Yuyang(Peter) RONG

(+1)530 · 601 · 3646 ◇ PeterRong96@gmail.com ◇ https://peterrong.netlify.app ◇ https://github.com/DataCorrupted

EDUCATION

UC Davis Sep 2019 - Jun 2024

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

- Research interests: **Fuzzing, Program Analysis, Software Security, Compiler Testing**
- Languages: **C/C++ (9/10), Rust (8/10), Python (8/10), Java**
- Tools: **LLVM (LLVM conributor w/ 23 commits and 2000+ LoC), libFuzzer, AFL++, Docker**

ShanghaiTech University Sep 2015 - Jun 2019

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

- GPA 3.79/4 (Ranking: 5/124)
- Excellent Undergraduate of Shanghai (Jun 2019) Scholarship of Academic Excellence (Nov 2017)

EXPERIENCE

Advanced Micro Devices (AMD) Jun 2022 - Dec 2022

Research Intern *San Jose, CA*

- Focused on testing compiler backend of AI Engine.
- Implemented a state-of-the-art fuzzer to accommodate for the compiler infrastructure, open-sourced.
- Found over 40 missing features in AI Engine. Found **58 confirmed bugs in LLVM, 32 fixed**, bug tracker.
- Lightning talk accepted to 2022 LLVM Developer's Meeting, recording.

Bytedance Jun 2020 - Sep 2020

Research Intern *Mountain View, CA*

- Focused on optimizing fuzzer Angora's gradient solver and alleviate branch collision problem.
- Implemented an LLVM pass in ~2000 lines of C++ and a new gradient solver in ~3000 lines of Rust, open-sourced.
- Improved branch coverage by 41% compared by Angora, 94% compare to AFL++.

Bytedance Sep 2018 - Aug 2019

Research Intern *Beijing, China*

- Assigned to find integer errors using Angora in Bytedance's codebase.
- Designed and implemented a sanitizer as an LLVM pass w/ runtime library using ~1500 lines C++ and ~2000 lines of Rust, maintained Angora ever since.
- Identified **8 crashing errors** that could cause deinal of service attack, CVE-2020-18869 and CVE-2020-18871 assigned; found 166 non-crashing errors that could cause program misbehave, reported to developers.

SELECTED PUBLICATIONS

IrFuzzer: Specialized Fuzzing for LLVM Backend Code Generation 2024

International Conference on Software Engineering (ICSE) *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 2023 Spring Quater

ECS032A: Introduction to Programming 2020 Fall Quater

CS110: Computer Architecture I 2018 Spring Semester