J. Parallel Distrib. Comput. 103 (2017) 22–31

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

A hybrid approach of mobile malware detection in Android

Fei Tong [a], Zheng Yan [a,b,*]

[a] State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, China
[b] Department of Communications and Networking, Aalto University, Espoo, Finland

HIGHLIGHTS

- Hybrid mobile malware detection based on both malware and normal patterns.
- Implementation and performance test based on an Android mobile platform.
- Self-improvement based on automatic optimization of pattern sets.
- Detection accuracy and generality showed through comparison.

ARTICLE INFO

Article history:
Received 22 May 2016
Received in revised form
2 August 2016
Accepted 13 October 2016
Available online 21 October 2016

Keywords:
Android
Malware detection
Pattern match
System call

ABSTRACT

Android security incidents occurred frequently in recent years. This motivates us to study mobile app security, especially in Android open mobile operating system. In this paper, we propose a novel hybrid approach for mobile malware detection by adopting both dynamic analysis and static analysis. We collect execution data of sample malware and benign apps using a net_link technology to generate patterns of system calls related to file and network access. Furthermore, we build up a malicious pattern set and a normal pattern set by comparing the patterns of malware and benign apps with each other. For detecting an unknown app, we use a dynamic method to collect its system calling data. We then compare them with both the malicious and normal pattern sets offline in order to judge the unknown app. Based on the test on a set of mobile malware and benign apps, we found that our approach achieves better detection success rate than some methods using either static analysis or dynamic analysis. What is more, the proposed approach is generic, which can detect different types of malware effectively. Its detection accuracy can be further improved since the pattern sets can be automatically optimized through self-learning.

1. Introduction

Mobile devices have become an open platform for executing various applications. Worldwide mobile app downloads are expected to reach 224,801 billion in 2016 and will continuously increase [21]. Due to the rapid growth of the smart phone industry and the rapid promotion of mobile communication technologies, more and more consumers use smart phones to access the Internet and consume various services. Mobile apps bring great convenience for our daily life by providing instant access to the wealth of information over the Internet, supporting constant communications anywhere and offering various functionalities. The fast growth of mobile apps plays a crucial role for the success of future mobile Internet and economy.

The security of mobile apps has to be paid special attention. The smart phones normally store private user data such as pictures, messages, and personal credentials. Thus, they become the target of many malicious attackers [30,15]. In the smart phone industry, devices with Android operating system hold a leading position. However, around 97% of mobile malware target the Android phones. In recent years, security incidents of Android mobile phones occurred frequently, some serious attacks happened also at Apple phones. This situation motivates us to study mobile app security, especially in Android.

Current research of malware detection can be classified into two methods: static analysis and dynamic analysis [15]. The static analysis aims at finding malicious characteristics or bad code segments of an app without executing them, while dynamic analysis focuses on collecting apps' behavior data in order to find

* Corresponding author at: State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, China.
E-mail addresses: tongfeisecurity@163.com (F. Tong), zyan@xidian.edu.cn, zheng.yan@aalto.fi (Z. Yan).

http://dx.doi.org/10.1016/j.jpdc.2016.10.012
0743-7315/

malware during apps runtime. Static analysis can find already known malware with high accuracy and efficiency. But it is helpless facing with camouflage techniques and encryption algorithms and cannot find intrusions suddenly happening at runtime. It cannot exhaust all malicious features to achieve comprehensive detection. On the contrary, the dynamic analysis can distinguish zero-day attacks. But this kind of methods often consumes huge operating resources with low efficiency and detection accuracy. At present, there are no good solutions to detect mobile malware by making use of the advantages of both methods, in order to effectively find runtime problems and identify malware and benign apps in a generic way through a uniform detection process. In the literature, mobile malware research is still in its infancy, even though malware authors shift their focus to smart phones. Few of the existing solutions can effectively detect mobile malware in a generic way with high accuracy. Some malicious mobile apps could intrude the mobile device suddenly after being used for a while. This threat challenges the research of mobile app security.

In this paper, we propose a novel approach to detect malware in a hybrid and generic way, especially for mobile malware in Android devices. We collect execution data of a set of known sample malware and benign apps to generate patterns of individual system calls and sequential system calls with different calling depth that are related to file and network access, and so on. By comparing the patterns (reflected by the above individual and sequential system calls) of malware and benign apps with each other, we build up a malicious pattern set and a normal pattern set that are used for malware detection and benign app judgment. When we need to detect an unknown app, we use a dynamic method to collect its runtime system calling data in terms of both individual calls and sequential system calls with different depth (e.g., about file and network access). Then we extract the target patterns (i.e., the frequency of sequential system calls with different calling depth) of the unknown app from its runtime system calling data. By comparing them with both the malicious pattern set and the normal pattern set, we judge the unknown app's good or bad. The proposed approach is a generic detection method suitable for various types of malware detection since the pattern set contains the patterns of various kinds of malware and benign apps. The malicious pattern set and the normal pattern set can be further optimized and extended based on the patterns of newly confirmed malware and benign apps. Specifically, the contribution of this paper can be summarized as below:

(1) We motivate applying both static and dynamic detection methods for mobile malware detection based on the generation and comparison with both malware patterns and normal patterns.
(2) We successfully implement the proposed approach based on an Android mobile platform and test its performance.
(3) We evaluate the performance of the proposed approach with regard to detection accuracy and generality, prove its self-improvement based on self-learning and automatic pattern optimization and show its better detection rate and sound effectiveness by comparing it with existing methods.

The rest of the paper is organized as follows. Section 2 briefly overviews related work. Section 3 introduces the hybrid approach for mobile malware detection by describing the algorithm for generating malware and normal pattern sets and the algorithm for malware detection. The pattern generation of the proposed approach is described in Section 4 followed by detection performance evaluation in Section 5. Finally, conclusion is presented in the last section.

2. Related work

Many novel techniques for detecting malware have been proposed in the literature. Most of them were introduced in comprehensive surveys [6,22]. However, mobile malware research is still in its infancy. The techniques available for detecting mobile malware and other security vulnerabilities have varying strengths and weaknesses.

Current mainstream Android malware detection methods fall into two categories: static analysis and dynamic analysis [22]. The static analysis uses disassemble technologies [29,13] to decompile Android app source codes in order to figure out malicious signature codes [16]. The dynamic analysis collects Android app runtime data to find out whether the app executes with malicious behaviors [27,2].

2.1. Static analysis

The static analysis is the way to find malicious characteristics or bad code segments in an app without executing it [1,9,10]. It is generally used in preliminary malware detection, when suspicious applications are first evaluated in order to find out any obvious security threats.

Some existing static analysis methods focus on classifying and detecting different types of malware. Li et al. proposed a static analysis method based on a characteristic tree [13]. Their research aimed to find and implement a novel API-usage characterization approach for Android on different layers of resolutions, namely packages, classes, functions and APIs. A tree structure called "Characteristic Tree" was used to contain such API-usage information on different layers of the tree structure. A comparison algorithm was also designed for calculating characteristic-tree similarity. This new detection method provides meticulous insights in classifying and detecting different types of Android malware in different code families. Yerima et al. proposed a proactive machine learning approach based on Bayesian classification in order to discover unknown Android malware via static analysis [27]. Sayfullina et al. presented a static algorithm for Android malware classification based on the features extracted from Android application package files, including AndroidManifest.xml, classes.dex and resources.arsc [18]. Wang et al. explored the permission-induced risk in Android apps on three levels in a systematic manner in order to detect mobile malware [24]. The proposed process includes thorough analysis on the risk of an individual permission and a group of collaborative permissions; evaluation on the usefulness of risky permissions for malware detection with support vector machine, decision trees, as well as random forest; in depth analysis on the detection results and their feasibility based on permission requests.

Other static analyses disassemble app codes to perform malware detection. Schmidt et al. developed a static malware detection technique that firstly disassembles the mobile application and extracts system calls (feature extraction) and then uses Centroid Machine, a lightweight clustering mechanism, to classify the mobile application as either malicious or benign (anomaly detection) [19]. Another static detection performs static taint analysis by disassembling mobile applications and constructing a control flow graph (CFG) [5]. The analysis considers the paths originating from sensitive sources, such as an address book, current GPS coordinates, keyboard cache, unique device ID, and other phone-related information. A dataflow analyzer checks any sensitive data transmitted from the sources to a remote server without notifying a user and thus causing privacy leakages. This method can only detect privacy leaks within a single application. It fails if two or more apps are transitively chained together. Enck et al. used a decompiler (for