University of Technology Sydney

Faculty of Engineering and Information Technology

Machine Learning Detection and Analysis on Obfuscated Android Malware

by

Yanxin Zhang

A Thesis Submitted

for the Degree of

Doctor of Philosophy

Sydney, Australia

January 20, 2022

Certificate of Authorship/Originality

I, Yanxin Zhang declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science, Faculty of Engineering and IT at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature: Signature removed prior to publication.

Date: January 20, 2022

Abstract

In recent years, the Android platform is the prevalent operating system platform, and it accounts for more than half of the world's mobile operating system market share. With the popularity of Android smartphones and Android tablets, Android-based malware has also developed rapidly, so malicious code detection technology based on the Android platform has been proposed, and Android application security must be carefully studied and have an extensive discussion. Meanwhile, machine learning methods can simulate the behavior of Android applications and distinguish between benign and malicious software; therefore, it is becoming the key to the effective detection of Android malware. However, malware developers may understand detection systems and take countermeasures against detection methods in the real world. Most current research work ignores this problem.

Recent research in machine learning has shown that learning-based systems are susceptible to well-designed adversarial examples. For instance, malware detection based on static analysis will encounter the challenge of code obfuscation, which is the countermeasure for malware authors to bypass detection. Therefore, there is a strong demand for research on Android malware detection from the perspective of defenders and attackers.

My research direction is to explore machine learning methods to ensure the security and stability of Android applications. Malware writers often use obfuscation techniques to obfuscate Android malware to evade detection by Android malware detectors. Knowing about the impact of code obfuscation on Android malware can provide valuable insights into obfuscated malware and, therefore, the design of resilient Android malware detectors.

First, labeling malware or malware clustering is essential for identifying new security threats, triaging, and building reference datasets. The state-of-the-art Android malware clustering approaches rely heavily on the raw labels from commercial AntiVirus (AV) vendors, which causes misclustering for a substantial number of

weakly-labeled malware due to the inconsistent, incomplete, and overly generic labels reported by these closed-source AV engines, whose capabilities vary greatly and whose internal mechanisms are opaque (i.e., intermediate detection results are unavailable for clustering). The raw labels are thus often used as the only significant source of information for clustering. To address the limitations of the existing approaches, we present ANDRE, a new Android Hybrid Representation Learning approach to clustering weakly-labeled Android malware by preserving heterogeneous information from multiple sources (including the results of static code analysis, the meta-information of an app, and the raw-labels of the AV vendors) to learn a hybrid representation for accurate clustering jointly. The learned representation is then fed into our outlier-aware clustering to partition the weakly-labeled malware into known and unknown families. The malware whose malicious behaviors are close to those of the existing families on the network is classified using a three-layer Deep Neural Network (DNN). The unknown malware is clustered using a standard density-based clustering algorithm. The evaluation shows that ANDRE effectively clusters weakly-labeled malware that cannot be clustered by the state-of-the-art approaches, achieving comparable accuracy with those approaches for clustering ground-truth samples.

Second, Android piggybacked malware (i.e., apps that piggyback malicious code) are becoming ubiquitous in app stores. Malware writers often use obfuscation techniques to obfuscate piggybacked apps to evade detection by Android malware detectors. Previous studies in this field have focused on the impact of code obfuscations on the detection of piggybacked malware, but the impact of code deobfuscation on detecting obfuscated piggybacked apps has rarely been investigated. Knowing about the impact of code deobfuscation can provide helpful insights into obfuscated piggybacked apps and, therefore, the design of resilient Android malware detectors. We conduct an empirical study of code deobfuscations on detecting obfuscated Android piggybacked apps, focusing on three types of malware detectors: commercial anti-malware products, machine learning-based detectors, and similarity-based detectors. We observe that code deobfuscations can have various impact on the detection rate for different types of malware detectors, such as similarity-based or

machine learning-based detectors. Also, we observe that the examined deobfuscation tools have a different impact on obfuscated piggybacked apps after deobfuscations.

Thrid, Android malware adversarial samples can help test the weaknesses of antivirus products to improve their robustness against emerging and sophisticated Android malware. A typical method to generate testing samples is to transform a single Android malware into a range of samples through code obfuscation. Existing obfuscation tools often have many obfuscation options, which can generate a large number of testing samples. However, accurately identifying top evasive samples to gain an optimal cost-benefit trade-off is a challenge. To address the above issue in adversarial malware generation for robustness testing of antivirus products, we present ALBS(Active Learning-based Sampling), an active-learning-based approach, by constructing an adversarial ranking model to select the most powerful evasive adversarial samples. First, ALBS generates all Android malware adversarial samples and then extracts the adversarial samples' representation using both unstructured and structural code features. Then, it utilizes the multi-view multi-learner (MVML) active learning method to train a model which ranks the adversarial samples using multiple learning-to-rank (LTR) algorithms. Our empirical experiments demonstrate the effectiveness of ALBS for generating adversarial samples for antivirus product robustness testing. The experiment result indicates that the proposed active learning approach can optimize the adversarial sample generation process and effectively select the most potent testing samples, improving the efficiency of Android antivirus products' robustness testing.

Acknowledgements

I want to express my deep gratitude to all those who have given me help and support in writing my thesis. First of all, I would like to thank my supervisors, Dr. Yulei Sui and Prof. Ivor Tsang, for their guidance, support, and advice during my PhD study. They discovered my potential, encouraged me to go further, and provided precious lessons in many different fields. Their meticulous academic attitude and dedicated work style have always been my role models. Also, I am grateful to the University of Technology Sydney for providing generous supports and attractive academic environment during my candidature career.

I would also like to thank the colleges in my group for discussing research work with me and sharing their knowledge and suggestions generously. They provided many insightful discussions for my research and offered various help to my personal life.

Finally, I would like to thank my parents and wife for their love, care, and continuous support and encouragement. Without their valuable support, it would be impossible for me to finish this thesis.

Sydney, Australia, January 20, 2022

List of Publications

Below is a list of publications that are included in this thesis as Chapter 3, 4, 5 respectively.

Chapter 3:

Zhang, Y., Sui, Y., Pan, S., Zheng, Z., Ning, B., Tsang, I. and Zhou, W., 2019. "Familial clustering for weakly-labeled android malware using hybrid representation learning." *IEEE Transactions on Information Forensics and Security, 15, pp.3401-3414.* Status: Published

Chapter 4:

Zhang, Y., Xiao, G., Zheng, Z., Zhu, T., Tsang, I.W. and Sui, Y., 2020, December. "An Empirical Study of Code Deobfuscations on Detecting Obfuscated Android Piggybacked Apps." *In 2020 27th Asia-Pacific Software Engineering Conference (APSEC) (pp. 41-50). IEEE.* Status: Published

Chapter 5:

Zhang, Y., Cai, H., Tsang, I.W. and Sui, Y. "ALR: Active Learning Based Ranking for Adversarial Android Malware Generation." *Submitted to IEEE Transactions on Reliability.* Status: Under review

Other publications during candidature:

Zhang, Y.. "Obfuscation Resilient Lightweight Android Repackaged Apps detection. " *Submitted to IEEE Transactions on Dependable and Secure Computing.* Status: Under review

Contents

List of Figures

List of Tables