Reproducing Taint-Analysis Results with ReproDroid

Felix Pauck[1], Eric Bodden[2], Heike Wehrheim[3]

Abstract:

More and more Android taint-analysis tools appear each year. Any paper proposing such a tool typically comes with an in-depth evaluation of its supported features, accuracy and ability to be applied on real-world apps. Although the authors spent a lot of effort to come up with these evaluations, comparability is often hindered since the description of their experimental targets is usually limited.

To conduct a comparable, automatic and unbiased evaluation of different analysis tools, we propose the framework ReproDroid. The framework enables us to precisely declare our evaluation targets, in consequence we refine three well-known benchmarks: DroidBench, ICC-Bench and DIALDroid-Bench. Furthermore, we instantiate this framework for six prominent taint-analysis tools, namely Amandroid, DIALDroid, DidFail, DroidSafe, FlowDroid and IccTA. Finally, we use these instances to automatically check whether different promises commonly made in the associated proposing papers are kept.

Keywords: Android Taint Analysis; Tools; Benchmarks; Empirical Studies; Reproducibility

1 The ReproDroid Study

Fig. 1: Overview of ReproDroid

Figure 1 depicts the conceptual idea behind our Android benchmark reproduction framework ReproDroid, which helps to (i) precisely specify the ground-truth of benchmarks and allows to (ii) automatically execute and (iii) evaluate benchmarks. First, the app-set representing a certain benchmark must be loaded into ReproDroid. In a semi-automatic manner the ground-truth can then be specified for the associated apps. Effectively a list of precisely defined sources, sinks and expected flows between them is generated this way. A *refined* benchmark, for which such a list exists, can be stored and reused anytime without having to redo this first refinement step. Once a refined benchmark becomes available in ReproDroid an arbitrary analysis tool can automatically be executed for each benchmark case. In order to do so, only

[1] Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany fpauck@mail.uni-paderborn.de
[2] Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany eric.bodden@uni-paderborn.de
[3] Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany wehrheim@uni-paderborn.de

doi:10.18420/SE2020_36

ReproDroid's configuration needs to be adapted to suit the designated analysis tool. In the end, results produced this way are collected and automatically compared against the beforehand specified ground-truth.

In our evaluation we took a look at three different categories of promises given in the proposing papers considering Amandroid [We14], DIALDroid [Bo17], DidFail [Kl14], DroidSafe [Go15], FlowDroid [Ar14] and IccTA [Li15]. The first type of evaluated promises (*feature-promises*) considers the features and sensitivities which are claimed to be supported by these six tools. Second, *accuracy-promises* are checked by attempting to reproduce the F-measure values reported (+/- 0.2) for certain benchmarks or subsets. The authors of the papers associated with all six tools promise that their tool is real world ready. Thus, lastly we checked these *real-world-promises*. Table 2 shows the number of promises given and how many could not be confirmed per category – a verbose version of these results is available in [PBW18]. As we can see, only a minority of promises cannot be confirmed. Nonetheless, the accuracy-promises are not strictly kept and in particular the real-world-promises are not fully kept by any tool.

Promises	Given / Unconfirmed	
Feature	64	5
Accuracy	12	6
Real-World	6	6

Fig. 2: Evaluation results

References

[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, pp. 259–269, 2014.

[Bo17] Bosu, Amiangshu; Liu, Fang; Yao, Danfeng (Daphne); Wang, Gang: Collusive Data Leak and More: Large-scale Threat Analysis of Inter-app Communications. In: Proceedings of AsiaCCS, 2017. ACM, pp. 71–85, 2017.

[Go15] Gordon, Michael I.; Kim, Deokhwan; Perkins, Jeff H.; Gilham, Limei; Nguyen, Nguyen; Rinard, Martin C.: Information Flow Analysis of Android Applications in DroidSafe. In: Proceedings of the 22nd NDSS, 2015. The Internet Society, 2015.

[Kl14] Klieber, William; Flynn, Lori; Bhosale, Amar; Jia, Limin; Bauer, Lujo: Android taint flow analysis for app sets. In: Proceedings of the 3rd SOAP, 2014. ACM, pp. 5:1–5:6, 2014.

[Li15] Li, Li; Bartel, Alexandre; Bissyandé, Tegawendé F.; Klein, Jacques; Traon, Yves Le; Arzt, Steven; Rasthofer, Siegfried; Bodden, Eric; Octeau, Damien; McDaniel, Patrick D.: IccTA: Detecting Inter-Component Privacy Leaks in Android Apps. In: Proceedings of the 37th ICSE, 2015. IEEE Computer Society, pp. 280–291, 2015.

[PBW18] Pauck, Felix; Bodden, Eric; Wehrheim, Heike: Do Android taint analysis tools keep their promises? In: Proceedings of the 26th ESEC/FSE, 2018. ACM, pp. 331–341, 2018.

[We14] Wei, Fengguo; Roy, Sankardas; 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, pp. 1329–1341, 2014.