A comparative study of static, dynamic and hybrid analysis techniques for android malware detection

[1]Prof. Vidhya Rao, [2]Khushboo Hande
[1]Assistant Professor, [2]MCA, SEM VI
[1,2]SIES College of Management Studies, Shri Chandrasekarendra Saraswathi Vidyapuram,
Plot 1-E, Sector V, Nerul, Navi Mumbai - 400706

Abstract— **With the popularity and increase in the number of smartphone users, the spread of mobile malware on Android platform has increased. Current intelligent terminal based on the Android has occupied most of the market, and the number of malware aiming at Android platform is also increasing with the increase in the smartphone users. The popularity of the smartphones, the large market share of android and the openness of the android market make android more sensitive platform for malware attacks. From a scientific point of view for understanding the threat to security and privacy, it is important for an analyst to analyze the behavior of the malicious application. Since a single approach may not be enough for detecting the malware against the advanced techniques, multiple approaches can be used for effective malware detection. This paper emphasizes on different types of android malware analysis techniques such as static analysis, dynamic analysis and hybrid analysis (combination of static and dynamic analysis). This paper also includes different approaches of these analysis techniques along with their functionality used for malware detection and a comparative study between these three types of analysis is highlighted. In this research, the effectiveness of hybrid analysis is also analyzed in comparison with static and dynamic analysis.**

Keywords—**Android malware, Static analysis, Dynamic analysis, Hybrid analysis**

I. INTRODUCTION

Android is the most popular operating system for mobile computing devices such as smartphones and tablets. Mobile devices are being widely used and contain sensitive user information, therefore malware's are being developed for stealing such information. According to an International Data Corporation (IDC) [1] analysis report on the market share of smartphone operating systems, the Android operating system makes up 78% of smartphone operating systems, up to the first quarter of 2015. Symantec report also indicates that the amount of malware targeting Android operating systems has increased significantly in recent years [2]. This large market for smartphones draws the attention of cyber criminals, such cyber criminals develop malicious software which is often designed to gain access to information within a smartphone. Android has various third party application stores which make it easy for cybercriminals to repackage Android applications with malicious payloads. The share of Android smartphone has grown exponentially in the smartphone market due to its open source characteristic and is more likely to be compromised than other systems. Therefore the analysis and detection of Android malware have become an important research area. CNCERT has detected 702, 861 mobile internet malicious sample programs in 2013, among which 99.5 percent aims at Android platform [3]. These malicious programs not only affect smartphone users' normal use but also have security threats such as malicious fee deduction, stealing information and remote control, which bring loss to smartphone users. To detect malicious applications we need to use analysis techniques. In this paper, we focus on 3 types of analysis techniques for android malware detection i.e. static, dynamic and hybrid analysis. In each type of analysis, we have different types of methods and tools, but we are focusing only on some important methods and tools of these analysis techniques. This paper also includes the comparative study of static, dynamic and hybrid analysis techniques and its effectiveness in detecting the malicious application respectively.

II. LITERATURE REVIEW

To analyze the mobile malicious application, there are two ways: static analysis and dynamic analysis. Static analysis is a way to recognize malicious permissions, methods, strings, and so on. The user, namely Inspector, may disassemble the application and find the malicious position of source code and manifest file. Dynamic analysis is a way to recognized malicious behaviors during runtime. It could be tested in a virtualized environment to be run and monitored. Static analysis can get and analyze all source code and is faster than dynamic analysis. But it is not appropriate for the suspicious malicious application which is concealed or obfuscated. Dynamic analysis can recognize malicious behavior which is executed only on targeted condition. But, it cannot cover malicious code that is not executed during runtime. Because of the massive increase in Android malware, the research community has suggested several security solutions, covering from static or dynamic analysis of applications [4].Static analysis is extensively used in desktop computers and Android phones, and the network attackers have developed obfuscation techniques to bypass static analysis [5]. Leonid Batyuk et al.[6] proposes a service that accesses android market applications via static analysis and provide detail reports to the user and then they describe a means to lessen security and privacy threats by applying automated reverse engineering techniques.

Static analysis is based on extracting features by inspecting an application's manifest and the disassembled code, it does not involve execution. On the other hand, dynamic analysis methods monitor and trace the application's behavior during its execution [7]. Hybrid methods integrate run-time data extracted from dynamic analysis into a static analysis algorithm to detect behavior or malicious functionality in the applications [8]. In the hybrid analysis, a combination of static and dynamic analysis features or information obtained from the static and dynamic analysis is used to detect malicious behavior. The Static analysis is usually lightweight and can be performed on a user's device while dynamic analysis is usually performed in an offline emulator due to simulation overhead. Static and dynamic analysis both have their advantages as well as disadvantages. Static analysis techniques can be defeated by malware packing and other malware obfuscation techniques. On the other hand, dynamic analysis techniques can be defeated if the malware notices it is running in an emulator or sandboxed environment [9]. The dynamic analysis consumes more resources and time as compared to static analysis but allows dynamic loading. In static analysis the malware is examined without actual execution whereas dynamic analysis executes malware in a monitored environment to observe its behavior and provides comparatively more information than static analysis and the result is more effective. Therefore dynamic analysis is less prone to code obfuscation [10].

III. OBJECTIVE OF STUDY

To study static, dynamic and hybrid analysis techniques for android malware detection using software tools, compare and determine the best technique.

IV. ANALYSIS & INTERPRETATION

To study static, dynamic and hybrid analysis techniques for android malware detection using software tools, compare and determine the best technique.

A. Static Analysis

In the static analysis, the analysis of the applications is done and the features are extracted without executing the application on an emulator or device. In comparison to other analysis techniques for android malware detection, static analysis consumes fewer resources and time as it does not involves execution of the application. The major disadvantage of this analysis is code obfuscation because of which detecting the malicious behavior of the application becomes difficult as pattern matching is not possible. This analysis can detect runtime errors, logical inconsistencies, and possible security violations. The most commonly used static features are the Permission and API calls.

Static analysis approach:

1) Signature-Based Approach

Signature based malware detection methods are mostly used by commercial anti-malware products. This method extracts the semantic patterns and creates a unique signature. A program is classified as a malware if its signature matches with existing malware signatures. Although signature based detection is very efficient for known malware, the major drawback is that it cannot detect the unknown malware types. Also because of the limited signature database, most of the malware remain undetected. The malware variants need to be immediately updated as detected.

2) Permission Based Analysis

For governing the access rights of any application, the permissions requested by the application plays an important role. When we install any application, by default it does not have any access to the data stored in the device and does not have any effect on system security. The permissions requested for the resources are in the AndroidManifest.xml file. While installing any application we have to allow the application to access all the resources requested by the application but all declared permissions are not necessarily the required permissions for that specific application. The drawback of this approach is that it only analyzes the manifest file and no other files.

B. Dynamic Analysis

Dynamic analysis is the testing and evaluation of a program by executing data in real-time. The objective is to find errors in a program while it is running, rather than by repeatedly examining the code offline. It is a detection technique which aims at evaluating malware by executing the application and the main advantage of this technique is that determines the application behavior during runtime and loads target data. The resource consumption in this analysis technique is more as compared to static analysis. Dynamic behavioral detection method constructs operation environment by using a sandbox, virtual machine, and other forms, and simulates the execution of the application to acquire the application's behavior model.

Dynamic analysis approach:

1) Anomaly-Based Detection

This technique uses machine learning algorithms to detect the malicious behavior of the applications. Features from the existing malware are used to train a model for an unknown malware. The tools which use this detection method provides deep analysis and thus require a lot of resources. To detect the malicious application, it is required for the application to be installed on the users' device. The drawback of this approach is that it may even classify the legitimate application as malware if it invokes more system calls.

***2) Taint Analysis*:**

The taint analysis is a popular method which checks which variables can be modified by the user input as user input can be dangerous if they aren't properly checked. The system TaintDroid which follows this approach uses a scientific technique called "dynamic taint analysis". This technique marks information of interest with an identifier called a "taint." That taint stays with the information when it is used. The tracking system then monitors the movement of tainted information. TaintDroid which provides system-wide information flow tracking for Android. It can simultaneously track multiple sources of sensitive data such as camera, GPS, and microphone etc. and identify the data leakage in third-party developer applications. TaintDroid can tag sensitive information or data automatically, as long as the tagged information leaves the system through any channel, it is recorded by TaintDroid. The application executing data sending is marked, and this function can identify suspicious applications. The drawback of this analysis method is that it cannot track information that leaves the channel and returns a network reply.

C. Hybrid Analysis

Hybrid Analysis is a combination of static and dynamic analysis. It is a technology or method that can integrate run-time data extracted from dynamic analysis into a static analysis algorithm to detect behavior or malicious functionality in the applications. The hybrid analysis method involves combining static features obtained while analyzing the application and dynamic features and information extracted while the application is executed. Though it could increase the accuracy of the detection rate, it makes the system cumbersome and the analysis process is time consuming.

Tools with hybrid analysis approach:

1) Mobile Sandbox:

Mobile sandbox is a combination of static and dynamic analysis. In a mobile sandbox, we use analysis of APK file for static analysis. In this, the anti-viruses, user permissions, parse the manifest.xml file for identifying the suspicious code are scanned. In dynamic analysis, they use an emulator for running the suspicious application and check the behavior of the application. The network traffic is also checked along with the native calls for better understanding the behavior of the application. Working of Mobile Sandbox is as shown is Fig.1. The sandbox is a security mechanism for separating running programs. It is often used to execute untested or untrusted programs or code, possibly from unverified or untrusted third parties, suppliers, users or websites, without risking harm to the host machine or operating system.

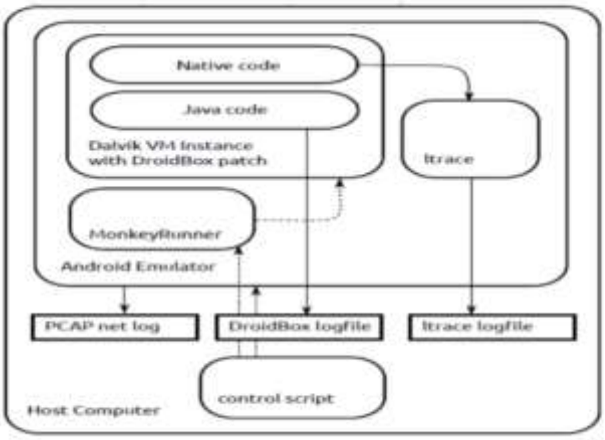


Figure 1: Mobile sandbox working

2) Andrubis

In Andrubis framework, for dynamic analysis result of static analysis is used so first we perform static analysis and then dynamic analysis which gives more effective results. In static analysis, this framework is concentrated of an android manifest.xml file and byte code. All the information which comes from static analysis is used 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 is shown in Fig2.

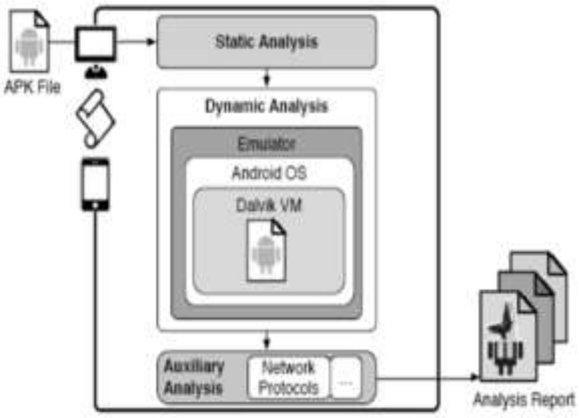
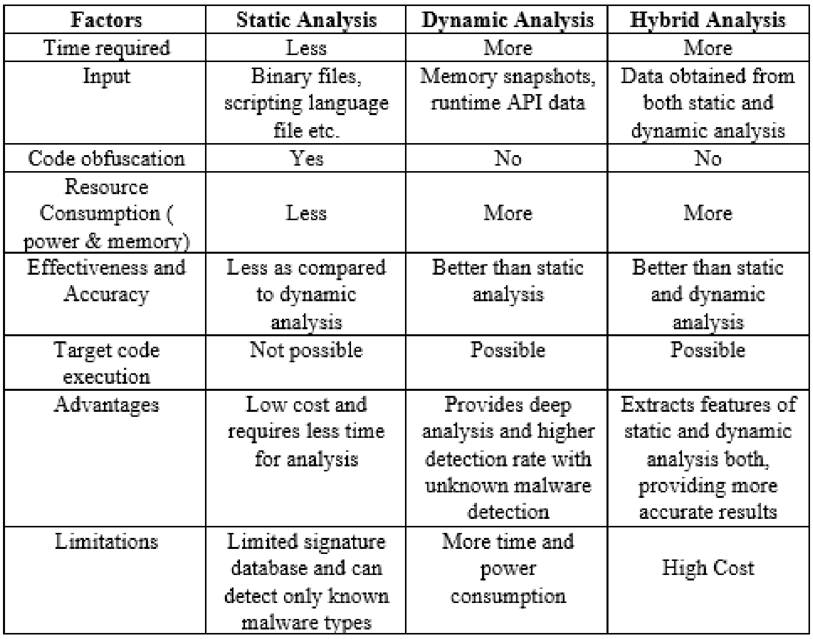


Figure 2: Andrubis framework

Comparison

Table 1: Comparison between static, dynamic and hybrid analysis



Factors	Static Analysis	Dynamic Analysis	Hybrid Analysis
Time required	Less	More	More
Input	Binary files, scripting language file etc.	Memory snapshots, runtime API data	Data obtained from both static and dynamic analysis
Code obfuscation	Yes	No	No
Resource Consumption (power & memory)	Less	More	More
Effectiveness and Accuracy	Less as compared to dynamic analysis	Better than static analysis	Better than static and dynamic analysis
Target code execution	Not possible	Possible	Possible
Advantages	Low cost and requires less time for analysis	Provides deep analysis and higher detection rate with unknown malware detection	Extracts features of static and dynamic analysis both, providing more accurate results
Limitations	Limited signature database and can detect only known malware types	More time and power consumption	High Cost

From analyzing the above table, it is observed that, hybrid analysis technique for android malware detection is more effective because of its dual nature of analysis.

V. CONCLUSION AND FUTURE WORK

Malware detection techniques that use either static detection techniques that can be easily obfuscated or those that use only dynamic detection techniques also do not provide the complete solution. These techniques are combined to overcome the drawbacks in detecting malicious applications. From the study it is concluded that, hybrid analysis, which is a combination of both static and dynamic analysis is more effective and provides more accurate results as compared to static and dynamic analysis individually. Malware detection techniques and tools should be improved as the malware families are increasing with the increase in the number of smartphone user. The hybrid analysis is becoming popular because it produces more accurate results in detecting malware applications. In future work a detailed study with the tools of android malware detection can be done. Smartphone users need to be aware enough so they read and understand the permissions requested by the application before agreeing to grant access.

REFERENCES

[1] IDC Smartphone OS Market Share, Q1 2015, http://www.idc.com/prodserv/smartphone-os-market-share

[2] Symantec, Security Response: Mobile Adware and Malware Analysis, version 1.0, Oct. 2013.

[3]"CNCERT/CC.CNCERT/CC Annual Report",(2013), http://www.cert.org.cn/publish/main/upload/File/2013 Annual Report.pdf.2014:53-56

[4] A. Reina, A. Fattori, and L. Cavallaro, "A system call-centric analysis and stimulation technique to automatically reconstruct android malware behaviors," EuroSec, April, 2013.

[5] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in the Proc. of Annual Computer Security Applications Conference", 2007.

[6] Leonid Batyuk, Markus Herpich, Seyit Ahmet Camtepe, Karsten Raddatz, Aubrey-Derrick Schmidt, and Sahin Albayrak "Using Static Analysis for Automatic Assessment and Mitigation of Unwanted and Malicious Activities Within Android Applications" in Malicious and Unwanted Software (MALWARE), 2011, 6th International Conference

[7] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck, "Drebin: Effective and explainable detection of android malware in your pocket." in Proceedings of the Network and Distributed System Security Symposium (NDSS), 2014.

[8] S. Zhao, X. Li, G. Xu, L. Zhang, and Z. Feng, "Attack tree based android malware detection with hybrid analysis," in Proceedings of the IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2014.

[9] V. Rastogi, Y. Chen, and X. Jiang, "Catch me if you can: Evaluating android anti-malware against transformation attacks," IEEE Transactions on Information Forensics and Security, vol. 9, no. 1, pp. 99–108, Jan 2014.

[10] Ankita Kapratwar , "Static and Dynamic Analysis for Android Malware Detection", San Jose State University, May2016.