ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (319-322), Month: April - June 2015, Available at: www.researchpublish.com

Similarity among Android Applications by GUI Feature Extraction

[1]Amit Kumar, [2]Vishal Verma

Department Of Computer Science & Engineering,Mahatama Gandhi Mission's Engineering College, Noida, India.

Abstract: **Code analysis and Malwares detection for Android applications are considered as a serious problem. There are many researches to apply new and creative techniques that can detect Malwares with low computational cost. These researches are being grown because of wide use of new applications. This thesis tries to attack this problem by presenting a new and creative method to analyze static code and measure similarity with available dataset of applications. Thesis immigrate its idea to a well-known and wide officially used antivirus project, which is considered as a part of famous antiviruses programs and called Androguard. We apply thesis idea in Androguard project to test its feasibility by making a comparison study between it and modified Androguard; we used Malware dataset of android applications for this research. As a result, proposed method saves about 60-70% of time with similar results and time distribution behavior in compare with original Androguard and little scarification in accuracy, this refer to simplicity of generating signatures and measuring similarity using SimHash algorithm.**

Keywords: **Androguard, similarity, applications, android applications, researches.**

1. INTRODUCTION

Smartphones are used everywhere, by everyone, for all purposes, it is the latest technology trend of the 21st century. Today's social life requires us to stay in always connection with internet by smartphones, also smartphones are being rapidly integrated into enterprises, government agencies, and even the military, In fact smartphones are used by all people around the world from all ages and for various usage. All of these are reasons for the wide development of smartphones hardware and software. Smartphones are based on several platforms; one of the most popular is Android. The popularity of Android has enabled the application marketplace to grow dramatically, the black market presence has also grown rapidly where paid applications are modified for free download and from untrusted websites or stores, smartphone user may exposed to various information security threats when he uses his phone, these threats can disrupt the operation of the smartphone, and transmit or modify the user data. For these reasons, Android applications must guarantee privacy and integrity of the information they handle.

There are several countermeasures and researches to guarantee privacy and integrity of apps by detecting and preventing Malware threat in mobile devices some of these are signature based antivirus scanners which efficiently detect known Malwares, others depends on detection and classification method in which they classify source code to detect Malware , even if it has no background of mobile applications. These countermeasures and researches are different in their accuracy and mobile resources consumption and there are a lot of researches which try to solve these problems.

2. ANDROID SYSTEM ARCHITECTURE

Android is a new open source mobile platform that was designed by Google .Android applications utilized advanced hardware and software to bring benefits and value to its users.

The architecture of Android is implemented as a software stack, customized for mobile devices. Figure 1 shows some of the most important components of this stack. The core of the Android platform is a Linux kernel. The kernel's responsible for handling device drivers, resource access, memory process, power management and other typical OS duties. The kernel also acts as an abstraction layer between the hardware and the software stack.

On the top of the kernel there are several native C/C++ libraries. Most of the application framework access these core libraries through the DVM. This access is based on Java APIs that are thin wrapper classes around the native code using the Java Native Interface.

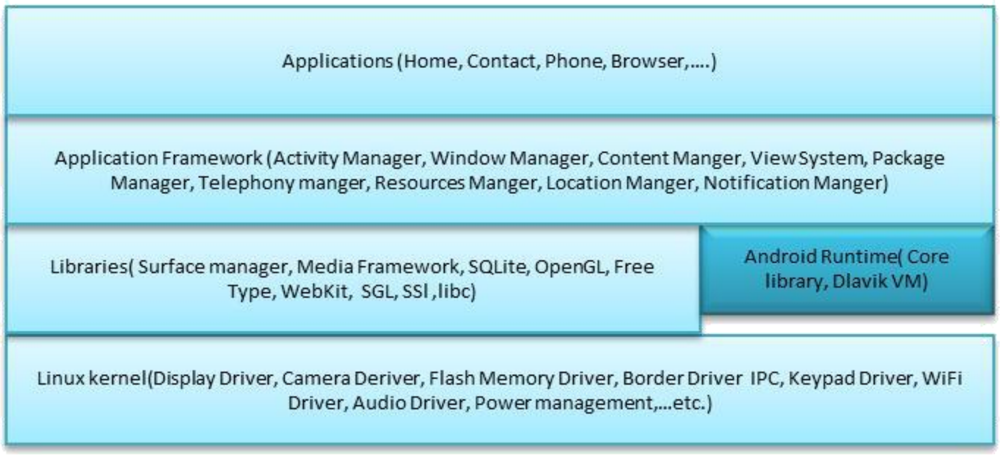

Applications (Home, Contact, Phone, Browser,....)
Application Framework (Activity Manager, Window Manager, Content Manger, View System, Package Manager, Telephony manger, Resources Manger, Location Manger, Notification Manger)
Libraries(Surface manager, Media Framework, SQLite, OpenGL, Free Type, WebKit, SGL, SSI ,libc)
Android Runtime(Core library, Dlavik VM)
Linux kernel(Display Driver, Camera Deriver, Flash Memory Driver, Border Driver IPC, Keypad Driver, WiFi Driver, Audio Driver, Power management,...etc.)

Fig.1 Android OS Architecture

3. ANDROID MALWARES

Malicious software is referred to as Malware, classified by its nature as either computer virus Trojan horse, worm, backdoor or rootkit. the most common Malware types are :

- **Virus:** Code that that inserts itself into another program and replicates, that is, copies itself and infects other computers. Nowadays often used as a generic term that also includes worms and Trojan horses.
- **Worm:** Self-replicating Malware which copies itself to other nodes in a network without user interaction using vulnerabilities. Worms do not attach themselves to an application like a virus do.
- **Trojan horse:** Malicious program which masquerades itself as being an application. Unlike viruses and worms, it does not replicate itself.
- **Rootkit:** Software that enables continued privileged access to a computer while actively hiding its malicious activity from administrators by modifying the operating system functionality.
- **Backdoor:** Specialized Trojan horse that masquerades itself as an installed program to enable remote access to a system and bypassing normal authentication.
- **Spyware:** Software that reveals private information about the user or computer system to eavesdroppers.
- **Bot:** Piece of Malware that allows the bot master, i.e. the author to remotely the infected system. A group of infected systems that are controlled are denoted as botnets, instructed by the bot master to perform various malicious activity such as distributed denial of services, stealing private information and sending spam.

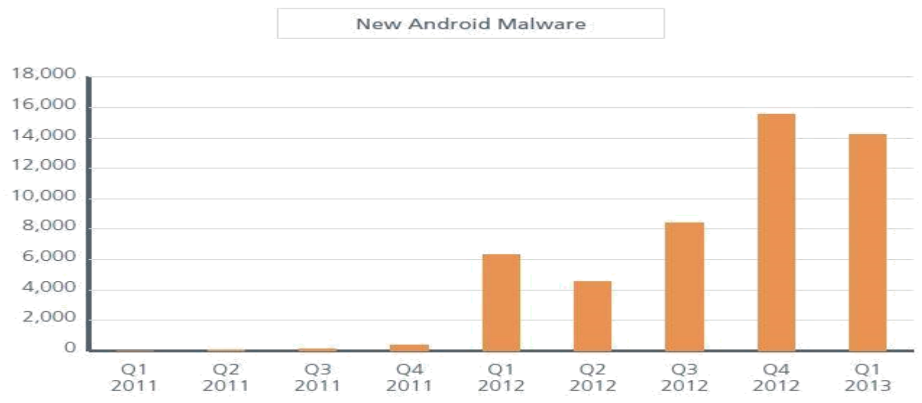


Fig.2 New Android Malwares

4. ANDROGUARD

Androguard is mainly a python tool done by Virus Total project, Virus Total is a subsidiary of Google, is a free online service that analyzes files and URLs enabling the identification of viruses, worms, Trojan horse and other kinds of malicious content detected by antivirus engines and website scanners. At the same time, it may be used as a means to detect false positives, i.e. innocuous resources detected as malicious by one or more scanners. Virus Total mission is to help in improving the antivirus and security industry and make the internet a safer place through the development of free tools and services.

Androguard play mainly with:

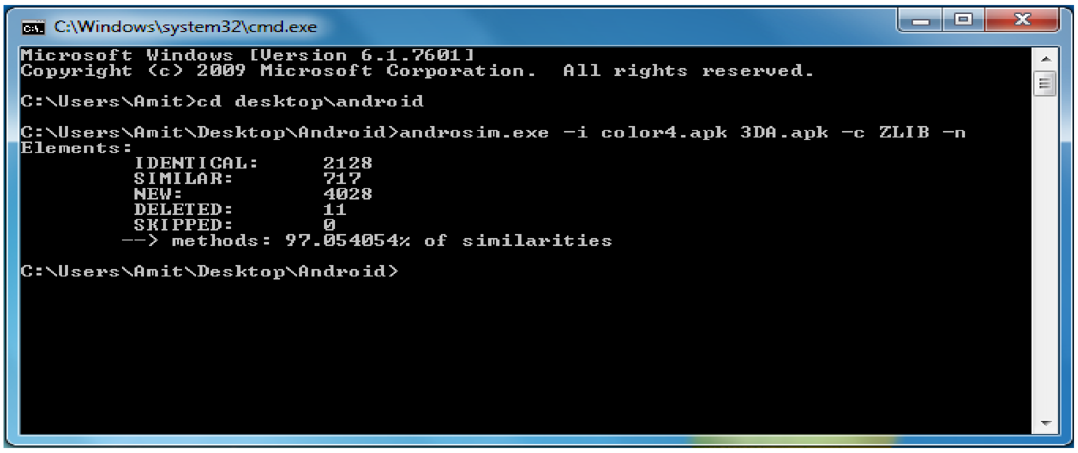
- Dex/Odex DVM , .dex disassemble, Decompilation .
- APK Android application .
- Android's binary xml.
- Android Resources.

Androguard has the following features **:**

- Map and manipulate DEX/APK format into full Python objects.
- Disassemble/Decompilation/Modification of DEX/APK format.
- Decompilation with the first native directly from dalvik byte codes to java source codes dalvik decompiler.
- Access to the static analysis of the code basic blocks, instructions, permissions and create analysis tool.
- Analysis a bunch of Android apps.
- Diffing of Android applications.
- Check if an Android application is present in a database.
- Open source database of Android Malware
- Reverse engineering of applications
- Transform Android's binary xml like AndroidManifest.xml into classic xml.

The most important feature that has been used in thesis is similarity measurement which will be used later to detect Malwares after measuring distance between it and given clustered datasets of Malwares families. Androguard uses Elsim project to measure distance between two text codes of different Android applications, Elsim has an important tool which used to measure similarity, this tool is called Androsim , This tool detects and reports identical methods, similar methods, deleted methods, new methods and skipped methods.

Moreover, a similarity between 0.0 to 100.0 is calculated upon the values of the identical methods and the similar methods. Androguard calculate the final values using text compressor. It is more interesting because an understandable value related to the similarity will be discovered.

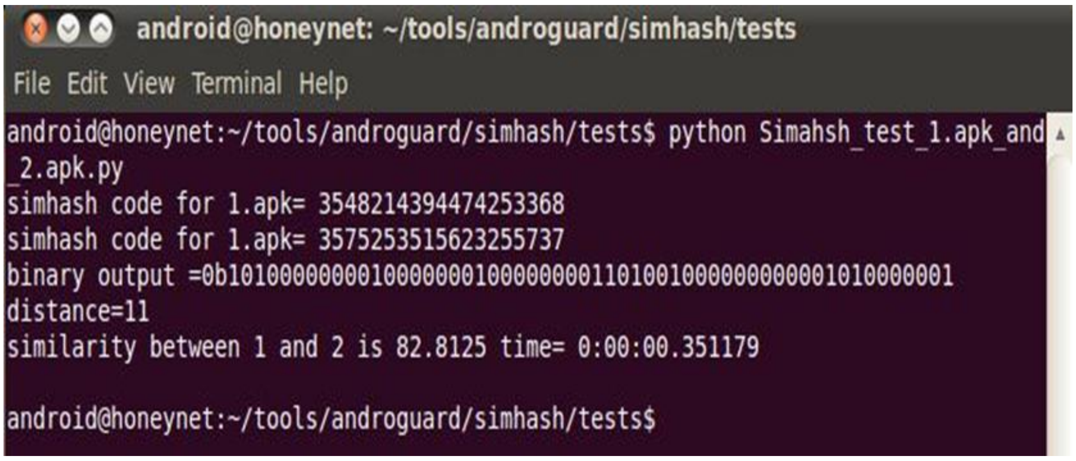

C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Amit>cd desktop\android

C:\Users\Amit\Desktop\Android>androsim.exe -i color4.apk 3DA.apk -c ZLIB -n
Elements:
         IDENTICAL:      2128
         SIMILAR:        717
         NEW:            4028
         DELETED:        11
         SKIPPED:        0
        --> methods: 97.054054% of similarities

C:\Users\Amit\Desktop\Android>
```

Fig.3 Andoguard to find similarity between two android applications

5. SIMHASH ALGORITHM

Measuring similarity between two strings has many algorithms one of them is SimHash algorithm which has been created by Moses Charikar from Google. It can compare easily between two strings, and by the way two datasets of any type, quickly and effectively.


android@honeynet: ~/tools/androguard/simhash/tests
File Edit View Terminal Help

```
android@honeynet:~/tools/androguard/simhash/tests$ python Simahsh_test_1.apk_and
_2.apk.py
simhash code for 1.apk= 3548214394474253368
simhash code for 1.apk= 3575253515623255737
binary output =0b1010000000001000000010000000011010010000000000001010000001
distance=11
similarity between 1 and 2 is 82.8125 time= 0:00:00.351179

android@honeynet:~/tools/androguard/simhash/tests$
```

Fig. 4

SimHash use a fingerprint system, instead of comparing the texts directly, it will compare their fingerprints, which is really more effective and usable in reality.

6. CONCLUSION

In this paper, In this shows the used methodology for thesis, First we describe how we ran Androguard project and how we modify it to use SimHash algorithm as build in tool, then we prove the feasibility of idea by experiments. The first experiment compare between Androguard and thesis model by applying two of them on a Malware data set, the results prove that proposed method is faster than Androguard with similar results and time distribution. Finally the second experiment detected a Malware application and compare detection evidence with Androguard.

REFERENCES

[1] Android. (n.d.). Activities. Retrieved 1 23, 2014, from Android Developers: http://developer.android.com/guide/components/activities.html

[2] Android. (n.d.). Application Fundamentals. Retrieved 1 28, 2014, from Android Developers.

[3] Android. (n.d.). Introduction to Android. Retrieved 1 23, 2014, from Android Developers: http://developer. android. com/guide/index.html

[4] APKInspector. (n.d.). Retrieved 6 3, 2014, from APKInspector: https://code.google.com/p/apkinspector/