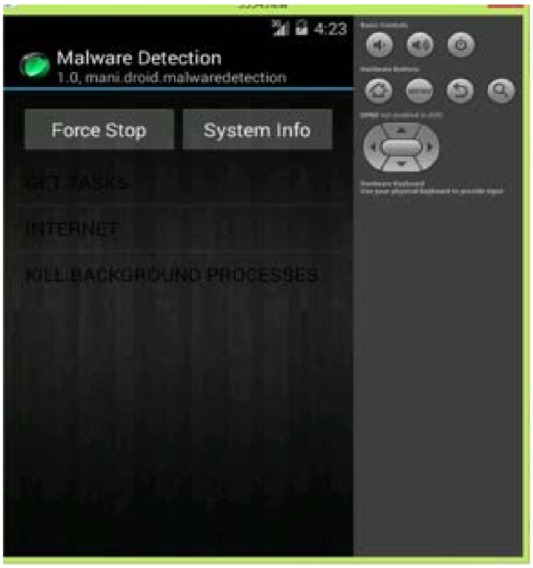
In android processes and applications are two different things. An application can stay running in the background without any processes eating up your mobile resources. Android operating system keeps the application in its memory, so it launches very quickly and return back to its prior state.



When your phone runs out of memory, android start killing the tasks automatically that we have not used in a while. Most of the malwares are running in the background without the user knowledge, which can send the data to other sites or receive the malicious data from the servers or websites. User can find the malicious application and remove it, if the user not opened any application but they automatically running in the background process, it is known as malware or malicious software.

C) Extract information:

Android system highly relies on permission based mechanisms. There are huge set of permissions in android that can access to different resources. If the user selects any application which is running in the device that shows all the set of permissions which are used by the specific application.



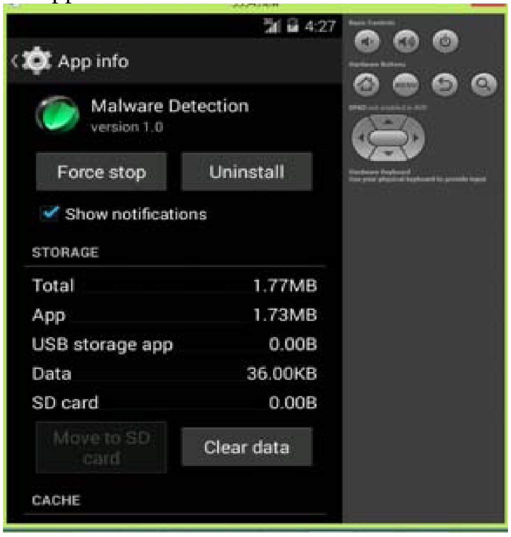
Malware Detection
1.0, mani.droid.malwaredetection

GET TASKS
INTERNET
KILL BACKGROUND PROCESSES

By seeing the permissions of running application user can easily identify the all the permissions of applications, which are valid to use and which are not to be used by the application. For example if the application uses network access permission , but if there is no need to use the same permission in the application then the application can connect to third party servers which are blocked by the Google.

D) Detection of malware:

In the websites many malwares are there to affect the android operating system, when we install the application from the third party sources. So we maintaining the huge database of malware in the local host or in the malware detection application.



Malware Detection
version 1.0

☑ Show notifications

STORAGE

Total	1.77MB
App	1.73MB
USB storage app	0.00B
Data	36.00KB
SD card	0.00B

CACHE

If the mobile user scan the all applications installed in the mobile device each application is compared to our malware database if any app found malware, our system shows error and instruct the user to uninstall the particular application. By maintaining the malware database in the local host when new malware found user can easily update the malware database

2.CODE ANALYSIS:

a) Save the APK file to new folder and rename the APK extension and save it.

b) Now download .apktool, dex2jar, jd-gui, signapk and keep all the files in same folder. Open the command prompt and give the path of the APK file what we want to check.

c) Use dex2jar and java decompile tools to get back the java class files.

d) APK tool, APK install window and resource frameworks are used to retrieve the xml files.

e) Now we can analyses the complete code of the class files and manifest files for the malicious code and other permissions used by applications.

Manifest permissions:

```
<uses-permission android:name="android.permission.GET_TASKS"/>
 <uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
```

V. CONCLUSION

The first thing we need to know is no antivirus is 100% safe, to detect the malware The major difficulties and issues of malware analysis and detection is that attackers uses various programming languages, packers to hide the code from the original code. The Reverse Engineering tools needs to be more advanced as they lack exposure over many programming languages and platform etc.

REFERENCES

[1] Dennis Distler, "Malware Analysis: A Introduction",RetrievedFrom:http://www.sans.org/reading_room/whitepapers/malicious/malware-analysis-introduction_2103.

[2] Nitin Padriya, Nilay Mistry, "Review of Behavior Malware Analysis for Android", International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 7, January 2013.

[3] Raj deep Chakra borty, "Detailed analysis of the continuously evolving threat of Malwares", Retrieved http://www.malwareinfo.org/library/whitepapers/MalwareAnalysisHow2.pdf, Last Accessed: 24 August, 2011

[4] Stephen. A.Ridley, "Android Malware ReverseEngineering",Retrievedfrom:http://dl.dropbox.com/u/2595211 /HelloMotoAndroidReversing.

[5] Google Android, Retrieved http://developer. android.com/guide/basics/what-is-android.html.

[6] Troy Vennon, "Threat Analysis of theAndroid Market",http://www.globalthreatcenter.com/wp-content/uploads/2010/06/Android-Market-Threat-Analysis-6-22-10-v1.pdf, Last Accessed: 24 August, 2011.

[7] Johannes Kinder, Stefan Katzenbeisser, Christian Schallhart, and Helmut Veith. Proactive detection of computer worms using model checking. IEEE Transactions on Dependable and Secure Computing, 7:424{438, October 2010.

[8] Dong-Jie Wu1, Ching-Hao Mao2 "DroidMat: Android Malware Detection through Manifest and API Calls Tracing"2012 Seventh Asia Joint Conference on Information Security. 978-0-7695-4776-3/12 /IEEE.