International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

Android Security Manager

Kartik Konje[1], Shivam Kharde[2], Siddhi Kelaskar[3], Prof. Swapnil Sonawane[4]

[1,2,3]*Students of Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, Maharashtra*
[4]*Professor of Department of Computer Engineering, Vidyalankar Institute of Technology, Mumbai, Maharashtra*

Abstract - *This demonstration presents an application for system security evaluation. This enhances the overall security level of the Android-based device. The proposed application is based on the android application that uses standard Android Studio as IDE. We followed up a modular approach that allows using our library independently from our application in any Android version. The application evaluates the overall security level, gives an explanation of each parameter, gives advice on how to improve the overall security of the device, and allows to simulate the evaluation that shows how each parameter affects the safety level. which will allow users to know how many apps there are on respective mobile devices & understand what permissions have been taken and how much RAM they are consuming. Also, prevent third-party applications from using hidden or unknown permissions & giving the user option to uninstall the application respectively and maintain security. Users can also change the apps permissions that are unnecessary or uninformed to the user by application and take actions accordingly.*

***Key Words*: Android Security, Android Studio IDE, Permissions, etc**

1. INTRODUCTION

The smart mobile device now plays an important role in people's lives. In recent years, due to increasing popularity of mobile devices and the mobile Internet services, as well as device functions becoming more diverse and powerful, ensuring device security is gradually being taken seriously. At present, the period when mobile devices are popular and mobile Internet prevalent, personal information are mostly stored in a person's device. If protection is not done well, private information could be leaked or stolen. Vulnerabilities of mobile devices are mostly caused by improper configuration setting. Therefore, remediation of those settings can reduce vulnerabilities and thus improve system security. As one of the foremost popular mobile platforms, the Android system implements an install-time permission mechanism to supply users with a chance to deny potential risky permissions requested by an application.

Android is nowadays the world's most widely used smartphone platform. Security management in Android platforms and applications is a remarkable challenge especially due to its popularity, the openness of the system and the difficulties in version control procedures. Attacks of various types make it possible to compromise an Android device and potentially other information systems to which it has connections. Android strives to form it clear to users once they are interacting with third-party applications and inform the user of the capabilities those applications have. Prior to installation of any application, the user is shown a transparent message about the various permissions the appliance is requesting. After install, the user isn't prompted again to verify any permissions.

Every time you put in an application onto your phone, you're asked to permit that app certain permissions: to use your camera, track your location, view your contacts, and more. While some of these permissions are necessary for the app to function, some apps take advantage of that process to gather and exploit information they may not actually need [2]. Depending on the app, some information is logical to offer. However, some permissions aren't so obvious, and people are those to be weary of. Sometimes, apps will explain why they invite particular permissions within the developer notes. Others are less upfront. For example, the newest update of Snapchat requires users to turn on Location Services to use filters: Is sharing your location worth turning your photo black and white? against apps that ask to send text messages in your name. Those apps usually aim to ask friends to their service using your name and number. Some apps are constantly connected to the web, and may upload your personal data like your private photos or documents to a foreign server without your knowledge or consent. 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. An app can request permission to use my network connection, for instance, but I'm never sure what it's actually using that connection for.

Some security apps, like Lookout Mobile Security, feature "privacy advisors" which will offer you a touch more detail on why an app would request certain permissions. At best, however, that's a workaround for a bigger problem. Even with the extra information from security apps, you never see explicit details as to why, say, a browser app wants access to your phone's SMS function. This problem will be sorted by proper configuring your android devices and maintaining a track about which all applications are installed, how much RAM they consume, how many and how much necessary permissions are required and providing functionalities to overcome these issues. Thus maintaining security on your device.

2. UNDERSTAND THIRD-PARTY APPLICATIONS

Android strives to make it clear to users when they are interacting with third-party applications and inform the user of the capabilities those applications have. Prior to installation of any application, the user is shown a clear

message about the different permissions the application is requesting [3]. After install, the user is not prompted again to confirm any permissions.

There are many reasons to means permissions immediately before installation time. this is often when user is actively reviewing information about the appliance, developer, and functionality to figure out whether it matches their needs and expectations. it's also important that they have not yet established a mental or financial commitment to the app and should easily compare the appliance to other alternative applications.

Some other platforms use a special approach to user notification, requesting permission at the start of each session or while applications are in use. The vision of Android is to possess users switching seamlessly between applications at will. Providing confirmations whenever would hamper the user and stop Android from delivering a superb user experience. Having the user review permissions at install time gives the user the selection to not install the appliance if they feel uncomfortable. Also, many interface studies have shown that over-prompting the user causes the user to start out out saying "OK" to any dialog that's shown. one among Android's security goals is to effectively convey important security information to the user, which cannot be done using dialogs that the user are getting to be trained to ignore. By presenting the important information once, and only it is vital, the user is more likely to believe what they're agreeing to.

Some platforms prefer to not show any information within the least about application functionality. That approach prevents users from easily understanding and discussing application capabilities. While it's impossible for all users to always make fully informed decisions, the Android permissions model makes information about applications easily accessible to a good range of users. for instance, unexpected permissions requests can prompt more sophisticated users to ask critical questions on application functionality and share their concerns in places like Google Play where they're visible to all or any users.

Android devices frequently provide sensitive data input devices that allow applications to interact with the encompassing environment, like camera, microphone or GPS [1]. For a third-party application to access these devices, it must first be explicitly provided access by the utilization r through the use of Android OS Permissions. Upon installation, the installer will prompt the user requesting permission to the sensor by name.

If an application wants to know the user's location, the appliance requires a permission to access the user's location. Upon installation, the installer will prompt the user asking if the appliance can access the user's location. At any time, if the user doesn't want any application to access their location, then the user can run the "Settings" application, attend "Location & Security", and uncheck the "Use wireless networks" and "Enable GPS satellites". this may disable location based services for all applications on the user's device. For instance refer Fig -1.

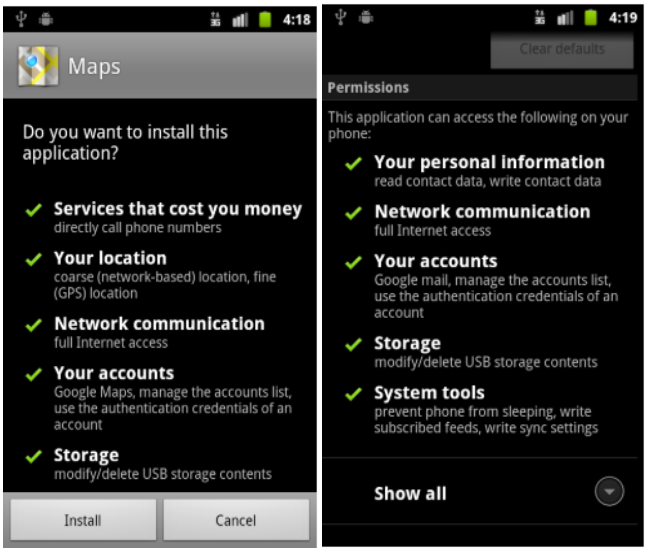

Fig -1: Display of permissions for applications

Android application have pretented almost every aspects of our lives, Android system adopts permission mechanism to limit Apps from accessing important resources of a smartphone, like telephony, camera and GPS location, users face still significant risk of privacy leakage because of the overprivileged permission. The overprivileged permission means the additional permisssion declared by the app but has nothing to try to to with its function some app take extra space that user not conscious of it [4]. Most of the days, the applications downloaded from unofficial sources pose a threat as there doesn't exist the required checks or mechanism to validate the authencity of those applications and perhaps infected with malware [4]. The malware infected applications can cause leakage of user's personal data. To resume this problem some applications are there but that apps only shows some what permissions it's taken and not overprivileged.

3. UNDERSTANDING PERMISSIONS

Firstly, we studied about the malware and therefore the refore the permissions and the different categories of permissions that an application seeks during its functioning [5]. These permission categories were available on google developer website [6] and provides clear understanding of how the working of the appliance is suffering from each permission the system requires both for software and hardware access. We studied about the difference between the benign and malicious applications and therefore the permission categories which were listed unsafe to be accessed by the user. The Android OS uses the permission-based model to access various resources and information [7]. These permissions aren't requests; they are declarations. These permissions are declared in AndroidManifest.xml file. Once the permissions are granted, the permissions remain

static for Android versions but 6 [8]. Android permissions are divided into several protection levels:

1) **Normal Permissions** are those permissions which have little or no risk of user‟s privacy. These permissions don't require user‟s involvement; these are granted by the Android system directly. eg. Bluetooth, Internet, Set_Alarm, Wake_Lock etc.
2) **Signature Permissions** are permission granted by the system at the install time. These are only granted to an app when it's signed by an equivalent certificate because the app that defines the permission. eg. Bind_Device_Admin, Bind_Nfc_Service, Read_Voicemail etc.
3) **Special Permissions** - The permissions that doesn‟t comes under the traditional and dangerous are the special permissions System_Alert_Window and Write_Settings are particularly sensitive, if an application wants to access these it must declare it within the manifest and access those with the assistance of intents.
4) **Dangerous permissions** - These are the permissions which require access to user‟s private data. For granting these permissions a message is prompt on the screen asking about the user‟s permissions. eg. Read_Calander, Camera, Read_Contacts, Write_Contacts, Call_Phone etc.
5) **Over-claiming of application permissions -** The permissions which may not be required for the app, but the application request for the particular permission, this is called over claiming of permissions [7]. It is the declaration to use irrelevant permissions that aren't in the least required for the appliance. It is the main reason for data theft in android application. The information is collected and sent to the concerned people. The developer of the app makes money by selling this information. Several third parties buy this information for various reasons like data mining etc., For example, in FlashLight Android app permission is given for full internet access. It is irrelevant for flashlight application to possess internet access.
6) **Double Authentication Failure** - App permissions are misused and due to that security breach is a possibility. Apps can read and send messages. Information is being transferred as a text message using Two-factor Authentication. For example, banks, online websites, resultantly failing this method.

4. PROPOSED SYSTEM

Proposed software would automatically analyze the permission use in Android applications also will perform accurate and in-depth analysis of the permission use in Android applications.

The main objective of this software is to reduce the vulnerabilities generated in the system due to incorrect user configuration settings and to uphold information security in Android mobile devices. Moreover, we believe proposed can help not only users to make informed decision about granting requested permissions, but also applications vendors to vet the permission requests of a large number of applications. Then, it would be easier for users to find the causes of these weaknesses and vulnerabilities on the device and their solutions, thus making the work of maintaining device security easier.

5. METHODOLOGY

The proposed application is based on the android application that uses standard Android Studio as a IDE. We followed up modular approach that permits to use our library independently from our application in any Android version. This presents an application for a system security evaluation. This enhances overall security level of the Android based device. Application evaluates overall security level, gives an evidence for each parameter, gives advices the way to improve overall security of the device, and allows to simulate the evaluation that shows how each parameter affects the security level. which will allow user to see how many apps are their in the phone & understand what permissions they have taken & how much RAM are they using. Also prevent the app from using certain permissions & giving the user option to uninstall the application respectively. Users can also change the apps permissions that are unnecessarily taken by other apps and also can uninstall the app. Also, code maintenance is simpler in contrast to Waterfall Approach.

In Waterfall methodology, there's no flexibility in changing the wants once we develop the project because we must understand the working flow of the project. Only if the planning process is completed, we will move to construction, testing, and support. For android security manager pre-requisite will be it will ask for certain permissions.

- Reading unknown applications permission
- Permissions to uninstall a particular application
- Permissions to ascess file manager
- Permissions to send data over TCP/UDP protocol
- Permissions to access notification logs

After above permissions are set it provides further operations, android security manager helps the user to:

1. Get all the information of the installed applications on the devices (like Desktop applications, etc) and contains metadata about applications packages, icon, space required, etc.

2. Allows to store the notifications triggered by the applications on devices from your PC’s or Laptops.

3. Discover and suggests similar apps based on the popularity and size which are safe and consume less space in the memory.

4. Display entire application information on the device and provides option to uninstall as well.

6. EASE OF USE

The test of the proposed system within the light of its workability, meeting user's requirements, effective use of resources and in fact , the value effectives. The prime focus of the feasibility is evaluating the practicality of the proposed system keeping in mind variety of things.

6.1 Technical Feasibility

Technical Technical feasibility assesses the present resources (such as hardware and software) and technology, which are required to accomplish user requirements within the software within the allocated time and budget. The technical issue usually raised during the feasibility stage of the investigation it includes the following:

- The technology used is integrated development environment for Google's Android operating system and will work on every android devices.
- The proposed equipment have the technical capacity to hold the data required to use the new system.
- The system can be upgraded if new version is developed.
- Android security manager is feasible and provides accuracy and data security.

6.2 Operational Feasibility

Operational feasibility aspects of the project are to be taken as a crucial a part of the project implementation. Some of the important issues raised are to check the operational feasibility of a project includes the following:

- The problem can be solved in the user's environment with existing and proposed system working.
- For user operational purposes sufficient permissions will be required to get access for each applications
- Being and android application its can be feasibile with all android devices.

6.3 Organizational Feasibility

Each of these technologies used are open source and the technical skills are easily manageable for organizational use.

7. CONCLUSIONS

We have proposed a software to automatically analyze the permission use in Android applications. which can perform accurate and in-depth analysis of the permission use in Android applications. the target of this software is to scale back the vulnerabilities generated within the system thanks to incorrect user configuration settings and to uphold information security in Android mobile devices. Moreover, we believe proposed can help not only users to form informed decision about granting requested permissions, but also applications vendors to vet the permission requests of an outsized number of applications. Then, it might be easier for users to seek out the causes of those weaknesses and vulnerabilities on their device and their solutions, thus making the work of maintaining device security easier.

ACKNOWLEDGEMENT

I would like to take the opportunity to thank and express my deep sense of gratitude to my Guide Prof. Swapnil Sonawane. I am greatly indebted for providing their valuable guidance at all stages of the study, their advice, constructive suggestions, positive feedback and encouragement and also would like to give whole hearted thanks and appreciation to the entire staff of the Vidyalankar Institute of Technology for their cooperation and assistance during the course of my project.

REFERENCES

[1] IEEE Reference Paper on Android system security evaluation - https://ieeexplore.ieee.org/document/8319325

[2] Android's Permission Problems - https://www.pcworld.com/article/251824/androids_permission_problems.html

[3] Understanding Third-party Libraries in Mobile App Analysis - https://ieeexplore.ieee.org/document/7965410

[4] IEEE Reference Paper on Overprivileged Permission Detection for AndroidApplications - https://ieeexplore.ieee.org/abstract/document/8761572

[5] IEEE Reference Paper on Permission based Android Malicious Application Detection using Machine Learning - https://ieeexplore.ieee.org/document/8938236

[6] Android developer permissions overview, [online] Available - https://developer.android.com/guide/topics/permissions/overview

[7] Android Security Issues and Solutions - https://ieeexplore.ieee.org/document/7975551

[8] Android (Nougats) security issues and solutions - https://ieeexplore.ieee.org/document/8394488