1. Revisiting the impact of common libraries for android-related investigations

Li Li (1); Riom, T. (1); Bissyandeacute, T.F. (1); Wang, H. (2); Klein, J. (1); Yves, L.T. (1)
Source: *Journal of Systems and Software*, v 154, p 157-75, Aug. 2019; **ISSN:** 0164-1212; **DOI:** 10.1016/j.jss.2019.04.065; **Publisher:** Elsevier B.V., Netherlands
Author affiliation: (1) University of Luxembourg, Interdisciplinary Centre for Security, Reliability and Trust, Luxembourg (2) Beijing University of Posts and Telecommunications, School of Computer Science, China
Abstract: The packaging model of Android apps requires the entire code to be shipped into a single APK file in order to be installed and executed on a device. This model introduces noises to Android app analyses, e.g., detection of repackaged applications, malware classification, as not only the core developer code but also the other assistant code will be visited. Such assistant code is often contributed by common libraries that are used pervasively by all apps. Despite much effort has been put in our community to investigate Android libraries, the momentum of Android research has not yet produced a complete and reliable set of common libraries for supporting thorough analyses of Android apps. In this work, we hence leverage a dataset of about 1.5 million apps from Google Play to identify potential common libraries, including advertisement libraries, and their abstract representations. With several steps of refinements, we finally collect 1113 libraries supporting common functions and 240 libraries for advertisement. For each library, we also collected its various abstract representations that could be leveraged to find new usages, including obfuscated cases. Based on these datasets, we further empirically revisit three popular Android app analyses, namely (1) repackaged app detection, (2) machine learning-based malware detection, and (3) static code analysis, aiming at measuring the impact of common libraries on their analysing performance. Our experimental results demonstrate that common library can indeed impact the performance of Android app analysis approaches. Indeed, common libraries can introduce both false positive and false negative results to repackaged app detection approaches. The existence of common libraries in Android apps may also impact the performance of machine learning-based classifications as well as that of static code analysers. All in all, the aforementioned results suggest that it is essential to harvest a reliable list of common libraries and also important to pay special attention to them when conducting Android-related investigations. [All rights reserved Elsevier]. (58 refs)
Inspec controlled terms: Android (operating system) - invasive software - learning (artificial intelligence) - mobile computing - program diagnostics
Uncontrolled terms: common library - Android apps - assistant code - Android libraries - potential common libraries - popular Android app analyses
Classification Code: C6190V Mobile, ubiquitous and pervasive computing - C6130S Data security - C6150G Diagnostic, testing, debugging and evaluating systems - C6150J Operating systems - C6170K Knowledge engineering techniques
IPC Code: G06F9/44 - G06F9/46 - G06F11/36 - G06F15/18 - G06F21/00 - G06N5/04
Treatment: Practical (PRA)
Database: Inspec
Data Provider: Engineering Village