Faculty of Information and Communication Technology

ENHANCED ANDROID MALWARE DETECTION FRAMEWORK USING API APPLICATION FRAMEWORK LAYER

Saidah Mastura binti Abdul Ghani

Master of Science in Information and Communication Technology

2023

ENHANCED ANDROID MALWARE DETECTION FRAMEWORK USING API APPLICATION FRAMEWORK LAYER

SAIDAH MASTURA BINTI ABDUL GHANI

A thesis submitted
in fulfillment of the requirements for the degree of Master of Science in Information and Communication Technology

Faculty of Information and Communication Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2023

DECLARATION

I declare that this thesis entitled “Enhanced Android Malware Detection Framework Using API Application Framework Layer” 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 : SAIDAH MASTURA BINTI ABDUL GHANI

Date : 6 JUNE 2023

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in term of scope and quality for the award of Master of Science in Information and Communication Technology.

Signature : .....

Supervisor Name : ASSOC. PROF. TS. DR. MOHD. FAIZAL BIN ABDOLLAH

Date : 6 JUNE 2023

DEDICATION

Thank you very much to:

my parents,
my family,
my supervisor and co-supervisor,
lecturers and teachers
the staff
and last but not least, to all my friends.

For,
understanding, supporting, believing and giving me inspiration and strength to me during this research periods.

Thank you.

ABSTRACT

Android is an open source mobile operating system which is gaining more popularity among users. Everyone is given the opportunities to develop Android application because of the existence of the API framework in Application Framework layer. Besides, Application Framework layer consists of manager blocks which can be used to access data in Android device, making the most vulnerable layer in which the malware developers like to exploit. This thesis was to develop an enhanced framework to detect Android malware application using Application Framework layer components. Static analysis technique was selected for developing the proposed enhanced framework. The proposed enhanced framework can be used to overcome the weaknesses of recent framework which are not categorizing the API into hierarchical level and used the inappropriate input for API data collection. Then, the experiment was carried out to test the effectiveness of the proposed enhanced framework using API and manager classes as parameters. By using Decision Tree, k-Nearest Neighbour and Random Forest algorithms, the results were analysed and the performance of the proposed enhanced framework was validated using Confusion Matrix calculation. The best performance of this proposed enhanced framework was obtained by using Decision Tree algorithms in both categories with accuracy of 82.75% for API category and 86.00% for manager classes category. Although the performance of detection rate was low, this proposed enhanced framework can still identify the malware behaviour using the categorization of API according to the hierarchical level of API. To improve the performance, a combination of parameters instead of using a single parameter should be utilised and a combination of dynamic and static analysis techniques should also be used for future research.

RANGKA KERJA PENGESANAN PERISIAN HASAD ANDROID YANG DIPERTINGKATKAN MENGGUNAKAN API PADA LAPISAN RANGKA KERJA APLIKASI

ABSTRAK

Android adalah sistem operasi mudah alih sumber terbuka yang semakin popular dalam kalangan pengguna. Setiap orang diberi peluang untuk membangunkan aplikasi Android kerana adanya rangka kerja API dalam lapisan Rangka Kerja Aplikasi. Selain itu, lapisan Rangka Kerja Aplikasi terdiri daripada blok-blok pengurus yang boleh digunakan untuk mengakses data dalam peranti Android, menjadikan lapisan yang paling terdedah di mana pemaju perisian hasad suka mengeksploitasi. Tesis ini adalah untuk membangunkan rangka kerja yang dipertingkatkan untuk mengesan perisian hasad di dalam aplikasi Android menggunakan komponen-komponen lapisan Rangka Kerja Aplikasi. Teknik analisis statik dipilih untuk membangunkan rangka kerja yang dipertingkatkan. Rangka kerja yang dipertingkatkan yang dicadangkan boleh digunakan untuk mengatasi kelemahan rangka kerja baru-baru ini yang tidak mengkategorikan API mengikut tahap hierarki dan menggunakan input yang tidak sesuai untuk pengumpulan data API. Kemudian, eksperimen dijalankan untuk menguji keberkesanan rangka kerja yang dipertingkatkan yang dicadangkan menggunakan API dan kelas pengurus sebagai parameter. Dengan menggunakan algoritma Decision Tree, k-Nearest Neighbour dan Random Forest, keputusan dianalisa dan prestasi rangka kerja yang dipertingkatkan telah disahkan menggunakan pengiraan Confusion Matrix. Prestasi terbaik rangka kerja yang dipertingkatkan ini telah diperoleh dengan menggunakan algoritma Decision Tree di kedua-dua kategori dengan ketepatan 82.75% untuk kategori API dan 86.00% untuk kategori kelas pengurus. Walaupun prestasi kadar pengesanan adalah rendah, rangka kerja yang dipertingkatkan yang dicadangkan ini masih dapat mengenal pasti tingkah laku perisian hasad dengan menggunakan pengkategorian API mengikut tahap hierarki. Untuk meningkatkan prestasi, gabungan parameter dan bukannya menggunakan parameter tunggal harus digunakan dan gabungan teknik analisis dinamik dan statik juga harus digunakan untuk penyelidikan masa depan.

ACKNOWLEDGEMENTS

In the name of ALLAH, Most Gracious and Most Merciful. All praises and thanks go to Allah. Alhamdulillah, finally, my Master thesis is successfully completed.

First of all, I would like to take this opportunity to acknowledge my supervisor, Associate Professor Ts. Dr. Mohd. Faizal bin Abdollah, and my co-supervisor, Associate Professor Ts. Dr. Robiah binti Yusof for their insightful supervision, support, encouragement and inspiration towards the completion of this thesis.

Also, this acknowledgement is given to both of my parents, Mr. Abdul Ghani bin Mohamad Nayan and Mdm. Rokiah binti Ahmad, for their sacrifices in ensuring my success throughout this journey. Thank you for your continuous support and blessing.

Thank you to UTeM and FTMK for providing me with good facilities for this project. Special thank you for Mr. Mohd. Rif'an bin Abdul Rahman, a technician in-charge for CMERP lab for helping with the laboratory problems. Thank you also to Ts. Dr. Mohd. Zaki bin Mas'ud, Dr. Fahmi bin Arif, Associate Professor Ts. Dr. Choo Yun Huoy and Dr. Zanariah binti Jano for sharing knowledge regarding this thesis. I would like to thank my classmate, Mr. Rudy Fadley bin Dolah for helping me in various ways.

Special thanks to my siblings and friends for their moral support. Lastly, thank you to everyone who has been involved directly and indirectly in this research endeavours.
Once again, thank you everyone for every good deed you have showered upon me.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF PUBLICATIONS

1. Ghani, S.M.A., Abdollah, M.F., Yusof, R., and Mas'ud, M.Z., 2015. Recognizing API Features for Malware Detection Using Static Analysis. *Journal of Wireless Networking and Communications,* 5(2A), pp. 6-12.

CHAPTER 1

INTRODUCTION

1.1 Introduction

In this cyber era, the 'always connect' philosophy is used by users so communication can take place with anybody at anytime and anywhere. The mobility of the devices is one of the features which attracts users. Therefore, size matters for users in their devices' selection. Other than that, the devices' sophistication also become one of the essential elements to users because they tend to update their device functionalities often to satisfy their needs. Hence, mobile devices such as smartphone and tablet are among the most popular devices preferred by users.

According to Meulen and Forni (2017), shipments of mobile phones are increasing steadily from year to year whereas the personal computer (PC) market is decreasing. Hence, mobile devices such as mobile phones are gaining popularity among users.

Sophisticated mobile phones have similar functions as PCs like browsing the Internet, performing online banking, and updating status in social network (Qadir et al., 2014). Other than that, mobile phones come in compact size compared with PCs which appeal to many users (Feizollah et al., 2015). Mobile phone also offer the all-time connection to the network where users can use their mobile devices every time and everywhere to connect to anybody from any place. The increase of social network media enhances users' needs to utilize the network. Additionally, the social media applications can be accessed through mobile devices, enhancing the popularity of mobile devices.

Android, one of the mobile operating systems, is used in mobile devices. It is an open source operating system which is developed by the Open Handset Alliance (OHA) led by Google Inc. (Guo-Hong, 2014; Kaur and Sharma, 2014) and based on Linux kernel (Wang et al., 2011; Ma et al., 2014; Shewale et al., 2014). It is also primarily designed for touchscreen mobile phones.

Android offers many applications to users. These applications can be downloaded from the official Android market called the Google Play or other Android markets. Android gives opportunities to anyone to develop its applications. Developers can also distribute their applications in Android market. Hence, Android applications are multiplying in volume in such a short time.

This opportunity has attracted malware developers to develop malwares using Android applications. Malware can cause system damage for any devices, leakage of confidential data or/and financial loss (Li and Clark, 2013; Purushothaman et al., 2014; Yoon et al., 2016).

Most applications which are developed and distributed to users are not going through a screening process (Raveendranath et al., 2014; Feizollah et al., 2015). This is one of the vulnerabilities of Android which may open the way for malware developers to spread their malware widely.

Therefore, this thesis enhanced a framework which statistically analysed the source code of normal and malware Android applications. The enhanced framework enabled the normal and malware Android applications behaviour to be distinguished using the application source code.

1.2 Research Problem

The increase of smartphone devices using Android as mobile operating system makes Android devices become the favourable target to malware developers to spread their malware. Most common technique used by malware developers is repackaging technique (Jiang and Zhou, 2013). Repackaging technique is a technique where normal and legitimate Android applications are repackaged with malicious content (Datta et al., 2014; Yoon et al., 2016). To develop a malware application, malware developers can download the popular legitimate applications, then disassemble the application by doing reverse-engineering to obtain the source code, insert the malicious code into the application and re-assemble the application and afterward, the new application is submitted to the official Android market and/or alternative Android markets to spread the malware application to users (Jiang and Zhou, 2013; Raveendranath et al., 2014).

Android provides Application Programming Interface (API) framework to developers by allowing them to use these APIs to develop their applications (Guo-Hong, 2014; Kaur and Sharma, 2014; Ma et al., 2014). API is needed by every Android application to interact with devices (Feizollah et al., 2015). API is also used to accomplish tasks like retrieving or modifying data (Wang et al., 2011). These APIs are also used by malware developers to develop malware applications for their malicious intention. These API frameworks are located at the Application Framework layer of Android architecture (Guo-Hong, 2014; Kaur and Sharma, 2014). Besides, Application Framework layer also contains manager blocks where these blocks are used to allow the applications to access data (Kaur and Sharma, 2014). According to Shewale et al. (2014), the Application Framework layer is the most vulnerable one where malware developers often exploit this layer to gather users' sensitive information.

The repackaging technique modifies the source codes of normal application. Therefore, to detect the malware Android application behaviour, their source codes need to be analysed. APIs and their manger blocks are the best features to analyse Android malware because they are situated at the most vulnerable layer. Static analysis can be used to analyse the Android application source codes which consist of API without the application execution (Chandramohan and Tan, 2012; Rami and Desai, 2013; Pandey and Mehtre, 2014; Qadir et al., 2014; Raveendranath et al., 2014). Recent static analysis frameworks which use API have some weaknesses. One of the weaknesses of the recent frameworks is the APIs are not categorized according to their hierarchical level. When APIs are not categorized appropriately, the misunderstanding of their hierarchical level can occur (Westyarian et al., 2015). Besides, when the hierarchical level of API is not used accordingly, the malware behaviour cannot be detected when using Android API (Li and Li, 2015). In addition, by categorizing the API, the steps (Aafer et al., 2013) can be simplified. Milosevik and Dehghantanha (2017) used the whole source code in their study. However, the usage of the whole source can make the malicious code in malware difficult to be found. Therefore, the usage of source code needs to be categorized according to their hierarchical level to easily detect the malicious code used in the malware.

Using binary expression or used API at once is another weakness of recent frameworks (Arp et al., 2014; Sheen et al., 2015; Jung et al., 2018) because APIs are used repetitively by an application to make the application achieves its functionality. Therefore, the usage of binary expression to analyse malware behaviour using API is not appropriate.

To overcome these recent weaknesses, this thesis developed an enhanced framework to detect Android malware application using API. This framework was

included in the categorization of APIs according to their hierarchical level. Besides, this enhanced framework utilised frequency analysis to the APIs which are used repetitively in Android application as the input for analysing malware behaviour.

Therefore, the statement of problem is as follows:

Static analysis can be used as an analysis technique in detecting malware Android applications. However, recent static analysis frameworks have weaknesses that need to be overcome. The weaknesses of these recent frameworks are the Android APIs are not categorized according to their hierarchical level and the binary expression is used on API which API utilises repetitively in an application. Therefore, this thesis developed an enhanced static analysis framework using API which was situated at the Application Framework layer to overcome the recent frameworks' weaknesses.

Thus, this thesis enhanced a framework to detect Android malware applications by analysing the recent frameworks. Table 1.1 shows the summary of the research problem.

Table 1.1: Summary of Research Problem

RP	Research Problem
RP1	Recent static analysis frameworks have weaknesses in detecting Android malware application behaviour when using parameter which is situated at Application Framework layer.

1.3 Research Questions

This thesis consists of four (4) research questions to identify the research problem discussed.

RQ1: What is the parameter used by recent frameworks in detecting the Android malware behaviour at the Application Framework layer?

This research question is constructed by considering the usage of parameter in Application Framework layer to detect Android malware application as stated in RP1 in Table 1.1.

RQ2: What are the recent frameworks' weaknesses?

This research question is formulated to identify the weaknesses of recent frameworks as highlighted in RP1 in Table 1.1.

RQ3: How to overcome these recent frameworks' weaknesses?

This research question is formulated to find the solutions to overcome the recent framework weaknesses as stated in RP1 in Table 1.1.

RQ4: How to validate the effectiveness of the proposed enhanced framework?

This research question is constructed to ensure the effectiveness of the component in the enhanced framework by validating the framework.

The summary of the research questions is shown in Table 1.2.

Table 1.2: Summary of Research Questions

RP	RQ	Research Questions
RP1	RQ1	What is the parameter used by recent frameworks in detecting the Android malware behaviour at the Application Framework layer?
	RQ2	What are the recent frameworks' weaknesses?
	RQ3	How to overcome these recent frameworks' weaknesses?
	RQ4	How to validate the effectiveness of the proposed enhanced framework?

1.4 Research Objectives

Based on the research questions illustrated in Table 1.2, the research objectives for this thesis were determined. The objectives of this thesis were as follows:

RO1: To identify the parameter in the Application Framework layer to distinguish between normal and malware Android applications.

This research objective was to identify the APIs and managers which were used in both malware and normal Android applications to differentiate their behaviour because these parameters were situated at Application Framework layer.

RO2: To develop an enhanced framework for detecting malware in Android applications using the Application Framework layer components.

Based on the recent frameworks' weaknesses and their solutions, an enhanced framework was proposed. This proposed framework used the parameters from the Application Framework layer.

RO3: To validate the proposed framework using machine learning algorithm.

The proposed framework was validated for its effectiveness in detecting Android malware applications using machine learning algorithms such as Decision Tree, k-Nearest Neighbour and Random Forest algorithms.

The summary of research objectives is depicted in Table 1.3.

Table 1.3: Summary of Research Objectives

RP	RQ	RO	Research Objectives
RP1	RQ1	RO1	To identify the parameters in the Application Framework layer to distinguish between normal and malware Android applications.
	RQ2	RO2	To develop an enhanced framework for detecting malware in Android applications using the Application Framework layer components.
	RQ3		
	RQ4	RO3	To validate the proposed framework using machine learning algorithm.