Building a Framework for Objective Evaluation of Malware Detection Methods
Bachelor Thesis

Supervisors: Prof. Dr. Alexander Pretschner, Aleieldin Salem
Email: pretschn, salem @ in.tum.de
Phone: +49 89 289 – 17, 314
Starting date: Immediately

Context
The research community has been working towards devising methods to detect Android malware (e.g., [7, 10, 13, 15]). Despite proving to be sometimes inconsistent [4], researchers continue to rely on VirusTotal [16] to either download training data to evaluate their newly-devised methods [14, 17, 19], or to label the apps they manually gathered from the wild (e.g., app marketplaces) [1, 6, 18], due to the lack of better, more feasible alternatives. Unfortunately, there are no standard procedures that instruct researchers on how to utilize or interpret the metadata acquired from VirusTotal. For example, despite proving to change over time [5, 8], some researchers may elect to utilize malware datasets (e.g., Drebin [1]), without downloading updated scan results from VirusTotal. Furthermore, given that the platform provides its users with scan results of around 60 antiviral software, instead of a binary label (i.e., malicious or benign), researchers tend to use their intuition and adopt ad-hoc methods to label the apps in the datasets they train their methods with or, more importantly, release to the research community as benchmarks. For example, based on VirusTotal’s scan results, Li et al. labeled the apps in their Piggybacking dataset as malicious if at least one scanner deemed an app as malicious [6], whilst Wei et al. labeled apps in the AMD dataset as malicious if 50% or more of the total scanners labeled an app as such [18].

The lack of standard evaluation procedures means that researchers adopt different combinations of the aforementioned three dimensions of (1) freshness of scan results (e.g., obtained from VirusTotal), (2) the strategy adopted to label apps, and (3) the dataset used to train or evaluate detection method. Such discrepancies in evaluation settings adopted by researchers might hinder the comparability of different detection approaches. Furthermore, it might incite researchers to dismiss promising detection approaches, because they underperform on a dataset with outdated labels, or because they utilize a different labeling strategy that does not reflect the true nature of the apps in the dataset. More importantly, it might give researchers false sense of confidence in the detection capabilities of their detection methods.

This issue has recently inspired researchers to devise frameworks to enable fair and comprehensive evaluation of detection methods [2, 9, 11]. For example, Pendlebury et al. implemented an evaluation framework, TESSERACT, to demonstrate the impact of spatial and temporal bias on the performance of Android detection methods, and to propose a space-time aware evaluation method to mitigate such biases [2]. However, TESSERACT does not discuss the impact of varying VirusTotal-related dimensions (e.g., labeling strategy), on the performance of detection methods, which proved to have a substantial impact on the composition of datasets and, in turn, the performance of detection methods trained with them [3, 12].
Goal
We attempt to complement the work of Pendlebury et al. [2] by advocating the consideration of VirusTotal-related dimensions upon evaluating the performance of (Android) detection methods. In particular, we refer to (a) the freshness of the scan results obtained from VirusTotal, and (b) the labeling strategy used to discern the malignancy of an app depending on its VirusTotal scan results.

Given access to TESSERACT’s code base, we plan to extend its functionality to accommodate for the aforementioned VirusTotal-related dimensions. After extending the framework, we plan on providing the reader with actionable use cases that demonstrate how the extended framework can be utilized to evaluate machine learning-based detection methods, compare their performance, and perhaps combine them as an ensemble of detection methods that yields better performance than individual methods.

Some of the issues and research questions we attempt to address while developing such a framework are:

- What is the definition of an evaluation dimension? What are the different types of dimensions? How to choose their values?
- How does varying the dimensions mentioned above affect the detection performance of detection methods?
- Why does varying the dimensions of (1) time, (2) labeling scheme, and (3) data set impact the detection performance of an Android repackaged malware detection method?
- How does an overall assessment of a detection method help enhance its performance?
- How can the devised evaluation framework be utilized to assess detection methods?

Work-plan

1. Get acquainted with the structure and code base of TESSERACT.
   a. Reading the TESSERACT paper [2].
   b. Understand the framework’s code base.
   c. Identify method to extend the framework
2. Extend the TESSERACT code base to support VirusTotal-related dimensions
3. Evaluate the extended framework.
   (a) Identify evaluation criteria and design experiments.
   (b) Prepare the evaluation dataset.
4. Document the design, implementation, and evaluation of Praetorian.

Required Skills
We are looking for a motivated student with the following expertise:

- Very good programming skills, particularly in Python.
- Good understanding and familiarity with the Android platform.
- Good understanding of machine learning concepts.
- Self motivation and ability to work independently.

Deliverables
The source-code and design of the implemented technique.

The evaluation dataset used to evaluate Praetorian.

A thesis document in accordance with TUM’s guidelines.

References


