Review Article

SECURITY CONTROLS AGAINST MOBILE APPLICATION THREATS OF ANDROID DEVICES

Authors: Ahmet Efe *, Şerife Özdamarlar

To cite to this article: Efe, A., Özdamarlar, Ş., (2021). Security Controls Against Mobile Application Threats of Android Devices, International Journal of Engineering and Innovative Research, 3(2), p 145-162.

DOI: 10.47933/ijeir.838873

International Journal of Engineering and Innovative Research

http://dergipark.gov.tr/ijeir

SECURITY CONTROLS AGAINST MOBILE APPLICATION THREATS OF ANDROID DEVICES

Ahmet Efe [1*], Şerife Özdamarlar [2]

[1] Dr., CISA, CRISC, PMP, Ankara Development Agency, TURKEY.
[2] Department of Computer Engineering, Yıldırım Beyazıt University, TURKEY.

[*]Corresponding Author: aefe@ankaraka.org.tr
(**Received:** 10.12.2020; **Accepted:** 03.02.2021)

https://doi.org/10.47933/ijeir.838873

ABSTRACT: In the ever developing world of technology, mobile applications are increasing day by day alongside with mobile cyber threats as the new cutting edge technology makes continuous advances. This fact is valid as a result of shifts from e-government to m-government and classical e-business to m-business solutions as a requirement of user friendly and secure mobile technology and applications. The main threat is to the critical and sensitive personal data and information that can be captured by malicious codes and hence dangerous results can be faced. In this paper, malicious software and security techniques of the android mobile applications are analyzed in addition to protection systems from user, developer aspects and even Google Play. The main issue of this paper is providing a current picture of the security concerns of the mobile applications and some sets of counter controls for covering the risks and vulnerabilities of mobile applications in the Android platforms.

Keywords: Cyber security, Android application security, malicious software, mobile security.

1. INTRODUCTION

Mobile device usage is increasing and applications are developed day by day. Mobile device generally is tablet computer and cell phone which is like a small computer instead of only cell phone functionality such as SMS and voice-calls. Banking operations, social interactions, shopping, reading and editing can be examples of these small computers. Privacy analysis or phishing can be made with the data generate through mobile applications. Application permissions are an important problem that threats privacy and business-critical information elements. Some applications request all permissions whether necessary or not and usually majority of users tend to give what they want. There are a little percentage of users who have knowledge about the permission if required and safe or not. Users' personal data which are identity information, photographs, affiliation, location information and messages can be captured with these permissions and attacker can use this information. Because of the fact that android is an open source and it's the top system which are mostly being used by developers and vendors, there are lots of attacks for Android operating system. So the security risk is emerging in the mobile applications and many are not aware of it as is shown in the figure 1.

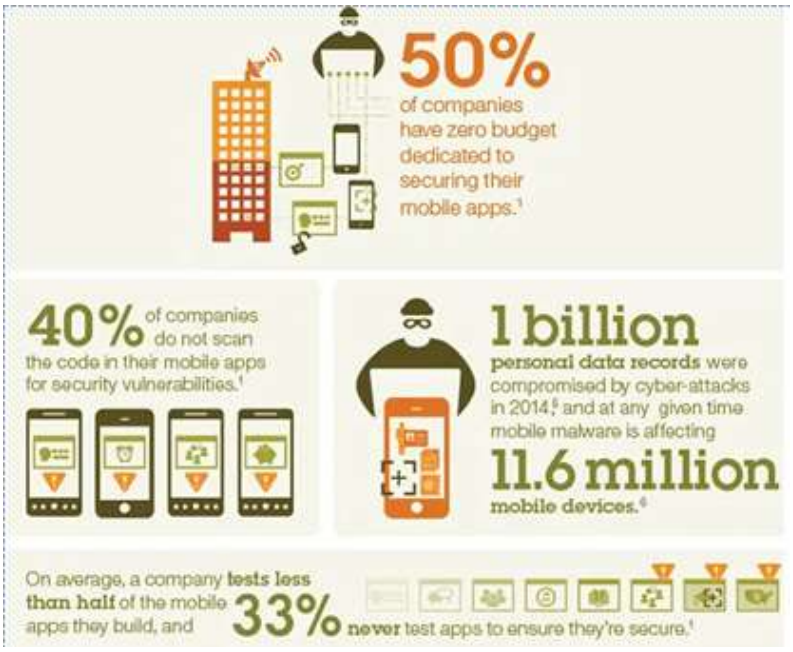

Figure 1. Impact of Weak Mobile App Security [IBM].

Like personal computers, mobile devices run on operating systems with their own vulnerabilities and security issues. Therefore, the increase in mobile device usage has led security experts to improve mobile application security processes while hackers improve their sophistication. Since the secure configuration of mobile devices as hardware is not commercially preferred, data must be protected by certain software on mobile devices and user awareness play a very crucial role here.

Mobile cloud computing is also one of the most important problems of mobile application security that needs to be discussed. Cloud computing on mobile platforms triggers a new wave of evolution as well as new security risks and vulnerabilities in the rapidly developing mobile world. While a few striking studies have been done on the computing counterparts of mobile technology, the cloud computing space for the mobile world seems largely unexplored. The research conducted by Swarnpreet et al. Introduced the Mobile Cloud Computing (MCC) concept, its internal processes and various applicable architectures related to MCC. Cloud computing is computing that provides virtualized IT resources as a remote service using Internet technology. In cloud computing, the user lends and uses IT resources such as software, storage, server, network and security as required, receives real-time scalability support according to the service load, and payments are made accordingly. In particular, the cloud computing environment distributes IT resources and allocates them according to the user's wishes, so some work should be done on the technology that manages these resources and deals with them effectively [1]

The problem faced by software products supporting mobile applications is insurmountable cyber security issues. Specifically, there are three main problems that are most cited:

- A hostile host can send code to another host with undesirable behavior. Currently, there seems to be no way to ensure that a hostile host cannot inject unsafe code into the mobile application system.
- A mobile application cannot be easily protected from a hostile host. Specifically, when a mobile app arrives at a host and starts execution programs, this mobile app is still in the host's compassionate hands. In other words, there is no guarantee that the host will

execute computer instructions accurately and securely. There is not even any guarantee that the host computer will run any particular software; and

- The mobile application cannot be sent or received securely to a host other than a group of trusted computers known as the Trusted Computing Base (TCB).

All these security problems related to mobile applications need to be overcome for mobile applications to be accepted as an alternative to traditional computing systems. Therefore, it is desirable to provide a mobile application security system and method that overcomes the above problems and limitations with conventional mobile application systems. For this purpose, the present technological invention has been directed to use mobile applications in most financial, commercial, administrative and military computer systems [2].

The top ten mobile application risks that are defined by OWASP are as demonstrated in the Fig.2. OWASP is an open source Mobile Security Project and a centralized resource intended to give developers and security teams the resources they need to build and maintain secure mobile applications [3].

Figure 1. Top 10 Mobile Risks in 2014 [3].

However in 2016 the top ten risks have been completely changed. This shows the degree of new technology and the characteristic of concomitant threats that are being renewed by time rabidly. The new ratings are as follows.

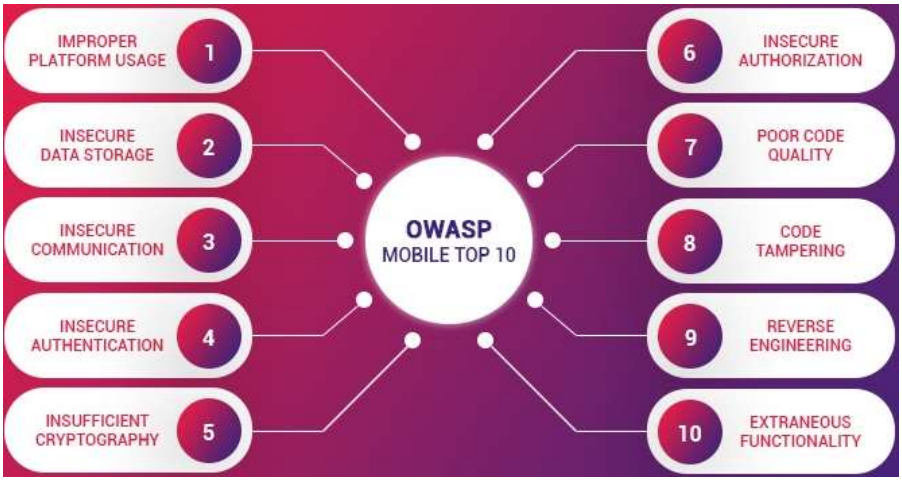

Figure 2. Figure 1.Top 10 Mobile Risks in 2016 [3].

Therefore, M1: Improper Platform Usage[1], M2: Insecure Data Storage[2], M3: Insecure Communication[3], M4: Insecure Authentication[4], M5: Insufficient Cryptography[5], M6: Insecure Authorization[6], M7: Poor Code Quality[7], M8: Code Tampering[8], M9: Reverse Engineering[9] and M10: Extraneous Functionality[10] are considered as the most dangerous threats of mobile applications. However, according to the latest reports, these risks have been completely changed due to new attack vectors that make the most of the newest technology advantages. Here is the new listing:
OWASP Top 10 Vulnerabilities in 2021 are:

1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfigurations
7. Cross-Site Scripting (XSS)
8. Insecure Deserialization
9. Using Components with Known Vulnerabilities
10. Insufficient Logging and Monitoring[11]

Unlike web and desktop applications where system information leaks from outside in more applications than other types of vulnerabilities, more mobile applications contain more system information leaks than any other vulnerability. Internal system holes contain information disclosed to other mobile apps installed on the same system, a much greater concern in the mobile world. When paired with the previous year, several types of vulnerabilities created a Cameo, including Weak Encryption, Insecure Storage: Insecure Deployment: Incomplete Jailbreak Protection and Weak Cryptographic Hash replacing SQL Injection [4].

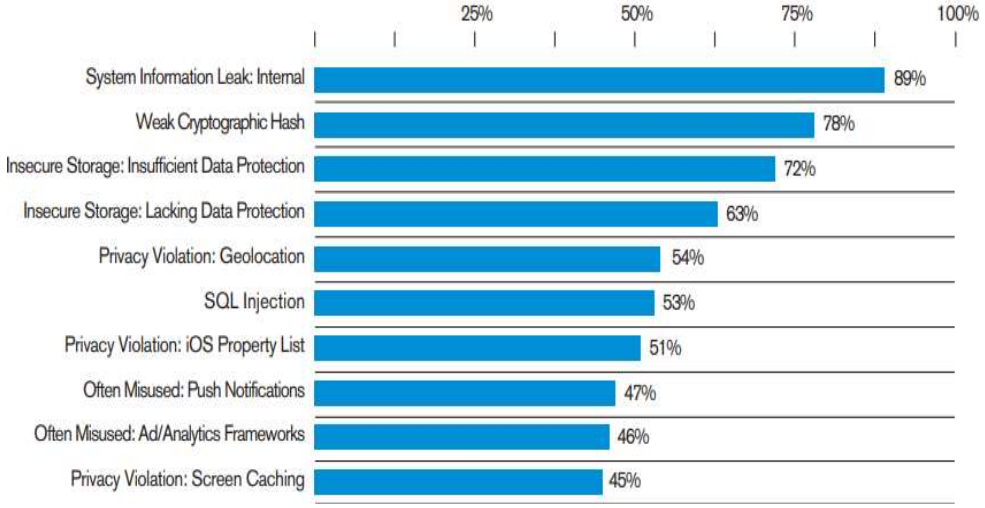

Figure 3. The 10 most commonly occurring vulnerabilities in the mobile applications dataset (Percentage of vulnerable mobile apps) [4].

[1] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M1-Improper_Platform_Usage
[2] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M2-Insecure_Data_Storage
[3] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M3-Insecure_Communication
[4] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M4-Insecure_Authentication
[5] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M5-Insufficient_Cryptography
[6]For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M6-Insecure_Authorization
[7] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M7-Poor_Code_Quality
[8] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M8-Code_Tampering
[9] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M9-Reverse_Engineering
[10] For details see: https://www.owasp.org/index.php/Mobile_Top_10_2016-M10-Extraneous_Functionality
[11] For the details of the data fort he 2021 year see: https://www.immuniweb.com/resources/owasp-top-ten/

In this paper, brief information will be given about risks of cyber-attacks on mobile applications. For this reason, detection and protection of mobile malicious software will be analyzed. Also, the claim which is Android platform is not secure will be refuted with some relevant data from security reports. In addition, some suggestions will be given to application users and developers.

2. RELATED WORKS

The rate of upgrading traditional mobile phones to smartphones is tremendous nowadays, due to the extremely high leap in functionality. One of the most attractive features of smartphones is actually the availability of numerous apps that users can download and install with user-friendly use. However, it also means that hackers can easily distribute malware to smartphones and launch various attacks via social media. This issue can be analyzed with both preventive approaches and effective detection techniques of the latest technology. A study by Daojing He, Sammy Chan, and Mohsen Guizani discusses why smartphones are so easily vulnerable to security attacks. They then presented the malicious behavior and threats of some malware, and then reviewed the existing malware prevention and detection techniques. They point out the efforts of app developers, app store administrators, and users to defend against this type of malware [5].

Sagiroglu and friends represented approaches in the literature and mentioned most important five mobile threats [13]. Also, mobile threats are assessed and some detection and protection methods are represented [6]. Another research about mobile application security is mobile mal-ware detection and protection systems [7]. These approaches bring light to understand detection and protection methods for android. Another approach is cyber security issues in mobile life [8]. Another paper mentions about android based mobile application development and its security. It represents static and dynamic analysis of android applications [7]. This paper also tries to explain android security framework. To understand android security framework is significant to find the causes of security vulnerabilities. Butler represents android phone market growth, android application software, applications and developer, and its security framework [6]. In addition to these approaches, there are mobile security reports. The important report is Android Security Report 2016 which is published by the Google [9].

This report has been published annually. Android security report provides to increase the security of Android. The other report is MacAfee mobile threat report published in 2019. This report handles new threats which affect Android OS. Those who use Android phones that were released before 2012 need to be particularly concerned, as these devices have lacked the security enhancements Google posted and are vulnerable amongst many others to the following 3 the most dangerous vulnerabilities.

1. BlueFrag: A critical vulnerability that could allow the device to be compromised to steal data and spread malware. This malicious code can be sent via the Bluetooth MAC address and users' data can be stolen. According to the security company statement, this BlueFrag vulnerability does not work on Android 10. In other words, there is no security risk from BlueFrag for users using Andorid 10.
2. Stagefright: first discovered in 2015, this vulnerability is used to infect malware via MMS message. As it can be understood, with this vulnerability, the attackers can obtain all personal information by running the malicious codes they want on their Android devices with the MMS they send. The most important issue caused by the vulnerability in question is that these malicious codes can delete themselves from your Android

device, making you not even aware. In addition, hackers can delete this MMS message when requested, and with this vulnerability, they can access the device's hardware, including the camera and SD card.

3. Joker: A vulnerability that appears as an official app on the Google Play store, but allows unauthorized access to the address book on devices when downloaded and used. The android malware called "Joker" was detected in 24 android applications in the Google Play Store and it was stated that these applications were downloaded more than 472,000 in total. The malware discovered by security researcher Aleksejs Kuprins enables users to spend their money on premium subscription services they use. Applications that host this malware secretly click an ad in the background and register on the site to which it is directed.

3. MOBILE ATTACKS

3.1. Cyber Attacks

Cyber-attacks are increasing with developing technology day by day. We can classify cyber-attacks into five groups. First is denial of service attacks and distributed denial of service attacks. Second group is malicious software which are viruses, worms, Trojan horses, key logger, ad words and spyware. Another cyber-attack is phishing. Fourthly, spam is a cyber-attack. At last but not least cyber-attack is listening traffic.

3.2. Mobile Attacks

Malicious software is divided into three groups which are malware, spyware and grayware. Malware can access to device to collect personal data or damage to device. When this software installed to the device, user personal data can be caught by attacker. Also, attacker can have unauthorized access. Another type is spyware. This software collects information which is messages, stored data and location. Spyware can be installed via physical access and personal data stored in this software. Last type is grayware. Grayware is defined as any unwanted software that can cause moderate to severe discomfort for users, including unwanted and unexpected behavior. Unlike a virus, it may not potentially harm the computer. The term grayware can refer to the fine line between a virus and legitimate software. This collects data which is about user. These data are used to marketing and statistical information about users.

4. MOBILE MALWARE

Mobile application area is developing day by day. Because android is an open source and widely used platform; it becomes a usual the target for attackers. In the figure 5 shows rate of using android platform.

The aim of the mobile malware is hacking the device, collecting personal data about user and earning income. Mobile malware has ability which is accessing user phonebook, sending message, remote access and locking the device. $12,000 money was earned without user knowledge according to a report which is published in 2013. There are lots of different mobile threats which are adware, Trojan-SMS, Trojan-banker and Trojan.

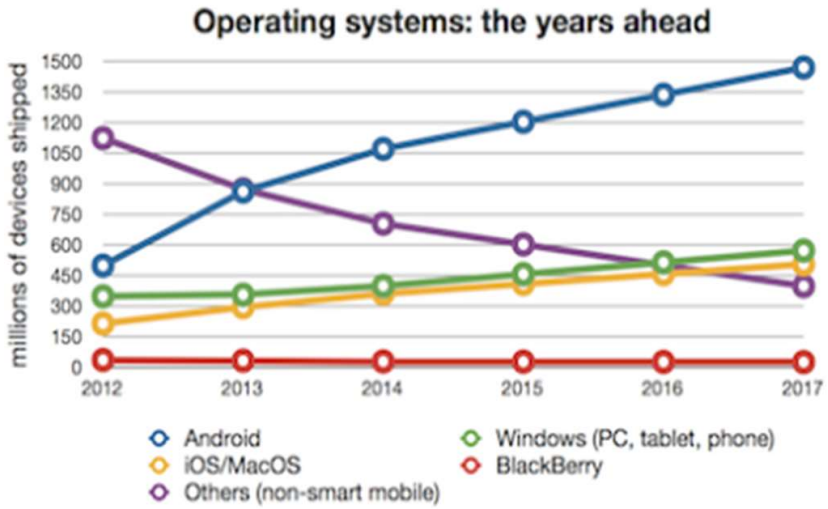

Figure 5. Market share by operating systems [10].

4.1. Adware

Adware is software which gives information about visiting websites, shopping preferences. If this collecting data is occurred without user's knowledge, this software will be malware. In this way, attacker can have information about victim and cheat them using this information.

4.2. Trojan-SMS

Trojan-SMS is the fastest and simple applications in the malicious software. Because SMS feature is placed from classic phones to smart phones, this way gives good solutions for attacker. Also, they earn lots of money via Trojan-SMS. This malware cause spending money via sending message from user phones to paid phone numbers.

Serious security vulnerabilities have been detected in Go SMS Pro, the messaging application widely used. In the vulnerability disclosed by TrustWave researchers, it is stated that the images that users have sent each other are collected by the company on a single server and that this server is configured to be accessed by third parties. It is known that Go SMS Pro, which is offered free of charge on Google PlayStore, has 100 million users worldwide. Go SMS Pro creates a special URL after uploading the files to an online server where everyone can access the images that the people using the application send to each other. Users can see images sent to each other via this URL. The checks made by the researchers also determined that these URL addresses are accessible to everyone and reported that the users' private data may have been disclosed.

4.3. Trojan-Banker

Trojan - Banker is a Trojan horse application to capture information in the online banking process. These applications can be seen harmless and useful by the user but they are prepared to perform illegal operations. Trojan horses use user to multiplication because they have not ability to multiplication. Zeus is the common Trojan - Banker application which can collect data without user knowledge. Trojan - Banker gather information with different ways. The first common way is that mobile device screen is saved when user access to the online banking application. Personal data of the user which are password and customer number with this way. The other way is environment listening via Trojan. While using phone banking, card information can be saved thanks to device microphone and attacker can use this information.

4.4. Trojan

Trojan is containing harmful applications and installing program and it works secretly inside any program. Trojan software is placed to system document of the popular applications. Since their capacities are very little, they take a small space in system documents. Trojan horses have not ability to self-processing. Because of this reason, they need user process. They are used with two different ways in the mobile devices. The first one is like keylogger. They save key motion and keep secret information like password. The second one is capturing session which is user session is captured and e-mail confirmation step can be passed easily while any attack are making.

5. ZERO-DAY EXPLOITS OF MOBILE APPLICATIONS

The cybercrime world is characterized by the rapid discovery and exploitation of any vulnerabilities or problems that may be found in a system or a machine. So-called "Zero Day" attacks are one of the most feared and dangerous security incidents, in addition, around 80% of large-scale attacks that occur are due to Zero Day vulnerabilities detected on hardware or software devices. Considering ever increasing wide range of usage of mobile applications, these types of attacks affect both home users and corporate environments. These attacks have the ability to exploit these identified vulnerabilities and malware variants to exploit for a specific malign purpose.

Vulnerabilities declared on the mobile platforms are available in the Exploit-DB[12] database. For an android application security, the zero-day is a free vulnerability in the Android kernel that could allow a privileged attacker or an application to elevate its privileges to gain root access to a vulnerable device and potentially take full control of the device. These are not only related with Android but also IOS. The most important measure that highlights what someone can (and should) do is to protect mobile device according to latest security tools and mechanisms. However, just installing an antivirus or a complete security solution isn't enough. To get the most out of these tools and to ensure protection, it is important that we know how to deal with the basic risks or at least the most important ones. Another important precaution to be implemented is to keep the software up to date. Both the operating system and the different programs used need security patches against vulnerabilities and discovered for zero-day attacks. Many people have been victims of attacks due to not keeping programs up to date. The complexity of Zero-Day attacks is very high. This is the importance that in addition to people working in technology, all users in general should be alert and take proactive measures. It may not be possible to reduce any type of cyberattacks 100%, but equally, reaching a significant and reasonable level of assurance against them can make a difference.
The table 1 below gives detailed information for a set of declared vulnerabilities of which detailes can be studied in the hyperlinks for each row.

Table 1. Vulnerabilities on EXPLOIT-DB for Android Applications in Hyperlinks.

Date	*Title*
2020-02-24	Android Binder - Use-After-Free (Metasploit)
2020-01-14	Android - ashmem Readonly Bypasses via remap_file_pages() and ASHMEM_UNPIN
2019-11-08	Android Janus - APK Signature Bypass (Metasploit)
2019-10-04	Android - Binder Driver Use-After-Free
2019-05-29	Qualcomm Android - Kernel Use-After-Free via Incorrect set_page_dirty() in KGSL

[12] For detailed information see: https://www.exploit-db.com/

2019-03-06	Android - getpidcon() Usage in Hardware binder ServiceManager Permits ACL Bypass
2019-03-06	Android - binder Use-After-Free via racy Initialization of ->allow_user_free
2019-02-20	Android Kernel < 4.8 - ptrace seccomp Filter Bypass
2019-02-12	Android - binder Use-After-Free of VMA via race Between reclaim and munmap
2019-02-12	Android - binder Use-After-Free via fdget() Optimization
2018-10-08	Android - sdcardfs Changes current->fs Without Proper Locking
2018-09-11	Android - 'zygote->init;' Chain from USB Privilege Escalation
2018-08-13	Android - Directory Traversal over USB via Injection in blkid Output
2018-02-07	Android - 'getpidcon' Permission Bypass in KeyStore Service
2018-01-11	Android - Hardware Service Manager Arbitrary Service Replacement due to getpidcon
2018-01-08	Android - Inter-Process munmap due to Race Condition in ashmem
2017-12-18	Outlook for Android - Attachment Download Directory Traversal
2017-11-28	Android Gmail < 7.11.5.176568039 - Directory Traversal in Attachment Download
2012-12-21	Google Android 4.2 Browser and WebView - 'addJavascriptInterface' Code Execution
2017-02-14	Google Android - android.util.MemoryIntArray Ashmem Race Conditions
2017-02-14	Google Android - Inter-process munmap in android.util.MemoryIntArray
2017-02-02	Google Android - 'rkp_set_init_page_ro' RKP Memory Corruption
2017-02-01	Google Android - RKP Information Disclosure via s2-remapping Physical Ranges
2017-02-01	Google Android - RKP EL1 Code Loading Bypass
2017-02-01	Google Android - Unprotected MSRs in EL1 RKP Privilege Escalation
2017-02-01	Google Android - 'cfp_ropp_new_key_reenc' / 'cfp_ropp_new_key' RKP Memory Corruption
2017-01-26	Google Android - 'pm_qos' KASLR Bypass
2017-01-19	Google Android TSP sysfs - 'cmd_store' Multiple Overflows
2017-01-06	Google Android max86902 Driver - 'sysfs' Interfaces Race Condition
2016-12-29	Google Android - get_user/put_user (Metasploit)
2016-12-20	Google Android - WifiNative::setHotlist Stack Overflow
2016-12-06	Google Android - 'IOMXNodeInstance::enableNativeBuffers' Unchecked Index
2016-12-06	Google Android - Inter-Process munmap with User-Controlled Size in android.graphics.Bitmap
2016-10-12	Google Android - Binder Generic ASLR Leak
2016-10-11	Google Android - 'gpsOneXtra' Data Files Denial of Service
2016-10-03	Google Android - Insufficient Binder Message Verification Pointer Leak
2016-09-27	Google Android 5.0 < 5.1.1 - 'Stagefright' .MP4 tx3g Integer Overflow (Metasploit)
2016-09-14	Google Android - getpidcon Usage binder Service Replacement Race Condition
2016-09-08	Google Android - libutils UTF16 to UTF8 Conversion Heap Buffer Overflow
2016-07-06	Samsung Android JACK - Local Privilege Escalation
2016-06-10	Google Android - '/system/bin/sdcard' Stack Buffer Overflow (PoC)
2016-04-11	Google Android - IMemory Native Interface is Insecure for IPC Use
2016-04-11	Google Android - IOMX 'getConfig'/'getParameter' Information Disclosure
2016-04-01	Google Android - 'ih264d_process_intra_mb' Memory Corruption
2016-03-28	Android One - mt_wifi IOCTL_GET_STRUCT Privilege Escalation
2016-02-08	Samsung Galaxy S6 - 'android.media.process' Face Recognition Memory Corruption
2016-01-26	Google Android ADB Debug Server - Remote Payload Execution (Metasploit)
2014-01-23	GoToMeeting for Android - Multiple Local Information Disclosure Vulnerabilities
2013-11-04	Google Android - Signature Verification Security Bypass
2013-07-03	Google Android - 'APK' code Remote Security Bypass
2015-11-03	Samsung Galaxy S6 - android.media.process Face Recognition Memory Corruption
2013-06-15	TaxiMonger for Android - 'name' HTML Injection
2011-11-03	Google Android 2.3.5 - PowerVR SGX Driver Information Disclosure
2015-09-17	Google Android - libstagefright Integer Overflow Remote Code Execution
2013-01-07	Facebook for Android - 'LoginActivity' Information Disclosure
2015-09-09	Google Android - 'Stagefright' Remote Code Execution
2012-09-12	Google Chrome for Android - Same-origin Policy Bypass Local Symlink
2012-09-12	Google Chrome for Android - Local Application Handling Cookie Theft
2012-09-12	Google Chrome for Android - Multiple 'file::' URL Handler Content Disclosure Vulnerabilities
2012-09-12	Google Chrome for Android - com.android.browser.application Extra Data Cross-Site Scripting
2011-08-02	Open Handset Alliance Android 2.3.4/3.1 - Browser Sandbox Security Bypass
2015-01-26	Android WiFi-Direct - Denial of Service
2014-12-28	WhatsApp 2.11.476 (Android) - Remote Reboot/Crash App (Denial of Service)
2014-11-18	Samsung Galaxy KNOX Android Browser - Remote Code Execution (Metasploit)
2014-06-17	Adobe Reader for Android < 11.2.0 - 'addJavascriptInterface' Local Overflow (Metasploit)

2014-04-15	Adobe Reader for Android 11.1.3 - Arbitrary JavaScript Execution
2014-02-07	Android Browser and WebView addJavascriptInterface - Code Execution (Metasploit)
2008-03-04	Google Android Web Browser - '.BMP' File Integer Overflow
2008-03-04	Google Android Web Browser - '.GIF' File Heap Buffer Overflow
2012-12-09	Google Android Kernel 2.6 - Local Denial of Service Crash (PoC)
2011-03-14	Google Android 2.0/2.1/2.1.1 - WebKit Use-After-Free
2011-02-02	Google Android 1.x/2.x - Local Privilege Escalation
2011-02-02	Android 1.x/2.x HTC Wildfire - Local Privilege Escalation
2010-11-15	Google Android 2.0/2.1 - Use-After-Free Remote Code Execution on Webkit
2009-08-18	Linux Kernel 2.x (Android) - 'sock_sendpage()' Local Privilege Escalation

Source: exploit-db.com

6. ANDROID MALICIOUS SOFTWARE DETECTION AND PROTECTION SYSTEMS

According to InfoWorld, there are three basic security elements in all smartphones. Your first important task as a mobile device user is to be aware of these layers and enable them on your devices:

Device Protection: Allowing remote "wiping" of data in case your device is lost or stolen.
Data Protection: Preventing corporate data from being transferred to personal applications running on the same device or personal network.
Application Management Security: To protect in-application information against interception.

Smartphone security is based on Mobile Device Management (MCY) technology that is installed not only on phones but also on company servers and controls and manages device security. Both have to work together to offer good security [26].

Android malicious software detection and protection systems are developing to get rid of attacks. We can analyze detection and protection system into four groups which are static analysis approach, dynamic analysis approach, signature-based analysis approach and cryptographic data transmission.

6.1. Static Analysis Approach

Static analysis approach provides a control mechanism with data of applications. This control mechanism detects malware and protect device before application installed to the device. Thanks to static analysis, malware detection is provided in the application before installation. We can observe some applications which are developed with static analysis approach for detection and protection. First one is DroidMat. This tool provides detection via API calling for manifest files and related permissions [14]. Secondly, Drebin detects malware with combining static analysis approach and machine learning approach. This tool uses source codes and manifest files of the application. Thus, it capture some data which are permissions which is need by application, API calling and network addressing. Drebin creates a vector with these data and detect malicious software [15].. The final system is Stowaway. This system provides detection of permission which is requested by the application unnecessarily. Stow-away consists of two parts which are determining API calling and matching permissions to APIs and detection permissions which are needed for API calling.

6.2. Dynamic Analysis Approach

Dynamic analysis approach works in runtime differently from static analysis approach [11]. In this approach, we will analyze some applications which are Crowdroid, RiskRanker, DroidMOSS and Paranoid Android. Crowdroid detects abnormal behavior on the Android applications. It classifies Android applications as harmless and malicious. While making this, it uses Strace command which is based on Linux in Android Kernel. Then, Crowdroid compile system callings and classifies the applications. RiskRanker [12] analyses application whether making some dangerous behavior or not. These dangerous behaviors are sending SMS in background and taking high level permission. DroidMOSSS purposes to detect repackaging applications with malicious software in Play Store. Repackaging process is adding malicious software to application in the market. This process can be harmful for user because some popular applications can be dangerous. ParanoidAndroid makes security scanning for android Applications [16].. While making this, it creates a copy for device on the virtual environment.

6.3. Signature Based Analyze and Protection

Applications are kept on the signature database. In this approach, there are server and signature databases. While central server is assigned to analyze and protect processes, database server keeps finding analyses and provides reusing in the next analyses. Some systems were developed in this system. The first one is TractorBeam. System images are created in the central server with TractorBeam and they are analyzed. Analysis application consists of detectors and loggers. Detectors contain malicious software techniques. On the other hand, loggers save activities potential malicious software. Secondly, MADAM is complex detection tool. It detects malicious software in the Kernel and application level. The final but not least system is DroidRanger. It detects new malicious software examples with using schema behavior which is based on permission [17].

PHA category	2016 share in PHA Category	2015-2016 change in PHA installs	2016 percent of total installs	2015-2016 percentage point change of total installs
trojan	77.8%	31.3%	2.58579%	-0.23835
backdoor	8.8%	229.8%	0.29347%	+0.16592
hostile downloaders	3.9%	-94.7%	0.12925%	-3.34500
privilege escalations	3.0%	66.4%	0.09873%	+0.01365
sms fraud	2.9%	108.6%	0.09730%	+0.03043
spyware	1.8%	272.7%	0.06078%	+0.03740
call fraud	0.5%	-61.8%	0.01536%	-0.04227
rooting malware	0.4%	-43.5%	0.01282%	-0.01969
phishing	0.4%	-46.2%	0.01262%	-0.02098
toll fraud	0.3%	-79.7%	0.00893%	-0.05418

Figure 6. Methods to defend against reverse engineering [6].

6.4. Cryptographic Data Transmission

The aim of the cryptographic data transmission is security data transferring and preventing security gaps. Pocatilu maintained an approach. In this approach, uses SMS information, e-mails, files are saved in the database. If saved data is needed by another application, data is taken from database and is decrypted and is transmitted to the application[18]. Android programming application contains javax.crypto package. This package provides symmetric encryption (AES, DES), public key cryptography and message digest classes. In the play store, apk files of the applications are taken and are modified with malicious software. Then, modified application presents with different names on the play store or web pages. There are some tools for modification processes. These are APKTool, smali, dex2jar and JD-GUI [19].

7. FINDINGS AND RESULTS

Android security threat should be analyzed into two groups which are client-side and developer side. There are some security gaps for clients. First security gap is some used applications can access other applications and realize operations on behalf of the user. Secondly, malicious software has ability to install some application without user knowledge. Finally, any application which requests unnecessary permission can be classified as malicious software. Lots of users have not knowledge about permissions and they confirm all permissions which requested from application. They cannot be aware whether these permissions are required or not. In this issue, user should be careful. The second aspect is developer side. While developers are implementing any application, some situations should be considered. First of all, application source code should be kept for security attack because malicious software which is placed into source code can be exposed to attack. The other is using wrong permission request. Another is that deprecated permissions should be removed. Then, using signature and system permission should not be used. In addition, some reviews should not be forgotten while testing process. The final and important problem is that copy-paste code should not be used because some unnecessary permission can be placed.

Protection steps can be classified into two groups like android security threat. First group is user protection steps to avoid installation of potential harmful application. First of all, user should turn off MMS auto retrieval. If you suspect under the threat of Stagefright malware, you should turn off MMS auto retrieval to protect from Stagefrigth. This malware infected from MMS and it take control of user personal data, camera and microphone. Another is updating your device regularly. Lots of update contains security fixes to previously unknown vulnerabilities on your device. Another important protection method is that user should not open messages from suspicious resource because lots of malware infect via message. SMiShing continue to improve phishing. If user is careful to open SMS and clicking on the link, the danger of SMiShing can be decreased dramatically. The final and the most important thing is that user should use comprehensive security software to protect device from cybercriminals. The other protection perspective is developer protection systems. First of all, developer must store sensitive data with encrypted. Developer can use javax.crypto class for encryption of sensitive data. The other is sensitive data should not be stored in the system log. In addition, application backup should be disabled because attacker can access data which stored from application thanks to backup. Another developer protection system is to use secure channel (HTTPs) for external communication. The most important thing is to protect against reverse engineering. Reverse engineering methods can cause a popular application to be embedded malware from

malicious people. In Figure 6, developer should use methods to defend against reverse engineering [20].
Besides the user and developer preventions I mentioned, there are protection methods that offered by Google. Google says there are over 6 billion app installs per day, and each of them is scanned for malware. According to report of Google Android Security 2016 Year in Review, Verify Apps, an app that helps to verify Android's apps, says that the Google Play
Store apps have dropped harmful activity from record level to 2015-2016 [21, 22]. In Figure 7 and 8 show decreasing rate effects of applications.

Technique	Defend Against	Description
Control Flow Obfuscation	Reverse Engineering	Make code flow difficult to follow
Symbol Stripping/Renaming	Reverse Engineering	Remove program symbols from application binaries and rename all human-understandable symbols
String Encryption	Reverse Engineering	Hide clear text strings through encryption
Anti-Debug	Reverse Engineering	Logic detects if debuggers is attached
Checksum	Tampering	Logic detects code/data changes
Self-Repair	Tampering	Logic erases attack changes
White-Box Cryptography	Cryptographic Key Theft	Implement cryptographic algorithms such that keys remain secure even when the code is subjected to white-box analysis
Class/asset/resource Encryption	Tampering/Reverse Engineering	Encrypt assets, resources or classes of the application

Figure 7. Malware effects of applications source from Google Play [9].

Android and Mac OS apps have become an essential element of the busy and daily lives of mobile device users, which is translating into a surge in mobile apps. Now even a new daily user can access a large number of applications through different platforms such as the play store, apple store. Due to certain vulnerabilities of mobile platforms, hackers develop mobile malware which is a threat and therefore the system can be subject to remote control and data privacy loss. Therefore, it is necessary to detect the threat level of a specific application installed on mobile devices and implement the necessary controls [27].

Before an app is available on Google Play, it passes an app review process to confirm that it complies with Google Play policies. Google analyze applications according to developed tools whether potential harmful or not. If an application is marked suspicious, it is sent to security analyst for manual review. After these processes, application is available on Google Play for user.

Google Play

PHA category	2016 share in PHA Category	2015-2016 change in PHA installs	2016 percent of total installs	2015-2016 percentage point change of total installs
trojan	54.2%	-51.5%	0.01623%	-0.01725
hostile downloaders	12.7%	-54.6%	0.00380%	-0.00458
backdoor	11.7%	-30.5%	0.00351%	-0.00154
sms fraud	9.9%	282.2%	0.00296%	+0.00219
phishing	6.2%	-73.0%	0.00185%	-0.00501
privilege escalations	2.5%	-77.6%	0.00076%	-0.00263
toll fraud	2.0%	592.8%	0.00060%	0.00051
commercial spyware	0.4%	-45.3%	0.00012%	+0.00004
call fraud	0.3%	-50.4%	0.00008%	-0.00008
ransomware	0.002%	-92.9%	0.000001%	-0.000009

Figure 8. Malware effects of applications source from outside of Google Play [9].

8. CONCLUSIONS

Businesses tend to rely on mobile devices for critical business operations, collaboration, and access to private data and information. Google continues to invest in innovative technology and artificial intelligence-based resources to further strengthen the security of the Android platform. Android's approach to open-source development seems to be an important part of its security. Developers, device manufacturers, security researchers, SoC vendors, academics, and the wider Android community are trying to create a collective level of competence for the entire ecosystem that finds and mitigates vulnerabilities. With Android, multiple layers of security can enable a variety of users to utilize states of an open platform, while also enabling adequate security to protect user and corporate data. In addition, Android platform security can keep devices, data, and applications safe through tools such as application sandboxing, exploit mitigation, and device encryption. A wide variety of management APIs can provide IT departments with tools to help prevent data leakage and ensure compliance in a variety of mobile usage risk scenarios. Work profiles allow organizations to create a separate, secure profile on user devices where mobile applications and important company data are kept secure and separate from personal information. "Google Play Protect", the world's most widely used

mobile threat protection service, can be provided with built-in protections on every android device. Supported by "Google machine learning", "Play Protect" tends to scan the device to catch, block and eradicate any PHA or malware. "Google Safe Browsing in Chrome" attempts to protect corporate users as they browse the web by warning of potentially harmful sites.

According to the latest statistics, the usage of mobile internet is one of the top trends which is acceding over % 90 of the total internet users. This is an indication that cyber risks and threats will also increase in the mobile world targeting social media and privacy.

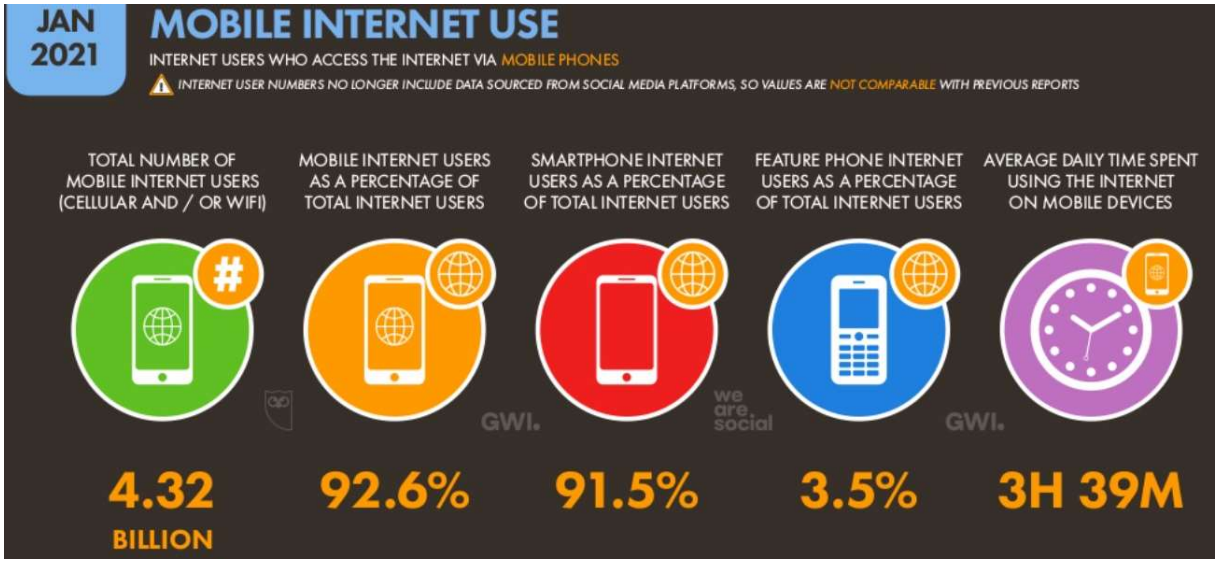

Figure 9. Latest statistics for mobile internet and mobile devices [25].

Like personal computers, mobile devices run on operating systems with their own vulnerabilities and security issues. The increase in the use of mobile devices has led security experts to improve mobile application security processes. Since the secure configuration of mobile devices in hardware is not commercially preferred, data must be protected by certain software and awareness of risk within mobile devices. In software measures, the key can be estimated from the runtime memory preview or the information in the application. Insufficient security measures can lead to situations such as data failure, privacy breach, credit card payment standards violation, identity theft and fraud.

An android application requires certain permissions from the user to access system resources and perform the necessary functions the user wants. Recently, the android market has been showing an exponential growth performance leading to an increase in malware applications. These applications are developed by hackers purposefully to gain access to users' private data and negatively affect the usability of the application through their malicious use. Appropriate tools are urgently needed to detect malware on Android systems, as malware can damage the user's system and data. Since both malware and cleanup security applications require similar types of permissions, distinguishing between them from the user's point of view becomes a very difficult task. A new algorithm should be developed to identify malware-based applications by investigating permission patterns [28]. Since mobile applications make people's lives easier, there are many mobile applications in the markets that are customized and ready to use for people's needs. While app markets provide a platform for people to download apps, it is also a platform that malware developers can use to distribute their malicious apps. Permissions on Android are used to prevent users from installing apps that may violate their privacy by providing warnings to increase their awareness. From the point of view of privacy and security,

it can be well understood if the functionality of applications in Android systems is given in sufficient detail in the descriptions, if the requested permissions are required. This is defined in the literature as permit compliance by definition [29].

If you are a mobile application manufacturer, your competitors' insufficient sensitivity to security measures will play a positive role in your preference with the security measures you increase in your applications. Although it is a reasonable justification for other application developers to prioritize performance in the operation of the process as well as time, and to consider that the usage performance of the application will decrease when the security measures are increased, it is a fact that your applications will remain on the smartphones of the users you are addressing for a long time and their use will be continuous if they like it. If the user is too involved with the application; It will increase the risk of encountering security problems. So, getting your mobile app to get the attention it deserves will not only be a positive improvement, but also bring some risks. Considering that there are too many Android applications on a subject, any security problem faced by users will lead your user to choose the product of the rival application developer from your application.

As a result, android platform is still getting safer and more secure. Although android platform is known as unsafe, Google improve safety level day by day. According to Google Android Report, android application has decreased malware proportion especially within one year. Because increasing of android application safety, android platform can be used mind at peace. Security is an important problem both user and developer. Both of them should be consider security steps to protect personal data.

9. RECOMMENDATIONS

It is recommended to take the following security measures in order to gain a good place in the application market where you will take a place for a long time with the "mobile application security" that you will prioritize:

Protecting Application Integrity against Attacks: The installation files of applications should be prevented from being published in different markets by assuming that attackers can change them. The way to do this is to make application files controllable on the server.

Correcting the Use of Authorization: It is recommended to only include the necessary authorizations while developing an "Android Application ".

Overlooked Errors in Description Lines: One of the most common mobile application mistakes is that the notes taken by the application developer in the description lines and the passwords they use are easily visible later. Therefore, this issue is also among the things that need attention.

Considering Data Storage Sensitivity: No matter if "iOS" and "Android" applications, the data needed to be kept should be encrypted and stored in a suitable folder. It is extremely inconvenient to save sensitive and critical information on any mobile device while the application is in use without any security measure. When it is necessary to keep it, using the password and locked-folder methods effectively will allow you to avoid an important security problem.

Considering Privacy and Personal Data Leak Risks: Confirmation of access to a lot of information such as personal information, phone records, address books and locations of the

users through the mobile application in the first stage of obtaining the product is a widely used data collection method. But there is an overlooked risk factor, which is that personal data accumulated in mobile applications is the target of malicious third parties. Data leakage risks are one of the most common security problems that can be encountered and are a matter of extreme concern.

On the other hand, using broken crypto algorithms, providing data entry from unreliable sources, and keeping the controls performed on the weak server side also constitute important security problems of your application. Effective measures against security problems will play an important role in gaining the trust of users who prefer products in both iOS application and Android applications in mobile market.

ACKNOWLEDGEMENTS

We acknowledge significant contributions of Mr. Volkan Evrin, (CISA, CRISC, CEHv9, CDPSE, COBIT 2019F, ISO 20000-22301-27001 LA) Enterprise Applications and Information Security Manager at KAREL; Part-time instructor at Bilkent University Information Systems and Technologies (CTIS).

REFERENCES

[1]. Swarnpreet Singh Saini, R. B. (2012). Architecture of Mobile application, Security issues and Services involved in Mobile Cloud Computing Environment. IJCER.
[2]. Rygaard, C. A. (2006). Patent No. Mobile application peer-to-peer security system and method. US 7046995 B2.
[3]. OWASP, https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#Top_10_Mobile_Risks
[4]. White Paper of 2017 Application Security Research http://files.asset.microfocus.com/9395/en/9395.pdf
[5]. He, D., Chan, S., & Guizani, M. (2015). Mobile application security: malware threats and defenses. IEEE Wireless Communications.
[6]. Butler, M. (2011). Android: Changing the Mobile Landscape. IEEE Pervasive Computing, 10(1), pp.4-7.
[7]. Holla, S. and Katti, M. (2012). Android Based Mobile Application Development and its Security. International Journal of Computer Trends and Technology, 3(3), pp.486-490. http://ijcttjournal.org/Volume3/issue-3/IJCTT-V3I3P130.pdf
[8]. Gokce, K., Sahinaslan, E. and Dincel, S. (2014). Cyber Security Approach in Mobile Life. 7th International Conference on Information Security and Cryptology.
[9]. Google. (March, 2017). Android Security 2016 Year In Review.
[10]. Intel Security. (2016). Mobile Threat Report Whats on the Horizon for 2016.
[11]. OWASP, https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#Secure_M-Development
[12]. https://mbatraveller.wordpress.com/
[13]. Kabakus, A., Dogru, I. and Cetin, A. (2015). Android Malware Detection and Protection System. Erciyes University Journal of the Institute of Science and Technology, 31(1), pp.9-16.
[14]. M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, RiskRanker: Scalable and Accurate Zero-day Android Malware Detection, in Proceedings of the 10th international conference on Mobile systems, applications, and services - MobiSys 12, 2012, pp. 281294.
[15]. Arslan, B., Gunduz, M. and Sagiroglu,. (2014). Current Mobile Threats and Precautions to Be Taken.
[16]. T. Vidas, N. Christin, and L. F. Cranor, Curbing Android Permission Creep, in In Proceedings of the 2011 Web 2.0 Security and Privacy Workshop (W2SP 2011), 2011.
[17]. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, Android permissions demystified, in Proceedings of the 18th ACM conference on Computer and communications security - CCS 11, 2011, p. 627.
[18]. Dynamic Analysis vs. Static Analysis, Intel, 2013. [Web]. Retrieved from: https://software.intel.com/sites/products/documentati on/doclib/
[19]. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, Crowdroid: behavior-based malware detection system for Android, Science (80-.)., pp. 1525, 2011
[20]. G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, Paranoid Android: Versatile Protection for Smartphones, in Annual Computer Security Applications Conference (ACSAC), 2010, pp. 347 356.

[21]. M. Guido, J. Ondricek, J. Grover, D. Wilburn, T. Nguyen, and A. Hunt, Automated identification of installed malicious Android applications, Digit. Investig., vol. 10, pp. 96104, 2013.
[22]. G. Dini, F. Martinelli, A. Saracino, and D. Sgandurra, MADAM: A Multi-level Anomaly Detector for Android Malware, in Computer Network Security, vol. 7531, I. Kotenko and V. Skormin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 240 253.
[23]. Barrera, P. C. Van Oorschot, and A. Somayaji, A Methodology for Empirical Analysis of Permission-Based Security Models and its Application to Android Categories and Subject Descriptors, in Proceedings of 17th ACM Conference on Computer and Communications Security, 2010, pp. 7384.
[24]. D.-J. Wu, C.-H. Mao, T.-E. Wei, H.-M. Lee, and K.P. Wu, DroidMat: Android Malware Detection through Manifest and API Calls Tracing, in 2012 Seventh Asia Joint Conference on Information Security, 2012, pp. 6269.
[25]. Dataportal, (2021) Digital 2021 Global Overview Report (January 2021) v01, https://www.slideshare.net/DataReportal/digital-2021-global-overview-report-january-2021-v01
[26]. Gruman, Garen (2015) Mobile security: iOS vs. Android vs. BlackBerry vs. Windows Phone https://www.infoworld.com/article/2987635/mobile-security-ios-vs-android-vs-blackberry-vs-windows-phone.html
[27]. Deepa D., Jena S., Ganesh Y., Roobini M.S., Ponraj A. (2021) Threat Level Detection in Android Platform Using Machine Learning Algorithms. In: Mallick P.K., Bhoi A.K., Chae GS., Kalita K. (eds) Advances in Electronics, Communication and Computing. Lecture Notes in Electrical Engineering, vol 709. Springer, Singapore. https://doi.org/10.1007/978-981-15-8752-8_55
[28]. Shrivastava G. Kumar P. (2019) Android application behavioural analysis for data leakage, *Wiley Online Library,* https://doi.org/10.1111/exsy.12468
[29]. Alecakir, H., Can, B. & Sen, S. (2021). Attention: there is an inconsistency between android permissions and application metadata!. *Int. J. Inf. Secur.* https://doi.org/10.1007/s10207-020-00536-1