COMPUTERS & SECURITY 68 (2017) 36–46

Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

PIndroid: A novel Android malware detection system using ensemble learning methods

Fauzia Idrees [a,*], *Muttukrishnan Rajarajan* [a], *Mauro Conti* [b], *Thomas M. Chen* [a], *Yogachandran Rahulamathavan* [c]

[a] *School of Mathematics & Engineering, City University London, London EC1V 0HB, UK*
[b] *Department of Mathematics, University of Padua, 35122 Padova, Italy*
[c] *Institute for Digital Technologies, Loughborough University in London, London, UK*

ARTICLE INFO

Article history:
Received 28 September 2015
Received in revised form 21 March 2017
Accepted 25 March 2017
Available online 31 March 2017

Keywords:
Malware classification
Permissions
Intents
Ensemble methods
Colluding applications

ABSTRACT

The extensive use of smartphones has been a major driving force behind a drastic increase of malware attacks. Covert techniques used by the malware make them hard to detect with signature based methods. In this paper, we present PIndroid – a novel Permissions and Intents based framework for identifying Android malware apps. To the best of our knowledge, PIndroid is the first solution that uses a combination of permissions and intents supplemented with Ensemble methods for accurate malware detection. The proposed approach, when applied to 1,745 real world applications, provides 99.8% accuracy (which is best reported to date). Empirical results suggest that the proposed framework is effective in detection of malware apps.

1. Introduction

In past few years, smartphones have transformed from simple mobile phones into mobile computers, making them suitable for personal and business activities. Smartphones have become the major target for mobile malware due to increased reliance on them for daily activities such as storing private data, financial transactions, emailing, socializing and online shopping.

Android being the most widely used platform for smartphones is under constant attacks. Existing anti-virus solutions are not capable of eliminating the exponentially increasing malware threats due to their reliance on signature-based detection. Moreover, resource constrained smartphones are unsuited for continuous malware scanning. There is a need to have an efficient method capable of overcoming the current challenges of outdated signatures, code obfuscation and resource constraints.

Permissions are used to guard against misuse of system resources and user data, however, some of Android's features like *intents* can break this shield. A lot of research has been done on permissions; however, intent is an under investigated area (in malware detection), providing opportunity for the evolving malware threats.

We propose a malware detection approach which classifies apps against certain combinations of permissions and intents which are unique to malware apps. These combinations form an efficient detection pattern to differentiate between malware and benign apps with a granularity to classify malware families. We evaluate the efficacy of proposed approach by

* *Corresponding author.*
E-mail address: fauzia.idrees.1@city.ac.uk (F. Idrees).
http://dx.doi.org/10.1016/j.cose.2017.03.011

applying machine learning algorithms. A comparative study of classifiers is carried out against different performance measures to select the most accurate and efficient classifier. We apply the ensemble methods to optimize the results.

1.1. Contributions

The main contributions presented in this paper are:

1. To the best of our knowledge, this is the first work that combines intents and permissions for collaborative malware detection. This work combines *permissions* and *intents* of applications to generate a distinguishing matrix that is used for efficient and accurate detection of malware and its associated families. Our method is capable of achieving 99% detection accuracy by combining permissions and intents.
2. We propose a new approach using ensemble methods to optimize the classification results. Our results show a detection accuracy of 99.8% by connecting multiple classifiers laterally with a meta-classifier.
3. We apply statistical significance test to investigate the correlation between permissions and intents. We found statistical evidence of a strong correlation between permissions and intents which could be exploited to detect malware applications.

1.2. Organization

Section 2 discusses the related work; Section 3 provides an overview of Android permissions and intents and Section 4 discusses about the analysis carried out on permissions and intents. Section 5 presents the proposed framework; Section 6 describes the model evaluation, experimental settings, and results. Section 7 concludes the paper.

2. Related work

There is a plethora of research work on Android security covering vulnerability assessments, malware analysis and detection. Faruki et al. (2014) present an overview on the current malware trends. Malware analysis leverage static, dynamic and hybrid methods. In static malware analysis, properties of apps are extracted by analysing different static features without running the code. In dynamic analysis, the runtime profiles of apps are generated by monitoring and collecting the memory consumption, CPU usage, battery usage and network traffic statistics (Shabtai et al., 2012) and (Burguera et al., 2011). Here, we provide an overview of related works in this area.

2.1. Static malware analysis on Android platform

Different static features such as permissions, API calls, Interprocess communication (IPC), code semantics, intents, hardware, components and developer ID have been used for malware detection. However, permissions, API calls, and IPC have attracted more attention from the researchers. There are a few works in which different features have been combined for malware detection. Here, we discuss the relevant works which use permissions, ICC/intents or hybrid features.

2.1.1. *Permission analysis*

Permission is the most investigated feature in malware detection. Barrera et al. (2010) examined 1,100 apps for permission usage and found the high frequency of certain permissions. Peng et al. (2012) calculated the risk scores of apps by analysing the requested permissions. Enck et al. (2009c) identified dangerous combinations of permissions and developed the security rules to identify malicious apps. Vidas and Christin (2014) identified the unnecessary permission requests by the apps. Zhang et al. (2013) examined mapping between API calls and permissions for behaviour profiling. Sarma et al. (2012) calculated risks and benefits of requested permissions to discern the adverse affects of app. Felt et al. (2011) is a tool to check the overprivilege of apps by mapping requested permissions with APIs. Au et al. (2012) is another tool which extracts permissions from the source code and maps them with URIs. Most of these methods aim to provide help to app developers and security analysts. These methods may be used as add-ons with malware detection solutions.

2.1.2. *Inter-component communication/intents analysis*

ICC and intents have not been explored the way permissions have been investigated. Most of the existing ICC based studies focus on finding the ICC related vulnerabilities. Enck et al. (2009a, 2009b) investigated the IPC framework and interaction of system components. Chin et al. (2011) detects the ICC related vulnerabilities. Kantola et al. (2012) suggested improvement in ComDroid by segregating the communication messages into inter and intra-applications groups so that the risk of inter-application attacks may be reduced. Maji et al. (2012) characterized Android components and their interaction. They investigated risks associated with misconfigured intents. Lu et al. (2012) examined vulnerable public component interfaces of apps. Avancini and Ceccato (2013) generated test scenarios to demonstrate the ICC vulnerabilities. Gordon et al. (2015) performs information flow analysis to investigate the communication exploits. Gallingani et al. (2015) investigated intents related vulnerabilities and demonstrated how they may be exploited to insert the malicious data. Their experiments found 29 out of a total of 64 investigated apps as vulnerable to intent related attacks. All of these works focus on finding communication vulnerabilities, and none of them used ICC and intents for malware detection.

2.1.3. *Malware analysis with hybrid features*

In this category, different features are combined for effective malware detection. Most relevant works are: Arp et al. (2014), Wu et al. (2012) and Lindorfer et al. (2015) as they use permissions and intents in addition to other features for malware classification. Arp et al. (2014) examines the manifest file and code of apps to check the permissions, API calls, hardware resources, app components, filtered intents and network addresses. It uses Support Vector Machines (SVM) for malware classification. Wu et al. (2012) analyses features extracted from the manifest and smali files of disassembled codes. These features include permissions, components, intent messages and API calls. It applies K-means algorithm and Singular Value