RISC-V Security

Arm® TrustZone® Technology vs RISC-V MultiZone™ Security

April, 2019
Did you feel the Earth Shake in Feb?

Base Score Metrics

Exploitability Metrics

- **Attack Vector (AV)**
- **Attack Complexity (AC)**
  - Low (C:L)  High (C:H)
- **Privileges Required (PR)**
  - None (PR:N)  Low (PR:L)  High (PR:H)
- **User Interaction (UI)**
  - None (UI:N)  Required (UI:R)
- **Scope (S)**
  - Unchanged (S:U)  Changed (S:C)

Impact Metrics

- **Confidentiality Impact (C)**
  - None (C:N)  Low (C:L)  High (C:H)
- **Integrity Impact (I)**
  - None (I:N)  Low (I:L)  High (I:H)
- **Availability Impact (A)**
  - None (A:N)  Low (A:L)  High (A:H)
Security Through Separation

Systems are composed of a stack of 100s of libraries

<table>
<thead>
<tr>
<th>Functional Code Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Stack</td>
</tr>
<tr>
<td>Root of Trust</td>
</tr>
<tr>
<td>Crypto Libraries</td>
</tr>
<tr>
<td>User Code</td>
</tr>
</tbody>
</table>

- stack
- heap
- uninitialized data (bss)
- initialized data
- text

0x5 HEX-Five Security
Security Through Separation

Arm® TrustZone® Technology

Non-Secure
RichOS

Trusted OS
and Apps

Non-Secure
Stack

Non-Secure
Heap

Non-Secure
Uninitialized
Data (BSS)

Non-Secure
Text

Trusted
Stack

Trusted
Heap

Trusted
Uninitialized
Data (BSS)

Trusted
Text

RISC-V MultiZone™ Security

Root of
Trust

Network
Stack

Crypto
Libraries

OTA Update

Rich OS

Linux / RTOS

Each Zone
Compiled and
Linked Separately

Compiled
Non-Secure
Privileged
Stack

Compiled
_non-Secure
Pluggable
Heap

Compiled
Non-Secure
Uninitialized
Data (BSS)

Compiled
Non-Secure
Text

Compiled
Trusted
Stack

Compiled
Trusted
Heap

Compiled
Trusted
Uninitialized
Data (BSS)

Compiled
Trusted
Text

Compiled
Trusted
Uninitialized
Data (BSS)

Compiled
Trusted
Text

Compiled
Trusted
Uninitialized
Data (BSS)

Compiled
Trusted
Text

Arm and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Arm® TrustZone® Trusted Firmware

Secure InterZone™ Communications – no shared memory

Secure MultiZone™ nanoKernel – boot room
Key Components of SoC Platform Security

- **Trusted Execution Environment**
  - Hardware enforced separation between code, data and memory mapped resources

- **Root of Trust**
  - Tamper-proof storage for unique ids and certificates
  - Tamper-proof storage for secrets (priv keys)

- **Secure Boot**
  - Prevent boot of unauthenticated code

- **Tools**
  - Transparent, intuitive and integrated with tools and processes the designer is already familiar with
Hardware Comparison
Arm® TrustZone® Technology vs. RISC–V Privileged Architecture
RISC-V Security Toolchest

Simplicity and Openness
- Simple and Modular ISA
- Open ISA and open cores
- Transparency and high assurance
- End of security through obscurity

Execution Privilege Levels
- Machine – always present
- Supervisor – OS’s (e.g., Linux)
- (Hypervisor – work in progress)
- User - Applications

User-mode Interrupts (“N”)
- Optional extension
- Interrupts delegated to userland
- Hardware transfers control directly to U-mode
- Intended for securing constrained embedded devices
  - M + U mode

Physical Memory Protection
- Whitelist-based
- Number of PMP entries can vary
- Configurable by the M-mode
- Controls accesses of U- and S-mode to memory
Hardware Security

Arm® TrustZone® for Armv8-A
Linux/Android Systems

Arm® TrustZone® for Armv8-M
RTOS or Bare Metal Systems

RISC-V Privileged
Specification v1.10

Two Domains Hardcoded in Hardware

Optional Memory Protection Controller
Optional Peripheral Protection Controller

Hardware Enforced Software Defined Domains

Arm and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
Software Comparison
OP-TEE and PSA vs. MultiZone™ Security
Armv8-A / OP-TEE Software Model

Two worlds – Mobile Phone / Gateway

- Code Size: 244kB\(^1\); RAM Requirements: 32MB\(^2\)
- Configuration and tools are outsourced from multiple Arm Ecosystem partners

“...the design complexity associated with correctly implementing [security] technologies like memory protection units (MPUs) often results in them not being used at all.

Brandon Lewis, Editor-in-Chief, Embedded Computing Design

Source: https://www.linaro.org/blog/op-tee-open-source-security-mass-market/

1. From https://www.op-tee.org/faq/

Awaiting detailed feedback from Mr. Joseph Yiu at Arm on more precise numbers.
Armv8-M PSA Software Model

IoT Endpoint / RTOS – Smart Watch / Sensor

- Boot Loader: 3,366 lines / 38kB
- Kernel Size: 6,596 lines / 75kB
- Solution ships with TCP/IP stack
- HW is just rolling out, L1 of PSA software is available
  higher levels pending
- Unified Development / Debug requires 3rd party tools such as Kiel MDK or IAR EWARM

Source: https://developer.arm.com/products/architecture/security-architectures/platform-security-architecture

Arm and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
2. trustedfirmware.org repo
RISC-V MultiZone™ Security Software Model

Zero Trust Model for Unlimited # of Equally Secure Worlds

- **Tiny**: 0 stage Boot Loader: 600B, nanoKernel Size: 1.6kB
- Designed for Formal Verification
- **Simple**: existing open source tools – gcc / gdb and Eclipse IDE
  All security settings in a single flat file
- **Fast**: Context Switch ~100 instructions, <0.01% of core cycles
- **Universal**: Works on standard RISC-V Cores; runs your existing code
- **Open Source** – available on github.com/hex-five
# Copyright(C) 2018 Hex Five Security, Inc. - All Rights Reserved

Tick = 10 # ms

Zone = 1 #
irq = 16, 17, 18 # BTN0 BTN1 BTN2
base = 0x20410000; size = 64K; rwx = rx # FLASH
base = 0x80001000; size = 16K; rwx = rw # RAM
base = 0x10025000; size = 0x100; rwx = rw # PWM
base = 0x10012000; size = 0x100; rwx = rw # GPIO
base = 0xC000000; size = 0x400000; rwx = rw # PLIC

Zone = 2 #
base = 0x20420000; size = 64K; rwx = rx # FLASH
base = 0x80005000; size = 16K; rwx = rw # RAM
base = 0x60000000; size = 8K; rwx = rw # XEMACLITE

Zone = 3 #
base = 0x20430000; size = 64K; rwx = rx # FLASH
base = 0x80000000; size = 4K; rwx = rw # RAM
base = 0x20000FF8; size = 0x8; rwx = r # RTC
base = 0x10012000; size = 0x100; rwx = rw # GPIO

Zone = 4 #
base = 0x20440000; size = 64K; rwx = rx # FLASH
base = 0x8000A000; size = 4K; rwx = rw # RAM
base = 0x10013000; size = 0x100; rwx = rw # UART
base = 0x10012000; size = 0x100; rwx = rw # GPIO
MultiZone™ Security Live Demo

<table>
<thead>
<tr>
<th>Zone #1</th>
<th>Zone #2</th>
<th>Zone #3</th>
<th>Zone #4</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTOS</td>
<td>TCP/IP Stack</td>
<td>Key Management</td>
<td>TEE CONSOLE</td>
</tr>
<tr>
<td>[FreeRTOS]</td>
<td>[picoTCP]</td>
<td>[wolfSSL]</td>
<td>[MultiZone]</td>
</tr>
<tr>
<td>GPIO / IRQs</td>
<td>ETHERNET</td>
<td>OTP</td>
<td>USB / UART</td>
</tr>
</tbody>
</table>

InterZone™ Secure Communications

Hex Five MultiZone™ nanoKernel

X300 Bitstream (RV32ACIMU + Ethernet)

Ethernet

Untrusted Network

SPI / USB

USB / UART

0x5 HEX-Five Security
Hex Five MultiZone™ Security

Hex Five Security, Inc. is the creator of MultiZone™ Security, the first trusted execution environment for RISC-V. Hex Five patent pending technology provides policy-based hardware-enforced separation for an unlimited number of security domains, with full control over data, code and peripherals. Contrary to traditional solutions, Hex Five MultiZone™ Security requires no additional cores, specialized hardware or changes to existing software. Open source libraries, third party binaries and legacy code can be configured in minutes to achieve unprecedented levels of safety and security.