Protecting Embedded Systems from Zero-Day Attacks

Professor Stephen Taylor
Thayer School of Engineering at Dartmouth
stnh.email@icloud.com  (603) 727-8945
Research Support

Current Support:
• AFRL ICCyRIS Program – “System-on-Chip Nano-Marshal”
• OSD SBIR Phase II – “Network Security Appliance”

Seedling Support:
• DARPA BAA 14-39 – “Hiding in Hardware”

Dartmouth College:
• DARPA CRASH Program – “Attacking Time”
• DARPA MRC Program – “Resilient Diffusive Clouds”

https://github.com/SCSLaboratory/BearOS
A Quiet Revolution in Embedded Systems

SoC Devices with on-chip FPGA
High-Level Synthesis
Lightweight OCI-compliant Containers

What can these innovations do for Security?
• Reverse Engineering
• Vulnerability Amplification
SoC Devices with on-chip FPGA
High-Level Synthesis
Lightweight OCI-compliant Containers
FPGA – A New Resource Within the Chip’s Security Perimeter

2012:
• Xilinx + Altera Announce ARM SoCs
• Dual 32-bit A9 1GHz
• Xilinx Zynq GA ~‘Q3

2013:
• Altera GA ~‘Q2
• Broad Adoption
• Multiple Awards

2014:
• Xilinx + Altera Announce 64-bit ARM
• Intel Announces XEON

2015:
• Intel buys Altera
• Xilinx Demonstrates UltraScale: Quad 64bit Cortex A53 + Dual 32bit Cortex R4 + Hypervisor

Xilinx Zynq Family

- Co-locate trust-boundary with chip-boundary
- No exposed busses
- Reverse engineering requires exotic techniques (e.g. acid etching)

Mitigates Reverse Engineering

- ZYBO ~ $200

6
FPGA SoC Product Space

Many Embedded Systems will fit inside the FPGA
Bus Mastering from the FPGA Fabric

Processor Peripherals:
- Gigabit Ethernet
- USB
- SDIO, SPI, UART
- DDR & DMA Controllers
- Timers
- GPIO

Shared Peripherals:
- Encryption Engine
- Analog to Digital Converter

General Purpose Ports
- 60 to 140 36 kb BRAMs (240kB-540kB)
- 17k to 53k Lookup Tables

High Performance Ports
- 35k to 106k Flip-Flops

Accelerator Coherency Port
- 28k to 85k Logic Cells

FPGA Fabric
- 17k to 53k Logic Cells

Multiple AXI interconnect options
Hidden Hardware Monitors

Processes have invariants:
- Hash of program code
- Bounded variables:
  ```
  /* 0 <= i <= 10 */
  buffer[i] = '!';
  ```

Processes have behavioral signatures:
- Call graph
- Resource (i.e. network) utilization

Monitor invisible to privileged code running on processor
Monitoring Process Code Integrity using SHA-256

Detect any single-bit change to code
SoC Devices with on-chip FPGA
High-Level Synthesis
Lightweight OCI-compliant Containers
High Level Synthesis

Accelerates Design: C specification
Accelerates Verification: C test bench to HDL test bench automation
Expands Developer Base: Accessible to C Programmers

Rapid Technology Evolution with *Separable Accreditation*
# Cost of High Level Synthesis

<table>
<thead>
<tr>
<th>Metric</th>
<th>SW</th>
<th>HW AXI-Lite</th>
<th>HW AXI</th>
<th>HW HLS AXI-Lite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Management Code Size</td>
<td>491</td>
<td>111</td>
<td>54</td>
<td>111</td>
</tr>
<tr>
<td>Core Clock Cycles</td>
<td>3100</td>
<td>2800</td>
<td>1500</td>
<td>4800</td>
</tr>
<tr>
<td>Flip-Flops</td>
<td>-</td>
<td>1%</td>
<td>4%</td>
<td>3%</td>
</tr>
<tr>
<td>Lookup Tables</td>
<td>-</td>
<td>1%</td>
<td>7%</td>
<td>5%</td>
</tr>
<tr>
<td>Slices</td>
<td>-</td>
<td>1%</td>
<td>9%</td>
<td>8%</td>
</tr>
<tr>
<td>BRAM 18</td>
<td>-</td>
<td>1%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BRAM 36</td>
<td>-</td>
<td>-</td>
<td>5%</td>
<td>6%</td>
</tr>
<tr>
<td>Max FPGA Clock</td>
<td>-</td>
<td>120 MHz</td>
<td>105 MHz</td>
<td>159 MHz</td>
</tr>
</tbody>
</table>

**HLS Development / Maintenance “Ease of Use” Costs:**
- ~3-5x Resource Utilization Cost
- ~2x Performance Cost
- HLS yielded higher max clock

Dahlstrom & Taylor, “Migrating an OS Scheduler into Tightly Coupled FPGA Logic to Increase Attacker Workload” MILCOM 2013.
Hardware Monitor Performance

Hardware Monitor exposes a “hashing” signal – logic high when hashing text segment (PMOD JC, Pin 1 = Hardware Monitor on Core 0, Pin 2 = Hardware Monitor on Core 1)

<table>
<thead>
<tr>
<th>Text Segment Size</th>
<th>Time Per Hash (uS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 kilobytes</td>
<td>1ms</td>
</tr>
<tr>
<td>100 kilobytes</td>
<td>10 ms</td>
</tr>
<tr>
<td>1 megabyte</td>
<td>100 ms</td>
</tr>
<tr>
<td>10 megabytes</td>
<td>1 s</td>
</tr>
</tbody>
</table>

**Key Findings:**
~10 megabytes/sec per monitor
~40x speedup over software
Memory Interface Independent (compute bound)
HLS Provides high level IP abstraction; still far from transparent
Monitoring Critical Processes

- Cortex-A9 800 MHz
- 32k L1 I/D Cache
- 512k L2 Cache
- 256k Static RAM
- 1 GB DDR SDRAM
- Non-critical Process
- Critical Process
- Non-critical Process
- Non-critical Process
- Kernel
Hardware Monitoring with Code Refresh

DARPA BAA-14-39 Secure PLC Prototype

• Hidden Code Monitoring on Process Code
• Hardware *refresh* from a gold-standard & secure boot

**Mitigates Zero-Day Attacks & Persistence**
Mitigating Persistence

Non-Deterministic Refresh to Deny Persistence

Code Refresh from Encrypted, Signed, Read-only Base-of-Trust
MicroArx DT-Series Modbus Diodes & Apiotics

- Passive data tap, monitoring Modbus data
- No changes to existing Modbus network
- Extract data from the network without introducing additional latency
- Forward that data to a server of your choice via WiFi, Cellular, or Ethernet

MicroArx.com

Apiotics.com

Raspberry PI

TI TM4C Series

Zynq Boards

RailsConf 2018
Pittsburgh
April 17-19

Control of Embedded Systems through Ruby WebApps
SoC Devices with on-chip FPGA
High-Level Synthesis
Lightweight OCI-compliant Containers
Open Containers Initiative (OCI)

Standards for container specification and management used by Docker, Resin.io, and many other container software vendors

**Enhanced scaling**
More efficient resource sharing than virtualization (when use same base OS)
Cloud hosting services (e.g. Amazon ECS)

**Simplify application maintenance**
Associate OS version, patches, language packs, binaries, libraries, credentials with a given containerized application
Mitigate maintenance interference between applications – e.g. library version mis-matches

**Improve portability and distribution**
Easy to move between dev, test, and production
Speeds application development

www.opencontainers.org
AFRL OpenSource Nanomarshall
An OCI Compliant Runtime

/* OCI compliant container interface – the following operations are REQUIRED */

/** \brief Create a container with the given ID and program bundle */
nm_error_t container_create(id_t *id, uint8_t *bundle);

/** \brief Start the container with the specified ID */
nm_error_t container_start(id_t *id);

/** \brief Query the state of the container of the specified ID */
nm_error_t container_query_state(id_t *id, container_state_t **ctr_state);

/** \brief Kill the container with the specified ID */
nm_error_t container_kill(id_t *id, int32_t signal);

/** \brief Delete the container with the specified ID */
nm_error_t container_delete(id_t *id);

Hardware Base-of-Trust for Containers that Replaces a Hypervisor
Core Affinity API

/** brief Set CPU Affinity – Pin a container to a core */
container_set_cpu_affinity (id_t *id, uint32_t affinity_flags);

Containers associated with one or more physical cores to create **Container Security Domains** with dedicated processing resources.
Container Security RoadMap

Diversity

Memory

Variant #1

Variant #2

Variant #3

Variant #4

Disk Image

Disk Image

Kanter and Taylor

“Attack Mitigation through Diversity”
MILCOM 2013

“Diversity in Cloud Systems through Runtime & Compile-Time Relocation”
IEEE-HST 2013

Memory Encryption

SoC

Processor

On-chip RAM

Hardware Crypto Engine

Off-Chip Memory

P(N)

P(k)

P(1)

Henson and Taylor

“Beyond Disk Encryption: Protection on Security Enhanced Commodity Processors”
Int. Conf. on Applied Crypto and Net Security ‘13

“Attack Mitigation through Memory Encryption Of Security Enhanced Commodity Processors”
Int. Conf. on Information Warfare and Security ‘13

Hardware Hiding

Dahlstrom and Taylor

“Migrating an OS Scheduler into FPGA Logic to Increase Attacker Workload”
MILCOM 2013, Proceedings of

“Hardware-Based Code Monitors on Hybrid Processor-FPGA SoC Architectures”
MILCOM 2015, Proceedings of
Summing Up

Own the Base-of-Trust in Hardware

- Mitigate Reverse Engineering (single-chip SoC & on-chip encryption)
- Mitigate Zero-day Exploits (hidden code monitoring)
- Mitigate Persistence (code refresh)
- Mitigate Re-infection & Vulnerability Amplification (diversity)
- Mitigate Insider Tampering (journaling, out-of-band management, 2-factor authentication, anti-tamper)
- Rapid Technology Evolution & Accreditation (HLS, Containers & Apiotics)