The Journal of Systems and Software 000 (2017) 1–13

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

An empirical study of collaborative model and its security risk in Android

Ajay Kumar Jha, Woo Jin Lee*

School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea

ARTICLE INFO

Article history:
Received 22 March 2016
Revised 8 July 2017
Accepted 27 July 2017
Available online xxx

Keywords:
Android applications
Inter-application communications
Collaborative application model
Permission-based security
Security risk assessment

ABSTRACT

Android provides a framework for the development of collaborative applications, which is considered as one of the reasons behind its success. Collaborative model provides flexibility to an application in utilizing services offered by other applications. This approach offers several advantages to developers, such as allowing them to dedicate all of their resources in developing only core functionalities of an application while leveraging services offered by other applications for its auxiliary functionalities. However, the collaborative model also has some disadvantages, such as opening of attack surfaces in an application during exposure of some of its components as it offers its services. Malicious actions can be performed through the exposed components of the application. Android provides permission-based security to protect the exposed components. However, developers must implement the security correctly. In this paper, we empirically evaluate the scale of the collaborative model adopted by Android applications. We also investigate various methods to achieve collaboration among applications. Furthermore, we evaluate the scale of security risk instigated by the collaborative model and perform several other empirical studies on 13,944 Android applications.

1. Introduction

Android operating system, which currently holds the largest market share (Smartphone, 2017), is one of the major players in the smartphone industry. Although open source is considered as the main reason behind its remarkable success, it can also be attributed to the incentives available for each stakeholder including developers. Developers can develop Android applications on multiple platforms. Most importantly, they can utilize the collaborative framework during application development. Android application developers can also utilize large number of APIs provided by the platform. These incentives not only facilitate but also encourage an individual developer to develop and publish applications in the market, which is evident as Google Play Store has overtaken other application stores based on the number of available applications, although it generates less revenue for developers than its main rival, the iOS App Store (Revenue for developers in Google play store vs iOS app store 2017). The framework for developing collaborative applications is a unique feature of Android and it is considered as one of the key elements behind Android success.

Collaborative model is not a new approach. Android only brought this approach into application development practice, which cannot be considered as a trivial step, but rather as a paradigm shift in the way applications collaborate. Some forms of collaboration exist among traditional software. For example, plug-ins are developed to collaborate with the target software. Android brought this traditional approach to a new height. Unlike traditional software where plug-ins act only as supplementary, an Android application that offers services to other applications can also perform its own tasks. For example, a camera application performs its own tasks as well as offers services to other applications. Other applications can easily utilize the services offered by the camera application. This seamless collaboration between applications is a unique feature of Android. The collaborative approach provides several advantages to developers, such as allowing them to focus on an application's core functionalities while leveraging services offered by other applications for its accessory functionalities, which increases the quality of the application and decreases the development time, resulting in less development cost and competitive advantage in the market.

Along with several advantages, the collaborative approach has some disadvantages. Android applications are composed of components. An application offers its services by exposing one or more of its components, which opens attack surfaces (Enck et al., 2009; Chin et al., 2011). These attack surfaces can be easily exploited by malicious applications. Android provides protection against these

* Corresponding author.
E-mail addresses: ajaykjha123@yahoo.com (A.K. Jha), woojin@knu.ac.kr (W.J. Lee).

http://dx.doi.org/10.1016/j.jss.2017.07.042
0164-1212/

Please cite this article as: A.K. Jha, W.J. Lee, An empirical study of collaborative model and its security risk in Android, The Journal of Systems and Software (2017), http://dx.doi.org/10.1016/j.jss.2017.07.042

attack surfaces at various places. Applications are vetted for malicious behavior when they are installed into Google Play Store (Bouncer - 2017). The procedure prohibits malicious applications from entering the store and collaborating with benign applications. Given that the vetting procedure is not 100% effective (Oberheide and Miller, 2012), some malicious applications may enter into the store. The presence of malicious applications in the store has been reported occasionally (Chen et al., 2015; BrainTest – 2017). Third party stores are other but major source of malicious applications (Zhou et al., 2012) against which Google provides a verification process (Verify apps - 2017). However, the process is optional for users. Android also provides permission-based security at the application level to protect the attack surfaces. However, developers must implement the security correctly. Several studies have indicated that developers fail to implement the collaborative model and its security correctly. A common developer's mistake in Android is to expose a component unintentionally (Kantola et al., 2012), which leaves it unprotected. Developers may fail to implement protection mechanisms correctly because most of the developers are not security experts (Davi et al., 2011; Felt and Helen, 2011).

In this paper, we empirically evaluate the scale of the collaborative model adopted by Android applications. The evaluation results can shed light on some important points. If the collaborative model has been adopted in large scale, then it can be concluded that the collaborative model has succeeded and it is contributing to the success of Android. The success of the collaborative model in Android may also open doors for collaborative applications or software on other platforms. Meanwhile, small-scale adoption indicates that the collaborative model has failed and it does not contribute to the success of Android. We also empirically evaluate the security risk instigated by the collaborative model. If the collaborative model has been adopted in large scale and the empirical study indicates that most of the exposed components are protected, then it can be concluded that the collaborative model does not pose significant security risk. Meanwhile, a large-scale collaborative model with a large number of unprotected exposed components indicates that either the developers are unaware of the security risk associated with the exposed components or they are implementing the security incorrectly. In either case, the collaborative model poses high security risk, which may affect its further development in Android or other platforms. Furthermore, we investigate the common security, reliability, and availability issues associated with the collaborative model and perform empirical studies on 13,944 free Android applications downloaded from the Google Play Store.

The main objective of this study is to investigate the collaboration among Android applications and the security risk instigated by the collaborative approach. In this direction, the contributions of this paper can be summarized as follows:

- Performs empirical studies on 13,944 popular Android applications.
- Assesses the scale of collaboration among Android applications and reports on different collaboration approaches involving the participation of different types of components.
- Evaluates the security risk in the collaborative approach by analyzing unprotected components and sensitive resources used by the applications.
- Identifies incorrect implementations of the collaborative model and its security and discusses their implications.

The rest of the paper is organized as follows. Section 2 describes the background on the collaborative model in Android and its security. Related works are discussed in Section 3. Several empirical studies converging toward the collaborative model and its security risk are performed in Section 4. We also discussed the incorrect implementations of the collaborative model and its security along with their implications in Section 4. The results obtained from the empirical studies are discussed in Section 5. Threats to the correctness of the obtained empirical results are discussed in Section 6. Finally, the paper concludes in Section 7.

2. Collaborative model and its security

Android applications comprise four kinds of components: activity, service, broadcast receiver, and content provider. In an application, a component can complete a task independently. However, components may interact with each other to complete the task. Furthermore, the components of an application can also communicate with the components of other applications. For example, a user may need to attach an image in an email. In this case, an email application may communicate with a camera application to obtain the image. This kind of collaboration among applications is a unique feature of Android, which can be described in terms of a client-server architecture. In a collaboration event, an application that offers services acts as a server whereas an application that avails the services acts as a client. Unlike the client-server architecture, Android applications are neither clients nor servers exclusively. In the example, the email application acts as a client whereas the camera application acts as a server for the event. However, the condition may reverse in another event. For example, a user captures an image through a camera application, and then sends the image through an email application. In this collaboration, the camera application acts as a client whereas the email application acts as a server.

The collaboration in Android is achieved through intents and intent filters (Intents and Intent Filters 2017). An intent describes an operation to be performed by a component. Depending on the presence or absence of a target component, an intent is categorized as either explicit or implicit. An explicit intent is directly addressed by a target component whereas a target component for an implicit intent is resolved by the system. Intent filters play a crucial role in the target resolution process. A component advertises its capability to handle a specific operation through an intent filter, which is registered in the system. If an implicit intent requests for a target component, then the system matches the operation to be performed by the intent with the operation advertised by the components. The component with the matching operation is served as a target component for the intent. Although collaboration can be achieved through an explicit intent, it requires prior knowledge of the target application, which results in some restrictions on the collaboration. Meanwhile, the collaboration achieved through an implicit intent does not impose these restrictions. In this form of collaboration, the collaborating applications do not have knowledge of each other. An application advertises its services through intent filters, and any other application can avail those services if they satisfy the security condition imposed by the service provider application.

The collaborative model and its security are implemented in Android through a manifest file, which is a configuration file where all components of an application must be declared. However, the broadcast receiver components can also be declared in the source code. An application willing to offer services to other applications must be declared through the manifest file. Given that services are offered through one or more components of an application, the application must expose its components. A component of an application makes itself available to other applications by setting its exported flag attribute to true. In the case of activity, service, and broadcast receiver components, the default value of the exported flag depends on an intent filter. The presence of one or more intent filters in a component indicates that the component is exported. In the case of a content provider component, if an application uses an

Please cite this article as: A.K. Jha, W.J. Lee, An empirical study of collaborative model and its security risk in Android, The Journal of Systems and Software (2017), http://dx.doi.org/10.1016/j.jss.2017.07.042