CAE Lecture Series

21 August 2025

(1 pm EST)

Dr. Carter Yagemann, The Ohio State University, Beyond Memory Safety: Expanding the Scope of Automatic Software Vulnerability Analysis

Mark your calendars and come join your colleagues in the CAE community for the CAE Lecture Series!

CAE Lecture Series are free and conducted live in real-time over MS Teams so no travel is required. NSA's CAE PMO office hosts the presentations via MS Teams which employs slides, VOIP, and chat for live interaction. Just click on the link and enjoy the presentation(s).

Bio: Carter Yagemann is an Assistant Professor of Computer Science and Engineering at The Ohio State University, where he teaches and conducts research in systems security. He earned his Ph.D. in Computer Science from the Georgia Institute of Technology and both his B.S. and M.S. from Syracuse University. Before academia, he worked at JPMorgan Chase in ethical hacking and cyber-threat intelligence. Carter's research interests are in systems and software security, including vulnerability discovery, exploit prevention, fault injection, cyber-physical systems, and financial market security. His broader research interests include malware analysis, machine learning, and agent-based simulation.

Abstract 1 pm EST: Memory safety violations—such as out-of-bounds accesses and use-after-free errors—have long been a primary vector for software exploits, enabling attackers to compromise systems. In response, the software industry is undergoing a broad transition toward memory-safe programming languages like Rust, Go, and Python, which eliminate entire classes of such vulnerabilities by construction. However, memory safety alone is not a panacea. Even in memory-safe environments, serious security threats persist, including logic bugs and resource exhaustion attacks that can cripple critical services, as demonstrated by recent exploits targeting the Go standard library.

This talk opens with a review of the software vulnerability lifecycle and established program analysis techniques for uncovering memory safety issues in legacy code. It then argues for the continued relevance of these techniques in the context of memory-safe languages, highlighting recent findings that exposed and led to the remediation of logic vulnerabilities in Go. The presentation concludes with a discussion of emerging challenges and research opportunities in securing the next generation of memory-safe software.

MS Teams Information:

Microsoft Teams Need help?

Join the meeting now

Meeting ID: 270 506 632 671 3

Passcode: zc9ck9VZ

Dial in by phone

+1 872-239-6004,,924963532# United States, Chicago

Find a local number

Phone conference ID: 924 963 532#

For organizers: Meeting options | Reset dial-in PIN

Note: This Lecture series cannot be recorded/posted online, we encourage you to attend live.