Yonghao Zou

Email: yonghaoz1994@gmail.com Mob: (41) 76 276 93 46 Homepage: zouyonghao.github.io

RESEARCH INTEREST

Program Analysis, Symbolic Execution, Verification, Fuzzing, Operating Systems, Distributed Systems

EDUCATION

EPFL (Advisor: Prof. George Candea). *Sep. 2023-Now*
Research Assistant
Research Topics: Program Verification, Memory Analysis, Operating Systems
Honor & Award: EDIC Fellowship

Tsinghua University (Advisor: Prof. Shi-Min Hu & Jia-Ju Bai) *Sep. 2019-Jun. 2022*
M.E. in Computer Technology
Main Courses: Computer System Performance Measurement, Advanced Operating Systems
Honor & Award: Outstanding Graduate, Tsinghua University

Zhejiang University *Sep. 2011-Jun. 2015*
B.E. in Computer Science and Technology

PUBLICATION

- **Yonghao Zou**, Jia-Ju Bai, Zu-Ming Jiang, Ming Zhao, Diyu Zhou. Blackbox Fuzzing of Distributed Systems with Multi-Dimensional Inputs and Symmetry-Based Feedback Pruning. NDSS, 2025.
- Can Cebeci, **Yonghao Zou**, Diyu Zhou, George Candea, Clément Pit-Claudel. Practical Verification of System-Software Components Written in Standard C. SOSP, 2024.
- Kai-Tao Xie, Jia-Ju Bai, **Yong-Hao Zou**, Yu-Ping Wang. ROZZ: Property-based Fuzzing for Robotic Programs in ROS. ICRA, 2022.
- **Yong-Hao Zou**, Jia-Ju Bai, Jielong Zhou, Jianfeng Tan, Chenggang Qin, Shi-Min Hu. TCP-Fuzz: Detecting Memory and Semantic Bugs in TCP Stacks with Fuzzing. USENIX ATC, 2021.
- **Yong-Hao Zou**, Jia-Ju Bai. Effective Crash Recovery of Robot Software Programs in ROS. ICRA, 2021.

EXPERIENCE & RESEARCH

Work Experience: Software backend engineer Shanghai, China
Software architecture group member in China Merchants Bank *Jul. 2015-Apr. 2018*

1. Develop a microservice framework, including configuration management systems, RPC framework, service management and discovery, and code generation tools.
2. Develop a message system for scale and availability, providing critical functions including at least once message delivery, high availability servers, server load balance, and manual message management.

Research on Robot Operation System (ROS) reliability Beijing, China
Project core member *Sep. 2019-Mar. 2020, May. 2021-Jul. 2021*

1. Develop a new lightweight recovery tool named RORY with checkpoint and message replay to recover ROS nodes effectively. RORY can recover six standard ROS programs in both virtual and realistic environments. (ICRA 20)
2. Develop a ROS fuzzing tool named ROZZ with three essential techniques, including a multi-dimensional generation method, a distributed branch coverage, and a temporal mutation strategy, to test ROS nodes effectively. ROZZ has successfully found 43 actual bugs on ten standard robotic programs in ROS 2. (ICRA 22)

Research on reliability and correctness testing of TCP Stacks Beijing, China
Project leader. *Apr. 2020-Feb. 2021*

1. Develop a novel fuzzing framework, TCP-Fuzz, to effectively test TCP stacks and detect bugs using three essential techniques: a dependency-based generation strategy, a transition-guided fuzzing approach, and a differential checker. (ATC 21)
2. TCP-Fuzz is adapted to 5 TCP stacks and finds 56 bugs.

Research on verification tool systems and specifications written in C. in Lausanne, Switzerland
Project core member *Sep. 2023-Mar. 2024*

1. Propose the use of the parallel CI pipeline to verify systems in parallel to improve the efficiency of verification.
2. Improve the Klee-based verifier and verify the correctness of Komodo with specifications written in C. (SOSP 24)

Research on reliability and correctness testing of distributed systems in Beijing, China
Project leader *Mar. 2022-Now*

1. Propose a coverage-guided fuzzing approach using fault injection and different checkers to test distributed systems.
2. Develop a novel fuzzing framework that has found dozens of bugs in distributed systems. The framework can also find bugs in distributed relational databases after further extension. (NDSS 2025)

SKILLS

- System programming, including Linux driver development, scheduling, networking, distributed systems, and ROS
- Static and dynamic analysis based on LLVM
- Symbolic execution and verification tools: Klee, Z3
- Programming language: Java, C/C++, Python, SQL and Clojure

TEACHING

- CS-311 The software enterprise - from ideas to products (TA)