H. Langweg, M. Meier, B.C. Witt, D. Reinhardt (Hrsg.): Sicherheit 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 279

Turning the Table Around: Monitoring App Behavior

Nurul Momen[1]

Abstract: Since Android apps receive whitecard access through permissions, users struggle to understand the actual magnitude of app access to their personal data. Due to unavailability of statistical or other tools that would provide an overview of data access or privilege use, users can hardly assess privacy risks or identify app misbehavior. This is a problem for data subjects. The presented PhD research project aims at creating a transparency-enhancing technology that helps users to assess the magnitude of data access of installed apps by monitoring the Android permission access control system. This article will present how apps exercise their permissions, based on a pilot study with an app monitoring tool. It then presents a prototypical implementation of a networked laboratory for crowdsourcing app behavior data. Finally, the article presents and discusses a model that will use the collected data to calculate and visualize risk signals based on individual risk preferences and measured app data access efforts.

Keywords: App Behavior, Privacy Preservation, Transparency.

1 Introduction

User data is an important key for today's customer driven economy. An elusive, complex and service-centric business model took over the usual consumer-product model. In this new model, most of the users remain in wonderland consuming 'free' services while turning themselves into a product. In order to support the revenue model, mobile apps are being used to profile users to a great extent. Service providers are supposed to utilize such user data to deliver more customer-centric packages, but the probable and rather obvious privacy risks are usually broomed under the carpet. Apps are demanding too many permissions [Mo17], leaving the actual access to data opaque to the users, thus creating a data protection problem. Our project focuses on designing and developing transparency-enhancing tools (TETs) [He08] to uncover some of the risks in order to help users by making more informed decisions to protect their privacy.

The app markets are thriving with popularity metrics and crowdsourced user opinions which are very unlikely to be based on security and privacy factors. The rapid growth of the app market has outpaced the development of adequate transparency and control over user information. In the KAUdroid project we make an effort to ease users' remorse by utilizing TETs. In this paper, we introduce an app-behavior-analysis-model which includes a prototype app, a client-server architecture to document app behavior, a database and a visualization tool.

doi:10.18420/sicherheit2018_25

2 Background

A significant amount of research effort has been invested on the access control model of Android. In this section, a brief overview of related efforts is illustrated along with our main research goals.

2.1 Related Work

Several empirical studies pointed out that users face difficulties to perceive appropriate consequences of granting permissions to apps; for instance [Ke12, Pe12, KCS13]. Absence of regulation enforcement and technical measures to ensure the principle of least privilege is also held responsible for leaving sensitive user information in a vulnerable state [Fe11, HBM17]. In fact, apps are asking for more information than ever before [Au12, We12]. In order to aid the user in preserving privacy, we focus on investigating *how often* permissions are being accessed by the installed apps.

2.2 Research Goals

Since the introduction of runtime permission architecture, user consent is required during first-time use of the corresponding permission[3]. Granted privileges remain unchanged unless the user explicitly revoke them. Additionally, no statistics or qualitative information is available which could support reassessment of initial decisions. We argue that it diminishes the effectiveness of runtime permissions to some extent. As an initial inquiry, this project has performed a pilot study [Mo17], which found that apps are accessing granted resources more frequently and the interface is unable to offer neither quantitative, nor qualititative usage information to the user. Later on, we demonstrated the risks concerning partial identity generation from questionable and inconsistent resource usage patterns by apps [FM17]. Now we aim to investigate in a larger scale. Hence, measuring infrastructure is built to generate larger dataset in order to commence controlled experiments. This project is intended to answer following research questions: (1) How can privilege-induced privacy risks be communicated effectively to the user? (2) How can tools and methods assist users to benefit from ensuring the principle of least privilege for apps?

3 Model

In this section, we describe the proposed model for data analysis, determination of app's privacy impact factor (*PIF*) and app's risk score (*RS*). The model will be tested within an experimental lab that is being built for quantitative data collection purpose. The model is elaborated as following:

$M = (\delta, \alpha, \rho, \tau)$where,
δ = a finite set of participating devices/nodes;
α = a finite set of apps installed on the device;
ρ = a finite set of permissions requested by apps;

[3] `https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html`

τ = logged instances (timestamp) for permission usage.

For a given device δ_1, app α_1 and a time frame $\Delta\tau = (\tau_t - \tau_0)$, the total usage of permission ρ_1, is calculated as: $R_{\rho_1} = COUNT(\tau_i)$. Later on model M is used to accumulate data based on m number of different scenarios, S = s_1, s_2, ... s_m , where the elements refer to boolean values.

s_1 - *active/passive usage:* s_1 depends on whether the user is interacting with the app or not. It could be unlikely for a user to keep using an app throughout the day and night, but the app possesses corresponding permission to accumulate data (e.g. ACCESS_FINE_LOCATION) without any time constraint. We observed idle-time usage of the location permissions in our primary experiment. Thus we plan to document resource access events based on user interaction. This could reveal the app behavior throughout the day, week and month.

s_2 - *surrounding environment:* s_2 depends on whether the device is located in a crowded place or not. We hypothesize that apps could be greedy to discover user's peers in a crowded area which has the potential to extract the social graph of users. In our earlier work [FM17], the likelihood of partial identity (e.g. social graphs) extraction was presented. In this controlled scenario, devices are intended to be taken to crowded places (e.g. a restaurant, library, concert, shopping mall, etc.) and to observe the app behavior with respect to isolated usage.

s_3 - *motion and network connectivity:* stationary usage and wifi-only-connectivity of the device were the limitations for the previous study. With a view to overcome these shortcomings, scenario s_3 includes device deployment through short and long journeys with connectivity through wifi and mobile telephony. Devices will be kept and used during a journey (by train or by car) and document app behavior comparing with their stationary permission usage.

Further scenarios (s_4, s_5 and so on) can be defined and determined from individual preferences. Based on permission usage and scenarios, a *PIF* is calculated, which opens the option for accommodating tolerance and individual prioritization factor (p_m):

$$PIF = p_1.s_1 + p_2.s_2 + ... + p_m.s_m \tag{1}$$

If there are m number of $PIFs$ associated with corresponding scenarios, for a given app α with permission ρ running on a device δ for a selected time frame $\Delta\tau = (\tau_t - \tau_0)$, a risk score ($RS_\alpha$) is defined as following:

$$RS_\alpha = \sum_{i=0}^{m} R_{\rho_i}.PIF_{\rho_i} \pm E = \sum_{i=0}^{m} R_{\rho_i} \left(\sum_{j=0}^{n} p_j s_j \right) \pm E \tag{2}$$

The error margin (E) is yet to be defined and tested through analysis. Currently, we undertake data collection to fill the database with sufficient test data for populating the model under the scenarios. The experimental setup for testing the model is elaborated in the next section.

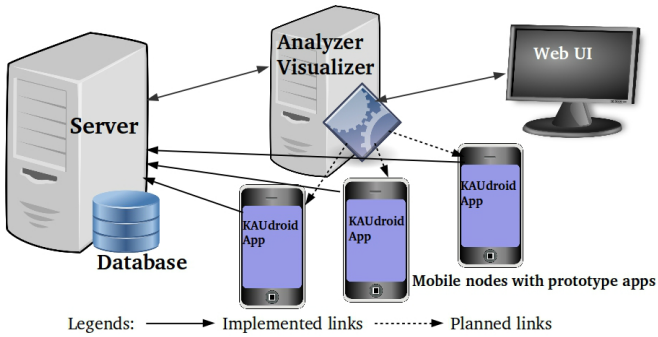

Fig. 1: Overall architecture.

4 System Architecture

In this section, we describe the system architecture of experimental setup that has been built to carry out controlled experiments [Ca18]. The system architecture is depicted in Fig. 1 and can be segmented into three different parts: a) logging, b) data collection, and c) analysis and visualization.

4.1 Logging

A prototype app has been developed that runs as a service on mobile devices. It is a monitoring app and is able to log each of the resource access events. The prototype is transparent to itself, and the log file also documents resource usage activities of the prototype app. The log is forwarded to the server through encrypted channel and stored remotely in the server.

AppOpsCommand is used to log the events[4]. The prototype checks for resource access events by each of the installed apps and writes respective events in a pre-defined format. The app stores the log records in a JSON file which contains three fields: 1) the name of the package/app, 2) the name of the accessed resource and, 3) the time of the resource access event. The following example shows a sample log record.

```
#root: adb shell appops get com.google.android.youtube
{"Package":"com.google.android.youtube","Permission":"READ_EXTERNAL_STORAGE",
"Timestamp":"Fri Mar 03 09:56:35 GMT+01:00 2017"}
```

4.2 Data Collection

The server is responsible for collecting, processing and storing logs sent from the mobile devices. The server is able to register participating devices/profiles. All the logs are first stored locally on the mobile device in a file (JSON) and is later sent to the server periodically (typically once a day). The server is also responsible for parsing the data and

[4] https://android.googlesource.com/platform/frameworks/base/+/android-6.0.1_r25/cmds/appops/src/com/android/commands/appops/AppOpsCommand.java;Accessed:2017-11-21

perform insert operations for the database. The database is consisted of three tables: 1) Main_info, 2) Time and 3) Coordinate. Figure 2 depicts the database schema. Package (or, app) and permission names are stored according to app's manifest. Time is stored in 'date+GMT+aa:aa:aa' format which enables it to handle data generated from any geographic location.

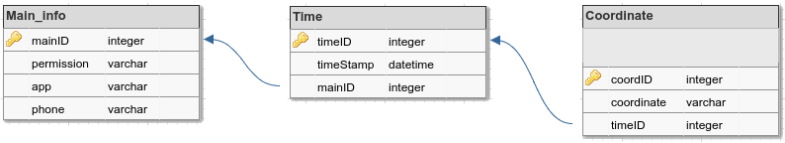

Fig. 2: Database Schema.

4.3 Analysis and Visualization

This part of the project is still in infant stage. Currently, a web-based interface can provide a brief summary to the resource usage events. We are also discussing the potentialities and properties of an effective visualization interface. For example, mean value of minimal resource usage could be calculated from the accumulated data within a given scenario and use it as a reference point to assess other permissions and apps. For similar scenarios ($S = S'$) and negligible discrepancy between intervals ($\Delta\tau \simeq \Delta\tau'$), condition for privacy-preserving behavior of an app is: $\overline{RS_\alpha} \geq RS'_\alpha$; where $\overline{RS_\alpha}$ = average risk score calculated from reference database and RS'_α = risk score of the target app.

Apart from the web interface, a personalized visualization feature is also included within the prototype app. In [MP17], we proposed a model to incorporate individual preference and to provide feedback based on self-defined threshold. The model includes a privacy measuring scale that possesses scalability within itself and it has the potential to offer additional *PIF* values. As the model is able to accommodate user-defined privacy preferences, it can be utilized to produce nudges in order to push the user toward privacy-preserving behavior. However, a meaningful threshold mechanism is yet to be defined and implemented. In Fig. 1, this is illustrated with dotted lines.

5 Discussion and Conclusion

From the pilot study, results and outcomes could be observed from two different angles. First, excessive permission usage and infraction from the principle of least privilege were highlighted [Mo17]. Second, a model for deriving partial identities from app permissions was presented which was induced from Pfitzmann and Hansen's terminology for privacy [PH10]. It also highlighted on the risks and likelihood of partial identity extraction [FM17].

Mobile app users pay with their personal data which is rather considered as an open secret. However, there is a hidden condition: price is undefined and uncontrolled; which leads to privacy risks. The installed apps possess unlimited access to sensitive user data without any time, frequency or volume constraint. This project aims to unearth such privilege induced risks and to provide a usable interface for assessing them. In order to inspect app behavior, the project collects log of resource access events, analyzes the data, displays summarized

statistics and seeks for a meaningful recommendation. It will allow the user to compare and be aware of privacy invasive behavior of apps and take initiatives to protect their private data.

References

[Au12] Au, Kathy Wain Yee; Zhou, Yi Fan; Huang, Zhen; Lie, David: Pscout: analyzing the android permission specification. In: Proceedings of the 2012 ACM conference on Computer and communications security. ACM, pp. 217–228, 2012.

[Ca18] Carlsson, Adrian; Pedersen, Christian; Persson, Fredrik; Söderlund, Gustaf: KAUDroid : A tool that will spy on applications and how they spy on their users. Technical report, Karlstad University, 2018. Working paper, Januari 2018.

[Fe11] Felt, Adrienne Porter; Chin, Erika; Hanna, Steve; Song, Dawn; Wagner, David: Android permissions demystified. In: Proceedings of the 18th ACM conference on Computer and communications security. ACM, pp. 627–638, 2011.

[FM17] Fritsch, Lothar; Momen, Nurul: Derived Partial Identities Generated from App Permissions. Open Identity Summit 2017, pp. 117–129, 2017.

[HBM17] Hammad, Mahmoud; Bagheri, Hamid; Malek, Sam: Determination and Enforcement of Least-Privilege Architecture in Android. In: Software Architecture (ICSA), 2017 IEEE International Conference on. IEEE, pp. 59–68, 2017.

[He08] Hedbom, Hans: A Survey on Transparency Tools for Enhancing Privacy. In: FIDIS. Springer, pp. 67–82, 2008.

[KCS13] Kelley, Patrick Gage; Cranor, Lorrie Faith; Sadeh, Norman: Privacy as part of the app decision-making process. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 3393–3402, 2013.

[Ke12] Kelley, Patrick Gage; Consolvo, Sunny; Cranor, Lorrie Faith; Jung, Jaeyeon; Sadeh, Norman; Wetherall, David: A conundrum of permissions: installing applications on an android smartphone. In: International Conference on Financial Cryptography and Data Security - FC2012 Workshops. Springer Berlin Heidelberg, pp. 68–79, 2012.

[Mo17] Momen, Nurul; Pulls, Tobias; Fritsch, Lothar; Lindskog, Stefan: How much Privilege does an App Need? Investigating Resource Usage of Android Apps. The Fifteenth International Conference on Privacy, Security and Trust (PST), 2017.

[MP17] Momen, Nurul; Piekarska, Marta: Towards Improving Privacy Awareness Regarding Apps' Permissions. Proceedings of the Eleventh International Conference on Digital Society and eGovernments - ICDS 2017, 2017.

[Pe12] Peng, Hao; Gates, Chris; Sarma, Bhaskar; Li, Ninghui; Qi, Yuan; Potharaju, Rahul; Nita-Rotaru, Cristina; Molloy, Ian: Using probabilistic generative models for ranking risks of android apps. In: Proceedings of the 2012 ACM conference on Computer and communications security. ACM, pp. 241–252, 2012.

[PH10] Pfitzmann, Andreas; Hansen, Marit: Anonymity, unlinkability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology. In: Designing privacy enhancing technologies. TU Dresden, pp. 1–9, 10-Aug-2010.

[We12] Wei, Xuetao; Gomez, Lorenzo; Neamtiu, Iulian; Faloutsos, Michalis: Permission evolution in the android ecosystem. In: Proceedings of the 28th Annual Computer Security Applications Conference. ACM, pp. 31–40, 2012.