Felix Rath | Curriculum vitae

Kamper Straße 17 – 52064 Aachen – Germany
✱ 1991-03-06 • +49 (172) 7017211 • felixm.rath@gmail.com
felix-rath.dev • futile • frath • eSAXVJwAAAAJ
Mastodon: @frath@mas.to

Experience

Priceloop GmbH **Berlin, Germany (Remote)**
Software Engineer *2023–2024*
Development of a Web-based no-code platform for data driven pricing decisions. Backend and frontend work using Scala and ScalaJs, as well as Rust, with a focus on functional programming. Work in developing and analyzing complex SQL queries and interactions for PostgreSQL. CI and CD work based on GitHub actions, as well as cloud and infrastructure maintenance based on AWS (lambda, ecs, rds) and terraform.

Chair of Communication and Distributed Systems **RWTH Aachen University, Germany**
Ph.D. Student *2016–2022*
Research into program analysis, mostly using Symbolic Execution and Distributed Symbolic Execution, as part of the ERC-sponsored SYMBIOSYS Horizon 2020 research project, focused on KLEE and KleeNet (both C++). Work on a new concolic and symbolic execution engine written in Rust. Advisor to several student theses related to these projects.

Chair of Embedded Software **RWTH Aachen University, Germany**
Student Assistant *2011–2015*
GUI development in Java using the Eclipse framework for the ARCADE program analysis and model checking tool (`https://arcade.embedded.rwth-aachen.de/`).

Publications

[1] Felix Rath, Johannes Krude, Jan Rüth, Daniel Schemmel, Oliver Hohlfeld, Jó Á. Bitsch, and Klaus Wehrle. SymPerf: Predicting Network Function Performance. In *Proceedings of the SIGCOMM Posters and Demos*, SIGCOMM Posters and Demos '17, page 34–36, New York, NY, USA, 2017. Association for Computing Machinery.

[2] Felix Rath, Daniel Schemmel, and Klaus Wehrle. Interoperability-Guided Testing of QUIC Implementations using Symbolic Execution. In *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, EPIQ'18, page 15–21, New York, NY, USA, 2018. Association for Computing Machinery.

[3] Johannes Krude, Jan Rüth, Daniel Schemmel, Felix Rath, Iohannes-Heorh Folbort, and Klaus Wehrle. Determination of Throughput Guarantees for Processor-Based SmartNICs. In *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '21, page 267–281, New York, NY, USA, 2021. Association for Computing Machinery.

Master thesis

Title: *Abstract Memory Layout for Symbolic Execution*

Supervisor: Professor Klaus Wehrle

Advisors: Daniel Schemmel, Oscar Soria Dustmann

Description: My master thesis presents a memory layout for symbolic execution which is focused on pointer tracking using provenance semantics, implemented in C++ for KLEE. By tracking allocation site information together with pointer values through the execution of LLVM IR instructions, it becomes possible to detect new bugs, such as mismatching provenances, during symbolic execution.

Bachelor thesis

Title: *Realistic Rendering of Real-time Weather Conditions*

Supervisor: Professor Leif Kobbelt

Advisor: Jan Robert Menzel

Description: In my bachelor thesis I present the design and implementation of a system to render realistic weather-conditions based on real-time weather information, implemented in C++ using OpenGL and Qt.

Education

RWTH Aachen University — **Aachen, Germany**
M.Sc. Informatik — *2013–2016*

RWTH Aachen University — **Aachen, Germany**
B.Sc. Informatik — *2010–2013*

Languages

German: Native language

English: Fluent in text and speech

French: Basic words and phrases only

Japanese: Basic words and phrases only

Professional Skills

■□□□□	basic knowledge
■■□□□	intermediate knowledge with some project experience
■■■□□	extensive project experience
■■■■□	deepened expert knowledge
■■■■■	expert / specialist

	Level	Skill	Years	Comment
Programming Languages	■■■■■	Rust	10	*Extensive experience with Rust since 2014 (before 1.0). Deep familiarity with language features, tooling, and ecosystem. Contributed to several Rust open-source projects, including the Rust compiler.*
	■■■■□	C++	16	*Extensive experience with C++ from several projects, my bachelor and master theses, and my Ph.D. work. Contributed Itanium ABI-based exception unwinding analysis capabilities to the KLEE symbolic execution engine.*
	■■■■□	C	12	*Extensive experience with C from C++, Linux and program analysis that was focused on C programs.*
	■■■■□	Scala	4	*Deepened knowledge through multiple personal and professional projects, as well as deep general experience with functional programming.*
	■■■□□	HTML&CSS	4	*High experience with HTML&CSS through various personal and professional projects, including working with modern reactive web frameworks.*
	■■■□□	Python	6	*Experience with Python from scripting and automation, as well as development of a meta build system for larger research projects.*
	■■□□□	Java	5	*Experience with Java through my work on the ARCADE project, mostly around GUI development based on the Eclipse framework.*
	■■□□□	Bash	7	*Experience with Bash from general Linux usage, scripting and automation, CI as well as NixOS usage. No large „projects“ in Bash, since I tend to avoid it if possible.*
	■■□□□	Go	1	*Basic experience with Go from personal projects and university work.*

	Level	Skill	Years	Comment
Concepts	■■■■□	Program Analysis	8	*Main focus of my Ph.D. work, with an emphasis on symbolic execution and program testing in general.*
	■■■■□	OOP	12	*Familiarity with OOP concepts from working with different OOP languages, including C++, Python and Java.*
	■■■■□	Async/Await	3	*Familiarity with async-/await-based programming from Rust and experience with low-level network programming.*
	■■■□□	Network Programming	6	*Experience with low-level network programming, such as Ethernet, IPv4/6, UDP, TCP, QUIC and asynchronous IO.*
	■■■□□	GUI Programming	6	*Experience with GUI programming from developing Eclipse- as well as Qt-based applications. Familiarity with concepts such as MVP and event-/observable-based architectures.*
	■■■□□	Graphics Programming	3	*Experience in developing 2D and 3D applications using OpenGL and GLSL, from university courses, my bachelor thesis as well as personal projects.*
	■■■□□	Functional Programming	8	*Experience and familiarity with imperative languages that implement many functional features, such as Rust and Scala. Experience with purely functional programming, for example in Haskell and Lisp.*
	■■■□□	Assembly and IRs	6	*Experience with LLVM IR from my work on symbolic execution. Experience with assembly from personal projects and IT security.*
	■■■□□	Databases (SQL)	3	*Experience with databases through professional work, university courses, personal interest and general programming knowledge.*
	■■□□□	Cryptography&Security		*Understanding of, and experience with, common cryptography techniques such as (asymmetric) encryption, certificates, as well as reverse engineering and malware analysis.*
	■■□□□	Procedural Macros		*Experience with procedural macros from Rust, as well as personal work on lexing and parsing.*

	Level	Skill	Years	Comment
Technologies	■■■■□	Git	12	*Extensive experience with git through years of personal use, university use, and professional use.*
	■■■□□	GitHub	12	*High familiarity with GitHub through personal use and open-source involvement.*
	■■■□□	GitLab	6	*High familiarity with GitLab through professional use including CI.*
	■■■□□	Docker	6	*High familiarity with Docker through professional and personal use, e.g., for intermediate caching during CI.*
	■■■□□	CMake	10	*High familiarity with CMake through work with C++ and C.*
Linux	■■■■□	General	12	*Extensive experience with Linux in general, through personal and professional use, including experience with low-level networking and kernel-level work.*
	■■■□□	NixOS	2.5	*Experience with NixOS through personal use, contribution to NixOS on GitHub, and currently my daily driver.*
	■■■□□	Arch Linux	8	*Experience with Arch Linux through longtime personal use.*