Faculty of Information and Communication Technology

ENHANCEMENT OF STATIC CODE ANALYSIS MALWARE DETECTION FRAMEWORK FOR ANDROID CATEGORY-BASED APPLICATION

Azmi Bin Aminordin

Doctor of Philosophy

2021

ENHANCEMENT OF STATIC CODE ANALYSIS MALWARE DETECTION FRAMEWORK FOR ANDROID CATEGORY-BASED APPLICATION

AZMI BIN AMINORDIN

A thesis submitted
in fulfilment of the requirements for the degree of Doctor of Philosophy

Faculty of Information and Communication Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I declare this thesis entitled "Enhancement of Static Code Analysis Malware Detection Framework for Android Category-Based Application" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	…………………………..
Name	:	…………………………..
Date	:	…………………………..

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Doctor of Philosophy.

Signature	:	……………………………………………..…
Supervisor Name	:	……………………………………………….
Date	:	……………………………………………….

DEDICATION

This thesis is dedicated to Almarhumah Zainaf Bte Maidin, family, supervisors and all my friends which without whom none of my success would be possible.

ABSTRACT

Android has become the number one mobile operating system in term of worldwide market share since May 2012. The highest demand and the open source factors had brought Android operating system into main target of malware creator. Two approaches introduced to detect malware in Android mobile environment namely static analysis and dynamic analysis. Static analysis is where the static features are examined. Too many features used, features extraction time consuming and the reliability of accuracy result by various machine learning algorithm are the main issues spotted in static analysis approach. As such, this thesis investigates the whole Android static analysis framework in detecting and classifying mobile malware. The early study found that two static features that are often used (permission and API calls) with the right mapping are sufficient to analyse the Android malware. The new permission(s) toward API call(s) mapping for Android level 16 to 24 is constructed based on Android official developer guideline references where previously these two features are mapped without using the standard guideline. On experimenting and analysing the framework, there are 4767 benign applications from 10 different categories was collected from Android official market place and 3443 malware applications was collected from AndroZoo dataset. All benign files are then scanned through VirusTotal to ensure that all collected files are free from virus. On extracting the desired features, a new automation of feature extraction using Depth First Search (DFS) with sequential search are introduced and succeed to extract the targeted features with consideration of no limitation on application file size also no limitation on file number. In order to enables machine learning to train faster and reduces the complexity of a machine learning model, the information gain features selection is applied towards the extracted features. Four types of machine learning algorithm were tested with four different kind of splitting dataset techniques separately. The result shows that the detection of malware within application category achieves higher accuracy compared to application with non-category based. In increasing the reliability, the results obtained are then validated by using statistical analysis procedure which each machine learning classification algorithm are iterate 50 times. The validation results show that Random Forest with 10-folds cross validation spitting dataset achieved 8 highest performance compared to benchmark study and two other classifiers. This study suggests the work to combine the optimization of feature selection and algorithm parameters to achieve higher accuracy and acquire more reliable comparison.

PENINGKATAN RANGKA KERJA PENGESANAN PERISIAN ANALISA KOD STATIK BAGI APLIKASI ANDROID BERASASKAN KATEGORI

ABSTRAK

Semenjak Mei 2012, Android telah menjadi sistem pengoperasian mudah alih nombor satu dari sudut pasaran di serata dunia. Faktor permintaan yang tinggi dan konsep keterbukaan sumber telah mendorong sistem pengoperasian Android menjadi sasaran utama oleh pembangun perisian hasad. Terdapat dua pendekatan yang diperkenalkan untuk mengesan perisian hasad dalam persekitaran mudah alih Android iaitu secara analisa statik dan analisa dinamik. Analisis statik adalah di mana ciri statik dikaji. Terlalu banyak ciri yang digunakan, pengekstrakan fitur memakan masa dan kebolehpercayaan ketepatan yang dihasilkan oleh pelbagai algoritma pembelajaran mesin adalah masalah utama yang dilihat dalam pendekatan analisis statik. Tesis ini menyiasat rangka kerja bagi mengesan perisian hasad di dalam perisian mudah alih Android menerusi kaedah analisa kod statik dan mencadangkan penambahbaikkan terhadap rangka kerja analisa statik bagi perisian Android. Kebenaran yang memetakan panggilan API untuk Android tahap 16 hingga 24 telah dibangunkan berdasarkan piawai daripada pembangun rasmi Android di mana sebelum ini kedua-dua fitur tersebut dipetakan tanpa menggunakan garis panduan piawai. Selanjutnya, proses dan algoritma baru bagi mengekstrak kebenaran dan panggilan API untuk mempercepat proses pengekstrakan dibangunkan. Di dalam perlaksanaan ekperimen dan penganalisian rangka kerja, sebanyak 4767 aplikasi bersih dari 10 kategori yang berbeza dikumpulkan dari pasaran rasmi Google dan 3443 aplikasi perisian hasad dikumpulkan dari dataset AndroZoo. Semua fail bersih kemudian diimbas melalui VirusTotal untuk memastikan semua fail yang dikumpulkan bebas dari virus. Automasi pengekstrakan fitur yang diperkenalkan berjaya mengekstrak fitur-fitur yang disasarkan dengan tidak mempunyai had pada saiz fail aplikasi juga tidak mempunyai had pada bilangan fail. Empat jenis algoritma pembelajaran mesin telah diuji dengan empat jenis teknik dataset secara berasingan. Hasil kajian mendapati, pengesanan perisian hasad berasaskan kategori mencapai ketepatan yang lebih tinggi berbanding aplikasi yang tidak berasaskan kategori. Hasil yang diperolehi kemudiannya disahkan melalui prosedur analisis statistik di mana setiap algoritma pengelasan pembelajaran mesin akan berulang sebanyak 50 kali. Hasil pengesahan menunjukkan bahawa pembelajaran mesin secara Random Forest dengan 10 kali liputan pengesahan silang mengumpul sebanyak 8 prestasi tertinggi dibandingkan dengan kajian tanda aras dan tiga pengelas lain. Tesis ini menyarankan kerja lanjutan terhadap kombinasi pengoptimuman pemilihan fitur dan parameter algoritma dilaksanakan bagi mencapai ketepatan yagn lebih tinggi serta memperoleh perbandingan yang lebih baik.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Gracious and the Most Merciful, for without Him, I would never be where I am today. Not to forget, greetings and eulogy to our beloved, The Great Prophet Muhammad SAW, for without him, we'll still be living the age of darkness.

I am deeply indebted to lecturers of UTeM, especially those in FTMK, and to my supervisors: Associate Professor Dr. Hj. Mohd Faizal Abdollah and Associate Professor Dr. Hjh. Robiah Yusof for their helps, supports, interests, valuable hints, guides and patience.

I am truly and forever obliged to Syarimasni Shamsuddin, for without her endearing support and loving company, I wouldn't make this far, and with my kids that I have the greatest blessing from Him and the joy of my life. Especially, I would like to forever give my deepest and most special thanks to my parents, my parents-in-law, my siblings, whose life-long teaching enabled me to complete this work and have pushed me to go this far. Special thanks to all my colleagues at UiTM Melaka and my friends who's always supported in many ways for completing this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

AAPT	-	Android Assets Packing Tool
AMD	-	Android malware detection
APK	-	Android Package
arff	-	Attribute-relation file format
ART	-	Android Runtime
AOSP	-	Android Open Source Project
AOT	-	Ahead of Time
API	-	Application Programming Interface
CEO	-	Chief Executive Officer
CFS	-	Correlation Feature Selection
CSV	-	Comma Separated Values
DDoS	-	Distributed Denial of Service
DEX	-	Dalvik Executable
DFS	-	Depth first search
DT	-	Decision Tree
DVM	-	Dalvik Virtual Machine
FPR	-	False Positive Rate
GA	-	Genetic Algorithm
GeFS	-	Genetic Features Selection
ID	-	Identification

IDS	-	Intrusion Detection System
IG	-	Information Gain
IMEI	-	International Mobile Equipment Identity
IMSI	-	International Mobile Subscriber Identity
IRC	-	Internet Relay Chat
JNI	-	Java Native Interface
JVM	-	Java virtual machine
KPCA	-	Kernel Principal Component Analysis
KNN	-	K-Nearest Neighbour
MAPE	-	Multiple Android Package Extractor
MIB	-	Management Information Base
MLP	-	Multi-Layer Perceptron
MMS	-	Multimedia Messaging Service
OHA	-	Open Handset Alliance
OS	-	Operating system
PC	-	Personal Computer
PIN	-	Personal Identification Number
RPC	-	Remote Procedure Call
SDK	-	Software Development Kit
SMS	-	Short Message Service
SMO	-	Sequential Minimal Optimization
SPSS	-	Statistical Product and Service Solutions
SVM	-	Support vector machine
TF-IDF	-	Term Frequency Inverse Document Frequency
TPR	-	True Positive Rate

VBA	-	Visual Basic for Application
VM	-	Virtual machine
WEKA	-	Waikato Environment for Knowledge Analysis

LIST OF PUBLICATIONS

Aminordin, A., M.A., F. And Yusof, R., 2018. Android malware classification base on application category using static code analysis. *Journal of Theoretical and Applied Information Technology, v*ol. 96, no. 20, pp.6853-6863.

Aminordin, A., Faizal, M.A., Robiah, Y., Mukhlis, A. and Arif, F., 2018. Multiple android package files extractor in mining request permissions and API calls. *Journal of Advanced Manufacturing Technology (JAMT),* vol. 12 (2), pp.11-24.

Aminordin, A., Abdollah, M.F., Yusof, R. and Ahmad, R., 2017. Preliminary findings: revising developer guideline using word frequency for identifying apps miscategorization. *Proceedings of the Second International Conference on the Future of ASEAN (ICoFA), Perlis Malaysia*, vol. 2, pp.123-131.

CHAPTER 1

INTRODUCTION

Background

In year 2019, Kaspersky lab detected about 3.4 million malicious installation packages (Chebyshev, 2020) which was nearly 1.5 times fewer than the previous year. Despite a decrease in the number of detected malicious installation packages, the number of attacks on personal data grew from 40,386 in 2016 to 67,500 in 2019.

Furthermore, in second quarter of 2019 a steady rise in the number of mobile threat detected (Chebyshev et al., 2020). Even though Android Open Source Project (AOSP) is committed to secure Android smartphone OS, it is also susceptible to the social-engineering attacks (Faruki et al., 2015). Every year, Android malware threat report is published by several anti-virus companies for public access, yet, the smartphone users awareness on its potential risk is still at the lower rate (Qiao et al., 2016). Numerous attack vectors occur which compromise smartphone security (Samani and Beek, 2018). At present, Android is holding a global market share of 74.44%, hence, Android OS devices attack phenomena cannot be overstated. Information stealing and monetisation are two serious threats toward Android based smartphones which lead to financial charges for Android users.

Most mobile phone users think that their phones are safe and assume they can proceed to do the tasks that they prefer without putting them in any risk. A study by mobile application security firm (Reed, 2019) found that 70% out of 250 applications collected from Android official market place suffer from vulnerabilities that could lead to privacy leakage. Even though Google introduces permission-based guard, in which the mechanism is to restrict access of the third-party Android applications to critical resources on Android devices through their Bouncer security in 2012, the mobile security firm still asserts that millions of Android users are facing high-risk vulnerabilities. This is because, not only sensitive information can be spread illegally, but it also could give negative impact to the economy. Hence, the existing technique of malware detection is continuously being explored and revised.

A detection of malicious activities in Android mobile phone is a crucial analysis process where it can be divided into two main approaches namely static analysis, and dynamic analysis. Static analysis detection refers to examining potential security problem program without performing or running the program. In other words, the inspection is done through each programming line and text used in various files. Figure 1.1 shows a general framework in detecting Android malware which consists of four main phases namely data collection, features extraction phase, features selection phase and classification phase.

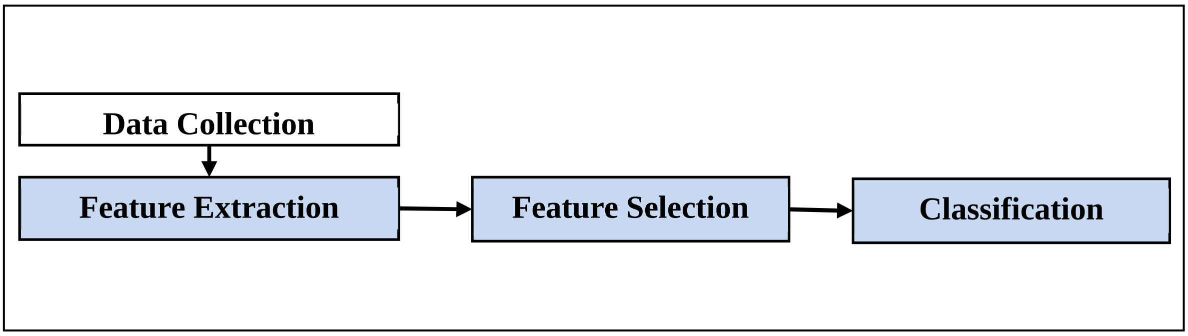

Figure 1.1: General framework of static analysis malware detection (Rana and Sung, 2018)

Static analysis is the most preferred method by many researchers (Bakour et al., 2018) because of its low computational time, ease of implementation and effectiveness in detecting malicious application. In the first phase, the dataset is prepared and processed to be in a suitable format to extract the desired features that will be used to structure the pattern that