

Matthew Cole

✉ mcole8@binghamton.edu

☎ +1 (206) 790-8791

🏢 Department of Computer Science

Thomas J. Watson College of Engineering and Applied Science

4400 Vestal Parkway East

Binghamton, NY 13902-6000

Curriculum Vitae

Education

- | | |
|---------------------|--|
| May 2024 (expected) | Ph.D. Computer Science , State University of New York at Binghamton University, Binghamton, New York.
Dissertation: <i>Enforcing Integrity Models Through Hardware-Software Cohesive Systems</i> |
| August 2018 | M.Sc. Computer Science , State University of New York at Binghamton University, Binghamton, New York.
Thesis: <i>Integrity Models</i> |
| May 2005 | B.Sc. Computer Science , United States Naval Academy, Annapolis, Maryland.
Graduated <i>With Merit</i> , Upsilon Pi Epsilon |

Dissertation

Advisor: Aravind Prakash

Integrity models are a principled defense mechanism that express a property of well-functioning software, then enforce that property continually at runtime. Unfortunately, these integrity models are often implemented in a way that compartmentalizes hardware from software. We present work that unifies these in a single cohesive view. First, we show that existing hardware resources can be repurposed to support software-layer defenses without onerous impacts to performance. Then, we present a modified LLVM compiler toolchain used to enforce a diverse body of integrity models through code and data tagging. Next, we investigate how to optimize a label-based integrity model to minimize binary size increases while maximizing expressiveness of the integrity model. Finally, we propose using tagging to enforce a read- and write-limited data model, thus bringing forward compile-time data type qualifiers as run-time assurances.

Experience

Research Experience

- | | |
|-----------|--|
| 2018-2022 | Research Assistant Research Foundation for the State University of New York. Binghamton, New York |
|-----------|--|
- Repurposed Intel Memory Protection Extensions for generalized storage, and implemented analyses for the LLVM compiler toolchain to replace memory accesses with these inlined register operations.
 - Engineered an implementation of the RISC-V architecture employing inline code for integrity models from compiler-driven static program analysis techniques, using the LLVM compiler toolchain.
 - Extended our implementation of the RISC-V architecture and LLVM compiler toolchain to perform out-of-band data tagging, with the ability to perform relaxation and linkage resolution at link-time using a modified version of the LLD linker.
 - Developed a prototype extending data tagging for a read- and write-limited data model that specifically enforces compile-time `const` qualifiers as run-time assurances.

Experience (continued)

2015-2016 **Graduate Assistant** Binghamton University. Binghamton, New York

- Developed a technique for extracting design patterns from C++ source code and encoding as a finite state machine with an XML machine-readable representation
- Researched scientific utilization of performance benchmark tools for the computer security domain.

Research Interests

Hardware-software cohesive design for resilient system security through compiler design, program analysis, and reverse engineering.

Teaching Experience

2022-Present **Lecturer** Binghamton University. Binghamton, New York

- Fall 2023 – CS 580U: Programming Systems and Tools
- Fall 2022 – CS 580U: Programming Systems and Tools
- Spring 2022 – CS 458, CS 558: Introduction to Computer Security

2017-2018 **Teaching Assistant** Binghamton University. Binghamton, New York

- Spring 2023 – CS 458, CS 558: Introduction to Computer Security
- Spring 2018 – CS 480, CS 580: Special Topics: Software Security
- Fall 2017 – CS 220: Computer Systems II, Architecture and C Programming

Courses Prepared to Teach

Systems programming in C, Rust, and Assembly Languages; Compilers: lexers, parsers, optimizers, and code generation. Computer security: authentication and cryptographic techniques, intrusion detection, access control, security policies, reverse engineering, offensive and defensive technologies in software security; Programming systems and tools with C, C++ or Python.

Professional Experience

Service

Professional Service

2021 ACSAC Artifact Committee <https://www.acsac.org/2021/committees/artifact/>
 2020 ACSAC Artifact Committee <https://www.acsac.org/2020/committees/artifact/>
 2017 ACSAC Artifact Committee <https://www.acsac.org/2017/committees/#artifact>

University Service

2018-2019 Binghamton University Graduate Student Organization Judicial Officer
 2016-2018 Binghamton University Graduate Student Organization Senate

Skills

Languages	C, C++, Python, x86/RISC-V/ARM assembly, LLVM IR, Rust, Java
Operating Systems	Unix/Linux (Debian/Ubuntu/Solaris), MacOS
Benchmarking	SPEC CPU 2006/2017, Hayai, Google Benchmark
Testing	CUnit, Google Test, Boost.Test, Python unittest, LLVM Lit

Skills (continued)

Build/Deploy	GNU Make, CMake, Github Actions, Travis-CI, Git
Reverse Engineering	Ghidra, Hopper

Miscellaneous