1. Automated Third-Party Library Detection for Android Applications: Are We There Yet?

Xian Zhan (1); Lingling Fan (2); Tianming Liu (3); Sen Chen (2); Li Li (3); Haoyu Wang (4); Yifei Xu (5); Xiapu Luo (1); Yang Liu (2)

Source: *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, p 919-30, 2020; **ISBN-13:** 978-1-4503-6768-4; **DOI:** 10.1145/3324884.3416582; **Conference:** 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), 21-25 Sept. 2020, Melbourne, VIC, Australia; **Publisher:** IEEE Computer Society, Los Alamitos, CA, USA

Author affiliation: (1) Hong Kong Polytechnic University, China (2) Nanyang Technological University, Singapore (3) Monash University, Melbourne, VIC, Australia (4) Beijing University of Posts and Telecommunications, China (5) Southern University of Science and Technology, China

Abstract: Third-party libraries (TPLs) have become a significant part of the Android ecosystem. Developers can employ various TPLs with different functionalities to facilitate their app development. Unfortunately, the popularity of TPLs also brings new challenges and even threats. TPLs may carry malicious or vulnerable code, which can infect popular apps to pose threats to mobile users. Besides, the code of third-party libraries could constitute noises in some downstream tasks (e.g., malware and repackaged app detection). Thus, researchers have developed various tools to identify TPLs. However, no existing work has studied these TPL detection tools in detail; different tools focus on different applications with performance differences, but little is known about them. To better understand existing TPL detection tools and dissect TPL detection techniques, we conduct a comprehensive empirical study to fill the gap by evaluating and comparing all publicly available TPL detection tools based on four criteria: effectiveness, efficiency, code obfuscation-resilience capability, and ease of use. We reveal their advantages and disadvantages based on a systematic and thorough empirical study. Furthermore, we also conduct a user study to evaluate the usability of each tool. The results showthat LibScout outperforms others regarding effectiveness, LibRadar takes less time than others and is also regarded as the most easy-to-use one, and LibPecker performs the best in defending against code obfuscation techniques. We further summarize the lessons learned from different perspectives, including users, tool implementation, and researchers. Besides, we enhance these open-sourced tools by fixing their limitations to improve their detection ability. We also build an extensible framework that integrates all existing available TPL detection tools, providing online service for the research community. We make publicly available the evaluation dataset and enhanced tools. We believe our work provides a clear picture of existing TPL detection techniques and also give a road-map for future directions. (0 refs)

Inspec controlled terms: Android (operating system) - data mining - invasive software - mobile computing - security of data - smart phones

Uncontrolled terms: automated third-party library detection - Android applications - third-party libraries - TPLs - Android ecosystem - app development - malicious code - vulnerable code - popular apps - mobile users - app detection - different tools focus - comprehensive empirical study - publicly available TPL detection tools - code obfuscation-resilience capability - systematic study - thorough empirical study - code obfuscation techniques - tool implementation - open-sourced tools - existing available TPL detection tools - enhanced tools - existing TPL detection techniques

Classification Code: C6130S Data security - C6170K Knowledge engineering techniques - C6190V Mobile, ubiquitous and pervasive computing

IPC Code: G06F9/44 - G06F9/46 - G06F21/00 - G06N5/04 - H04M1/725

Treatment: Practical (PRA)

Database: Inspec

Data Provider: Engineering Village