Pervasive and Mobile Computing 18 (2015) 137–151

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Power-aware anomaly detection in smartphones: An analysis of on-platform versus externalized operation

Guillermo Suarez-Tangil *, Juan E. Tapiador, Pedro Peris-Lopez, Sergio Pastrana

COSEC—Computer Security Lab, Department of Computer Science, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911, Leganés, Madrid, Spain

HIGHLIGHTS

- We assess the power-consumption trade-offs among different strategies for off-loading, or not, certain security tasks to the cloud.
- We evaluate different functional tasks in machine learning based detection systems.
- We provide consumption models for such tasks that can be used to obtain power consumption estimates and compare detectors.
- Experiments show that outsourced computation is significantly the best option in terms of power consumption.
- Our findings also point out noticeable differences among different machine learning algorithms.

ARTICLE INFO

Article history:
Received 4 March 2014
Received in revised form 16 July 2014
Accepted 25 October 2014
Available online 4 November 2014

Keywords:
Smartphone security
Anomaly detection
Outsourced security
Power consumption

ABSTRACT

Many security problems in smartphones and other smart devices are approached from an anomaly detection perspective in which the main goal reduces to identifying anomalous activity patterns. Since machine learning algorithms are generally used to build such detectors, one major challenge is adapting these techniques to battery-powered devices. Many recent works simply assume that on-platform detection is prohibitive and suggest using offloaded (i.e., cloud-based) engines. Such a strategy seeks to save battery life by exchanging computation and communication costs, but it still remains unclear whether this is optimal or not in all circumstances. In this paper, we evaluate different strategies for offloading certain functional tasks in machine learning based detection systems. Our experimental results confirm the intuition that outsourced computation is clearly the best option in terms of power consumption, outweighing on-platform strategies in, essentially, all practical scenarios. Our findings also point out noticeable differences among different machine learning algorithms, and we provide separate consumption models for functional blocks (data preprocessing, training, test, and communications) that can be used to obtain power consumption estimates and compare detectors.

1. Introduction

The past few years have witnessed a rapid proliferation of portable "smart" devices with increasingly powerful computing, networking and sensing capabilities. One of the most successful examples of such devices so far is smartphones

* Corresponding author. Tel.: +34 91 624 6260; fax: +34 91 624 9429.
E-mail addresses: guillermo.suarez.tangil@uc3m.es (G. Suarez-Tangil), jestevez@inf.uc3m.es (J.E. Tapiador), pperis@inf.uc3m.es (P. Peris-Lopez), spastran@inf.uc3m.es (S. Pastrana).

http://dx.doi.org/10.1016/j.pmcj.2014.10.007
1574-1192/

and tablets, but new appliances are appearing at a steady pace, including watches, glasses, and other wearable systems. One key difference between such smart devices and traditional, non-smart platforms is that they offer the possibility to easily incorporate third-party applications (“apps” for short) through online markets. The popularity of smartphones has been recurrently corroborated by commercial surveys, showing that they will very soon outsell the number of PCs worldwide [1] and that users are already spending nearly as much time on smartphone applications as on the Web (73% vs. 81%) [2].

In many respects, devices such as smartphones present greater security and privacy risks to users than traditional computing platforms. One key reason is the presence in the device of numerous sensors that could leak highly sensitive information about the user's behavioral patterns (e.g., location, gestures, moves and other physical activities) [3], as well as recording audio, pictures and video from their surroundings. As a consequence, the development of smartphone technologies and its widespread user acceptance have come hand in hand with a similar increase in the number and sophistication of threats tailored to these platforms. For example, recent surveys have warned about the alarming volume of smartphone malware distributed through alternative markets [4] and the spread of new forms of fraud, identity theft, sabotage, and other security threats.

1.1. Anomaly detection in smart devices

Many security issues can be essentially reduced to the problem of separating malicious from non-malicious activities. Such a reformulation has turned out to be valuable for many classic computer security problems, including detecting network intrusions, filtering out spam messages, or identifying fraudulent transactions. But, in general, defining in a precise and computationally useful way what is harmless or what is offensive is often too complex. To overcome these difficulties, many solutions to such problems have traditionally adopted a machine learning approach, notably through the use of classifiers to automatically derive models of good and/or bad behavior that could be later used to identify the occurrence of malicious activities.

Anomaly-based detection strategies have proven particularly suitable for scenarios where the main goal is to separate “self” (i.e., normal, presumably harmless behavior) from “non-self” (i.e., anomalous and, therefore, potentially hostile activities). In this setting, one often uses a dataset of self instances to obtain a model of normal behavior. In detection mode, each sample that does not fit the model is labeled as anomalous. This notion has been thoroughly explored over the last two decades and applied to multiple domains in the security arena [5–7].

More recently, many security problems related to smartphone platforms have been approached with anomaly-based schemes (see, e.g., [8–12]). One illustrative example is found in the field of continuous – or implicit – authentication through behavioral biometrics [13–15]. The key idea here is to equip the device with the capability of continuously authenticating the user by monitoring a number of behavioral features, for instance the gait – measured through the built-in accelerometer and gyroscope –, the keystroke dynamics and the usage patterns of apps. These schemes rely on a model learned from user behaviors to identify anomalies that, for example, could mean that the device is mislaid, in which case it should lock itself and request a password.

Proposals for detecting malware in smartphones have also made extensive use of anomaly detection approaches. Most schemes are built upon the hypothesis that malicious apps somehow behave differently from goodware. The common practice consists of monitoring a number of features for non-malicious apps, such as the amount of CPU used, network traffic generated, system/API calls made and permissions requested. These traces are then used to train models of normality that, again, can be used to spot suspicious behavior. Modeling app behavior in this way is particularly useful in two scenarios. The first one is related to the problem of repackaged apps, which constitute one of the most common distribution strategies for smartphone malware. In this case, the malicious payload is piggybacked into a popular app and distributed through alternative markets. Detecting repackaged apps is a challenging problem, in particular when the payload is obfuscated or dynamically retrieved at runtime. The second problem is thwarting the so-called grayware, i.e., apps that are not fully malicious but that entail security and/or privacy risks of which the user may not be fully aware. For instance, an increasingly number of apps access user-sensitive information such as locations frequently visited and contacts, and send it out of the phone for obscure purposes [3]. As users find it difficult to define their privacy preferences in a precise way, automatic methods to tell apart good from bad activities constitute a promising approach.

1.2. Motivation

Essentially all machine learning-based anomaly detection solutions can be broken down into the following functional blocks:

- *Data acquisition*. Activity traces are required both for (re-)training the model of normality and in detection mode. The nature of the data collected varies across applications and may include events such as system calls, network activities and user-generated inputs.
- *Feature extraction*. Machine learning algorithms require data to be expressed in particular formats, commonly in the form of feature vectors. A number of features are extracted from the acquired activity traces during a preprocessing stage. The complexity of such preprocessing depends on the problem and ranges from computationally straightforward procedures (e.g., obtaining simple statistics from the data) to more resource intensive transformations.