PASCHAL C. AMUSUO

(765)-746-9312 pamusuo@purdue.edu West Lafayette, Indiana https://ampaschal.github.io

RESEARCH STATEMENT

A software security expert specializing in applying program analysis, formal methods, and artificial intelligence to proactively secure software systems. My PhD research created new systematic vulnerability discovery methods, and developed techniques to guide and support artificial intelligence agents to automate these methods. My work has produced 8 publications, uncovered 37 security vulnerabilities in widely-used software, and earned a Qualcomm Innovation Fellowship.

EDUCATION

Purdue University, IN, USA *2021 - 2026*
Ph.D. Candidate, Electrical and Computer Engineering *(CGPA: 3.97/4.00)*
Qualcomm Innovation Fellow, 2025 – 2026
Magoon Excellence in Teaching Award, 2022

Purdue University, IN, USA *2021 - 2024*
Master of Science, Electrical and Computer Engineering *(CGPA: 3.98/4.00)*

Federal University of Technology, Owerri, Nigeria *2013 - 2018*
Bachelor of Engineering, Electrical and Electronic Engineering *(CGPA: 4.88/5.00)*
Best Graduating Engineering Student (Out of >1000 Engineering Students)

RESEARCH EXPERIENCE

PhD SWE Intern - *Google Open Source Security Team, Google* *May 2025 - Aug 2025*
Project: Agentic AI Systems for Fuzzing and Vulnerability Discovery
- Designed an LLM-based agent to autonomously reason about, debug, and validate the security impact of fuzzing crashes.
- Developed methods to automatically infer and use implicit function expectations to improve AI-generated fuzz driver quality.
- Integrated agents in OSS-Fuzz-Gen, agentic AI system for fuzz driver generation, and reduced false positive crashes by 64%.

Graduate Research Assistant – *Duality Lab, advised by Prof. James C. Davis* *Jan 2022 – May 2026 (Expected)*
Project: Automated Unit Proof Generation for Code-level Security Verification
- Designed AutoUP, multi-agent system for unit proof generation, precondition refinement, and memory safety verification.
- Implemented sub-agents to detect and autonomously fix build, timeout, and coverage issues during unit proof generation.
- Developed program-analysis-based tools that enable LLM agents efficiently access verification coverage and error reports.

Project: Systematic Unit Proof Development for Compositional Bounded Model Checking
- Designed a systematic methodology to develop environment models and verification bounds for Bounded Model Checking.
- Evaluated the effectiveness and cost of Bounded Model Checking for discovering vulnerabilities in embedded software.
- Verified components of four embedded operating systems using Bounded Model Checking and discovered 20 vulnerabilities.

Project: Systematic Dynamic Testing of Embedded Software for Vulnerability Discovery
- Studied vulnerabilities in embedded network stacks to identify the input packet structure and sequences that trigger them.
- Designed a systematic state exploration and packet mutation algorithm to expose vulnerabilities in network stacks.
- Implemented a dynamic analysis framework to validate embedded network stacks and uncovered 7 new vulnerabilities.

Engineering Intern - *Qualcomm Product Security Initiative, Qualcomm* *May 2024 - Aug 2024*
Project: A Static Analysis Tool for Vulnerability Discovery in Qualcomm Drivers
- Studied and characterized known vulnerabilities in Qualcomm drivers to identify root causes and develop detection strategies.
- Developed *IAAFinder*, an LLVM-based static analyzer for out-of-bounds array accesses in Qualcomm drivers.
- Applied IAAFinder to five Qualcomm drivers and uncovered 11 zero-day vulnerabilities in the drivers (7 rated critical severity).

Student Researcher, Google - *Google Open Source Security Team* *May 2023 - Aug 2023*
Project: Runtime Security Framework to Mitigate Software Supply Chain Vulnerabilities in Java
- Proposed *Zero-Trust Dependencies (ZTD)*, a policy-based architecture to secure vulnerable dependencies.
- Developed a dependency isolation system, ZTD_{Java}, to monitor dependency resource accesses and enforce security policies.
- Evaluated ZTD_{Java}, demonstrating its effectiveness in blocking exploits and applicability to real-world systems.

PROFESSIONAL SOFTWARE ENGINEERING EXPERIENCE

Software Engineer, Seamfix Ltd *Mar 2020 - Jun 2021*

- Designed backend services in Java and Quarkus for user and organization management on the Seamfix Base platform.
- Built APIs for payment and wallet management using Java and Spring Boot in the Seamfix Payment platform.
- Developed the Seamfix Mobile Device Management (MDM) app in Java for tracking and controlling device fleets.

Mobile Applications Developer, Naija Cowry Technologies Ltd *Jun 2019 - Mar 2020*

- Developed and maintained the E-Dey Shop merchant app in React Native for inventory and sales management.

Graduate Software Engineer Trainee, Seamfix Ltd *Jan 2019 - Mar 2019*

- Added audit and error-reporting features to the Seamfix BioSmart app for SIM card registration.

PUBLICATIONS

Amusuo, Cochell, Le Lievre, Patil, Machiry and Davis. *Do Unit Proofs Work? An Empirical Study of Compositional Bounded Model Checking for Memory Safety Verification*. Accepted at the 48th IEEE/ACM International Conference on Software Engineering. Preprint: https://arxiv.org/abs/2503.13762 (**ICSE'26**).

Amusuo, Liu, Mendez, Metzman, Chang and Davis. *FalseCrashReducer: Mitigating False Positive Crashes in OSS-Fuzz-Gen Using Agentic AI*. Submitted to the 48th IEEE/ACM International Conference on Software Engineering (Software Engineering in Practice). Draft: https://arxiv.org/abs/2510.02185 (**Arxiv '25**).

Tanksalkar, Muralee, Danduri, **Amusuo**, Bianchi, Davis and Machiry. *LEMIX: Enabling Testing of Embedded Applications as Linux Applications*. Accepted at the 34th USENIX Security Symposium. Preprint: https://arxiv.org/abs/2503.17588 (**USENIX Security'25**).

Amusuo, Robinson, Torres-Arias, Machiry, Simon, and Davis. *ZTD$_{Java}$: Mitigating Software Supply Chain Vulnerabilities via Zero-Trust Dependencies*. Proceedings of the 47th IEEE/ACM International Conference on Software Engineering. Preprint: https://arxiv.org/pdf/2310.14117 (**ICSE'25**).

Amusuo, Patil, Cochell, Le Lievre, and Davis. *A Unit Proofing Framework for Code-level Verification: A Research Agenda*. Proceedings of the 47th IEEE/ACM International Conference on Software Engineering - New Ideas and Emerging Results track (ICSE-NIER '25) Preprint: https://arxiv.org/abs/2410.14818 (**ICSE-NIER'25**).

Amusuo, Méndez, Xu, Machiry, and Davis. *Systematically Detecting Packet Validation Vulnerabilities in Embedded Network Stacks*. Proceedings of the 38th ACM/IEEE International Conference on Automated Software Engineering (**ASE'23**).

Amusuo, Sharma, Rao, Vincent, and Davis. *Reflections on Software Failure Analysis*. Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering — Ideas, Visions, and Reflections track (**ESEC/FSE-IVR'23**).

Srinivasan, Tanksalkar, **Amusuo**, Davis, and Machiry. *Towards Rehosting Embedded Applications as Linux Applications*. Proceedings of the 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN'23**).

SECURITY VULNERABILITIES REPORTED IN OPEN-SOURCE SOFTWARE

FreeRTOS (2): CVE-2024-38373 (CVSS 9.6), CVE-2025-5688 (CVSS 8.4).
Zephyr-RTOS (5): CVE-2025-1675 (CVSS 9.1), CVE-2025-1674 (CVSS 8.2), CVE-2025-1673 (CVSS 8.2), CVE-2025-9558 (CVSS 7.6), CVE-2025-9557 (CVSS 7.6).
Contiki-ng (6): CVE-2024-41126 (CVSS 8.4), CVE-2024-41125 (CVSS 8.4), CVE-2023-34100 (CVSS 7.3), CVE-2023-34101 (CVSS 9.1), CVE-2023-37459 (CVSS 5.3), CVE-2023-37281 (CVSS 5.3).
PicoTCP (4): CVE-2023-35847 (CVSS 7.5), CVE-2023-35846 (CVSS 7.5), CVE-2023-35849, CVE-2023-35848 (CVSS 7.5).
RIOT-OS (9): *Reported and still under investigation*

SKILLS

Programming: C/C++, Python, Java, JavaScript
Agentic AI: Agent design, tool development, prompt and context engineering, AI Frameworks (Google ADK, PyTorch)
Program Analysis: Static Analysis (LLVM, CodeQL), Fuzzing (AFL, OSS-Fuzz), Verification (CBMC, Kani)
Systems: Embedded software, real-time operating systems (RTOS), network protocol stacks, operating system drivers
Security: Container isolation, zero-trust architecture, policy enforcement, secure OS stacks