Pattern-based Inter-Component Communication Discovery for Android Applications

Vinicius Souza
Federal University of Pernambuco
Recife, Brazil
vcps@cin.ufpe.br

Leopoldo Teixeira
Federal University of Pernambuco
Recife, Brazil
lmt@cin.ufpe.br

Marcelo d'Amorim
Federal University of Pernambuco
Recife, Brazil
damorim@cin.ufpe.br

ABSTRACT

A number of program analysis have been proposed to identify inter-component communication (ICC) between Android applications. These analysis are often designed to account for every possible way (*flow? context?*) of setting up such communications. In this paper, we explore the fact that such communications often follow some patterns. Therefore, we propose a lightweight, pattern-based approach for detecting ICC. We apply this analysis to XX applications from the Play store, identifying $YY\%$ of ICC specifications. Compared to the state-of-the-art, we reduce in $ZZ\%$ the precision (*or recall*), while, on the other hand, we take $TT\%$ of the time to perform our analysis. These results show that our analysis, although imprecise, can be integrated

CCS Concepts

•General and reference → Empirical studies; •Software and its engineering → Automated static analysis;

1. INTRODUCTION (1 PAGE)

Android marketshare, numbers, importance. Briefly explaing that Android applications interact through intents and ICC

Discuss past analysis, how they often are built for soundness, and therefore suffer on scalability. Also mention about trying to capture each and every way of building intents and ICC

Explain that intents and ICC are built following patterns. Use a small example? We can explore this to create lightweight analysis, that are imprecise, but can account for most of the scenarios, as we show in the evaluation (hopefully).

Brief idea of the solution.

We make the following contributions:

- We present a tool for computing Android ICC in a pattern-based way;
- We perform a study with XX applications, comparing the results with the state-of-the-art analysis for detecting ICC [3, 2, 1]
- something about client analysis here? *We analyze the impact that our solution has on tools that perform taint analysis on Android applications...*

FSE '16 Seattle, USA

DOI: 10.475/123_4

2. MOTIVATION (1-2 PAGES)

Quickly explain what is ICC and give an example of intent and communication between Android components

Go over existing solutions and explain why they are so expensive. Is this dependent on the size of the program, or in the number of ICC points?

- epicc: build the call graph amounts to large part of the time, *still getting the grasp of the paper, but the problem seems to be that they model the entire application (therefore causing the graph to be too big), while we are mainly interested on specific points of communication*
- ic3: builds on top of epicc, including the constraint propagation solver for resolving strings and URIs. Therefore, it can only be as fast as epicc is, but not faster; depending on the intent values it will also spend some time resolving the strings
- gator (atanas rountev): still working through the ICSE paper and have also downloaded the tool to figure it out

Characterize patterns for creating intents and establishing ICC. Gather the data collected by Vinicius and present here, that in general, half of the ICCs are explicit, and we can infer such information without sophisticated analysis. (*challenge seems to be implicit intents*)

A explicacao e apresentacao de padroes se daria aqui ou na proxima secao?

3. SOLUTION (1-2 PAGES)

Explain the idea of patterns using 1-2 concrete examples. (1 from motivation and maybe other here)

The intuition is that patterns can often occur together, therefore we might calculate the intersection and give a precise answer over the ICC specification in most of the program points.

3.1 Explicit ICC

- Intent creation with class name (anonymous object)

- Use setComponent/setClass on Intent object

Both patterns can be specialized when the class name is in a local variable, static class attribute, and so on... Explain this.

3.2 Implicit ICC

- Intent with get/put extras
- Intent with Action, Data, and Category fields

3.3 Dynamic Broadcast Receivers

Still working on which patterns we can establish for this case.

3.4 Implementation

Explain the overall implementation, we collect the information based on the patterns established above.

Then we have to perform a "join" of this information, to make it more precise

This results in a number of ICC specifications, or interfaces. Explain the idea using the 1-2 examples given before.

4. EVALUATION (4 PAGES)

4.1 Experimental Setup

Explicitly detail the environment in which we performed all of the analysis detailed below

Discuss in which

4.2 Quantitative Analysis

Analysis on precision/recall of the specifications.

Epicc evaluation consists of inferring specifications automatically, and if there is only "one" possible specification, they consider it precise. One way would be to adopt the same strategy for considering our precision/recall.

I am going through FlowDroid [?] and Atanas' GATOR paper [?], to understand how they considered the ground truth.

4.3 Comparison with existing ICC tools

Compare deltas with Epicc [3], IC3 [?], ComDroid [1]

4.4 Application to Vulnerabilities

Use IccTA [2] with our ICC information and compare performance and analysis results.

5. RELATED WORK (1 PAGE)

Epicc [3]
IC3 [?]
IccTA [2]
FlowDroid [?]
GATOR [?]
Soundiness manifesto [?]
William Enck's papers [?] (grab more)

6. CONCLUSIONS (1/2 PAGE)

7. REFERENCES

[1] E. Chin, A. P. Felt, K. Greenwood, and D. Wagner. Analyzing inter-application communication in android. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 239–252, New York, NY, USA, 2011. ACM.

[2] L. Li, A. Bartel, T. F. Bissyandé, J. Klein, Y. Le Traon, S. Arzt, S. Rasthofer, E. Bodden, D. Octeau, and P. McDaniel. Iccta: Detecting inter-component privacy leaks in android apps. In *Proceedings of the 37th International Conference on Software Engineering - Volume 1*, ICSE '15, pages 280–291, Piscataway, NJ, USA, 2015. IEEE Press.

[3] D. Octeau, P. McDaniel, S. Jha, A. Bartel, E. Bodden, J. Klein, and Y. Le Traon. Effective inter-component communication mapping in android with epicc: An essential step towards holistic security analysis. In *Proceedings of the 22Nd USENIX Conference on Security*, SEC'13, pages 543–558, Berkeley, CA, USA, 2013. USENIX Association.