Thomas Brüggemann

Master Thesis
im Fach Allgemeine Wirtschaftsinformatik

Automating the privacy risk assessment for mHealth apps

Themensteller: Jun.-Prof. Dr. Ali Sunyaev

Vorgelegt in der Masterprüfung
im Studiengang Information Systems
der Wirtschafts- und Sozialwissenschaftlichen Fakultät
der Universität zu Köln

Köln, Februar 2016

Contents

Index of Abbreviations

API	Application programming interface
APK	Android application package
DRM	Digital rights management
mHealth	Mobile health

1. Problem Statement

The market for mobile smart device applications (apps) is growing extensively in recent years. It is increasingly easier for small companies or even single developers to create unique apps that reach millions of users via app stores. This market growth also effected mobile health (mHealth) apps. More and more apps are available that support the users in solving their health related issues and information deficiencies. Users are asked to input their personal information in order to tailor the app to their custom needs. The users are asked to expose vulnerable information about their health status while it remains mostly unclear how and where the data is processed, stores or tossed along.

The information about privacy practices of the app providers should be found in the privacy policy document provided by the app provider. Processing these privacy policies-requires a higher level of education and time to read through large documents of text to find the relevant information. Additionally, the important information is hidden in legal language or insufficiently addressed, if at all.[1]

Aside from the data usage beyond users control, it is also challenging for users to assess what kind of privat information an app asks for, in order to tailor the app experience. Users have to download the apps of interest and try them out, before it becomes clear what information is used. This leads to low inter comparability between apps. If users are looking for specific functionality in a mHealth app, it is challenging to find the app that offers the desired functionality at an acceptable privacy risk. Even if users would persue the goal to find and compare mHealth apps of similar functionality, the high volume of apps available on the app stores[2] makes it not feasable to review all of them by hand.

Resolving the challenges in evaluating the privacy risk of mHealth apps, before usage and of large volumes, will result in an improved decision making process for users. It also reduces the danger of exposing vulnerable information. A way to assess privacy risks of mHealth apps would be to automate the review of each individual app. Automating the review process for large scale app assessments has the potential to grow new privacy evaluation service markets.[3] Automating the app assessment for potential privacy risk

---

1 Dehling, Gao, Sunyaev (2014), p. 11

2 Enck et al. (2011), p. 1

3 Enck et al. (2011), p. 14

factors can be done by downloading and analyzing the source of each app to trace data leaks. Static code analysisis used in the field of informatics to analyse application source code to detect faults or vulnerabilities.[4] A static code analysiscould potentially be used to assess the privacy risks that mHealth apps pose.

The automated process of assessing the privacy risk helps to reduce the costs of reviewing each individual app and enhances the information experience users get while researching mHealth apps. Additionally, it exposes new possibilities for research in the privacy risk area. The research could be conducted on providing solutions and best practices for minimizing the privacy risk of apps. It is unclear if, and to what degree, the concepts of static code analysis and privacy risk assessment can be combined in order to automate the app assessment. This leads to the research question of this master thesis. How and to what degree can the privacy risk assessment of mHealth apps be automated?

TODO: PREVIOUS RESEARCH Previous research in this field revield...

2. Objectives

3. Definitions

Certain terms are used in the remainder of this thesis that need to be defined:

3.1 Mobile health applications

3.2 Static code analysis

3.3 Decompilation

4. Methods

4.1 Automating the privacy risk assessment

In order to automate the privacy risk assessment of mHealth apps via static code analysis, it is necessary to gain access the source code of the apps. While uploading a new app to the Apple AppStore, Apple's digital rights management (DRM) system encrypts the binary file in a way that makes recovering the source code difficult. There are approaches

---

[4] Baca, Carlsson, Lundberg (2008), p. 79

on decompiling the Apple app binary back into its source code. These approaches involve unlocking and jailbreaking [5] an Apple iPhone or iPad, which is a violation of the Apple terms of service and therefore forbidden[6]

The Google PlayStore on the other hand hosts Android application in APK containers that are non encrypted and allow for decompilation back into the original source files. In order to automate the download process of APK files to our local computer, we found a hidden Google API that reveals access to the APK files from the PlayStore. The API can be queried by sending an Android device id along, effectively pretending to be a requesting Android device. The result of the query is a binary APK file.

Xu2015 provision a repository of 10000 apps that are extracted from the Google PlayStore within the categories ”Health & Fitness” and ”Medical”. Due to the obstacles of gaining access to the Apple iOS binary files, we restrict the dataset to the available Android apps and conduct our automated privacy risk assessment on these apps.

4.2 Evaluation

5. Structure

6. Expected Results

7. Problems

So far there are no open questions or problems. Though, a problem could arise from the fact that the core of this thesis relies on a undocumented Google API for downloading the APK files of the apps. If this API is shut down or somehow secured from open usage, it would not be as easy to gather the needed APK files for the static code analysisas it currently is.

---

[5] Jailbreaking revers to the action of removing ... Kweller (2010), p. 1

[6] Kweller (2010), p. 1

References

Baca, Carlsson, Lundberg (2008)

Dejan Baca, Bengt Carlsson, Lars Lundberg: Evaluating the cost reduction of static code analysis for software security. In: Proceedings of the third ACM SIGPLAN workshop on Programming languages and analysis for security - PLAS '08. 2008, p. 79

Dehling, Gao, Sunyaev (2014)

Tobias Dehling, Fangjian Gao, Ali Sunyaev: Assessment Instrument for Privacy Policy Content: Design and Evaluation of PPC. In: WISP 2014 Proceedings. 2014,

Enck et al. (2011)

William Enck, Damien Octeau, Patrick McDaniel, Swarat Chaudhuri: A Study of Android Application Security. In: USENIX security .... Nr. August, 2011, pp. 21–21

Kweller (2010)

Abby Kweller: Jailbreaking and Unlocking the iPhone: The legal implications. In: Polymer Contents. Nr. 8, Jg. 27, 2010, pp. 480–523