An app to demonstrate the risks of granting permissions in Android OS

Jakub Michálek[1], and Václav Oujezský[2]

[1]Brno University of Technology, The Czech Republic

[2]Brno University of Technology, The Czech Republic

E-mail: 186140@vut.cz, oujezsky@vut.cz

Abstract—This article presents an application for demonstrating the risks of granting unsafe authorizations on Android devices. These permissions allow the application to access sensitive user data. The application uses practical examples to illustrate what user data can be abused by enabling permission. Nowadays, the security of user data is a growing topic. Therefore, mobile users themselves must be aware of the risks they expose themselves to by granting permissions.

Keywords—Android, Kotlin, mobile application, permissions

1. INTRODUCTION

Almost everyone has a smartphone these days. Over 70 % of them are based on the Android operating system. As these devices provide users with many different functions, they are increasingly used for security-critical purposes such as internet banking, payment confirmation, or logging into online accounts. They also constantly collect sensitive user data such as text messages, location information, call history, photos, and videos.

Permissions allow support user privacy by preventing applications from accessing confidential user data and confidential actions. Each android application runs in its own isolated space and by default has no permission to perform any operations that could potentially adversely affect other applications, the operating system, or the user. Therefore, if it needs to use resources or information outside of this isolated space, it needs to declare permissions to access those resources and set up requests for those permissions. The three most important permissions groups are defined, namely permissions granted at installation, application runtime, and special permissions. Each of these groups has a limited range of data that can be accessed and actions that the application can perform once granted [1].

Permissions granted while the application is running are also known to be dangerous. Permissions give access to sensitive user data to an application and allow it to perform actions that significantly affect the system and other applications. In Android 6.0 and above, permissions are required when the application runs. In earlier versions, the operating system required all permissions to be enabled when an application is installed on a device.

This article describes a developed application whose purpose is to demonstrate to users one of the security risks of Android, namely the granting of insecure permissions. Enabling these permissions on a device allows sensitive data to be extracted from the user's device. The risks of each permission are illustrated with practical examples directly in the application.

2. THE APPLICATION OF THE RISKS OF SECURITY AUTHORISATIONS

2.1. Proposal

The application's primary purpose is to present the risks of granting dangerous permissions to users. It consists of two parts – mobile and server. The mobile part is programmed using Kotlin language, the one of the official programming language used today to create Android applications. The server part uses the PHP (Hypertext Preprocessor) language to communicate with the MySQL database and the HTML (Hypertext Markup) language to display data from the database and interact with the user.

2.2. Implementation

A user is presented with a menu that allows them to navigate to examples with risk permissions, settings and information about the application, shown in Figure 1. In the settings, it is possible to change the

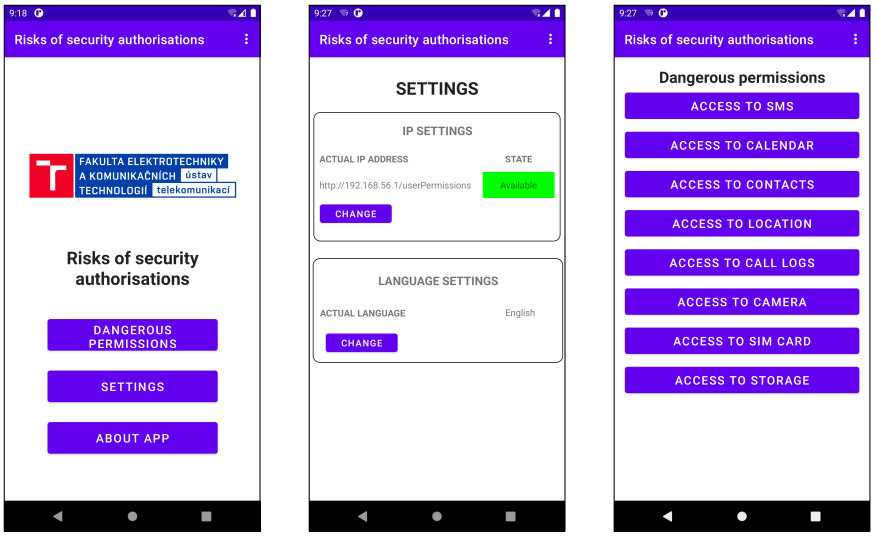

Fig. 1: The main menu screen, settings and list with the examples of dangerous permissions.

application language and the address of the server with which the application communicates. The server availability is required to use all functions. If the user navigates to the risk permissions examples, a menu with eight different permissions is displayed, which are listed with their function in Table I.

After selecting one of the examples, the user is first introduced to what is involved in enabling this permission and what is required to view the practical example. The user's account is also created in the database on the server. If a practical example is selected, the server is available, and the permission has already been enabled, sensitive data and photos are uploaded. If the server is unavailable, a dialog with the option to display an offline example is displayed, as is shown in Figure 2. If this option is selected, the offline example is displayed with the data intended to be sent to the server. If the permission has not been granted, a dialog is displayed depending on the user's previous decisions. If required for the first application run, the dialog contains an option to allow or deny permissions. Suppose the dialog has already been rejected and the user still wants to go to a practical example. In that case, the permission must be approved directly in the phone settings in the application manager.

The practical example then shows what sensitive data can be obtained with the selected permission. The web application loaded in the mobile is used for this purpose. The user's login credentials are generated and after logging in, it is shown that the sensitive data is outside the device itself and could be used by a potential attacker to his advantage. When the application exits, all sensitive user data stored on the server are deleted. The server hosts a web application and a MySQL database. The web application is used to display data from the database that has been stolen from a device using the mobile application to demonstrate the dangers of granting permissions. It is a simple application that consists of a user

	Permissions used	Allows
1	READ_CALENDAR	Read calendar events
2	READ_CALL_LOG	Read call logs
3	CAMERA	Access camera
4	READ_CONTACTS	Get contacts
5	ACCESS_FINE_LOCATION	Access phone location
6	READ_PHONE_STATE	Get SIM card information
7	READ_SMS	Read SMS messages
8	READ_EXTERNAL_STORAGE	Access files in storage

Tab. I: The permissions used in the application.

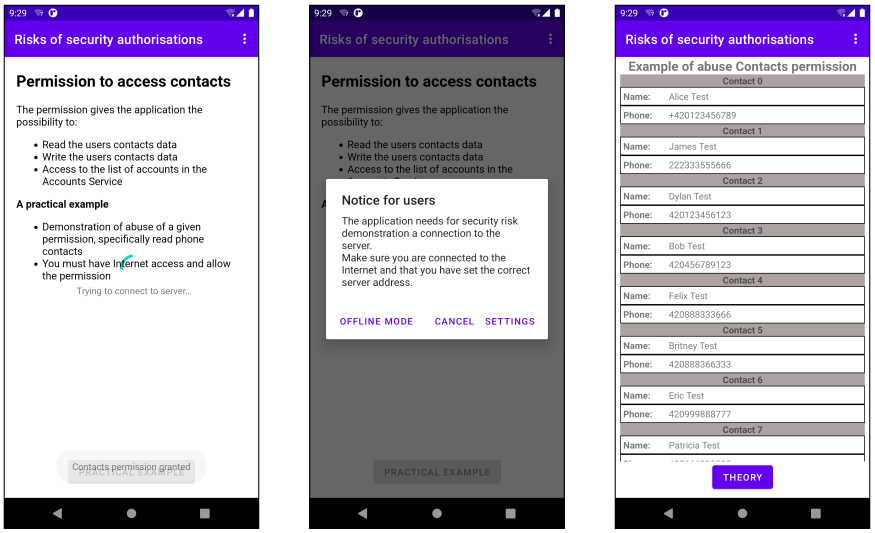

Fig. 2: A screen with the theory (left), unavailable server dialog (center) and an offline practical example of Contact access permission abuse.

login/logout page and a database data display page. The login page contains a form that requests the user's ID and password. This data is generated by the mobile application itself and stored in a database located on the server. If the correct login credentials are entered, the user is redirected to a page displaying the data from the database. If the user is not in the database an error message is displayed stating that the login details are incorrect. After a successful login, the retrieved data from the user's device stored in the database is displayed in the form of tables based on the login data. Figure 3 shows the example of SMS access permission abuse. If the stored data also includes photos from the user's devices, these are also displayed. The MySQL database is used to store the device data. It contains a static table for storing the users of the application. Tables storing data directly from devices using the mobile application are dynamically added depending on the currently displayed practical example.

Communication between the mobile application and the webserver is only one-way and takes place from the application to the webserver. Functions from the Volley library are used to send the data. This is the official HTTP (Hypertext Transfer Protocol) and HTTPS (Secure) library developed by Google for fast and easy communication over the Internet on Android [2]. In the presented version of the application, HTTP protocol communication is used. PHP scripts are created on the webserver to receive data from the application.

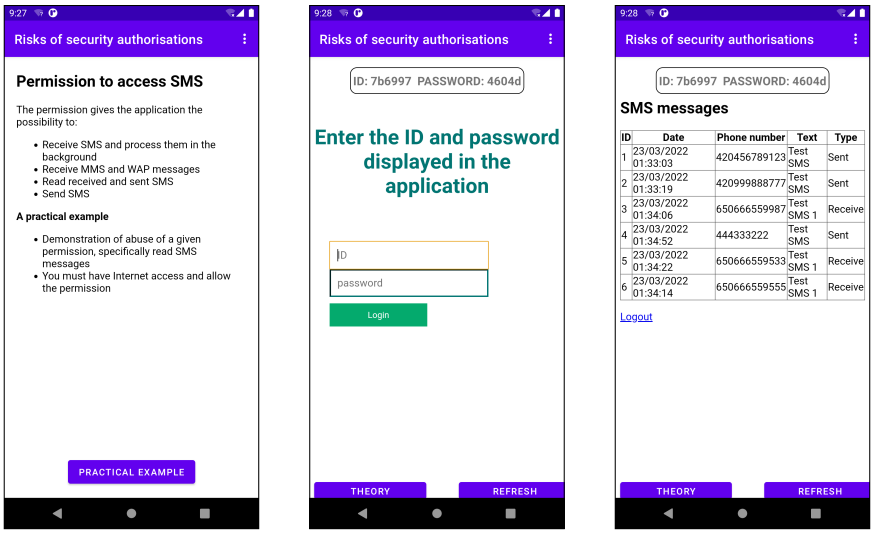

Fig. 3: A screen with the theory (left) and a practical example of SMS access permission abuse.

2.3. Verification

The application is now in the development stage and testing phase. The latest version of the application is available on the developer's website [3] and the source code on GitHub [4]. A video demonstrating all the functionalities of the application is uploaded on the YouTube platform [5].

The full functionality of the latest version of the application has been tested by using three tests. The first test involves testing the application in the mobile device emulator in the Android Studio IDE using a local server. This test is used to detect critical errors and complex functions. The second test is a user test with a server located on the hosting. This test involves testing the application on several real devices. It is used to detect errors that may not appear on the emulator. It also tests the functions for sending data from the device. The third test is performed in the Firebase CodeLab environment. This test is used to test whether the application is stable.

The application has passed all the tests mentioned above. All critical errors and shortcomings have been identified and corrected. As the testing phase is still ongoing, the application may contain some other bugs that have not yet been detected but should not affect the main functionality.

3. CONCLUSION

The aim of the paper was to present the developed application "Security of risk permissions". The purpose of this application is to present to users the risks that can occur when granting unsafe permissions. Enabling these permissions then gives the application the ability to access sensitive data stored on the mobile device and misuse it by an attacker without any knowledge of the application user. The risks posed by each permission are first described in a short theory and then demonstrated with a practical example. With the help of this application, the user can realize the risks that can arise by unthinkingly approving any permissions and thus protecting his sensitive data from applications that he does not trust completely. In the next version of the application, HTTPS protocol support will be implemented to communicate with the webserver and more examples will be added.

REFERENCES

[1] Developers, "Permissions on android," Google LLC, 2022. [Online]. Available: https://developer.android.com/guide/topics/permissions/overview

[2] V. Tyagi, "Volley library in android," GeeksforGeeks, 2022. [Online]. Available: https://www.geeksforgeeks.org/volley-library-in-android/

[3] J. Michalek, "Security application," 2022. [Online]. Available: http://unsecureapp.tode.cz/

[4] vaklur, "Userpermissions," GitHub, Inc., 2022. [Online]. Available: https://github.com/vaklur/UserPermissions

[5] J. Michalek, "Application – risks of security authorisations," Youtube, 2022. [Online]. Available: https://www.youtube.com/watch?v=MntV5ga39Ls