TaintBench: Automatic Real-World Malware Benchmarking of Android Taint Analyses

Linghui Luo\(^1\), Felix Pauck\(^2\), Goran Piskachev\(^3\), Manuel Benz\(^4\), Ivan Pashchenko\(^5\), Martin Mory\(^6\), Eric Bodden\(^7\), Ben Hermann\(^8\), Fabio Massaci\(^9\)

Abstract: Due to the lack of established real-world benchmark suites for static taint analyses of Android applications, evaluations of these analyses are often restricted and hard to compare. Even in evaluations that do use real-world applications, details about the ground truth in those apps are rarely documented, which makes it difficult to compare and reproduce the results. Our recent study fills this gap. It first defines a set of sensible construction criteria for such a benchmark suite. It further proposes the TaintBench benchmark suite designed to fulfil these construction criteria. Along with the suite, this paper introduces the TaintBench framework, which allows tool-assisted benchmark suite construction, evaluation and inspection. Our experiments using TaintBench reveal new insights of popular Android taint analysis tools.

Keywords: Taint analysis; Benchmark; Real-world benchmark; Android malware

1 Summary

In our recent study [Lu22], we proposed the TaintBench framework and suite. To build the TaintBench framework, we mainly extended: (I) Jadx [T] to be able to construct and document benchmark apps, (II) ReproDroid [T] to automatically load and evaluate benchmarks, and (III) VSC [T] (Visual Studio Code) to inspect and understand benchmark results (cf. Figure 1). We employed the TaintBench framework to construct the TaintBench suite. In a thorough and long-lasting process we selected 39 malware apps to be included in the final suite and precisely documented 221 taint flows as a baseline ground truth. 186 of these taint...
flows are *expected* to be found by taint analysis tools. The remaining 35 taint flows are *not* expected (*unexpected*) to be found, however, these taint flows have been detected by analysis tools and were identified as false positives.

After constructing the suite, we employed the framework again to automatically re-evaluate two state-of-the-art analysis tools (*AMANDROID* [We14] and *FLOWDROID* [Ar14]). We considered two different versions of both tools. On the one hand, the same version as the one used in the *REPRODROID* study [PBW18], and on the other hand, the up-to-date version (when we refer to the latter, we mark it with an asterisk). For comparison purposes we also employed the *DROIDBENCH* micro benchmark suite in our experiments. Figure 2 shows the results of two experiments we conducted. For the first experiment (DB1 and TB1) we configured the analysis tools such that the default list of sources and sinks was used. Lists of sources and sinks with respect to the respective benchmark suite have been used during the second experiment (DB2 and TB2). The outcome already shows three findings: (1) The tools are less effective on real-world malware apps in *TaintBench*—they have especially low recall, (2) The tools detect way more correct taint flows when using benchmark specific lists of sources and sinks, and (3) *AMANDROID* and *FLOWDROID* are less precise than their predecessors. These effects are harder to notice in the context of *DroidBench*. Hence, we suggest to employ *TaintBench* in future evaluations of Android taint analyses.

### 2 Data Availability

The *TaintBench* framework, the suite and all results of the study [Lu22] can be obtained from the project's website: https://TaintBench.github.io (also available on Zenodo: https://zenodo.org/record/5734328)

### Bibliography

[Ar14] Arzt, Steven; Rasthofer, Siegfried; Fritz, Christian; Bodden, Eric; Bartel, Alexandre; Klein, Jacques; Traon, Yves Le; Octeau, Damien; McDaniel, Patrick D.: FlowDroid: precise
context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps. In: Proceedings of PLDI, 2014. ACM, 2014.

[Lu22] Luo, Linghui; Pauck, Felix; Piskachev, Goran; Benz, Manuel; Pashchenko, Ivan; Mory, Martin; Bodden, Eric; Hermann, Ben; Massacci, Fabio: TaintBench: Automatic real-world malware benchmarking of Android taint analyses. Empir. Softw. Eng., 27(1):16, 2022.

[PBW18] Pauck, Felix; Bodden, Eric; Wehrheim, Heike: Do Android taint analysis tools keep their promises? In: Proceedings of the 26th ESEC/FSE, 2018. ACM, 2018.

[We14] Wei, Fengguo; Roy, Sankadas; Ou, Xinming; Robby: Amandroid: A Precise and General Inter-component Data Flow Analysis Framework for Security Vetting of Android Apps. In: Proceedings of CCS, 2014. ACM, 2014.

[T] Tools, https://github.com/skylot/jadx, https://FoelliX.github.io/ReproDroid, https://code.visualstudio.com (last accessed: 01.12.2021).