Ph.D. Proposal

Enabling Privacy-sensitive Context Monitoring in Crowdsourced Software Developments

Supervisors

Walter Rudametkin (Associate professor, Spirals) <Walter.Rudametkin@inria.fr>
Romain Rouvoy (Associate professor, HDR, Spirals) <Romain.Rouvoy@inria.fr>
Lionel Seinturier (Professor, Spirals) <Lionel.Seinturier@inria.fr>

Research team

The Ph.D. student will join the Spirals project-team.

Spirals Project-team
https://team.inria.fr/spirals

Inria
Parc Scientifique de la Haute Borne
40, avenue Halley - Bat. B, Park Plaza
59650 Villeneuve d'Ascq – FRANCE

Scientific Context

Developing mobile applications, also known as *apps*, has become extremely popular. This is in part due to the popularity of the mobile devices themselves, which now outnumber desktops, and in part due to the time users spend on them [1]. The app ecosystem is supported by app stores, such as Google Play, Apple App Store, Amazon Appstore, and F-Droid, among many others. There are currently millions of apps available from these stores [2], many are downloaded thousands of times per day. For example, in 2013 users downloaded more than 50 billions of apps from the Google Play Store [3], and it is estimated that the number of downloads will reach 300 billion in 2017 [4].

No non-trivial app is without bugs. Nevertheless, app developers are faced with the demanding task of ensuring their apps run correctly on a multitude of devices. Mobile devices are far from identical, there are many brands, makes and models. Differences between mobile devices can be subtle to an app developer, such as the screen technology used (OLED vs IPS).

Or they can be quite important, such as different CPU architectures (ARM vs Intel). There are even many variant devices that share the same model name (*e.g.*, Samsung Galaxy S6 has at least 6 different models). Furthermore, the same device may be running different versions or variants of the underlying Operating System, libraries or SDK. This is called *device fragmentation* [5], and it complicates the task of developing apps because it is too costly and time consuming to test an app for every device that exists. Furthermore, some bugs only appear on certain devices or under certain conditions.

Crash reproduction is essential for app developers to promptly fix their apps. To reproduce crashes, users provide information regarding the conditions that produced the failure. Developers then attempt to reproduce the issue, but due to different context conditions (*e.g.*, network conditions, location, temperature) and device fragmentation, it is difficult to simulate the different conditions necessary to produce the crash. Furthermore, the information that users provide to developers may contain privacy sensitive data that the user may not wish to share. This often causes users to be reluctant to share the information that lead to a crash.

Ph.D. Project

Positioned in the context of mobile application development, this Ph.D. project will focus on numerous issues developers face when attempting to develop, monitor, and debug mobile applications that run in heterogeneous environments.

This Ph.D. will build upon our results with the crowdsourcing tool MoTiF [6], which which monitors the execution of Android apps in mobile devices to detect app crashes. Specifically, MoTiF collects *user interactions (e.g., clicks, touches)* and *context information* (*e.g.*, sensors state, memory state) during the execution of apps in a multitude of real devices. Then, MoTiF identifies *crash patterns* (minimum set of steps and context conditions) which are relevant to reproduce the crashes. This Ph.D. will focus on providing developers with crash reproduction, while ensuring that the user's privacy is not compromised. This can be done at two levels: (i) locally on the device, the monitoring service can ensure that sensitive data is wiped from the reports, and (ii) on the server side, the collection service should provide an anonymisation process to ensure that the reports do not compromise the user's privacy and cannot lead back to the submitter.

The objective of this Ph.D. is to crowd-source application monitoring to provide large amounts of data that will be used to reproduce crashes, all while ensuring the user's sensitive data remains private.

In order to do so, we propose the following strategy:

- **Evaluate the state-of-the-art software approaches in mobile application development and crash reproduction.** There exists much work in the area of Application Monitoring [7], Crowdsourcing [8], Crash Reproduction [9], and Mobile Data Privacy [10] that is related.

- **Build a taxonomy of *crash patterns* that can be extracted from mobile applications.**
  This is a first step in understanding crash and in building families of crashes that can lead to their reproduction. Such information can be extracted from MoTiF.
- **Automatically convert *crash patterns* into *test cases.*** The test cases should reproduce, as best as possible, the context and conditions that lead to the failure.
- **Build a taxonomy of sources and types of sensitive data in mobile applications.** In order to ensure that sensitive data is not submitted in crash reports, and that crash reports cannot be used to, e.g., uniquely fingerprint devices [11], the sources of the data need to be classified.
- **Provide device-side data privacy.** Develop techniques to analyse and remove privacy-sensitive data from crash reports before they leave the device, all while retaining the utility of the crash report itself. It is essential to retain crash reproducibility. This goes beyond simply finding passwords, names and other data. Collecting configuration and device data can lead to highly discriminating fingerprints that are statistically unique [11].
- **Provide server-side anonymisation.** The collected data should be used in aggregate, removing any statistically identifying information from the reports. Aggregate reports can then be provided to developers and used to reproduce crashes.

The outcome of this work will be a privacy sensitive monitoring tool that uses crowdsourcing to create privacy friendly, anonymized, reproducible crash tests.

Skills Summary

The Ph.D. candidate will develop her/his skills in Java, Android, Spoon, Linux, Docker, Git, among many other technologies.

As is a common practice in the Spirals research team, all source code will be open sourced using either the GPL or the Apache License. It is expected that the student participate in related open source communities. This should also assist in the technological transfer from academic prototype to industrial-ready tools.

Experimentations to demonstrate the effectiveness of developed tools should be performed on popular mobile applications that exhibit real world issues.

References

1. **“Mobile internet usage soars by 67%,”** http://gs.statcounter.com/press/.
2. **“Number of apps available in leading app stores as of July 2015”**, http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/
3. “**Cumulative number of apps downloaded from the Google Play Android app store as of July 2013 (in billions)**.”

http://www.statista.com/statistics/281106/number-of-android-app-downloads-from-google-play

4. L. Columbus. **Roundup of mobile apps and app store forecasts**, 2013, June 2013.
5. "**Android Fragmentation Visualized**" https://opensignal.com/reports/2015/08/android-fragmentation/
6. M. Gomez, R. Rouvoy, and L. Seinturier, "**Reproducing Context-sensitive Crashes in Mobile Apps using Crowdsourced Debugging**", Research Report RR-8731, Inria Lille; INRIA. 2015 https://hal.inria.fr/hal-01155597/document
7. S. Herbold, J. Grabowski, S. Waack, and U. Bunting. "**Improved bug reporting and reproduction through non-intrusive gui usage monitoring and automated replaying**". In Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on, pages 232–241. IEEE, 2011.
8. M. Gomez, M. Martinez, M. Monperrus, and R. Rouvoy. "**When App Stores Listen to the Crowd to Fight Bugs in the Wild**". In 37th *International Conference on Software Engineering* (ICSE), track on *New Ideas and Emerging Results* (NIER), Firenze, Italy, May 2015. IEEE.
9. T. Roehm, N. Gurbanova, B. Bruegge, C. Joubert, and W. Maalej. **Monitoring user interactions for supporting failure reproduction**. In Program Comprehension (ICPC), 2013 IEEE 21st International Conference on, pages 73–82. IEEE, 2013.
10. Marouane Fazouane, Henning Kopp, Rens Van Der Heijden, Daniel Le Métayer and Franck Kargl. **Formal verification of privacy properties in electrical vehicle charging.** In International Symposium on Engineering Secure Software and Systems (ESSOS15), Mar 2015, Milan, Italy. 2015
11. Andreas Kurtz, Hugo Gascon, Tobias Becker, Konrad Rieck and Felix Freiling. **Fingerprinting Mobile Devices Using Personalized Configurations.** Proceedings on Privacy Enhancing Technologies (PoPETS), 2016 (1) , 4–19, to appear 2016.

CONTROL OVER PERSONAL DATA: TRUE REMEDY OR FAIRY TALE?
http://script-ed.org/wp-content/uploads/2015/06/lazaro_metayer.pdf

http://www.assemblee-nationale.fr/14/rapports/r3119.asp

Formal verification of privacy properties in electrical vehicle charging.
https://hal.inria.fr/hal-01089925

Mobilitics: Analyzing Privacy Leaks in Smartphones
http://ercim-news.ercim.eu/images/stories/EN93/EN93-web.pdf

Par exemple : http://planete.inrialpes.fr/~achara/mobilitics/