AppGuard – Enforcing User Requirements on Android Apps

Michael Backes\textsuperscript{1,2}, Sebastian Gerling\textsuperscript{1}, Christian Hammer\textsuperscript{1}, Matteo Maffei\textsuperscript{1}, and Philipp von Styp-Rekowsky\textsuperscript{1}

\textsuperscript{1} Saarland University, Saarbrücken, Germany
\textsuperscript{2} Max Planck Institute for Software Systems (MPI-SWS)

Abstract. The success of Android phones makes them a prominent target for malicious software, in particular since the Android permission system turned out to be inadequate to protect the user against security and privacy threats. This work presents AppGuard, a powerful and flexible system for the enforcement of user-customizable security policies on untrusted Android applications. AppGuard does not require any changes to a smartphone’s firmware or root access. Our system offers complete mediation of security-relevant methods based on callee-site inline reference monitoring. We demonstrate the general applicability of AppGuard by several case studies, e.g., removing permissions from overly curious apps as well as defending against several recent real-world attacks on Android phones. Our technique exhibits very little space and runtime overhead. AppGuard is publicly available, has been invited to the Samsung Apps market, and has had more than 500,000 downloads so far.

1 Introduction

Mobile devices nowadays store a plethora of sensitive information about us – both private and business-related. Usually, this information can be accessed in predefined locations, such as address books or photo folders, and is thus easily locatable by an attacker. Most of these locations, however, lack comprehensive access control and protection mechanisms. When users install a new app on Android, they have no choice but to grant an app all requested permissions at install time, and these permissions cannot be revoked later on. At the same time, these permissions are coarse-grained and their impact is hard to understand for the average user. In the past, several incidents have been reported where private information was deliberately leaked to external servers. Even widely used major apps like Twitter and WhatsApp used to clandestinely send the phone’s whole address book to their servers to mine for possible contacts (for iOS, similar behavior was revealed, e.g., for the Facebook app).

In order to overcome this unsatisfactory situation, this paper presents AppGuard, a tool based on inline reference monitoring (IRM)\textsuperscript{413} that allows the user to enforce fine-grained security and privacy policies on third-party apps. These policies enforced by AppGuard restrict the outreach of vulnerabilities both in third-party applications and the operating system. In short, the IRM
algorithm proceeds in two steps. First, app binaries are rewritten to invoke at runtime a security monitor before each security-relevant program operation, usually before each function call to the Android system libraries. Second, the security monitor dynamically checks whether any of the currently enforced security policies allows the attempted operation, and then either grants the execution or executes alternative code (e.g., to return a mock value to prevent the app’s termination due to an exception). Since IRM only affects the app binary and not the operating system, AppGuard allows for enforcing policies without rooting phones or changing the operating system.

AppGuard is deployed as a stand-alone app, has been installed on about 500,000 phones so far, and will be soon released to the Samsung Apps market after an explicit invitation from Samsung. The experimental evaluation and case studies discussed in this paper demonstrate the effectiveness of our approach: AppGuard exhibits very little overhead in terms of space and runtime, and it can be used to revoke permissions of excessively curious apps, to enforce complex policies, and to prevent several recent real-world attacks on Android phones.

Although several approaches for enforcing policies in Android based on IRM have recently been presented in the literature [7,2], AppGuard is the only IRM based security tool that has been deployed on a large scale and provides a fully automated on-the-phone instrumentation for third-party apps. In the remainder of this paper, we focus on the architecture and on usability and deployment aspects of the tool: for more details on the IRM algorithm and an extensive discussion of the related work, we refer to [1].

2 AppGuard

Architecture. AppGuard uses caller-site rewriting to inline the reference monitor into existing third-party apps. Fig. 1 provides an overview of its components.

Policies. AppGuard provides a set of built-in security and privacy policies. The tool, in particular, provides general purpose policies that aim at the revocation and restriction of critical Android permissions, such as the Internet-, Contacts- and SendSMS-permission. The Internet policy, for example, provides, besides a general on/off switch option, the possibility to specify a set of servers an app is allowed to connect to. The current version of AppGuard contains 24 different policies in total. Security policies are specified in an aspect-oriented programming style and include a detailed specification of all function calls that are to be controlled by the security monitor (cf. Section 3).

Rewriter. Policies constitute the working basis for the rewriting component to inline the specified checks in front of function calls. The rewriter takes an existing application package (.apk file), extracts the classes.dex file, and disassembles it. After analyzing the converted assembly code, the rewriter merges the security checks specified by the policy into the existing application code. Finally, it reassembles the classes.dex file and repackages the apk file. Our implementation handles both reflective JAVA calls and virtual methods.