Yuyang(Peter) RONG

530 601 3646 ◇ PeterRong96@gmail.com ◇ Website ◇ GitHub

EDUCATION

UC Davis Sep 2019 - Jun 2024

Ph.D. candidate in Computer Science *Davis, CA*

- Research interests: **Compiler optimization and fuzzing, LLM for fuzzing**
- Skills: **C/C++ (9/10), Rust (8/10), Python (8/10)**
- Community impacts: **LLVM contributor. Bug fixes in backends, e.g. X86, AArch64, etc.**
- Leadership & project management: **Manage a six-person team working on two projects simultaneously**

ShanghaiTech University Sep 2015 - Jun 2019

B.E. Computer Science and Technology *Shanghai, China*

- GPA 3.79/4 (Rank: 5/124) Excellent Undergraduate of Shanghai (2019) Scholarship of Shanghai (2016)

EXPERIENCE

Advanced Micro Devices, Inc. (AMD) Jul 2023 - Present

Research Intern *San Jose, CA*

- Focused on aeveloping a scheduling algorithm based on reinforcement learning for AI Engine (AIE).
- Preliminary results shows the new scheduling algorithm can reduce 5% cycles in the generated assembly code.
- Implementing GitHub Action script for weekly fuzzing of AIE, cooperate with DevOps to deploy the fuzzer.

Advanced Micro Devices, Inc. (AMD) Jun 2022 - Dec 2022

Research Intern *San Jose, CA*

- Focused on testing compiler backend (CodeGen) of AI Engine (AIE).
- Implemented IRFuzzer in 2000 LoC C++ to accommodate for the compiler infrastructure.
- Found over 40 missing features in AIE. Found **74 confirmed bugs in LLVM, 44 fixed**, bug tracker.
- Lightning talk accepted to 2022 LLVM Developer's Meeting, recording.

Bytedance Ltd. Jun 2020 - Sep 2020

Research Intern *Mountain View, CA*

- Focused on optimizing fuzzer Angora's gradient solver and alleviate its branch collision problem.
- Implemented a fuzzer Valkyrie with a runtime in ~2000 LoC in C++ and a gradient solver in ~3000 LoC in Rust.
- Found six bugs in open-source libraries, improved branch coverage by 41% compared to Angora.

Bytedance Ltd. Sep 2018 - Aug 2019

Research Intern *Beijing, China*

- Assigned to maintain Angora and use it to find integer bugs in Bytedance's codebase.
- Implemented a sanitizer as an LLVM pass w/ runtime library using ~1500 LoC in C++ and ~2000 LoC in Rust.
- Identified **8 crashing and 166 non-crashing bugs**. CVE-2020-18869 and CVE-2020-18871 assigned.

SELECTED PUBLICATIONS

Code Representation Pre-training with Complements from Program Executions 2024

The International Conference on Learning Representations (ICLR) *Under peer review*

IrFuzzer: Specialized Fuzzing for LLVM Backend Code Generation 2024

International Conference on the Foundations of Software Engineering (FSE) *Under peer review*

Understanding Programs by Exploiting Fuzzing Test Cases 2023

Association for Computational Linguistics (ACL)

Valkyrie: Improving Fuzzing Performance Through Principled Techniques 2022

Software Quality, Reliability, and Security (QRS) ***(Best paper award)***

An Inexact First-order Method for Constrained Nonlinear Optimization 2022

Optimization Methods and Software

IntEgrity: Finding Integer Errors by Targeted Fuzzing 2020

Security and Privacy in Communication Networks (SecureComm)

TEACHING

ECS153: Computer Security ***(Best TA award)*** Spring 2023

ECS032A: Introduction to Programming Fall 2020