1. Borrowing your enemy's arrows: the case of code reuse in Android via direct inter-app code invocation

Jun Gao (1); Li Li (2); Pingfan Kong (1); Bissyandé, T.F. (1); Klein, J. (1)

Source: *ESEC/FSE 2020: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, p 939-51, 8 Nov. 2020; **ISBN-13:** 978-1-4503-7043-1; **DOI:** 10.1145/3368089.3409745; **Conference:** ESEC/FSE '20: 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 8-13 Nov. 2020, Virtual Event, USA; **Sponsor:** SIGSOFT; **Publisher:** ACM, New York, NY, USA

Author affiliation: (1) University of Luxembourg, Luxembourg (2) Monash University, Melbourne, VIC, Australia

Abstract: The Android ecosystem offers different facilities to enable communication among app components and across apps to ensure that rich services can be composed through functionality reuse. At the heart of this system is the Inter-component communication (ICC) scheme, which has been largely studied in the literature. Less known in the community is another powerful mechanism that allows for direct inter-app code invocation which opens up for different reuse scenarios, both legitimate or malicious. This paper exposes the general workflow for this mechanism, which beyond ICCs, enables app developers to access and invoke functionalities (either entire Java classes, methods or object fields) implemented in other apps using official Android APIs. We experimentally showcase how this reuse mechanism can be leveraged to “plagiarize" supposedly-protected functionalities. Typically, we were able to leverage this mechanism to bypass security guards that a popular video broadcaster has placed for preventing access to its video database from outside its provided app. We further contribute with a static analysis toolkit, named DICIDer, for detecting direct inter-app code invocations in apps. An empirical analysis of the usage prevalence of this reuse mechanism is then conducted. Finally, we discuss the usage contexts as well as the implications of this studied reuse mechanism. (0 refs)

Inspec controlled terms: Android (operating system) - application program interfaces - Java - mobile computing - program diagnostics - security of data - software engineering
Uncontrolled terms: studied reuse mechanism - code reuse - direct inter-app code invocation - app components - functionality reuse - Inter-component communication scheme - different reuse scenarios - app developers
Classification Code: C6190V Mobile, ubiquitous and pervasive computing - C6110B Software engineering techniques - C6130S Data security - C6150G Diagnostic, testing, debugging and evaluating systems - C6150J Operating systems
IPC Code: G06F9/44 - G06F9/46 - G06F11/36 - G06F21/00
Treatment: Practical (PRA)
Database: Inspec
Data Provider: Engineering Village

Copyright 2020, The Institution of Engineering and Technology