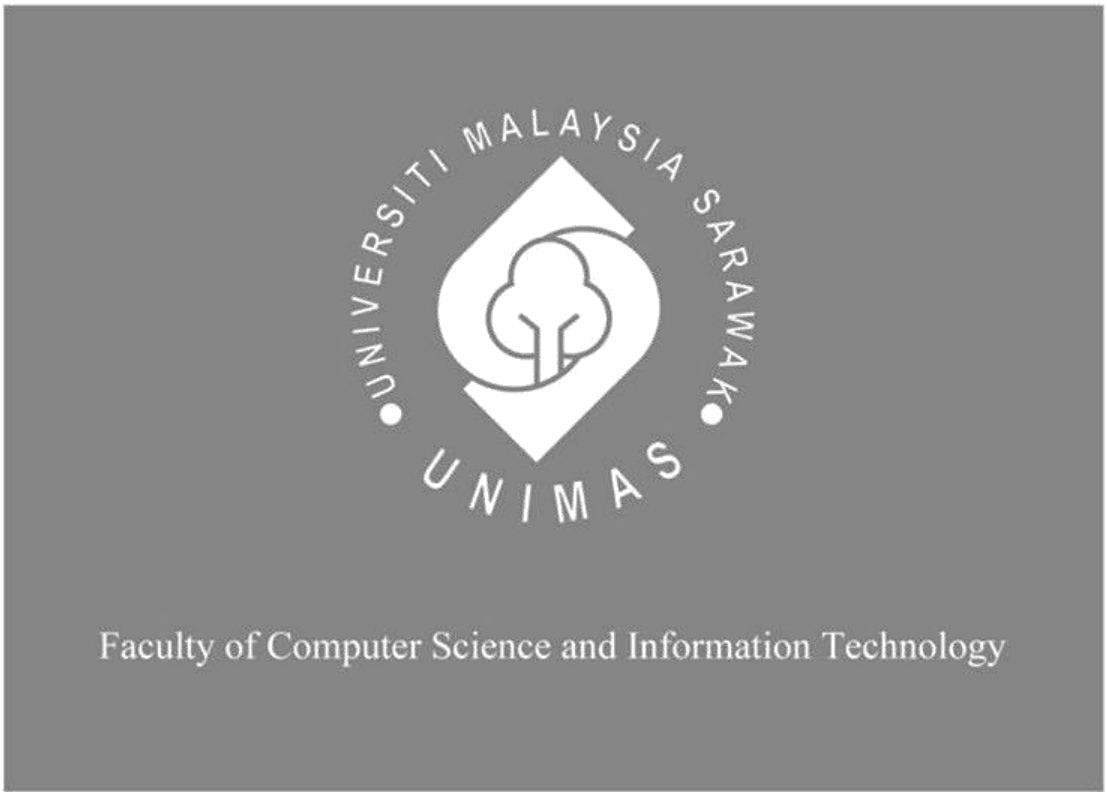

A COLLABORATIVE FRAMEWORK FOR ANDROID MALWARE IDENTIFICATION USING DYNAMIC ANALYSIS

THAYAALENI RAJANDRAN

Bachelor of Computer Science with Honours
(Network Computing)

2019

A COLLABORATIVE FRAMEWORK FOR ANDROID MALWARE IDENTIFICATION USING DYNAMIC ANALYSIS

THAYAALENI RAJANDRAN

This project is submitted in partial fulfilment of the

requirements for the degree of Bachelor of

Computer Science with Honours

(Network Computing)

Faculty of Computer Science and information Technology

UNIVERSITI MALAYSIA SARAWAK

2019

KERJASAMA RANGKA KERJA UNTUK PENGENALAN ANDROID MALWARE MENGGUNAKAN ANALISIS DINAMIK.

THAYAALENI RAJANDRAN

Projek ini merupakan salah satu keperluan untuk

Ijazah Sarjana Muda Sains Komputer dan

Teknologi Maklumat

(Pengkomputeraan Rangkaian)

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2019

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE: **A COLLABORATIVE FRAMEWORK FOR ANDROID MALWARE IDENTIFICATION USING DYNAMIC ANALYSIS.**

ACADEMIC SESSION: 2018/2019

THAYAALENI RAJANDRAN

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (√)

☐	CONFIDENTIAL	(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)
☐	RESTRICTED	(Contains restricted information as dictated by the body or organization where the research was conducted)
√	UNRESTRICTED	

Validated by

_____ _____

(AUTHOR'S SIGNATURE) (SUPERVISOR'S SIGNATURE)

Permanent Address
No.2, Jalan Tun Teja 6,
Taman Tun Teja,
48000 Rawang,
Selangor Darul Ehsan.

Date: _____ Date: _____

Note * Thesis refers to PhD, Master, and Bachelor Degree
** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

DECLARATION

I hereby declare that this thesis with the title of "A Collaborative Framework for Android Malware Identification Using Dynamic Analysis" is my research work except for some information that quoted and cited from other resources which have been acknowledged. I also declare that no portion of the work that referred to in this report has been submitted in support of application for another degree at Universiti Malaysia Sarawak (UNIMAS).

Signature:

……

Author Name: Thayaaleni Rajandran

Date: 27.04.2019

ACKNOWLEDGEMENT

This project work has been developed to meet the academic requirements of University Malaysia Sarawak (UNIMAS) for the completion of Bachelor of Degree (Hons) in Computer Science. I would like to my express my gratitude and gratefulness to all the people who have been involved either directly or indirectly in the successful completion of my Final Year Project (FYP). Firstly, I would like to show my deepest appreciation to my supervisor, Mr. Rajan Thangaveloo for his motivation, guidance, encouragement by helping me to structure my Final Year Project. His willingness to spend his time and his patience has been very much appreciated. Next, I would like to thank to Final Year Project coordinator, Professor Dr Wang Yin Chai for providing guidelines and coordination throughout the conduction of Final Year Project. Last but not least, I am thankful to my parents and friends who have been always supporting and encouraging me throughout the development of the project.

ABSTRACT

The project proposed a dynamic analysis technique in Android malware detection. The objectives of the project are to investigate the Android malware using dynamic analysis technique and to enhance the accuracy of malware detection. The scope of this project focuses on Android Malware detection by using dynamic analysis. The methods to implement this project is through data collection, feature extraction, feature selection, and classification process. The machine learning algorithm is used to train and test datasets with the percentage of 70% which is 140 samples from malware and benign applications and 30% of total datasets which is 60 samples from malware and benign applications respectively. The Correlation-based Feature Selection Evaluator (CfsSubset) algorithm is applied in feature selection process in order to improve the classification process. Lastly, the classification result is generated. The proposed project will extract the features of system calls, network packets, CPU usage and battery usage of the application. The proposed project achieves overall accuracy level of 96.67% using Sequential Minimal Optimization classifier.

ABSTRAK

Projek ini mencadangkan teknik analisa dinamik dalam pengesanan malware Android. Objektif projek adalah untuk menyiasat malware Android menggunakan teknik analisis dinamik dan untuk meningkatkan ketepatan pengesanan malware. Skop projek ini memberi tumpuan kepada pengesanan Android Malware dengan menggunakan analisis dinamik. Kaedah untuk melaksanakan projek ini adalah melalui pengumpulan data, pengekstrakan ciri, pemilihan ciri, dan proses klasifikasi. Algoritma pembelajaran mesin digunakan untuk melatih dan menguji dataset dengan peratusan sebanyak 70% iaitu 140 sampel dari perisian malware dan aplikasi jinak dan 30% daripada jumlah dataset yang masing-masing 60 sampel dari aplikasi malware dan jinak. Algoritma Pemilihan Ciri-ciri Pemilihan Korelasi (CfsSubset) digunakan dalam proses pemilihan ciri untuk meningkatkan proses klasifikasi. Terakhir, hasil pengelasan dihasilkan. Projek yang dicadangkan akan mengeluarkan ciri-ciri panggilan sistem, paket rangkaian, penggunaan CPU dan penggunaan bateri aplikasi. Projek yang dicadangkan mencapai tahap ketepatan keseluruhan 96.67% menggunakan Pengelas Pengoptimuman Minimum Sequential.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Project Title

A Collaborative Framework for Android Malware Identification Using Dynamic Analysis.

1.2 Introduction

Smart devices are electronic devices with an operating system. Over the decade, smart devices had gained immense popularity and most importantly they have changed the way we do our daily things. Smart devices can be connected to other devices through different wireless networks such as Wi-Fi, NFC, Bluetooth and many more. It helps us to do our daily tasks such as online banking, internet browsing and many more. One of the most prominent mobile operating system is Android. Android is also the most powerful operating system in smartphones, and it is user-friendly. Android phones are much easier to hack when compared to other mobile platforms.

According to (Broersma, 2017), there are more than one hundred thousand Android devices been running by botnet to launch distributed denial-of-service (DDoS) attacks. There are over hundreds of malicious applications been downloaded from Google Play Store, traced by researchers (Broersma, 2017). (Goodin, One of 1st-known Android DDoS malware infects phones in 100 countries, 2017) stated about three hundred applications in Google Play market found to be botnet. (Somarriba & Zurutuza, 2017) stated the security concerns of Android have gotten attackers interest lately, that seeking to take advantages of its vulnerabilities, and these issues are being dealt in both academia and industry.

(Singh & Hofmann, Dynamic Behavior Analysis of Android Applications for Malware Detection, 2017) stated that about 99% attacks are in Android compared to other operating

systems. In the IT threat evolution report (Chebyshev, Sinitsyn, Parinov, Liskin, & Kupreev, 2018) shows that Bangladesh and China were shortlisted as top ten countries of Android

malware detection with 31.17% and 31.07% respectively in second quarter of 2018. The malicious applications can be any type, such as trojan horse, ransomware, spyware and so on. It is vital to study about the behaviour to identify the malicious applications and this can be performed by malware analysis. There are three primary techniques to identify malware analysis which are static analysis, dynamic analysis and hybrid analysis. These techniques are used to analyse the behaviours and test system call, API calls, permission and network packets of the Android applications.

1.3 Problem Statement

A recent analysis exhibits that Android is being the most leading operating system for smartphone and tablets. Due to this, Android platform is being the most vulnerable target for malware attacks. In recent years, the number of Android malware are distressing due to the increased number of malicious applications (Singh, Jaafar, & Zavarsky, 2018) . Android operating system has an open source platform where the source codes are freely available for developers. One of the main reasons for these attacks is to steal user's data or money. This is because smartphones become a place to store personal information such as banking information.

However, due to rapid increase in malicious applications, it is important to find an efficient solution for malware detection. Even though some researchers used in their proposed features, but they still face some difficulties in detecting the malwares more accurately. Next, the restraints of static analysis in Android malware detection technique. The static analysis unable

to find the malicious behaviour of the Android applications during the run-time executions of the applications.

1.4 Objectives

This project aimed:

1. To investigate the Android Malware by using dynamic analysis technique.
2. To enhance the accuracy level of malware detection.
3. To determine detection technique that produce higher accuracy results.

1.5 Methodology

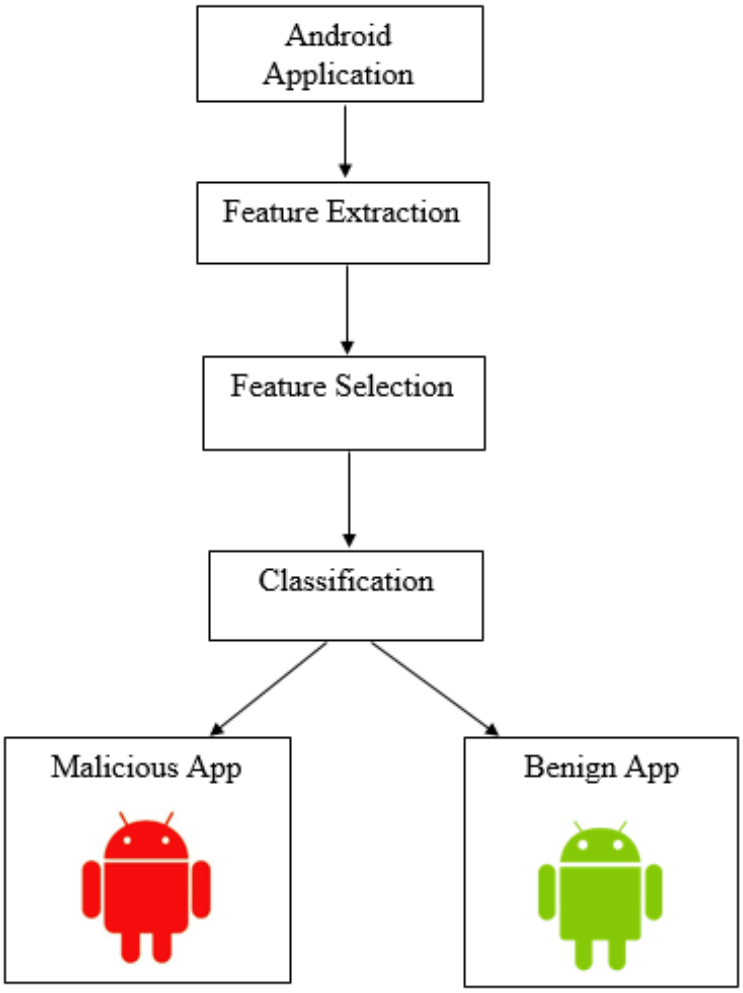

Figure 1. 1: General architecture of malware detection.

The Figure 1.1 shows the general architecture of Android malware detection. There are four steps involved in malware detection using dynamic analysis technique. The first step is Android applications is installed in the emulator and execution is initiated followed by feature extraction, feature selection using Correlation-based Feature Selection (CfsSubset) Evaluator and classification using Sequential Minimal Optimization (SMO) algorithm and lastly results is indicated as malicious and benign application.

1.6 Scope

The scope of this project focuses on Android Malware detection by using dynamic analysis.

1.7 Significance of Project

The significance of this project is to identify and analyse the behaviour of Android application using dynamic analysis technique. The goal of this project is to detect the abnormal behaving applications. By carrying out this project, we can increase level of accuracy for Android Malware detection.

1.8 Expected Outcome

The outcome of this project is an Android Malware detection technique.

1.9 Thesis Outline

Chapter 1: Introduction

Chapter 1 includes proposed project title and general introduction of the project. The introduction composed of problem statement, objectives, methodology, scope, expected outcome, project schedule, significance of the project and thesis outline.

Chapter 2: Literature Review

This chapter presents the literature review of the project. In this chapter, literature review includes the dynamic analysis techniques, classification and features. Besides that, a comparison was made regarding the different frameworks used in previous researches.

Chapter 3: Methodology

Chapter 3 explains about the methodology of the project. This chapter illustrates the detailed description of the proposed framework techniques been used for Android Malware detection. This chapter discussed on software requirements and tools required for this project. Diagrams and tables attached together as an evidence.

Chapter 4: Implementation

In chapter 4, it provides a detailed explanation about the implementation of the proposed technique. Experiment and processing setup environment with software requirements where organized and screenshot of the results are provided in this section.

Chapter 5: Results

Chapter 5 indicated the results of dynamic analysis, data collection after feature extraction and experimental results. A comparison made between new and existing results. The final accuracy results presented after pass through the True-False technique.

Chapter 6: Conclusion

Chapter 6 explains about the contribution, limitation, and overall conclusion of the project has been completed. The future works to improve the proposed technique of this project also discussed in this chapter.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter illustrates the summarize, evaluate and explain of some related studies and researches on Android malware detection techniques. The detection of malware in smartphone gadgets is one of the vital topics in cybersecurity. There are three types of techniques to detect Android malware on Android smartphones. The three techniques are static analysis, dynamic analysis and hybrid analysis. Static analysis detection used to extract some features from an application and analysed the application before the execution. In dynamic analysis detection, the applications are executed in a simulator and determined according to the log files. Hybrid analysis detection is the combination of static analysis and dynamic analysis. This project focuses on the dynamic analysis technique to detect the malware behaviours in Android smartphones.

2.2 Related Work

The three types of analysis techniques which is dynamic analysis, static analysis and hybrid analysis techniques are discussed in this section. The techniques use feature sets, classification research and frameworks to detect the malicious applications. The comparison among techniques is discussed in this section.

2.2.1 Dynamic Analysis

Dynamic analysis also known as behavioural-based analysis. Dynamic analysis is the testing and evaluation of a program by executing data in real-time. (Yang, Wei, Xu, He, & Wang, 2016) proposed an effective dynamic analysis method which is DroidWard for the detection of Android malware. The goal of DroidWard is to extract the most effective and significant features to indicate the malicious behaviour. It also aims to enhance the detection accuracy of malicious application. The machine learning classifiers that used to perform malware classifications are Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF). In this work, Droidbox was customized to extract dynamic features of the applications such as the size of the network packets and sensitive API calls. The experiment proposed used 15 features which includes 9 features from existing works and 6 proposed novel features. The proposed 6 novel types of dynamic features are anti-stimulator, hidden application's icons, packet size, number of distinct sensitive API calls, number of API calls per unit time and requested root permissions. The existing 9 features are file access, encrypted operations, data leakage, permission leakage, dynamic loading, SMS operation, network traffic sent, network traffic received and telephone operation. The DroidWard runs the applications, extract features in order to identify malicious and benign applications. The evaluation results of DroidWard shows that it classifies 98.54% of malicious applications with 1.55% of false positive.

(Koli J. , 2018) introduced Randroid system which applies various machine learning techniques. The machine learning techniques that used to perform malware classifications are Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB) and Random Forest (RF). The proposed system uses permission, API calls along with existence of key application's information. (Koli J. , 2018) presented system of Android malware detection which extract relevant features to be used in machine learning classifiers in order to identify malicious and

benign Android applications. The system design divided into two parts, (i) present the research methodology and (ii) explains the classification model used in the system. Figure 2.1 shows the research methodology used in Randroid design (Koli J. , 2018). There are four phases in the methodology. Phase 1 is the reverse engineering process where the applications were decompiled into their source code in the form of *AndroidManifest.XML* and java classes by using malware analysis tool which is Androguard. Androguard is a toolkit that built in Python which provides reverse engineering and malware analysis for Androids. Next, Phase 2 is the feature extraction process. Using python module, the necessary features are interpreted from the source. This is saved in the document-oriented MongoDB database. The features that are extracted includes requested permissions, API calls, together with existing of key information approaches such as: is_crypto_code; is_dynamic_code; is_native_code; is_database_code and is_reflection_code. Phase 3 is used to transfer from extracted features of every application into a binary vector that can be useful for machine learning algorithms. Every application showed as a unique instance with binary vector of features and the class label determines the applications whether malicious or benign. Finally, the phase 4 is to model the classifiers by using four supervised machine learning algorithms which are Support Vector Machine, Decision Tree, Random Forest and Naïve Bayes with binary vector of the sample applications.

The second part of the system design is classification model. Figure 2.2 shows the classification model used in the system design (Koli J. , 2018). The model consists of two phases that are the training and prediction phase. In the training phase, a set of features are inferred and extracted from the source code of the sample malware and benign applications. The features that has been extracted will be depicted in the form of binary vector. Then, the set of binary vectors of applications along with the information label parsed into various machine learning algorithms. In prediction phase, the exact set of features extracted from source code and binary vector

generated for testing of malware and benign. Next, the generated binary vector passed into classifier module with the aid of classification models built in training phase.

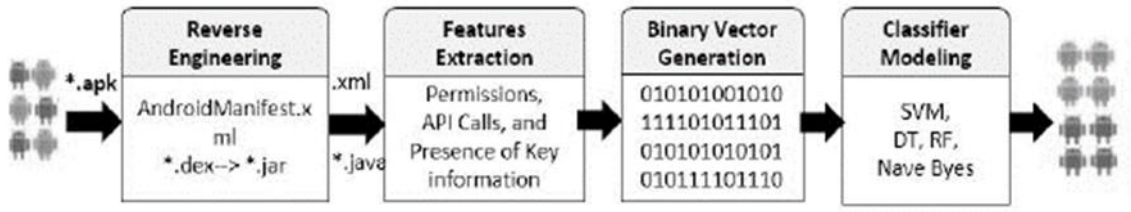

Figure 2. 1: Illustrates research methodology of Randroid design.*(Koli J. D., 2018)*

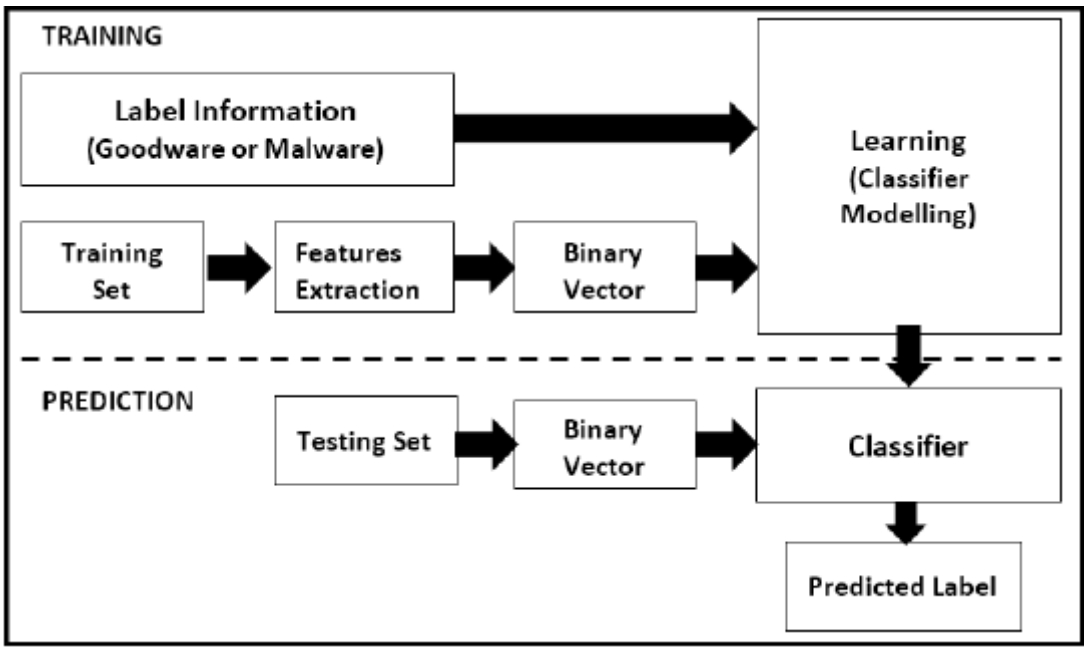

Figure 2. 2: The classification model of system design.*(Koli J. , 2018)*

(Somarriba & Zurutuza, 2017) proposed a framework for examining the Android applications in a platform-independent manner in order to detect restricted API calls used during the run-time of the applications. The Dynamic Analysis (DA) implies the execution of the application conducted via instrumenting or virtual machine monitoring to inspect the behaviour of application. This paper was focused on capturing requested URL for identifying malicious transactions commenced by an application that running on Android smartphone. Next, the evaluation of combining and correspond two approaches which is top-down detection by recognizing malware domains using Domain Name System (DNS) network traffic and bottom-up detection using the classical dynamic analysis. The whole procedure is divided into four stages. Firstly, generation of application traces in the first stage followed by data collection and analysis, monitoring of network traffic and fourth phase is visualization. In the first phase, the application traces are the malware traces that capable to capture the DNS queries done by the application under test. The second phase consists of the log aggregation and transport of data generated, the extraction of URLs from application traces and DNS network-service traces done with the aid of Python-language scripts. Next, the third phase which is the search and analytics where the correlation of the monitored events in the previous task must be performed. The last phase which is the visualization phase carried out to perform a visual analysis of the platform using *Kibana* from the Elastic Stack.

(Liu, Gu, Li, & Su, 2017) have proposed Realdroid which is a dynamic and emulator-based analysis. It can abduct Android evasive malware and is capable of large-scale malware detection. There are five main components in Realdroid which are file disguise, API disguise, property disguise and hardware extension and interactive stimulation. Evasive malware can be influenced to expose its malicious behaviour. Realdroid can efficiently detect the evasive malware. Realdroid includes UI traversal Android Test Engine (ATE) so that the applications can be analysed mechanically in a vast scale without manual interference. Next, the process-

level behaviour monitoring technique been used to keep track of system calls invokes by the target applications. Thereafter, the dynamic behaviours of the applications been analysed, and malware is identified by its malicious behaviours.

(Faiella, et al., 2017) have proposed a Distributed BRIDEMAID (D-BRIDEMAID) framework for coordinated analysis of Android applications. The goal is to detect the new threats via distributed dynamic analysis. D-BRIDEMAID framework is constituted by host application and it is a lightweight application which does not interrupt the daily usage of smartphone device. The functions of D-BRIDEMAID were to report the suspicious behaviour to the user and detect malicious applications. When installed, D-BRIDEMAID host applications validate the tester over IMEI and IMSI in real environment. This is to assure that the application is executing using a real environment. This list of unknown applications is proposed to let the user to test. The tester will determine whether to run the application. In order to test the application, the user need to install and launch the D-BRIDEMAID. The D-BRIDEMAID framework is capable to check the background time of applications while the application is running. After the tester evaluated application, a report will be submitted based on the eventual BRIDEMAID alerts. A mathematical model been used by D-BRIDEMAID to analyse the reliability of reports by combining them into a single matrix. The decision of application trustworthiness is analysed and collected by using an aggregator.