Deep Dynamic Analysis of Android Applications

By

ERIC DAVID GUSTAFSON
B.S. (California Polytechnic State University, San Luis Obispo) 2011

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Hao Chen, Chair

Matthew Bishop

Karl Levitt

Committee in Charge

2014

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

To my parents, for instilling in me the determination to follow my dreams.

Contents

List of Figures

List of Tables

Abstract

Deep Dynamic Analysis of Android Applications

The smartphone revolution has brought about many new computing paradigms, which aim to improve upon the computing landscape as we knew it. Chief among these is the "app", packetizing and trivializing the distribution and installation of software. This has led to a boom in the mobile software industry, but also an increased burden on security researchers to ensure the millions of apps available do not harm users.

This paper presents a partial solution to that problem, Pyandrazzi, a practical dynamic analysis system for Android applications. Pyandrazzi aims to be more scalable, more compatible, and more thorough than any existing system, and to provide more informative data to analysts than was previously thought possible. The system is a true black-box solution, and is able to perform this analysis without any source code or prior knowledge of the application whatsoever. Unlike other similar systems, which rely heavily on unrealistic modifications to Android, our system employs the original Android virtual machine and libraries, to provide a more natural environment for apps, and to ease portability to new Android versions. Novel contributions include an algorithm for more thoroughly exploring application modeled on common user interface design patterns, a platform version-independent means of obtaining method trace data, and a method of using this data to calculate the method coverage of an application execution.

To evaluate the performance and coverage of the system, we used 1750 of the top applications from the dominant Google Play app market, and executed them under a variety of conditions. We demonstrate that the algorithm we developed is more effective than random user interface interactions at achieving method coverage of an application. We then discuss the performance of the system, which can execute all 1750 apps, for two minutes of run-time each, under heavy instrumentation, in about 7 hours.

We then explore two practical applications of the system. The first is a Host-based Intrusion Detection System (HIDS) concept implemented using application re-writing techniques. The system uses signatures based on high-level API call activity, as opposed to binary fingerprints or system call traces used in other systems. In our tests, we were able to reliably detect three families of malware for which we created signatures with zero false positives.

Secondly, we explore Pyandrazzi's role in a recent study of advertising fraud on Android, covering over 130,000 Android applications. The system was used to analyze those apps that did not generate ad-related traffic without user interaction. Of the 7,500 apps without such traffic, we found that 12.8% of applications would have generated ad traffic, if they had been properly interacted with via their user interfaces. We then explore augmenting Pyandrazzi to avoid interacting with advertising so that fraudulent behaviors can be better detected. Using a set of rules based on advertising industry standards and common design patterns, we were able to avoid ad-related interactions in 97.6 percent of a test set of 1,000 apps.

Acknowledgments

Firstly, I would like to thank Kristen Kennedy, without whom the Pyandrazzi project would not exist. Her dedication and tenacity helped this project get to where it is now, despite all the challenges we've faced.

Secondly, I owe a huge debt of gratitude to my colleague Benjamin Davis for his invauable assistance with the coverage measurement portion of the work, and the use of the RetroSkeleton rewriting framework.

I'd also like to thank Bogdan Copos for his collaboration on the Intrusion Detection System extension to Pyandrazzi and the associated analysis.

Additionally, I'd like to thank Ryan Stevens and Jon Crussell for their help in brining Pyandrazzi and the MAdfraud project together, as the findings have benefitted all of our work significantly.

Lastly, I'd like to thank the entire UC Davis Security Lab, especially my advisor Hao Chen, for their thoughtful support and encouragement throughout the project