TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 1, February 2020, pp. 485∼499
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i1.13288

A survey on android security: development and deployment hindrance and best practices

Ratul Sikder[1], Md Shohel Khan[2], Md Shohrab Hossain[3], Wazir Zada Khan[4]
[1,2,3]Department of CSE, BUET, Dhaka, Bangladesh
[4]Department of CS and IT, Jazan University, Jazan, Saudi Arabia

Article Info

Article history:

Received Jun 10, 2019
Revised Dec 21, 2019
Accepted Dec 31, 2019

Keywords:

Android security
Developers guideline
System permission
User privacy
Vulnerability

ABSTRACT

Android OS is the most popular mobile OS for the past few years. Vulnerabilities arise with respect to the increasing functionality of Android OS, impolitic app development practices of developers, end-user incautious and interestingly remediation for the vulnerabilities has been introduced frequently as well. To mitigate security risk factor Google has been updated, deprecated and restricted many system level APIs for 3rd party developers. Considering the consequences, this paper provides a wide overview of Android's system level app development, privacy issues, and guideline for the developers about what measure they should consider while developing apps. We also discussed the historical development of Android OS and the end-users role to maintain privacy and to minimize security risks.

Corresponding Author:

Ratul Sikder,
Department of CSE,
BUET, Dhaka, Bangladesh.
Email: ratulsikder121@gmail.com

1. INTRODUCTION

Current age is considered as the age of mobility. To communicate in a long-distance, we don't have to wait for days or hours; now we can communicate almost in real-time. The rapid and accelerated development in communication technology and mobile devices in recent years have made this possible. From the early nineteenth century to date, the development of mobile devices boosts massively [1]. Earlier, the only use of mobile devices is to talk with someone in a long distance but today's mobile phones specifically smartphones are powerful hand-held computers. Like any traditional computer, every smartphone operates based on its operating system. Android, iOS, Tizen, KaiOS are the major of this kind [2].

Today's smart phone Operating System (OS) allows other software to run on the phone to provide diverse functionalities to the users. It enhances the user experience but security and privacy is the main concern by allowing 3rd party apps on users' private device quirolgico2011vetting. Moreover, unfortunately, security and privacy are not one of the main targets of many small to big 3rd party app developers [3]. As a result, smartphone OS developers naturally don't want to allow 3rd party apps to access root level and sensitive information. Being a flexible smart phone OS at the beginning, Google's Android is also following the restrictive access method. Accessing system-level information, system log and other sensitive information are now being restricted continuously. On our studies, we have found that development of many device optimization and security-related apps had stopped due to permission depreciation.

On the other hand, more problems arise with the non-guided practices by the developers. Developers often don't find the necessity of following the rules and recommendations for developing apps on the mobile platform, and it is very hard to monitor and mine the source code and app behavior to detect unwise programming and harmful activities of the apps. Though the developers, engineers and some machine learning based technologies are always trying to find harmful apps on Google Play Store [4, 5].

There are a few research works on Android's security issues in the app development and adaptation phase. Jha et al. [6] studied on 13,483 real-world Android applications and found only 2,373 apps with no configuration errors; this is a development phase scenario. These security issues become more severe when studies found that security and privacy are not the primary tasks of the developers [3]. Security and privacy are shared responsibilities of both the app service providers and the end users. Usage-pattern and misuse: both intentionally and unintentionally may raise the probability of security threats to the end users. Google's Android help and support center provide some simple guidelines for the Android device users for helping the device and information safe and secure [7].

A detailed survey on application and android ecosystem found some improvements over the traditional software systems [8]. But while improving some aspects of the ecosystem, it has also introduced a new range of problems. Moreover, a detailed analysis of Android and iOS showed that "Privacy by design" is better for mobile platform [9]. This ultimate power of platform should be enforced by the authorities to define and strictly regulate the privacy boundaries. There exists no such survey on these factors where the readers may find the current state of Android security and privacy violation, major changes in Android from the developers' point of view, restrictions and best practices for the developers as well as the users.

We have studied mobile app security related papers and blogs, privacy policies, tested different 3rd party apps, open source projects and analyzed some of the helper classes of Google. There are very few resources on the current system level Android development. The objective of this paper is to summarize our findings from the mobile development history to the present age, including android development hindrance in different aspects, security and privacy issues, guidelines towards safe development and significant facts about the platform as well as the whole ecosystem.

The contributions of this paper are (i) discuss security vulnerabilities and possible solutions, (ii) development restrictions, (iii) recent changes and improvements, best practices for both the developers and users as well as some suggestions for the manufacturers in an organized fashion so that both the Android developers and the users could find it useful in a simpler way.

The rest of the paper is organized in the following sections. In section 2, we briefly describe the mobile device as well as the smartphone operating system development history. Section 3 characterizes the overall scenario of mobile app development, restrictions, and guidelines. Major issues of system level android app development are described in section 4. Best practice for the users to avoid security and privacy threats and recommendations for the developers for safe and secure development are explained in section 5 This section also includes the demand from the manufacturers as the developers' and the users' point of view about which security-related features should be added for the future release of Android. Applications of our findings, discussions as well as concluding notes are expressed in section 6.

2. MOBILE OS DEVELOPMENT HISTORY

iOS is a powerful mobile operating system, developed by Apple Inc. originally unveiled in 2007. It is the second most popular OS till now. At that time, Google was still working on Android secretly; but in November of that year, the company started to reveal its plans for Android and its functionalities. Finally, Android was released at the end of 2008. This is the beginning of today's Android revolution in the smartphone market. Now, Android is the most popular mobile OS worldwide [10]. We will discuss the mobile phone's operating system and the mobile phone development history in the following two subsections.

(a) Mobile devices and pre-android development
(b) Android OS development

2.1. Mobile devices and pre-android development

In 1947, Bell laboratory introduced the term Mobile Network. The first automated mobile phone system for private vehicles launched in Sweden in the year 1956. The device in the car used vacuum tube technology and weighed approximately 40 kg. An engineer of the Soviet Union developed and presented

a number of experimental pocket-sized communications radio in 1957-1961. The weight of one model was only 70 g and it was palm-sized. The first commercial mobile phone was introduced by Motorola. In 1973, they made the first public mobile phone call on a device that weighed 1.1 Kg [11]. In the past 30 years, there were some major changes in the mobile phone architecture. Numerous revolution had been occurring in that era. Later in the late '90s, a closed-source mobile operating system called Symbian has been developed. People in that decade experienced a new form of the mobile OS which was specially designed for multitasking. On the other hand, Series 40 was a software platform, worked as OS, introduced by Nokia in 1999. It was one of the world's most widely used mobile phone platforms but it is not considered as a smartphone operating system because of its limitations. In the meanwhile, Symbian was the most popular smartphone OS until the end of 2010. Obviously, the market was slowly captured by the iOS and the Android from 2007. Timeline 1 shows the historical development of mobile phones, mobile operating systems and finally the development of Android OS till date.

Google had at least two alpha builds of Android released internally before the release of version 1.0 at the end of 2008 [12]. The leading manufacturers, such as HTC, Motorola, Qualcomm, Texas Instruments and carriers including T-Mobile agreed on a formation for future mobile and related software production which is called Open Handset Alliance. For the promotion of the Android platform as a reliable smartphone operating system, OHA members are forbidden from producing devices based on incompatible forks of Android. Now, there are in total 84 firms under this agreement and they contribute to the open standards for the smartphone technology.

TIMELINE 1: *Evolution- Mobile Devices & OS*

1947	AT and T Bell Labs develop the idea of cellular phones
1956	The first automated mobile phone system for private vehicles launched in Sweden
1957-61	Soviet Union developed experimental pocket-sized communications radio
1973	First mobile handset invented by Motorola Inc.
1973-93	Mobile phones use embedded systems to control operation
1994	The first smartphone, the IBM Simon, has a touchscreen, email, and PDA features
1996	Palm Pilot 1000 PDA is introduced with the Palm OS
1998	Symbian OS was developed by Symbian Ltd. The OS was used by phone manufacturers
1999	Nokia S40 and N7110 was introduced officially
2000	Symbian was the first modern mobile OS with the launch of the Ericsson R380.
2008	Android released it's first version internally
2010	Android version Gingerbread released, one of the most popular Android version
2011-14	Android released five popular versions in this era
2015-17	Marshmallow, Nougat and Oreo released. Currently installed on 64.4% android devices
2018	Android Pie, the most recent released OS

Now, the Android open source project is developed and maintained by Google and Open Handset Alliance.

2.2. Android development

From the very initial release, Google has been releasing new versions of Android OS every year, containing major changes in both the base architecture and the user interface.

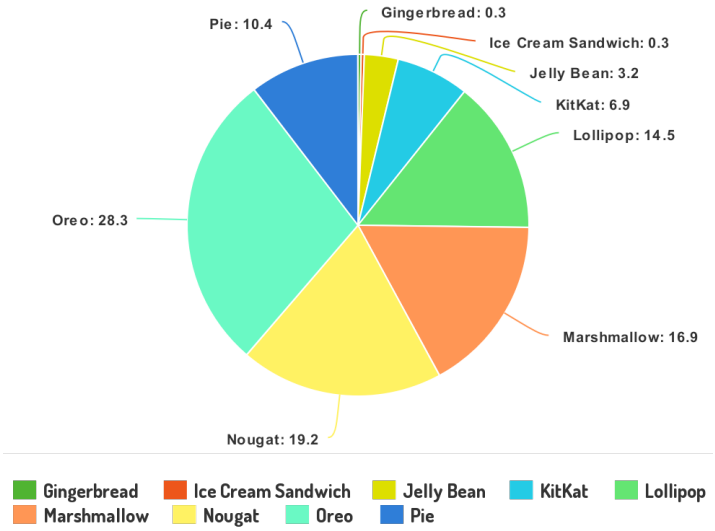

Figure 1. Adaptation of android OS versions(%)

The pie chart in Figure 1 shows the percentage of devices running different versions of the Android platform. In Oct. 2009, a year after the launch of Android 1.0, Google released version 2.0 of the OS. This version adds text-to-speech support and introduced live wallpapers, multiple account support, and Google Map's navigation. This release contains many other new features and improvements. Android 2.3 Gingerbread was launched in Sept. 2010, is currently the oldest version of the most popular versions of Android. Basically, Android started dominating the smartphone market with the huge success of this version of the Android OS. Android 4.0, 4.2 and 4.4 ruled the market for several years and Android smartphone became a strong preference for the general people because of its' functionalities like easy file sharing and backup, enriched app-store, gaming and so on alongside with its comparatively lower price tag. Moving on, Android versions from 5.0 to 9.0 gradually have grabbed around 75 percent of the whole mobile market share [13].

3. BIG PICTURE

Android OS is significantly popular than any other existing smartphone OS. Table 1 shows the smartphone OS market share in 2019. According to Global Stats, Android OS is the dominating smartphone OS with its 75 percent market share [14].

Table 1. Smartphone operating system market share worldwide, 2019 [14]

S/N	Mobile OS	Market Share (%)
1	Android OS	75.33
2	iOS	22.4
3	KaiOS	0.84
4	Windows OS	0.61
5	Unknown	0.36
6	Windows	0.28

The number of active mobile users around the globe is 4.93 Billion [15]. The primary goal of developing a mobile phone was to communicate with others. But, today's phone can do much more. Due to the increasing functionality of mobile phones, vulnerabilities arise which has become a serious concern for both the manufacturers and the developers. Many of the security vulnerabilities are unintentional, e.g., poor programming practices, app developers fail to validate input from the web, allowing adversaries to access the

protected files, etc. Vulnerabilities can also be intentional as well as malicious and can be hidden within a seemingly safe and legitimate app, i.e. a simple paint app asks for internet and GPS access [16]. Security vulnerabilities and unwise programming practices can lead to the following issues:

(a) Users' privacy violation
(b) Performance degradation
(c) Heavy battery drain
(d) Poor end-user satisfaction
(e) Malware, virus, adware attack, etc.

Vulnerabilities result from security threats which are created with the collaboration of a group of hackers and unethical employees. The top security threats are discussed in the following:

3.1. Malicious app

Malicious apps are specially designed to attack smartphone systems. These malware apps significantly relay on the exploitation of OS and software technology of smartphone. We can enlist the malicious apps into the following four categories: [17]

(a) Spyware
(b) Trojans
(c) Phishing
(d) Hidden processes

3.1.1. Spyware

Spyware is unsought software that pervades one's computing devices by robbing his/her internet usage data and other sensitive information like personal information without his/her knowledge of it. Spyware is a kind of malware designed to gain access to one's device. The intentions of using spyware are diverse e.g., for tracking login information, selling internet usage data, capturing credit/debit card information, etc. Some spyware is able to install additional applications and they can change the settings of the victim's smartphone. According to the Norton Cyber Security Insights Report, in the year 2017, nearly 978 million people from 20 countries were attacked by cybercrime and victims lost 172 billion USD globally and spyware caused more damages than other types of malicious app [18].

3.1.2. Trojans

Trojan is a kind of malware that is often faked as legitimate software. Trojans can be devoted by hackers and other cyber criminals to gain access to someone's computing device and can severely damage the system, e.g., deleting, copying, disrupting, blocking and modifying data. Some common form of trojans includes Trojan-SMS, Trojan-Notifier, Trojan-Spy, Trojan-Mailfinder and so on [19]. Sometimes it is hard to ensure the absence of trojans in a system as they may not harm directly to the users rather steal the private and sensitive information silently.

3.1.3. Phishing

It is a type of social engineering attack designed to gain access to someone's private information, e.g., credit card information and login credentials. The cyber thieves accomplish this by misleading people in very convenient ways. For example, one may receive an email which claims that his/her password for a specific website is about to expire within 24 hours and put a fake link which looks very legitimate to renew the password. Once the victim inserts his/her login credentials, the attacker captures the original information and eventually get access to his/her private data.

3.1.4. Hidden processes

These are the applications in which some anonymous activities are embedded without providing any knowledge to the users. For example, a gaming application scans for the nearby wireless devices which is not necessary for any of the gaming functionalities. These types of hidden operations can harm users and user experience.

3.2. Malware downloader

A malware downloader (i.e. trojan downloader) is a harmful application, basically installed by an exploit or some other fraudulent causes like an email attachment or a downloaded image that triggers to install the malicious program onto a victim's computer [20].

3.3. Fake operation

Android OS family is very diverse. There are numerous official as well as unofficial versions of this OS. This open nature of the platform has given the attackers to introduce various fake operations. Faking operator's identity, model, version, software update as well as fake apps' goal, etc are some common examples of fake operations.

3.4. Hidden ads

"It won't hurt if you don't know it." is a common proverb but unfortunately, this phase isn't suitable for today's smartphone security risks. Many of the free apps contain excessive ads that are available in the app store. That is legal because they acknowledge both parties that the app contains ads. But some malicious app contains hidden ads that may be harmful to users. Often these apps cause slowing down the device, sucking mobile data, draining the battery and so on. A recent study has shown that more than 5000 apps of both the major smartphone platforms contain hidden apps. It also causes a huge amount of loss to the advertising organization. They lose about $85 million per year because of the hidden ads [21].

3.5. Premium text

Sometime we may receive some messages from a four or five digit phone number e.g., get jokes for USD1 per month or send STOP to cancel the service. Majority of the users may not activate the service so they are not concern about it but after a month they get a bill of USD1. This unintentional or fake registration to a service is done by some scammers and fraudsters. They sign up for the victim by using the victim's phone number from some websites [22].

3.6. Mobile spy

Mobile spying applications have been developed to monitor child or employee's mobile and tablet usage. Targeted ads are a major source of income for an ad network which may enable this type of attackers to mine the personal information of a user [23]. Table 2 shows the top enlisted mobile threats in 2016-2017 along with their percentages.

Table 2. Enlisted mobile threats in 2016-2017 [26]

Mobile Threat	Percentage
Malicious App	39.2
Malware Downloader	16.1
Fake Operation	5.2
Hidden Ads	4.8
Premium Text	4.1
Mobile Spy	3.2
SMS Blocker	2.3
Mal Dropper	2.1
Downloader	1.7
Dropper	1.7
Fake App	1.7
SMS Stealer	1.7
Rootnik	1.6
Lotoor	1.4
Reg SMS	1.2
Fake Inst	1.2
Hidden App	0.8
Lock Droid	0.8

Being the most popular smartphone OS, Android has a much bigger target for malicious attacks. According to an industry research firm "J. Gold Associates" companies that manufacture Android-powered devices should take necessary measures to make global policies to mitigate security risks that the platform may pose due to its' massive growth [24]. On the other hand, though Google's Android OS is an open-source OS, it has restricted a lot of features for 3rd party developers. The 3rd party developers have minimized developing security related stuff due to Android's core level restrictions. For example, AutoStart restriction has been introduced in Android version 3.1 (Honeycomb), Android 6.0 (Marshmallow) update restricted apps' ability to find the current running task by using the getRunningTasks() API [25].

4. MAJOR ISSUES OF SYSTEM LEVEL ANDROID DEVELOPMENT

Android OS has been developed, changed and modified significantly in recent days. Security-related development, as well as process handling API, is becoming deprecated for the 3rd party developers and Google has clearly mentioned that this is the job for mobile manufacturers [27].

Currently, developing apps for the root level platform optimization is not feasible without root access which is only available for manufacturers and trusted developers. This is mainly due to the continuous permission restrictions in every major release of the Android OS. This domain will be discussed in the following three subsections.

(a) Android security restrictions for 3rd parties
(b) Recent changes in android permission
(c) Feasibility of developing security related apps

4.1. Android security restrictions for 3rd parties

Android is continuously restricting access to 3rd party apps for different types of resources and raw data. Gradually, the majority of the apps developed to serve the purpose of further platform optimization such as battery optimization, security checking, process optimization, etc., had stopped from developing further. Some of the restrictions are shown below.

(a) Limiting directory access: world-writable directories may lead to security weaknesses and enable an application to manipulate trusted files. A proper file scanning scheme is mandatory for a security-related 3rd party app for better threat detection [27].
(b) Logging data: it increases the risk of the exposure of core level system data and reduces system performance upon excessive requests. On the other hand, log information is necessary to gather information about the battery, CPU and network usage which are mandatory for device optimization and security analysts [27].
(c) Device Metadata: Android also restricts access to data that does not seem directly sensitive, but that could lead to revealing characteristics about the user, user preferences, and other stuff [28].

4.2. Recent changes in android permission

Android has changed as well as depreciated some system level permissions and APIs. It has led some old applications to a non-workable state in the newer versions of Android. Figure 2 summarizes the changes in recent Android versions which includes battery and memory optimization, performance enhancement and others.

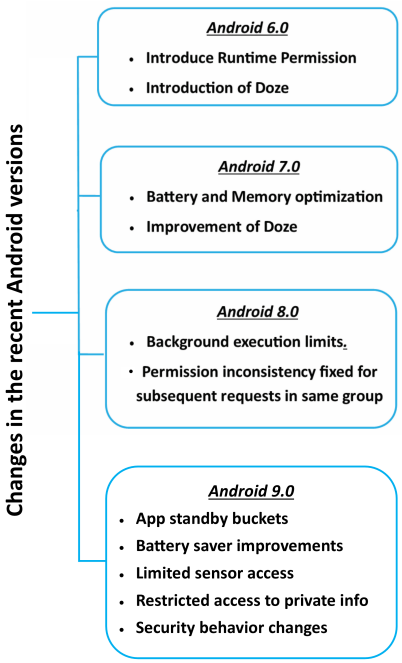

Figure 2. Major changes in the recent android versions

4.2.1. Changes in Android 6.0 (Marshmallow)

Runtime Permission: In Android 6.0, a new mechanism, called runtime permission, is added so that the user can identify what permissions are needed for a particular app and this gives them a chance to review the permissions to judge whether it is worthy or not. This technique reduces security risks via permission violation [29].

Doze and App Standby: If a device is stationary and idle along with the screen off, the device goes into doze mode. It is like a sleep state of the system and app standby technology enables the system to perceive that an app is idle.

4.2.2. Changes in Android 7 (Nougat)

Battery and Memory: In Android 7.0, the system's characteristics was changed to enhance the battery life and the memory optimization of devices.

Improvement of Doze: Battery life is improved by 'Doze' by limiting CPU and network activities. Doze triggers when a user keeps device unplugged; could be moving or stationary, with the screen turned off.

4.2.3. Changes in Android 8 (Oreo)

(a) Background execution limits
(b) Android background location limits

Android Go: Android Go is a lightweight version of Android OS specially designed for low-end devices to run apps and other processes smoothly. The operating system was introduced alongside Android 8.0 (Oreo) and it's based on Android 8.0. This lightweight smartphone OS is optimized to run on smartphones having 1 GB or lower memory and it takes almost half of the storage than the regular Android versions [30].

Android Go devices are 15% faster in terms of apps opening than the regular versions. On the other hand, the apps may be faster and lightweight on Android Go but, they are lack of some features. The positive thing is that the developers can easily optimize their apps for the Android Go platform by following Google's "Building for Billions" development guidelines.

4.2.4. Changes in Android 9 (Pie)

Android 9.0 is the newest version of this OS offered from Google. It introduces a number of changes to the system behavior of the OS. Important behavioral changes are briefly described below [31]. Power management: Android 9 (API level 28) introduces brand new features to improve the power management of Android-based devices. These power management features are two types as follows:

(a) App standby buckets: Analyzing the user's usage pattern, the system limits access to various resources like battery or CPU to the apps.
(b) Battery saver improvements: The existing battery saving feature is improved with a wide area of restrictions.

Privacy changes:

(a) Limited access to sensors in background: In Android 9, apps running in the background can not access the camera, microphone, sensors using the continuous reporting mode as well as on-change and one-shot reporting mode.
(b) Android 9 restricts apps' access to call logs and telephone numbers.
(c) The system restricts access to Wi-Fi location and connection information.

Restrictions on use of non-SDK interfaces: The platform restricts the use of some popular non-SDK methods and fields if the developers attempt to access these directly, via reflection or using JNI.
Security behavior changes:

(a) The system's TLS implementation has experienced numerous modifications in Android 9.
(b) Android 9 additionally restricts the system calls available to apps that use privileged syscalls.
(c) Android secure encrypted files are no longer supported.
(d) Network address lookups can cause network violations that require name resolution. This might invoke network I/O and might be considered blocking actions. This can result in pauses on the main thread.

4.3. Feasibility of developing security related apps

We have studied different types of utility apps developed for mobile device(Android) optimization and battery life enhancement. We have found some interesting facts, that is most of these apps didn't do anything right rather draining the battery and slowing down the performance. Android OS structure has been changing rapidly and the security involvements, process handling API, etc. are becoming deprecated for 3rd party developers. Google has clearly mentioned that this is the job for mobile manufacturers.

5. BEST PRACTICES

The overall security and the privacy of an Android mobile phone user depend on the end user's usage pattern and awareness. The security and privacy related principals and efforts put by the developers in this field is also equally important for ensuring security and privacy. The first party such as manufacturers have a big opportunity to play important roles in delivering the latest security features, patches to the phone quickly. These three portions of security and privacy related practices all together result in greater protection and safety for the end users.

5.1. Best practices for the users

Almost every smart phone user has at least 100 apps on his/her mobile phone according to the survey of Fortune [32]. But, the user doesn't use all the apps. A user may need an application for a while, but after getting the job done, he/she may not delete the application, even though he/she does not require the app anymore. In addition, one may stop using many such applications. These unused applications can be detected as malicious apps [33]. Many of these unused applications may use resources, like the internet, geographic locations, call logs and other user permissions. It is harmful to a user. So, every user should follow some rules to keep his/her devices healthy, and some of the rules are shown in the Figure 3. In the following, we will discuss some important security practices that should be considered for avoiding unwanted vulnerabilities.

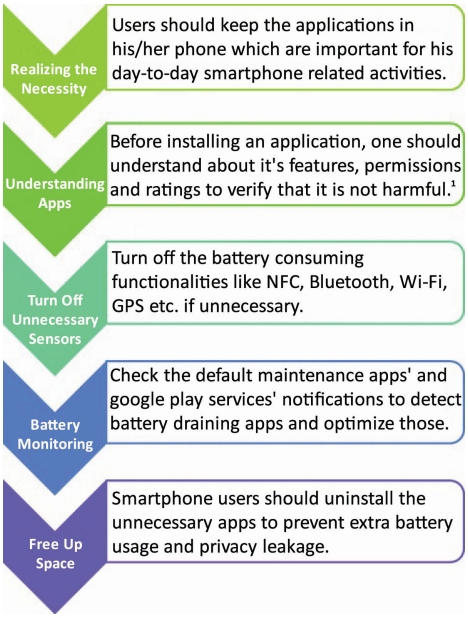

Figure 3. Best practices for users

5.1.1. Software update

Google Inc. and other manufacturers provide system updates which include security patches, features and functionalities, UI improvements and so on to overcome vulnerabilities and to ensure smoother user experience. To get the finest user experiences, it is definitely a good idea to update the phone software regularly. The newest versions of the software help the users to run their phones more smoothly and quickly with minimum numbers of lags and security vulnerabilities.

5.1.2. Installation of applications from untrustworthy sources

Unlike Google, may other third-party app stores never seem to concern about malicious applications on their stores. Many developers also offer beta apps that don't follow some of the Google's guidelines. Hence, it is a good practice not to install apps from any untrustworthy sources as well as the beta versions of smartphone apps.

5.1.3. Understanding app permissions

Android OS is getting better in terms of security enhancement day by day. From Android v6.0, runtime permission request is added. It means the user needs to agree with critical permission(s) during app usage. Though this process is safer than the previous versions' of agreement, people often make mistakes while opening the app for the first time: they often grant permissions without reading and knowing the consequences of it. They also do not check the list of permissions during apps installation. Instead, they just accept the requests without thinking about the consequences. It may be harmful because the developer could take advantages of it.

5.1.4. Data encryption and remote phone wipe

Encrypting the data of a smartphone can help to improve users' security and privacy one step ahead. A user can only get access to the encrypted data with a valid password or key. Encryption of a smartphone enables the needs for a password or key in every boot up of the smartphone. Apart from that, it doesn't change anything how a user uses his/her smartphone. From Android version 6.0, data encryption is enabled by default. Encryption may cause slowing down the performance of some older smartphone, but it doesn't affect today's Android devices. If an application does not meet certain challenges, it should not be installed on phone [34]. Anti-virus provider Kaspersky lab published a report that an IT-based company developed an app called "skygofree" which was released with 48 different commands that was able to risk a user's safety in several ways like it relies on 5 different exploits to achieve root privilege which allows it to bypass security key and it was also capable of location-based recording, capturing image, videos, calendar data and other personal information [35].

Although applying all the security measurement available, it won't feel good if the phone got stolen, unfortunately. If someone faces such circumstances that he/she won't regain his/her device again, it becomes necessary to wipe the phone. Google provides "find my device" option for all Google account linked Android smartphones. Victims need to head over to the "Google's Device Manager" website and log into the Google account and complete the desired operation. If the device has internet access, it is possible to call, set alert, find the exact location, lock or wipe the phone's data remotely from the device manager. The user must make sure that the Google account's password is strong enough so that anyone else can't wipe his/her smartphone.

5.1.5. Lock-screen and biometric scanner

Almost one-third of the total users don't concern about the lock-screen security and they use the traditional swipe-to-unlock method [36]. Though it helps to protect the phone from accidental touches when the phone is in the pocket but the phone can not provide any security barrier if the phone got stolen or compromised somehow. All type of Android smartphone offer PIN, password, and pattern (mostly) to secure the phone which can easily be enabled from the security options in the settings. Additionally, modern smartphones have been introducing biometric sensors like fingerprint sensor, iris, and faceID to enhance smartphone security. Among these multiple biometric-based methods, fingerprint-based biometrics is the most secure way to date.

5.1.6. Online backups

Limited storage on any smartphones could create some troubles, especially on lower storage smartphones. With the increasing media consumption and day to day online interactions and activities, the inbuilt phone storage is getting occupied quickly. This lead to slowing down the smartphone's performance, increase battery consumption and reduce the quality of user experience. A non-limited storage or at least visibly enough storage would solve the problem. On the other hand, if the phone is lost, stolen or damaged then all the important data and media might be lost forever if the data is not backed up. To preserve important data and access them from anywhere and from any device, cloud storage is a proper solution. Many online cloud service provider serves this purpose according to their own structure. For example, one can backup unlimited photos and videos to Google Photos free of cost. Dropbox, Google Drive, One Drive, etc. are offering online cloud storage.

5.1.7. Online password selection and two-step verification

Thousands of smartphone users use easy passwords like 123456, phone number, birth date, and so on to remember it which is very simple to guess for an attacker. So, selecting passwords, especially for online accounts, should not be that much simple and straightforward job. A user should not use a single password to handle all his/her accounts because compromising one account's password can led compromise all other accounts of that user. To minimize the vulnerabilities, the selection of passwords must be based on some criteria. For example, every person should make his/her own reasoning for each password so that every time he/she can put it by remembering the reasoning developed earlier. Additionally, using two-step verification can add extra strength to an account; even if an intruder or hacker achieves the password of a particular account, he/she can't access it without compromising the two-factor authentication media such as cell phone or email account configured earlier for the verification system. So, all the passwords including lock-screen password/PIN, Google accounts, Facebook, Twitter and so on should be selected wisely in order to remember and protect them easily.

5.2. Best practices for the developers

The scenario of Android development is quite different from the earlier versions. Things have changed a lot in recent Android architectures. It seems that new things which were introduced to provide functionalities, nowadays these may cost user's safety. Along with Google, many other companies and researchers work on these issues and come up with several different decisions. That is why many developers, who started developing Android applications before the deployment of Android Studio and Android version 4.4, have faced many difficulties. Moreover, many developers, especially developers from small companies are not interested in reading the privacy policies let alone maintain the policies. A research on app developers found the alarming things [3] as follows.

(a) Developers find privacy policies hard to read.
(b) Writing privacy policies is not considered useful to small developers.
(c) Privacy is not a primary task.

International Association of Privacy Professionals (IAPP) is a web tool that allows the user to compare, read and access ten different privacy policies from the US, Australia, and Europe. After examining the policies, many similarities and overlapping were found and the whole policies can be described with just the four major privacy practices [3] as follows.

(a) Developers should decrease the data to be collected. This will reduce the obligation of developers and save the users from unnecessary data collection.
(b) Old data should not be retained and older unnecessary data should be regularly deleted.
(c) Privacy policies should be enforced to every sort of communication between the two parties.
(d) Developers should encrypt every sensitive data to be stored and all communications should be through an encrypted channel.

Figure 4. Developers' perspective: Guidelines summary [3]

Figure 4 depicts the summary of developers' practice for maintaining users' security and privacy. On the other hand, developers, working on security and core device related apps, are discoursed by Google, i.e., Android Power Usage APIs are not open for the 3rd party developers [37].

Developers can play the most important role in securing the users' activities and privacy as well. Here, we will briefly discuss some important practices that should be abided by the developers to ensure the proper and desired security level and this will define the proper approach throughout the applications lifecycle [38].

5.2.1. Secure the server

Attacking on the server and its API is a common method from the attackers. The developers must secure the corresponding server and API to establish controls and prevent any sort of unauthorized access. Introducing web application firewall and conducting code reviews can help overcome this challenge.

5.2.2. Data encryption

All sensitive data stored on the mobile device should be encrypted. Additionally, the source code and data transmitted between applications should also be encrypted. High-level data encryption always protects valuable data from attackers.

5.2.3. Code obfuscation

It is important to protect the source code from human analyzer and decompiler preserving the operations correctly. This process of code obfuscation not only enhance the security of the app but also provides the confidentiality of intellectual properties.

5.2.4. Strong user authentication system

Two-factor authentication system, wise session management, hashing and encrypting login information can help protect sensitive information. Using advanced authorization tools like OAuth, JSON web tokens, etc. is also essential. These ensure secure and integrated access gateways for the corresponding app.

5.2.5. Regular updation and testing

Hackers always try to find vulnerabilities in apps and exploit them and it is a regular job for the developers to test their apps, repair the breaches and enhance the security. Google regularly updates its software to fix the vulnerabilities of Android platform but its the duty of the developers to find their own faults which may result from poor programming practices, technological changes, and unconsciousness.

5.2.6. Client side data storage

Data stored in smartphone can be exposed if lost or theft. On the other hand, a smartphone might not be secured always because some user roots their device for additional features or more control on the device. Therefore, sensitive data should be stored on the server side.

Alongside with the above recommendation, the Android platform maintainer Google has recommended the following checklist to the developers for securing their apps and sensitive information corresponding to the apps [39].

(a) Enforce secure communication
   - a Apply signature-based permissions
   - b Disallow access to your app's content providers
   - c Ask for credentials before showing sensitive information
   - d Use implicit intents and non-exported content providers
   - e Apply network security measures
   - f Use SSL traffic
   - g Add a network security configuration
   - h Create your own trust manager
   - i Use WebView objects carefully
   - j Use HTML message channels

(b) Provide the right permissions
   - a Use intents to defer permissions
   - b Share data securely across apps

(c) Store data safely
   a Store private data within the internal storage
   b Use external storage cautiously
   c Use scoped directory access
   d Check validity of data
   e Store only non-sensitive data in cache files
   f Use SharedPreferences in private mode
(d) Keep services and dependencies up-to-date
   a Check the Google Play services security provider
   b Update all app dependencies

5.3. Gestures for the manufacturers

When any android phone manufacturers provide an update, they do not provide it for all the previous models that it has produced. However, Apple does not have such problem. When Apple provides a new security patch, is is provided for almost all the models of Apple product. Therefore, android developers, sometimes fail to cope up with every version of Android and they have to think about the same task in different ways for different devices. The possible solution for the problem is to make a sustainable policy for android version upgrade. These policies will be applicable to both the manufacturers and Google to push the update in a defined process. So that the developers don't have to think about handling different API levels.

As Google changes the android's core level security architectures very frequently and they are enforcing more and more restrictions day by day, it is very difficult for other developers to monitor and offer security. Google has already introduced some Google Play Services to monitor apps, harmful activities, malware, etc. But it is not still sufficient to provide full security for the users. Google should take over all the portions of security-related features and developments.

6. CONCLUSION

With the increasing popularity of Android smartphones, proper security of the devices is also becoming a serious concern. As the architecture of the system for smartphones differs somewhere from traditional devices, the existing security system is not enough for securing smartphones. Major market-leading companies don't provide enough flexibility to the 3rd party developers and Google is becoming one of them. Now, the 3rd party developers should not concern about the core security-related process of the Android OS rather than developing utility apps by following platform standards and policies. Platform security and end users' privacy is a shared responsibility among the platform maintainers, the app developers, and the users. All parties should follow the well-defined guidelines, recommendations and take proper responsibility for their actions to ensure a safe, secure and trusted smartphone platform. Android smartphone manufacturers and Google's Android development team should collaborate properly in a defined protocol to ensure security patches timely for all or the majority of the active devices. After all, proper development practices should be spread and made available to the developers by the platform maintainers.

REFERENCES

[1] uSwitch Mobiles, "History of mobile phones," [Online], Available: http://bit.ly/2SfFmwu, Accessed: 11 March 2019, April 2018
[2] ShoutMeLoud, "Top 10 mobile phones operating systems," [Online], Available: http://bit.ly/2Y2KREh, Accessed: 11 March 2019, November 2017
[3] R. Balebako and L. Cranor, "Improving app privacy: Nudging app developers to protect user privacy," *IEEE Security Privacy*, vol. 12, no. 4, pp. 55–58, 2014.
[4] Google, "Help protect against harmful apps with google play protect - google play help," [Online], Available: https://bit.ly/2Lq5vcm, Accessed: 14 March 2019
[5] G. Corporation, "Android: Google play protect," [Online], Available: https://www.android.com/play-protect/, Accessed: 14 March 2019
[6] A. K. Jha, S. Lee, and W. J. Lee, "Developer mistakes in writing android manifests: an empirical study of configuration errors," in *Mining Software Repositories (MSR), 2017 IEEE/ACM 14th International Conference on*, IEEE, 2017, pp. 25–36.

[7] Amy, "Help keep your android device safe - android help," [Online], Available: https://bit.ly/2DPTqa1, Accessed: 14 March 2019, 2018
[8] Y. Acar, M. Backes, S. Bugiel, S. Fahl, P. McDaniel, and M. Smith, "Sok: Lessons learned from android security research for appified software platforms," in *Security and Privacy (SP), 2016 IEEE Symposium on*, IEEE, 2016, pp. 433–451.
[9] D. Greene and K. Shilton, "Platform privacies: Governance, collaboration, and the different meanings of "privacy" in ios and android development," *New Media Society*, vol. 20, no. 4, pp. 1640–1657, 2018. [Online]. Available: https://doi.org/10.1177/1461444817702397
[10] Statistic, "Global mobile os market share 2012-2017," [Online], Available: https://goo.gl/QmcSA1, Accessed: 17 March 2019, January 2018
[11] "History of mobile phones and the first mobile phone," [Online], Available: http://bit.ly/2SfFmwu, Accessed: 17 March 2019
[12] J. Callaham, "The history of android os: its name, origin and more," [Online], Available: https://www.androidauthority.com/history-android-os-name-789433/, Accessed: 24 March 2019, January 2018
[13] StatCounter, "Operating system market share worldwide," [Online], Available: http://gs.statcounter.com/os-market-share, Accessed: 25 March 2019, February 2018
[14] "Mobile Operating System Market Share Worldwide — StatCounter Global Stats," [Online], Available: http://gs.statcounter.com/os-market-share/mobile/worldwide, Accessed: 20 Apr 2019, April 2019
[15] Statista, "Number of mobile phone users worldwide 2013-2019," [Online], Available: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/, Accessed: 25 March 2019, January 2018
[16] S. Quirolgico, J. Voas, and R. Kuhn, "Vetting mobile apps," *IT Professional*, vol. 13, no. 4, pp. 9–11, 2011.
[17] Veracode, "Details on malicious mobile application security," [Online], Available: https://www.veracode.com/security/rise-malicious-mobile-applications, Accessed: 2 Apr 2019, 2017
[18] "What is spyware? And how to remove it," [Online], Available: https://goo.gl/rnXgfp, Accessed: 25 Apr 2019, Nov 2018
[19] "What is a Trojan Virus — Trojan Virus Definition — Kaspersky Lab," [Online], Available: https://www.kaspersky.com/resource-center/threats/trojans, Accessed: 26 Apr 2019, Nov 2018
[20] S. Intellect, "What is a trojan downloader?," [Online], Available: https://bit.ly/2vBlNo3, Accessed: 2 Apr 2019, 2018
[21] G. Tinari, "Cult of android - your phone could be slower due to hidden ads," [Online], Available: https://www.cultofandroid.com/74838/hidden-ads/, Accessed: 4 Apr 2019, July 2015
[22] A. O'Donnell, "How to protect yourself from premium sms text message scams," [Online], Available: https://goo.gl/7sCZ59, Accessed: 14 Apr 2019, March 2017
[23] E. Terkki, A. Rao, and S. Tarkoma, "Spying on android users through targeted ads," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, Jan, pp. 87–94, 2017.
[24] L. Mearian, "Android vs ios security: Which is better?, computerworld," [Online], Available: https://bit.ly/2l2fOqb, Accessed: 17 Apr 2019, August 2017
[25] G. Developers, "Activitymanager, android developers," [Online], Available: https://bit.ly/2O2puLX, Accessed: 18 Apr 2019
[26] Symantec, "Internet security threat report 2017," [Online], Available: https://www.symantec.com/security-center/threat-report, Accessed: 7 March 2019, April 2017
[27] G. LLC, "Implementing security, android open source project," [Online], Available: https://bit.ly/2LoiJpZ, Accessed: 18 Apr 2019
[28] Google, "Application security, android open source project," [Online], Available: https://source.android.com/security/overview/app-security, Accessed: 23 Apr 2019
[29] "Android 6.0 changes," [Online], Available: https://bit.ly/2VHDfG7, Accessed: 26 April 2019
[30] T. A. Authority, "Android go: Everything you need to know," [Online], Available: https://www.androidauthority.com/android- go-773037/, Accessed: 26 April 2019, March 2018
[31] "Behavior changes: all apps — Android Developers," [Online], Available: http://bit.ly/2VyS0LS, Accessed: 27 Apr 2019, May 2019

[32] P. Elmer-Dewitt, "108 apps per iphone, fortune," [Online], Available: http://fortune.com/2011/01/28/108-apps-per-iphone/, Accessed: 6 May 2019, January 2011
[33] I. Singh, S. V. Krishnamurthy, H. V. Madhyastha, and I. Neamtiu, "Zapdroid: managing infrequently used applications on smartphones," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1475–1489, 2017.
[34] G. Dini, F. Martinelli, I. Matteucci, M. Petrocchi, A. Saracino, and D. Sgandurra, "Risk analysis of android applications: A user-centric solution," *Future Generation Computer Systems*, vol. 80, pp. 505–518, 2018.
[35] D. Goodin, "Found: New android malware with never-before-seen spying capabilities, ars technica," [Online], Available: https://goo.gl/LFqXM1, Accessed: 6 May 2019, January 2018
[36] "Best Android security practices," [Online], Available: https://www.androidauthority.com/best-android-security-practices-700393, Accessed: 6 May 2019, Jun 2016
[37] S. Overflow, "Android - battery usage details," [Online], Available: https://bit.ly/2J1uxwi, Accessed: 8 May 2019, 2016
[38] "Android App Security Best Practices To Build Secure Application," [Online], Available: https://www.rishabhsoft.com/blog/android-app-security-best-practices, Accessed: 11 May 2019, Jan 2019
[39] "App security best practices — Android Developers," [Online], Available: https://developer.android.com/topic/security/best-practices, Accessed: 12 May 2019, May 2019