A thesis submitted in partial satisfaction of the requirements
for the degree of Master of Computer Science and Engineering
in the Graduate School of the University of Aizu

Detecting Android malware applications using Static Analysis and Machine learning

by

Cristian Naoki Kamia

March 2020

The thesis titled

Detecting Android malware applications using Static Analysis and Machine learning

by

Cristian Naoki Kamia

is reviewed and approved by:

Main referee
Senior Associate Professor
Vitaly Klyuev
Professor
Pyshkin
Professor
Paik

THE UNIVERSITY OF AIZU

March 2020

Contents

List of Figures

List of Tables

List of Abbreviations

List of Symbols

Acknowledgment

Abstract

Chapter 1

Introduction

Android operating system, which is provided by Google, is predicted to continue have a dramatic increase in the market with around 1.5 billion Android-based devices to be shipped by 2021 sta. It is currently leading the mobile OS market with over 80% market share compared to iOS, Windows, Blackberry, and Symbian OS as Figure 1.1. The availability of diverse Android markets such as Google Play, the official store, and third-party markets makes Android devices a popular target to not only legitimate developers, but also malware developers. Over one billion devices have been sold and more than 65 billion downloads have been made from Google Play []. Android apps can be found in different categories, such as educational apps, gaming apps, social media apps, entertainment apps, banking apps, etc.

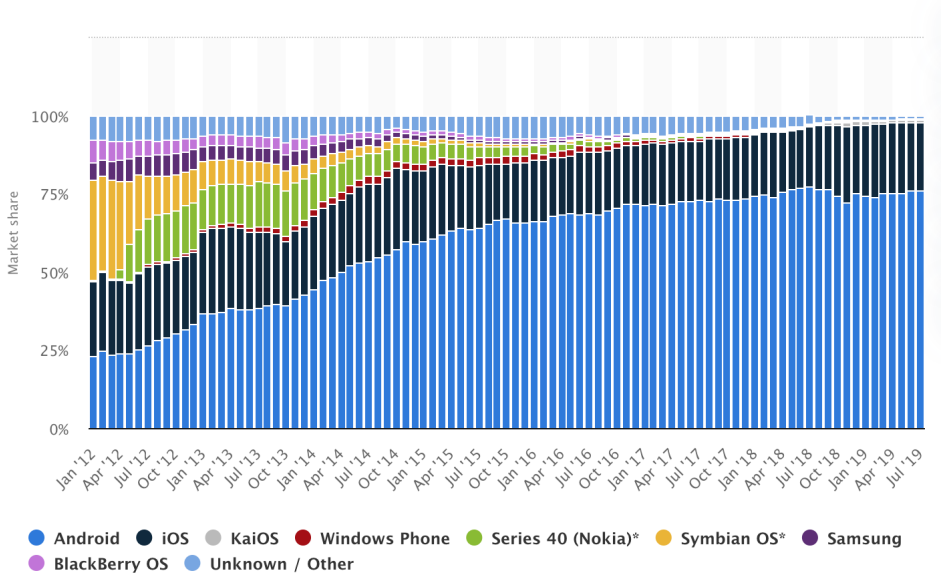

Figure 1.1: Mobile operating systems's market share worldwide from January 2012 to July 2019 []

As a technology that is open source and widely adopted, Android is facing many challenges especially with malicious applications. The malware infected apps have the ability to send text messages to premium rate numbers without the user acknowledgment, gain access to private data, or even install code that can download and execute additional malware on the victims device. The malware can also be used to create mobile botnets []. Over the last few years, the number of malware samples attacking Android has significantly increased as you can see at Figure 1.2. Attacks on other connected things around the house gained momentum as well. While hidden apps and Adware remain by far the most common form of mobile threats in

Android, the others are growing and learning how to infect other types of devices as well. According to a recent report from McAfee [], detections of backdoors,cryptomining, fake apps, and banking Trojans all increased substantially in the latter half of the year.

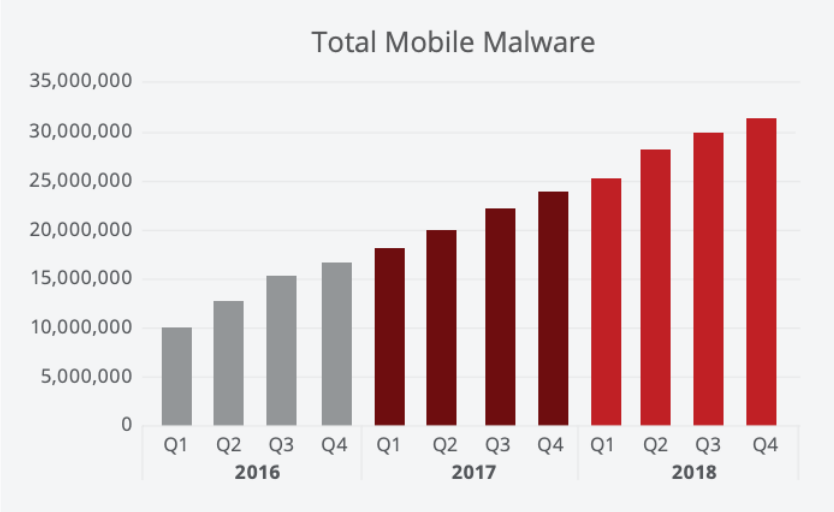

Figure 1.2: Total mobile malware

In order to mitigate the spread of malware, Google introduced a detection mechanism to its app market in Feb 2012 called Bouncer. Bouncer tests submitted applications in a sandbox for five minutes in order to detect any harmful behaviours; however, it has been shown that bouncer can be evaded by means of some simple detection avoidance methods (Oberheide and Miller, 2012). Alongside Bouncer, Google introduced Google Play protect in the Google 2017 event (Google Play, 2018). Google Play Protect is an always-on service that scans the applications automatically even after installation to ensure that the installed applications remains harmless 24/7. It has been reported that over 50 billion apps are scanned and verified every day regardless of where they were download from. However, according to McAfee, Google Play Protect also failed when tested against malware discovered in the previous 90 days in 2017 [McA]. Furthermore, most third-party stores do not have the capability to scan and detect submitted harmful applications. Clearly, there is still a need for additional research into efficient methods to detect zero-day Android malware in the wild in order to overcome the aforementioned challenges.

Cyber attacks manage to produce unprecedented levels of disruption, where attackers usually leverage diverse tools and tactics, such as zero-day vulnerabilities and malware []. This situation makes malware detection techniques worth studying and improving, in order to prevent and/or mitigate the effects of cyber attacks. Machine learning techniques can help to satisfy this demand,building classifiers that discern whether a precise Android application is malware or benignware. Algorithms such as Decision Trees [], Support-Vector Machines [] and Naive-Bayes [], to name a few, are able to build such classiers.Going further, ensemble methods for machine learning []Preprint submitted to Information Fusion 5th February 2019 aim at effectively integrating many kinds of classification methods and learners to benefit from each ones advantages and overcome their individual drawbacks, hence improvingthe overall performance of the classication.

Various approaches have been proposed in previous works with the intention of detecting Android malware. These approaches are categorized into static analysis, dynamic analysis or hybrid analysis (where static and dynamic are used together). The methods based on static analysis reverse engineers the application for malicious code analysis. Arp et al. (2014), Aafer et al. (2013), Yerima et al. (2015a), Fan et al. (2017), Yerima et al. (2015b), Kang et al. (2016b), Cen et al. (2015), Westyarian et al. (2015) and Kang et al. (2016a) are few examples of detection methods using static analysis. By contrast, dynamic analysis executes the application

in a controlled environment such as an emulator, or a real device with the purpose of tracing its behaviour. Several dynamic approaches, such as Enck et al. (2010); Alzaylaee, M.K., Yerima, S. Y., and Sezer S. (2016) DroidBox; Rastogi et al. (2013); Tam et al. (2015)Tracedroid, NVISO ApkScan have been proposed. However, the efficiency of these approaches rely on the ability to detect the malicious behaviour during the runtime while providing the perfect environment to kick-start the malicious code. And many approaches not have a end-to-end system to detect malware. They has some separated steps to predict and is not easy to use for users.

In this paper, we will focus on Static analysis methods to detect malware in android application files (APKs) using features extracted from these APK files with reverse engineering methods. We used a tool named AndroPytool[] to extract 7 different Static features described below.

- Activities
- API calls
- Opcodes
- Permissions
- Receivers
- Services
- System commands

Using these features we use a Stacking method [] to train our models. This method is often used by data scientists in Kaggle [] competitions to improve the final prediction and is one of the state-of-art method in table data competitions. Moreover, we build a end-to-end system which uses for input a android APK and output the result of detection, benign application or malware application.

Chapter 2

Background

2.1 Overview of Android OS

In the Figure 2.1, the Android software stack items in green are the written in C/C++ and the blue ones are written in Java which executed using the Dalvik VM [5]. Here the Android Linux Kernel is a modified Linux Kernel which includes features like wake locks (memory management for optimizing the memory consumption), Binder IPC Drivers and other features which play a key role in mobile embedded platform [21]. The libraries item plays a vital role in optimizing CPU, memory consumption as well as the audio and video codecs for the device.

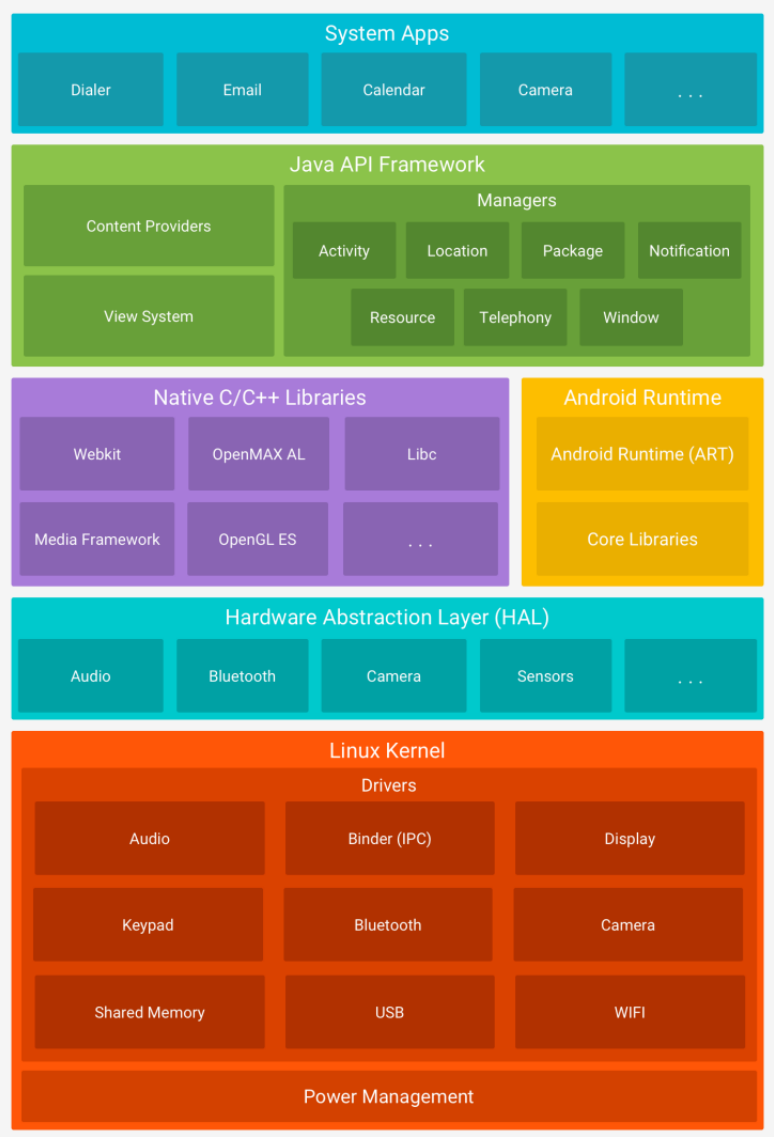

Figure 2.1: Android operating system architecture

References