Demo: RecDroid— An Android Resource Access Permission Recommendation System

Bahman Rashidi, Carol Fung, Gerrit Bond, Steven Jackson, Marcus Pare
Dept. of Computer Science, Virginia Commonwealth University
Richmond, Virginia, USA
{rashidib,cfung,bondg,jacksonst4,pareme}@vcu.edu

Tam Vu
Dept. of Computer Science and Engineering,
University of Colorado Denver
Denver, Colorado, USA
tam.vu@ucdenver.edu

ABSTRACT

Nowadays, it is prohibitive for apps market places, such as Google App Store, to thoroughly verify an app's resource permission requests to be legitimate or malicious. As a result, mobile users are left to decide for themselves whether an app is safe to use or not. To assist users to make correct decisions as for whether to accept a permission request or not, we propose RecDroid. RecDroid is a crowdsourcing recommendation framework that collects apps' permission requests and users' responses to those requests, from which an experts ranking algorithm is used to seek expert users in the system and a recommendation algorithm is used to suggest appropriate responses to permission requests (accept or reject) based on experts' responses. In this demo, we demonstrate a user case to show how the RecDroid system assists users in permission control. We also explain the major principles and processes behind that support the RecDroid recommendation system.

Keywords

Permission; Android; Recommendation; Policy-based; Smartphone

1. INTRODUCTION

In current Android architecture, users are the key to resource control to applications. Users need to grant all resource access requests before installing and using an app. This type of permission control mechanism, however, has been proven ineffective to protect users privacy and resource from malicious apps. Study shows that more than 70% of smart phone apps request to collect data irrelevant to the main function of the app. When installing a new app, only a small portion (3%) of users pay attention and make correct answers to permission granting questions, since they tend to rush through prompted permission request screens to get to use the application. In addition, the current Android permission warnings do not help most users make correct security decisions.

Realizing these shortcomings in the current Android architecture, much efforts have been put towards to address the problems.

MobiHoc'15, June 22–25, 2015, Hangzhou, China.
ACM 978-1-4503-3494-5/15/05.
http://dx.doi.org/10.1145/2746285.2764930.

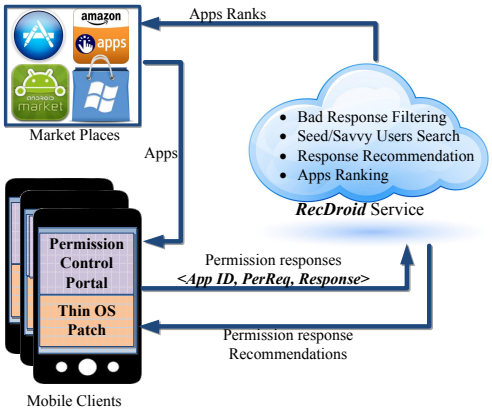


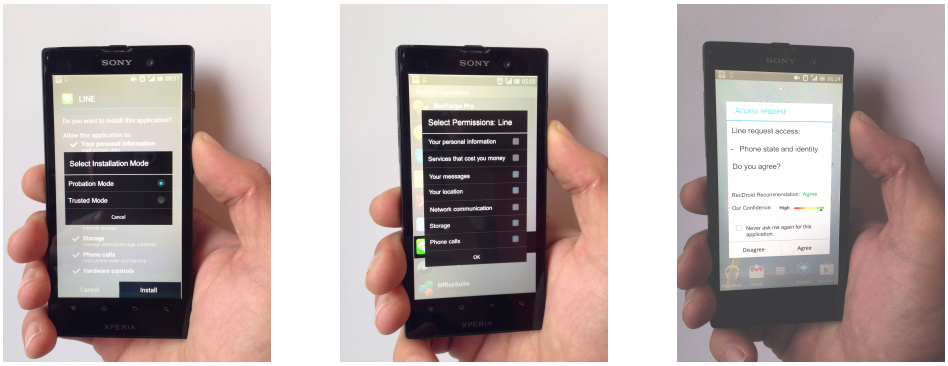
Figure 1: RecDroid Service Overview

In particular, we provide a framework, called *RecDroid*, to assist mobile users to control their resource and privacy. First, the framework allows users to use apps without giving all permissions and receive help from expert users when permission requests appear. RecDroid allows users to install untrusted apps under a *"probation"* mode, while the trusted ones are installed in normal "*trusted*" mode. In probation mode, users make real-time resource granting decisions when apps are running. Second, the framework also facilitates a user-help-user environment, where expert users' decisions are recommended to inexperienced users. The framework provides the following functionalities [1, 2]:

- Two app installation modes for apps: *trusted mode* and *probation mode*. In the probation mode, an app requests permissions from users at run time to access sensitive resources (e.g. GPS traces) when the resource is needed. In the trusted mode, the app is fully trusted and all permissions are granted.
- An architecture to intercept and collect apps' permission requests and responses, from which recommendations are made on whether a permission should be granted or not.
- A recommendation system to guide users with permission granting decisions, by serving users with recommendations from expert users on the same apps.
- A user-based ranking algorithm to rank security risks of mobile apps.

2. SYSTEM DESIGN AND IMPLEMENTAION

We designed and implemented RecDroid with four major functional components. In particular, RecDroid (1) captures resource



(a) (b) (c)

Figure 2: An example of RecDroid on Line app: (a) probation and trusted installation modes; (b) Users pick which critical resources to be monitored; (c) Pop-up for permission granting with suggestion from RecDroid and its confidence

access request on the phone and pops up dialogs to interact with users when such events happen, (2) recommends users with low-risk responses to permission requests based on expert users' responses, (3) collects users' permission-request responses and reports them to the server, and (4) allows users to configure monitored resources through a management portal. The architecture overview of RecDroid system is shown in Figure 1. The system consists of of a *thin OS patch* and a *RecDroid server*. The *thin OS patch* enables RecDroid to capture local permission requests and interact with users regarding permission control. It also communicates with the server to report user responses to those permission request and receive recommended responses from the server.

We implemented the request capturing function of RecDroid by modifing the permission management component of the Android OS. We set up a RecDroid server on a desktop to handle data storage and recommendation algorithm. We also created an Android app named *Permission Control Portal* for users to monitor and manage resource access permissions at fine-grain level.

3. DEMONSTRATION

3.1 User Interaction

The RecDroid supports users with fine-grained permission control and guided permission decision making with RecDroid recommendation system. We use the app "Line" as an example. The first screenshot (Figure 2(a)) shows two options on app installation, *probation mode* or *trusted mode*. If a user selects the probation mode, the application will be added to the list of RecDroid monitored apps. Otherwise, if the user selects the trusted mode, the application will be installed with all requested permissions granted.

For the installed apps, users can use the RecDroid's *Permission Control Portal* app to view a list of apps which have been installed under the probation mode. If a user selects an app in the list, a list of its requested permissions (Figure 2(b)) will be shown and the user can select some of them to be monitored resources. By default all sensitive resources are monitored.

If an app is installed under the probation mode, whenever the app attempts to access a monitored resource, RecDroid will detect it and the user will be notified through a pop-up dialog (Figure 2(c)). The RecDroid system provides a recommendation with a confidence level to assist users to make decisions regarding whether or not to grant the resource access. If the user select agree, the requested resource will be served; otherwise the access is blocked.

3.2 Experiment Scenario

In our demo, we will show the performance and usability of the RecDroid system through an interactive process. We will show that how the RecDroid system discovers expert users and makes low-risk recommendations to permission requests. In the demo, we will show the interception design of RecDroid and explain the background processes that support the functions of RecDroid. We will show the entire process of RecDroid from installing an app to permission decision making assisted by RecDroid recommendation.

To prepare for the demo, we build a customized Android ROM (Android 4.3, Jelly Bean) equipped with RecDroid system and install it on 5 LG Nexus 4 devices. We also install 20 apps including known Malicious and non Malicious apps. In addition to installed apps, we have developed some apps with different required permissions and users can install them by themselves under probation or trusted mode. After setting up the phones, users will work with the devices. Whenever an installed app requests to get access to a resource, users will be informed through a pop-up dialog to decide whether to accept or reject the request.

RecDroid records all responses from users and store them on an online server hosted by the Computing Center of Virginia Commonwealth University. We use the collected responses to identify the expertise levels of users through our expert seeking algorithm. After ranking the users, we aggregate the responses to a permission request from different users using a weighted voting technique to generate a recommendation on whether to grant the resource access request. We will demonstrate that the RecDroid recommending system can generate high quality recommendations to assist users to protect their security and privacy.

In this demonstration we are planning to show the following aspects of the RecDroid system:

Interaction. In this part of the demo, we show the usability of the system by demonstrating the entire process of the interaction between RecDroid and users. We also demonstrate how users can use RecDroid app to configure the system.

Recommendation System. In this part of the demo, we will show how RecDroid identifies expert users and how it utilizes expert users to make recommendations.

3.3 Equipmet and Settings

We will bring 1 laptops, 4 mobile phones (Nexus 4) and some brochure to demonstrate RecDroid. In addition, we ask for a demonstration table, a poster board, and an electronic screen (around 31.5inch and above) for the video demonstration. We also need the WiFi access for our Mobile phones to communicate with the server.

4. REFERENCES

[1] B. Rashidi, C. Fung, and T. Vu. Recdroid: A resource access permission control portal and recommendation service for smartphone users. In *Proceedings of the ACM MobiCom Workshop on Security and Privacy in Mobile Environments*, SPME '14, pages 13–18, New York, NY, USA, 2014. ACM.

[2] B. Rashidi, C. Fung, and T. Vu. Dude, ask the experts!: Android resource access permission recommendation with recdroid. In *14th IFIP/IEEE International Symposium on Integrated Network Management (IM 2015)*. IEEE, Ottawa, Canada, 2015.