Nathan Lilienthal

nathan@nixpulvis.com ∘ +1 (202) 701-4368 ∘ **Available Now**
`nixpulvis.com` ∘ `github.com/nixpulvis`

Languages: Ruby, Rust, C/C++, LUA, Racket, Shell. ECMAScript (JS), Python, Java, and more...
Systems: UNIX, Linux, Git + Hub/Lab, Rails, Heroku, Postgres, SQLite, AVR/ARM, WoW (ask me about it).

Professional Experience

- **Northeastern University** **Boston, MA**
  Research Programmer, Intelligence Advanced Research Projects Activity, HECTOR *Aug. 2019 – Aug. 2020*
  - Collaberativly developed a hybrid-mode secure programming language design for multi-party computation (MPC)
  - Represented my team, at both remote and in-person technical exchange meetings with other researchers
  - Built a prototype implementation of our language, which is forked from the Rust programming language
  - Began a formalism for our language(s), which will include sound typing rules, and reductions

  Teaching Assistant, Fundamentals of Computer Science 1 *Fall of 2012, 2013, 2014, and 2015*
  - Conducted mini lectures, monitored class progress, and answered students' questions
  - Discovered new ways to present concepts that facilitated student understanding
  - Assisted students during established office hours

- **Forward Financing Inc.** **Boston, MA**
  Sr. Software Engineer *May 2018 – Aug. 2019*
  - Developed a client wrapper for an Algolia search implementation
  - Performed various application performance improvements, often caused by unacceptable response times, quickly
  - Planned architecture refactoring, including object model improvements, and a new data permissions system
  - Led efforts to create an orchestration CLI for managing a complex Heroku + Salesforce microservice system
  - Mentored the co-op university students by providing deep code reviews and pairing on problems

- **HOMER Energy** **Boulder, CO**
  Software Developer, Summer Intern *Jul. 2017 - Nov. 2017, Summer 2012*
  - Built API integrations for the HOMER C# application, including REST and CSV file APIs, which involved a general refactoring of the code which imports data, complete with added tests
  - Developed an internal tool to view the Google Protocol Buffer used to pass values between all parts of the application, allowing developers to quickly see inputs and outputs
  - Created a web based front-end for HOMER in Rails, which served at the starting point for another version and provided a proof of concept for how to integrate the HOMER API with a webserver

- **Apple Inc.** **Cupertino, CA**
  Software Engineer *Jan. 2015 – Aug. 2015, Jul. 2016 – Jul. 2017*
  - Built a Ruby library (`radic`) and CLI (`radish`) for interacting with Apple's bug management system (aka Radar)
  - Participated in a cross-cutting web design work group to help create common components for the hardware teams
  - Contributed to an internal tool for managing hardware validation, which was inspired in part by Travis CI
  - Contributed to another internal tool for analysing large amounts of pre-production device test data

- **Americas Test Kitchen** **Boston, MA**
  Web Developer *Jan. 2014 – June 2014*
  - Pushed code to the front-end and back-end for all four Americas Test Kitchen websites, including bug fixes and technical infrastructure upgrades
  - Built modularized components to abstract functionalities found common throughout the company's codebase

- **Bluesocket - Adtran** **Burlington, MA**
  Software Developer *Jan. 2013 – June 2013*
  - Developed an automated build system, which reduced turnaround time, allowing anyone to easily run a build
  - Addressed user reported issues in Ruby/Rails and LUA, including hardening validations and updating database migrations for old versions of the software
  - Designed a class/model structure for users and accesspoints, which allowed the back-end to represent clients of individual accesspoints

Interests: Teaching, Microelectronics, Woodworking, Music, Gaming, Skiing, Cats, ...

Education

- **Northeastern University** **Boston, MA**
  College of Computer and Information Science *2011 – 2016*
  Bachelor of Science in Computer Science
  - Relevant Courses:
    * Programming Languages
    * Special Topics in Programming Languages
    * Compilers
    * Systems and Networks,
    * Computer Organization
    * Software Development (aka HELL),
    * Theory of Computation
    * Algorithms and Data Structures
    * Fundamentals of Computer Science 1 & 2,
    * Object Oriented Design,
    * Artificial Intelligence
    * Logic and Computation,
    * Combinatorics
  - Clubs & Extracurriculars:
    * NU Hacks
    * Hack Beanpot
    * ACM