

# IOSEC: Protection and Memory Safety for Input/Output Security

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

Second Annual RISE Conference

London, 21 November 2019

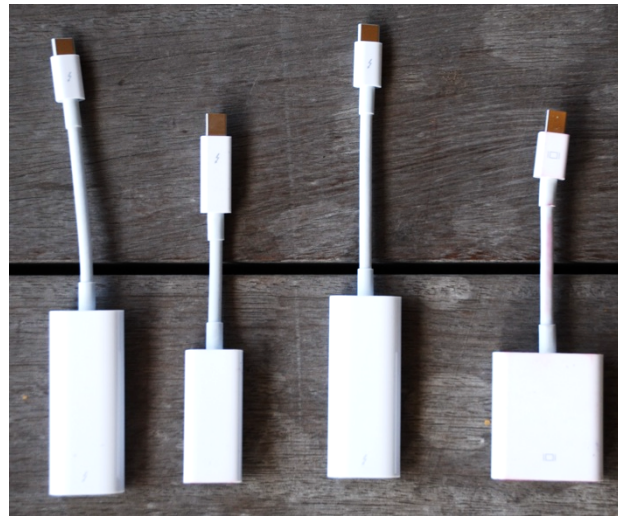
Funded by EPSRC under the RISE initiative (ref: EP/R012458/1)

# Thunderclaps and lightning...

- Background to I/O security
- The Thunderclap FPGA research platform
- Thunderclap attacks
  - Markettos et al, *Thunderclap: Exploring Vulnerabilities in Operating System IOMMU Protection via DMA from Untrustworthy Peripherals*, in NDSS 2019
- Media reaction
- Interaction with vendors
- Standardisation progress
- Next steps...

# Smaller laptops, more external peripherals

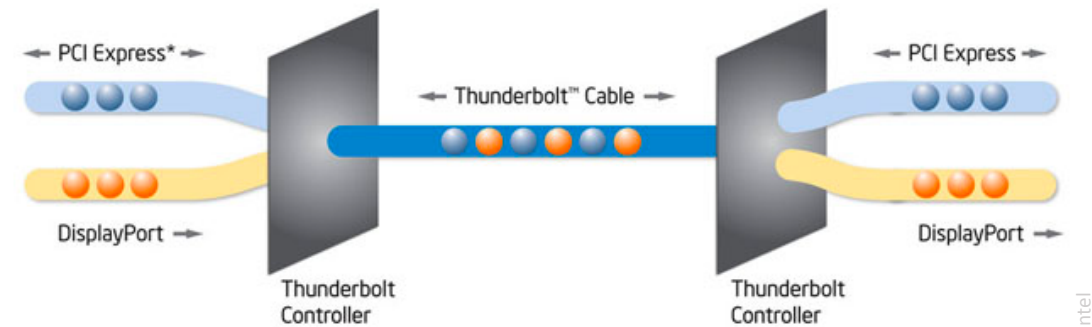
- Laptops getting smaller, more devices are going external
  - Chargers, dongles, docking stations
  - Common to borrow external peripherals (power, dongles, displays) from others
- Performance is increasingly more of a constraint
- Security?



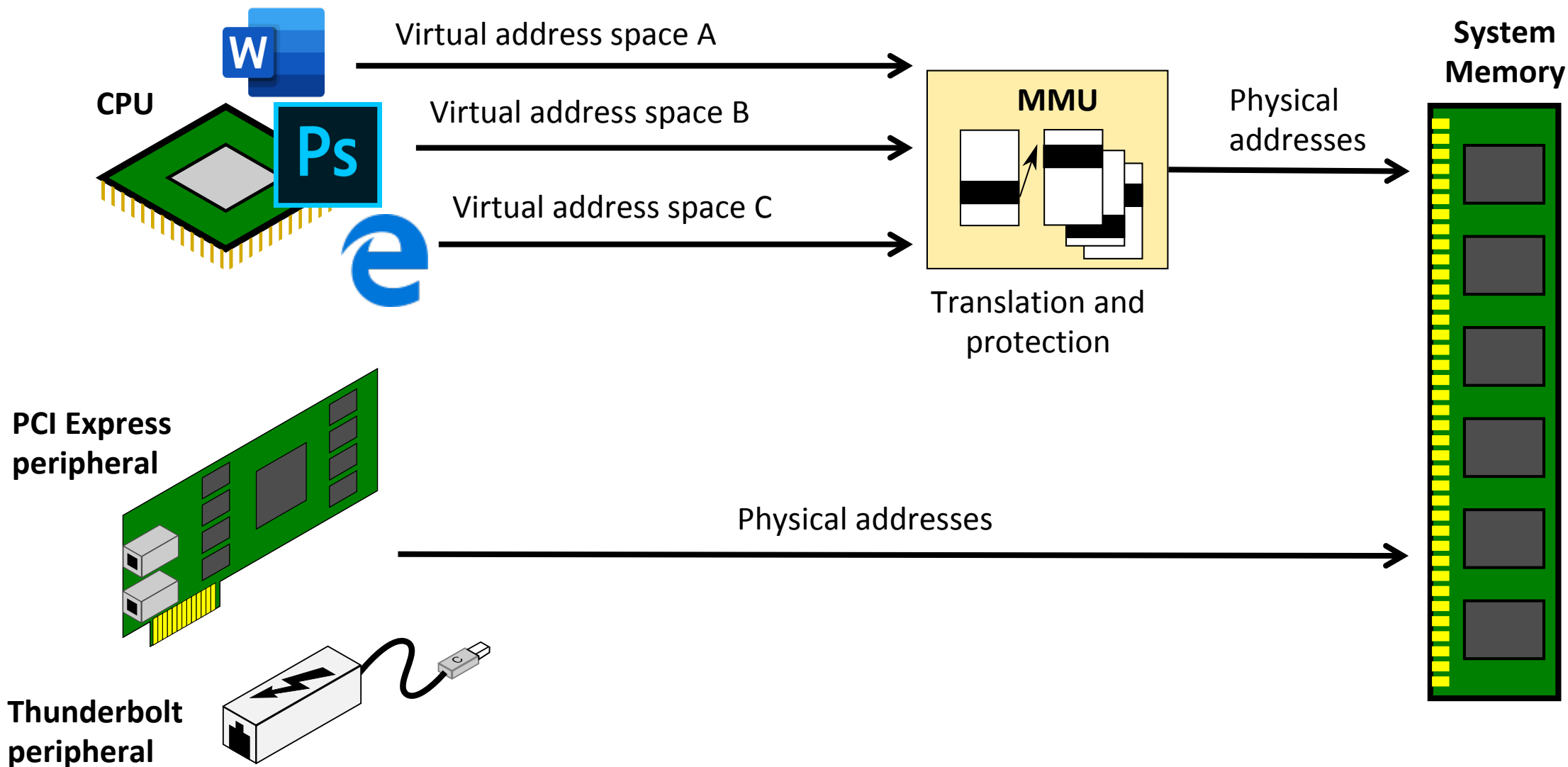
Wikimedia/Amin CC-BY-SA-4.0

# Security?

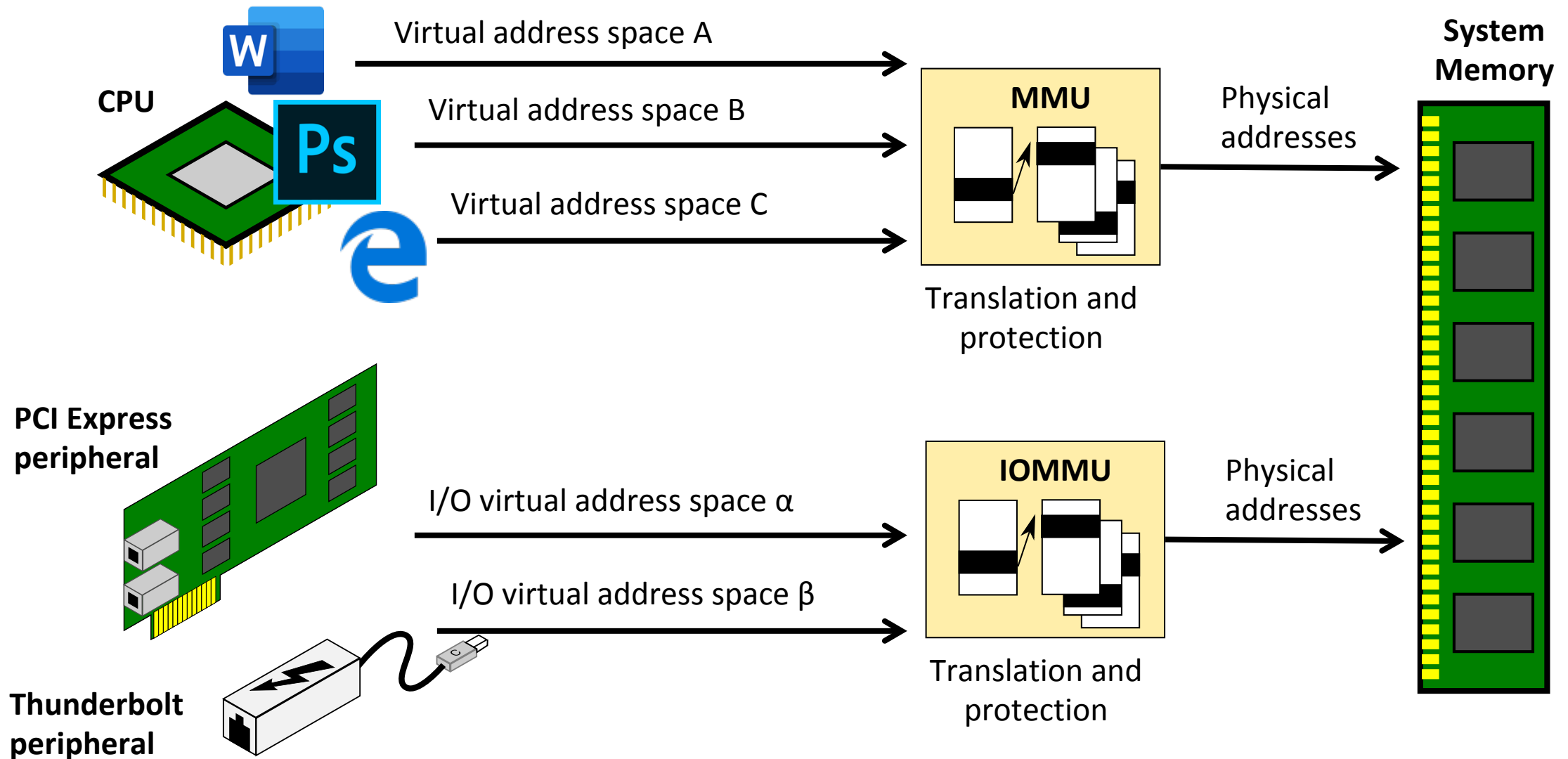
- USB is a packet-based protocol
  - like the internet, only little scrutiny
  - attackers craft bad messages
  - reprogram devices to send bad messages
  - trip up and exploit device drivers
  - defences: firewalls, filtering, fuzzing etc
- Thunderbolt carries PCI Express, which is a memory-based protocol
  - DMA: *direct memory access*
  - access the full state of your machine
  - read your files, your passwords
  - inject arbitrary code...
- USB Type C carries both, and power and video, on the same cable



# Memory Management Unit: process isolation



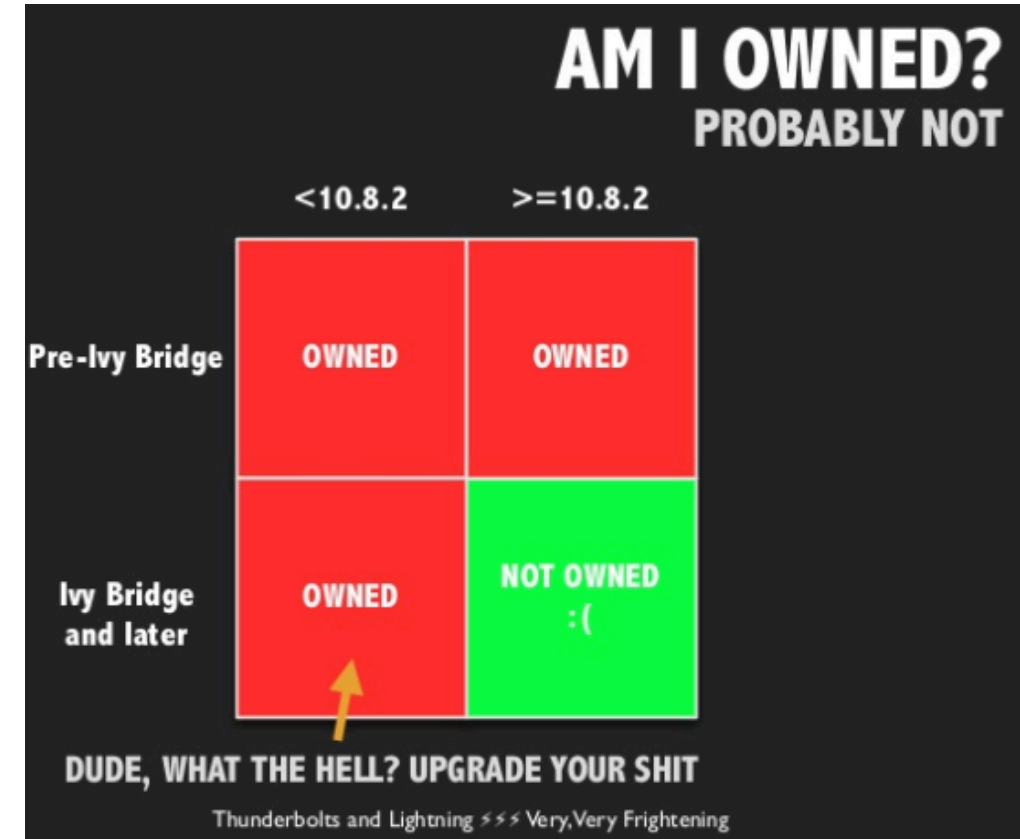
# I/O Memory Management Unit: device isolation





# Attacks from devices

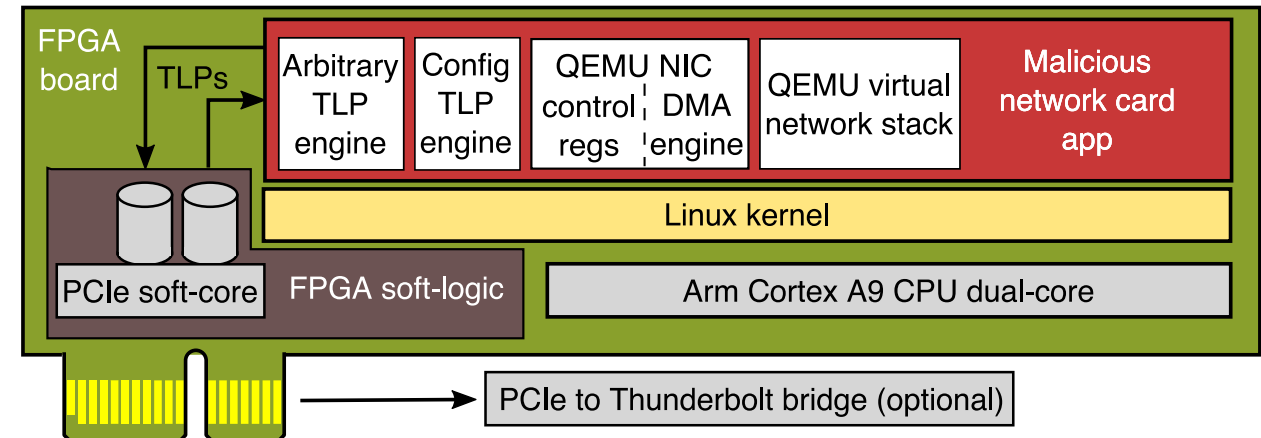
- General understanding before our work: “when the IOMMU is enabled, attacks are foiled”
  - these are simple memory-probing attacks
  - no interactions with driver or kernel
- actually, the attack surface is much more nuanced
- what attack surface does a real I/O device have?
  - what accesses can it make?
  - how does it interact with the device driver stack?
  - as the OS increasingly trusts it, what extra vulnerabilities does it open up?



snare and rzn, *Thunderbolts and Lightning* – Very Very Frightening (2014)

# Thunderclap: a research platform for I/O security

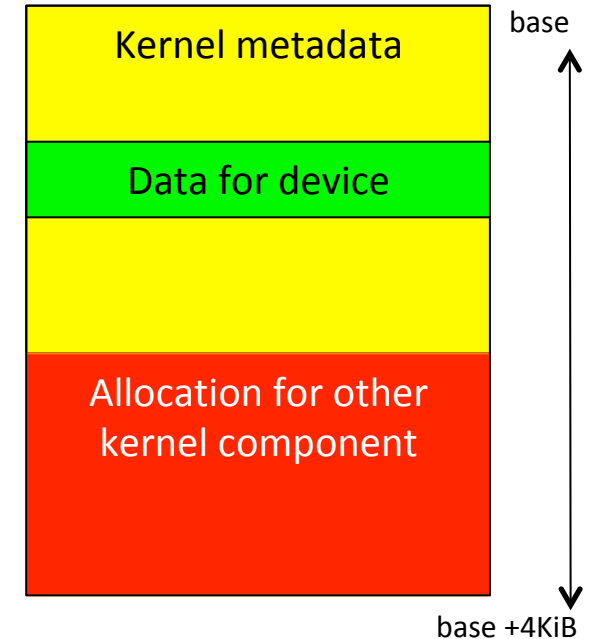
- We built a fake network card (NIC):
  - software device model of an Intel E1000 PCIe ethernet card from QEMU
    - software = easy to change, add malicious behavior
  - run it on a CPU on an FPGA (Arm Cortex A9 on Intel Arria 10, running Ubuntu)
    - FPGA logic can send and receive arbitrary PCIe packets
    - QEMU model responds to PCIe packets and generates 'DMA' like a real NIC
  - runs on FPGA dev boards, attached via PCIe or Thunderbolt dock
  - hardware/software open sourced
  - designed physical embodiments
    - Thunderbolt dock implant
    - malicious projector, charger
    - not fully engineered/productized
    - not released





# IOMMU vulnerability taxonomy

- *Spatial vulnerability*
  - 4KiB page granularity isn't fine enough to distinguish data fields in complex data structures
  - Read or write memory we aren't supposed to access
    - Kernel code pointers for control flow takeover
- *Temporal vulnerability*
  - Exploit the time gap between asking for a window to be closed and closure taking place
  - Memory gets reused for something else in the interim
- *Spatio-temporal vulnerability*
  - Force data visible for longer to exploit repeated spatial vulnerabilities



# Our IOMMU attacks

- Windows 10: barely uses the IOMMU, mostly unprotected from malicious devices
- MacOS: uses IOMMU since 2012 but in a limited way
  - ran a root shell
  - extracted private VPN traffic
- FreeBSD: IOMMU not enabled by default
  - when enabled, tries to properly segregate devices using IOMMU
  - root shell, private data extraction
- Linux: most distros don't enable the IOMMU by default
  - when enabled, tries to segregate devices using IOMMU
  - when enabled, could see private network traffic, kernel data, code pointers etc
  - simply set a bit in a PCIe packet to fully bypass the IOMMU!
- All exploitable from a malicious Thunderbolt dock



iMac (victim)

PCIe

Thunderclap on FPGA  
(Intel e1000 model)

### Device discovery and driver attachment

Hi! What are you?

I'm an ... Intel e1000 NIC .. I promise!

Oh cool, I've got the perfect large, buggy, and highly vulnerable vendor-provided device driver just for you!

Attacker selects their device driver of choice via the returned PCI device ID

### Device-driver/NIC protocol enters steady state.

Here are the descriptor rings, other parameters.

Great, because I'm a NIC.

The attacker can source and sink packets, allowing it to interact with OS state: respond to DHCP, make and accept TCP connections, trigger OS services launching, etc.

Use spatial vulnerability to look in IOMMU windows for sensitive leaked data, change it.

Exposed kernel control-flow pointers and their parameters allow arbitrary ROP-like code execution.

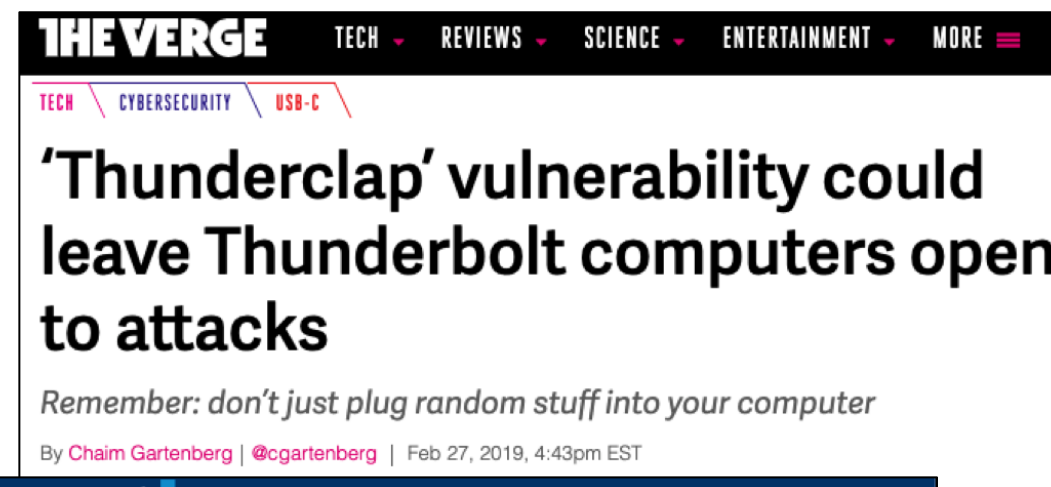


# The IOMMU attack surface

- The attacks shared-memory devices can do are rich, complex and nuanced
  - Substantially more powerful than attacks by message-passing devices such as USB
  - Shared memory interface like the syscall interface, but without hardening
- OS kernels are barely protected from devices by the IOMMU and accesses from devices
  - A large body of buggy and poorly tested device driver code
  - Often provided by third-parties
  - Malicious device can pick its shape to target the most vulnerable device driver
  - Performance is a key reason why IOMMU protections aren't fully used

# Media interest

- NDSS publication picked up by ~70 media outlets across the world



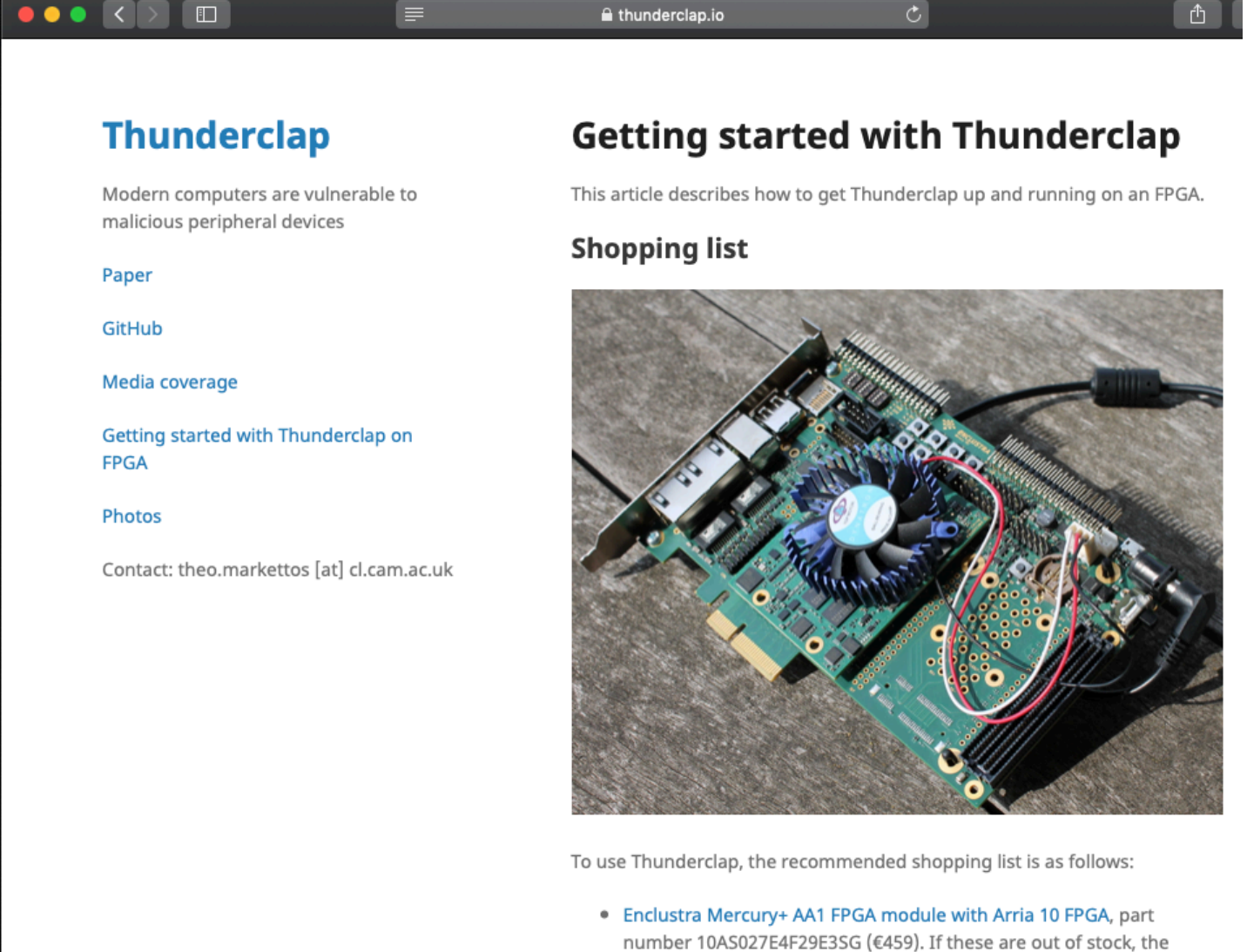
# Mitigations and impact

- Collaborating with vendors since 2016
- Apple mitigated specific exploit in MacOS 10.12.4
  - encrypt the kernel pointer, hide the flags
- Microsoft shipped Kernel DMA Protection for Thunderbolt 3 in Windows 10 1803
  - IOMMU enabled for Thunderbolt devices (only)
  - Requires post-1803 firmware, ie new products only
  - Best practice guidelines for businesses: 'Standards for a highly secure Windows 10 device'
- Intel enabled IOMMU for Thunderbolt in Linux 4.21 (now 5.0rc), disabled ATS
  - Thunderbolt devices are now less trusted than internal ones
- Major laptop vendor: we won't ship Thunderbolt until we understand this attack vector better



# Thunderclap.io transition to industry

- Vendors want to audit security from malicious devices, but don't have the skill set
- Our hardware and software has been open-sourced
- Worked hard to make it accessible to software folks
- Major vendors are now using it internally



The screenshot shows the Thunderclap.io website in a browser window. The page has a dark header with the site name and navigation icons. The main content area is white and features the 'Thunderclap' logo in blue. Below the logo is a tagline: 'Modern computers are vulnerable to malicious peripheral devices'. A list of links is provided: 'Paper', 'GitHub', 'Media coverage', 'Getting started with Thunderclap on FPGA', and 'Photos'. A contact email is listed at the bottom: 'Contact: theo.markettos [at] cl.cam.ac.uk'. On the right side, there is a section titled 'Getting started with Thunderclap' with a sub-header 'Shopping list'. Below this is a photograph of a green FPGA development board with a blue fan and various cables connected. At the bottom of the page, there is a paragraph stating: 'To use Thunderclap, the recommended shopping list is as follows:' followed by a bullet point: '• Enclustra Mercury+ AA1 FPGA module with Arria 10 FPGA, part number 10AS027E4F29E3SG (€459). If these are out of stock, the'.

**Thunderclap**

Modern computers are vulnerable to malicious peripheral devices

[Paper](#)

[GitHub](#)

[Media coverage](#)

[Getting started with Thunderclap on FPGA](#)

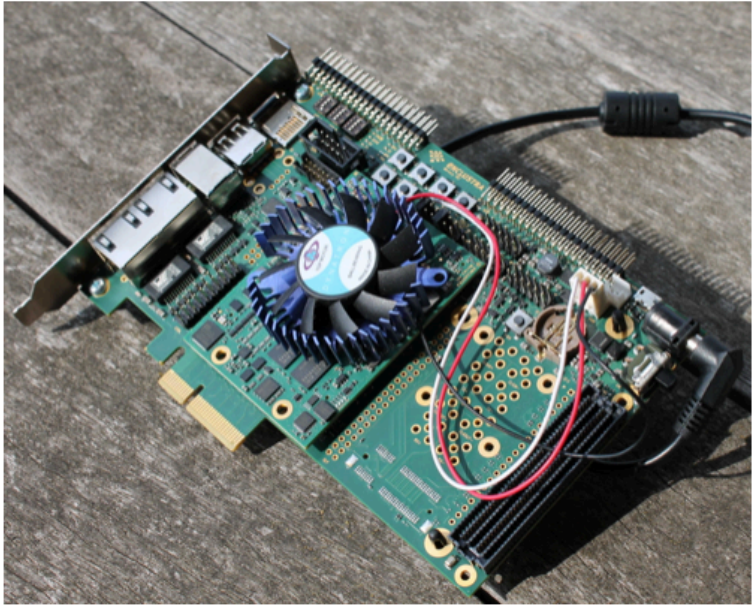
[Photos](#)

Contact: theo.markettos [at] cl.cam.ac.uk

## Getting started with Thunderclap

This article describes how to get Thunderclap up and running on an FPGA.

### Shopping list



To use Thunderclap, the recommended shopping list is as follows:

- [Enclustra Mercury+ AA1 FPGA module with Arria 10 FPGA](#), part number 10AS027E4F29E3SG (€459). If these are out of stock, the



# USB 4 standard

- Our paper substantially woke up industry to this threat
- Industry friends who saw our early draft pushing for improved defences in upcoming standards
- USB 4 = Newly published standard combining USB and Thunderbolt
- Imports our recommendations wholesale 😊
- Only security-related words in USB 4 spec 😞

## Universal Serial Bus 4 Specification

Apple Inc.  
Hewlett-Packard Inc.  
Intel Corporation  
Microsoft Corporation  
Renesas Corporation  
STMicroelectronics  
Texas Instruments

Version 1.0

August, 2019

### 11.2.3 PCIe Transaction Layer

A USB4 host needs to be hardened against malicious devices and malformed requests. The Transaction Layer in an Internal PCIe Port in a USB4 host, in conjunction with the System Software, needs to be able to provide appropriate protection against requests from rogue endpoints. The mechanism to provide such protection is implementation specific, but System Software needs certain functionality to be provided by the hardware.

In a USB4 Host, the Transaction Layer shall additionally provide functionality to:

Ensure that a transaction received on the PCIe Root Port is appropriate for the Requester ID. This can be done using ACS Source Validation or by an implementation-specific mechanism that is more appropriate for the architecture of the Host.

# Ongoing work

- Is there a better way?
  - Use the IOMMU better?
  - What are the limitations of the IOMMU?
    - Performance bottlenecks?
    - Techniques to manage the IOMMU
- Exploring other protection mechanisms
  - How to achieve performant, safe, DMA?

# Conclusion

- The IOMMU attack surface is a new and rich field for vulnerabilities
- We've helped vendors and standards bodies make the world a less-worse place
- Industrial evaluation and improvement still ongoing
- In the general case, the problems are harder than they appear
- Source code and FAQ: [thunderclap.io](https://thunderclap.io)

# Architectural Security Workshop?

- We had a lot of problems publishing the Thunderclap work
  - Mainstream security conferences didn't really understand hardware
  - Hardware security venues hacker-oriented or dominated by physical layer/crypto
- Perhaps we need to start our own venue?
  - Focusing on architectural security and the hardware/software interface
  - Better to co-locate with an arch conference than a security conference?
- Potential calls for workshops:
  - ISCA 2020, Valencia, Spain, May 2020. CfW closes 8 January 2020
  - MICRO 2020, Athens, Greece, October 2020, CfW closes ~June 2020
- If you'd be interested in taking part in a workshop, come and chat!
  - [theo.markettos@cl.cam.ac.uk](mailto:theo.markettos@cl.cam.ac.uk) [thunderclap.io](https://thunderclap.io)