Intl J Engg Sci Adv Research 2016 March; 2(1):70-72 ISSN NO: 2395-0730

Evaluating and Recognizing Mechanism of Android Malware through Dismantling and Visualization

Sanjay Sharma
Research Scholar
Computer Science & EngineeringNational Institute for TechnicalTeacher Training & Research , Chandigarh
sanjay.cse@nitttrchd.ac.in

Anubhav Bewerwal
Research Scholar
Software EngineeringRajiv Gandhi Proudyogiki Vishvavidyalay, Bhopal
bewerwalanu@gmail.com

Arvind Lal
Research Scholar
Computer Science & Engineering
National Institute for Technical Teacher Training & Research, Chandigarh
arvind.cse@nitttrchd.ac.in

Abstract-**It is essential requirement to enroot evaluation and recognizing quick fix in current scenario of advancement. Certainly many of the safeguards are contributing in narrow consideration of mobile malwares and its cultivated evaluation. As Android is a prominent medium to put forth the process of evaluation technique of Malwares resembling to its actual malware families, our target is to do an observable comparison among prevailing android malwares and making a significant decision on their intensity of alikeness in this paper which overall contributes in dissemination of it to its investigator relevantly.**

Keywords— Android malware, malware analysis, Smartphonesecurity.

I. INTRODUCTION

By rapid growth of the speed of wired and wireless networks and sharp increase of smartphone adoption, the number of financial transactions with the smartphone has increased in recent years. Unfortunately, this comes with evolution of mobile malware, especially financial fraud that utilizes the vulnerability of smartphone and installs malwares on your smartphone. They then flow out personal information to use the premium SMS services and micro-payments on mobile phones. For the advantages that can be obtained from the smartphone infected with malware, the number of mobile malicious code is rapidly increasing. Specially, many mobile malwares are found that are repackaged the legitimate applications with malicious codes[1].

In this paper, we propose a method of analyzing and deciding malware on the basis of similarity with existing malware families on the popular platform, Android. In particular, we express in visualization the suspected Android application and compare with malware families. According to the degree of similarity, it helps them distribute to inspector discriminately.

II. RELATED WORKS

To solve the mobile malware, similar initiatives have emerged on the Android platform. Zhou *et al.* find that about 86.0% malwares are repackaged version of legitimate application with malicious codes. Also they characterized of existing Android malware, ranging from their installation, activation, to the carried malicious payloads. In this paper, although they classified as above, we find that malwares are repacked in diverse ways, that is hackers disassemble existing malwares or legitimate apps, enclose malicious payloads, and then re-assemble and submit the new apps or update apps to official and/or third party Android market[1].

M. Cho *et al*. proposed AndroScope, a performance analysis tool for the Android platform and provides a trace mechanism for tracing not only the Android applications but also all software layers of the Android platform that is, Dalvik VM, core libraries, Android libraries, and even Linux kernels[2]. AndroScope provided advanced Traceview[3] which is graphical viewer to load the trace file and display such basic information as the enter and exit times of each method.

J. Ko et al. proposed techniques to determine similarity of Android application via reversing and k-gram birthmarking. Although they developed the system to identify software reuse illegally, this system decompiles the Android apps and made the birthmarks based on k-gram and determines the similarity between the sample Android apps by comparing the birthmarks[4].

III. PROPOSED APPROACHES

We propose the system which makes Android malware family and measuring similarity among Android applications. Especially, we produce CFG(Call Flow Graph) to visualize characteristics of Android application, select the representative CFG of each family and then suspected Android application check degree of similarity. According to the degree of similarity, malwares are distributed to inspectors discriminately.

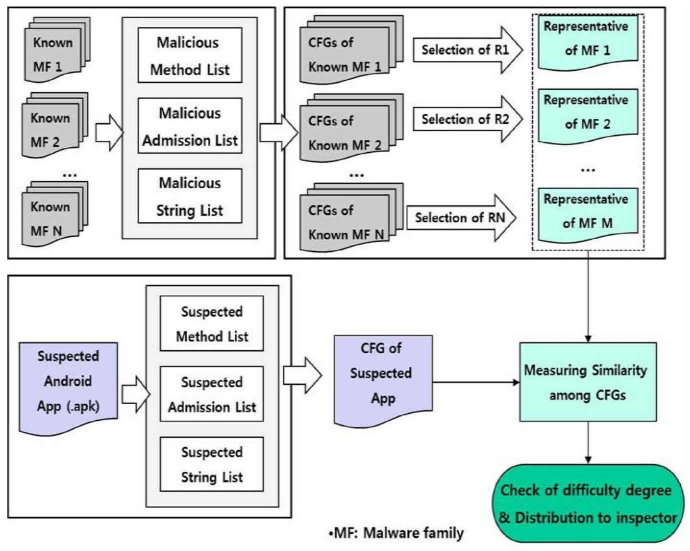

Figure 1 : Procedures of the system

Figure 1 illustrates our approach to visualize of Android malware, measure similarity with the existing malware families and then distribute to inspector according to level of difficulty appropriately. Our system consists of 4 sub-systems: system of making CFGs of known malware families, system of selection of Representative of malware families, system of making CFG of Suspected Android application, and system of Measuring similarity and Distribution malware to inspector.

The first step is to make CFGs of all known and collected malwares. To make them, system disassembles all malwares and removes identifiers in order to increase the accuracy. And then, system extracts and lists malicious methods, needless permissions and malicious strings. CFGs are made by analyzing call among methods, relevant threads, life-cycle of Activities. Additionally, the system abstracts the call graph information not only to decrease the computation but to increase the accuracy by removing insignificant calls. The second step is to select a representative in each malware family. Each representative is selected the ones with the highest degree of similarity among malware family members. At this time, methods of calculating of similarity as are Isomorphism, Edit distance, Maximum common sub-graph, Statistical similarity, Node and edge matching methods, and so on [5]. The Next step is to make CFG of suspected Android application. This is almost the same as making CFGs of known malware families. However, instead of malicious methods, permissions and strings, suspected them is used for making CFG. Finally, the system compares suspected malware CFG with representatives of malware family CFGs and produces the degree of difficulty to analyze a suspected application. Especially, it is important to measure the degree of difficulty. According to the degree of difficulty, malwares are distributed to inspectors discriminately. We propose the rules for measuring difficulty are as follows:

- Inverse proportion to the number of application in each known malware family.
- Proportion to the appearance frequency within the limited time of the last in each known malware family.
- Inverse proportion to the number of application in a known malware family.
- Proportion to whether to apply the code obfuscation.
- Inverse proportion to the degree of similarity among the known malware families.

Figure 2 illustrates flow chart of our system to detect whether malware. First of all, suspected A ndroid application should be disassembled to know if malware is not. And then, prior to making the CFG the system checks suspected permission list, for example SEND_SMS, READ_SMS, WRITE_SMS, RECEIVE_SMS,READ_PHONE_STATE,KILL_BACKGROU D_PROCESSES and so on. If the application is suspected, the system progress to make a CFG. To make the CFG, methods, relevant threads, activities and suspected strings are extracted. Also system may remove identifiers in order to increase the accuracy. After making CFG, it measures similarity with existing malware family and analysis difficulty. If difficulty is the highest thing, the malware is placed in the highest expert. However, the malware is the latest fashion and difficulty is low relatively, the malware is sent to beginner. After an inspector finishes analyzing and detecting Android application, it is joined malware family or passed. If it is malware, the family is updated and selects a new representative optionally.

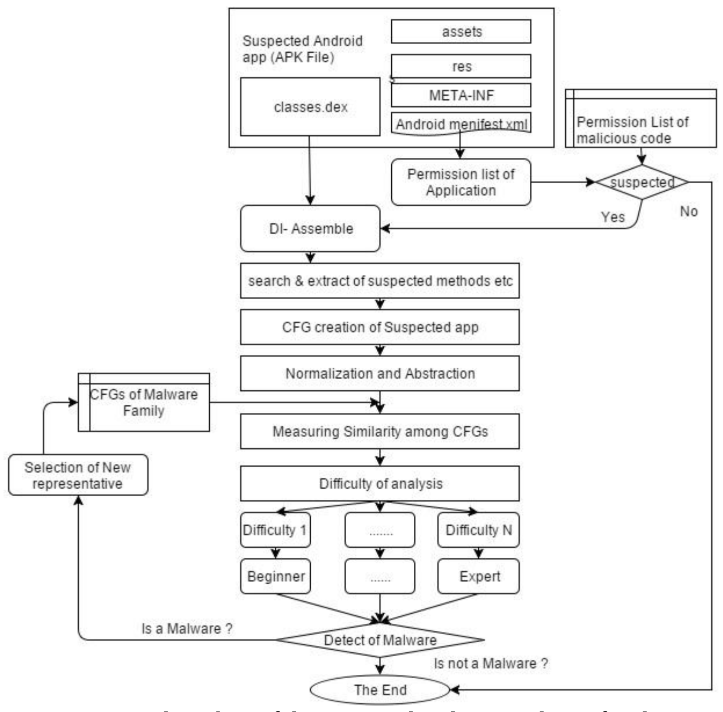
Suspected Android app (APK File)
assets
res
META-INF
classes.dex
Android menifest.xml
Permission List of malicious code
Permission list of Application
suspected
Yes
No
DI- Assemble
search & extract of suspected methods etc
CFG creation of Suspected app
Normalization and Abstraction
CFGs of Malware Family
Measuring Similarity among CFGs
Selection of New representative
Difficulty of analysis
Difficulty 1
....
Difficulty N
Beginner
......
Expert
Is a Malware ?
Detect of Malware
Is not a Malware ?
The End

Figure2. Flow chart of detecting and updating malware family

IV. CONCLUSION AND FUTURE WORK

This work presents the method of detecting an Android malware with visualization of application and measuring similarity among known malware families. Also, we proposed that malware is distributed to various inspectors discriminately according to degree of difficulty to analyze. In particular, we suggested the rules for measuring difficulty. In the future, we plan to develop this system and apply in real-world.

REFERNCES

[1] Yajin Zhou, Xuxian Jiang, "Dissecting Android Malware : Characterization and Evolution" Proceeding 33rd IEEE Symposium Security and Privacy, 2012

[2] AndroScope: An insightful performance analyzer for all software layers of the android-based systems, ETRI Journal, 2013

[3] Traceview, "http://developer.android.com/tools/help/traceview.html"

[4] J. Ko, et al. "Measuring Similarity of Android Applications via Reversing and K-gram Birthmarking", RACS'13 pp.336-341, Oct, 2013

[5] L. Zager, "Graph similarity amd matching", MS Thesis, EECS, MIT, 2005.