Paul Ellis

1111 Fake Address Rd
Some, Where 11111

555-555-5555
redacted@email.com

Keyword SEO

ARM Architecture, TrustZone, Virtualization, Confidential Compute, TEE, Global Platform, C, C++, Python, gTest, gMock, Scons, PlantUML, Ditaa, Markdown, org-mode, LaTeX, Doxygen, Security, Secure Boot, Middleware, Trace32, gdb, Linux, cgroups, Namespaces, SELinux, RTOS, Firmware, Operating Systems, Capabilities, Kernel, Runtime, Agile, Scrum, JIRA, Workfront, Git, Gerrit, Perforce, Code Collaborator, Github

Experience

Principle Engineer, Manager
2010 - Current

Qualcomm Technologies, Inc
Security Systems Group

- I currently lead a global team of 21 engineers located in the US, India and China.
- My team is responsible for development and maintenance of secure/confidential compute platforms across Qualcomm products. This includes a TrustZone-based TEE (QTEE) and Linux-based TEE.
- Boundaries of team responsibilities include: QTEE application SDK, proprietary middleware across QTEE and OS images, QTEE kernel, QTEE application runtime, QTEE EL3, Global Platform standard interface compliance, Linux VM containers, etc...
- The QTEE application SDK includes auto-generating documentation, a scons-based build system, example applications and a POSIX-based emulation environment.
- Proprietary middleware implements IDL-based IPC supporting transport across: Domain sockets, Android binder, non-secure/secure boundary, secure kernel/secure user boundary.
- Secure kernel responsibilites include object-based capability system, threading, scheduling, application lifetime management, dynamic linking, process isolation, memory management, etc...
- Secure userspace responsibilities include application runtime, secure storage, TZ user/kernel ABI, secure memory allocation container, POSIX middleware implementations, etc...
- EL3 responsibilities include TrustZone-based TEE virtualization without stage-2 protection.
- My individual software feature contributions include: secure firmware image loader, confidential firmware OTA updates, initial QTEE memory management implementation, initial QTEE AArch64 EL3 implementation, QTEE meltdown vulnerability mitigation (similar to Linux KPTI), Scons-based application build system, etc...
- Some individual non-software contributions: ARM FF-A standard collaboration, QTEE service virtualization design, Linux-based TEE design, development processes, project management processes, etc...

Ending Title: **Senior Engineer**
2005 - 2010

Qualcomm Technologies Inc,
MediaFLO

- Developed and maintained features in the driver and protocol stack for hardware responsible for receiving broadcast signal
- Collaborated with systems engineers on protocol design to make use of hardware features related to multi-frequency network decoding, dynamic interference cancellation, mobility, detransportized packet buffering, MAC layer packet reordering and system and service reacquisition
- Software ran in a concurrent multi-threaded RTOS and had to balance memory use and CPU budget to meet deadlines for hardware and bus latencies

Education:

2000-2004: University of California, San Diego
Bachelor of Science, Computer Engineering

References available upon request