# Kevin Nause

*Senior Security Software Engineer*

## Contact

425.626.7520

kevin@nause.engineering
linkedin.com/in/kevinnause
github.com/Nauscar

## Programming

Rust, C, C++, C#, x86 & ARM Assembly, Go, Python, Java

## Frameworks

OpenCL, OpenMP, Hadoop, Thrift, Qt, ASP .NET

## Interests

Embedded Systems, Photography, Performance Vehicles

## About

I enjoy low level programming on platforms such as embedded systems and operating systems. Working on wearable hacks and obtaining root access on mobile devices are also side interests. Computer security and static analysis are key interests of mine. I have been a Linux enthusiast since I typed "Hello World" for the first time and have adored penguins ever since. The first thing I do when I sit down at a computer is change the keyboard layout to Dvorak and plug in an IBM Model M keyboard.

## Education

**2016** **Bachelor of Applied Science (B.A.Sc.)** — University of Waterloo
Computer Engineering

## Experience

**Feb 2020** (3 yrs 2 mos) **Microsoft** — Redmond, Washington
*Senior SW/FW Engineer, Senior Security Software Engineer*
- Lead security procedures between core, build, validation, and post-launch teams
- Vulnerability identification, mitigation, and incident response
- Performed threat modeling, static analysis, dynamic testing, and pentesting
- Scoped security features and utilized cryptographic standards
- Communicated severity of known issues and compliance requirements
- Researched future security technologies for business success

Products: Microsoft Devices
Hardware: NXP i.MX RT ARM Cortex-M33
Languages Used: Rust, C, C++, x86 & ARM Assembly

**Nov 2018** (1 yr 3 mos) **NCC Group** — Seattle, Washington
*Senior Security Consultant*
- Design audit for Bootloaders, RTOS, Linux, Android, and Windows
- Scoped and lead multi-week hardware focused client engagements
- Hardware teardown, flash dumping, bus probing
- Reverse engineering C, C#, and Java binaries
- Vulnerability assessment, code review, network pentesting
- ARM shell code creation and control highjacking attacks
- Automotive (CAN SAE J1939) and Robotics (ROS) security experience

Languages Used: C, ARM Assembly, Python, Go

**Jul 2016** (2 yrs 5 mos) **Microsoft** — Redmond, Washington
*SW/FW Engineer II, SW/FW Engineer*
- Working on ECs for platforms with Intel CPUs and Nvidia GPUs
- Experience with power sequencing, battery, and thermal subsystems
- Implementing inter-bus communications via USB, UART, SPI, I2C, SMBus
- Working with communication protocols such as TCP/IP, HID, RS-232
- Proficient with oscilloscopes and logic analyzers
- Experience with schematics, reference manuals, and errata for hardware peripherals

Products: Surface Hub, Surface Laptop
Hardware: NXP/Freescale K22 ARM Cortex-M4
Languages Used: C, ARM Assembly, C#, PowerShell

**Aug 2015** (5 months) **Pebble Technology** — Kitchener, Ontario
*Embedded Firmware Engineer*
- Implemented device drivers, recovery firmware, and system applications on the Pebble OS (based on FreeRTOS)
- Primary focus was porting the current firmware to an older device with significantly less flash storage and a black and white screen
- Optimized anti-aliasing on 8-bit displays, and dithering on 1-bit displays

Products: Pebble, Pebble Time, Pebble Time Round
Hardware: STM32F4 ARM Cortex-M4, STM32F2 ARM Cortex-M3, TI CC2564
Languages Used: C, ARM Assembly, Python

**Jan 2015** (4 months) **Motorola** Kitchener, Ontario
*Security Engineer*
- Discovered and patched vulnerabilities, resource leaks, and concurrency problems in Android OS, Motorola's MSM kernel, and Moto X sensor hub
- Used static analysis to assist in discovering security vulnerabilities
- Traced execution flow to isolate false positives or potential exploits

Products: Moto E/G/X, Moto 360
Hardware: TI OMAP 3, Qualcomm PM8921 PMIC, NXP 44701 NFC
Languages Used: C, C++, Java

**Sep 2014** (8 months) **Computer Aided Reasoning Group** Waterloo, Ontario
*Undergraduate Research Assistant, Unviersity of Waterloo*
- Member of the Computer Aided Reasoning Group, reported to Dr. Vijay Ganesh
- Researched the topic of SAT solvers and their underlying heuristics
- Studied the relevance of backdoor variables and community structure for the VSIDS decision heuristic
- Experience with static analysis, symbolic execution, and Return Oriented Programming (ROP)

Languages Used: C, C++, x86 Assembly, Java

**May 2014** (4 months) **ON Semiconductor** Waterloo, Ontario
*Embedded Tools Developer*
- Designed Bluetooth Low Energy GATT services for functions such as data streaming, audio streaming, and status updates
- Embedded programming with BLE enabled medical devices such as hearing aids, insulin monitors, and heart rate monitors
- Interfaced with Windows and Android client devices

Hardware: Nordic nRF51822 Bluetooth Low Energy Controller, ARM Cortex-M0
Languages Used: C, C++, Java, ARM Assembly

**Sep 2013** (4 months) **eSolutionsGroup** Waterloo, Ontario
*Mobile Developer*
- Designed a real-time transit prediction system using GTFS data and protocol buffers
- Database design, MVC server communications, and mobile application development

Languages Used: C# (ASP .NET), SQL, JavaScript

**May 2012** (16 months) **Regional Municipality of York** Richmond Hill, Ontario
*Transit Management Systems*
- Worked with GTFS data and real-time prediction feeds and contributed to the OneBusAway project
- Hands on work with transit embedded systems and fare management systems

Languages Used: C#, Java

# Projects

*Weekend projects demonstrating my learnings in Rust over the past year*

**Nov 2021** **Determinate** Github Repo
Procedural macro attributes to mark a function as determinate or indeterminate for testing runtime determinism.

**Oct 2021** **NRF52 Firmware Demo** Github Repo
A demonstration of the Real-Time Interrupt-driven Concurrency (RTIC) framework running on the nrf52840.

**Sep 2021** **Bare Metal Runtimes** Github Repo
Bare metal Rust runtimes for no_std for ARM and RISC-V toolchains.

**May 2021** **Persistant Variables** Github Repo
A persistent variable type that serializes/deserializes its value to/from disk upon declaration and drop.