Cooperative Android App Analysis with CoDiDroid

Felix Pauck[1], Heike Wehrheim[2]

Abstract: Novel Android app analysis tools as well as improved versions of available tools are frequently proposed. These proposed tools often tackle a specific single issue that cannot be handled with existing tools. Consequently, the best analysis possible should use the advantages of each and every tool. With CoDiDroid we present an analysis framework that allows to combine analysis tools such that the best out of each tool is used for a more comprehensive and more precise cooperative analysis. Our experimental results show indeed that CoDiDroid allows to setup cooperative analyses which are beneficial with respect to effectiveness, accuracy and scalability.

Keywords: Android Taint Analysis; Tools; Cooperation; Precision

1 Cooperative Analysis

Combinations of different Android app analysis tools have been used before to enhance existing analysis. One well-known example is IccTA, a tool that beneficially combines IC3 with FlowDroid. Thereby the intra-component analysis of FlowDroid is lifted-up to inter-component level. However, this combination is hard-coded in the tool itself. Its components cannot be swapped out and the analysis cannot be extended by including another tool without adapting IccTA. In a *cooperative analysis* as proposed by Pauck and Wehrheim [PW19] it is possible to combine arbitrary analyses. The following example illustrates what a cooperative analysis is capable of. The section thereafter introduces the framework required to compose a cooperative Android app analysis, namely CoDiDroid [Pa19].

Example

The example depicted in Figure 1 shows two *taint flows* that leak sensitive information. A taint flow describes the connection of a *source* and a *sink*. A source extracts sensitive information. In the example the device identification number represents the extracted sensitive data which is read in statement s_1 (see Figure 1). Two sinks (s_4, s_7) may leak this information via logging or sending an SMS. In order to detect both leaks an analysis tool must be (1) lifecycle-aware, (2) able to resolve reflection, (3) analyze native code and (4) successfully handle Inter-App-Communication (IAC). Sadly, there exists no such analysis tool. However, FlowDroid is able to perform an intra-component, lifecycle-aware taint analysis, DroidRA is able to resolve reflection, NOAH can discover sources and sinks in native code, IC3 allows to gather information about an app's exit and entry points which can be used by PIM to find connections between those that are realized via intents (IAC).

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

doi:10.18420/SE2021_30

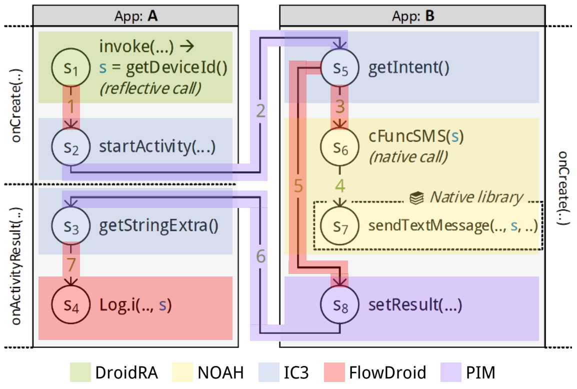

Fig. 1: Example: Two taint flows that include reflection, IAC and a native method call

Most of these tools were designed as standalone tools without having cooperation in mind. Nonetheless, CoDiDroid allows to compose a single analysis that not least successfully analyzes the described example by detecting both taint flows. The figure visualizes which parts of these taint flows are detected by which tool.

2 The CoDiDroid framework

The cooperative (and distributed) analysis framework CoDiDroid takes (1) a task in form of a query as input, (2) generates subtasks to answer each part of the query, (3) distributes these subtasks onto tools available in its configuration and in the end (4) merges tool answers to respond to the initial query. The Android App Analysis Query Language (AQL) and its execution system (AQL-System) is extensively used to do so. Along with CoDiDroid we thus proposed AQL-WebServices that allow to execute tools in different environments and exchange results. Also by employing the AQL, ReproDroid [PBW18] could be used to evaluate CoDiDroid against the state-of-the-art, revealing that cooperative analysis pays off by outperforming standalone tools with respect to effectiveness, scalability and accuracy in terms of precision, recall and F-measure.

Note, detailed evaluation results and all the reference of the mentioned tools can be found in the original paper [PW19].

Bibliography

[Pa19] Pauck, Felix: , CoDiDroid, 2019. `https://FoelliX.github.io/CoDiDroid` last accessed 11/16/2020.

[PBW18] Pauck, Felix; Bodden, Eric; Wehrheim, Heike: Do Android taint analysis tools keep their promises? In: Proceedings of ESEC/FSE 2018, Lake Buena Vista, FL, USA. ACM, 2018.

[PW19] Pauck, Felix; Wehrheim, Heike: Together strong: cooperative Android app analysis. In: Proceedings of ESEC/FSE 2019, Tallinn, Estonia. ACM, 2019.