Chapter 9
A Novel Security Framework for Managing Android Permissions Using Blockchain Technology

Abdellah Ouaguid
Hassan II University, Morocco

Noreddine Abghour
Hassan II University, Morocco

Mohammed Ouzzif
Hassan II University, Morocco

ABSTRACT

This article presents a new framework named ANDROSCANREG (Android Permissions Scan Registry) that allows to extract and analyze the requested permissions in an Android application via a decentralized and distributed system. This framework is based on the emerging technology Blockchain whose potential is approved in the matter of transparency, reliability, security and availability without resorting to a central processing unit judged of trust. ANDROSCANREG consists of two Blockchains, the first one (PERMBC) will handle analysis, validation and preparation of the raw results so that they will persist in the second Blockchain of Bitcoin already existing (BTCBC), which will assume the role of a Registry of recovered permissions and will save the permissions history of each version of the applications being scanned via financial transactions, whose wallet source, recipient wallet and transaction value have a precise meaning. An example of a simulation will be presented to describe the different steps, actors, interactions and messages generated by the different entity of ANDROSCANREG.

INTRODUCTION

The android ecosystem continues its world domination through operating systems and takes pole position with 86,8% in market share in 2016Q3 (*IDC: Smartphone OS Market Share*, 2016) by profiting from a light increase of 1,1% of the world market of Smartphones.

DOI: 10.4018/978-1-5225-6201-6.ch009

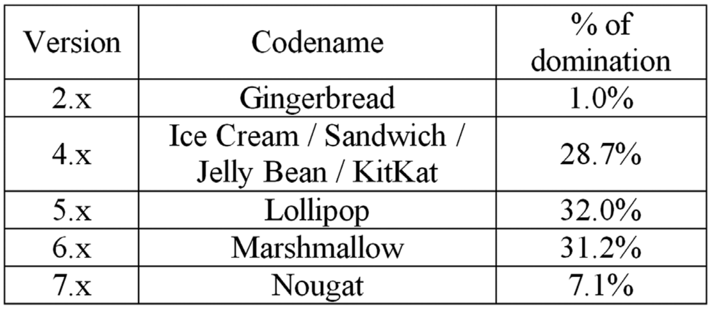
This position of quasi monopoly is due to its ‘Open Source’ nature that encourages telephone constructors to adapt it to the large scale and also to the large number of developed applications (+2,7 millions applications) (*Number of Android applications*, 2016). These are made accessible through Google’s official store (Google Play) or Third-Party stores such as Amazon, AppShop, Baidu App Store, Opera Mobile App Store...etc. Android’s popularity has made it the preferred target for hackers (Symantec, 2016) that take advantage of the uncorrected vulnerability (*Android, système d’exploitation le plus vulnérable*, 2017) of the Operating System in order to launch refined attacks through malwares.

These are designed specifically to take control over the targeted device and access the sensitive data of the users (Feizollah, Anuar, Salleh, & Wahab, 2015). Recently, a malware targeting clients of large banks was detected, and it was thought to be a Flash Player. The great danger of this malware resides in its capacity to steal authentication of 94 different applications of mobile banking (*Android banking malware masquerades as Flash Player, targeting large banks and popular social media apps*, 2016).

Limiting the field of action of applications is a solution, among many more, that target reducing the improper use of the users’ sensitive data. This is what Google tried to apply by implementing a control mechanism of permissions that is inspired by a Linux security model. However, this mechanism showed its weakness (Fang, Han, & Li, 2014), especially when the applications’ developers demanded unnecessary permissions that are never used in their applications (over privilege) (Felt, Chin, Hanna, Song, & Wagner, 2011). This can lead to discreetly transforming a legitimate application to malware through a manipulation of authorization with the objective geared towards accessing users’ sensitive data (Geneiatakis, Fovino, Kounelis, & Stirparo, 2015). Since the launching of 6.0 version of Android, the permissions system management has clearly improved by giving the user the right to manage the permissions of the installed application. Yet, this is considered insufficient since: 1) the users underestimate the impact of giving permission about their private life to another source, 2) the majority of users of Android (61,7%) always work through an earlier version of 6.X (Table 1) and 3) wherein the multitude of permissions are accompanied by an incomplete documentation (Felt et al., 2011) of how to use them reasonably. This requires having an autonomous, reliable and trusted entity that analyzes the permissions of each application before the installation to define the level of legitimacy of the permissions requested (Neisse, Steri, Geneiatakis, & Fovino, 2016).

The studies (Fang et al., 2014) conducted on the static analysis of permissions favor the centralized approach of analysis; which means either 1) submitting a verification request to a distant analysis platform

Table 1. Division of Android versions



Version	Codename	% of domination
2.x	Gingerbread	1.0%
4.x	Ice Cream / Sandwich / Jelly Bean / KitKat	28.7%
5.x	Lollipop	32.0%
6.x	Marshmallow	31.2%
7.x	Nougat	7.1%

Source: Dashboards | Android Developers,2017