VALERA: An Effective and Efficient Record-and-replay Tool for Android

Yongjian Hu
University of California, Riverside
yhu009@cs.ucr.edu

Iulian Neamtiu
New Jersey Institute of Technology
ineamtiu@njit.edu

Abstract

We demo VALERA, a Versatile-yet-lightweight Record-and-replay tool for Android. Record-and-replay is useful across the Android development lifecycle, from bug reproducing to systematic testing. VALERA uses a novel technique named *sensor-oriented replay* (recording and replaying sensor and network input, event schedules, and inter-app communication via intents) to achieve high accuracy and low overhead. VALERA can be used as an effective replay tool on both real phones and emulators. Evaluation on more than 50 popular Android apps shows that VALERA's performance overhead for either record or replay is just 1%. We demonstrate how VALERA can be used in many development scenarios: bug reproducing, regression testing, event-driven race reproduction and verification, mutation testing via fuzzy replay, and cross-app testing.

Categories and Subject Descriptors D.2.4 [*Software Engineering*]: Software/Program Verification—Reliability, Validation; D.2.5 [*Software Engineering*]: Testing and Debugging—testing tools

General Terms Reliability, Verification

Keywords Mobile applications, Record-and-replay, Google Android, App testing, Event-based races

1. Introduction

As smartphones and the applications ("apps") running on them continue to grow in popularity [6], there is a growing trend to develop high quality, reliable, secure and energy efficient apps. The mobile development lifecyle includes activities such as systematic testing, reproducing bugs, and performance/energy profiling. Recording and replaying the execution of smartphone apps is particularly useful in these contexts.

However, record-and-replay on mobile devices has proven difficult: smartphone apps revolve around concurrent streams of events that have to recorded and replayed with precise timing. To keep overhead low, prior record-and-replay approaches for smartphones only capture GUI input [5] which hurts accuracy as they cannot replay input from the network or sensors; or events, to reproduce event-based race [7]. Prior work on record-and-replay for desktop and server platforms [1, 8] has relied on techniques such as hardware changes, or instruction logging which is too heavyweight for mobile apps. To address these challenges, we have developed VALERA (VersAtile yet Lightweight rEcord and Replay for Android) [4], a novel *sensor- and event-stream driven* approach to record-and-replay; VALERA is practical and has been designed to meet several key desiderata:

1. *Support I/O (sensors, network) and record system information required to achieve high accuracy and replay popular, full-featured apps.*
2. *Accept APKs as input—this is how apps are distributed on Google Play—rather than requiring access to the app source code.*
3. *Work with apps running directly on the phone, rather than on the Android emulator which has limited support for only a subset of sensors.*
4. *Low overhead to avoid perturbing the app's execution.*
5. *Require no hardware, kernel, or VM changes.*

Our experiments [4] have shown that VALERA is able to replay 50 popular Android apps which use a variety of sensors with low overhead (about 1% for either record or replay). VALERA is effective for reproducing bugs by replaying the input and event schedule that led to an error state. With the support of deterministic replay of event schedules, VALERA is able to reproduce hard-to-debug event-driven races. VALERA can also verify races reported by a race detector via event flipping, and classify the race to be benign, harmful of false positive by comparing the outcome states of different schedules. Finally, VALERA provides useful test-

MobileSoft '16, May 16-17 2016, Austin, TX, USA.
Copyright © 2016 ACM 978-1-4503-4178-3/16/05…$15.00.
http://dx.doi.org/10.1145/2897073.2897712

ing techniques such as semantic sensor data alteration and cross-app testing to help developers and testers.

2. Installation and Usage

2.1 Building and Installing

VALERA is open source; build instructions are available on its website.[1] While VALERA supports both the Android emulator and real devices version, for real devices additional hardware drivers are required, e.g., proprietary drivers for Google's Nexus series phones [2].

2.2 Basic Usage

In this demo, we run VALERA on an open source app named Nori on a Galaxy Nexus phone. First, the user needs to run `adb root` (the log data is saved in the app's private location and VALERA needs root privilege to access it). Second, the app is installed via `adb install nori.apk`. Third, the user needs to write a *config* file describing the app. The config file must have the app's package name, main entry activity, options for tracing (on/off) and recording network (on/off) or not and the app's home folder. Here is an example:

```
PKG=pe.moe.nori
MAIN=pe.moe.nori.SearchActivity
TRACING=0
NETWORK_REPLAY=1
APP_HOME_DIR=/data/data/$PKG
```

To start recording the app, the user runs the command:

```
./script/run.sh --config=config.txt --cmd=record
```

Users can then run the app as usual. VALERA will automatically save some key information into log files, e.g., `input.bin` for input events, `io.bin` for HTTP/HTTPS network data, `record.trace` for event schedules.

Replaying the app is very simple, the user just invokes:

```
./scripts/run.sh --config=config.txt --cmd=replay
```

VALERA then automatically replays the recorded execution with deterministic input and schedule.

3. Applications

VALERA has many applications in Android development; we demo several.

3.1 Reproducing Bugs

Reproducibility is key to bug fixing. VALERA provides significant help for reproducing bugs by recording executions until a bug is encountered and then replaying the trace. We tested VALERA's bug reproducing capabilities using actual bugs in popular apps. The bugs varied from incorrect file format, to invalid input, to stress testing, to erroneous scripts or plugins [5].

[1] `http://spruce.cs.ucr.edu/valera/`

3.2 Semantic Sensor Data Alteration

To test the stability of an app, or create new test cases by fuzzing existing executions, VALERA provides features to alter the recorded sensor data in a semantically meaningful way [3]. For example, VALERA can alter GPS readings (e.g., inject a `null` location object to simulate GPS hardware exceptions), blur or darken the pictures captured by camera to emulate different physical environments, or change the sample rate of the audio data to test the reaction of the app towards different audio qualities.

3.3 Reproducing Event-driven Races

Event-driven races in Android are hard to debug and reproduce by current record-and-replay tools. With deterministic event order recorded, VALERA can help reproduce these races by preserving the exact event ordering [4]. Our experiments show that VALERA can do this effectively on several open source apps: we were able to reproduce harmful event-driven races that crash the app (e.g., Tomdroid) or lead to incorrect GUI view state (e.g., NPR News).

4. Conclusions

We have demoed VALERA, an approach and tool for versatile, low-overhead, record-and-replay of Android apps. Experiments with using VALERA on popular apps from Google Play, as well as replaying event race bugs, show that our approach is effective, efficient, and widely applicable to various development tasks such as testing, finding and fixing bugs.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1064646.

References

[1] G. W. Dunlap, S. T. King, S. Cinar, M. A. Basrai, and P. M. Chen. Revirt: enabling intrusion analysis through virtual-machine logging and replay. In *OSDI'02*.

[2] Google. Binaries for Nexus Devices, 2015. `https://developers.google.com/android/nexus/drivers`.

[3] Y. Hu and I. Neamtiu. Fuzzy and cross-app replay for smartphone apps. In *AST'16*.

[4] Y. Hu, T. Azim, and I. Neamtiu. Versatile yet lightweight record-and-replay for android. In *OOPSLA'15*.

[5] L. Gomez, I. Neamtiu, T.Azim, and T. Millstein. Reran: Timing- and touch-sensitive record and replay for android. In *ICSE '13*.

[6] M. Ronkko and J. Peltonen. Software industry survey, 2013. `http://www.softwareindustrysurvey.org/`.

[7] P. Maiya, A. Kanade, and R. Majumdar. Race detection for android applications. In *PLDI'14*.

[8] S. Narayanasamy, G. Pokam, and B. Calder. Bugnet: Continuously recording program execution for deterministic replay debugging. In *ISCA '05*.