

PEDroid: Automatically Extracting Patches from Android App Updates (Artifact)

Hehao Li ✉

Shanghai Jiao Tong University, China

Yizhuo Wang ✉

Shanghai Jiao Tong University, China

Yiwei Zhang ✉

Shanghai Jiao Tong University, China

Juanru Li ✉🏠

Shanghai Jiao Tong University, China

Dawu Gu ✉

Shanghai Jiao Tong University, China

Abstract

We propose an approach to automatically identify and extract patches from updated Android apps by comparing the updated versions and their predecessors. PEDROID, a prototype patch extraction tool against Android apps, consists of two phases: differential analysis and patch identification. We

evaluated it with a set of popular open-source apps to demonstrate its effectiveness. PEDROID achieves a recall of 92% in differential analysis and successfully identifies 28 of 36 patches in patch identification. We also provide specific guidance on reproducing the experimental results.

2012 ACM Subject Classification Software and its engineering → Software evolution

Keywords and phrases Diffing, Patch Identification, Android App Analysis, App Evolution

Digital Object Identifier 10.4230/DARTS.8.2.24

Funding This work was supported by the National Key Research and Development Program of China (No.2020AAA0107803).

Acknowledgements We are grateful to our reviewers for their valuable support and suggestions.

Related Article Hehao Li, Yizhuo Wang, Yiwei Zhang, Juanru Li, and Dawu Gu, “PEDroid: Automatically Extracting Patches from Android App Updates”, in 36th European Conference on Object-Oriented Programming (ECOOP 2022), LIPIcs, Vol. 222, pp. 21:1–21:31, 2022.

<https://doi.org/10.4230/LIPIcs.ECOOP.2022.21>

Related Conference 36th European Conference on Object-Oriented Programming (ECOOP 2022), June 6–10, 2022, Berlin, Germany

Evaluation Policy The artifact has been evaluated as described in the ECOOP 2022 Call for Artifacts and the ACM Artifact Review and Badging Policy.

1 Scope

We propose a bytecode-level patch extraction approach, named PEDROID, to automatically locate the patches in updates of Android apps. PEDROID consists of two phases: 1) differential analysis to locate the modified methods in two versions of an app, and 2) patch identification to identify patches among the modified methods. Differential analysis is implemented in Python, and we disassemble the Dex bytecode of APK files by the tool *baksmali*. For patch identification, our taint analysis is based on the taint engine provided by Find Security Bugs [1], and the analysis of internal semantics is implemented in Java on top of Soot [2], a framework for analyzing and transforming Java and Android apps.



© Hehao Li, Yizhuo Wang, Yiwei Zhang, Juanru Li, and Dawu Gu; licensed under Creative Commons License CC-BY 4.0

Dagstuhl Artifacts Series, Vol. 8, Issue 2, Artifact No. 24, pp. 24:1–24:2



DAGSTUHL ARTIFACTS SERIES
Schloss Dagstuhl – Leibniz-Zentrum für Informatik,
Dagstuhl Publishing, Germany



24:2 PEDroid: Automatically Extracting Patches from Android App Updates (Artifact)

In addition to the programs and the requirements, the benchmark *dBench* we collected also could be found here. And we provided detailed instructions to guide how to produce the results in the paper.

2 Content

The artifact package includes:

- a Docker image that includes the programs and data;
- a benchmark *dBench* used in paper;
- a documentation (in Markdown format) that provides guidance on how to use the artifact and obtain the results in the paper.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: <https://github.com/huawanbibi/PEDroid>.

4 Tested platforms

We have carried out all the experiments on a server running Ubuntu 18.04 x64 with two Intel Xeon Gold 5122 Processors (each has eight logical cores at 3.60 GHz) and 128GB RAM.

5 License

The artifact is available under MIT license.

6 MD5 sum of the artifact

b0b89511355472af631311792fdf8f89

7 Size of the artifact

1.51 GiB

References

- 1 Find security bugs, accessed: November 2021. URL: <https://find-sec-bugs.github.io/>.
- 2 Soot - a java optimization framework, accessed: November 2021. URL: <https://github.com/soot-oss/soot>.