Paul Kelly

12738 25th Ave NE
Seattle, WA, 98125

pjk2563@rit.edu
(484) 949-5943

Objective

To continue research and increase my experience in the embedded cybersecurity industry

Education

Rochester Institute of Technology Rochester, NY
Bachelor of Science in Computer Engineering *Expected Graduation: May 2022*

- **GPA**: 3.71
- **Courses**:
  Digital System Design I, II
  Intro to Embedded Systems
  Computer Science I, II
  Computer Architecture
  Digital Signal Processing
  Applied Programming in C

Technical Skills

Languages: C/C++, Rust, Python, ARM Assembly, VHDL, C#, Java, Golang

Software: Binary Ninja, Bash, Burp Suite, Windows, Linux, Vim

Technologies: FPGA, ARM, ESP32, Digital Multimeter, SDR, Bluetooth and BLE, 802.1x, 3G/4G/5G Cellular, Signal Analyzer, Solder Iron

Experience

IOActive, Inc Seattle, WA
Senior Security Consultant *June 2022 - June 2025*

- Improved customer product lines by performing embedded analysis tests to identify security vulnerabilities
- Leveraged engineering experience to perform security assessments in the automotive industry
- Conducted internal research to explore under tested technologies to improve the overall security posture of the tech world

Saab Defense and Security Syracuse, NY
Electrical Engineer Intern *May 2019 - Dec 2019*

- Verified VHDL modules by generating files of known good inputs and outputs with Python, then writing a VHDL testbench to simulate the module with the test data
- Removed uncertainties in board redesign by creating and running test processes on potential components
- Reduced test times by automatically generating mixer intermodulation tables through instrumentation of test equipment with Python and the NI VISA API

Projects

RIT Formula SAE

- Designed a PCB for use in a custom battery management system on the electric car
- Made wireless telemetry possible through an Xbee network transmitting serial messages to a Raspberry Pi running an internet access point with a webserver
- Designed the dashboard PCB to hold a teensy microntroller that managed a TFT screen driver

Boosted Android

- Enabled wireless and USB gadget drivers by pulling LineageOS 16.1 source tree and modifying kernel code
- Ported Hijacker source code to work with custom driver access
- Added custom functionality to apps by reverse engineering compiled apks without source code

MIPS Processor Model

- Designed the instruction fetch, instruction decode, execute, memory access, and write back stages of the MIPS processor in VHDL