

## EXPERIENCE

### Qualcomm | Senior Software Engineer, May 2020 – Present

Working on the Secure Processor Unit [SPU] solution. SPU is a crypto co-processor inside the Snapdragon SoC which provides security related services and features like secure boot, cryptographic acceleration, key provisioning, anti-replay protection etc.

The focus of my work was designing and developing [C/C++] features for our proprietary kernel running on SPU.

Main contributions:

- Deep dived into hardware assisted memory virtualization module and delivered virtual memory support for our proprietary kernel.
- Studied kernel IPC [inter process communication] and RPC [remote process communication] mechanisms, designed and delivered RPC loop-back functionality. This allowed verification of RPC code in pre-silicon stage while running SPU in standalone mode [without any other subsystems].
- Improved performance of software-based encryption algorithms by 500%.
- Improved performance of NVM commit operations by 30%.
- Debugging and maintenance of complex system wide features spanning boot loaders, Linux kernel drivers [platform and character] and SPU kernel.

Tech stack and skills: C, C++, lauterbach trace32, DMA, Linux kernel.

### Cisco Systems | Software Engineer, July 2018 – April 2020

Worked on Application Visibility and Control [AVC] solution. The AVC solution allows to recognize, analyze and optimize network traffic as well as help with anomaly detection, network troubleshooting etc.

The focus of my work was designing and developing [C/C++] code for Deep Packet Inspection [DPI] engines running on proprietary Network Processing Unit [NPU].

Main contributions:

- Deep dived into SIP protocol, designed and developed SIP Parser Engine which allows recognition of SIP traffic with minimal latency hit. The feature involved end to end understanding and implementation of data and control plane with hard memory and runtime constraints.
- Designed and developed Dynamic Flow Attributes, the feature allows marking [decision on what traffic to mark was received from network controller] of traffic and performing QoS [quality of service].

Tech stack and skills: C, C++, gdb, T-Rex traffic generator, low level performance aware code [cache coherency and performance], high concurrency.

### Security Researcher | Technion R&D Ltd, February 2018 – July 2018

Research of kernel level Windows vulnerabilities by means of static analysis. The goal of the research was to automatically detect double fetch vulnerabilities in device drivers.

Tech stack and skills: C/C++, WinDBG, IDA Pro, KMDF.

### Q Core Medical | Embedded Software Engineer [part-time position], July 2016 – February 2018

Bare-metal firmware development in C.

Tech stack and skills: DMA, SPI, complex hardware debugging using Oscilloscope.

---

## PERSONAL PROJECTS

### Network Interface Card Driver for the Linux Kernel

The goal of this project is to gain understanding on how Linux kernel networking drivers work. First step of the project involved of reducing the E1000 driver to minimal core functionality, followed by writing the NIC driver code from scratch.

Tech stack and skills: C, QEMU/KVM, gdb, Linux kernel, DMA, PCI, Wireshark

### Anti-Ransomware Windows Kernel Mode Solution

The goal of this project was to gain understanding on how to write Windows kernel drivers. The main component of this project is a mini-filter driver [running in kernel mode] which tracks all IO operations and tries to identify malicious processes.

Tech stack and skills: C++, WinDbg, Windows kernel [mini-filter driver], KMDF.

---

## EDUCATION

### Technion – Israel Institute of Technology | Bachelor of Science, Computer Engineering

Specialization in operating systems and computer security.

Interesting courses: CPU Design, Operating Systems Engineering, Lab in RDMA programming.

---

## TECHNICAL INTERESTS

Low-level system programming, virtualization, operating systems internals and computer networks and security.