Gabriel Sherman

Salt Lake City, UT | gabesherman6@gmail.com | +1 207-307-1490

linkedin.com/in/gabe-sherman | github.com/gabe-sherman

Research Interests

My research focuses on fuzz testing, with a strong passion for proactively securing widely used software. I am particularly interested in extending fuzz testing to previously unexplored areas. My current work centers on automating harness generation for open-source libraries. Some of the bugs I have found can be found here: futures.cs.utah.edu/bugs/?search=gabe+sherman

Education

Ph.D., Computer Science	University of Utah	2024-*current*
B.S., Computer Science	University of Utah	2020-2024

Publications

1. **No Harness, No Problem: Oracle-guided Harnessing for Auto-generating C API Fuzzing Harnesses**
   Gabriel Sherman, Stefan Nagy
   International Conference on Software Engineering **(ICSE '25)**

Experience

Ph.D. Research Assistant, University of Utah – Salt Lake City, UT — August 2024 – *current*

- Currently conducting research under Dr. Stefan Nagy to expand the capabilities of automatic harness generation through testing, development, and evaluation of lab-developed approaches.
- Authored and published a paper detailing an automated harness generation approach for C-based libraries.
- Performed extensive harnessing and fuzzing of software libraries using AFL++.
- Performed detailed triage on crashes seen during fuzzing trials to determine the root cause of the error.
- Uncovered and reported over 50 bugs in software harnessed by our developed approaches, with over 40 confirmed.

Summer Research Intern, Trail of Bits – Salt Lake City, UT — May 2024 – August 2024

- Advanced the development of an innovative automatic harness generation approach for C-based libraries. Enhanced its functionality to support more complex library harnessing routines and optimized the overall efficiency of the method.
- Integrated Trail of Bits' code indexing tool, Multiplier, to perform static analysis of the library under test. This static analysis involved parsing the abstract syntax tree of the build artifact to extract relevant information about the library.
- Created harnesses for several widely-used open-source libraries. Performed fuzzing and bug triage, identifying and reporting 6 confirmed bugs to the respective library developers.
- Engaged in a dedicated and technical environment focused on cybersecurity.
- Presented my summer research work at Empire Hacking in New York City.

Invited Talks & Articles

Introduction to Fuzzing – University of Utah Cybersecurity Club.	03/2025
Automated Bug Finding – Guest Lecture at Kahlert School of Computing.	09/2024
Automated Harness Generation – Mountain West Undergraduate Research Showcase.	11/2023

Technical Skills

Build Systems: Make, CMake, Autotools, Ninja, Meson

Debugging and Profiling: GDB, perf, strace, ASan/UBSan

Software testing: Fuzzing, Static & Dynamic Testing, Crash Triage

Languages: Python, C, C++, Bash/Shell, Java

Projects & Activities

Market Simulation Game

- Designed and implemented a browser-based market simulation game where users buy and sell products to maximize profits.
- Developed the backend using Spring Boot and MongoDB, exposing core functionalities through a REST API for seamless integration.
- Built a responsive and interactive frontend using React, leveraging the REST API to display real-time market data in a user-friendly interface.

University of Utah CTF Team

- Competed in cybersecurity competitions, solving challenges in binary exploitation, digital forensics, reverse engineering, and web security.
- Specialized in identifying and exploiting vulnerabilities in binaries and analyzing digital artifacts for forensic investigations.
- Collaborated with team members to develop creative solutions and improve overall performance in competitions.

Operating System Fundamentals

- Built an early prototype of a kernel based on the xv6 architecture. This prototype included booting, interrupt handling, and running processes in user space.
- Implemented a reduced linker program. Parsed the ELF file to perform relocation on all entries in the relocation table.