ISSN (Online) : 2455 - 3662
SJIF Impact Factor :5.148

EPRA International Journal of

Multidisciplinary Research

Monthly Peer Reviewed & Indexed International Online Journal

Volume: 5 Issue: 5 May 2019

Published By :EPRA Publishing

Volume: 5 | Issue: 5 | May 2019 || SJIF Impact Factor: 5.148 ISSN (Online): 2455-3662

EPRA International Journal of
Multidisciplinary Research (IJMR) *Peer Reviewed Journal*

AN ANALYSIS OF ANDROID SECURITY: WHAT MIGHT HAPPEN WHEN YOU GRANT AN APP TO ACCESS YOUR PHONE?

Rolando Real Codilan
Faculty,
College of Computer Studies
Eastern Samar State University Main Campus,
Borongan City Philippines

ABSTRACT

The Android permission system is intended to inform users about the risks of installing applications. This study inspected the source code of 3 android applications. Findings show that the permissions declared by the 3 android applications in the manifest are the same with the permissions required before installing those 3 android apps, only the Ringtone Application used those permissions declared in the manifest on its actual project code. While the Ghost Hunter and Kids Memory have not used those permissions declared in the manifest on its actual project code.

KEYWORDS – ***Android security, Android apps, risk of installing an application***

INTRODUCTION

The Android operating system is the most accepted for mobile devices and tablets in the four corners of the world. In the study of Neil Dupaul (2012), Security is a major part of any Android device. Android was created with openness in mind, and is conducive to the use of third-party applications and cloud-based services. Android seems to be a secure and usable operating system for mobile platforms. Securing mobile devices has become increasingly important in recent years as the number of devices in operation and the uses to which they have expanded dramatically [2]. The problem is compounded within the enterprise as the ongoing trend toward it. Android devices are open to a number of different threats, most of which come from downloaded apps. During installation, each app requests specific permissions, such as the ability to access the contacts list or to open websites. The Android operating system tells the user what systems and data an app will access, but the OS will not block any app activity after installation.

According to PatrickGage Kelley,(2012) When a user tries to install an android application (or an app), a list of permissions required by that app is shown to the user before initiating the installation process. Android asks the user if he or she wishes to continue installing the app and grant those permissions to that app. Most casual users, however, are not too interested in those permissions[3]. Recent studies have shown that the majority of users tend to ignore permission warning messages at installations time. Warning messages pop up on the screen when users have already decided to install an app; at that stage, users probably just want to continue with the installation without being interrupted. Furthermore, in the study of Andrienne Porter Felt(2012), When a user installs an application, he

or she has the opportunity to review the application's permission request and cancel the installation if those permissions are excessive or objectionable [1]. Even for users who pay careful attention to permissions being requested, permissions descriptions are often confusing and are hard to understand. This is a huge concern because more apps are increasingly asking for access to sensitive information on your phone to function properly.

The main focus of this study is to examine the source code of an Android App. This study downloaded three android application from a third Party App Store. The study inspected three android application manually and comprehensively.

OBJECTIVES

This study aimed to:

1. Examine the source code of three Android Application.
    a. Ringtone Application
    b. Ghost Hunter
    c. Kids Memory Game
2. a. Determine if the permission asked shown to the user before the installation process was the same in the android manifest.

b. Determine whether the permission declared on the manifest was used in the actual part of the project code of the application.

II METHODOLOGY

Data Gathering

This study focused on downloading the three android applications. The study downloaded it from www.Greenhotworld.com and these applications were: Ringtones Application Template; Kids Memory Game, and; Ghost Hunters.

Installation of Application

This study installed three android applications. During that time, the study took some screenshots of the permissions that the applications asked before installing the three android application. These can be seen in the Results and Discussion part of this paper.

Research Design

The study was an analysis design, whose main objectives were to: examine each source code of the three android applications; determine if the permissions required were shown to the user before initiating the installation, and; determine if the permission requested on the manifest was used in the actual project code.

The study chooses 3 android application based on the availability of the source code and apk file. The android application named as follows: Kids Memory Game, Ghost Hunters & Ringtone Application. The study used the software application notedpad++ to analyze the source code of the 3 android application, compared the android manifest and actual source code of the application. The study compared it based on the permission asked before the installation process and the permission declared on the manifest. This study opened the source code of the 3 android application using notepad++ and manually analyzed it line by line. To test the Ringtone Application which was installed on an android phone and connect to the firebase database and to know what happens to the contact data of the user.

III RESULTS AND DISCUSSIONS

Based on the assessment of the source codes of the three android applications, Tables 1. Shows the application name, the number of classes and the number of lines used in the application were analyzed in this study. In the Ringtone Application, there were 13 classes and 2,242 number of lines of code used by this application. For the Kids Memory game, this game used 10 classes and 1,401 number lines of code. The Ghost Hunter used 15 classes and 1233 number lines of codes. After analyzing the source code of three android applications, this study compared the permissions declared on the manifest and the permissions asked before the installation process. This was also to determine if the permissions declared on the manifest were used in the actual part of the source code.

Table1. Summary of the source code of 3 android application

Application Name	Application Category	No.of classes	No. of lines
Ringtone Application	Availability of source code and APK file	13	2242
Ghost Hunter	Availability of source code and APK file	10	1401
Kids Memory Game	Availability of source code and APK file	15	1233

The table above shows upon installing the 3 android application, some permissions were requested by the android systems, these permissions appeared in Table 2. This study compared those permissions with the permissions declared on its manifest. It was shown that all permissions that asked before installing this application were similar to the permissions declared on the manifest. However, five out of seven of those permissions were dangerous and the other two are normal permission. First, the *read phone status identity* allows read-only access to phone state including the phone number of the device, a list of any phone accounts registered on the device. Second *Read Contacts,* allows an application to read users contact data. Third, *Write Contacts* allows an application to write users contact data. Fourth, *Write External Storage,* it allows an application to write to external storage. Fifth *Write Settings* allows an application to read or write the system settings. Sixth, the *Internet* this permission allows the application to open network sockets. This permission was used when your application wants to connect the internet. The last was *Access Network State* this permission allows the application to access information about networks. This was used when you check the network status whether you are connected or not to a network.

Access Present	Present in Manifest? YES/NO		
	Ringtone Application	Kids Memory Game	Ghost Hunter
Read phone status and identity	YES	NO	NO
Modify your Contacts	YES	NO	NO
Modify SD card	YES	NO	NO
Find account on the device	YES	NO	NO
Full network access	YES	YES	YES
Network connection	YES	YES	YES
Modify system settings	YES	NO	NO

Table 2. Summary of Android manifest and permission of 3 android application

The table above shows the device Second *Read Contacts,* allows an application to read users contact data. Third, *Write Contacts* allows an application to write users contact data. Fourth, *Write External Storage,* it allows an application to write to external storage. Fifth *Write Settings* allows an application to read or write the system settings. Sixth, the *Internet* this permission allows the application to open network sockets. This permission was used when your application wants to connect the internet. If not allowed, a mobile app won't be able to connect the internet. The last was *Access Network State* this permission allows the application to access information about networks. This was used when you check the network status whether you are connected or not to a network. Moreover, you can observe that all permissions asked will be shown to the user before the installation process was the same in the android manifest.

Figure 1. Comparison of Android manifest and permission Ringtone app

Figure 2. Comparison of Android manifest and permission Kids Memory Game

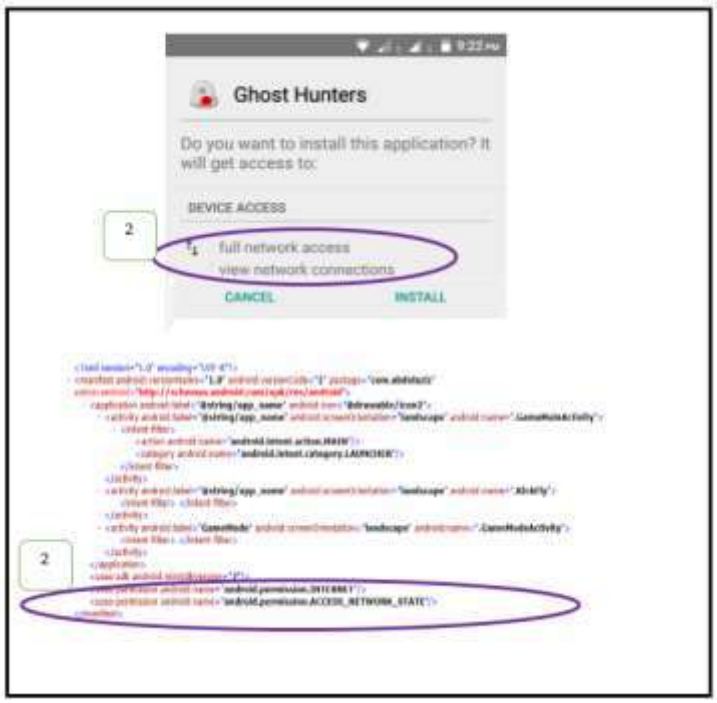

Figure 3. Comparison of Android manifest and permission Ghost Hunter

Table 3. Android.permission.WRITE_EXTERNAL_STORAGE of Ringtone Application

Permission	Class	Function Name	Actual Project code	Is permission declared in the manifest used in the actual source code? Yes/No
<uses permission android name:"Android.permission.WRITE_EXTERNAL_STORAGE"/>	ListRingtonesAdapter/ Util	setDefaultRingtone setDefaultAlarm setDefaultNotice setDefaultRingtone / assignRingtoneToContact	if(Environment.getExternalStorageState().equals(android.os.Environment.MEDIA_MOUNTED)){dir=new File(Environment.getEXTERNALSTORAGEDIrectory(), what);	yes

Table 3 shows the list of permissions of the Ringtone Application, it was also shown the class that was used in this application, Function name, Actual Project code, and the question if the permissions declared on the manifest was used in the actual project code. The permission WRITE_EXTERNAL_STORAGE was used in the class ListRingtonesAdapter and Util. In the class ListRingtoneAdapter you can observe the function name that was used in this class, the setDefaultRingtone, setDefaultAlarm and setDefaultNotice. While in the class Util you will observe the function name assignRingtoneToContact. You can also observe the code for this permission WRITE_EXTERNAL_STORAGE. This permission was used in these functions to check if the external memory was mounted into the mobile phone and also to get the exact path/directory of the external storage.

Table 4. Android.permission.WRITE_SETTINGS of Ringtone Application

Permission	Class	Function Name	Actual Project code	Is permission declared in the manifest used in the actual source code? Yes/No
<uses permission android name:"Android.permission.WRITE_SETTINGS"/>	ListRingtonesAdapter/ Util	setDefaultRingtone setDefaultAlarm setDefaultNotice setDefaultRingtone / assignRingtoneToContact	InputStreaminputStream =context.getResources(). openRawResource(info.getAudioResource());OutputStreamoutputStream=new FileOutputStream(file);byte[] buffer = new byte[1024];int length; while((length=inputStream.read(buffer)) > 0) {outputStream.write(buffer, 0, length);} outputStream.flush(); outputStream.close(); inputStream.close();	yes

Table 4 Shows the list of permissions of the Ringtone Application, it was also shown the class that was used in this application its Function name, Actual Project code and the question if the permissions declared on the manifest was used in the actual project code. The permission WRITE_SETTINGS was used in the class ListRingtonesAdapter and Util. Within this two classes you will observed some function name that was used, setDefaultRingtone, setDefaultAlarm, setDefaultNotice, DeleteRingtone and assignRingtoneToContact. For the permission WRITE_SETTINGS. WRITE_SETTINGS permission was used in this function to save/write the file into the mobile phone.

Table 5. Android permission.READ_CONTACT Ringtone Application

Permission	Class	Function Name	Actual Project code	Is permission declared in the manifest used in the actual source code? Yes/No
<uses permission android name:"Android.permission.READ_CONTACTS"/>	Util	getAllContacts	ArrayList<Contact>();Cursor cursor =context.getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, null, null, null, ContactsContract.Contacts.DISPLAY_NAME); if(cursor != null) { while (cursor.moveToNext()) { Stringname=cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME)); Stringid=cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts._ID)); name); contacts.add(contact); } } } cursor.close();	yes

Table 5. Shows the permission of the Ringtone Application, it was also shown the class that was used in this application, its Function name, Actual Project code, and the question if the permissions declared on the manifest was used in the actual project code. The permission READ_CONTACTS was used in the class Util. Wherein this class used the function name getAllContacts. READ_CONTACTS permission was used in this function to get all contacts saved in the phone.

Table 6.Android.permission.WRITE_CONTACT Ringtone Application

Permission	Class	Function Name	Actual Project code	Is permission declared in the manifest used in the actual source code? Yes/No
<uses permission android name:"Android.permission.WRITE_CONTACTS"/>	Util	getAllContacts	ArrayList<Contact>(); Cursorcursor=context.getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, null, null,null,ContactsContract.Contacts.DISPLAY_NAME);if(cursor != null){while (cursor.moveToNext()) {Stringname=cursor.getString(cursor.getColumnIndex (ContactsContract.Contacts.DISPLAY_NAME));Stringid=cursor.getString(cursor.getColumnIndex (ContactsContract.Contacts._ID)) ; name); contacts.add(contact); }}} cursor.close();	yes

The table above Shows that the Ringtone Application used permission WRITE_CONTACTS. The permission WRITE_CONTACTS was used in the class Util. WRITE_CONTACTS permission was used in this function to write the user's contact data in the phone.

Table 7: Android.permission.GET_ACCOUNT of Ringtone Application

Permission	Class	Function Name	Actual Project code	Is permission declared in the manifest used in the actual source code? Yes/No
<uses permission android name:"Android.permission.GET_ACCOUNTS"/>	Util	getAllContacts	ArrayList<Contact>();Cursor cursor=context.getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, null, null, null, ContactsContract.Contacts.DISPLAY_NAME);if(cursor != null) { while (cursor.moveToNext()) { Stringname=cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME)); Stringid=cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts._ID));name); contacts.add(contact); }}}cursor.close();	yes

The table above shows that Ringtone Application used permission GET_ACCOUNTS, this permission allows access to the list of accounts in the Accounts Service. This permission was used in the class named Util. Inside this class, a function name called getAllContacts this permission was used to get all list of accounts in the mobile phone.

Table 8.Android.permission.READ_PHONE_STATE of Ringtone Application

Permission	Class	Function Name	Actual Project code	Is permission declared in the manifest used in the actual source code? Yes/No
<uses permission android name:"Android.permission.READ_PHONE_STATE"/>	Util	getAllContacts	ArrayList<Contact>();Cursor cursor=context.getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, null, null, null, ContactsContract.Contacts.DISPLAY_NAME);if(cursor != null) { while (cursor.moveToNext()) { Stringname=cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME)); Stringid=cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts._ID));name);contacts.add(contact);}}}cursor.close();	yes

The table above shows that the ringtone application has permission READ_PHONE_STATE this permission allows the application to read-only access to phone state, including the phone number of the device, current cellular network information, the status of any ongoing calls, and list of any Phone Accounts registered on the device. The permission READ_PHONE_STATE was used in the class Util. READ_CONTACTS permission was used in this function to get all the contacts saved in the phone.

V. SUMMARY, CONCLUSIONS, AND RECOMMENDATION

Conclusion

This study test android security. The study examines the source code of the applications Kids Memory Game, Ghost Hunters, and Ringtone Application. The analysis showed that all of the three android applications have the same permissions, from the permission asked before installation and permissions declared on the manifest. In addition, it also showed that out of the three android applications, only the Ringtone Application used those permissions in the actual part of the source code. The Kids Memory Game and Ghost Hunters did not use the permissions that were required before installation. Consequently, the application Kids Memory Game and Ghost Hunter was safe to the users. However, the Ringtone Application was kind of suspicious and possibly dangerous, because this application can get and save all contacts of the phone. This study will help users be aware of the dangers of installing applications without reading or understanding the permissions required before the installation process. In this study, it was shown that the Ringtone Application can access some sensitive data such as the users' contacts. However, this application has not done any harm to the user.

REFERENCES

1. *Andrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. (2012), Android Permissions: User Attention, Comprehension, and Behavior. In Proceedings of the 8th Symposium on Usable Privacy and Security (SOUPS).*

2. *Neil Dupaul. (2012), :http://www.veracode.com/security/android-security*
3. *PatrickGage Kelley, Sunny Consolvo, LorrieFaithCranor, Jaeyeon Jung, Norman Sadeh, and David Wetherall. (2012), A Conundrum of Permissions: Installing Applications on an Android Smartphone. In Proceedings of the 16th Financial Cryptography and Data Security.*