Dr. Marcel Böhme

Last updated: October, 2026

ORCID: 0000-0002-4470-1824
Email: marcel.boehme@acm.org
Website: mboehme.github.io
Research group: boehme-lab.github.io

CISPA C0
Stuhlsatzenhaus 5
66123 Saarbrücken
Germany

Summary

Marcel Böhme leads the Scalable Software Security Laboratory at the CISPA Helmholtz Center for Information Security based in Munich, Germany. His group has made foundational contributions to fuzzing which has become one of the most successful techniques for the automatic discovery of security flaws in practice. The technology developed in his group has been widely deployed in industry where it is continuously squashing security bugs in thousands of security-critical projects on hundreds of thousands of machines. Marcel serves as a Guest Editor-in-Chief and Associate Editor of the ACM TOSEM journal and on the steering committees of ASE (as PC Chair of ASE'25) and ISSTA (as PC Chair of ISSTA'26), two of the largest, premier conferences in his area. He won a 2024 ERC Consolidator grant, a 2019 ARC DECRA (Australia's ERC Starting), a 2019 Google Faculty Research Award, and five ACM Distinguished Paper awards, two research highlights, and one ICLR spotlight at the premier publication venues for security and software engineering. Marcel received his PhD at the National University of Singapore (NUS) in Singapore where he was named a 2024 NUS Outstanding Computing Alumni. He is an ACM Distinguished Member. ▶ *Watch this 2 min video summary*.

Professional Appointments

Sep 2026 – on
Faculty Member (W3)
CISPA Helmholtz Center for Information Security, Germany

Sep 2021 – Aug 2026
Faculty Member (W2)
Max Planck Institute for Security and Privacy, Germany

Mar 2018 – Aug 2025
Adjunct Senior Research Fellow
← **Senior Lecturer**
← **Lecturer**
Faculty of Information Technology
Monash University, Australia

Jul 2015 – Feb 2018
Senior Research Fellow
← **Research Fellow**
School of Computing
National University of Singapore, Singapore

Jul 2014 – Jun 2015
Research Fellow
CISPA and Saarland University
Saarbrücken, Germany

Education

2014 **PhD in Computer Science**, National University of Singapore, Singapore
(GPA 4.9/5.0; Advisor: Abhik Rouchoudhury)

2009 **Dipl.-Inf (BSc & MSc) in Computer Science**, Technische Universität Dresden
(Advisor: Uwe Aßmann; Minor in Robotics and Process Automation)

Awards & Honors

- **ACM Distinguished Member** (2025)
  For contributions in Software Security and Fuzz Testing.
- **ERC Consolidator Award** (2024)
  Highest distinction for a mid-career researcher in Europe.
- **NUS Outstanding Young Computing Alumni Award** (2022)
  Recognition of NUS alumni who have brought honour to the University/School.
- **ARC Discovery Early Career Researcher Award** (2019)
  Highest distinction for an early-career researcher in Australia.

Peer Esteem

- **ACM SIGSOFT Distinguished Paper Award @ ASE** (2026)
- **ACM SIGSOFT Distinguished Paper Award @ ISSTA** (2025)
- **Spotlight Paper @ ICLR** (2025)
  Invited as opinion piece @ IEEE Security and Privacy (Vol. 23; Issue 3).
- **ACM SIGSAC Distinguished Paper Award @ CCS** (2024)
- **CACM Research Highlight** (2023)
  CACM is the monthly journal sent to all members of the ACM.
- **MaxPlanckResearch Spotlight** (2023)
- **Google Open Source Peer Award** (2023)
- **ACM SIGSOFT Research Highlight** (2021)
- **ACM SIGSOFT Distinguished Paper Award @ ICSE** (2020)
- **ACM SIGSOFT Distinguished Paper Award @ FSE** (2020)
- **Monash Dean's Award for Research Excellence (ECR)** (2019)
- **Google Faculty Research Award** (2019)
- **Universität des Saarlandes Best Seminar in Computer Science** (2014/15)
- **NUS Research Achievement Award** (2013)
- **Invited Keynotes @ ICST'26, ISEC'26, SBFT'25, RAID'24**
- **Distinguished Reviewer Awards @ ICSE'25, CCS'23, ICSE'23, ICST'22**

Invitations to Expert Panels and Keynotes

2026 **Keynote @ ICST**: 19th IEEE Int'l Conference on Software Testing, Verification & Validation

Keynote @ ISEC: 19th ACM India Innovations in Software Engineering Conference

Keynote @ Paderborn IT Security Day: 20th Paderborner Tag der IT-Sicherheit

2025 **CASA Summer School** w/ Elissa Redmiles (Georgetown U), Konrad Rieck (TU Berlin), Marco Lorenzi (Inria), Thomas Schneider (TU Dortmund).

Keynote @ SBFT: 18th International Workshop on Search-Based and Fuzz Testing

SBFT Panel of Experts w/ P. Krishnan (Oracle), A. Arrieta (Mondragon U), and A. Crump (CISPA).

2025 **ERC Mock Interview Panel** w/ F. Sarro (UCL), P. Tonella (USI), and C. Cadar (Imperial College).

2024 **Keynote @RAID**: 26th International Symposium on Research in Attacks, Intrusions & Defenses.

NUS Fuzzing Summer School with Abhishek Aarya (Google), Mathias Payer (EPFL), Van-Thuan Pham (Melbourne U), and Andreas Zeller (CISPA).

2023 **EPFL SuRI Summer School** with Henry Corrigan-Gibbs (MIT), Minlan Yu (Harvard), Davide Balzarotti (EURECOM), Alessandra Gorla (IMDEA), Keith Winstein (Stanford) and others.

2023 **It Will Never Work in Theory** with Prem Devanbu (UC Davis), Shurui Zhou (U of Toronto), Alexander Serebrenik (TU Eindhoven), Ariana Mirian (UCSD) and others. Youtube

2022 **Search-based Software Testing (SBST) Panel of Experts** with Andreas Zeller (Saarland/CISPA), Lionel Briand (Luxembourg), Mark Harmann (UCL/Meta), Myra Cohen (Iowa) & P. Tonella (USI).

2022 **University of Zürich IFI Colloqium** with Reid Holmes (UBC) and others.

2021 **Microsoft DS3 Panel on Fuzzing** with Justin Campbell (Principal SE; Fuzzing @ Microsoft) and Kostya Serebryany (Principal SE; Dynamic Tools @ Google) Youtube

2021 **ISSTA Discussion with Experts** with Mathias Payer (EPFL)

2021 **ISSTA Summer School** with Eric Bodden (Paderborn / Fraunhofer IEM), Claire Le Goues (CMU), Satish Chandra (Meta), and Andreas Rossberg (Dfinity) Youtube

2021 **ETH Zürich, Workshop on Dependendable and Secure Software Systems** with Deepak Gark (MPI-SWS), George Candea (EPFL), Ninging Xie (Cambridge), Justin Gottschlich (Intel), Byron Cook (Amazon), Nadia Polikarpova (UCSD), and Anders Miltner (UT Austin)

2020 **FuzzCon Europe** with Caroline Lemieux (UC Berkeley), Christian Holler (Mozilla), Kostya Serebryany (Google), Andreas Zeller (Saarland/CISPA), Rakshith Amarnath (Bosch) and others.

Invited Talks and Guest Lectures

2026 Guest Lecture on *When to Stop Fuzzing* (Live Programming), Ruhr Uni Bochum, Germany; hosted by Flavio Toffalini.
2022 Guest Lecture on *When to Stop Fuzzing* (Live Programming), Saarland Uni / CISPA, Germany; hosted by Andreas Zeller.
2022 Guest Lecture on *Foundations of Software Testing*, U of Oulu, Finnland; hosted by Burak Turhan
2022 Guest Lecture on *The Curious Case of Fuzzing for Automated Software Testing*, RUB, Germany (Tag der Informatik); hosted by the Dean of the Faculty of Computer Science, Alex May.
2022 Invited Talk on the *Reliability of Coverage-Based Fuzzer Benchmarking*, ETH Zürich, Switzerland; hosted by Z. Su.
2022 Invited Talk on the *Reliability of Coverage-Based Fuzzer Benchmarking*, UZH Zürich, Switzerland; hosted by A. Bacchelli.
2021 Guest Lecture on *Academic Career Planning for Early Career Researchers*, NUS, Singapore; hosted by A. Roychoudhury
2021 Guest Lecture on *Foundations of Software Testing*, ISSTA Summer School, Virtual, hosted by Frank Tip (IBM).
2020 Invited Talk on *Software Testing as Species Discovery*, MPI-SWS, Germany; hosted by Gilles Barthe
2018 Invited Talk on *Software Testing as Species Discovery*, COINSE, KAIST, Korea; hosted by Shin Yoo
2018 Invited Talk on *Software Testing as Species Discovery*, TSUNAMi, NUS, Singapore; hosted by A. Roychoudhury
2015 Invited Talk on *On the Efficiency of Automated Testing*, U of Darmstadt, Germany; hosted by M. Pradel
2015 Invited Talk on *On the Efficiency of Automated Testing*, CREST, University College London, UK; hosted by D. Clark
2015 Invited Talk on *On the Efficiency of Automated Testing*, Nanyang Technical University, Singapore; hosted by Liu Yang
2015 Invited Talk on *On the Efficiency of Automated Testing*, Singapore University of Technology & Design; hosted by Sun Jun

Grants & Fellowships

2M EUR	Single-PI, European Research Council. ”In-vivo Security Analysis at Scale”, **Consolidator grant**.
678k AUD (419k EUR)	Single-PI, Australian Research Council. “Fortifying Our Digital Economy: Advanced Automated Vulnerability Discovery”, **Discovery Early Career Researcher Award** (ARC DECRA Fellow).
850k EUR	Co-PI (1/42), Deutsche Forschungsgesellschaft (DFG), ”Cyber Security in the Age of Large-Scale Adversaries (CASA 2.0)”, **DFG Excellence Cluster** (Total: 35M EUR).
571k AUD (353k EUR)	Co-Pi (w/ Prof Abhik Roychoudhury), a Singapore Government Agency, “FuzzInfer: Fuzzing Protocol Implementations”
58k USD (48k EUR)	Single-PI, Google, “Pushing the Coverage Frontier when Fuzzing Progress is Stalled”, **Google Faculty Research Award**
20k EUR	Mentor on the FNR CORE Junior Grant (PI: Dr Ezekiel Soremekun) at the Uni. Luxembourg, “GTDebug: Ground Truth-Based Program Debugging” (Total: 634k EUR)
20k AUD (12k EUR)	PI (w/ Prof Andreas Zeller), DAAD and Universities Australia, “Learning to Discover Security Flaws in Stateful Programs”
2x PhD positions	Co-PI with Konrad Rieck and Kevin Borgolte, CASA Fundamental Research Project, “Testing and Explaining the Limits of Machine Learning For Automated Vulnerability Discovery”

Open Science

I lead my group with an Open Science and Reproducibility Policy, which includes publishing the experimental infrastructure, tools, data, and the scripts to produce tables and figures. Most of the published repositories continue to be actively maintained by the community and receive external contributions. Apart from this group-level policy, I am also actively involved in community efforts to promote preregistration and artifact evaluation as a way to improve the soundness and reprodicibility of our empirical evaluations as well as the soundness of our peer review process.

For instance, I am leading a grassroots initative to introduce a novel preregistration-based publication process for fuzzing research that consists of two main stages: In the first stage, the program committee (PC) evaluates all submissions based on: (i) the significance and novelty of the hypotheses or techniques and (ii) the soundness and reproducibility of the methodology specified to validate the claims or hypotheses—but explicitly not based on the strength of the (preliminary) results. These draft registered reports are presented and improved at the FUZZING’22 workshop. After the workshop, the final versions of the registered reports are re-checked and approved by the PC. In the second stage, the PC and the Artifact Evaluation Committee (AEC) check whether the experimental methodology as laid out by the authors was correctly followed. I am excited that the outcome of this stage have been published in the ACM Transactions on Software Engineering and Methodology (TOSEM) via the Preregistration track (which I was invited to establish initially assisting Cristian Cadar, which, as of the end of 2026, I have now been heading for the past four years as the journal’s Guest Editor-in-Chief). In 2027, I am an inaugural chair of the Registered Reports track at the 20th IEEE International Conference on Software Testing, Verification and Validation (ICST).

Open-source Software

2027 **CTGrep** [S&P'27]
A scalable testing tool for compiler-induced time side channels in implementations of crypto protocols.
Artifact Evaluation Committee evaluated our artifact as **Available**, **Functional**, and **Reusable**.
10.5281/zenodo.22837955

2026 **Cleverest** [FSE'26]
An LLM-based test generator for commits and pull requests for programs like compilers or interpreters.
Artifact Evaluation Committee evaluated our artifact as **Available** and **Functional**.
github.com/niMgnoeSeeL/cleverest

SZZ-Agent and **Simple-SZZ-Agent** [ASE'26]
The most effective tool to date to identify the bug-inducing commit from a bug-fixing commit.
Artifact Evaluation Committee evaluated our artifact as **Available**.
github.com/niklasrisse/agents-for-szz
10.5281/zenodo.19336271

Compass [ASE'26]
A grammar-aware fuzzer which allows to target specific features by disabling parts of the grammar.
10.6084/m9.figshare.31855447

Vital [TOSEM'25]
A fuzzer implementing vulnerability-oriented symbolic execution.
github.com/haoxintu/Vital-SE

Cottontail [S&P'26]
A fuzzer implementing LLM-driven concolic execution for highly structured inputs.
github.com/haoxintu/cottontail

2025 **AFLLive** [ICSE'25]
A fuzzer that can be injected into any program to amplify actual executions.
Artifact Evaluation Committee evaluated our artifact as **Available**, **Functional**, and **Reusable**.
github.com/OctavioGalland/afllive

MendelFuzz [FSE'25]
A fuzzer that uses minimal corruption of the initial seed for more efficieny.
Artifact Evaluation Committee evaluated our artifact as **Available**, **Functional**, and **Reproduced**.
github.com/HexHive/MendelFuzz-Artifact

2024 **ChatAFL** [NDSS'24]
An LLM-based protocol fuzzer based on AFLNet.
10.5281/zenodo.8378945
github.com/ChatAFLndss/ChatAFL

LMTest [CCS'24]
A tool that finds side-channels in crypto implementations running on future microarchitectures.
github.com/hw-sw-contracts/leakage-model-testing

2023 **Rete+PSP and Mythril+PSP** [ASE'23]
Integration of precise data-driven approximation of the valid state space into varios static analyses.
10.5281/zenodo.7902214 (tools, data)

2022 **SGFuzz** [CCS'22]
A stateful greybox fuzzer that keeps track of the explored state space by using state variables and values.
github.com/bajinsheng/SGFuzz

Grammar2Fix [ISSTA'22]
A human-in-the-loop oracle learner for semantic bugs in string processing programs.
github.com/charakageethal/grammar2fix

2021 **AFLChurn** [CCS'21]
A greybox fuzzer for regression bugs that focusses on code that has been changed more often / recently.
www.kaggle.com/marcelbhme/aflchurn-ccs21 (data and analysis scripts)
github.com/aflchurn/aflchurn

2020 **Entropic** [ESEC/FSE'20]
A greybox fuzzer that maxmizes the information about the program's behavior per generated input.
10.6084/m9.figshare.12415622 (tool, data, analysis scripts)
github.com/llvm/llvm-project (Been made the default power schedule in LibFuzzer)
github.com/google/clusterfuzz (Been integrated into OSS-Fuzz and Clusterfuzz, finding security flaws in 500+ open source programs and the Chrome browser)

TimeMachine [ICSE'20-1]
Android Fuzzer that restores the most progressive previously visited state once progress is slow.
10.5281/zenodo.3672076 (tool, benchmarks, experimental infrastructure, data, analysis scripts)
github.com/DroidTest/TimeMachine

HyDiff [ICSE'20-2]
Differential analysis (e.g., to find regressions, side-channels, or adverserial examples).
10.5281/zenodo.3627893 (tool, benchmarks, data, analysis scripts)
github.com/yannicnoller/hydiff

Learn2Fix [ICST'20-1]
A human-in-the-loop automatic program repair tool for semantic bugs in number-processing programs.
github.com/mboehme/learn2fix (Original tool and data)
github.com/charakageethal/learn2fix-journal-ext/ (Journal extension tool and data)

AFLNet [ICST'20-2, TSE'25]
A network-enabled greybox fuzzer that maximizes coverage of a protocol's code and state space.
github.com/aflnet/aflnet

Defect Prediction Guided Fuzzer [ASE'20]
A search-based fuzzer for Java that is guided towards program locations that are more likely defective.
github.com/SBST-DPG

2019 **AFLSmart** [TSE'19]
An input-structure aware greybox fuzzer to fuzz program that take structured inputs.
Reimplemented as interactive book chapter in the Fuzzing Book (Greybox Fuzzing with Grammars)
github.com/aflsmart/aflsmart

2018 **Pythia** [TOSEM'18]
A greybox fuzzer that can estimate the total coverage and the probability of increasing coverage.
Reimplemented as interactive book chapter in the Fuzzing Book (When to Stop Fuzzing)
github.com/mboehme/pythia

2017 **AFLGo** [CCS'17]
A directed greybox fuzzer that can steer the test input generation towards a given set of targets.
Reimplemented as interactive book chapter in the Fuzzing Book (Greybox Fuzzing)
github.com/aflgo/aflgo

DbgBench [ESEC/FSE'17]
A human-annotated benchmark for automated software testing, debugging, and repair.
Poster: How Developers Debug Software The DBGBENCH Dataset
dbgbench.github.io/

2016 **AFLFast** [CCS'16, TSE'18]
A greybox fuzzer that used to outperform the state-of-the-art by an order of magnitude.
Reimplemented as interactive book chapter in the Fuzzing Book (Greybox Fuzzing)
github.com/mboehme/aflfast

2014 **CoREBench** [ISSTA'14]
A benchmark for automated software testing, debugging, and repair of regression errors.
www.comp.nus.edu.sg/ release/corebench/

Open Data and Analysis

2026 **Dependency-aware Residual Risk Analysis** [ICSE'26]
Docker-based setup and scripts, experimental data, analysis scripts, and paper.
Artifact Evaluation Committee evaluated our artifact as **Available** and **Functional**.
10.5281/zenodo.18176114

In Bugs We Trust? On Measuring the Randomness of a Fuzzer Benchmarking Outcome [FSE'26]
Experimental setup, benchmark, data, analysis scripts
github.com/ardier/in_bugs_we_trust/

Incoherence as Oracle-less Measure of Error in LLM-Based Code Generation [AAAI'26]
Experimental setup, incl. tools and benchmarks
github.com/mpi-softsec/difftrust

Scaling Security Testing by Addressing the Reachability Gap [ICSE'26]
Tool, experimental setup, result data
github.com/GPSapia/ReachabilityAgent_ICSE

On Interaction Effects in Greybox Fuzzing [ICSE'26]
Dataset, analysis scripts, figures
github.com/kitsiosk/muofuzz
10.5281/zenodo.17391101

2025 **Top Score on the Wrong Exam Data** [ISSTA'25]
Benchmark datasets, analysis scripts, figures
github.com/niklasrisse/TopScoreWrongExam

Accounting for Missing Events in Statistical Information Leakage Analysis [ICSE'25]
Prototype, experimental setup, and results
Artifact Evaluation Committee evaluated our artifact as **Available** and **Functional**.
github.com/niMgnoeSeeL/ChaoMI

How Much is Unseen Depends Chiefly on Information About the Seen Data [ICLR'25]
All data, scripts, and figures
github.com/https://github.com/niMgnoeSeeL/UnseenGA

Impact of Defect Prediction Quality on Defect-Prediction-guided SBST Data [TOSEM'25]
Prototype, data, and analysis scripts.
10.6084/m9.figshare.16564146

2024 **Limits of ML for Vulnerability Detection Data** [USENIX Sec'24]
Benchmark datasets, analysis scripts, figures
github.com/niklasrisse/LimitsOfML4Vuln
github.com/niklasrisse/USENIX_2024

Coverage Rate Extrapolation Data [ICSE'24]
Data and analysis for extrapolating the coverage rate in a greybox campaign
10.5281/zenodo.10575734 (data and analysis scripts)

2023 **Statistical Reachability Data** [ESEC/FSE'23]
Data and analysis for our statistical analysis.
10.5281/zenodo.7612964 (data and analysis scripts)

Debugging Assumptions Data [ICSE'23]
Data, analysis, and FAQ for reproducing our experiments
https://debugging-assumptions.github.io/

Reachable Coverage Data [ICSE'23]
Data and analysis for estimating when a fuzzing campaign saturates.
10.5281/zenodo.7571359 (data, analysis, figures)

Green Fuzzing Data [ISSTA'23]
Data and analysis for terminating a fuzzing campaign when coverage of defective code saturates.
10.5281/zenodo.7944722 (data, analysis, figures)
github.com/tum-i4/green-fuzzing-artifact

2022 **On the Reliability of Coverage-Based Fuzzer Benchmarking** [ICSE'22]
10.5281/zenodo.6045830 (experimental infrastructure, data, analysis scripts)

2021 **Estimating Residual Risk in Greybox Fuzzing** [ESEC/FSE'21]
Reimplemented as interactive book chapter in the Fuzzing Book (When to Stop Fuzzing)
10.5281/zenodo.4970239 (data, analysis scripts, simulation)

Locating faults with program slicing: an empirical analysis [EMSE'21]
10.6084/m9.figshare.13369400.v1 (data and analysis scripts)

2020 **Fuzzing: On the Exponential Cost of Vulnerability Discovery** [ESEC/FSE'20]
10.6084/m9.figshare.11911287.v1 (data and analysis scripts)

Open Source Textbooks

2021 **The Fuzzing Book**
www.fuzzingbook.org/ (Interactive and executable textbook well-suited for live programming)
github.com/uds-se/fuzzingbook

Academic Service

In computer science, research is disseminated primarily via conferences, not journals. My work on software security is at the intersection of software engineering and cyber security. The flagship venues in Software Engineering are ICSE, ESEC/FSE, ISSTA, and ASE. In cyber security, they are CCS, USENIX Sec, NDSS, and S&P.

Steering Committee Member

2025 – 2031 ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)

2025 – on ACM/IEEE International Conference on Automated Software Engineering (ASE)

2025 – on International Workshop on Fuzzing (FUZZING)

Journal Editorial Board Membership

Since 2022 ACM TOSEM **Guest Editor-In-Chief** for Registered Papers.
The ACM Transactions on Software Engineering and Methodology is the flagship journal for Software Engineering.

Since 2022 ACM TOSEM **Editorial Board Member**.
The ACM Transactions on Software Engineering and Methodology is the ACM flagship journal for Software Engineering.

Conference Organization

2028 ICSE **Area Chair** for Testing and Analysis

2027 FSE New Faculty Mentoring Symposium Chair

ICST Registered Reports Co-Chair

2026 ISSTA **Program Committee Co-Chair**
Chairing 240 program committee members and 13 area chairs (0.9k submissions).

ASE New Faculty Symposium Chair

FUZZING Workshop Co-Chair

2025 ASE **Program Committee Co-Chair**
Chairing 325 program committee members and 20 area chairs (1.2k submissions).

FUZZING Workshop Co-Chair

2024 ICSE **Area Chair** for Dependability and Security.
Chairing area-specific program committee for 1 of 7 areas.

ASE **Area Chair** for Security.
Chairing area-specific program committee for 1 of 7 areas.

ASE Tool Demo Chair.

FUZZING Workshop Co-Chair

2023 ICSE Sustainability Co-Chair

FUZZING Workshop Co-Chair

Dagstuhl on Software Bug Detection: Challenges and Synergies
Co-organized with M. Christakis (MPI-SWS), R. Padhye (CMU), A. Zeller (Saarland/CISPA), and K. Serebryani (Google)

2022 ESEC/FSE Social Media and Publicity Co-Chair

FUZZING Workshop Co-Chair
Introduces a new preregistration-based publication model. Accepted reports will be published in ACM TOSEM.

2021 ECOOP/ISSTA Workshops Co-Chair

2020 ISSTA Doctoral Symposium Co-Chair

2019 NII Shonan: Fuzzing and Symbolic Execution: Challenges, and Opportunities
Results published in IEEE Software: "Fuzzing: Challenges and Reflections" by M. Böhme, C. Cadar, and A. Roychoudhury

2018 ICECCS Workshop Chair

Program Committee Member

2027 ICSE, FSE

2026 ASE

2025 ICSE

2024 ICSE, FSE, ISSTA

2023 CCS, ICSE, ISSTA

2022 CCS, ESEC/FSE, ICSE SEIP, ICST

2021 CCS, ICSE, ISSTA, ASE, ICST

2020 ICSE, ISSTA, ASE, ICST, ICSE-NIER

2019 ICSE, ASE, TAP

2018 ICECCS, SBST, ASWEC, ISSTA AEC, ICSE SRC, ASE Demo, MSR MC

2016 FSE Demo, ISSTA Artifact Evaluation

2015 ICSE NIER (session chair), ISSTA Artifact Evaluation

University / Institute / Community Service

2025 – 2026 **MPRGL Spokesperson** – representing all CPTS research group leads at Max Planck Gesellschaft.

2025 – 2026 Guest of the **MPG CPTS Perspective Commission**
Develops recommendations for the Chemistry, Physics, and Technology Section on all scientific and strategic topics.
8 full-day meetings a year (Administrative HQ in Munich or online).

2025 – 2026 Guest of the **MPG CPTS Section**
3 two-day meetings a year (Harnack Haus in Berlin).

2023 – 2026 **Deputy Ombudsperson** for MPI-SP at Max Planck Gesellschaft

2025 Member of the **SIGSOFT Impact Paper Award Committee** at ACM SIGSOFT

2025 Member of the **CASA Dissertation Awards Committee** at CASA Cluster of Excellence

2022 – 2023 Co-Chair and Member of the **Internship Committee** at CIS Max Planck Gesellschaft

2022 – 2023 Member of 2x **Chair Professor (W3) Recruitment Committees** at RUB

2020 – 2021 **Director of Engagement** in the Executive Committee of the Department of Software Systems and Cybersecurity, Faculty of IT, Monash University, Australia

2019 – 2021 **Inaugural Chair** of Software Engineering Researchers Australasia (SERA)
A working group to build a recognized SE research community (with C. Treude & K. Blincoe)

2019 **Inaugural Engagement Coordinator** (reporting to A/Dean Engagement)

Other Service

2019 – on **Member** of the OpenSSF Linux Foundation [Security Tooling WG], a cross-organizational effort to make OSS and the world's digital infrastructure more secure

2021 – on **PhD Dissertation Panel Member** for
Ezekiel Soremekun (2021 CISPA, DE; Advisor: Andreas Zeller; moved to RHUL, UK),
Dongdong She (2023 Columbia U, US; Advisors: S. Jana, B. Ray; moved to HKUST, HK),
Ahmad Hazimeh (2023 EPFL, CH; Advisor: Mathias Payer; moved to BugScale, CH),
Andrea Fioraldi (2023 EURECOM, FR; Advisor: Davide Balzarotti; moved to Apple, France),
Moritz Schloegel (2024 RUB, DE; Advisor: Thorsten Holz; moved to ASU, US),
David Baker-Effendi (2025 Stellenbosch, SA; Advisors: B. v.d. Merwe, W. Visser, F. Yamaguchi),
Elia Geretto (2025 Vrie University, NL; Advisors: Herbert Bos, Cristiano Giuffrida),
Haoxin Tu (2025 Singapore Management University; Advisor: Lingxiao Zhang; moved to NUS),
Hoang Lam Nguyen (2025 Humboldt University, DE; Advisor: Lars Grunske),
Phillip Görz (2025 RUB, DE; Advisor: Thorsten Holz),
Giacomo Priamo (2026 Sapienza University of Rome, IT; Advisor: Daniele Cono D'Elia),
Johann Glock (2026 Universität Klagenfurt, AT; Advisor: Martin Pinziger),
Ketai Qiu (2027 Università della Svizzera Italiana, CH; Advisors: Mauro Pezzè & Luca Di Grazia),
Samuele Pasini (2027 Università della Svizzera Italiana, CH; Advisor: Paolo Tonella),
Andreas Happe (2027 TU Vienna; AT; Advisor: Jürgen Cito).

2022 – 2024 **Mentor** on Luxembourg FNR CORE Junior grant (PI: Ezekiel Soremekun). CORE Junior projects are piloted by a junior researcher who is supported by a more senior faculty member.

2012 **Focus Group Discussions** with Ministry of Education, Singapore
Nominated and selected to participate in discussions with the MoE about the future of graduate studies in Singapore.

2011 – 2012 **Organizer** of CSTalks, a seminar-style talk series

2010 – 2011 **Graduate Student Representative** @ Graduate Liaison Committee

2009 – 2020 **University Ambassador** @ Technische Universität Dresden, Germany

Software Security Group

Research Fellows and Assistants

2023 – 2025 Dr. Seongmin Lee (Research Fellow @ MPI-SP; CASA Jump.Start Fellow)
(now Prof at POSTECH, Korea)

2022 – 2023 Dr. Behrad Garmany (Research Fellow @ MPI-SP)
(now Co-Founder of BINAMIC Security)

2022 – 2023 Octavio Galland (Research Assistant @ MPI-SP)
(now Security Engineer at Canonical, the company which maintains Ubuntu)

2020 – 2023 Dr. Zahra Mirzamomen (Research Fellow @ Monash University)
(now Lecturer at Monash University)

2020 – 2021 Bharath Krish (Research Assistant, managed collab with Google)
(now Specialist Lead [Manager] at Deloitte)

2018 – 2020 Dr. Van-Thuan Pham (Research Fellow @ Monash University)
(now Prof. at Uni Melbourne; awarded the ARC DECRA)

PhD Students

2025 – on Junming Liu (ERC: MPI-SP→ RUB)

2025 – on Jing Liu (ERC: MPI-SP → RUB)

2025 – on Yasamin Golzar (MPI-SP; main advisor: Yixin Zou)

2024 – on Gaetano Sapio (ERC: MPI-SP → CISPA)

2024 – on Ardi Amadi (MPI-SP → CISPA)

2023 – on Leon Weiß (CISPA / RUB; main advisor: Kevin Borgolte)

2022 – on Tobias Holl (CISPA / RUB; main advisor: Kevin Borgolte)

2022 – 2026 Niklas Risse (MPI-SP / CASA; co advisor: Kevin Borgolte)

2023 – 2025 Philipp Görz (CISPA / RUB; main advisor: Thorsten Holz)

2020 – 2024 Danushka Liyanage (Monash University; co-advisors: Chakkrit Tantithamthavorn)

2019 – 2023 Charaka Geethal (Monash University; co-advisors: Aldeida Aleti, Thuan Pham)

2018 – 2023 Anjana Perera (Monash University; main advisor: Aldeida Aleti)

BSc/MSc Thesis

2026 Annika Kuntze (MSc thesis @ Uni Bonn)

2026 Hannah Freudenberger (MSc thesis @ H-BRS)

2023 Robert Stark (MSc thesis, co-advisor: Veelasha Moonsamy [RUB] and Christopher Huth [Bosch])

2022 Tobias Wienand (BSc thesis, co-advisor: Behrad Garmany [MPI-SP]). Now PhD student at RUB.

Visiting Professors

2025 Prof. Jooyong Yi (UNIST, Korea); 2 weeks

2024 Prof. Abhik Roychoudhury (NUS, Singapore); 1 week

2024 Prof. Eleonora Losiouk (U of Padova, Italy); 4 months

Visiting PhD Students and Undergrad Interns

Candidates were accepted into our competitive summer internship program: www.cis.mpg.de/internships/.

2026 Giorgi Merabishvili (NC State U, USA)

Hannah Freudenberger (H-BRS, DE)

Arjun Sekar (IIT Gandhinagar, IN).

2025 José (Pepe) Zamudio (CISPA, DE)

Thomas Valentin (ENS Paris-Saclay, France)

Ruixiang Qian (Nanjing U, China)

2024 Dr. Dr. Haoxin Tu (SMU, Singapore). Now PostDoc @ NUS, Singapore

Yasamin Golzar (Sharif U, Iran). Now PhD @ MPI-SP, DE

Eric Tang (CMU, US)

Jing Liu (UC Irvine, US). Now PhD @ MPI-SP, DE.

2023 Dr. Abhishek Shah (Columbia U, US). Now at an Online Safety Non-Profit, US.

Rafaila Galanopoulou (U of Athens, Greece). Now PhD @ EPFL, Switzerland.

Shreyas Minocha (Rice University, US). Now PhD @ Georgia Tech, US.

Simran Kathpalia (Amrita U, India). Now MSc @ CISPA, DE.

2022 Dr. Stephan Lipp (TU Munich, DE). Now SWE @ BINAMIC Security, DE.

Dr. Nikhil Parasaram (UCL, UK). Now Software Engineer @ Uber, Netherlands

Octavio Galland (U of Buenos Aires, Argentinia). Now Security Engineer @ Canonical, Argentina

Dongjia Zhang (U of Tokyo, Japan). Now PhD @ EURECOM, France.

Christian Presa Schnell (TU Munich, DE)

Publications

In computer science, research is disseminated primarily via conferences, not journals. My work on software security is at the intersection of software engineering and cyber security. The flagship venues in Software Engineering are ICSE, ESEC/FSE, ISSTA, and ASE. In cyber security, they are CCS, USENIX Sec, NDSS, and S&P.

Authored Books

2020 **The Fuzzing Book**
A. Zeller, R. Gopinath, M. Böhme, G. Fraser, C. Holler.
Interactive textbook. 27 chapters. Printed 1000+ pages.
https://www.fuzzingbook.org

Book Chapters

2013 **Regression Testing of Evolving Programs**
M. Böhme, A. Roychoudhury, B. Oliveira.
In Atif Memon (Ed.), *Advances in Computers* 89, pp. 53–88,
doi:10.1016/B978-0-12-408094-2.00002-3.

Refereed Conference Proceedings (Full Papers)

12x ICSE, 10x ESEC/FSE, 7x ASE, 4x CCS, 4x ISSTA, 2x USENIX Sec, 2x S&P, 1x AAAI, 1x ICLR, 1x NDSS, 1x ICST

S&P'27 **Analyzing Compiler-induced Constant-time Violations with Compiler at Scale**
J. Liu, L. M. Gavier, Z. Zhang, M. Böhme, and G. Barthe
Practical and scalable detection of compiler-induced side channels using the compiler itself.

ASE'26 **LLM-based Agents for Identifying Bug-Introducing Commits**
N. Risse and M. Böhme.
First paper to substantially improve on SZZ in 20 years using an agentic approach with grep skills.
IEEE/ACM International Conference on Automated Software Engineering (ASE'26). 13 pages.
Received the **ACM SIGSOFT Distinguished Paper Award** (Top 8% of accepted papers).
doi:10.1145/3832783.3834337

ASE'26 **Feature-Focused Test Generation**
J.A.Z. Amaya, G. Sapia, A. Turcotte, D. Benavides, M. Böhme, and A. Zeller.
Marries software variability & grammar-based fuzzing. Configure grammars to focus on features.
IEEE/ACM International Conference on Automated Software Engineering (ASE'26). 13 pages.
doi:10.1145/3832783.383747

ASE'26 **Accurate and Low-Cost Residual Risk Assessment via Sampled Profiling and Structure-Aware Coverage Amplification**
S. Lee, I. Özgü, M. Böhme, M. Kim.
Explores the utility of no-overhead, hardware-based coverage profiling for residual risk estimation.
IEEE/ACM International Conference on Automated Software Engineering (ASE'26). 13 pages.
doi:10.1145/3832783.3837519

FSE'26 **In Bugs We Trust? On Measuring the Randomness of a Fuzzer Benchmarking Outcome**
A. Madadi, S. Lee, C. Aschermann, and M. Böhme.
First paper to establish concordance of a benchmarking method to measure the "randomness" of the benchmark outcome.
ACM Symposium on the Foundations of Software Engineering 2026 (FSE'26), 21 pages.
doi:10.1145/3797112

AAAI'26 **Incoherence as Oracle-less Measure of Error in LLM-Based Code Generation**
T. Valentin, A. Madadi, G. Sapia, and M. Böhme.
How to estimate the correctness of an LLM-generated program when we have no specification or ground-truth available.
Annual AAAI Conference on Artificial Intelligence (AAAI'26). 14pp.
doi:10.1609/aaai.v40i39.40616

ICSE'26 **Scaling Security Testing by Addressing the Reachability Gap**
G. Sapia and M. Böhme.
How to configure and interact with any software system to execute a given target functionality (and run invivo fuzzing)?
IEEE/ACM International Conference on Software Engineering (ICSE'26). 12pp.
doi:10.1145/3744916.3773116

ICSE'26 **Dependency-aware Residual Risk Analysis**
S. Lee and M. Böhme.
First work to account for dependencies among coverage elements in residual risk estimation.
IEEE/ACM International Conference on Software Engineering (ICSE'26). 12pp.
doi:10.1145/3744916.377318

ICSE'26 **On Interaction Effects in Greybox Fuzzing**
K. Kitsios, M. Böhme, and A. Bacchelli.
First work to identify and leverage interaction effects between mutation operators in fuzzing.
IEEE/ACM International Conference on Software Engineering (ICSE'26). 12pp.
doi:10.1145/3744916.3773128

S&P'26 **Cottontail: LLM-Driven Concolic Execution for Highly Structured Test Input Generation**
H. Tu, S. Lee, Y. Li, P. Chen, L. Jiang, and M. Böhme.
First work to introduce grammar-awareness to whitebox fuzzing via LLMs.
IEEE Symposium on Security and Privacy (SP'26). 18pp.

ISSTA'25 **Top Score on the Wrong Exam: On Benchmarking in ML for Vulnerability Detection**
N. Risse, J. Liu, and M. Böhme.
Renders almost all existing work as invalid. ML4VD as binary classification problem is ill-defined.
ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'25), 22 pages
Received the **ACM SIGSOFT Distinguished Paper Award** (Top 8% of accepted papers).
doi:10.1145/3728887

ICLR'25 **How Much is Unseen Depends Chiefly on Information About the Seen**
S. Lee and M. Böhme.
Significant progress on a beautiful statistical riddle. Can estimate data representativeness.
International Conference on Learning Representations (ICLR'25), 22 pages.
Selected as **ICLR'25 Spotlight** (Top 5% of accepted papers).

FSE'25 **MendelFuzz: The Return of the Deterministic Stage**
H. Zheng, F. Toffalini, M. Böhme, and M. Payer.
Ground-breaking work that re-introduced the deterministic stage which was disabled by default.
ACM Symposium on the Foundations of Software Engineering 2025 (FSE'25), 21 pages.
Adopted as **default mode** **in the most widely-used fuzzer** AFL++ since v4.10c.
doi:10.1145/3715713

ICSE'25 **Invivo Fuzzing by Amplifying Actual Executions**
O. Galland, and M. Böhme.
First work to suggest injecting fuzzers into arbitrary systems to amplify any state.
IEEE/ACM International Conference on Software Engineering 2025 (ICSE'25), 13 pages.
doi:10.1109/ICSE55347.2025.00172

ICSE'25 **Accounting for Missing Events in Statistical Information Leakage Analysis**
S. Lee, S. Minocha, and M. Böhme.
How to estimate software privacy (information leakage) in the small sample regime?
IEEE/ACM International Conference on Software Engineering 2025 (ICSE'25), 13 pages.
doi:10.1109/ICSE55347.2025.00018

USENIX Sec'24 **Uncovering the Limits of Machine Learning for Automatic Vulnerability Detection**
N. Risse and M. Böhme.
Are machine learning models for vulnerability discovery as good as they seem? How to find out?
USENIX Symposium on Security (USENIX Sec'24), 19 pages.

CCS'24 **Testing Side-Channel Security of Cryptographic Impl. Against Future Microarchitectures**
G. Barthe, M. Böhme, S. Cauligi, C. Chuengsatiansup, D. Genkin, M. Guarnieri, D.M. Romero, P. Schwabe, D. Wu, Y. Yarom.
First comprehensive analysis of the security impact of future microarchitectures.
ACM Conference on Computer and Communications Security (CCS'24), 16 pages.
Received the **ACM SIGSAC Distinguished Paper Award**.
doi:10.1145/3658644.367031

NDSS'24 **Large Language Model guided Protocol Fuzzing**
R. Meng, M. Mirchev, M. Böhme, and A. Roychoudhury.
Let the fuzzer ask ChatGPT about the correct structure / order of messages as specified in the RFC.
Network and Distributed System Security Symposium (NDSS'24), 18 pages.
Number 24 in the **Normalized Top-100 Security Papers** of all time.
doi:10.14722/ndss.2024.24556

ICSE'24 **Extrapolating Coverage Rate in Greybox Fuzzing**
D. Liyanage, S. Lee, C. Tantithamthavorn, and M. Böhme.
*First work to *predict* the coverage rate of a greybox fuzzer in the future.*
IEEE/ACM International Conference on Software Engineering 2024 (ICSE'24), 13 pages.
doi:10.1145/3597503.3639198

ESEC/FSE'23 **Statistical Reachability Analysis**
S. Lee and M. Böhme.
First quantiative program analysis using a statistical rather than an analytical approach.
ACM Symposium on the Foundations of Software Engineering (ESEC/FSE'23)
doi:10.1145/3611643.3616268

ASE'23 **Precise Data-Driven Approximation for Program Analysis via Fuzzing**
N. Parasaram, E. T. Barr, S. Mechtaev, and M. Böhme.
First work to approximate the valid program state space from above and from below.
IEEE/ACM International Conference on Automated Software Engineering (ASE'23). 12 pages
doi:10.1109/ASE56229.2023.00185

ICSE'23 **Reachable Coverage: Estimating Saturation in Fuzzing**
D. Liyanage, M. Böhme, C. Tantithamthavorn, and S. Lipp.
First work to solve the undecidable reachability problem as a statistical estimation problem.
ACM/IEEE International Conference on Software Engineering (ICSE'23). 13 pages
doi:10.1109/ICSE48619.2023.00042

ICSE'23 **Evaluating the Impact of Experimental Assumptions in Automated Fault Localization**
E. Soremekun, L. Kirschner, M. Böhme, and M. Papadakis.
Important work studying assumptions that researchers make during debugging tool evaluations.
ACM/IEEE International Conference on Software Engineering (ICSE'23). 13 pages
doi:10.1109/ICSE48619.2023.00025

ISSTA'23 **Green Fuzzing: A Saturation-based Stopping Criterion using Vulnerability Prediction**
S. Lipp, D. Elsner, S. Kacianka, A. Pretschner, M. Böhme, and S. Banescu.
We suggest to stop a fuzzing campaign when the coverage of potentially vulnerable code saturates.
ACM/SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'23), 13 pages
doi:10.1145/3597926.3598043

ICSE'22 **On the Reliability of Coverage-Based Fuzzer Benchmarking**
M. Böhme, L. Szekeres, J. Metzman
First paper to study agreement instead of correlation between coverage and bug finding.
IEEE International Conference on Software Engineering 2022 (ICSE'22), 11 pages
doi:10.1145/3510003.3510230

USENIX SEC'22 **Stateful Greybox Fuzzing**
J. Ba, M. Böhme, Z. Mirzamomen, A. Roychoudhury.
Navigating an unknown state space by identifying and monitoring state variables values.
USENIX Security Symposium (USENIX SEC) 2022, pp. 3255-3272
isbn: 978-1-939133-31-1

ISSTA'22 **Human-in-the-loop Oracle Learning for Semantic Bugs in String Processing Programs**
C. Geethal, V.T. Pham, A. Aleti, M. Böhme.
Learning to identify semantic bugs for string processing programs
ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'22), 12 pages
doi:10.1145/3533767.3534406

ESEC/FSE'21 **Estimating Residual Risk in Greybox Fuzzing**
M. Böhme, D. Liyanage, and V. Wüstholz.
First paper to study estimators of bug finding probability (residual risk) when no bugs were found.
ACM Symposium on the Foundations of Software Engineering (ESEC/FSE) 2021, 12 pages
doi:10.1145/3468264.3468570
- J. Campbell (Microsoft) would use this in OneFuzz to maximize bug finding per unit time.
- First work my PhD student, Danushka Liyanage. Very proud!

CCS'21 **Regression Greybox Fuzzing**
X. Zhu and M. Böhme
Finds and explains the constant rate of new bug reports, despite the exponential cost in fuzzing.
ACM Conference on Computer and Communications Security (CCS'21), pp. 2169–2182
doi:10.1145/3460120.3484596

ESEC/FSE'20 **Boosting Fuzzer Efficiency: An Information Theoretic Perspective**
M. Böhme, V. J.M. Manès, S.K. Cha.
First work to show and leverage a connection between fuzzer efficiency and information theory.
ACM Symposium on the Foundations of Software Engineering (ESEC/FSE'20), pp. 970-981
doi:10.1145/3368089.3409748
Independent evaluation available on https://www.fuzzbench.com!
Entropic is the default schedule in LibFuzzer powering Google's OSSFuzz & MS's OneFuzz!
Received the **ACM SIGSOFT Distinguished Paper Award** (1x Accept, 2x Award Quality).
Selected as **ACM SIGSOFT Research Highlight**.
Selected as **CACM Research Highlight** (accompanied by a Technical Perspective: "What's all the fuss about fuzzing?" by Prof Gordon Fraser).

ESEC/FSE'20 **Fuzzing: On the Exponential Cost of Vulnerability Discovery**
M. Böhme, B. Falk.
First work to shed light on the scalability of fuzzing across a large number of available machines.
ACM Symposium on the Foundations of Software Engineering (ESEC/FSE'20), pp. 747-758
doi:10.1145/3368089.3409729
Nominated for **ACM SIGSOFT Distinguished Paper Award** (2x Accept, 1x Award Quality).

ICSE'20 **Time-Travel Testing for Android Apps**
Z. Dong. M. Böhme, L. Cojocaru, A. Roychoudhury.
First work to tackle the problem of statefulness when testing Android testing by user interactions.
ACM/IEEE International Conference on Software Engineering (ICSE'20), pp. 481-492
doi:10.1145/3377811.3380402
This quick animation captures nicely the key idea of time-travel testing.
Received the **Outstanding Prototype Award** on ChinaSoft2020.
Received the **ACM SIGSOFT Distinguished Paper Award** (2x Accept, 1x Award Quality).

ICSE'20 **HyDiff: Hybrid Differential Software Analysis**
Y. Noller, C. Pasareanu, M. Böhme, Y. Sun, H. Nguyen, and L. Grunske.
First work to introduce a hybrid approach to differential software analysis (incl. RNNs).
ACM/IEEE International Conference on Software Engineering (ICSE'20), pp. 1273-1285
doi:10.1145/3377811.3380363

ICST'20 **Human-In-The-Loop Automatic Program Repair**
M. Böhme, C. Gheetal, V.T. Pham.
First work to achieve automated repair of semantic bugs starting from just one failing test case.
IEEE International Conference on Software Testing, Verification and Validation (ICST'20), pp. 274-285, doi:10.1109/ICST46399.2020.00036
First work by my first main-advised PhD student, Charaka Gheetal. Very proud!
Featured in the IEEE Software Practitioner's Digest.

ASE'20 **Defect Prediction Guided Search-Based Software Testing**
A. Perera, A. Aleti, M. Böhme, and B. Turhan.
First work to integrate defect prediction and automated software testing.
IEEE/ACM International Conference on Automated Software Engineering (ASE'20), pp. 448–460
doi:10.1145/3324884.3416612
ℹ First work by my first co-advised PhD student, Anjana Perera. Very proud!

ESEC/FSE'18 **Verifying the Long-Run Behavior of Probabilistic System Models in the Presence of Uncertainty**
Y.R.S. Llerena, M. Böhme, M. Bruenink, G. Su, and D.S. Rosenblum.
First work to tackle modelling uncertainties in probabilistic verification of steady-state properties.
ACM Symposium on the Foundations of Software Engineering (ESEC/FSE'18), pp. 587–597.
doi:10.1145/3236024.3236078

CCS'17 **Directed Greybox Fuzzing**
M. Böhme, V.-T. Pham, M.-D. Nguyen, A. Roychoudhury.
First fuzzing work to cast reachability as an optimization rather than a satisfiability problem.
ACM SIGSAC Conference on Computer and Communications Security (CCS'17), pages 2329–2344
doi:10.1145/3133956.3134020
ℹ Invited to present to industrial partners of the Singapore Cyber Security Consortium.

ESEC/FSE'17 **Where is the Bug and How is it Fixed? An Experiment with Practitioners**
M. Böhme, E.O. Soremekun, S. Chattopadhyay, E. Ugherughe, and A. Zeller.
First large-scale user study to establish the ground-truth for automatic debugging & repair.
ACM Symposium on the Foundations of Software Engineering (ESEC/FSE'17), pages 117-128
doi:10.1145/3106237.3106255

ASE'17 **Detecting Information Flow by Mutating Input Data**
B. Matthis, V. Avdiienko, E. Soremekun, M. Böhme, A. Zeller.
First work to introduce lightweight input probing to detect information flows using fuzzing.
IEEE/ACM International Conference on Automated Software Engineering (ASE'17), pp. 263–273
doi:ASE.2017.8115639
ℹ Result of my first BSc student, Björn Mathis. Very proud!

CCS'16 **Coverage-based Greybox Fuzzing as Markov Chain**
M. Böhme, V.-T. Pham, A. Roychoudhury.
First work to introduce systematic path exploration to random testing, using adaptive sampling.
ACM SIGSAC Conference on Computer and Communications Security (CCS'16), pp. 1032–1043
doi:10.1145/2976749.2978428
ℹ The journal extension appeared at the SE flagship journal IEEE TSE in 2018.

ASE'16 **Model-based Whitebox Fuzzing for Program Binaries**
V.-T. Pham, M. Böhme, A. Roychoudhury.
First work to integrate structure-ware fuzzing and symbolic execution.
IEEE/ACM International Conference on Automated Software Engineering (ASE'16), pp. 543–553

ESEC/FSE'14 **On the Efficiency of Automated Testing**
M. Böhme, S. Paul.
First work to shed light on the unusual efficiency of random testing vs the most effective technique. Solves a riddle that has been baffling researchers for more than thirty years (Duran et al., TSE'84).
ACM Symposium on the Foundations of Software Engineering (ESEC/FSE'14), pp. 632–642
doi:10.1145/2635868.2635923
ℹ The journal extension appeared at the SE flagship journal IEEE TSE in 2016.

ISSTA'14 **CoREBench: Studying Complexity of Regression Errors**
M. Böhme, A. Roychoudhury.
ACM International Symposium on Software Testing and Analysis (ISSTA'14), pp. 398–408
doi:10.1145/2610384.2628058
Ranked among Top10 most downloaded articles on ACM DL (SIGSOFT) from Aug'14 to Oct'14.

ESEC/FSE'13 **Regression Tests to Expose Change Interaction Errors**
M. Böhme, B. Oliveira, A. Roychoudhury.
First work to shed light on the unusual efficiency of random testing vs the most effective technique.
ACM Symposium on the Foundations of Software Engineering (ESEC/FSE'13), pp. 339–349
doi:10.1145/2491411.2491430

ICSE'13 **Partition-based Regression Verification**
M. Böhme, B. Oliveira, A. Roychoudhury.
First work to introduce a scalable, interruptible verification of the absence of regression bugs.
ACM/IEEE International Conference on Software Engineering (ICSE) 2013, pp. 300–309
doi:10.1109/ICSE.2013.6606576

Refereed Journal Articles

6x IEEE TSE, 5x ACM TOSEM, 2x EMSE, 1x CACM, 1x IEEE Security & Privacy (S&P), 1x IEEE Software

TOSEM 2026 **Vital: Vulnerability-Oriented Symbolic Execution via Type-Unsafe Pointer-Guided Monte Carlo Tree Search**
H. Tu, L. Jiang, and M. Böhme.
How to guide symbolic execution towards the most vulnerable parts of the execution tree.
ACM Transactions on Software Engineering and Methodology, pp. 1-25

IEEE S&P 2025 **How to Solve Cybersecurity Once and For All**
M. Böhme.
Elicits fundamental challenges in cybersecurity and discusses what to do about them.
IEEE Security and Privacy, Volume 23, Issue 3, pp. 1-4, doi:10.1109/MSEC.2025.3551590
Invited opinion piece. Elaborates my invited RAID'24 keynote.

TOSEM 2025 **Software Security Analysis in 2030 and Beyond: A Research Roadmap**
M. Böhme, E. Bodden, T. Bultan, C. Cadar, Y. Liu, and G. Scanniello.
Challenges and opportunities for the security analysis of our software systems of the future.
ACM Transactions on Software Engineering and Methodology, pp. 1-25, doi:10.1145/3708533.
Invited perspective paper.

TOSEM 2025 **Fuzzing: On Benchmarking Outcome as a Function of Benchmark Properties**
D. Wolff, M. Böhme, A. Roychoudhury.
How would fuzzer ranking change if programs were larger or initial seeds had more coverage?
ACM Transactions on Software Engineering and Methodology, pp. 1-24

TSE 2025 **AFLNet Five Years Later: On Coverage-Guided Protocol Fuzzing**
R. Meng, V.T. Pham, M. Böhme, A. Roychoudhury.
State- and code-coverage-guided greybox fuzzing.
IEEE Transactions on Software Engineering, pp. 1-14, doi:10.1109/TSE.2025.3535925.
Extension of our ICST'20 Tool Demo.

TSE 2024 **Human-In-The-Loop Automatic Program Repair**
C. Geethal Kapugama, M. Böhme, and V.-T. Pham.
First to fully automate the debugging process end-to-end by interrogating the bug-reporting user.
IEEE Transactions on Software Engineering, pp. 1-24, doi:10.1109/TSE.2023.3305052.
Extension of our ICST'20 work.

TOSEM 2024 **On the Impact of Lower Recall and Precision in Defect Prediction for Guiding Search-Based Software Testing**, A. Perera, B. Turhan, A. Aleti, and M. Böhme, ACM Transactions on Software Engineering and Methodology.

CACM 2023 **Boosting Fuzzer Efficiency: An Information Theoretic Perspective**
M. Böhme, V. Manès, and S.K. Cha
First work to show and leverage a connection between fuzzer efficiency and information theory.
Communications of the ACM (Volume: 66; Issue: 11), November 2023, pp 89-97
doi:10.1145/3611019
CACM is the monthly journal sent to all members of the ACM.
CACM Research Highlight for the month of November 2023 from our ESEC/FSE 2017 paper.
CACM Technical Perspective: "What's all the fuss about fuzzing?" by Gordon Fraser.

TSE 2022 **An Experimental Assessment of Using Theoretical Defect Predictors to Guide Search-Based Software Testing**, A. Perera, A. Aleti, B. Turhan, M. Böhme, IEEE Transactions on Software Engineering, pp. 1-18, doi:10.1109/TSE.2022.3147008

Software 2021 **Fuzzing: Challenges and Reflections**
M. Böhme, C. Cadar, A. Roychoudhury.
Establishes the key open challenges in fuzzing and symbolic execution as a community result.
IEEE Software Journal. Accepted for publication, pp. 1–9, doi:10.1109/MS.2020.3016773.
Outcome of a 3-day meeting of thought leaders & rising stars, both in industry and academia.

EMSE 2021 **Locating Faults with Program Slicing: An Empirical Analysis**
E. Soremekun, L.Kirschner, M. Böhme, A. Zeller.
Empirical Software Engineering Journal, pp. 1–49
doi:10.1007/s10664-020-09931-7 (Paper)
doi:10.6084/m9.figshare.13369400.v1 (Dataset)

TSE 2020 **Smart Greybox Fuzzing**
V.T. Pham, M. Böhme, A.E. Santosa, A.R. Căciulescu, and A. Roychoudhury.
Makes greybox fuzzing aware of input structure, handles corrupted inputs, and maximizes validity.
IEEE Transactions on Software Engineering (Volume: 47, Issue: 9), pp. 1980-1997
doi:10.1109/TSE.2019.2941681
In the news at Security Week, The Register, and Nacked Security.

TOSEM 2018 **STADS: Software Testing as Species Discovery**
M. Böhme.
First work to establish the statistical foundations of automated software testing. Borrows from ecological biostatistics to estimate and extrapolate quantities in software testing.
ACM Transactions on Software Engineering and Methodology (Volume: 27, Issue: 2), pp. 1–52
doi:10.1145/3210309
Selected as journal-first contribution presented at ESEC/FSE 2018.
Reimplemented as interactive book chapter in the Fuzzing Book (When to Stop Fuzzing?)

TSE 2018 **Coverage-based Greybox Fuzzing as Markov Chain**

M. Böhme, V.-T. Pham, A. Roychoudhury.

First work to introduce systematic path exploration to random testing, using adaptive sampling.

IEEE Transactions on Software Engineering (Volume: 45, Issue: 5), pp. 489-506 doi:TSE.2017.2785841

- AFLFast evaluated by the community finds 6 bugs in Perl and several bugs in Erlang VM.
- AFLFast was featured on Hackernews, the most popular news website for security experts.
- Main ideas integrated into AFL by Google's Director of Information Security, Michał Zalewski.
- Google Security awards USD 2000 in bug bounties for vulnerabilities we reported.

EMSE 2017 **A Correlation Study between Automated Program Repair and Test-Suite Metrics**

J. Yi, S.H. Tan, S. Mechtaev, M. Böhme, and A. Roychoudhury.

Empirical Software Engineering Journal (Special Issue on Automated Program Repair), 23(5), pp. 2948–2979. doi:10.1007/s10664-017-9552-y

- Established test suite metrics are good predictors of the quality of auto-generated repairs.
- Selected as Journal-First contribution presented at ICSE 2018.

TSE 2016 **A Probabilistic Analysis of the Efficiency of Automated Software Testing**

M. Böhme and S. Paul

First work to shed light on the unusual efficiency of random testing vs the most effective technique. Solves a riddle that has been baffling researchers for more than thirty years (Duran et al., TSE'84).

IEEE Transactions on Software Engineering, 42(4), pp. 345–360. doi:10.1109/TSE.2015.2487274

- Invited talk at the CREST Open Workshop, University College London (UCL), United Kingdom.
- Invited talk at the Singapore University of Technology and Design (SUTD) in Singapore.
- Invited talk at the Nanyang Technical University (NTU) in Singapore.
- Invited talk at the Technische Universitaet Darmstadt in Germany.
- Invited talk at the Center for It-Security, Privacy and Accountability (CISPA) in Germany.
- Ranked among Top-50 most popular IEEE TSE articles for six months (Apr'16–Sep'16).[1]

Other Refereed Publication Outputs

5x ICSE short papers (2x New Ideas, 1x Doctoral Symposium, 1x Poster), 1x ASE NIER and 1x ICST Tool paper.

ASE'26 NIER **Empirical Computation: Prompting versus Programming**

Eric Tang, Jing Liu, and M. Böhme

Vision paper positions empirical computation as an interesting new topic in software engineering.

41st IEEE/ACM International Conference on Automated Software Engineering (NIER track) (ASE'26 NIER), 5 pages

ICSE'22 NIER **Statistical Reasoning about Programs**

M. Böhme

Vision paper to challenge traditional program analysis and opportunities for new perspectives.

IEEE International Conference on Software Engineering 2022: New Ideas and Emerging Results (ICSE'22 NIER), 5 pages, doi:10.1145/3510455.3512796

ICST'20 Testing Tool **AFLNet: A Greybox Fuzzer for Network Protocols**

V.T. Pham, M. Böhme, Abhik Roychoudhury

IEEE International Conference on Software Testing, Verification and Validation 2020 (ICST'20), Testing Tool Track, pp. 60-465, doi:10.1109/ICST46399.2020.00062 Youtube

[1] Issue April'16, Issue May'16, Issue June'16, Issue July'16, Issue August'16, Issue September'16.

ICSE'19 NIER **Assurance in Software Testing: A Roadmap**
M. Böhme
IEEE International Conference on Software Engineering 2019: New Ideas and Emerging Results (ICSE'19 NIER), pp. 5-8, doi:10.1109/ICSE-NIER.2019.00010 Slideshare

ICSE'17 Poster **How developers debug software the DbgBench dataset: poster**
M. Böhme, E.O. Soremekun, S. Chattopadhyay, E. Ugherughe, A. Zeller
ACM/IEEE International Conference on Software Engineering 2017: Poster (ICSE'17 Poster), pp. 244-246, doi:10.1109/ICSE-C.2017.94

ICSE'12 Doctoral Symposium **Software Regression as Change of Input Partitioning**
M. Böhme, ACM/IEEE International Conference on Software Engineering 2012: Doctoral Symposium (ICSE'12 DS), pp. 1523-1526, doi:10.1109/ICSE.2012.6227046

Teaching

While I have always enjoyed teaching, I have been a *research-only staff* for most of my academic training and career. At NUS, it was uncommon for PhD and PostDocs to teach. At Monash, I received a prestigious ARC DECRA award which comes with an automatic teaching relieve for three years. At MPI-SP, I do not have any teaching obligations.

2026 Guest Lecturer
• 211067 Advanced Automatic Testing @ RUB (Germany) on *When to Stop Fuzzing*; hosted by Prof Flavio Toffalini.

2025 Lecturer
• Seminar on Automated Software Security @ RUB (Germany).

2024 Lecturer
• One-day course on the Foundations of Software Security @ Singapore Summer School.

2023 Lecturer
• 212125 Seminar Software and Internet Security @ RUB (Germany) with Prof Kevin Borgolte

2022 Guest Lecturer
• 1819 Generating Software Tests @ Saarland Uni / CISPA (Germany) on *When to Stop Fuzzing*; hosted by Andreas Zeller.
• 811602S Advanced Software Quality and Security @ Uni. of Oulu (Finnland) on the *Foundations of Software Testing*; hosted by Burak Turhan

2021 Guest Lecturer
• CS6210 The Art of CS Research @ NUS (Singapore) on *Career Planning for Academics*; hosted by A. Roychoudhury.
• Summer School @ ISSTA conference on the *Foundations of Software Testing*; hosted by Frank Tip (IBM).

2020 Guest Lecturer
• FIT2093 Introduction to Cyber Security @ Monash University (Australia) on *Software Security*; hosted by Ron Steinfeld

S1 2020 FIT3077 Software Architecture and Design @ Monash University
Chief-Examiner

S1 2019 FIT3077 Software Architecture and Design @ Monash University
Co-Lecturer

S1 2018 FIT3077 Software Architecture and Design @ Monash University
Co-Lecturer

S2 2015/16 CS4218 Software Testing @ National University of Singapore
Co-lecturer and Lab Coordinator. Prepared slides and taught in the second half of the semester. Coordinated two teaching assistants and prepared material.

S2 2014/15 Seminar on Automated Debugging @ Saarland University
Lecturer. Organized as reading group. *Voted Best Seminar*. Voted Top-3 among *all* courses in Computer Science.

S2 2013/14 CS4218 Software Testing and Debugging @ National University of Singapore
Weekly exercises, labs, tutorials, and grading. I was involved in designing the schedule and details for exercises, labs, and tutorials. This was the first time the course was provided.

S1 2010 CS3215 Software Engineering Project @ National University of Singapore
Consultations, lab, and grading.

Practical Impact

Our tools found many security-critical vulnerabilities in widely used open-source projects and libraries, such as php (4), valgrind, gdb, coreutils (13), binutils (56), libiberty (8), libdwarf (7), libxml2 (4), ffmpeg (10), wavepac (4), libming, and libav, and Live555 Media Server. Our tools were discussed in the news (Security Week, The Register, Nacked Security, Hackernews), and by the coreutils package maintainer Pádraig Brady. Google Security awarded USD 2,000 for hardening of security-critical OSS libraries. Most vulnerabilities were detected and analyzed during experiments of Van-Thuan Pham and myself. Our most recent fuzzer, Entropic, was integrated into Google's LibFuzzer and now runs on more than 100,000 machines every day to discover security vulnerabilities in the Chrome browser and more than 1,000 open source projects.

We published 82 security advisories (CVEs), a considerable number even for security professionals:
CVE-2016-2226, CVE-2016-4487, CVE-2016-4488, CVE-2016-4489, CVE-2016-4490, CVE-2016-4491, CVE-2016-4492, CVE-2016-4493, CVE-2016-6131, CVE-2017-6965, CVE-2017-6966, CVE-2017-6969, CVE-2017-7209, CVE-2017-7210, CVE-2017-7223, CVE-2017-7224, CVE-2017-7225, CVE-2017-7226, CVE-2017-7227, CVE-2017-7299, CVE-2017-7300, CVE-2017-7301, CVE-2017-7302, CVE-2017-7303, CVE-2017-7304, CVE-2017-7578, CVE-2017-8392, CVE-2017-8393, CVE-2017-8394, CVE-2017-8395, CVE-2017-8396, CVE-2017-8397, CVE-2017-8398, CVE-2017-9047, CVE-2017-9048, CVE-2017-9049, CVE-2017-9050, CVE-2017-9051, CVE-2017-9052, CVE-2017-9053, CVE-2017-9054, CVE-2017-9055, CVE-2018-10372, CVE-2018-10373, CVE-2018-10536, CVE-2018-10537, CVE-2018-10538, CVE-2018-10539, CVE-2018-10540, CVE-2018-12458, CVE-2018-12459, CVE-2018-12460, CVE-2018-13300, CVE-2018-13301, CVE-2018-13302, CVE-2018-13303, CVE-2018-13304, CVE-2018-13305, CVE-2018-13785, CVE-2018-19539, CVE-2018-19540, CVE-2018-19541, CVE-2018-19542, CVE-2018-19543, CVE-2018-19543, CVE-2019-7314, CVE-2019-15232, CVE-2021-38380, CVE-2021-38381, CVE-2021-38382, CVE-2021-38383, CVE-2021-39282, CVE-2021-39283, CVE-2021-41396, CVE-2021-41397, CVE-2021-41687, CVE-2021-41688, CVE-2021-41689, CVE-2021-41690, CVE-2023-0215, CVE-2023-37117, CVE-2023-51713.