ANDROID MALWARE ANALYSIS : A SURVEY PAPER

Tarang Kumar Barsiya[1], Dr. Manasi Gyanchandani[2]and Dr. Rajesh Wadhwani[3]

[1]Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal
[2]Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal
[3]Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal

Abstract

The prevalence of the smart phones, the large market share of android and the openness of the android market make android more sensitive platform for malware attacks. For understanding the threat to security and privacy it is important to analyze the behavior of the malicious application. For a forensic point of view an analyst need to understand the behavior of the application to find out the suspicious application In this paper, we are focusing on a different type of android malware analysis techniques: static analysis, dynamic analysis and hybrid analysis (combination of static and dynamic analysis). This paper also presence different methods of these analyses along with their functionality and working.Comparisons of these analyses have been done along with their advantages and disadvantages.

Keywords

Android malware,static analysis, dynamic analysis, hybrid analysis, Android security.

1. Introduction

Smart mobile devices have been widely used and lots of sensitive information is saved in smart phones. A huge number of malwareis being developed for stealing the private data.There are lots of smart phones available in the marketusing different types of operating system. Android is one of the operating system for mobile devices and is popular in the market. Android is an open source operating system for mobile devices and it is based on the Linux kernel. Since 2008 Android is the most popular operating system for smart phone. Now a day peoples stores all their confidential data, financial information and personal content like photos, videos bank details in their smart phones, so these smartphones become more vulnerable and an attacker can attack on their devices to steal their information. Due to the openness of the android system, itbecomes very easy for an attacker to develop malware which can harm any Android device. In thispaper wehave discussed how to find the malicious application and how to analyze them. For analyzingmalware we have discussed the different type of techniques,but this paper focus on only 3 techniques of analysis i.e. static, dynamic and hybrid analysis. In each type of analysis we have different types of methods and frameworks, but we are focusing only some important methods and frameworks of these analysis techniques.

2. Android Malware analysis

The android market is very large, so it is very easy to develop malware in the markets. Now days there are a very large number of malware available in the market and they are known as malware family [2]. These malware families can be classified according to their functionality like: privacy escalation, remote control, financialchange and private information stealing.There is also lots of spyware application which can steal your personal information and send it to a remote server. For forensic [1] point of view we need to analyze all this application and we have to find that it is a good-ware application or malware application. So for analyzing android malware we have three techniques, namely, static analysis, dynamic analysis and hybrid analysis.In static analysis, importance is given to the code of the applicationto find the malicious code and fix them. But in dynamic analysis, the application is run inthe sandbox and the abnormal behavior of the application is found out.

3. Static Analysis

Android app is packaged into APK file [13]. It is a ZIP archive based on JAR file format.The structure of APK file is as shown in figure 1.The APK filecontains app bytes code, manifest file UI layouts. The manifest file is mandatory because without manifest the information of the application cannot be installed and execute. Manifest file containsMeta information such as requested permission, services, broadcast receivers, content provider, activity and SDK version. For analysis of application, we need to check all the contained information of manifest file as well as examine the byte code to extract all the Javaclasses, objects as well as methods.Here we are discussing same methodology to analysis the APK file.

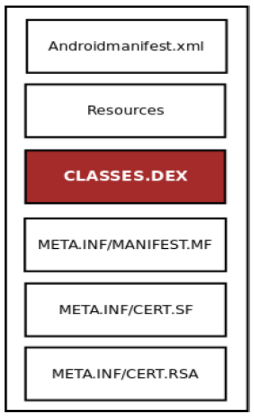

Figure 1APK file format

Signature: -It contains the signature of the message digest of APK file. If signature of any APK file is changed that means there is something wrong with APK file and it can be easily identified. The analyst also collectsthe signature of the malware to quickly identify the malware**.**

Bytecodes: -Android coding is written in the Java platform so all the executable part of the application are stored in class.dex.This folder contains all the compiled classes of the program in the form of bytecodes. So we need to use reverse engineering to understand the byte code so for that we use the DexToJar converter tool. For analyzing the code we need to take care of some important point like code obfuscation, string encryption and anti-forensic code.

Resources: -resources are non-executable part of the program. Most of the resources are user interface component such as bitmaps, menu and layout. In most of the cases, malwarerunning in the background and does not have any user interface. For example AndroidManifest.xml is a resource file. It is encoded in binary format in APK file. It contains the component of the application and permissions request of an application.

Component: -Activity, services, broadcast receiver and content provider are the component of the malware who run in the background may use the services component as well as a receiver component in order to get the booting access of the system. So while checking the component we can find the behavior of the application.

Permissions: -To access some protected APIs of android we need to request some permission [12]. Like sendTextMessage, getPAckageInfo, getSimCountryIso, READ_Contect, Internet. If we found any suspicious permission in any simple application, then we can understand that it's a malware.

4. Static Analysis Methods

The attacker uses some anti-forensic techniques by which analyst unable to find the suspicious code. There are some methods which are used for anti-forensics point of view.

4.1 Code Obfuscation

Android is written in Java so attacker user Obfuscation code [12] in which all the classes, packages, methods are renamed to single alphabet, so it is very hard to distinguish different parts of the code and even it difficult to understand the functionality of the code.

4.2 String Encapsulation

The string is very important for information point of view in reverse engineering. Many malware developers use strong encryption to avoid the plaintext detection. For string encryption they use DES or AES algorithms. So that an analyst cannot decrypt the cipher text easily.

4.3 Environment verification

Some type of malicious code only runs on some specific type of mobile device. All mobile verify the code before executing them. So if a malware run on the different IMSI device so it will stop executing so an analyst cannot understand the behavior of the malware.

4.4 Self-define communication protocol

Some malware are executing remotely that means they attach with the code and execute remotely on specific event. They work like client server and communicate each other and steal private information from the device.

4.5 Sensitive data access

In android AKP, thereis lots of userdefining permission. If any simple applicationuser READ_CONTECT type permission they it can be suspicious. The whole processes of static analysis are shown in figure 2.

5. Dynamic Analysis

In dynamic analysis we generally analysis the behavior of the application. In dynamic analysis, we monitor the file system and network flow as well as outgoing SMS phone call and loading of the addition Dex or native code during runtime. There are some more additional analysis like stimulation, taint tracking, method tracing and system level analysis [10]. Now a days there are lots of framework are available for analysis of the malware. There are some frameworks which work with both static as well as dynamic analysis, which are called as hybrid analysis framework. Start from the beginning, we are going to summarize the framework with their features.

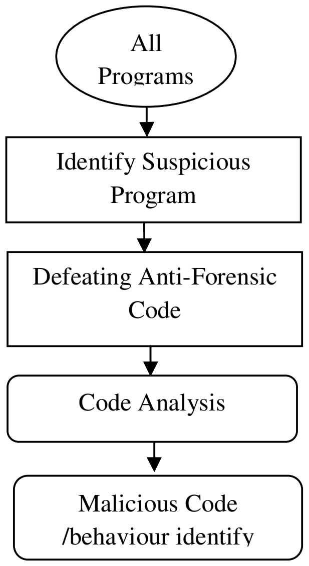
All Programs
Identify Suspicious Program
Defeating Anti-Forensic Code
Code Analysis
Malicious Code /behaviour identify

Figure 2Static Analysis Process

5.1. Taintdroid

Taintdroid [4] is a multiple granularity taint tracking approach on android system. FirstlyTaint Droidautomatically taints all data from sensitive source and put labels as sensitive data. When these labelled data transmitted through the network or leave the system than Taint droid logs the data’s labels and identify transmitting the data. Taint droid only tracks data flow not control flow that means only explicit data can be traced not implicit data. Taintdroid is very effective for tacking sensitive information, but it causer signification false positive when the tracked information contains configuration identifiers. Thus, the taint source should not contain a configuration identifier for more accuracy.The framework of Taint droid is shown in figure 3.

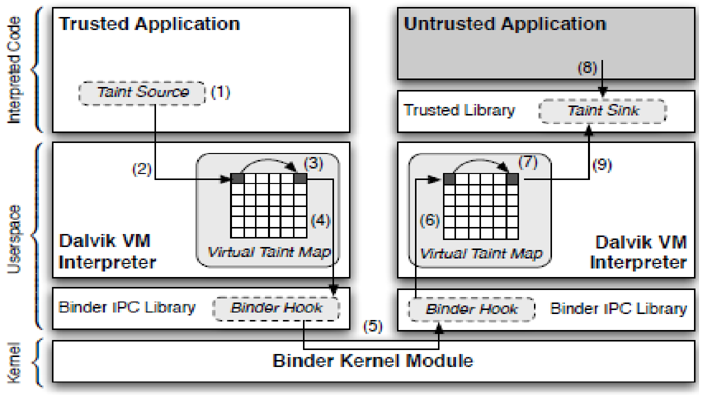

Figure 3Taint Droid Framework

5.2 DroidScope

To facilitate custom analysis, Droid Scope [8] exports three tiered APIs that mirror the three levels of an Android device: hardware, OS and Dalvik Virtual Machine. On top of DroidScope, we further developed several analysis tools to collect detailed native and Dalvik instruction traces, profile API-level activity, and track information leakage through both the Java and native components using taint analysis. For an analysis point of view they have implemented 4 analysis plug-in, namely, API tracer, native instruction tracer, Delvik instruction tracer and taint tracer.API tracer monitor that how an application interact with the rest of the system through system calls and libraries call. Native instruction tracer gathers information about each instruction, including row instruction with its operands and values (register and memory). Delvik instruction tracer works same as a native instruction tracer and log the decoded instruction to a file in the dexdump format. It logs all the classes, fields and all available symbol information. The working framework of Droid Scope shows in figure 4.

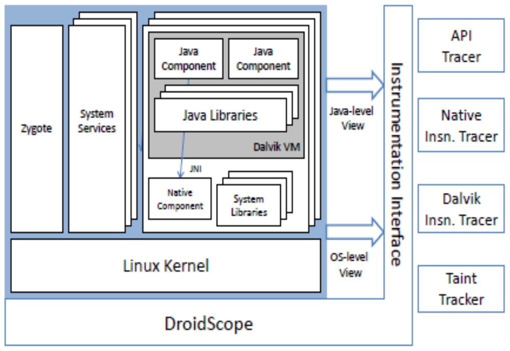

Figure 4 Framework of Droid Scope

5.3 Profile Droid

Profile Droid [6] this framework the analysis part is divided into 4 layers. Static specification, user interaction, operating system and network. All 4 layers consist two part first one is monitoring and second one is profiling. Monitoring system logs all the activity of the application and sends it to a specific computer for profiling. At the static layer they use ApkTool for analysis of APK file. On this layer they mainly focus on manifest.xml files and the byte code files. At the user layer, the focus on user generated event, i.e., the event occurs when a user interact with the android device at the run time. To get the information from user layer they use a combination of log-cat and get event tools.The log - cat is used to capture the system debugging output and log message from the application. Get-event tool readsthe /dev/input/event* to capture user input from the input device.At OS layer they monitor the operating system activity with the help of system calls. The working of Profile Droid is shown in figure 5.

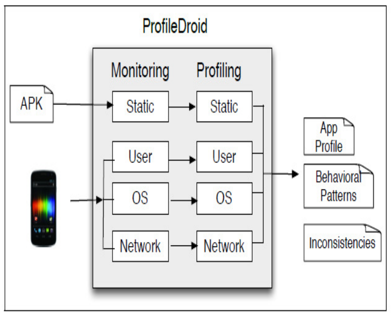

Figure 5 ProfileDroid

6. Hybrid Analysis

Hybrid analysis [5] is a combination of static and dynamic analysis, In android market, there are lots of malware which need both static and dynamic analysis, for analysis these types of malware we user hybrid type of framework which do both the analysis. With the help of hybrid analysis malware detection become easier and accuracy of malware detection also increases

6.1 **Mobile Sandbox**

Mobile sandbox [7] is a combination of static as well as dynamic analysis. In mobile sandbox we analysis APK file for static analysis. In this we scan the anti-viruses, user permissions, parse the manifest.xml file for identifying thesuspicious code. In dynamic analysis they use emulator for running the suspicious application and check the behavior of the application. They also check the native call as well as network traffic for better understanding the behavior of the application. Working of Mobile Sandbox is as shown is figure 6.

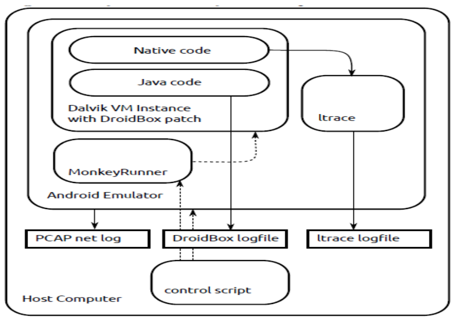

Figure 6Mobile Sandbox

6.2 **Andrubis**

In Andrubis [5] framework firstly we do static analysis and the result of the analysis is used for dynamic analysis which give more effective results.In static analysis this framework is concentrated of android manifest.xml file and bytecode. Allthe information which comes from static analysis isused for dynamic analysis. In dynamic analysis they do following analysis, namely, Stimulation, taint tracing, method tracing, system level analysis. The component and the framework of Andrubis are shown in figure 7.

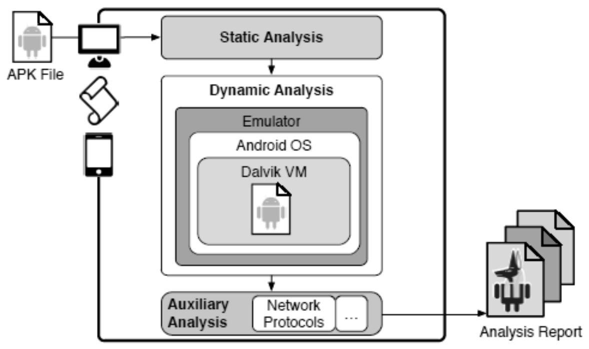

Figure 7Andrubis Framework

7. Comparison

The present paperdiscusses all the different type of analysis of android malware. All have necessary to detect malicious behavior of the application. The first step of analysis is static analysis. It is necessary for every application to go through static analysis because most of malware developers change the code as well as permissions in APK file, but now a day's analyst wants to know the behavior of the application, which type of event attacker wants to execute, for that they need to run the application on sandbox and analyze the behavior of the application. For malware analysis, we need static and as well as dynamic analysis, that's why hybrid analysis is used for most of the framework because it can do static and dynamic analysis simultaneously.

8. Conclusion

In this paper, we discussed all types of analysis of the android malware, their methods and frameworks. In static analysis, we discuss code obfuscation, anti-forensics technique, and different types of suspicious permissions. In dynamic analysis we have seen different type of frameworks, their functionality, process and limitations. And at last we discussed about hybrid analysis frameworks which workwith both static as well as dynamic analysis.Finally, we conclude that the hybrid analysis framework is more effective because of the dual nature of the analysis. It compares all the possible conditions for analysis of malware. But still there are lots of points we have to cover for analysis of android application, because android market is growing very fast and there are lots of new types of malware families are developing.

9. References

1. Thomas Eder, Michael Rodler, Dieter Vymazal and Markus, "ANANAS – A Framework For Analyzing Android Applications", International Conference on Availability, Reliability and Security2013.

2. Alessandro Reina, Aristide Fattori and Lorenzo Cavallaro,“A System Call-Centric Analysis and Stimulation”,Techniqueto Automatically Reconstruct Android Malware Behaviors, EuroSec ’13, April 14 2013.
3. Juanru Li, Dawu GU anduhao Luo, “Android Malware Forensics:”, 32nd International Conference on Distributed Computing Systems Workshops Events, 2012.
4. William Enck, Peter Gilbert and Byung-Gon Chun, “TaintDroid: An Information-Flow Tracking System for Real-time PrivacyMonitoring on Smartphones”, 9th USENIX Symposium on Operating Systems Design and Implementation
5. Martina Lindorfer, Matthias Neugschwandtner and Lukas Weichselbaum, “ANDRUBIS: A View on Current Android Malware Behaviors”.
6. Xuetao Wei, Lorenzo Gomez, Iulian Neamtiu and Michalis Faloutsos,“ProfileDroid: Multi-layer Profiling of Android Applications”, MobiCom’12, Istanbul, Turkey August 22–26, 2012.
7. Michael Spreitzenbarth,Felix Freiling, Michael Spreitzenbarth,Felix Freiling and Johannes Hoffmann,“Mobile-Sandbox:Having a Deeper Look into Android Applications”,18-22, 2013.
8. Lok Kwong Yanand Heng Yin“DroidScope:Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis”
9. Min Zheng, Mingshen Sun and John C.S. Lui,” DroidAnalytics: A Signature Based Analytic System to Collect, Extract and Analyze” , 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications2013.
10. Iker Burguera and Urko Zurutuza and Simin Nadjm-Tehrani,“Crowdroid: Behavior-Based Malware Detection Systemfor Android”, SPSM’11, Chicago, Illinois, USA, October 17, 2011.
11. Juanru Li, Dawu GU and Yuhao Luo, “Android Malware Forensics: Reconstruction ofMalicious Events”, 32nd International Conference on Distributed Computing Systems Workshops,2012.
12. Zarni Aung and Win Zaw, “Permission-Based Android Malware Detection”, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 2, ISSUE 3, MARCH 2013
13. Parvez Faruki, Vijay Ganmoor and Vijay Laxmi,“AndroSimilar: Robust Statistical Feature Signature for Android Malware Detection”, SIN ’13, Astray, Turkey November 26-28, 2013.

Author

1. **Tarang Kumar Barsiya** received his B.E. degree in Computer Science and Engineering fromIES IPS Academy Indore. He is currently pursuing M.Tech (Information Security) in Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India.

2. **Dr. ManasiGyanchandani** holds a Ph.D in Computer Science and Engineering from Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India. She has more than 20 years of teaching experience. She is currently working as Assistant Professor in the Department of Computer Science and Engineering in Maulana Azad National Institute of Technology. Her research area includes domains of Information retrieval, Artificial Intelligence and digital image processing. She is lifetime member of ISTE.

3. **Dr. Rajesh Wadhvani** holds a Ph.D in Computer Science and Engineering from Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India. He has more than 12 years of teaching experience and has guided more than 12 M.Tech. scholars. He is currently working as Assistant Professor in the Department of Computer Science and Engineering in Maulana Azad National Institute of Technology. His research area includes domains of Information retrieval, data mining.