

Advanced Hands-on Training
Software Development Tools
Winter Semester 23/24

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


Matching & Extending Missing Code

Many code snippets are not usable, because of missing code pieces.

- **Task:** Extract knowledge from a manually analyzed project to train a machine learning model that creates stubs for missing code pieces.
- **Languages & Frameworks:** Any machine learning model &  docker
- **Suitable for:** 1 - 2 people
- **Contact:** leonid.glanz@tu-darmstadt.de


Support for Machine Learning of Texts

Currently, text classification is a very popular tasks but needs a pre-labeled data set to enable classifications for new categories.

- **Task:** Develop a GUI that shows text and let users label specific parts which can be fed to a machine learning model as an advanced feature.
- **Languages & Frameworks:** Any language we support &  docker
- **Suitable for:** 1 - 2 people
- **Contact:** leonid.glanz@tu-darmstadt.de


Deep Google Play Store Crawler

There are many crawlers for the Google Play store, but none provide the essential information that a security analyst needs.

- **Task:** Develop a crawler that downloads the APK, app name, icon, and all relevant privacy information for a given package id.
- **Languages & Frameworks:** Any language we support &  docker
- **Suitable for:** 1 - 2 people
- **Contact:** leonid.glanz@tu-darmstadt.de

Improve Android Bytecode Analyses with more Information

Security analysts need additional information to assess Android apps. This information includes Android archive (AAR), ZIP, JAR, POM or even license files.

- **Task:** Develop a scraper that searches & downloads the above-mentioned files in various repositories, such as Sonatype. It should search for associated files from specific packages or GroupId + ArtifactId.
- **Languages & Frameworks:** Python or Go &  docker
- **Suitable for:** 1 - 2 people
- **Contact:** leonid.glanz@tu-darmstadt.de


Automated Dynamic Analysis of Android Applications

We want to run automated dynamic analyses on Android applications using a headless emulator in VMS.

- **Task:** Set up an easily deployable, general framework that automatically runs android applications and records their behaviour.
- **Languages & Frameworks:** Docker, qemu, mitmproxy
- **Suitable for:** 1 person
- **Contact:** leonid.glanz@tu-darmstadt.de

Automated Config Documentation for OPAL

Our static analysis framework OPAL makes heavy use of JSON configuration files. The configuration options are hard to discover and understand without documentation.

- **Task:** Define a documentation comment syntax for our configuration files & develop a tool (preferably a plugin to the Scala Build Tool sbt) that converts this to a browsable HTML documentation.
- **Languages & Frameworks:** JSON,  **Scala**, (sbt, OPAL)
- **Suitable for:** 1 - 2 people
- **Contact:** dominik.helm@tu-darmstadt.de

Positions & Theses

If you are interested in **HiWi Positions** or **Bachelor- or Master theses** contact:

leonid.glanz@tu-darmstadt.de

Seminar

Foundations of Static Analysis

Winter Semester 23/24

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: “[FoSa23] : 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 9th:** Send the final version (4 pages/person + appendix, acmart - <https://www.acm.org/publications/proceedings-template>)
- **About end of february:** Blockseminar (date: tba)
(Details will be announced)

Precision and Recall Measurement for Call Graphs

Call graphs link methods to the methods they invoke. They are the foundation for all inter-procedural static analysis. Precision and recall are important metrics for the quality of call graphs. The goal of this topic is to survey methods used to determine the precision and/or recall of call graphs.

Task: Search and read papers that measure the precision and/or recall of call graphs; summarize the methods used

Suitable for: 1 - 2 people

Languages and Frameworks: Java, possibly C/C++, etc.

Starting point:

- <https://dl.acm.org/doi/10.1145/1251535.1251542>
- <https://ieeexplore.ieee.org/document/9984208>
- <https://dl.acm.org/doi/pdf/10.1145/3377811.3380441>

Contact: helm@cs.tu-darmstadt.de

Call-Graph Challenges in Dynamic Languages

Call graphs link methods to the methods they invoke. They are the foundation for all inter-procedural static analysis. Dynamic languages such as Python or JavaScript pose challenges to call-graph construction that are different from static languages. The goal of this topic is to survey existing works on such challenges.

Task: Search and read papers on call-graph challenges in Python and JavaScript; summarize the challenges in a structured way

Suitable for: 1 - 2 people

Languages and Frameworks: Python, JavaScript

Starting point:

- <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9402076>
- <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9392986>
- https://software-lab.org/publications/issta2023_wasm_call_graphs.pdf

Contact: helm@cs.tu-darmstadt.de

Analysis of Cross-Platform Frameworks

Cross platform development frameworks allow building apps for multiple platforms from a single platform-independent codebase. Currently, Flutter and React Native are the most widely used frameworks. Research existing work (publications/tools) that analyzes apps built using a cross-platform framework (reverse engineering, static/dynamic analysis etc.)

Task: Study existing research, summarize research challenges in a structured way. Study existing tools, classify them by approach, features, shortcomings etc.

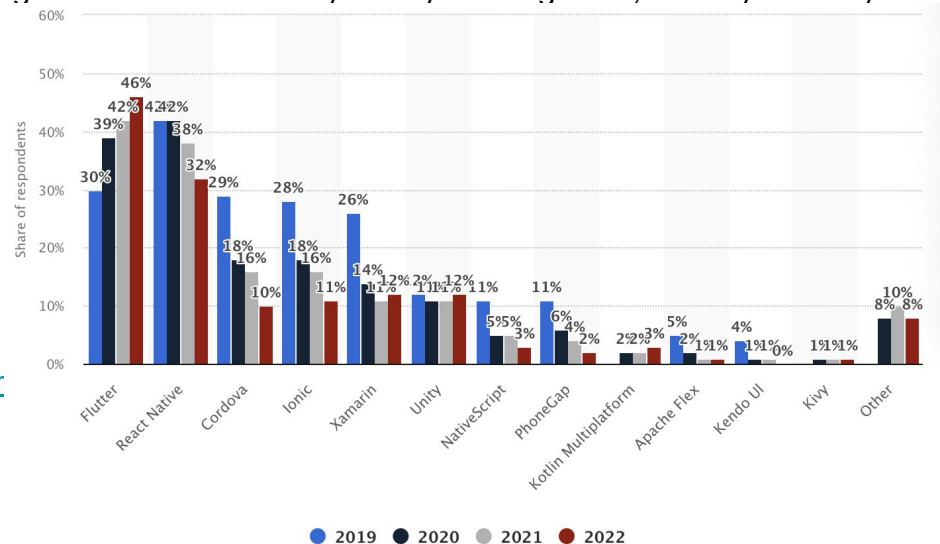
Suitable for: 1 - 2 people

Languages and Frameworks: (Java, C++)

Starting point:

- <https://github.com/ptswarm/reFlutter>
- https://is.muni.cz/th/gogzu/The_Security_of_Flutter_s_Ar

Contact: naeumann@cs.tu-darmstadt.de



Analysis of Multi-Language Software

Modern software consists of multiple-languages, that interact with each other. Multi-language interaction encompasses data flows, malicious behavior, bugs or vulnerabilities. Static analysis can reveal this. There are already approaches to analyze multi-language software statically. For instance, the given paper.

Task: Read the given paper. Search for about 10 related papers. Summarize and compare them in a structured way.

Suitable for: 1 - 2 people

Starting point:

- https://link.springer.com/chapter/10.1007/978-3-030-88806-0_16

Contact: roth@cs.tu-darmstadt.de