International Journal of Advanced Science and Technology
Vol.105 (2017), pp.37-50
http://dx.doi.org/10.14257/ijast.2017.105.04

Understanding Mobile Apps and Related Permissions for Android Platform

Yatin Jog, Pramod Damle, Abhishek Tambulwadkar and Muthukumar Konar

Symbiosis Institute of Telecom Management,
Symbiosis International University, Pune, India
yjog@sitm.ac.in, pdamle@sitm.ac.in, abhishek.tambulwadkar@sitm.ac.in,
muthukumar.konar@sitm.ac.in

Abstract

There are different categories within mobile applications based on the platform and code. Native apps are designed for a single platform whereas web apps are developed to support multiple platforms. Cross platform apps are designed for specific set of OS and features. Every type of app needs a range of permissions at the OS level to execute the operations of the app. When the app is installed on any device, user has to allow these permissions before start using the app. Many times users are not aware of the permission and meaning of that permission.

Some apps are constantly connected to the Internet, and can upload the personal data--such as private photos or documents--to a remote server without knowledge or consent of the user. While iOS users can generally depend on Apple's app-curating process to keep their data safe, Android users pretty much have to fend for themselves, left to rely on a cryptic system that doesn't seem to be working. According to Joe Keehnast, a product manager for Norton, very few people actually look through an app's permissions before installing it. Even if you were to read through the alert, you may not come away with much information: The permissions list can be extremely unclear and unhelpful.

Security is big concern in allowing apps and related permissions if user don't know the exact meaning of the permission. Android apps need permissions in order to work. However, cybercriminals can exploit them for their personal gain. This paper focuses on understanding different types of apps and related permissions.

Keywords: *Mobile Applications, Apps, Permissions, Security, Android, App Permissions, Group Permissions*

1. Introduction

The advent of cell phone was primarily for wireless voice communication. But over the period mobile phones acquired a lot more features and functions. Mobile phone has been capable of performing almost all tasks done by a computer. The initial mobile phone in 1970s had no screen and was very similar to a cordless phone [1]. With the introduction of the screen and the gradual growth in its size the number of mobile apps for the phone has increased too [2].

Initially the apps were designed only for basic purposes like contact list, call log, calculation, calendar, ringtone selector and editor etc. The Operating system used to be closed and designed by the phone company themselves due to trade secrecy. Games were the first set of popular apps. As internet connectivity was introduced in mobile phones through Wireless Application Protocol (WAP), there rose a need for a browser to surf the web [3]. It was the first step of mobile phones in the direction of being mini hand held computers. It was followed by camera which enabled the entry of photo editing and video players. Soon proprietary mobile platforms and operating systems were introduced which initiated the journey of mobile phones to smart phones. Cheaper memory drives, larger

ISSN: 2005-4238 IJAST

battery life and added on hardware features boosted the app market. The platforms with maximum availability of apps started proving successful.

The open source operating system caused a disruption in app market by overtaking the previously reliable platform. The open source coding platforms encouraged developers to experiment with apps. The process for adding apps to the app market also got easier with minimum stringent. It spurred the developers to integrate social media networks, e-commerce, banking etc. which proved to be the game changers. It enabled users to do all the tasks on mobile phones, thereby reducing the need of computers [4].

The future of app development looks even brighter as big companies have developed apps for their own to promote their products and services along with customer interaction. FINTECH (Financial Technology) has high hopes from app to deliver more value to their business by making monetary transactions quicker and easier through apps [5]. Mobile Apps have managed to penetrate all other industries too, like healthcare, education and transport and will further better other industries too with constant improvisation and innovation [6].

This paper will give an insight on the current structure of development and launching of apps and different permissions related to the apps.

1.1. Native Apps

The native applications are developed specifically for a particular OS. Each OS has a required specific coding to build their native applications. The native applications have the access to all the device interfaces for *e.g.*, camera, accelerometer, GPS location services, contact book, media gallery, etc. This enables the app to be able to be able to take pictures using camera, know the speed of the user carrying the mobile phone, the names in contact book, location of mobile phone user, the pictures received on phone respectively [7-9].

Table 1. Mobile OS Type and Platforms

Mobile OS Type	Platform
Apple iOS	C, Objective C, SWIFT
Google Android	Java (Dalvik VM)
Windows Mobile	C#
Tizen	JavaScript, CSS, HTML5, and W3C widget

The native apps are generally known for faster performance and high responsiveness. Android has the richest diversity in terms of devices manufactured by different manufacturers. Google's Compatibility Definition Document (CDD) defines the protocol for all android device manufacturers. As per Compatibility Definition Document, the measurement of the diagonal of the display of the devices must be at least 2.5 inches while the display size should be at least 320 x 240 pixels (QVGA). The aspect ratio should range from minimum 4:3 to maximum 16:9. Four different classes of screen sizes namely small, normal, large and extra-large have been defined. Additionally, there are four different density classes: Ldpi (120 dpi), mdpi (160 dpi), hdpi (240 dpi) and xhdpi (320 dpi)[9].Other than that the processor capabilities variation in OS versions is also a major challenge for android native app development. IDE used for Android app developments are Eclipse and Android studio [12].

iOS on the other hand has an advantage that the products are limited and hence the range of screen sizes is also limited. Swift by Apple is used to develop native apps for iOS which uses the same compiler, Low Level Virtual Machine (LLVM) as Objective-C.

LLVM transforms the source code of Swift to optimum native source code for all Apple devices. C++ is used for programming LLVM [11].

Native apps are generally developed targeting a single operating system. The developers aim at providing superior user experience (UX) in spite of having to code one application in three different codes which would raise the costs and also more man hours[12].

1.2. Web Applications

A web application is running on a web server out in the cloud and is accessed through a web browser like Chrome, Safari or Internet Explorer. Data is typically saved in the cloud and the application is formatted to be easy to use on a phone or tablet. These are usually accessible from a desktop, laptop and mobile devices.

Mobile Web Applications are developed using JavaScript, CSS and HTML using the browser as the runtime environment and thus make the application flexible for all mobile operating systems. This method of development involves building an app like a website which is interpreted by the mobile browser of the device. The mobile browser of a device is capable of optimising a website as per the needs and specifications of a mobile device. This particular function saves the complexity of developers while designing native apps suitable for every device in spite of having a common platform. The browser standardizes the web application enabling it run smoothly on different devices. Initially the web apps didn't get the permission to use device hardware. Web apps are basically websites wrapped in GUI of a mobile application. They consume data like websites

Progressive Web Application development has overcome those problems. The salient features of progressive web apps are

- Speed-Most of the users quit on an app or website if it takes more than 3 seconds to load. In progressive web apps the key is to minimise or at least optimise the data that downloads on apps of users. Progressive Web apps display improved code efficiency.

- Homogeneity-With the variety in platforms and OS like Android further divided with handsets manufactures by different entities having different specifications and features, the progressive web apps need to run equally smooth on all platforms and devices.

- Applications operating at low bandwidth and high latency are always preferred by users since they can operate under poor network and slow data conditions. Progressive Web Apps use PRPL pattern *i.e.*, push, render, pre-cache and lazy-load where-
    - Critical resources are pushed for initial URL route
    - The initial route is rendered.
    - The remaining routes are re-cached
    - Lazy loading is done so that the object is called only when it is needed.

The PRPL approach provides performance efficiency [13].

Web applications being flexible are cost effective as well as more preferred choice for commercial app. The maintenance is easier compared to native apps but they consume more data too. In terms of offline usage, they have limited capabilities whereas native apps tend to give a richer and a fuller experience to its users. Web Apps are easier to share through social media websites and apps whereas native apps can't be shared across all channels. Web apps also have a better and easier advertisement integration process and can be linked to Google Adwords. Web applications tend to provide a better Return on Investment for a company which floats its app in the market [14].

1.3. Cross-Platform

A cross platform app is where a developer writes a single code for an app. A framework or process is applied on that code to make different apps which are compatible to their specified platforms. Amongst them, we are going to discuss two techniques to convert a single code into platform specific codes in this paper.

1.3.1. Cross-Compilation

Here there is a build environment in which the code is written and there is a target environment which is specific to the required platform. An independent API is used to design the User Interface (UI), data persistence and the business logic using main stream programing languages like Ruby, Java and JavaScript. The written code is then processed through a cross compiler into a platform specific code. This code generated after cross compilation can be deployed like a native app on a device and executed smoothly.

The advantage here is that since the code now runs like a native app, it provides improved user experience and has full access to the device hardware and functions. The cons of cross compilation are that the complexity of the code needs to be consistent with fragmented mobile platforms and operating systems available.

Rhodes Framework, Xamarin, Appcelerator, PhoneGap, MoSync, Whoop are some of the examples of cross platform development frameworks. Appropriate framework needs to chosen based on app requirement. Some are Java based platforms whereas some are C# based platforms and every framework supports different combinations of platforms.

1.3.2. Virtual Machine

Virtual Machine abstracts the details of the target platform from the apps running code. For this the API as well as the runtime environment that the application will run on, both have to be provided. The virtual machine takes time to interpret the data and instructions in the runtime environment from the written code to target platform code, which gives rise to low latency factor. The advantage of this method is that VMs are easy to maintain provide more flexibility in terms of modifying the code in future.

The disadvantage is the latency which could be very low for some complex and heavily coded apps. Also minor adjustments are needed for each platform even after using VM. This method doesn’t support some advanced devices with new functions and features [15].

1.3.3. Hybrid

A hybrid app development method is the one where we combine Native apps and Web Apps technique to build an app which has flexibility of web apps and UX of native apps.HTML5 web apps code is embedded in a native container where WebPages are usually embedded within the apps. This enables them to get an access to the device hardware like native apps.

This method is applied using Hybrid application frameworks like PhoneGap. It involves two-step process. In the first step it provides an embedded web browser that runs the app code and in the second step it provides bridges which allow the web code to escape the browser and get a full access to all native resources of a device.

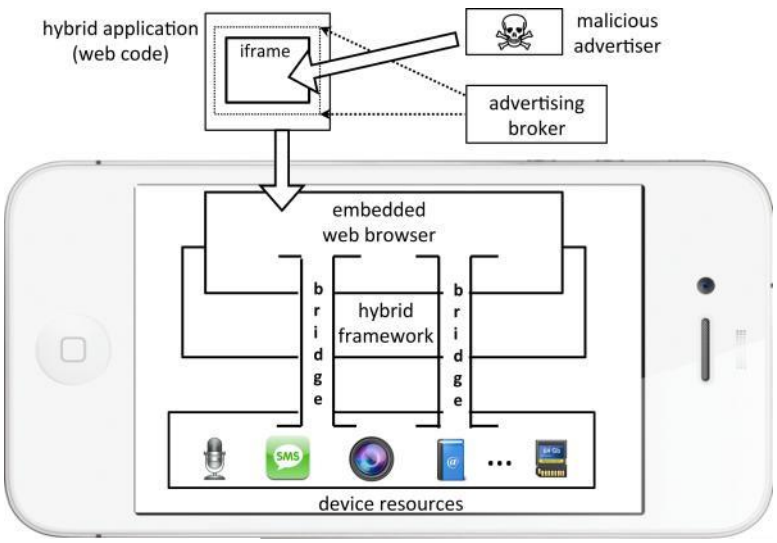

Figure 1. Hybrid Mobile App Framework

A hybrid framework consists of two parts. There is a local half, where the code is written in language specified for the platform for *e.g.*, C# and Objective-C. This code runs as a local process in the device OS and helps in providing actual device access which includes functions like contact list read-write, location services, camera access etc. For execution, the local half creates an instance of a platform-specific embedded Web browser—for *e.g.*, WebView in Android and UIWebView in iOS—and the code of the app is run within the browser. The Web app part of the framework is basically a JavaScript library. JavaScript library which uses its API to access local resources on the device forms the app code.

The advantage of this app building method is that flexibility of a web app is achieved with the UX and accessibility of a native app. Commercial app developers find this method economical since one saves time and money to write dedicated codes for each platform. At the same time the user gets the fine experience of using the native apps which means more responsive and minimal lag. Commercial apps which need camera and other location based services now can use hybrid method instead of web coding which won't get them access to the device peripherals and the complexity in building a separate native app for each platform is also avoided at the same time.

The negative side of hybrid app development is the vulnerability in terms of security. The access policies of web codes and local or native codes are different. Web codes are not allowed an access to any device features which are reserved by local code. The bridges added by the Hybrid framework hold the same access rights as the entire application and aren't protected properly by the same origin policy. This invites fracking attacks where any code in web app format gets an access to the device. If a malicious code is embedded within the web code; it will gain the local access of the device which can cause great damage. It can allow a hacked to pierce through all layers using a web code and hybrid framework bridges and gain access to the device completely. Vulnerabilities are common as far as all hybrid frameworks are concerned [16].

2. Permissions in Mobile Apps

App permissions aren't requests, they're declarations. When you install an app from the Play Store, you'll get a pop up listing all the permissions that the app requires, things like access to your storage, phone calls, network communication etc. Permission is an integral part of an app installation process. Without a proper access to the device peripherals an app cannot function fully.

Third party applications i.e. applications developed by an entity other than the operating service developers seek for certain permissions in order for smooth functioning of their app. Although they are supposed to be harmless hackers and manipulators can misuse these permissions in many ways. Users need to be educated about these permissions and should also know about the negative aspects of these. There are various types of permissions, but we confine ourselves with 15 often-required permissions.

2.1. In-App Purchases

This permission allows the app to access to use Google play to bill you for all the purchases made under paid features of an application. For android most of these purchases are secured by Google but in certain apps, especially games where users buy certain currency or coins Google isn't liable.

For payments not secured by Google the app developer might charge unaccounted fees when given access. Users have to be careful with subscriptions where auto debit is carried every month. Users need to monitor their payments to avoid any such situations. Users must be careful of fake charges and extortions through this permission.

2.2. Device and App History

This permission allows the app to read the sensitive log data, bookmarks, browsing history. It will also let the app gain access to system internal state and retrieve apps. In short it can reveal sensitive information about your device to the app seeking permission.

This permission allows the app to know about other running apps. It is sought by apps which are heavy and need greater RAM in order to run. Social media apps need to know about your browsing history while alternate browsers on phone need them to import the settings and history on their app. An app manager needs this permission to know about the apps which are already installed on the phone. IP based calling apps generally seek this permission so they do not interrupt a voice call on mobile network at the same time cut short an IP call when a call on mobile network is received.

It will be suspicious if apps other than mentioned above ask for this permission. The result of other apps gaining an access to the log data could be misused against the user.

2.3. Cellular Data Settings

This app permission is needed especially for messaging apps which can send and receive data through cellular network. At the same time, they queue up multimedia files like images and videos for downloading on Wi-Fi data.

Although this is a harmless permission, a data guzzling app could use this to upload all accessible information from your phone to its server thus accelerating your usage and threatening your privacy.

2.4. Identity

The identity permission allows aps to access the account information. It is exclusive to Android OS which seeks information about a user's Google account, Facebook, WhatsApp, Skype, DropBox *etc*. It enables the user to login to other apps using their existing Facebook and Google account thus eliminating the need to sign up for that app.

The app gains access to your Facebook and Google account through this app where all the Facebook posts and Google search behaviour of a user could be revealed to the app seeking permission. Users must be careful and not let this permission to be given to an app other than one which needs logins using Facebook or Google.

2.5. Contacts

The contacts permission gives the app full access to the contact list in the device which includes both reading and writing new contacts. Messaging and IP calling apps like Viber and WhatsApp generally use this app. The messaging apps also use this permission to invite people listed in contacts to install their app.

The downside of this permission is that the app gets all the names and contact numbers in your list. This database could be sold or manipulated if such permission is given to a lesser known app. Apps other than messaging and social media apps shouldn't be granted this particular permission.

2.6. Calendar

A planner or organiser app generally would ask for the permission of your calendar. There are some payment apps which also seek calendar permission so they remind you about payments to be made. Generally, a harmful permission, the chances of exploitation of user through this app is very rare.

2.7. Location

This permission as the name suggests, gives apps an access to your location coordinates. Within this, there are two different methods of determining your location, one of which is more precise than the other and more battery intensive.

Approximate location – It determines the location based on Cell ID of mobile towers and WiFi networks which a user is connected to. Comparatively this feature consumes less battery.

Precise location – IT uses GPS/GLONASS as well as the network-based factors above to determine the user location.

The types of apps seeking this permission are as follows-

Maps-The mapping apps are the ones which need this permission primarily to help a user navigate efficiently.

Fitness apps-To determine the distance covered by you while a user runs, walk or cycles.

Food ordering apps– In these apps this permission helps a user, to navigate through all the nearby restaurant options.

Social Media apps-For check-ins the social media apps need the location services of a device.

2.8. SMS

This permission allows app to send messages which would cost money to the user. It also allows the app to read the SMS received and edit them.

Apps use this permission to invite new users via SMS to download and install their app. Similarly, some apps which use OTP (One Time Password) during registration process read the OTP on their own and ease the registration process for the user.

Apps can inflate bills of a user by sending lot of text messages to expand their subscription base. Similarly, apps can access OTPs from other apps especially payment and banking apps which is a very dangerous scenario.

2.9. Phone

The phone permission is useful for all apps that allow you to place and receive calls within an app (think Android dialler apps). Its functions include-

- Call phone numbers (even without your intervention)
- Read and write call logs
- Reroute calls
- Modify phone state

Mostly food ordering apps and third party browsers need this permission to call the numbers with just a touch of a button and thus user doesn't have to punch the entire number again.

A malicious app can take advantage of this permission to call anyone within the contacts or even outside the contact list of the user.

Photos/Medias/Files

This permission allows the app to access the user gallery. It allows apps to:

- Read, modify or delete, multimedia contents of a device such as photos or videos
- To mount, unmount and format external storage of the device which could be an SD card or inbuilt memory of a phone.

Social media and messaging apps generally seek this permission in order to be able to share multimedia files. The apps need access to the media gallery for the same.

Hackers can use this permission to steal pictures and videos and infringe into user's privacy. Hence, users need to be suspicious when a non-social media or non-messaging app asks for this permission.

2.10. Camera

This permission allows the apps to take pictures or videos without sound. Social Media and Messaging apps have the feature to take a picture or a video and directly upload on the user's page. It needs the permission to use camera.

The types of app which need camera permission are-

- Social Media and messaging-In order to click and upload pics instantly on to the app.
- Barcode and QR cod scanner apps-To be able to scan the respective codes, camera access is must.
- Photo Editor-Some photo editing apps also have the feature of clicking pictures directly to be edited further.
- Third party Camera-There are which apps claim to provide better exposure and picture quality and need permission to the native camera for their functioning.

Other apps like Dubsmash need camera to upload video clips in order to synchronise them with voice clips and make funny videos. Although it's a harmless permission the camera can be used by an app for spying purpose.

2.11. Microphone

Apps with voice recording or voice transmission capability need an access to the microphone. Messaging apps which have VOIP (voice over IP) or Wi-Fi calling feature need microphone for seamless voice calling. Apps which have voice command functions also need access to microphone. This permission is comparatively harmless and has no potential threats as such.

2.12. Wi-Fi Connection Information

This permission allows app to gain access to all Wi-Fi information. The app need an access to the Wi-Fi network to send and receive data. Similarly, there are some files queues by messaging app to be downloaded on Wi-Fi network. These files are downloaded after the app senses that the phone is connected to the Wi-Fi network.

2.13. Bluetooth Connection Information

This permission gives apps access to the Bluetooth settings of the device. Further, there are three different kinds of Bluetooth access:

- Simple access-It allows apps to connect to Bluetooth devices which are already paired.
- Admin access – It allows apps to pair and discover other new Bluetooth devices.
- Privileged access –In this the appsgain both the access as mentioned in simple and admin access. Hence it is called privileged access, because no third party app is granted this permission and Google in case of android handles this permission.

2.14. Wearable Sensors/Activity Data

This permission is sought by fitness apps. High end devices have various sensors like accelerometers, gyroscope, proximity sensor etc. The apps to measure speed, direction, light in lumens etc. need access of these sensors.

2.15. Device ID & Call Information

This permission gives apps access to your IMEI (International Mobile Equipment Identity) and IMSI(International Mobile Subscriber Identity). It guards against piracy and helps the app to know if the device is busy with a call.

2.16. Other

All other permissions which do not fall under above mentioned categories are listed under other permissions. These are mostly customised permission which has to go under stringent check of mobile device operating system.

If not thoroughly checked and verified permissions under this category can cause serious damage to the users [17-19]. User's generally treat the permissions list like an EULA, but skipping over these permissions could mean the difference between having your data securely on your device or having all of it at the fingertips of unscrupulous app developers.

3. App-Permission Analysis

Mobile app needs permissions in order to work but cybercriminals can exploit them for their personal gain. Following are the most commonly requested permissions and the way they can be misused [20].

Table 2. App-Permission Analysis Details Table

Permission Name	Type of apps	Meaning	How it can be misused
Network-based Location	location apps, check-in apps	It allows apps to retrieve an approximate location through network-based location sources like cell sites and Wi-Fi. App developers can use it to gain profit from location-based ads.	Malicious apps use it to launch location-based attacks or malware. For example, cybercriminals can direct Russia-based mobile users to malicious Russian language sites.
GPS Location	location apps, check-in apps, social media apps	It grants apps access to your exact location through the Global Positioning System (GPS) and other location sources like cell sites and Wi-Fi. Like network-based location, GPS location can also be used by app developers to gain profit from location-based ads.	Malicious apps use it to load location-based attacks or malware.
View Network State	location apps, check-in apps, social media apps	It allows apps to check for cellular network connections, including Wi-Fi. Apps require network connectivity to download updates or connect to a server or site.	Malicious apps use it to spot available network connections so they can perform other routines, like downloading other malware or sending text messages. Malicious apps can switch on these connections without your knowledge, draining your battery and adding to data charges
View Wi-Fi State	browser apps, communication apps	It gives apps to access Wi-Fi network information, such as the list of configured networks and the current active Wi-Fi network	Cybercriminals take advantage of device bugs to steal Wi-Fi passwords and hack into the networks you use
Retrieve Running Apps	task killer apps, battery monitoring apps, security apps	It lets apps identify currently or recently running tasks and the processes running for each one	Cybercriminals use this to steal information from other running apps. They can also check for and “kill” security apps.
Full Internet Access	browser apps, gaming apps, communication apps, productivity apps	This allows apps to connect to the Internet	Malicious apps use the Internet to communicate with their command centers or download updates and additional malware.
Read Phone State and Identity	mobile payment apps, gaming apps, audio and video apps	It lets apps know if you’re taking calls or are connected to a network. It also gives them access to information such as your phone number, International Mobile Equipment Identity (IMEI) number, and other	Information-stealing malicious apps often target device and phone information.

		identifying information. Apps often use this to identify users without requiring more sensitive information	
Automatically Start at Boot	task killer apps, battery monitoring apps, security apps	Apps use this to tell the OS to run the application every time you start your device.	Malicious apps use this to automatically run at every boot.
Control Vibrator	communication apps, gaming apps	This gives apps access to your device's vibrator function	Malicious apps use it to stop vibrations, which can alert you of premium service notifications or verification text messages before the malicious app can intercept them.
Prevent From Sleeping	audio and video apps, gaming apps, browser apps	It keeps the processor from sleeping or the screen from dimming.	Malicious apps use this to prevent phones from going into sleep mode, so they can continuously run malicious routines in the background. This can also lead to battery drainage.
Modify/Delete SD Card Contents	camera apps, audio and video apps, document apps	This lets apps write on external storage, like SD cards.	Cybercriminals use this to store copies of stolen information or save files onto your SD card before sending them to a command center. Malicious apps can also delete photos and other personal files on your SD card.
Send SMS Messages	communication apps, social media apps	This allows apps to send text messages.	Premium service abusers use this to send messages to premium numbers. This leaves you with unexpected charges. Cybercriminals can also use it to communicate to command centers.
Use accounts on the device	communication apps, social media apps	Lets the app check with built in Account Manager on whether you have any accounts on services such as Google, Facebook and so on.	A malicious app can take advantage of this permission to get your password by phishing you.

4. Group Permissions

Google Play now groups app permissions into groups of related permissions. An app that wants to read your incoming SMS will require the "Read SMS messages" permission. When user install it via the Play Store, it asks for the "SMS" permission group. Install the app and it is giving the access to all SMS-related permissions. The app can now automatically update and gain the ability to send SMS messages without asking the user. If a user have apps on the device that user trust, to read SMS messages, but not send them, now those apps can gain the ability to send SMS messages without prompting the user. The developer just has to update the app. The only way to prevent this from happening is

to disable automatic updates and verify app permissions manually every time an app wants to update

Permission Groups Contain Both Safe and Dangerous Permissions. The big problem is that groups can contain both normal, basic permissions as well as more dangerous permissions. For example:

- Location: An app that asks for your approximate, network-based location can now gain permission to track your exact location with your device's GPS.
- SMS: An app that only needs to receive text messages can now gain the permission to send SMS messages in the background, potentially costing you money.
- Phone: An app that asks to read your call log can now gain permission to reroute outgoing calls and make phone calls without asking you.
- Photos/Media/Files: An app that needs to read the contents of your USB storage or SD card can now format your entire external storage device.
- Camera/Microphone: An app that has permission to take pictures and videos (for example, a camera app) can now gain the permission to record audio. The app could listen to you when you use other apps or when your device's screen is off.

Most of the apps need Internet access these days. User may want to use a live wallpaper, flashlight, or keyboard app without giving the Internet access. One of the security features for third-party keyboards in Apple's iOS is that those keyboards can't access the Internet unless you specifically allow them to. All keyboards on Android can now access the Internet.

Removing the Internet access permission can be one solution. App developers still have to declare that they want Internet access when putting together the app. Users can no longer see the Internet access permission when installing an app but the current apps that don't have Internet access can now gain Internet access with an automatic update without prompting the user.

5. Conclusion

With the increased use of internet data and connecting to public wifi and hotspots, app permissions are playing a major role in securing the device and data from any kind of misuse. Users tend to store all personal data and sensitive information on the deice which can be easily accessed and misused. Whatever content you post online can be used against you. Statements you make can be twisted and used for defamation. Details about your home can give crooks a better picture of how they can break in. Even a very trivial piece of information like your pet's name can be used to hack into your accounts, if you chose that as the answer to a security question. Some users indiscriminately download free apps from app stores. When a game like Angry Birds gets popular, users often just search for it and then hit install without reading through the product page and the permissions they're granting the app. This kind of user behavior often leads to malware infection. Using free apps also raises privacy concerns. Free mobile apps often have ad libraries that collect your data as you tap away on your device. So while you're idly playing your game, your mobile and personal data is being gathered and sent in the background. And often, this form of data collection is legitimate.

- If a user think that the mobile device is secure enough, it needs to be reconsidered. To protect the data in the mobile devices, one has to understand apps, app behavior and meaning of permissions.
- It's important for the user to keep questioning whether each app really needs access to all parts of the phone
- Google may think to revamp the permission system so that it is easier to understand, and app developers have to clearly explain what their apps can access

References

[1] J. L. Funk, "The co-evolution of technology and methods of standard setting: the case of the mobile phone industry", Springer-Verlag, **(2008)**.
[2] L. Findlater and J. McGrenere, "Impact of Screen Size on Performance, Awareness, and User Satisfaction with Adaptive Graphical User Interfaces", University of British Columbia, Vancouver, Canada, **(2008)**.
[3] T. S. H. Teo and S.H. Pok, "Adoption of WAP-enabled mobile phones among Internet users", The International Journal of Management Science, **(2003)**.
[4] D. Dagon, T. Martin and T. Starner, "Mobile Phones as Computing Devices: The Viruses are Coming", IEEE CS and IEEE ComSoc.
[5] D. Lee KuoChuen, "Emergence of FinTech and the LASIC Principles, SimKee Boon Institute for Financial Economics", Singapore Management University.
[6] M. N. Kamel Boulos, S. Wheeler, C. Tavares and R. Jones, "How smartphones are changing the face of mobile and participatory healthcare", BioMedical Engineering OnLine, **(2011)**.
[7] A. Charland and B. LeRoux, "Mobile Application Development: Web vs. Native", Communications of the acm (Vol 54/No5), doi:10.1145/1941487.1941504, **(2011)**.
[8] H. Kang and K. Kim, "Introduction to TIZEN SDK, S-Core", Samsung, Linux Foundation Collaboration Summit, **(2012)**.
[9] A. Holzinger, P. Treitler and W. Slany, "Making Apps useable on multiple different mobile platforms: On interoperability for business application development on smartphones", Institute for Medical Informatics, Statistics & Documentation (IMI) Research Unit Human-Computer Interaction, **(2012)**.
[10] Google Inc (2016), Android 7.1 Compatibility definition, http://static.googleusercontent.com/media/source.android.com/en//compatibility/android-cdd.pdf.
[11] C. González García, J. Pascual Espada, B. Cristina Pelayo G-Bustelo and J. Manuel Cueva Lovelle, International Journal of Artificial Intelligence and Interactive Multimedia, vol. 3, no. 3, **(2015)**, DOI: 10.9781/ijimai.2015.3310.
[12] E. Marinho and R. Ferreira Resende, "Native and Multiple Targeted Mobile Applications", Conference: Proceedings of the 15th international conference on Computational Science and Its Applications - Volume Part IV, At Banff, AB, Canada, DOI: 10.1007/978-3-319-21410-8_42, **(2015)**.
[13] Google Developers, Progressive Apps, https://developers.google.com/web/progressive-web-apps/, **(2016)**.
[14] R. Doom, "When to build a web application vs a mobile application", Web Ascender Blog, http://www.webascender.com/Blog/ID/562/When-to-Build-a-Web-Application-vs-a-Mobile-Application#.WIU1qht97IV, **(2014)**.
[15] B. Nielsen, "Cross platform mobile development", Master's Thesis, Department of Computer Science, University of Aarhus, **(2015)**.
[16] M. Georgiev, S. Jana and V. Shmatikov, "Breaking and Fixing Origin-Based Access Control in Hybrid Web/Mobile Application Frameworks", Internet Society, NDSS Symposium, ISBN 1-891562-35-5 http://dx.doi.org/10.14722/ndss.2014.23323, **(2014)**.
[17] Google Developers, API Guide-Permissions, https://developer.android.com/guide/topics/permissions/index.html, **(2016)**.
[18] Apple Inc. white paper, iOS Security-9.3 or later, https://www.apple.com/business/docs/iOS_Security_Guide.pdf, **(2016)**.
[19] M. Kolenkina, "What settings or permissions do I need to enable on my iPhone?", Whispersystem.org, https://support.whispersystems.org/hc/en-us/articles/213133077-What-settings-or-permissions-do-I-need-to-enable-on-my-iPhone-, **(2016)**.
[20] Trend Micro, Android app permissions, Threat encyclopaedia, http://about-threats.trendmicro.com/us/library/image-gallery/12-most-abused-android-app-permissions, **(2016)**.

