Bradford Larsen, Computer Scientist

11 years software security-relevant industry experience · CS PhD coursework at Tufts

brad@bradfordlarsen.com
bradfordlarsen.com
linkedin.com/in/bradfordlarsen
github.com/bradlarsen

Summary

A strong generalist with 11 years of security-relevant industry experience. Subject matter expert in static program analysis, symbolic execution, abstract interpretation, compiler and interpreter development, fuzzing and property-based testing, and appsec review. Currently working to empower security operators using modern machine learning.

Looking to have outsized impact in areas such as application security, language implementation, or program analysis at an early- to mid-stage product company.

A programming language polyglot; has shipped commercial software in Python, C++, C, Rust, Haskell, Go, R, C#, JavaScript, and Scala.

Experience

Principal Security Engineer, Static Analysis and ML — **August 2021 – Present**
Praetorian, Austin, TX (full-time remote)

- Served as security subject matter expert for a 4-person team building static analysis tools based on deep learning and large language models
- Was the primary code author of Nosey Parker, a state-of-the-art detector of hardcoded secrets that combines classical pattern matching and machine learning
- Developed and owned data pipelines and labeling tools for hardcoded secret data, instrumental in producing a dataset of 20k hand-labeled examples of hardcoded secrets

Senior Security Engineer, AppSec and Research — **July 2019 – July 2021**
Trail of Bits, New York, NY (full-time remote)

- Conducted 20+ multi-week white-box application security audits for clients using a combination of manual review, static and dynamic analysis tools, and fuzz testing; applications included embedded software, cryptographic libraries, distributed ledgers, and a very popular video chat app
- Modified the Manticore symbolic execution tool (github.com/trailofbits/manticore) to improve support for ARMv7 and Linux system calls as part of the DARPA LADS project
- Created fuzz targets with custom structure-aware mutators for a new Datalog language being implemented under the DARPA AMP project; prevented dozens of bugs from shipping by automatically fuzzing in CI

Senior Software Developer, Infrastructure and Security **January 2016 – June 2019**
Ab Initio Software, Lexington MA

- Established a system for continuous use of static analysis tools for a huge C++ codebase
- Found over 100 memory corruption, undefined behavior, and logic errors in Ab Initio base code; collaborated with dozens of code owners to remediate these issues
- Ported CPython 2.7 to z/OS; updated 30 internal patches for CPython 2.5 to CPython 2.7
- Ported GNU Make to z/OS; added a profiling mechanism for build times to GNU Make 4.2
- Found and fixed a bug introduced in SQLite 3.11 that prevented it from working on EBCDIC systems; submitted a patch that was accepted upstream
- Improved developer productivity company-wide by implementing custom GDB extensions and pretty-printers for data types in Ab Initio base code

Senior Software Engineer, Binary Static Analysis **May 2012 – December 2015**
Veracode, Burlington MA ($614M exit)

- Increased speed of Veracode's flagship production system by 4x via improved job scheduling algorithms and coaching others on use of sampling profilers
- Saved 160+ hours per week by overhauling legacy QA infrastructure, automating test failure triage, and improving build times
- Saved 20+ hours per week in manual buffer overflow vulnerability triage by implementing a symbolic execution postprocessor, built on top of the Z3 SMT solver
- Fixed 50+ memory corruption errors by refining core data types *(interfaces really matter!)*
- Found and fixed 1000+ errors in Veracode's security rules domain-specific language by creating a custom lint tool (written in C#) and integrating it in the development process
- Identified dozens of latent defects in Veracode's flagship product by collecting production metrics, then analyzing them using R, ggplot2, and SQL; delegated repair of these defects

Research Assistant **2010 – 2012**
Tufts University Computer Science, Medford MA

- Identified interpreter design decisions that have significant affect on runtime performance by developing and evaluating 924 interpreter implementations for Lua in OCaml
- Formalized part of a distributed revision control system in the Coq proof assistant
- Implemented a clear & efficient binary decision diagram library in ANSI C, along with Haskell bindings (github.com/bradlarsen/bdd)

Research Assistant **2007 – 2010**
UNH Computer Science, Durham NH

- Designed and implemented Barracuda, a statically typed, array-based language for numeric problems; its optimizing compiler for GPUs generated faster code than NVIDIA's BLAS library for certain operations
- Designed and implemented Switchback, a shortest path algorithm that ran up to 10x faster than earlier algorithms and required no explicit heuristic function
- Improved processing time by 100x in space weather data conversion tools by reducing I/O
- Made software builds repeatable by replacing a legacy build system with GNU Autotools

Software Contractor **2007**
Evergreen Solar, Marlboro MA

- Automated a solar panel simulation workflow using Python and wxPython on Windows

Visiting Research Assistant **2007**
Friedrich-Alexander University, Erlangen, Germany

- Improved performance by 3x in a software model checker and a naive graph isomorphism checker by implementing object caching in a distributed shared-memory JVM

Performance Algorithms Group Intern **2005 – 2006**
Mercury Computer Systems, Chelmsford MA

- Fixed bugs, added tests, and added new functions to Mercury's Vector Signal Image Processing Library, a high-performance signal processing library in C
- Maintained an Eclipse debugger plugin for remote programs running on a Cell processor

Education

2010 – 2012	Ph.D. Computer Science (incomplete), Tufts University
2010	M.S. Computer Science, University of New Hampshire
2009	B.A. Philosophy, University of New Hampshire
2008	B.S. Computer Science, University of New Hampshire

Publications

Carson Harmon, Bradford Larsen, and Evan Sultanik. *Toward Automated Grammar Extraction via Semantic Labeling of Parser Implementations.* LangSec 2020. IEEE Security & Privacy, May 21, 2020.

Jan Midtgaard, Norman Ramsey, and Bradford Larsen. *Engineering definitional interpreters.* PPDP 2013. ACM, September 2013.

Bradford Larsen. *Simple optimizations for an applicative array language for graphics processors.* DAMP '11. ACM, January 2011.

Bradford Larsen, Ethan Burns, Wheeler Ruml, and Robert Holte. *Searching without a heuristic: efficient use of abstraction.* AAAI-10. AAAI Press, July 2010.

Ronald Veldema, Bradford Larsen, and Michael Philippsen. *A DSM protocol aware of both thread migration and memory constraints.* PDCS '08. ACTA Press, November 2008.

CVEs

- CVE-2019-19274: DoS from out-of-bounds read in CPython v3.8.0a1–v3.8.0a3
- CVE-2019-19275: DoS from out-of-bounds read in Python typed_ast library 1.3.0 and 1.3.1

Awards

2010 – 2012	Dean's Fellowship, Tufts University, School of Engineering
2009	University of New Hampshire Barlow Prize, Honorable Mention
2008 – 2009	NASA New Hampshire Space Grant Consortium Graduate Fellowship
2007	UNH International Research Opportunities Program Fellow
2003	Eagle Scout, Boy Scouts of America