SEAN N. ANDERSON

Hillsboro, OR 97124 **607.744.2019**
www.linkedin.com/in/sean-noble-anderson **sean.noble.anderson@protonmail.com**

Leverage Diverse Theory and Systems Background into Novel, Grounded Research
Expertise in early stage research coupled with systems experience. Fluent in programming language theory and formal logic. At ease adopting and working in new theoretical frameworks. Strong technical writing and interpersonal skills.

EDUCATION

Doctor of Philosophy (PhD) In Progress, Computer Science, Portland State University, Portland, OR, exp. June, 2024
Dissertation: Permissive Memory Safety and Customized Security in a C Dialect Backed by Hardware Tags
Supervisor: Andrew P. Tolmach
Summary: I introduce Tagged C, a C dialect whose semantics are parameterized by tag-based security policies, intended to be backed by an underlying reference monitor such as PIPE (Programmable Interlocks for Policy Enforcement). With its large legacy codebases that may rely on unsafe coding idioms, C benefits from the strong protections of runtime monitoring. Legacy codebases in particular may need flexible enforcement: strict in security-critical contexts, permissive when the codebase relies on low-level idioms. Ideally, such protections should be written and reasoned about at the source level. To this end, my core contribution is a semantics and reference interpreter for Tagged C, written in Coq. I show that Tagged C can enforce important permissive models of memory safety from the literature, a range of customized program-specific policies, and a compartmentalization policy that isolates the risks of permissive enforcement from parts of the program that need stronger guarantees.

Bachelor of Science (BS), Computer Science, Minor in Mathematics, with distinction, Clarkson University, Potsdam, NY

PUBLICATIONS

Zhenyang, D. et al., ***Verifying Rust Implementation of Page Tables in a Software Enclave Hypervisor.***
April 2024. In proc., 29th ACM Intl. Conf. on Architectural Support for Programming Languages and Operating Systems

Anderson, S., Naaktgeboren, A., and Tolmach, A. ***Flexible Runtime Security Enforcement with Tagged C.***
October 2023. In proc., 23rd International Conference on Runtime Verification.

Chhak, C., Tolmach, A., and Anderson, S. ***Towards Verified Compilation of Concrete C.*** Submitted for Publication.

Anderson, S., Blanco, R., Lampropoulos, L., Pierce, B., and Tolmach, A. ***Formalizing stack safety as a Security Property.***
July 2023. In proc., 36th IEEE Computer Security Foundations Symposium.

Chhak, C., Tolmach, A., and Anderson, S. ***Towards Formally Verified Compilation of Tag-Based Policy Enforcement.***
January 2021. In proc., 10th ACM SIGPLAN Intl. Conf. on Certified Programs and Proofs.

TECHNICAL SKILLS

Theories

- Expertise in Theoretical Security, Trace Properties
- Expertise with Formal Semantics (esp. C)
- Proficient in Theorem Proving, Randomized Testing
- Familiar with Algorithms and Complexity Theory
- Familiar with Computational Learning Theory, Markov Models, Deep Learning, "Good-Old-Fashioned AI"

Languages & Tools

- Proficient in Ocaml
- Proficient in C
- Expertise with Coq Proof Assistant, QuickChick
- Expertise with Linux, Git, LaTeX workflows
- Familiarity using SVM classification in Python
- Experience implementing language models in Python and Go

RESEARCH PROJECTS

Tagged C **2022 – present**
https://github.com/SNoAnd/Tagged-C – Source semantics, reference interpreter, and proofs for a C variant with tag-based security enforcement. Implemented in Coq and extracted to Ocaml.

Stack Safety as a Security Property **2021 – present**
https://github.com/SNoAnd/stack-safety – Formal properties describing a specification for stack safety, against which a stack safety enforcement mechanism may be validated. Proof of concept tests PIPE policies on RISC-V simulator using QuickChick randomized property-based testing.

PROFESSIONAL EXPERIENCE

Graduate Research Assistant in PIPE Tagged Architecture Portland State University, Portland, OR **2018 – present**
Participated in multi-institution project to enforce security policies through metadata-tagged hardware, expanding scope of policies to those applied at C source-level. Designed compartmentalization policies, stack protection policies, and associated properties. Now developing Tagged C, a C-dialect with metadata tags built into its semantics.

- Implemented proof-of-concept compiler in Coq, interfacing with Ocaml parser via extraction.
- Applied information flow control and memory safety policies, with novel compartmentalization policy.
- Designed formal properties compatible with lazy enforcement.
- Developing full C formal semantics, with reference interpreter and proof of correctness, in Coq with Ocaml backend.

  Presentation January 2020, Flexible Tag-based Policies for Compartmentalized C, presented at Principals of Secure Compilation 2020, New Orleans, LA - Link to Talk

CERTIK, Remote

Intern, Research and Development **June – September 2021**
Formally verified Rust-based hypervisor implementation, Hyperenclave. Expanded proof automation for functional verification proofs in Coq. Developed memory model and Rust formal semantics for use in proofs. Led team of four interns on overall functional verification effort.

- Formalized top-level security theorem for non-interfering trusted execution environments.
- Outlined proof structure connecting top-level theorem with functional specifications.
- Prioritized critical subset of the code-base for verification.
- Identified frequent proof bottlenecks and automated proof tac tics for them.

CERTIK, Remote

Intern, Research and Development **June – September 2020**
Added features to CompCert-based verified compiler backend. Strategically planned compilation chain to maximize proof reuse. Developed novel memory model unifying Ethereum storage and memory pointers and implemented compilation to integer and hash pointer models for EVM backend. Integrated memory model with Web Assembly backend for stack-allocated large variables.

- Extended verified compiler in Coq.
- Designed novel memory model based on "extended identifiers" enabling easier proofs in the presence of pointers.

IBM: LINUX TECHNOLOGY CENTER, Hillsboro, OR **2013 – 2018**

Software Engineer, Embedded Linux
Developed and supported customized Linux-based operating systems for embedded devices.

- Engaged with project management principles to plan for long-term supportability of specialized OS.
- Drove implementation of continuous integration using Jenkins, smoothing workflow for very active customers / developers, enabling adoption by other teams. Earned Manager's Choice Award for Agile development practices.
- Worked closely with customers to develop clear support expectations motivated by long-term client success.

VOLUNTEER EXPERIENCE

CASCADIA WILD, Portland, OR, **Wolverine Tracking Project** **October 2018 – January 2023**
Surveyed for large predators on Mt. Hood. Identified tracks and recorded measurements.