Stratos and Morello?

Joakim Bech - 2020-12-15

Capability based security systems

What it is

- Capabilities are unforgeable tokens
- Access to resources goes via capabilities

Idea is to

- limit the damage when software misbehaves
- Implement principle of least privilege
  - I.e., use no more privileges than needed
- Implement principle of intentional use
  - I.e., the answer to the “confused deputy” problems. Avoid trick a more privileged program into misusing its authority
- Formally verify the system

Existing software

There are software implementations / variants of this, Capsicum (FreeBSD), Linux capabilities, seccomp etc.

CHERI

What is it?

- A project that has been running for roughly 10 years already …
- Architectural protection model
- Leverage hardware to implement a capability based system
- ISA extended to include security primitives

Idea is to

- Mix software and hardware to come up with a capability based system that is more robust to well known attacks, as for example
  - Buffer overflows
  - Return oriented programming (ROP)
  - And many other known vulnerability classes …
- Implement compartmentalization (high granularity)
- Use hardware to get the needed performance and atomicity

CHERI #2 - 128-bit capability

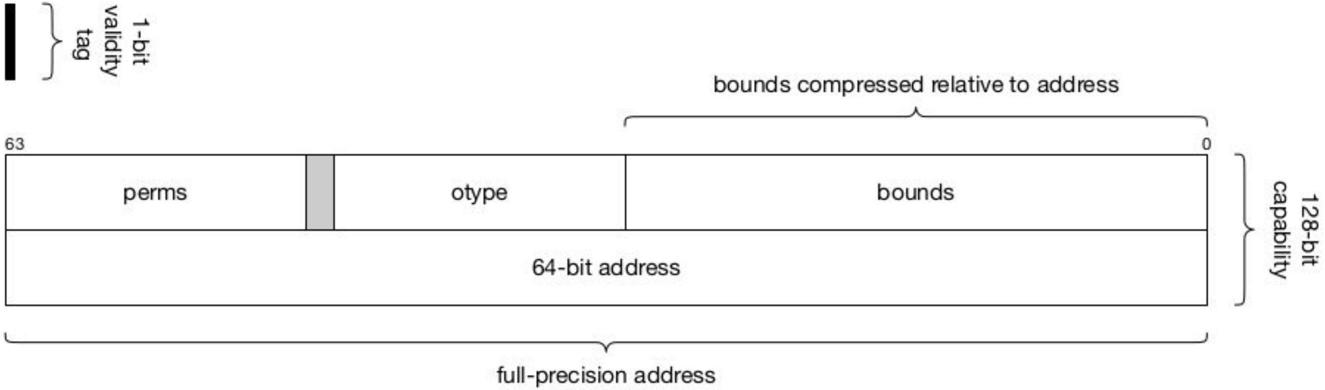


Image source: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-941.pdf

- A capability in CHERI embeds
  - Permission bits (perms)
  - Can be used to “seal” capabilities (otype)
  - Bounds that pointers can operate on (bounds)
  - A traditional pointer (at the last half of the capability)
  - A tag bit telling whether the capability is trustworthy or not

CHERI #3 - compilation modes

Pure-capability code

- Implements all C/C++ pointer types, as well as all implied pointers (e.g., return addresses, the stack pointer, and so on) using capabilities.
- ABI disruptive change

Hybrid-capability code

- Implements pointer types using integers by default, interpreted with respect to a global default data capability (DDC) able to address code, globals, heap, and stack(s).

CHERI #4 - Software

<u>How does software deal with this?</u>

- Toolchains: Involves lots of changes to LLVM, LLDB, GCC, GDB etc
- In short all software must go through a “CHERIfication” to be able to leverage all this
  - Add / user compiler intrinsics
  - Change pointers to capabilities
  - Adapt syscall layers
  - Do pure- or hybrid-capability implementations

DSbD - Digital Security by Design

- It’s a (UK) government funded project headed by UK Research and Innovation (UKRI)

- Quote from the UKRI DSbD page:

  > *“The ISCF Digital Security by Design challenge aims to radically update the foundation of the UK’s insecure digital computing infrastructure. The challenge will:*
  >
  > - *Increase cyber security for businesses, government and the wider public and economy;*
  > - *Increase productivity to the UK through reduction of days lost to cyber-attacks;*
  > - *Make the UK market-leader through new capabilities fostering the trust that is necessary for successful adoption of future digital services in areas such as artificial intelligence (AI) and the Internet of things (IoT).”*

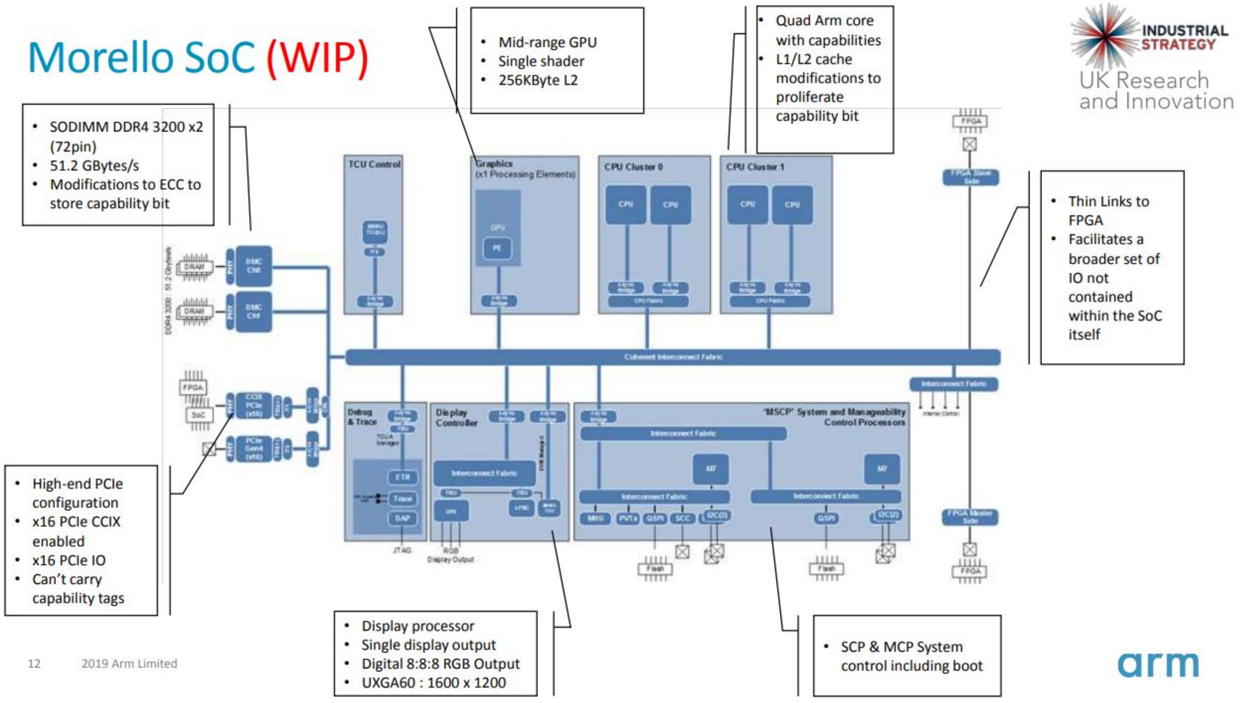
The Arm Morello project

- On a high level: Implement and run CHERI on Armv8-A
- Focuses on Linux kernel and user space (Android “nano” environment)
  - Think framework enablement and commonly used tools
    - Python, Ruby, Go, QT ...
    - I.e., to be able to attract other people than researchers, it must be possible to engineers to convert and run their scripts, binaries etc without too much work
- Firmware, boot loaders and secure side (TrustZone) is low priority for the moment in the Morello project

The Arm Morello board

- Prototype board based on Arm Neoverse N1
- Released to partners somewhere in Q3 2021
- Virtual platforms - Arm Morello FVP will be used until hardware exists
    - Morello FVP will be released to the public in October 2020

The Arm Morello board



Morello SoC (WIP)

- SODIMM DDR4 3200 x2 (72pin)
- 51.2 GBytes/s
- Modifications to ECC to store capability bit

- Mid-range GPU
- Single shader
- 256KByte L2

- Quad Arm core with capabilities
- L1/L2 cache modifications to proliferate capability bit

- Thin Links to FPGA
- Facilitates a broader set of IO not contained within the SoC itself

- High-end PCIe configuration
- x16 PCIe CCIX enabled
- x16 PCIe IO
- Can't carry capability tags

- Display processor
- Single display output
- Digital 8:8:8 RGB Output
- UXGA60 : 1600 x 1200

- SCP & MCP System control including boot

12 2019 Arm Limited

Image source: https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/cheri-morello.html

Food for thought - Stratos on Morello?

Motivation?

- Morello focus is on the non-secure side
- It’s about framework enablement
- Many of the deliverables for Stratos is about framework improvements on the non-secure side (i.e., a plain Linux environment)
- Seems like making Stratos ready for Morello could be worthwhile
    - Combine Morello project goals with areas of interest to Linaro members
- In short: Hardening hypervisors

Challenges and risks?

- “CHERIfication” is a bit invasive, so it won’t be easy to share code base
- The Morello project is a research/prototype project, i.e., no guarantees that it’ll be successful
- Morello hardware can be a challenge

Resources

- **University of Cambridge**
  - Main page:
    http://cheri-cpu.org
  - Morello page:
    https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/cheri-morello.html
- **UKRI**
  - Digital Security by Design
    https://www.ukri.org/innovation/industrial-strategy-challenge-fund/digital-security-by-design
  -
- **Arm**
  - Arm Morello program
    https://developer.arm.com/architectures/cpu-architecture/a-profile/morello
  - Morello Architecture specification (pdf)
    https://documentation-service.arm.com/static/5f7460641b758617cd95ab98
  - Neoverse N1
    https://developer.arm.com/ip-products/processors/neoverse/neoverse-n1

Thank you