Andrew Haberlandt

(937) 344-1676
andrew.haberlandt@gmail.com

Education

Ohio State University, *B.S. Computer Science & Engineering with Honors, minor in Mathematics* **Columbus, OH**
Expected May 2022

I will enter a Computer Science PhD program in Fall 2022. GPA: 3.977

TA/Grader

- Graded projects, held regular office hours, and answered student lab questions for the following courses, over 6 semesters:
    - **CSE 2421** (4 semesters) - systems programming and computer organization, including **C** and **x86 assembly** Jan/20 - Present
    - **CSE 2221** (2 semesters) - introductory software design in Java Jan/19 - Dec/19

Clubs / Activities

- **Undergraduate Research (see below)** Aug/20 - Present
- **Cybersecurity Club @ Ohio State (co-lead)** - Run meetings, develop CTF-style challenges, create workshops and presentations, organize BuckeyeCTF competition. Aug/18 - Present
- **Code 4 Community** - create and deliver computer science workshops for K-12 students Jan/20 - May/21
- **Collegiate Cyber Defense Competition** Oct/19 - Feb/20
- **Engineer's Council, Hall Council**

Research Experience

Advised by Dr. Zhiqiang Lin, Ohio State University

(in progress) Identifying Repeated Code in Binaries for Decompilation Jan/21 - Present

- Working with the Angr binary analysis framework, I'm developing (source-less) techniques to identify repeated (e.g. macro-like) code in binaries. These techniques support the identification of application-specific patterns in real-world binaries, including expression substitutions within these patterns. This work will improve reverse-engineering tools by simplifying decompiler output and will also aid other automated program analyses. We also plan to evaluate the impact of these simplifications on manual reverse-engineering through a human study. This work is planned to be submitted to USENIX Security in February 2022.

(in progress) Fuzzing Intel SGX Programs Aug/20 - Present

- Although programs running in Intel's SGX secure enclave cannot be inspected by normal means, page faults can still be observed by the host operating system. I developed a prototype using Intel PIN to collect a page-fault trace of program execution, and modified the AFL fuzzer to use this page-fault-based trace to guide fuzzing. I then modified the Linux kernel to force page faults on (nearly) every memory access, and to collect the same type of page-fault trace as the PIN prototype. I evaluated this fuzzer on programs from the LAVA-M dataset.

Work Experience

Apple, *Software Engineering Intern* **Cupertino, CA (remote)**
May/21 - Aug/21

- Developed a dynamic analysis tool that discovered 10+ security-critical bugs.

Caesar Creek Software, *Software Engineering / Reverse Engineering Intern* **Springboro, OH**
May/20 - Aug/20
May/19 - Aug/19

- Utilized static and dynamic analysis tools (Ghidra, GDB, Frida) to tackle real-world reverse engineering challenges.
- Researched a popular, off-the-shelf IoT device (ARM): vulnerability research, developed fully-remote multi-stage exploit (mem. corruption to RCE), engineered stealth implant (in C) for remote access via Android app. (2020)
- Designed new and improved existing automated tools in C and Python for distributed vulnerability discovery, making significant modifications to the KVM hypervisor and QEMU. (2019)

Ohio State University (CSE Department), *Grader/TA and Research Assistant* **Columbus, OH**
Jan/19 - Present

- Graded projects, held regular office hours, and answered student lab questions (see above)
- Research in binary analysis to improve decompilation, and in fuzzing Intel SGX programs (see above)

Air Force Research Laboratory (WPAFB), *Research Intern* **Dayton, OH**
May/18 - Aug/18
Jun/17 - Aug/17

- Developed a visualization in Python for a modular AI platform
- Designed and implemented data processing and visualization utilities in Python and Javascript for large datasets of location-tagged imagery.

Side-Projects

More information at https://andrewh.tech

CTF Challenges and Infrastructure for BuckeyeCTF

- Developed a variety of reverse-engineering, binary exploitation, and web exploitation challenges. My challenges (written in C, C++, x86 assembly, and ARM assembly) have incorporated ROP, heap exploitation, custom virtual machine architectures, dynamic binary instrumentation, and more.
- Designed and implemented infrastructure on Amazon Web Services using Terraform (infrastructure-as-code) for securely running vulnerable containerized (Docker) services.

"Grades for Students" iOS App

- Developed an iOS app in Objective-C for students at my high school to track their grades. It is still used by over 700 students daily as of Fall 2021.

Code 4 Community Projects

- Code 4 Community is a student organization at Ohio State which designs computer science workshops for middle and high school students. I designed a web-based game (in Javascript) targeted at middle-school students which helps them learn the fundamentals of computer science.

Open Source Contributions

- I have regularly contributed back to open-source software, including Canvas (LMS used by many large universities), MediaWiki, and more.

CTF (cybersecurity competitions)

Individual Awards

- CSAW CTF Finals Qualifier (one of 60)

Team Awards

- 3rd in redpwnCTF 2021
- 4th in b01lersCTF 2021
- 7th in UTCTF 2021
- 6th in DAMCTF 2021

Selected CTF write-ups are available at https://github.com/ndrewh/ctf

Other Awards

- Google Code-in 2015 Winner (one of ~24)
- Apple WWDC 2015 Scholarship Recipient (one of ~200)
- Bug bounty ($500) for a information disclosure vulnerability in a platform with 10M daily users (2020)
- CSE Department Scholarship (one of ~25) for 2021-2022 ($2000)
- Ohio State University “Maximus” Scholarship ($4000/yr)