An Investigation into Permissions Requested by Mobile Banking on Android Platform

Latifa Er-Rajy, Cadi Ayyad University, Marrakesh, Morocco
M. Ahmed El Kiram, Cadi Ayyad University, Marrakesh, Morocco

ABSTRACT

In the past, banking took place only inside bank rooms, which was a task for customers and bankers at the same time. But in our day, thanks to the high-speed development and growth of mobile technology, the mobile phone platform had the power to create great opportunities for customers of the physical bank due to its capabilities and coverage of the population; this can be proved by the number of mobile subscriptions that approximates the world population figures. In order to explore these opportunities, most banks have already launched their mobile apps or have redesigned the mobile version of their websites. Among the advantages of using mobile banking is that users have the ability to make banking transactions, online payments or transfers, anywhere and at any time. In this article, we investigated the danger of the permissions requested by mobile banking applications, their effects on sensitive user data and their relationship with the attack called “Man in the middle” and its different forms. We took Morocco as a case of study.

KEYWORDS:
Android, Applications, M-Banking, MITMo, Permissions

INTRODUCTION

Mobile banking is simply the service that allow a mobile customer to use freely his bank account for different services (Poustitchi & Schurig, 2004). The credit for success of mobile banking is due to its convenience, ease of use, ubiquity and reliability. This leads to the presence of confidential information belong to Mobile Banking users. So, it is a requirement to secure the users data in order to prevent the hacker from attacking and stealing sensitive data. The consumers percentage who use mobile banking have increased over the past years and continues to grow in the world and even in developing countries like Morocco. In fact, most banks in Morocco have added the mobile channel as an additional information channel. However, the transactional services provided to clients have low-income. Even for AttijariWafa Bank and Albarid Bank that are the leaders of the mobile financial services development (Final Report Mobile financial services in Mediterranean Partner Countries, 2012). In the 2015 survey, the mobile banking continued to rise, reaching to 43% of mobile-phone users with bank accounts and 23% of smartphone users with bank accounts (Board Of Governors of the Federal Reserve System, 2012).

To alert users about the security and privacy ramifications of installing an application, whether banking or otherwise, Android uses Mandatory Access Control (MAC) (Paraboschi, Bacis, & Mutti, 2015), which means that at the time of installation, an application must request permission to

DOI: 10.4018/IJMCMC.2018040102
access to system resources such as the user’s location, Internet or cellular network. So, an interface appears to the user, which allows him either to accept all the requested permissions or to cancel the installation since it is not possible to selectively accept or refuse them. Thus, many users simply accept these authorization requests without considering their implications, which put their private data in the danger zone. Like Android, the iOS system uses a permissions-based model for each application. The deference is that in the iOS system, the user can download an application and decide which permissions the application can use. In Android, the user must agree to allow access to all permissions requested by the app before downloading and he cannot disable any authorization once the application is installed. The opposite is true in the iOS system (Au, Zhou, Huang, Gill, & Lie, 2011). The permissions required by an application can endanger the user when using this application. Mobile users are concerned but often do not understand the security risks that might be involved when making financial transactions via a mobile application because of permissions granted to it.

In this research, we focus on Android mobile banking application permissions and their relation with Man-In-The-Mobile attacks.

In summary, in this article, we make the following contributions:

1. We describe the permissions requested during installation by mobile banking application for the Android platform. And we show to what extent these permissions can be very dangerous for the data exchanged during the execution of banking transactions. And to better explain this danger, we give as an example the attack called “Man-in-the-Phone” that can take advantage of these permissions to perform malicious actions.
2. We show that sometimes the permissions requested by mobile banking applications have no relation to the features and services provided by them. And to prove this idea, we compare two applications for the two leaders of mobile banking in Morocco.
3. We present the scan result of a set of applications from around 100 apps collect from Google Play Store. For ease and speed of the analysis task, we used our tool called “PerUpSecure” that we developed in order to analyze the permissions requested by Android applications before authorizing their installations.

BACKGROUND

Android Architecture

The Android platform as shown in Figure 1 is a stack with different layers running on top of each other layers, lower layers provide services to higher layers. The Linux Kernel is at the bottom. It handles the hardware abstraction (Zhang et al., 2013). This is the layer where the entire hardware driver specific to the device will be convened, allowing hardware vendors to develop drivers in a familiar environment. It also imposes some of the most basic separation between applications. On top of the kernel we find the native libraries which are code modules that are compiled to native code machine for the device interest, and they offer a number of common services that are accessible by Android applications and other programs. They include the surface manager (Responsible for graphics on the device screen) (Elenkov, 2014), the 2D and 3D graphics libraries, Web Kit (the web rendering the engine that power the default browser) and SQLite (the basic database technology for the Android platform). These native libraries are executed by processes in the underlying Linux kernel and in the runtime implementation. Each application owns its instance of Android runtime environment for its execution, and each instance core is a Dalvik virtual machine (Nolan, 2015).

Deconstructing Application Installation

Any developer and even if he isn’t registered with Google has the power to develop and distribute his Android applications in the official Google market known as “Play Store” or through third-party
Depth-Vision Coordinated Robust Architecture for Obstacle Detection and Haptic Feedback
www.igi-global.com/article/depth-vision-coordinated-robust-architecture-for-obstacle-detection-and-haptic-feedback/142529?camid=4v1a

Meet your Users in Situ Data Collection from within Apps in Large-Scale Deployments
www.igi-global.com/article/meet-your-users-in-situ-data-collection-from-within-apps-in-large-scale-deployments/144334?camid=4v1a