Android Malware Detection: A Review

1st Kerem Safa Dirican
Computer Engineering
Istanbul Kultur University
Istanbul, Turkey
1800002205@stu.iku.edu.tr

2nd Emre Özay
Computer Engineering
Istanbul Kultur University
Istanbul, Turkey
1700004229@stu.iku.edu.tr

***Abstract*—Android is the most popular operating system in the market now. This makes the Android users a big target for attackers. Because the user base is so wide and diverse, protecting the system and creating a safe Application Store isn't enough. Users also should be protected from themselves. When doing this, experienced users should not be bothered by these security rules. A balance must be struck between security and freedom. To do this, exploits and harmful applications should be detected before they reach the target. As malware evolves, detection methods evolve as well.**

***Index Terms*—android, malware, mobile, security**

I. Introduction

In this era of technology, people from 7 to 70 have at least one mobile device and are the target of many threats. The main cause of these threats is malware. Malware, on the other hand, is the code prepared for the purpose of performing the actions desired by the person who made the software on the infected system or device. They settle on their devices for many reasons such as user unconsciousness, the misleading of this prepared software, and they have more authority than them. The main operation is explained by a 3-step cycle. First, the malware reaches the targeted device and settles. Then it achieves its intended purpose on this device. Finally, he sets a new goal to achieve the same goal.

A lot of information, from the user's credit card password to the address, can be obtained through a device belonging to the user. It is aimed to access this data in attacks on mobile devices. After this user's data is obtained, it can be used illegally for any purpose desired by the attacker. Thanks to these transactions, millions of users suffer material and moral damages. For this reason, in order to avoid such damages, users should be made aware, threats and their types should be informed, systematic vulnerabilities should be eliminated and these precautions should be taken before attacks are carried out. The main targets of mobile attackers are applications prepared on the basis of the operating system. The most used mobile operating systems in the world are Android and IOS. The android operating system is in the first place with a 70% usage rate. IOS comes in second place with a usage rate of 28.3%. Other operating systems have a usage rate of 1.7%.

The number of applications in the stores of these operating systems; Android applications: 2.7 million, IOS applications: 2.2 million.

In this article, we examined malware on Android, its detections, solutions, and what can be done in this area in the future. [2]

II. Android Overview

Google and the Open Handset Alliance developed Android, a free and Linux-based operating system for mobile devices. Despite the fact that the system is open source, Google has kept a small but critical portion of its code confidential. Google offers it for free since the system is evolving at a rapid pace, is utilized by many well-known businesses, and so helps its ads to reach a larger audience. Google makes money by displaying advertisements on the Google Play market for Android games and applications. Android's supported app extension is ".apk". [4]

A. Architecture of Android

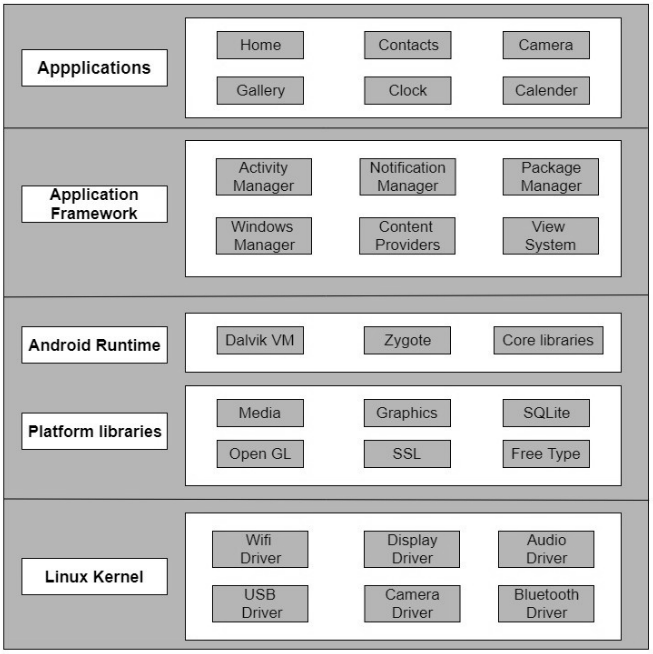

Fig. 1. Android Architecture
[1]

Identify applicable funding agency here. If none, delete this.

The main parts of the Android operating system are the Linux Kernel, Android Runtime, Platform Libraries, Applications, Application Framework.

1) **Linux Kernel**
According to this architecture, there is the Linux Kernel at the bottom layer. The Linux kernel must be in one-to-one communication with the hardware. So the Linux kernel contains the drivers for the hardware. In addition, the fact that Android systems are Linux-based is a convenience for developers to access other platforms. [6]
2) **Libraries:**
The layer on top of the Linux kernel is the Libraries layer. This layer contains the libraries of the system. The functions of these libraries are as follows:
    - **Surface Manager:** Used to manage screens and windows.
    - **Media Framework:** This allows the use of various compression and decoding types for media playback and recording.
    - **SQLite:** SQLite database is used for database management in Android systems and this library is used.
    - **WebKit:** A library for browser engine rendering.
    - **OpenGL:** A library used to properly display 2D and 3D content on the screen.

    Libraries used in Android systems are written in C and C++ languages, with most of them coming from Linux. The biggest difference between Android systems and Linux-based systems is that Linux does not use the library called libc. Instead, Android systems use a library called bionic. Bionic is a modified version of the libc library. [8]
3) **Android Runtime:**
In the Libraries layer, there is the Android Runtime area as a different area. This area includes Core Libraries and Dalvik Virtual Machine.
    - **Core Libraries:** This layer includes the core APIs for Java, network access, file access, data structures, and graphics components.
    - **Dalvik Virtual Machine:** DVM, one of the most important components of Android systems, plays a role in the operation of applications. Each application running on Android systems means a DVM run. It provides minimum usage of memory and Zygote is used for this. It converts these codes to dex format for applications written in Java to work with DVM. [9]
4) **Application Framework:**
The Application Framework layer provides various features for running applications. Application developers offer these features to develop applications. Application Framework offers the following features:
    - **Activity Manager:** It controls all aspects of the activity stack and application lifecycle.
    - **Content Provider:** Allows applications to share their data with other applications.
    - **Resource Manager:** Provides access to built-in resources such as strings, color settings, UI.
    - **Notifications Manager:** Used for applications to show notifications to users.
    - **View System:** It is used to create the user interface.
5) **Applications:**
The Applications layer contains built-in and 3rd party applications that run by taking advantage of the lower layers. [10]

B. Architecture of APK

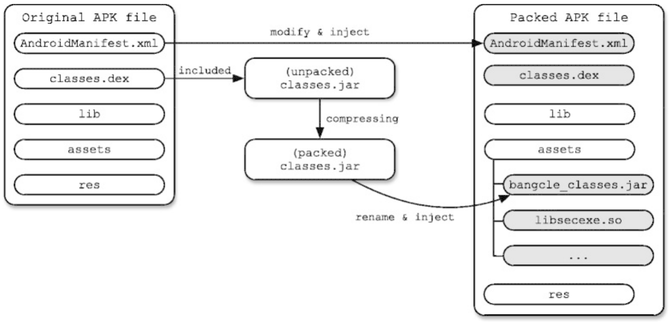

Fig. 2. APK Architecture
[7]

An APK file contains all the necessary files for an Android program. Listed below are the important folders and files you might find in an APK file:

- **META-INF:** This contains the signature file and the manifest file with a list of resources in the archive.
- **lib:** The native library that runs on a particular architecture of the device.
- **res:** The uncompiled source in sources.
- **assets:** Raw files of resources packaged with applications by developers.
- **AndroidManifest.xml:** Gives a detailed account of the APK file's content, version, and name.
- **Classes.dex:** Compiled Java classes to run on the device.
- **Resources.arsc:** compiled resources used by the application, such as strings. [11]

III. Malware Types on Android

Malware is an umbrella term for any malicious software. There are different types of malware on the internet and they can infiltrate a user device in different ways. All malwares are not acting the same and some of them are not as harmful as others. Because of how they work and how they damage the users, they are categorized under different terms. The most of them can be classified as:

A. Adware

Adware means advertisement malware. It's a malicious or unwanted application that contains unwanted ads and shows it to the user. Ads itself can be harmful or not, but they are usually shown to users in inappropriate places. They are usually bundled with other basic apps. Depending on permissions, Adwares can do more than showing annoying ads, like stealing user information or encrypting the user data.

B. Backdoor

Backdoors bypass the system security measures and allow attackers access to the device or the application. Backdoors can be the result of a bug or system vulnerability, can be deliberately created by developers of the legitimate app or created by installing a harmful application. Backdoors are usually rooted to the android devices. After they access the device they usually wait for an execution command that is triggered when a condition is met. They can also be chained with adware to lure users and gain access to the system.

C. File Infecting Viruses

This type of malware that attaches itself to APK files. This apk file can be a legitimate application or game too. They usually use system resources and slow down the device and collect user information like IMEI, status etc. Depending on the application permissions, they can access or modify files. Also if the device is unlocked or not up to date, they can even root the device and get full permissions.

D. PUAs or PUPs

They are potentially unwanted programs that come with legitimate programs. They are usually distributed in the Free or Freemium model. They are not always that harmful depending on their purpose. They can be addressed as Adware if they are showing Ads, if they steal user information they can be called as spyware.They also can act like a backdoor for other malicious applications.

E. Ransomware

Ransomware encrypts a user's data and asks for ransom to decrypt that data. Most of them also make the system not usable after encryption complete. Ransom is usually paid with a crypto payment option even after paying the ransom, the decryption process is still not guaranteed.

F. Riskware

These apps are usually not designed to be a harmful app but because of the critical vulnerabilities they have, they are classified as malware. Some of them don't have public vulnerabilities but they perform risky actions that potentially put the system in danger. They most likely ended up at an entry point for an attack.

G. Scareware

Scareware is a fear coaxer that directs users to call or text to a paid number, pay or download for malicious software. These types of malwares try to get device and user info to give more convincing customized warnings.

H. Spyware

This type of malwares steals user sensitive information and data and sells them to advertisers or agencies. Depending on the permissions they have, they can get as much data as they can(location, network info, passwords,messages etc.). If it's a targeted attack, they can get 2FA keys from sms and access sensitive accounts of the targeted user.

I. Trojan

Trojans are the biggest malware category that represent many malware categories. They can hide in the background and do tasks or act like a legitimate app for phishing attacks. This type of malwares usually deals with delete, blocking or stealing data from services provided by the OS.

IV. Malware Distribution Scenarios

Malwares typically distributed through web pages, emails and messages. This type of distributions need user input and because of android don't allow installing apps from unknown sources and warn the user when they try to enable installing apps from unknown sources option, this makes tricking the target user is not that easy as on Windows. Because of the more strict rules of Android, both Android users and attackers exhibit different behaviors than Windows users and attackers. Because of the more strict policies of Android, attackers have a harder time to distribute their harmful applications, but on the other hand some users may wish to circumvent these limitations and prefer unknown sources to get their applications and as a result, that users can be more vulnerable to attacks.

A. Google Play Approved Apps

Google Play has an automated control mechanism on uploaded applications. This will result in more harmful applications having managed to get in the store compared to Apple's App Store which have manual control mechanisms. Listing in Google Play is not enough byself. The app should be downloaded to devices by users. Because of that this type of attacks can be started with deployment of the non malicious code on store. After the app get enough user base, attackers deploy a malicious update and users get the malware because of Google Play's automatic update feature.

B. 3rd Party App Stores

This type of 3rd Party App Stores are more popular in markets Where Google services are not available. There are big companies other than Google that provide app stores and SDK's too (Samsung, Huawei, Amazon, Tencent etc.) and it can be said that most of them have the same level of security measures as Google Play Store, but the real problem is Apk Mirror stores. They don't have agreements with publishers or developers. Developer or Publisher doesn't have any control of the application on that market. The APK can be legit but can be an old version. The worst situation is that a Modified version of the APK can be uploaded and distributed through that store. This modification can include adding malicious code to a legitimate application.

C. Unknown Sources

APK files can be distributed through direct downloads too. Some users go that way because of their own reasons. Some of the reasons are:

- **Application Availability:** Some applications are available only in Google Play Store and Some Applications are not available in Google Play Stores because they violate the Store rules. The users in these situations are choses to download applications directly from the web. Some apps provide official APKs directly from their website but most of them don't provide such a thing. This results in some users choosing to get APK files from untrusted websites.
- **3rd Party Clients for Legitimate Services:** Some users want to use a service with a different client. Because this type of apps mostly violate the service's rules, distributing from an App Store is not an option. This application can contain malicious codes because there is no security check. Even though 3rd Party Clients are safe and even they are open source, most of the users will directly download a compiled version of this application. And this may result in a potential security risk because the compiled version of the app can be different than the original and contains malicious codes.
- **Modified Apps and Piracy:** Because of android system allows installing APKs from an unknown source, Some users want a modified version of their favorite app or get paid apps for free or unlock paid features for free. This will result in users searching 3rd party websites for these apps. Also because some applications have DRM's, in order to run that pirated apps, they need to be modified by someone. This will result in more security problems and users search deeper on the web to find modified and not signed versions of these apps. Also Android Piracy scene is not in a good state when it comes to reliability when compared to other piracy scenes.

D. Bootloader Unlocked Devices and Root level Users

Some users are using their daily devices with root privileges or using a device with an unlocked bootloader. This will result in a huge security vulnerability because that means a malicious program can get an Root access and do everything with that device hardware, files, applications and data.

E. Application, System and Hardware Exploits

This scenario is usually applied in targeted attacks. Attackers use a vulnerability on an app or system itself. Depending on the scope of the attack it can be a known vulnerability on an old version of a common app or it can be a zero day exploit if the target is high value. This type of attacks usually need minimal user input (in some cases no input needed) and this makes this type of attacks more dangerous.

V. Android Malware Detection and Protection Mechanisms

Protection as the variety and number of Android malware grows the number of methods is also increasing and diversifying. Malware detection and protection systems in the literature Static Analysis Approach, Dynamic Analysis Approach, and Signature-Based Approach are grouped under 3 headings.

A. Static Analysis

This approach ensures that malware detection is done before apps are installed on the device. Thus, the mobile device is not affected by the malicious function of the application. This approach is a fast and inexpensive approach that detects malicious characteristics and bad pieces of code before the application is run. DroidAnalyzer is a static analysis tool that identifies potential vulnerabilities of Android applications and the presence of root privilege exploits. in the study App permissions, risky APIs, and keywords indicating the presence of root privilege exploitation were examined. Risky permissions and keywords were first identified on the data set, then target applications were examined. An algorithm based on the comparison of the MD5 cryptographic hash values of the applications and the doubt levels assigned to the keywords is used. [12]

B. Dynamic Analysis

Burguera and his collaborators have proposed Crowdroid, a framework with components that give enough resources and techniques to identify malware on the Android platform. This application is in charge of monitoring Linux kernel system calls and transmitting them to a central server in preprocessed form. Users will help by contributing anonymous yet behavioral data for each program they use, according to the crowdsourcing principle. The data will be parsed and a system call vector generated for each user interaction in the applications will be handled by the remote server. As a result, for each application utilized, behavioral data collection will be created. Finally, a divisive clustering technique is used to cluster each data set. Small system call patterns could be discriminated between authorized and malicious apps in this way. To develop the normalcy model and detect aberrant behavior in Android applications, the vectors obtained in the previous phase are evaluated and grouped. Crowdroid collects system calls using the Strace utility that comes with Linux. According to the results of the studies, monitoring system calls is a viable way for detecting malware. [13]

C. Signature-Based Approach

The signature based method, which is one of the static analysis methods, is the first detection method used in the field of mobile security. This method varies according to which features of the malware will be used to extract the signature and what algorithm will be used to create this signature. Usually, certain parts of the code are taken and a signature specific to the malware is extracted from them. If an example is given for Android applications, the permissions it uses, the information in the code, the system calls used can be given as examples. After creating a data set extracted from these data, it is calculated according to the selected algorithm and a value that will define the application is obtained.

Generally, hash algorithms are used for signature extraction. However, these algorithms were not successful in the signature extraction process, as they produced very different results in the slightest change in the application. Due to this problem, fuzzy hash algorithms have been used and a relationship can be established between two signature values. But this was not enough either. Since there is a need to calculate other signatures from a piece of signature and to perform correlation calculations in small-sized data, different algorithms should be produced. Algorithms that enable searching on the character strings used in this study and Rolling hash algorithms within this algorithm are presented as a solution. The most important rule in signature algorithms is to ensure singularity for each sample. Otherwise, the algorithm fails. [14]

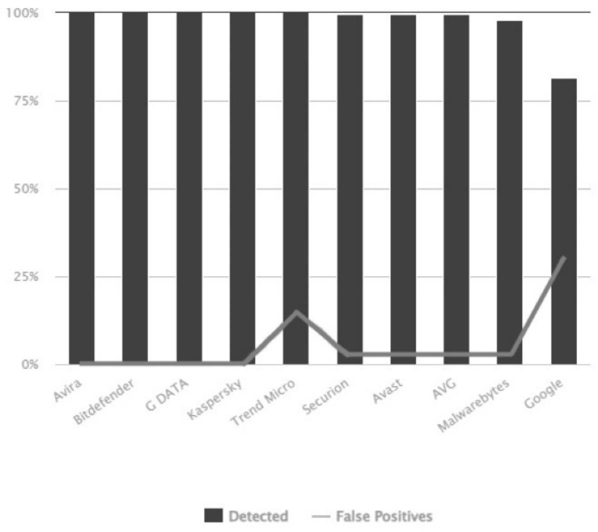

Fig. 3. Antivirus Chart
[5]

VI. Future of Android Malware and Detection

Android becomes more secure everyday and after. More strict security rules and sandbox technologies are the most important factors in this. Also advancement of preventive measures like Archive-Unpacker analysis, Emulations, Heuristic based detection, Machine Learning and Cloud based detection technologies are important factors too. But some of that progression doesn't mean anything to most of the users because of the lack of updates on Android devices. Because the manufacturer stops providing updates to their devices, most android devices, even the popular flagship ones stop getting security updates maximum 2 or 3 years after the launch. That means that devices no longer get patches for critical security vulnerabilities. Because of the system design, every new update including security patches should be optimized by the manufacturer for a specific device before deployment. This is not a healthy business model for Android, because most of the devices are designed to be cost effective, providing long software support has a financial cost to manufacturers and this doesn't suit their business models. Google mandates 2 year security update support for popular phones in the market but this is not enough when we consider both time and device scope of the system. Google works on separating system(feature) and security updates in the system in that way users will now depend on Google instead of the manufacturer about security updates. But this system doesn't mean anything for users who live in markets where Google Services are not available. Also some app developers stop providing updates for older versions of Android and that leaves users who have older Android devices vulnerable to exploits in the old version of that app. Apple's privacy policy influences the industry and this bothers attackers as other data collector companies like Facebook. In the future, Android will adopt such privacy principles and reduce the prevalence of Spyware and PUAs. As time goes by, exploits and zero day attacks will be more valuable than before and will be the main attack vector in the Android Scene.

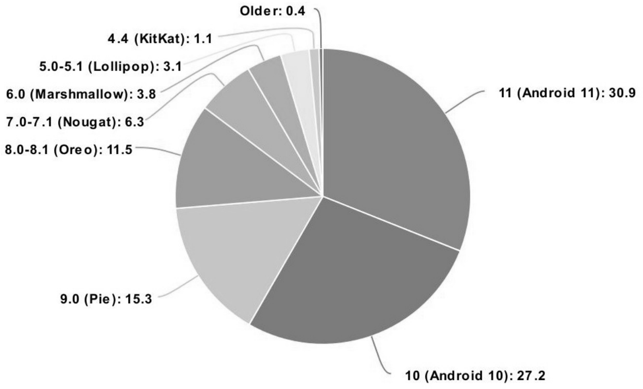

Fig. 4. Andorid Version Chart
[3]

References

[1] Tanweer Alam. Middleware implementation in manet of android devices. *Tanweer Alam." Middleware implementation in MANET of Android Devices.", International Journal of Electronics and Information Engineering*, 12(2), 2020.

[2] Kevin Allix, Quentin Jérome, Tegawende F Bissyandé, Jacques Klein, Radu State, and Yves Le Traon. A forensic analysis of android malware–how is malware written and how it could be detected? In *2014 IEEE 38th Annual Computer Software and Applications Conference*, pages 384–393. IEEE, 2014.

[3] AppBrain.com. Top Android OS versions. https://web.archive.org/web/20211127182910/https://www.appbrain.com/stats/top-android-sdk-versions, 2021. [Online; accessed 27-November-2021].

[4] Saba Arshad, Munam Ali Shah, Abid Khan, and Mansoor Ahmed. Android malware detection & protection: a survey. *International Journal of Advanced Computer Science and Applications*, 7(2):463–475, 2016.

[5] AV-Comparatives. Mobile Security Review 2021. https://web.archive.org/web/20210814055738/https://www.av-comparatives.org/tests/mobile-security-review-2021/, 2021. [Online; accessed 27-November-2021].

[6] Parvez Faruki, Ammar Bharmal, Vijay Laxmi, Vijay Ganmoor, Manoj Singh Gaur, Mauro Conti, and Muttukrishnan Rajarajan. Android security: a survey of issues, malware penetration, and defenses. *IEEE communications surveys & tutorials*, 17(2):998–1022, 2014.

[7] Jongsu Lim and Jeong Hyun Yi. Structural analysis of packing schemes for extracting hidden codes in mobile malware. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):1–12, 2016.

[8] Mahmudur Rahman, Mizanur Rahman, Bogdan Carbunar, and Duen Horng Chau. Search rank fraud and malware detection in google play. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1329–1342, 2017.

[9] Vaibhav Rastogi, Yan Chen, and Xuxian Jiang. Catch me if you can: Evaluating android anti-malware against transformation attacks. *IEEE Transactions on Information Forensics and Security*, 9(1):99–108, 2013.

[10] Sagar Sabhadiya, Jaydeep Barad, and Jaydeep Gheewala. Android malware detection using deep learning. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1254–1260. IEEE, 2019.

[11] Vikas Sihag, Manu Vardhan, and Pradeep Singh. A survey of android application and malware hardening. *Computer Science Review*, 39:100365, 2021.

[12] Mingshen Sun, Xiaolei Li, John CS Lui, Richard TB Ma, and Zhenkai Liang. Monet: a user-oriented behavior-based malware variants detection system for android. *IEEE Transactions on Information Forensics and Security*, 12(5):1103–1112, 2016.

[13] Ke Xu, Yingjiu Li, and Robert H Deng. Iccdetector: Icc-based malware detection on android. *IEEE Transactions on Information Forensics and Security*, 11(6):1252–1264, 2016.

[14] Win Zaw Zarni Aung. Permission-based android malware detection. *International Journal of Scientific & Technology Research*, 2(3):228–234, 2013.