



Seminar

Foundations of Static Analyses

Winter Semester 22/23

TuCan-No: 20-00-1028-se
Course Type: 2SWS / 3CPs
Workload: ~90hours

Prof. Dr.-Ing. Mira Mezini

Process

- **Today:** Send your favorite 3 topics to roth@st.informatik.tu-darmstadt.de
Subject: “[FoSa22] : Topic Selection”
- **Tomorrow:** We inform you about your assigned topic via e-mail
- **Next:** Contact your supervisor and schedule a meeting to discuss the topic and requirements in detail
- **End of January:** Discuss a preliminary version with your supervisor
- **February 6th:** Send the final version (4 pages/person + appendix, acmart - <https://www.acm.org/publications/proceedings-template>)
- **About third week of february:** Blockseminar (date: tba)
(Details will be announced)

WebAssembly (Wasm) RoadMap

WebAssembly is a novel web-language at an early stage of development. The goal of this topic is to recap the current status and give an overview of the planned Wasm development, e.g., new language features.

Task: Familiarize yourself with the development of Wasm. Write an essay about Wasm's development - the current state and the future.

Suitable for: 1 - 2 people

Starting Point:

- <https://dl.acm.org/doi/10.1145/3062341.3062363>
- <https://webassembly.github.io/sign-extension-ops/core/index.html>
- <https://webassembly.github.io/spec/core/>
- <https://webassembly.org/>

Contact: florian.breitfelder@tu-darmstadt.de

Call Graphs in WebAssembly (Wasm)

Call graphs are essential for static analysis, e.g., for identifying data flows. They describe the call relation between subroutines of a program. Since it is not always easy to determine all call relations between functions because of different language features, it is necessary to use different techniques or over-approximation. For WebAssembly it is partly necessary to construct a call graph with over-approximation, which leads to a loss of precision.

Task: Familiarize yourself with call graphs in general, understand function calls in WebAssembly and possible strategies for call graph construction in WebAssembly. For this purpose, it is also worth to examine existing analysis frameworks.

Suitable for: 1 - 2 people

Starting Point:

- <https://dl.acm.org/doi/10.1145/3062341.3062363>
- https://www.software-lab.org/publications/asplos2019_Wasabi.pdf
- <https://github.com/acieroid/wassail>
- <https://github.com/WebAssembly/wasp>
- https://cris.vub.be/ws/files/75991494/informationflow_copyright.pdf
- <https://www.academia.edu/download/69022311/tse.1979.23418320210903-3806-1li6btb.pdf>

Contact: florian.breitfelder@tu-darmstadt.de

Cross-Language Vulnerabilities

While previous security research was focused on vulnerabilities of single language programs, multi-language software introduces new forms of vulnerabilities. The goal of this topic is to survey cross-language related vulnerabilities.

Task: Read the following paper as a starting point, search and read related papers and write a paper about multi-language related vulnerabilities.

Suitable for: 1 - 2 people

Starting Point:

- <https://www.usenix.org/conference/usenixsecurity20/presentation/lehmann>

Contact: roth@cs.tu-darmstadt.de

Call Graphs in JavaScript

Call graphs describe the call relations of a program and are the basis for client analyses, e.g., for determining data flows. JavaScript encompasses multiple challenges for determining call graphs due to its dynamic nature. The goal of this topic is to survey current approaches of computing call graphs in JavaScript.

Task: Read the following paper, search and read related ones, and summarize and compare the approaches

Suitable for: 1 - 2 people

Starting point:

- https://www.crysys.hu/publications/files/setit/cpaper_szte_AntalHTFGy19scam.pdf

Contact: roth@cs.tu-darmstadt.de

App SAST, DAST & SCA in Research and Industry

Currently, various analysis tools are available for Android and iOS apps, but there is no overview of which tools offer which benefits and in which licenses these tools are available.

Task: Find the most commonly used SAST, DAST, and SCA tools on the web or in a scientific context and summarize their benefits and licensing models.

Suitable for: 1 - 4 people

Possible Sources:

- <https://core.ac.uk/download/pdf/31219796.pdf>
- <http://library.usc.edu/ph/ACM/SIGSAC%202017/spsm/p33.pdf>
- <https://checkmarx.com>
- <https://www.microfocus.com/de-de/cyberres/application-security/static-code-analyzer>
- <https://mobsf.live>
- <https://www.immuniweb.com/mobile>

Contact: leonid.glanz@tu-darmstadt.de

Advanced Hands-on Training
Software Development Tools
Winter Semester 22/23

TuCan-No: 20-00-0673-pr

Course Type: 4SWS / 6CPs

Workload: ~**180hours**



Prof. Dr.-Ing. Mira Mezini

Process

- **Today:** Send an e-mail with your three preferred topics and your knowledge about the topics to: leonid.glanz@tu-darmstadt.de
- **Tomorrow:** Assignment of topics
- **Next:** Contact your supervisor to discuss details of your topic
- **During Hands-on Training:** Bi-weekly meetings with supervisor in an agile process
 - Discuss the current state and the next steps
- **End of March:** Final submission of artifacts



Android Analysis: Not Reinvent the Wheel but Just Change the Tires

Many different tools are currently used for the security analysis of Android apps. These tools provide useful results, but are very slow today because they are based on old frameworks.

- **Task:** Build different analysis tools on top of our analysis framework
- **Languages & Frameworks:**  **Scala**  **OPAL**
- **Suitable for:** 1 - 4 people
- **Contact:** florian.breitfelder@tu-darmstadt.de

Matching & Extending Analysis Summaries

Many analyses build summaries of code parts to analyze apps with similar code parts faster. This also makes it possible to analyze large apps such as Facebook Messenger. However, these summaries cannot be used if the code is slightly modified.

- **Task:** Match the modified code to summaries and extend the summaries with results of analyses.
- **Languages & Frameworks:**  **Scala**  **OPAL**
- **Suitable for:** 1 - 2 people
- **Contact:** florian.breitfelder@tu-darmstadt.de

Library Detection & Dependencies

Often libraries depend on others and are thus integrated together in an app. However, the connection between the libraries can no longer be traced afterwards and not all library versions are known in the app.

- **Task:** Build a crawler that extracts dependency information from Maven Central or MVNRepository and extract library information from apps
- **Languages & Frameworks:** JAVA &  **Scala**
- **Suitable for:** 1 - 2 people
- **Contact:** florian.breitfelder@tu-darmstadt.de

Mapping of Obfuscated Names

For security reasons, companies often obfuscate the code of apps. However, to identify the affected code locations for crash reporting, obfuscators provide mappings to all code locations.

- **Task:** Find common mapping formats and create a tool that parses the mapping formats and returns the original names for obfuscated names.
- **Languages & Frameworks:** **JAVA** or other languages
- **Suitable for:** 1 person
- **Contact:** florian.breitfelder@tu-darmstadt.de

Deobfuscation for the Security Community

We have developed a tool to recover obfuscated strings in Android apps. This tool is precious for the security analysis community, but currently, it is only usable in our environment.

- **Task:** Integrate our tool into the ghidra technology stack that is well known by security analysts.
- **Languages & Frameworks:** Ghidra &  **Scala**
- **Suitable for:** 1 - 2 people
- **Contact:** leonid.glanz@tu-darmstadt.de



Cross-language adapter for JavaScript in Android

OPAL is a static analysis framework currently focused on Java Bytecode. We want to analyze software written in multiple languages. There is already an analysis in OPAL that recognizes the JavaScriptEngine and collects all relevant information.

- **Task:** Write an analysis in OPAL that recognizes the JavaScript interface in Android and collects all relevant information
- **Languages & Frameworks:**  OPAL,  **Scala**, Java, JavaScript, Android
- **Suitable for:** 1 person
- **Contact:** roth@cs.tu-darmstadt.de

Cross-language adapter for Python

OPAL is a static analysis framework currently focused on Java Bytecode. We want to analyze software written in multiple languages. Currently there is an analysis in OPAL that recognizes the JavaScriptEngine and collects all relevant information.

- **Task:** Write an analysis in OPAL that recognizes the Java-Python interface and collects all essential informations
- **Languages & Frameworks:**  OPAL,  **Scala** , Java, Python
- **Suitable for:** 1 person
- **Contact:** roth@cs.tu-darmstadt.de