When Lightning Strikes Thrice: Breaking Thunderbolt 3 Security

Björn Ruytenberg
Eindhoven University of Technology

@0Xiphorus • bjornweb.nl
Who Am I

Björn Ruytenberg
@0Xiphorus

Vulnerability researcher
Main interests: hardware and firmware security, sandboxing, input validation
More about me: https://bjornweb.nl

MSc student in Computer Science @ TUE
• This work part of master’s thesis
PCI Express Basics – Quick Review

• A standardized interconnect for attaching hardware devices in a computer system

• Designed as CPU-architecture agnostic, internal I/O interconnect for low-latency, high-bandwidth

• Intended to overcome limitations of PCI, most notably:
  • Scalability: per-device configurable bandwidth, flexible link width
  • Networking: moves from bus to packet switching; allows for more flexible topology, QoS / congestion control

• Network topology: root complex, switch, endpoints, PCIe to legacy bridge (e.g. ISA/PCI/PCI-X)

• Direct Memory Access (DMA) primary CPU-peripheral mode of transport
Thunderbolt: A PCIe-based Interconnect

• High-performance, proprietary I/O protocol developed by Intel and Apple
• PCIe-based, Direct Memory Access (DMA)-enabled I/O
• Use cases
  • External graphics, docking stations, 5K monitors, high-speed external storage, peer-to-peer networking
• Thunderbolt 1 (2011) and 2 (2013) mostly exclusive to Macs
  • Mini-DisplayPort form factor – multiplexes TB, native DP
• Thunderbolt 3 (2015) first version to be widely adopted
  • USB-C form factor – multiplexes TB, native DP and/or USB-C
DMA attacks

- **Thunderbolt 1**: no protection against physical attacks
- Plug in malicious device
  → Unrestricted R/W memory access (DMA)
- Access data from encrypted drives
- Persistent access possible, by e.g. installing rootkit
DMA attacks (selected)

• **Owned by an iPod [Dornseif 2004]**
  • First research to demonstrate practical DMA attack
  • Malicious FW device presents Serial Bus Protocol 2 (SPB-2) endpoint, which triggers host controller to allocate DMA channel for fast bulk data transfers
  • Several authors release exploitation tools [Boileau 2006] [Plegdon 2007]
  • Improved upon for memory forensics [Witherden 2010]
  • “Improved upon” in law enforcement spyware such as FinFireWire [Gamma 2011]

• **Subverting Windows 7 x64 kernel with DMA attacks [Aumaitre 2009]**
  • First PCI-based attack through custom PCI device with DMA engine

• **Inception [Maartmann-Moe 2014]**
  • Improves upon Witherden’s libforensic1394 by presenting virtual SBP-2 interface through ExpressCard, FW device + TB-to-FW adapter

• **PCILeech [Frisk 2016]**
  • Native PCIe attack
  • DMA attack using FPGA with PCIe PHY (full size, ExpressCard, miniPCIe, M.2-NVMe), optionally tunneled through Thunderbolt enclosure
  • Improved later with various functionality: e.g. dumping FDE keys, dumping UEFI memory regions, patching Windows lock screen process

• **Thunderclap [Markettos et al. 2019]**
  • Replaces PCIe endpoint in TB device with malicious one, then performs DMA attack
  • Does not break Security Levels access control, but relies on tricking user into authorizing malicious device
Threat Model

- Brief physical access to victim system, aka “evil maid attack”
- Example real-world scenarios:
  - Laptop locked or set to sleep; left unattended in hotel room, while victim is out for dinner
  - Desktop systems locked or set to sleep; left unattended outside office hours
  - Cleaning crew has unfettered access
Threat Model

Industry measures against opportunistic physical access

1. BIOS access control
2. Secure Boot
3. Boot Guard
4. Full Disk Encryption

...
Threat Model

Industry measures against opportunistic physical access

1. BIOS access control
   • Prevents unauthorized modification of system settings
   • E.g. require password on entering BIOS
Threat Model

Industry measures against opportunistic physical access

1. BIOS access control

2. Secure Boot
   - Protects against malicious, unsigned code early in boot process
   - Cryptographically verify boot chain: OS bootloader, kernel, drivers
Threat Model

Industry measures against opportunistic physical access

1. BIOS access control
2. Secure Boot
3. Boot Guard
   - Protects against malicious firmware implants
   - Cryptographically verifies BIOS integrity
Threat Model

Industry measures against opportunistic physical access

1. BIOS access control
2. Secure Boot
3. Boot Guard
4. Full Disk Encryption
   - Protects against physical data extraction
   - Encrypts user data + OS root (depending on FDE config)
Threat Model

Industry measures against opportunistic physical access

1. BIOS access control
2. Secure Boot
3. Boot Guard
4. Full Disk Encryption
5. Thunderbolt Security Levels
Thunderbolt Security Architecture

• **Security Levels** – access control system enabling users to authorize trusted device only
• Introduced in Thunderbolt 2
• No authorization = No PCIe tunneling
Thunderbolt Security Architecture

Thunderbolt devices authenticate to the host using the following metadata:

- **Device ID**: 16-bit device identifier
- **Device name**: ASCII string
- **Vendor ID**: 16-bit vendor identifier
- **Vendor name**: ASCII string
- **Universally Unique Identifier (UUID)**: 64-bit number uniquely identifying device, fused in silicon

Source: Thunderbolt 3 and Security on Microsoft Windows 10 Operating System – Intel Corporation
# Thunderbolt Security Levels

<table>
<thead>
<tr>
<th>Security Level (SL)</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SL0</strong> None</td>
<td>• No security (legacy mode)</td>
</tr>
</tbody>
</table>
| **SL1** User        | • Device authorization ACL based on UUID  
|                     | • UUID fused in silicon  
|                     | • Default setting on all PCs |
| **SL2** Secure      | • Device authorization based on UUID (SL1), *plus*  
|                     | • Cryptographic device authentication (challenge-response) |
| **SL3** No PCIe tunneling | • Disable all Thunderbolt connectivity  
|                     | • USB and/or DisplayPort tunneling only |
| **SL4** Disable daisy-chaining | Terminate PCIe tunneling at first TB device  
|                     | (some Titan Ridge controllers only) |
| **Pre-boot protection** | PCIe tunneling enabled only if Thunderbolt device previously authorized by user |

Security Levels prevent malicious TB devices from accessing PCIe domain, thereby protecting against:
- Device-to-host DMA attacks
- Device-to-device (P2P) DMA attacks
- PCI ID spoofing to target vulnerable device drivers
- TLP source ID spoofing

Source: Thunderbolt 3 and Security on Microsoft Windows 10 Operating System – Intel Corporation
Introduction to Thunderspy

• Previous research:
  • Before Security Levels: attacks primarily focus on PCIe-level DMA attacks to compromise Thunderbolt security
  • After Security Levels: attacks require cooperation of user, i.e. inadvertently connecting malicious peripherals

• Thunderspy is a new class of vulnerabilities that breaks Thunderbolt protocol security

• First attack on Thunderbolt Security Levels

• 7 vulnerabilities and 9 practical exploitation scenarios
Identifying attack surfaces

- Thunderbolt is a proprietary standard
- Protocol specifications not publicly documented
- Hardware architecture not publicly documented
- Dissected various Thunderbolt devices and Thunderbolt-equipped systems
Our Analysis of TB Hardware Architecture
Identifying attack surfaces

• Thunderbolt is a proprietary standard
• Protocol specifications not publicly documented
• Hardware architecture not publicly documented
• Dissected various Thunderbolt devices and Thunderbolt-equipped systems
Thunderbolt Devices
NetStor Thunderbolt NVMe Enclosure

Intel JHL6540
TB 3 host/device controller
4-channel, dual port

2* TPS65983
USB Type-C PD Controller
Power Switch
High-speed Multiplexer

MX25R8035F
8 Mbit SPI Flash

JTAG?
NetStor Thunderbolt NVMe Enclosure

Intel JHL6540
TB 3 host/device controller
4-channel, dual port

2* TPS65983
USB Type-C PD Controller
Power Switch
High-speed Multiplexer

MX25R8035F
8 Mbit SPI Flash

I²C

JTAG?
Intel JHL6540 Thunderbolt Controller

• 4 channel, dual-port Thunderbolt 3 controller
• Up to 20 Gbit per channel
• Supports Host and Endpoint mode
• “Alpine Ridge” generation:
  • DisplayPort 1.2
  • Integrated HDMI 2.0 LSPcon
  • USB 3.1 passthrough
  • USB-PD + 100W charging

• BGA package
• No public datasheets
• Not much we can do without more invasive techniques
TPS65983 USB-PD Controller

1 Features
- USB Power Delivery (PD) Controller
  - Mode Configuration for Source (Host), Sink (Device), or Source-Sink
  - B-Phase Marked Encoding/Decoding (BMC)
  - Physical Layer (PHY) Protocol
  - Policy Engine
  - Configurable at Boot and Host-Controlled
- USB Type-C Specification Compliant
  - Detect USB Cable Plug Attach
  - Cable Orientation and Role Detection
  - Assign CC and VCONN Pins
  - Advertise Default, 1.5 A or 3 A for Type-C Power
- Port Power Switch
  - 5-V, 3-A Switch to VBUS for Type-C Power
  - 5-V to 20-V, 3-A Bidirectional Switch to or from VBUS for USB PD Power
  - 5-V, 600-mA Switches for VCONN
  - Overcurrent Limiter, Overvoltage Protector
  - Slow Rate Control
  - Hard Reset Support
- Port Data Multiplexer
  - USB 2.0 HS Data, UART Data, and Low Speed Endpoint
  - Sideband Use Data for Alternate Modes (DisplayPort and Thunderbolt™)
  - Power Management
- Gate Control and Current Sense for External 5-V to 20-V, 5-A Bidirectional Switch (Back-to-Back NFETs)
- Power Supply from 3.3-V or VBUS Source
- 3.3-V LDO Output for Dead Battery Support
- BGA MicroStar Junior Package
  - 0.5-mm Pitch
  - Through-Hole Via Compatible for All Pins

2 Applications
- Thunderbolt 3 Devices

3 Description
The TPS65983 is a stand-alone USB Type-C and Power Delivery (PD) controller providing cable plug and orientation detection at the USB Type-C connector. Upon cable detection, the TPS65983 communicates on the CC wire using the USB PD protocol. When cable detection and USB PD negotiation are complete, the TPS65983 enables the appropriate power path and configures alternate mode settings for internal and (optional) external multiplexers.

Device Information(1)

<table>
<thead>
<tr>
<th>PART NUMBER</th>
<th>PACKAGE</th>
<th>BODY SIZE (NOM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPS65983</td>
<td>BGA MICROSTAR JUNIOR (96)</td>
<td>6.00 mm x 6.00 mm</td>
</tr>
</tbody>
</table>

(1) For all available packages, see the orderable addendum at the end of the data sheet.
Macronix MX25R8035F

Ultra Low Power 8M-BIT [x 1/x 2/x 4] CMCS MXSMIO® (SERIAL MULTI I/O) FLASH MEMORY

1. FEATURES

GENERAL
- Supports Serial Peripheral Interface — Mode 0 and Mode 3
- 8,192,608 x 1 bit structure or 4,194,304 x 2 bits (two I/O mode) structure or 2,097,152 x 4 bits (four I/O mode) structure
- Equal Sectors with 4K byte each, or Equal Blocks with 32K/64K byte each
  - Any Block can be erased individually
- Single Power Supply Operation
  - Operation Voltage: 1.65V-3.6V for Read, Erase and Program Operations
- Latch-up protected to 100mA from -1V to Vcc +1V

PERFORMANCE
- High Performance
  - Fast read
    - 1 I/O: 108MHz with 8 dummy cycles
    - 2 I/O: 104MHz with 4 dummy cycles, equivalent to 208MHz
    - 4 I/O: 104MHz with 2+4 dummy cycles, equivalent to 416MHz
  - Fast program and erase time
    - 8/16/32/64 byte Wrap-Around Burst Read Mode
- Ultra Low Power Consumption
- Minimum 100,000 erase/program cycles
- 20 years data retention

SOFTWARE FEATURES
Thunderbolt 3 Controller Firmware

- Device ROM stores Thunderbolt device identity
  - Device name
  - Device ID
- Vendor name
- Vendor ID
- UUID? Yes, but only 2 out of 8 bytes

```c
struct tb_drom_header {
    /* BYTE 0 */
    u8 uid_crc8; /* checksum for uid */
    /* BYTES 1-8 */
    u64 uid;
    /* BYTES 9-12 */
    u32 data_crc32; /* checksum for data_len bytes starting at byte 13 */
    /* BYTES 13 */
    u8 device_rom_revision; /* should be <= 1 */
    u16 data_len:10;
    u8 __unknown1:6;
    /* BYTES 16-21 */
    u16 vendor_id;
    u16 model_id;
    u8 model_rev;
    u8 eeprom_rev;
} __packed;
```
Thunderbolt 3 Controller Firmware

- Embedded in firmware
  - Public key (fingerprint likely stored in silicon)
  - Signed digest

- Device ROM stores Thunderbolt device identity
  - Device name
  - Device ID
  - Vendor name
  - Vendor ID
  - UUID (partial)

- What is covered by the signature?
Thunderspy: Vulnerability 1 + 2

• What is covered by the signature?
  • Not the DROM...

• **Vulnerability 1: Inadequate firmware verification schemes**
  • Firmware authenticated when updating from host, but not adequately upon connecting device, during boot, or resuming from sleep
  • Signature verification does not cover Thunderbolt device identity

• **Vulnerability 2: Weak device authentication scheme**
  • None of the identifiers linked to Thunderbolt PHY or each other, cryptographically or otherwise
  • E.g. can spoof arbitrary vendor ID that doesn’t match vendor name
Thunderbolt 3 Controller Firmware

Thunderbolt™ 3 Security Features details and definitions

Authenticating newly attached device

Firmware and software supported feature that requires user approval before allowing a PCIe capable Thunderbolt™ connection for the first time, supported on Thunderbolt™ starting in 2013

Cryptographic Authentication

Cryptographic authentication of connection to help prevent a peripheral device to be spoofed to masquerade as an “approved” device to the user (authentication of the connection), supported from Thunderbolt™ 2 products onward, starting in 2014

Separating Thunderbolt™ data stream

Separating Thunderbolt™ data stream from display tunneling to help prevent walk-up access of PCIe unless it is specifically allowed.

Unique ID number

Every Thunderbolt™ controller has a unique ID fused in silicon during production, this allows to identify a specific device

Statement inaccurate, but interesting emphasis on TB3

Source: Thunderbolt 3 and Security on Microsoft Windows 10 Operating System – Intel Corporation
• UUID stored in plaintext, not covered by any signatures
Thunderbolt 2 Controller Firmware

- UUID stored in plaintext, not covered by any signatures
- TB2 device can spoof TB3 devices
- Device identified as previously authorized = profit!
**Thunderspy: Vulnerability 3 + 4**

- **Vulnerability 3: Use of unauthenticated device metadata**
  - DROM not cryptographically verified
  - When combined with vulnerability 1 + 2, enables arbitrary identities and cloning user-authorized devices

- **Vulnerability 4: Downgrade attack**
  - Backwards compatibility with subjects Thunderbolt 3 systems to vulnerability introduced by Thunderbolt 2 hardware

- **Exploitation scenarios**
  - 3.1.1 – 3.1.3: Cloning victim devices with and without physical device access
  - Demonstrates spoofing victim device identity on arbitrary attacker device
Device Controller Firmware Outline

Jump address

- **Host mode:**
  - 0x00
- **EP mode:**
  - 0x4000

Secure key dictionary

- Maps 8-byte host UUID to 32-byte key

PHY config

- PtoSPtoQWake
- EE_CIO

DROM (0x4000)

- Device identity

PHY config (continued)

- EE_PCIE
- EE_DMA
- EE_USB_PA / PB
- EE_PCIE_PHI
- EE_DP
- PATCHES
- DP_IN_UCODE
- ...

“RSA+EXP”

- public key

Signed digest

TPS USB-PD FW

Temporary FW update buffer for host-initiated updates

*Offset varies by controller model, FW revision, and presence of secure key dictionary
Identifying attack surfaces

• Thunderbolt is a proprietary standard
• Protocol specifications not publicly documented
• Hardware architecture not publicly documented
• Dissected various Thunderbolt devices and Thunderbolt-equipped systems
Thunderbolt-Equipped Systems

- **Five vendors, seven generations of systems:**
  Intel, Lenovo, HP, Dell, Apple (2013 – 2020)
- **Five generations of Thunderbolt controllers:**
Host Controller: Key Questions

- UEFI enables user switching Thunderbolt Security Levels
  - DXE programs TB controller upon setting SL, so UEFI stores SL state?
- SL1+2 require storing device UUIDs
  - Device ACL?
Host Controller Firmware Outline

Jump address
- Host mode: 0x00
- EP mode: 0x4000

No secure key dictionary (stored on OS disk; pre-boot auth appears based on UUID only)

Device ACL (UUIDs)

Host Security Level configuration

DROM (0x4000)
- Host identity

PHY config (continued)
- EE_PCIE
- EE_DMA
- EE_USB_PA / PB
- EE_PCIE_PHI
- EE_DP
- PATCHES
- DP_IN_UCODE
- …

“RSA+EXP” public key

Signed digest

TPS USB-PD FW

Temporary FW update buffer for host-initiated updates

PHY config
- PtoSPtoQWake
- EE_CIO

*Offset varies by controller model, FW revision, and currently active Security Level
Thunderspy: vulnerability 5

• **Vulnerability 5: Use of unauthenticated controller configurations**
  • Two state machines: UEFI and host controller FW maintain SL state
  • Host controller FW overrides UEFI state
  • FW signature does not cover security configuration

• **Exploitation scenario**
  • 3.2.1: Disabling Thunderbolt security (SL1/SL2), or restoring Thunderbolt connectivity when disabled (SL3)
    • Demonstrates attacking host controller firmware: patch SL to 0 (no security)
    • Works against every Security Level
    • Enables restoring TB connectivity, even user disabled it (SL3)
### SPI Flash: Write Protection

#### 7.1.6 Complement Protect (CMP)

The Complement Protect bit (CMP) is a non-volatile read/write bit in the status register (S14). It is used in conjunction with SEC, TB, BP2, BP1 and BP0 bits to provide more flexibility for the array protection. Once CMP is set to 1, previous array protection set by SEC, TB, BP2, BP1 and BP0 will be reversed. For instance, when CMP=0, a top 4KB sector can be protected while the rest of the array is not; when CMP=1, the top 4KB sector will become unprotected while the rest of the array become read-only. Please refer to the Status Register Memory Protection table for details. The default setting is CMP=0.

#### 7.1.7 Status Register Protect (SRP1, SRP0)

The Status Register Protect bits (SRP1 and SRP0) are non-volatile read/write bits in the status register (S8 and S7). The SRP bits control the method of write protection: software protection, hardware protection, power supply lock-down or one time programmable (OTP) protection.

<table>
<thead>
<tr>
<th>SRP1</th>
<th>SRP0</th>
<th>WP</th>
<th>Status Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>Software Protection</td>
<td>WP pin has no control. The Status register can be written to after a Write Enable instruction, WEL=1. [Factory Default]</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Hardware Protected</td>
<td>When WP pin is low the Status Register locked and can not be written to.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Hardware Unprotected</td>
<td>When WP pin is high the Status register is unlocked and can be written to after a Write Enable instruction, WEL=1.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>Power Supply Lock-Down</td>
<td>Status Register is protected and can not be written to again until the next power-down, power-up cycle.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>One Time Program[^2]</td>
<td>Status Register is permanently protected and can not be written to.</td>
</tr>
</tbody>
</table>

---

Special order, yet some TB controller flash samples appear to ship support
Thunderspy: vulnerability 6

• **Vulnerability 6: SPI flash interface deficiencies**
  - Host controller FW maintains SL state (vulnerability 5)
  - SPI flash write protection allows preventing user to change SL
    - On supported flash, irrevocable OTP write protection turns it into ROM

• **Exploitation scenarios**
  - 3.3.1 – 3.1.3: Rendering SL0 permanent and blocking future firmware updates
  - Demonstrates ability to patch SL to 0 (vuln 5), then render it permanent (vuln 6)
  - Shown in demo 1
## Summary: Thunderspy Attack Methods (selected)

<table>
<thead>
<tr>
<th>Attack method 1</th>
<th>Exploitation scenarios: 3.2.1, 3.3.1, 3.3.2, 3.3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack Thunderbolt host controller firmware to disable Thunderbolt security. System will accept any arbitrary attacker devices.</td>
<td></td>
</tr>
<tr>
<td>• Requires brief access to laptop (~ 5 min) and reprogramming host controller firmware</td>
<td></td>
</tr>
<tr>
<td>• Does not require access to victim’s Thunderbolt devices</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attack method 2</th>
<th>Exploitation scenarios: 3.1.1, 3.1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clone user-authorized Thunderbolt device identity to an arbitrary attacker device. System will accept attacker device as being legitimate, user-authorized device.</td>
<td></td>
</tr>
<tr>
<td>• Does not require reprogramming host controller firmware</td>
<td></td>
</tr>
<tr>
<td>• Requires brief access to one of victim’s Thunderbolt devices (~ 5 min)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Impact (both)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Unrestricted read and write access to system memory (DMA)</td>
</tr>
<tr>
<td>• Access data from encrypted drives</td>
</tr>
<tr>
<td>• Persistent access possible, by e.g. (i) exploiting Thunderspy vulnerability 6, or (ii) installing rootkit to ensure continued access without requiring Thunderspy</td>
</tr>
</tbody>
</table>

For additional exploitation scenarios, please refer to the [vulnerability report](#).
Demo – Unlocking Windows PC in 5 minutes using attack method 1

Edited to fit Black Hat session. Please refer to our YouTube recording for the complete real-time footage.
# Thunderbolt Security Levels – Revisited

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL0 None</td>
<td>• No security (legacy mode)</td>
</tr>
</tbody>
</table>
| SL1 User       | • Device authorization ACL based on UUID  
                 • UUID fused in silicon  
                 • Default setting on all PCs |
| SL2 Secure     | • Device authorization based on UUID (SL1), *plus*  
                 • Cryptographic device authentication (challenge-response) |
| SL3 No PCIe tunneling | • Disable all Thunderbolt connectivity  
                         • USB and/or DisplayPort tunneling only |
| SL4 Disable daisy-chaining | Terminate PCIe tunneling at first TB device  
                             (some Titan Ridge controllers only) |
| Pre-boot protection | PCIe tunneling enabled only if Thunderbolt device previously authorized by user |
### Thunderbolt Security Levels – Revisited

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Definition</th>
<th>What we found it to mean</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SL0 None</strong></td>
<td>No security (legacy mode)</td>
<td></td>
</tr>
<tr>
<td><strong>SL1 User</strong></td>
<td>Device authorization ACL based on UUID</td>
<td>UUID not so unique – can be spoofed, UUID not fused in silicon</td>
</tr>
<tr>
<td></td>
<td>• UUID fused in silicon</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default setting on all PCs</td>
<td></td>
</tr>
<tr>
<td><strong>SL2 Secure</strong></td>
<td>Device authorization based on UUID (SL1), plus Cryptographic device authentication (challenge-response)</td>
<td>Keys stored in plaintext on device SPI flash – can be cloned</td>
</tr>
<tr>
<td></td>
<td>• Cryptographic device authentication (challenge-response)</td>
<td></td>
</tr>
<tr>
<td><strong>SL3 No PCIe tunneling</strong></td>
<td>Disable all Thunderbolt connectivity</td>
<td>...until the attacker reprograms the controller firmware to SL0 (no security)</td>
</tr>
<tr>
<td></td>
<td>• USB and/or DisplayPort tunneling only</td>
<td></td>
</tr>
<tr>
<td><strong>SL4 Disable daisy-chaining</strong></td>
<td>Terminate PCIe tunneling at first TB device</td>
<td>To connect malicious device, simply unplug existing device or pick another TB port</td>
</tr>
<tr>
<td></td>
<td>(some Titan Ridge controllers only)</td>
<td></td>
</tr>
<tr>
<td><strong>Pre-boot protection</strong></td>
<td>PCIe tunneling enabled only if Thunderbolt device previously authorized by user</td>
<td>All security levels broken, so has no effect</td>
</tr>
</tbody>
</table>
Thunderspy PoC Tools

Thunderbolt Controller Firmware Patcher

https://github.com/BjornRuytenberg/tcfp
Thunderspy PoC Tools

**SPIblock**

https://github.com/BjornRuytenberg/spiblock
Thunderspy: Affected systems

• All Thunderbolt-equipped systems shipped between 2011-2020
  • All PCs released between 2011-2018 fully vulnerable
  • All Macs running Windows and Linux (Boot Camp) fully vulnerable
  • Some systems providing Kernel DMA Protection, shipping since 2019, partially vulnerable: https://thunderspy.io/#kernel-dma-protection
  • MacOS partially vulnerable: https://thunderspy.io/#affected-apple-systems

• Spycheck
  • Free and open-source tool to determine if your system is vulnerable: https://thunderspy.io
  • Alternatively, follow manual verification steps on website
Thunderspy: Intel’s response

Kernel DMA Protection

- Intel-suggested mitigation to Thunderspy
- Opt-in DMA remapping for Thunderbolt devices
- Requires Windows 10 >= 1803, Linux kernel >= 5.0
Device-to-Host DMA
Device-to-Host DMA with IOMMU
Thunderspy: Intel’s response

Kernel DMA Protection

- Intel-suggested mitigation to Thunderspy
- Opt-in DMA remapping for Thunderbolt devices
- Requires Windows 10 >= 1803, Linux kernel >= 5.0

However,

- Partial mitigation only
  - Mitigates only vulnerabilities 4-6
  - Prevents impact via DMA, but remaining vulnerabilities 1-3 expose system to BadUSB-style attacks
- Requires IOMMU and UEFI (BIOS) support
- UEFI support exclusively available on some >= 2019 systems
- Not available on systems < 2019
Thunderspy 2

• All Thunderbolt-equipped systems released 2011-2018, and several >= 2019, remain unpatched against Thunderspy
• Starting with Haswell (2013), a lot of Intel consumer systems feature an IOMMU, thus technically capable of supporting DMA remapping
• Thunderspy 2: OS-agnostic ACPI table upgrade patch
  • Brings Kernel DMA Protection to roughly 6 years worth of systems
    • Includes Thunderbolt 2!
  • Experimental OS-agnostic UEFI extension
    • Works with Windows 10 1803+ and Linux kernel 5.0+
    • Note: ACPI patching could also be turned into attack, i.e. disabling Kernel DMA Protection on supported systems. Recommended to self-sign TS2 extension and use measured boot (next slide)
• Protection level similar to officially supported systems at OS runtime
  • Does not protect against boot time attacks, but screenlocking + sleep mode are covered 😊
Thunderspy 2: Mitigations on Linux

• We are working with the Linux kernel hardware security team to develop kernel-level mitigations
  • Work around ACPI to enable Kernel DMA Protection on unsupported Thunderbolt systems

• Meanwhile, Linux users can use TS2 UEFI extension
  • Secure Boot: sign using your own keys
  • Combine with measured boot (e.g. TPM-enabled GRUB) for additional security
Demo 2 – Kernel DMA Protection patched onto unsupported machine
What’s Next?

The future of Thunderbolt-based interconnects

• What issues currently remain unaddressed?
  1. **Thunderspy vulnerabilities 1–3**: No means to distinguish between forged and legitimate DROMs. Devices that look legitimate physically could still be malicious.
  2. **Narrow scope of Kernel DMA Protection vs. Security Levels**: Enables PCIe tunneling without user interaction. Does not protect against malicious devices that
     • spoof arbitrary PCI IDs to target vulnerable device drivers
     • spoof TLP source IDs to hijack transactions

• How may these issues affect USB 4 and Thunderbolt 4?
  • To mitigate Thunderspy, Thunderbolt 4 now requires Kernel DMA Protection as part of vendor product certification
  • Backwards compatibility likely means susceptibility to (1), while (2) remains unaddressed
What’s Next?

The future of Thunderbolt-based interconnects

• What are potential avenues on mitigating these remaining issues?
  • **Thunderspy vulnerabilities 1–3:**
    Firmware embeds public key + digest; may allow to verify authenticity on host (driver, DXE) if Intel publishes digest scope
  • **Narrow scope of Kernel DMA Protection vs. Security Levels:**
    (1) Allow all DMA devices on boot. OS runtime: initially, “null-route” all new DMA devices using IOMMU. Require screen unlocking and explicit user authorization, then have IOMMU assign I/O memory range.
    (2) Virtualization-based security (VBS) may help prevent kernel memory safety issues
    (3) TB controller-assisted TLP source ID verification (similar to PCIe ACS)
  • **USB 4:**
    Implement UEFI toggle that controls Thunderbolt signaling (... and maintain state in UEFI only, please!)
• **Thunderspy**: a new class of vulnerabilities breaking Thunderbolt security
  - No fix from Intel for vulnerable systems released in 2011-2020; Kernel DMA Protection available only on some >= 2019 systems
  - Check if your system is vulnerable – use Spycheck or verify manually
  - Full vulnerability report: [https://thunderspy.io](https://thunderspy.io)

• **Thunderspy 2**: experimental, OS-agnostic mitigation to Thunderspy
  - Brings Kernel DMA Protection to all vulnerable systems with IOMMU

• **The future is PCI Express**
  - Thunderbolt is a powerful external interconnect enabling high-bandwidth, low-latency use cases previously not possible
  - USB 4 and Thunderbolt 4 upcoming, but adequate protection schemes remain absent (for now?)
Thank You

Questions?

Björn Ruytenberg

@0Xiphorus
https://bjornweb.nl