Arab J Sci Eng (2016) 41:479–493
DOI 10.1007/s13369-015-1665-2

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Preventive Policy Enforcement with Minimum User Intervention Against SMS Malware in Android Devices

Abdelouahid Derhab[1] · **Kashif Saleem**[1] · **Ahmed Youssef**[2] · **Mohamed Guerroumi**[3]

Received: 9 December 2014 / Accepted: 13 April 2015 / Published online: 3 May 2015

Abstract In this paper, we propose MinDroid, a user-centric preventive policy enforcement system against SMS malware in Android devices. The design of MinDroid takes into consideration the user's little understanding of the Android permission system. This can be done by deriving the policy rules from the behavioral model of the malicious SMS applications rather than adopting user-defined rules. MinDroid requires user intervention only during the first T time units from the application installation time. The user during this time period is notified to accept or reject the SMS-sending operations. MinDroid execution is specified as a finite state machine, and its security properties are formally proven using Metric Temporal Logic. We also show that MinDroid is resilient against threats trying to compromise its correct functionality. In addition, an analytical study demonstrates that MinDroid offers good performance in terms of detection time and execution cost in comparison with intrusion detection systems based on static and dynamic analysis. The detection efficiency of MinDroid is also studied in terms of detection rate, false positive rate, and ROC distance. A prototype implementation of MinDroid is tested under Android emulator.

Keywords SMS malware · Policy enforcement · Prevention · Android

✉ Ahmed Youssef
ahyoussef@ksu.edu.sa

Abdelouahid Derhab
abderhab@ksu.edu.sa

Kashif Saleem
ksaleem@ksu.edu.sa

Mohamed Guerroumi
mguerroumi@usthb.dz

[1] Center of Excellence in Information Assurance (COEIA), King Saud University, Riyadh, Kingdom of Saudi Arabia

[2] College of Computer and Information Sciences (CCIS), King Saud University, Riyadh, Kingdom of Saudi Arabia

[3] Faculty of Electronic and Computer Science, USTHB University, Algiers, Algeria

1 Introduction

The current era comes up with an enormous growth in the field of wireless communication with respect to solutions, devices, and their software [1]. The mobile devices offer many applications that become very essential for people to manage daily life [2]. Almost all the personal and confidential information nowadays is stored and transferred through these mobile devices [3]. Therefore, security becomes essential and is required to be employed at every layer [4–8] to deal with specific attacks. In mobile devices, the operating systems such as Android and iOS play an important role in managing the resources and providing services to applications [9].

The Android operating system is popular among users and dominant in the mobile world. According to a report from Strategy Analytics [10], Android had 85 % of the global mobile market share in the second quarter of 2014. This popularity makes Android attractive for malware developers, which aim at making monetary profit by infecting the maximum number of devices. A report study [11] has shown that 98.05 % of mobile threats targeted the Android operating system in 2013. In the third quarter of 2013, F-secure reported that 81.1 % of the mobile threats were profit-motivated and 97.14 % of the latter targeted Android OS [12]. Most of the profit-motivated malwares perform SMS-sending operations to perform their attacks. A recent report published in 2014 [13] revealed that 83 % of the Android malwares analyzed

by F-secure were performing SMS-sending activity, making it the most common activity among malware.

The attacks performed by the SMS-sending trojans (or SMS malware) are the following:

- *SMS spam* This is the most basic form of attack where unsolicited messages are sent to subscribers for commercial advertising [14]. Sending spam from a compromised mobile device reduces the risk of exposing the spammer and keeps the spammer's identity hidden. Users might find out about the existence of malware in their devices sending out SMS spam when their monthly phone bills arrive. The spams might also contain links to malicious applications.
- *SMS premium rate fraud* Unsolicited messages that trick subscribers to call premium rate numbers or subscribe for services that are charged to the bill of the victim. These SMSs result in costly sums being transferred from the user's account to that of the cybercriminals. We have seen a continuous growth of SMS-sending trojan families performing this attack such as FakeInst, OpFake, PremiumSms, and SmsSend. For instance, a malware named Trojan-SMS.AndroidOS.FakeInst.ef, masquerades as an application for watching porn videos. Once it is installed, it starts sending SMS messages to predefined premium rate numbers. The TOP 10 Android malware families reported in [13] showed that FakeInst family accounted for 45 % of threats followed by SmsSend variants for 34 %.
- *SMS privacy attack* Some malware such as Android/SmsHnd.A, AndroidOS.Bankrypt.BH, and Android.Nickispy can read sensitive information stored at the device such as bank account, phone identifiers (IMES, IMSI), and GPS location and send it to the attacker.
- *SMS flooding* It takes place when unsolicited SMSs are sent to a user, which can lead to a denial of service (DoS) in both the core network and the radio access networks [15].

In Android, applications can access resources such as telephony, network, and SMS functions using APIs. The latter are protected through a security mechanism called permissions. Each application must define the permissions it requests in its AndroidManifest.xml file. A user needs to grant all the requested permissions to install the application. Otherwise, the application cannot be installed. In Android 4.2 (Jelly Bean), the user is notified when an application attempts to send SMS to a premium rate number. However, a malware can hide this notification from the user. Beginning with Android 4.4 (KitKat), only one default SMS application, selected by the user, is allowed to write to the SMS provider and receives SMSs. The other SMS applications can only read the SMS provider and are notified when an SMS is received. However, this design choice cannot prevent the SMS attacks. According to [13], 0.1 % of the applications received from Google's Play Store were identified as malicious. Thus, a user might download a compromised SMS application that sends both legitimate and malicious SMSs.

Most of the users ignore or have little understanding of the Android permission policy. This was confirmed in a survey [16], which showed that only 17 % of users look at the permissions when installing applications. It has also been noticed that developers request more permissions than they actually need [17]. As legitimate and malicious applications can request the same permissions, it is often difficult for the users to determine during the installation time whether the requested permissions are harmful or not.

In order to help the user to accurately distinguish between legitimate and malicious SMS applications, we propose MinDroid, a new user-centric policy enforcement approach for Android devices, which notifies the user of all SMS-sending operations performed by an application within the time interval $[a, a + T]$, where a is the installation time of the application and T is the duration of the testing phase. Then, the user either accepts or rejects the operation. Knowing that a malicious application starts sending SMS within a short time $\theta \leq T$ after its installation on the device, it is sufficient that the user intervenes only during the first T time units of the application lifespan, in order to prevent all malicious SMS-sending operations. The main contributions of the paper are the following: Firstly, we design MinDroid by considering two types of SMS threats occurring during the testing phase: (a) SMSs are generated by a malicious application without the user's knowledge and (b) a compromised SMS application that sends both legitimate and malicious SMSs. To the best of our knowledge, this second threat has not been addressed before. Secondly, we extend the attack models to consider attacks performed by a malicious application for the first time after the expiration of the testing phase. Thirdly, we prove using Metric Temporal Logic (MTL) that any malicious SMS application that is installed with or without the user's knowledge is eventually blocked. Fourthly, we verify the security properties of MinDroid and its resilience against attacks, as well as its performance in terms of detection time, execution cost, and detection efficiency. Fifthly, we provide a proof-of-concept implementation of MinDroid under Android Emulator.

The rest of the paper is organized as follows. Section 2 provides related work. In Sect. 3, we present system model and basic idea. The description of MinDroid is given in Sect. 4. Security analysis, resilience analysis, complexity analysis, and detection efficiency analysis are provided in Sects. 5, 6, 7, and 8, respectively. In Sect. 9, we present proof-of-concept implementation. Finally, Sect. 10 concludes the paper.