

# Gilbert Hoermann

www.seal9055.com | seal9055@gmail.com | <https://github.com/seal9055> | [linkedin.com/in/gilberthoerm](https://www.linkedin.com/in/gilberthoerm)

## Experience

### Interrupt Labs Vulnerability Researcher

07/2023 - Current

- Chrome Javascript engine
  - Setup onboarding information on internal wiki pages and started documenting engine internals
  - Adapted the team's previous work on finding bugs in Safari's regular expression engine to Chrome and used it to find bugs in Chrome's regular expression engine. This involved custom patches to Chrome and fuzzer fine-tuning based on coverage reports and performance statistics
  - Performed root cause analysis and crash triaging for the found crashes using manual analysis and automated debugging scripts
  - Developed proof of concept exploits to reliably trigger the bugs and demonstrate their impact
  - Expanded on fuzzer-analysis tooling to more reliably generate post-fuzz-campaign reports
  - Performed manual code audits on Chrome's just-in-time compiler and V8 builtins in Google's custom Torque programming language
  - Spent several months writing a fully custom and configurable javascript fuzzer based on abstract syntax tree mutations while scaling up to hundreds of CPU cores that found bugs in V8's ignition bytecode interpreter
- Chrome Browser Sandbox
  - Audited various sandbox internals such as Moji-IPC interfaces
  - Wrote variant analysis scripts using Codeql and Weggli to perform more automated analysis
  - Found bugs on this target by writing and setting up fuzzers targeting mojo interfaces
  - Audited Webgpu interfaces accessible from V8
- Client Contract - Router
  - Worked on reverse engineering target software and auditing the code for security bugs while contributing to the team's target wiki/knowledge base and helping interns/junior members
  - Harnessed target using custom snapshot fuzzer and emulation to target pre-auth attack surface that found several segmentation faults reproducing on the target
  - Wrote scripts using binary ninja and gdb's scripting API to automate research tasks

### Trail of Bits Vulnerability Research Internship

01/2023 - 02/2023

- Designed and implemented an efficient coverage guided graybox snapshot fuzzer on top of Qemu
- The fuzzer was able to leverage an Android live ram-dump to find bugs on an old device

### UMass Cybersecurity Lecturer - <https://umasscybersec.org/cs390r.html>

01/2022 - 05/2023

- Teaching a course on reverse engineering and advanced vulnerability analysis to a class of 55 students
- Topics include re/code auditing, fuzzing, stack/heap exploitation, kernel-mode security, and automated dynamic/static software analysis w/ llvm, decompiler scripting, taint analysis, pin, and time-travel dbg

### Technical Director at UMass Cybersecurity Club

11/2021 - 05/2023

- Hosting workshops focused on low level security topics such as binary exploitation and fuzzing
- Created virtualized enterprise network as training grounds for CPTC pentesting competition

## Projects

### Fuzzing Research using Emulation/JIT - <https://github.com/seal9055/sfuzz>

09/2021 - 07/2022

- Wrote an emulation-based graybox fuzzer focused on performance, code coverage and scaling
- The emulator + custom JIT enable high levels of target introspection without requiring source code
- The fuzzer includes coverage guided seed selection, byte level permission checks, snapshot fuzzing, memory allocation hooks and linear scaling across cores without a source requirement

### Wyze Camera - CVE-2021-43726 & CVE-2021-43727

09/2022 - 11/2022

- Reverse engineered and emulated (using Qemu) device's firmware to find and exploit 2 critical bugs
- Rce via format string bug and remote image/video download by attackers via path traversal

**Officejet Pro 6835** - [https://github.com/seal9055/officejet\\_pro\\_6835](https://github.com/seal9055/officejet_pro_6835) 02/2023 - 05/2023

- Project to find rce in popular printer. Extracting firmware through multiple layers of non-standard compression/encryptions that binwalk could not handle
- Reverse engineered firmware to understand data parsing routines (PCL protocol)

## Skills

---

- Reverse engineering, static analysis (binary ninja, llvm), and dynamic analysis (pin, gdb, triton, unicorn)
- Exploit dev, writing n-day exploits for userland, kernel (Linux), and browser (Chrome/Safari) bugs
- Bug discovery techniques, fuzzer harnessing, dataflow-analysis, patch diffing and variant analysis
- Understanding of modern exploit mitigations (ASLR, SMAP, HVCI, Sandboxing, ...) and bypass strategies
- Systems programming in the context of writing high-performance emulators, compilers and hypervisors
- Embedded VR, extracting firmware from flash, reversing decompression, emulation/fuzzing setup
- C, C++, Rust, Javascript, Python, Scala, Assembly (X86, Arm, Mips, RISC-V)
- Ability to quickly tackle and get familiar with unfamiliar large codebases and binaries

## Education

---

**University of Massachusetts Amherst** 08/2020 - 05/2023

- Graduated with BS in computer science with a focus on systems and security - GPA 3.7
- Coursework: Cyber-effects, Computer Architecture, Operating Systems, Compilers

## Certifications & Achievements

---

**OSCP** - January 2021

**Sans Foundations** - June 2021

**Ret2 Systems Binary Exploitation Course** - [wargames.ret2.systems/course](https://wargames.ret2.systems/course)

03/2021

**Hypervisor Development for Security Researchers Course** - <https://tandasat.github.io/>

08/2022

**Ret2 Browser Exploitation Course** - <https://ret2.io/trainings>

05/2024

**Ctftime Top 7 US** - Binary exploitation player for k3rn3l4rmy