William Blair

Department of Computer Science
111 Cummington Mall
Boston, M.A. U.S.A.

email: wdblair@bu.edu
url: https://wdblair.io

Current position

Ph.D. Candidate, Boston University

Research Interests

I am interested in developing novel program analysis and verification tools for cybersecurity. Currently, I investigate how fuzz testing can detect Algorithmic Complexity (AC) vulnerabilities in Java programs. State of the art fuzzers such as afl and libFuzzer typically target binary programs and are optimized for discovering memory corruption vulnerabilities that allow remote adversaries to either leak information from a process or achieve code execution. In contrast, a threat model where adversaries degrade an application's performance by submitting inputs that trigger its worst-case execution time or space consumption, is much less studied from a program analysis perspective. Indeed, few fuzzers target applications written in high level languages where memory corruption vulnerabilities are less prevalent. I currently develop HotFuzz, a fuzz testing framework that detects Algorithmic Complexity (AC) vulnerabilities in Java libraries as a part of the DARPA Space and Time Analysis for Cybersecurity (STAC) program. HotFuzz has detected previously unknown vulnerabilities in the Java Runtime Environment (JRE) that have been confirmed by Oracle and IBM.

Education

2014-present PhD in Computer Science, Boston University
Advisors: Manuel Egele, Hongwei Xi

2012-2014 MS in Computer Science, Boston University
Project: *Dependent Types for Real Time Constraints*
Advisor: Hongwei Xi

2008-2012 BA in Computer Science, Boston University

Publications

2020 William Blair, Andrea Mambretti, Sajjad Arshad, Michael Weissbacher, William Robertson, Engin Kirda, Manuel Egele. HotFuzz: Discovering Algorithmic Denial-of-Service Vulnerabilities Through Guided Micro-Fuzzing. In Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS) San Diego, CA US, February 2020.

2017 William Blair, Hongwei Xi. Dependent Types for Multi-Rate Data Flows in Synchronous Programming. In the Post-Proceedings of ACM ML/OCAML Workshop 2015. EPTCS 241, pp. 36-44.

Talks

2021 Microservice-Aware Reference Monitoring through Hybrid Program Analysis
FloCon 2021 at CMU Software Engineering Institute (SEI)

2019 HotFuzz: Finding Space and Time Vulnerabilities in Java Programs
DARPA Space and Time Analysis for Cybersecurity P.I. Meeting

2016 Continuum: Finding Space and Time Vulnerabilities in Java Programs
DARPA Space and Time Analysis for Cybersecurity P.I. Meeting

2016 Side Channels and Worst Case Behavior in Java
Northeastern-WPI Seminar on Security

2015 Using a Portfolio of SMT Solvers in Software Development
NEPLS Fall at Tufts University

2015 Dependent Types for Real Time Constraints
ACM Sigplan ML Workshop at ICFP 2015

2015 Integrating SMT into Software Development
NEPLS Spring at Wesleyan University

2014 Debugging with Types in ATS
Boston Haskell Meetup

Service

2021 Shadow Program Committee member for IEEE Security and Privacy (Oakland)

2021 Sub-Reviewer for NDSS

2020 Sub-Reviewer for ACM CODASPY,DSN

2019 Sub-Reviewer for ACM CODASPY

2018 Artifact Evaluation Committee member for ACSAC

2018 Sub-Reviewer for ACSAC, RAID, DIMVA, ACM CODASPY

2017 Artifact Evaluation Committee member for ACSAC

2017 Sub-Reviewer for ACM CODASPY

Teaching

Fall 2020 *TF* for CS630 Graduate Design and Analysis of Algorithms

Fall 2019 Lectured on topics including Linear Algebra, LUP Decomposition, Complexity, Approximation Algorithms, Randomized Algorithms, and Linear Programming. Managed a small team of graders.

Spring 2015 *TF* for CS111 Introduction to Computer Science

Fall 2014 Assisted students through a breadth first introduction to Computer Science that covers programming in Functional, Imperative, and Object Oriented paradigms. Other topics such as Computer Organization, Assembly Programming, and Computational Complexity were briefly introduced as well. The class was adapted from the "CS For All" class developed at Harvey Mudd University. My role included leading discussion sections, grading, and holding office hours.

Spring 2014 *TF* for CS211 Object Oriented Programming

Assisted students with learning Objective C and writing applications for iOS devices. Students first built familiarity with the iOS environment by gradually constructing a Tweeting App in iOS, and then developed original apps on their own.

Miscellaneous

2020 3rd Place speaker at 7th Annual BU CISE Graduate Student Workshop (CGSW 7.0)

2019 2nd Place speaker at 6th Annual BU CISE Graduate Student Workshop (CGSW 6.0)

2018 Student Travel Award to the IEEE Symposium on Security and Privacy

2016 Sixth Summer School on Formal Techniques at Menlo College

2015 Verification Mentoring Workshop at the International Conference on Computer Aided Verification (CAV)

Professional Experience

2019-2021 *Research Intern* at IBM Research

Researched System Security topics in the Cyber Security Intelligence (CSI) Group.

2015 *Software Engineer Intern* at ViaSat

Assisted in developing a Business Process Engine (BPE) that provides a fault tolerant programming framework for integrating components of distributed systems.

2013 *Software Engineer Intern* at ViaSat

Investigated how mobile applications received multi-media from content providers. This required reverse engineering native ARM libraries in Android applications, and developing prototypes where a man-in-the-middle server augments the behavior of Javascript applications.

2009-2012 *Software Engineer* at 829 Studios LLC

Designed, implemented, and deployed OfferedLocal, a web application that allows businesses to run location based advertising campaigns across social networks like Facebook and Twitter. The start-up participated in Mass Challenge and was featured in the Demo Fall 2011 Conference.

Developed and maintained the back office system for the Licensing Industry Merchandisers Association (LIMA), along with an online directory of member companies.

2009-2010 *Technician* at Electronics Design Facility

Developed firmware for a medical prototype as a part of the FLARE project at Beth Israel Hospital. The system allowed an external device to control the power output of lasers and regulated their temperature using Peltier coolers. The firmware featured serial communication, measuring temperature from ADCs, and PID controllers that managed temperature through pulse width modulation.