UNIVERSITY OF CAMBRIDGE

Computer Science & Technology

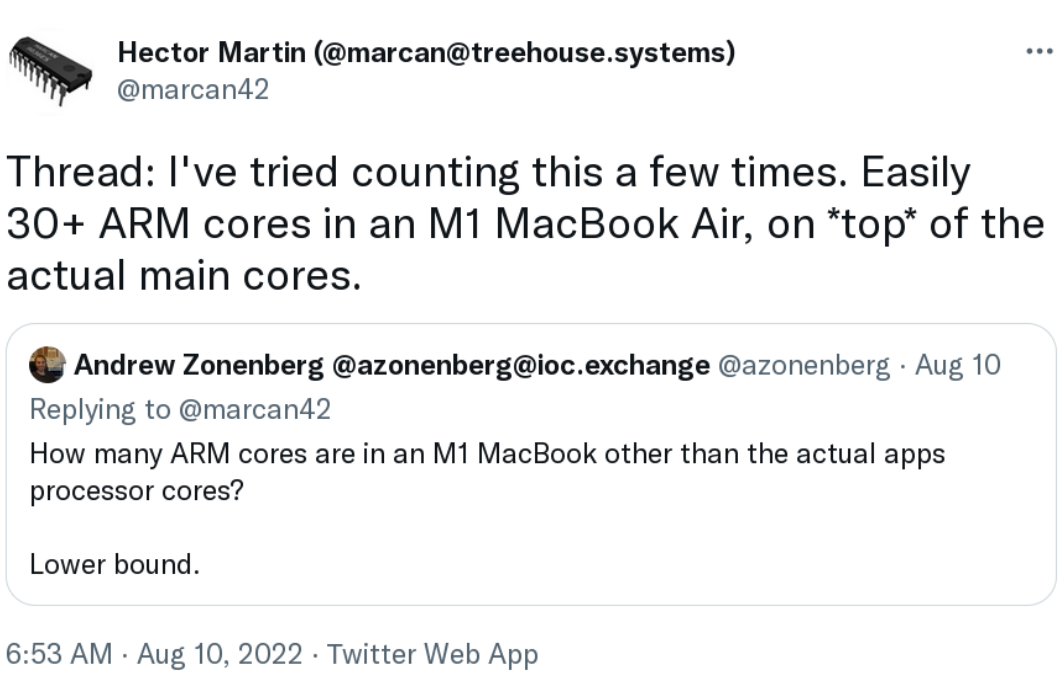
IOSEC: Protection and Memory Safety for Secure I/O

A. Theodore Markettos, **Simon W. Moore**, Robert N.M.Watson

Funded under the RISE initiative by EPSRC grant EP/R012458/1

Input/output and platform security

- Computer hardware is much more complicated than most people imagine
- What do all these cores do? How do we know they are secure?



Hector Martin (@marcan@treehouse.systems)
@marcan42

Thread: I've tried counting this a few times. Easily 30+ ARM cores in an M1 MacBook Air, on *top* of the actual main cores.

Andrew Zonenberg @azonenberg@ioc.exchange @azonenberg · Aug 10
Replying to @marcan42
How many ARM cores are in an M1 MacBook other than the actual apps processor cores?

Lower bound.

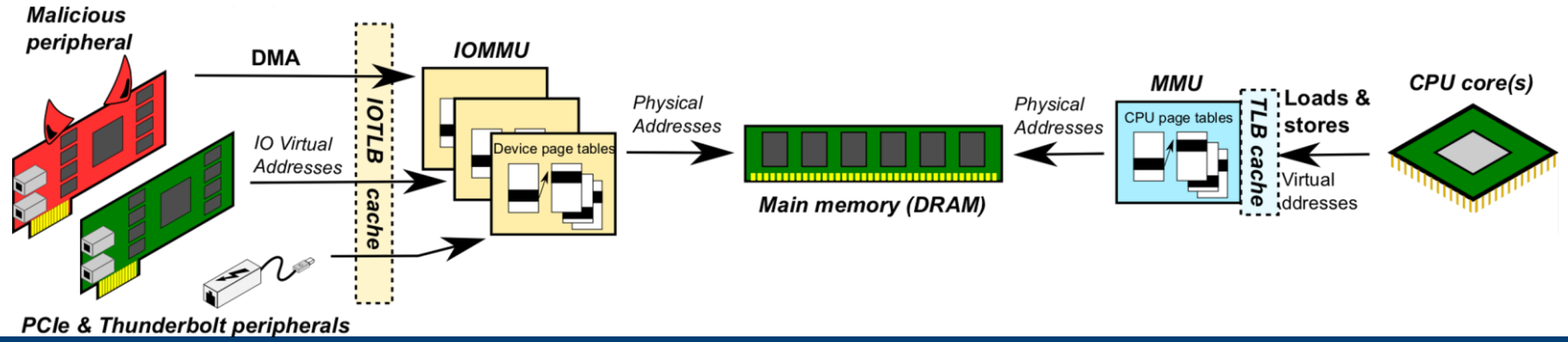
6:53 AM · Aug 10, 2022 · Twitter Web App

Peripheral DMA... the keys to the kingdom

- Many peripheral cores have 'Direct Memory Access' (DMA)
  = peripheral can directly access system memory
- DMA is a necessity for high performance I/O
- System memory is also where all your secrets live
- Full access to read or change system memory means complete system takeover
- Thunderbolt: allows hotplug of DMA-capable peripherals into your computer
- Thunderbolt over USB-C: every charger or projector could try to do DMA
- Bad peripherals can be very, very bad...

The IOMMU to the rescue!

- The Input/Output Memory Management Unit (IOMMU) interposes on memory access from peripherals
- Combines *address translation* and *memory protection*
- Can prevent access to memory the peripheral is not supposed to access
- MMU isolates software *processes*
- IOMMU isolates hardware *peripherals*
- Original design for connecting peripherals to virtual machines
  - Protection from malicious peripherals was an afterthought



Struck by a Thunderclap

- We built a research platform for exploring IOMMU protection in existing OSes
- Thunderclap = an FPGA running a software model of a network card
- Connected via PCIe or Thunderbolt (eg malicious docking station)
- Allows us to explore how much malice a real device can do
- OS sees us as a real device, sets up IOMMU so we can access memory
- Now we can (mis-)interact with the OS/driver attack surface
- How bad can things be?
- IOMMU was barely used and vulnerable across Windows, MacOS, Linux, FreeBSD
- Root shells, VPN snooping, full IOMMU bypass – lots of fun!
- See the Thunderclap paper for more details, and: http://thunderclap.io/

State of the art of protection (2019)

- Windows 10: Entirely unprotected from malicious DMA
- MacOS: tries to use the IOMMU
  - All peripherals can see the memory used by all other peripherals
  - In network packet memory are kernel function pointers...
  - We can change them and launch a root shell
- FreeBSD: the same as MacOS
  - The IOMMU is not enabled by default
  - When it is, the same attack works
- Linux: different but worse
  - Function pointers are protected, but VPN plaintext isn't
  - Simply need to set a bit in the PCIe packets and can entirely disable the IOMMU

Thunderclap outcomes

- USB Implementer's Forum picked up our work and made IOMMU use mandatory in the USB 4 specification
  - Now in shipping products
- Much more awareness across the community of malicious peripherals
- Thunderclap platform open sourced and used by companies for exploring IOMMU attacks
- Windows implemented 'Kernel DMA protection' and added support for the IOMMU

IOMMU state of the world (2022)

- OS' IOMMU implementations have seen some efforts at improvement
- Windows kernel DMA protection exists, but in practice can be hard to turn on
- IOMMUs being used more, but reluctance to use the IOMMU universally due to performance costs
  - Every memory access requires to be translated
  - Generates extra page table traffic and translation caches (IOTLB) are never big enough
  - Increases system and device complexity
    - Pixel 6 phone has 28 separate IOMMUs!

Beyond the IOMMU?

- Thought experiment: can we use CHERI capabilities for I/O?
  - CHERI Capability = bounded memory region with permissions and provenance
- Prevent software generating pointers to things it is not allowed to access
- If the hardware is trustworthy, it can enforce that software obeys the protection model
- If the hardware isn't trustworthy, interpose something that holds the references in a trustworthy way

Small systems

- IOMMUs are too expensive for microcontroller-scale devices
- Many memory accesses but relatively few references
- Peripheral can support secure reference (CHERI capability) manipulation
- Or a small table-like structure can hold capabilities
  - accesses from untrustworthy peripherals are constrained relative to these regions
- Published a paper on the 'small device' model (HASP 2020)
- Ongoing work exploring the issues in practice
  - Everyone's microcontroller software stacks are very different (and proprietary)
  - Hard to evaluate in a way that's meaningful more generally
  - Conversations with industrial folks about their workloads

Large systems

- What does this mean for a server or a phone with a CPU, GPU, neural processor, camera, NVMe, Wi-Fi, etc?
- Much more memory communication, more complex patterns
- IOMMU has a dual role
  - Address translation – deals with virtualisation, memory fragmentation
  - Memory protection – prevent malicious DMA
- Separating translation and protection
  - IOMMU for translation
  - Capabilities for protection

Progress – large systems

- Evaluation of different protection models in a large system needs work at all levels of the stack
  - CPU, peripherals, interconnect, firmware, device driver, OS, etc
  - Performance is a key metric, so it needs to be performance-realistic
  - Needs rich applications to accurately model effect on performance
  - Very hard to do such experiments piecemeal
- Developing a platform for experimentation of models across the stack (joint work with CAPcelerate project)
  - Arm Morello CPU (no capabilities in the I/O system) / CHERI RISC-V CPU (capability aware I/O)
  - Capability-aware peripherals
  - Interconnect handling capabilities and IOMMU
  - Capability-aware device drivers

Conclusions

- Be afraid of your I/O!
- System-wide protection is a hard problem as it requires both security and performance
- We believe there are better ways to achieve protection *and* performance
- Further information…

Thunderclap
thunderclap.io

CHERI
www.cl.cam.ac.uk/research/security/ctsrd/cheri/

Digital Security by Design
www.dsbd.tech