DATTA MANIKANTA SRI HARI DANDURI

ddanduri@purdue.edu | +1 (480) 937-9406 | 512 N 5th St #B, Lafayette, IN 47901 | linkedin.com/in/dandurisrihari | github.com/dandurisrihari

SUMMARY

PhD student and researcher with extensive experience in system software development, specializing in low-level software security, drivers, firmware, operating systems, and runtime libraries. My current research focuses on analyzing the design of the latest edge AI systems to understand different design choices and explore their system security aspects.

EXPERIENCE

Graduate Research Assistant, Purdue University *Aug 2023 - Present*
- Studied the design of Google Coral Dev board, TDA4VM (Texas Instruments), and Tinker Edge R (Asus) Embedded Edge AI accelerators to understand the interaction between the application processor and accelerator hardware.
- Analyzed flow from application level to firmware on the accelerator, Starting from application to Runtime libraries (vendor-specific userspace libraries, and framework runtimes like TensorflowLite, and ONNX), Linux kernel drivers and subsystems, and firmware on accelerator hardware.
- Answered research questions like Open Session, Memory Setup, Sending Request, Receive request, and close session.
- The goal is to see if confused deputy attacks are possible, can a malicious application confuse the accelerator to corrupt the kernel memory or other process memory.
- Found some design-level security vulnerabilities and reported them, Investigation by companies is under progress.

Silicon Firmware Development Engineer (Firmware Security), Intel Corporation *Jun 2021 - Aug 2023*
- Implemented security hardening features in Intel Boot Guard and Trusted Execution Technology.
- Developed various features in BIOS and SINIT ACMs (Intel Authenticated Code Modules), providing BIOS/UEFI security for Intel's silicon products, such as Intel Core processors and PCH chipsets.
- Worked closely with Intel silicon design teams and was responsible for ACM software and BIOS-related issues during the program's planning, development, and validation stages.
- Addressed customer issues and fixed bugs reported in the Common Vulnerabilities and Exposures (CVE).
- Served as point of contact for some of the critical features (x86 paging) in Client BIOS security.
- Ensured validation and mitigation of any risks associated with silicon bugs during the pre-silicon phase.
- Actively involved in planning for potential issues, executing system debugging, and ensuring smooth silicon power-on.

Software Development Intern (Firmware Security), Intel Corporation *Jan 2021 - Apr 2021*
- Implemented startup code in X86 Assembly Language (MASM and NASM).
- Re-structured the Build-system to incorporate support for various Compilers and Intel Crypto Libraries.
- Investigated and implemented hardening features against stack canary attacks.

Embedded Systems Firmware Intern, Praan *May 2020 - Aug 2020*
- Develop prototype firmware on the ST NUCLEO-F303RE development board (ARM 32-bit Cortex-M4).
- Utilized UART, USART, and SPI protocols to collect data from multiple sensors in the RTOS environment.
- Lead the board Bring-up stage and contributed to assessing the software/hardware requirements in the design.
- Working experience with HAL, CMSIS Apis, and FreeRTOS.

Assistant System Engineer-Trainee, Tata Consultancy Services (T.C.S) *Nov 2018 - Mar 2019*
- Worked for Verizon client in the backend development of IoT products (Oracle DB).

System Engineer Intern, Tata Consultancy Services *Nov 2017 - Apr 2018*
- Worked on the prototype project IOT-Autonomous Garbage / Package Collection System.

PROJECTS

Exploring Security Aspects Of Edge AI Accelerators - Purdue
- The current research project analyzes the security threats Edge AI Accelerators pose to Application processors.
- Explore the possibility of confused deputy attacks and the semantic gap between the AI Accelerator and host communication.

Rehosting Embedded Applications As Linux Applications - Purdue
- We are continuing the previous work of the initial idea paper.
- Wrote LLVM passes, optimized the porting pipeline, and different fuzzing techniques.
- The idea is to strip off all architecture-dependent parts of embedded applications and instrument the MMIO access followed by fuzzing to find security bugs.

CUDA Implementation of Convolutional Neural Networks and General Matrix Multiply - Purdue
- Implemented convolution CUDA kernels and optimized them by leveraging shared memory, memory coalescing using cache aligned stride patterns.
- Implemented and optimized GEMM cuda kernels and GEMM using both CUDA memory copies and unified virtual memory and compare the performance

CUDA Implementation of Filtering Noise from Images and Pooling - Purdue
- Implemented CUDA kernels that perform filtering of noise in images.
- Implemented CUDA kernels that perform Max, Min and Average pooling.

CUDA Implementation of AlexNet - Purdue
- Transformed convolutions into matrix multiply operations and implemented fully connected layers.
- Implemented AlexNet paper CUDA kernels by using all the basic blocks like GEMMS.

LLVM Playground - Purdue
- Implemented LLVM passes to develop a simple data flow analysis to detect divide-by-zero errors in C code.
- Improved simple divide-by-zero data flow analysis to handle pointer aliasing and allocated memory
- Build a dynamic analyzer to catch division-by-zero errors at runtime, by having LLVM passes instrument the program.
- Developed an LLVM pass to insert runtime checking and monitoring code into a given program.
- By instrumentation performed division-by-zero error checking and record coverage information for a running program.

Symbolic Execution Playground - Purdue
- Implement a dynamic symbolic execution (DSE) engine that automatically generates inputs to efficiently explore different program paths. Used an LLVM pass to encode C programs into our symbolic interpretation API and Z3 for constraint solving.

USB Drivers in Linux (Embedded C) - ASU
- Wrote a detailed report on USB implementation (including source code review) in Linux kernel (v3.19.8) its usage of the mechanism at kernel and user level and developed a kernel module that hacks and steals the USB device information.

Thread Event Tracing in Zephyr RTOS (Embedded C) - ASU
- Developed a new tracing backend, recorded the data in VCD format. Parsed the data using python script and analyzed.

Linux Kernel v4.19 and x86 based Device Drivers for HC-SR04 ultrasonic distance sensors - ASU
- Developed Linux kernel module to enable user-space device interface for HC-SR04, developed platform driver/platform device infrastructure for HC-SR04 sensors, and Tested the sysfs interface, with a Bash script file.

Generic Netlink Socket and SPI Device Programming (Embedded C) - ASU
- Developed a multithreaded user space program that transfers patterns to LED matrix (MAX7219) using Generic Netlink Socket, developed SPI device driver to asynchronously control LED matrix depending on distance measured.

A Dynamic Stack Dumping in Linux Kernel (Embedded C) - ASU
- Implemented two new system calls to insert and remove dump stack dynamically without re-building the kernel in the execution path of kernel programs by invoking dump_stack() in a pre-handler of the Kprobe.

Custom Bootloader - STM32F303RE-NUCLEO BOARD (Embedded C) - ASU
- Designed a custom bootloader for the STM32Nucleo-F303RE board to update the on-chip firmware through UART. The design will handle the flashing of multiple applications and various options at the time of boot.

Message Passing via Ports in Multiprocessor Operating Systems (Embedded C) - ASU
- Built a module that handles multiple server-clients with blocking receive and non-blocking sends. Built modules like TCB, Semaphores, queues, threads from scratch.

Thread programming and device driver in Zephyr RTOS (Embedded C) - ASU
- Developed HC-SR04 sensor driver with apis sensor sensor_sample_fetch_t, sensor_channel_get_t, sensor_attr_set_t.

EDUCATION

Doctor of Philosophy, Computer Engineering — *Aug 2023 - Present*
Purdue University — GPA: 3.37

Master of Science, Computer Engineering - Electrical Engineering (Embedded systems) — *Aug 2019 - May 2021*
Arizona State University, Tempe, AZ — GPA: 3.84

Bachelor of Technology, Electrical Electronics Engineering — *Aug 2014 - May 2018*
GVPCOE, Jawaharlal Nehru Technological University, Kakinada, India

SKILLS

Programming: C, C++, x86 Assembly, Python, bash scripting, Java
Architecture and Protocols exposure: x86, ARM, SPI, I2C, UART, CAN, PCI, PCIe, BLE
Tools and Development Skills: GNU toolchains, LLVM, MLIR, CUDA, PyTorch, TensorFlow, Multiple tools/utilities for Trusted Execution Environments, TPM, BIOS, platform security, cryptography, Windows Secure Launch, Board bring-up, Tboot, JTAG, SWV, FreeRTOS, Kernel programming, Device drivers, Shell scripting, Bare-metal firmware, debugging, and fault analysis, embedded Linux, DMA, Custom Bootloader, Resource management, Intel silicon products such as next-generation Intel Core processors and PCH chipsets, Intel Galileo Gen 2, NUCLEO, Raspberry Pi, cadence virtuoso, HSPICE, schematics, Git, JIRA, GitHub.
Relevant Coursework: Programmable Accelerator Architectures, Holistic Software Security, Real-Time Embedded Systems, Embedded Operating Systems Internals, Computer Architecture-2, Foundations of Algorithms, Digital Systems and Circuits, Distributed and Multiprocessor Operating Systems, Wireless Networks, Artificial Neural Computation, Mobile Systems Architecture.