Poster: Towards Sustainable Android Malware Detection

Haipeng Cai
Washington State University
Pullman, WA, USA
hcai@eecs.wsu.edu

John Jenkins
Washington State University
Pullman, WA, USA
john.jenkins@wsu.edu

ABSTRACT

Approaches to Android malware detection built on supervised learning are commonly subject to frequent retraining, or the trained classifier may fail to detect newly emerged or emerging kinds of malware. This work targets a *sustainable* Android malware detector that, once trained on a dataset, can continue to effectively detect new malware without retraining. To that end, we investigate how the behaviors of benign and malicious apps evolve over time, and identify the most consistently discriminating behavioral traits of benign apps from malware. Our preliminary results reveal a promising prospect of this approach. On a benchmark set across seven years, our approach achieved highly competitive detection accuracy that sustained up to *five* years, outperforming the state of the art which sustained up to two years.

ACM Reference Format:
Haipeng Cai and John Jenkins. 2018. Poster: Towards Sustainable Android Malware Detection. In *Proceedings of ICSE, Gothenburg, Sweden, (ICSE'18),* 2 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

A major defense technique for securing the Android ecosystem has been app *classification* based on machine learning (ML), which identifies malware by predicting a given app as benign or malicious. The technique typically works by first training a classifier based on a set of features extracted from labeled sample apps, and then applying the trained classifier to unlabeled apps using the same feature set. In the current relevant literature, ML-based malware detectors have achieved very-high accuracy. Yet primarily, this accuracy has only been attained for classifying apps developed *in the same time period* as the training data. The reason is that both the attack strategies of malware and the platforms of Android itself evolve rapidly. As a result, the classifier needs to be constantly retrained with new malware samples (i.e., they are not sustainable).

Developing a sustainable malware detector is crucial for Android because otherwise the detector, without retraining, would not be able to identify new malware, which presently continues to emerge and surge [1]. Even if retraining is an option, it is not always practical because getting training samples for emerging malware may not be possible. Meanwhile, the cost of computing features and training typically dominates the total detection cost. A state-of-the-art malware detector with demonstrated sustainability, MamaDroid [4], made important advances in this regard, yet it only sustained high accuracy for one year.

Towards a sustainable solution to detecting Android malware, we aim to explore a set of behavioral features that not only clearly differentiate benign apps from malicious ones, but consistently do so for a long period of time with expected rapid changes in both the Android platform and its user applications. To that end, we start with understanding how Android apps evolve over a long period of time through a longitudinal characterization study. This study led us to a set of *dynamic* features as regards to the *extent*, *frequency*, and *distribution* of the sensitive accesses in Android apps, which consistently separated benign apps from malware, despite the seven years of evolution across each group that we investigated. Based on these features, we developed a malware detector based on random-forest classification. Our preliminary results show that our approach substantially outperformed MamaDroid [4] as the baseline with high accuracy (F-measure of over 93%) sustained for as long as four years, while still keeping a reasonable performance (F-measure of 82%) even five years after training.

2 APPROACH

Feature engineering is at the core of any ML-based malware detection approach. We started with the 122 behavior metrics used in our prior work [3] which characterize the execution of an app in terms of the interaction between user code and libraries, distribution of components and inter-component communications, and classification of callbacks. We further considered metrics on the extent (e.g., percentage of callsites of one kind over the total callsites exercised), frequency (e.g., percentage of call instances of certain kinds), and distribution (e.g., categorization according to the kinds of information accessed) of source/sink invocations. Finally, we included metrics of dynamic control flows at method level concerning sensitive APIs (i.e., *sources* and *sinks*) only.

With all these metrics and using our characterization toolkit [2], We characterized 3,431 benign and 3,001 malicious samples from different years and diverse sources. We compared each of the metrics between the two groups in each year, and identified 52 of them that clearly differentiated benign apps from malware. All of the selected metrics are relative statistics (i.e., percentages) concerning sensitive data accesses via APIs, which were computed purely from a 10-minute trace per app, driven by random inputs. These 52 metrics focus on the distribution of source/sink invocations, such as the percentage of calls to sources that retrieved location information, as well as reached at least one sink on the dynamic call graph constructed from the trace.

A dynamic malware detector is then built on a ML-based classifier that uses the 52 metrics as features. Like existing

ICSE'18, , Gothenburg, Sweden

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

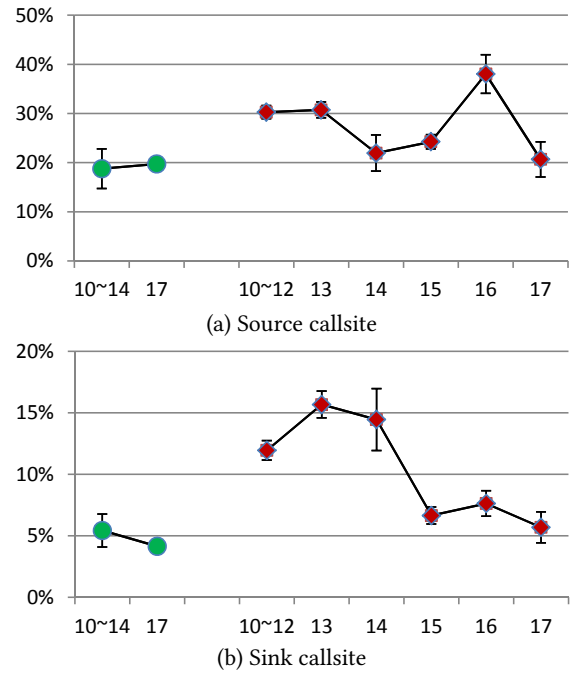

Figure 1: Mean percentage of calls for sensitive accesses, y axis) over years (x axis) in benign apps (green circle) and malware (red diamond). Error bars show the 0.95 confidence intervals of the means.

ML-based app classification systems, our detector works in two major phases: training and testing. In the training phase, the classifier is trained on a set of labeled app samples. Then in the testing phase, the trained classifier is applied to classify novel apps. In particular, given the motivation of this work, the goal of this detector is to work effectively (i.e., with high classification accuracy) some time after it is trained, so that *emerging* malware can be discovered without retraining it on samples of the new malware. Intuitively, the longer the time span the better, and more sustainable the detector.

3 RESULTS

As part of our characterization results, Figure 1 depicts the evolutionary patterns regarding the *extent* of sensitive accesses, in terms of the source and sink callsites. As shown, regardless of the fluctuations seen by both malware and benign apps, the differences in the extent of sensitive accesses between the two groups were very consistent: a visual horizontal dividing line between them could be drawn in each of the two charts. In other words, while both malware and benign apps have changed significantly within the respective group, the patterns of the *differences* appeared to be quite stable, suggesting the two *extent* metrics shown to be able to consistently differentiate malware from benign apps. We observed similar patterns with respect to all other 50 metrics.

We evaluated our malware detector in terms of classification performance and efficiency, versus *MamaDroid*. We chose *MamaDroid* as the baseline approach because it is a state-of-the-art ML-based malware detection system and shares a similar goal to ours (i.e., sustainable detection). Specifically, we

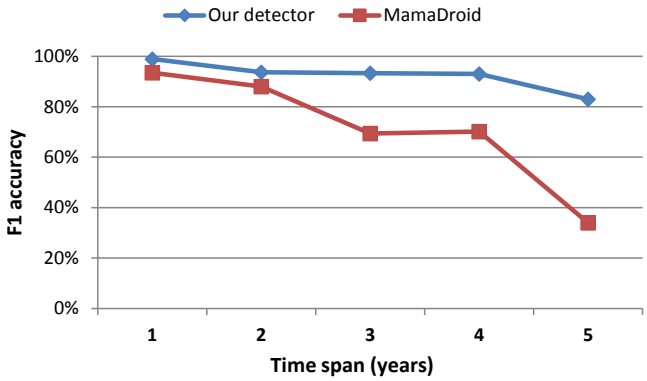

Figure 2: Classification performance (y axis) of our detector versus *MamaDroid*, trained on benign apps and malware from year 2012 or earlier, for detection over five years (x axis)–testing on benign app and malware developed in 2013 through 2017.

gauged the classification performance with three metrics: precision, recall, and F1-measure (accuracy) in detecting malware. We took all the apps from each year used in the characterization study as a separate dataset, and trained the detector using samples from one year while testing it against apps from other years. We ensured that no apps trained were ever used in the testing to avoid possible overfitting problems. Part of the results is shown in Figure 2, which depicts the trends of classification performance changes in terms of F1 accuracy with our detector versus *MamaDroid*. As shown, while both classification approaches deteriorate over time, ours did so much slower than the baseline. Not only was the performance of our approach always higher than *MamaDroid* at any of the five time spans studied, overall the gap enlarged continuously with the increase in the length of the time span. These contrasts suggest that our work can achieve high accuracy much longer than the state-of-the-art solution, thus clearly outperforming it in terms of sustainability.

4 CONCLUSION AND FUTURE WORK

This work explores a novel malware detection approach, based on the evolutionary dynamic characteristics of Android apps that model the distribution of their sensitive accesses. Our preliminary results show that the evolution-based malware detection outperformed the state-of-the-art peer approach, in terms of the length of sustaining period for high accuracy. Future work will expand both the evolution study and evaluation of the malware detection approach.

REFERENCES

[1] 2017. Mobile malware growth. https://www.gdatasoftware.com/blog/2017/04/29666-malware-trends-2017. (2017).
[2] Haipeng Cai and Barbara Ryder. 2017. DroidFax: A toolkit for systematic characterization of Android applications. In *Proceedings of International Conference on Software Maintenance and Evolution (ICSME)*. 643–647.
[3] Haipeng Cai and Barbara Ryder. 2017. Understanding Android Application Programming and Security: A Dynamic Study. In *International Conference on Software Maintenance and Evolution (ICSME)*. 364–375.
[4] Enrico Mariconti, Lucky Onwuzurike, Panagiotis Andriotis, Emiliano De Cristofaro, Gordon Ross, and Gianluca Stringhini. 2017. MAMADROID: Detecting Android Malware by Building Markov Chains of Behavioral Models. In *Proceedings of Network and Distributed System Security Symposium*.