SOFTWARE
TECHNOLOGY
GROUP

Seminar
Foundations of Static Analyses
Winter Semester 21/22

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

Prof. Dr.-Ing. Mira Mezini

Process

- **Today:** Send your favorite 3 topics to roth@st.informatik.tu-darmstadt.de
- **Tomorrow:** Assignment of topic (information via e-mail)
- **Next:** Meet with your supervisor to discuss the topic in detail
- **End of January:** discuss a preliminary version with your supervisor
- **February 7th:** Send the final version (4 pages + appendix, acmart - https://www.acm.org/publications/proceedings-template)
- **About second week of february:** Blockseminar (date: tba) (Details will be announced)

Immutability Facets in JavaScript and C

Immutability is the property of a program element stating that it is unchangeable or not changed after its creation. It brings several guarantees, for thread safety, safety and security or is also required by specific APIs. In Our CiFi Paper, we defined a model for analyzing immutability in object oriented languages like Java. The task is now to recognize these or related properties in languages like JavaScript or C.

Task: Read our CiFi-Paper and try to infer related immutability properties in JavaScript and C

Sources:

- https://developer.mozilla.org/en-US/docs/Glossary/Mutable
- https://homes.cs.washington.edu/~mernst/pubs/immutability-aliasing-2013-lncs7850.pdf
- https://en.cppreference.com/w/c
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

Contact: roth@cs.tu-darmstadt.de

Data Flow Analysis of Hybrid Systems

The number of softwares that use different languages arises. For instance core functions implemented in C can be reused on different platforms like Android or Windows. However, these hybrid software systems involve several challenges for static analysis.

Task: Read the following papers, search for related ones and survey the state of the art of hybrid analysis.

Sources:

- Sungho Lee et al. 2016. “HybriDroid: static analysis framework for Android hybrid applications.“
- Claudio Rizzo, 2020, “Static Flow Analysis for Hybrid and Native Android Applications”
- Bai et al. 2018, “BridgeTaint: A Bi-Directional Dynamic Taint Tracking Method for JavaScript Bridges in Android Hybrid Applications”

Contact: roth@cs.tu-darmstadt.de

Static/Dynamic Analysis of WebAssembly

WebAssembly (Wasm) is a new emerging binary format and was originally designed for the web. Today, Wasm is not limited to the web but can also be used to create client and server applications. Due to the popularity and rapid spread of Wasm, security of Wasm applications is getting more and more important. One way to foster security are static/dynamic analyses.

Task: Read the following papers, search for related ones and survey the state of the art of code analyses for the WebAssembly binary format.

Sources:

- Haas, Andreas, et al. 2017, "Bringing the web up to speed with WebAssembly."
- Lehmann, Daniel, and Michael Pradel. 2019, "Wasabi: A framework for dynamically analyzing webassembly."
- Stiévenart, Quentin, and Coen De Roover. 2020, "Compositional Information Flow Analysis for WebAssembly Programs."

Contact: breitfelder@cs.tu-darmstadt.de

Security Issues of WebAssembly

WebAssembly (Wasm) is a new emerging binary format and was originally designed for the web. Today, Wasm is not limited to the web but can also be used to create client and server applications. Due to the popularity and rapid spread of Wasm, security of Wasm applications is getting more and more important. One way to foster security are static/dynamic analyses.

Task: Read the following papers, search for related ones and survey the state of the art of Wasm security issues.

Sources:

- Haas, Andreas, et al. 2017 "Bringing the web up to speed with WebAssembly."
- Lehmann, Daniel, Johannes Kinder, and Michael Pradel. 2020, "Everything old is new again: Binary security of webassembly."
- Musch, Marius, et al. 2019, "New Kid on the Web: A Study on the Prevalence of WebAssembly in the Wild."
- Hilbig, Aaron, Daniel Lehmann, and Michael Pradel. 2021, "An Empirical Study of Real-World WebAssembly Binaries: Security, Languages, Use Cases."

Contact: breitfelder@cs.tu-darmstadt.de

What is fixed in security related bug fixes?

The SmartSHARK dataset combines mined information about code from various sources, e.g., version control system and issue tracking. Some of the bug fixes are manually validated to understand how the fix related to the issue, e.g., if all lines contribute to the fix.

Task: Identify security related bugs, understand what is fixed, and how it compares to all manually validated bugs.

Dataset: SmartSHARK dataset (Python, MongoDB)

Suitable for: 1 to 2 students

Misc: Aim to submit to MSR Mining Challenge in case of good results

Contact: wickert@cs.tu-darmstadt.de

Are security vulnerabilities introduced/removed as part of bug fixes or other commits?

The SmartSHARK dataset combines mined information about code from various sources, e.g., version control system and issue tracking. Some of the bug fixes are manually validated to understand how the fix related to the issue, e.g., if all lines contribute to the fix.

Task: Identify vulnerabilities and understand if the code is introduced/removed as part of a bug fix or within other commits.

Dataset: SmartSHARK dataset (Python, MongoDB)

Suitable for: 1 to 2 students

Misc: Aim to submit to MSR Mining Challenge in case of good results

Contact: wickert@cs.tu-darmstadt.de

Data Flow Analysis of iOS & Android Apps

Task: Read the FlowDroid & ICCTA papers and search for related problem solutions. Identify & survey all sub-problems of dataflow analysis

Sources:

- Arzt, S., Rasthofer, S., Fritz, C., Bodden, E., Bartel, A., Klein, J., ... & McDaniel, P. 2014, “Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps.”
- Li, L., Bartel, A., Bissyandé, T. F., Klein, J., Le Traon, Y., Arzt, S., ... & McDaniel, P., 2015 “Iccta: Detecting inter-component privacy leaks in android apps.”
- Späth, J., Nguyen Quang Do, L., Ali, K., & Bodden, E. 2016, “Boomerang: Demand-driven flow-and context-sensitive pointer analysis for java.”

Suitable for: 2 people

Contact: glanz@cs.tu-darmstadt.de

Library Detection in Android Apps

Task: Read the Orlis & Feichtner et. al. papers and search for related problem solutions. Survey the state of the art of Android library detection

Sources:

- Wang, Y., Wu, H., Zhang, H., & Rountev, A. 2018, "Orlis: Obfuscation-resilient library detection for Android."
- Johannes Feichtner and Christof Rabensteiner, 2019, "Obfuscation-resilient code recognition in android apps (ares)."
- Jiexin Zhang, Alastair R Beresford, and Stephan A Kollmann. 2019, "Libid: Reliable identification of obfuscated third-party android libraries."

Contact: glanz@cs.tu-darmstadt.de

Reverse Engineering of Software Product Line Models

Task: Read the Nadi et al. paper and search for related problem solutions. Survey the state of the art of Software Product Line Model Mining.

Sources:

- Sarah Nadi, Thorsten Berger, Christian Kästner, and Krzysztof Czarnecki. 2014, “Mining configuration constraints: static analyses and empirical results.”
- Galindo, J.A., Benavides, D., Trinidad, P. et al. 2019, “Automated analysis of feature models: Quo vadis?”
- Steven She, Rafael Lotufo, Thorsten Berger, Andrzej Wąsowski, and Krzysztof Czarnecki. 2011. “Reverse engineering feature models.”

Contact: mueller@cs.tu-darmstadt.de

Positions & Theses

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

roth@cs.tu-darmstadt.de