Funderbolt
Adventures in Thunderbolt DMA Attacks

Russ Sevinsky
Background

- Thunderbolt
  - Apple and Intel collaboration
  - Expansion port
  - PCI Express (PCIe) and DisplayPort using the same port
- DMA
  - Direct Memory Access
  - Processor becomes bottleneck for high-speed transfers
  - Lets devices read and write directly to RAM
Why external buses matter for security experts?

- Digital Forensics
  - Getting data to solve a mystery
- User protection
  - So RAM contents can be safe
- Sneaky DRM
  - Bus encryption
Background

• Current DMA research
  • I/O Attacks in Intel-PC Architectures and Countermeasures\(^1\)
  • Understanding DMA Malware\(^2\)

• Current Thunderbolt attacks?
  • Daisy chaining Thunderbolt and Firewire\(^3\)
  • Inception\(^4\)
  • De Mysteriis Dom Jobsivs (Mac EFI Rootkits)\(^5\)
Background

• Mitigations?
  • Epoxy (really?)
  • Input/Output Memory Management Units (IOMMUs)
    • Maps physical memory addresses to logical addresses
    • Think “VM for DMA”
    • Prevents devices from requesting physical addresses directly
    • Currently not implemented on Apple hardware
  • Secure Configurations$^6$
    • Prevents attacks when computer is locked
    • Boot attacks and social engineering still possible
I can has bypass
A Trip Down Memory Lanes
A Trip Down Memory Lanes

- PCI Express (PCIe)
  - High-speed serial bus
  - Data sent via “lanes”
  - A lane is made up of differential wire pairs
    - One + and one – wire
    - Helps to reduce noise
  - One lane (x1) is made up of two differential pairs
    - Transmit pair (PET)
    - Receive pair (PER)
A Trip Down Memory Lanes

- PCIe (cont)
  - Four lanes (x4) has eight wires, etc...
  - All lanes CAN use another differential pair for clock
    - REFCLK
    - Not required (could use a XTAL)
  - So per spec, x1 only needs 4 wires for data communication!
    - PET and PER
  - Typical x1 setups use a REFCLK
A Trip Down Memory Lanes

• PCIe (cont)
  • PCIe device types
    • Endpoints (Legacy and Native)
    • Switch/Bridge
    • Root Complex
  • Legacy endpoints
    • More permissive than Native endpoints for backwards compatibility
    • Supports “Locked” transactions, IO transactions and 32-bit addressing only
A Trip Down Memory Lanes

• PCIe (cont)
  • Data sent via “packets” called Transaction Layer Packet (TLP)
  • Point-to-point topology
    • No arbitration
    • Devices do not need to be “granted” access to the bus
  • Packets routed by address
    • Peer-to-peer transfers are possible
  • Packets going to RAM go through “Root Complex”
A Trip Down Memory Lanes
A Trip Down Memory Lanes

• PCIe (cont)
  • PCIe devices are discovered via “enumeration”
    • Starts at Bus 0 (root complex), Device 0, Function 0
    • Checks if it’s a bridge or an endpoint
    • Traverses tree of devices
    • Assigns ids and addresses
  • Hot plug device?
    • Enumerate just that port
  • Device info stored in expansion ROMs and read into configuration space
How My Adventures Went

• Improvised Tools for Analysis
  • Multimeter
  • Soldering station
  • Heat gun
  • Desoldering tools
  • Ethernet cable
  • Epoxy (really?)
  • Logic Analyzer
  • Image Editor
How My Adventures Went

• Reversing Thunderbolt – The Process
  • Research a product
  • Take it apart
  • Locate important ICs
  • Trace connections between ICs
  • Look for datasheets
  • Sniff buses
  • Develop a map
How My Adventures Went

• Looking at consumer products
  • Buffalo MiniStation Thunderbolt/USB3 Hard Drive
    • 500GB and 1TB model
    • USB3 and Thunderbolt
    • Decent form factor for reversing
  • Apple Thunderbolt to Gigabit Ethernet Adapter
    • Tiny
    • Small
    • Little
How My Adventures Went

- Buffalo MiniStation Thunderbolt/USB3 Hard Drive
  - Excellent Anandtech review
  - Tear-down instructions
  - Identified ICs for us!

![Buffalo MiniStation Thunderbolt/USB3 Hard Drive](image)
How My Adventures Went

• Main ICs
  • MLDU03
    • Medial Logic USB3.0 to SATA 6G Bridge
  • ASM1061
    • ASMedia PCIe to SATA Controller
  • DSL2210 (Peak Ridge)
    • Intel Thunderbolt Controller
    • Supports PCIe x1
  • LPC1114
    • NXP ARM Cortex Mo
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went

• ASMedia ASM1061
  • PCIe/SATA Controller
  • Datasheets?
  • ROMs/Flashes?
How My Adventures Went

ASM1061 SATA6G

Reserve internal 1.2V voltage

1.2V delay from 3.3V 90% > 0ms
How My Adventures Went
How My Adventures Went

PCIe RST#
PCIe REFCLK+
PCIe REFCLK-
PCIe TX-
PCIe TX+
PCIe RX-
PCIe RX+
How My Adventures Went

- Patch PCIe Controllers’ SPI ROM to send DMA read requests?
How My Adventures Went

- NXP LPC1114
  - ARM Cortex Mo
  - Used for... ??
  - No ROMs or Flashes
  - TONS of info online
  - Connects into DSL2201
    - How do I know?
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went

• Intel DSL2210
  • Thunderbolt Controller
  • No Datasheets
  • Promo info only
  • ROMs/Flashes?
How My Adventures Went

• Continuity testing SPI ROM...
  • Beep! It’s Thunderbolt!
  • hasROM = TRUE;
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went

Thunderbolt Connector
How My Adventures Went

- Thunderbolt Connector
  - 1 pair of HighSpeed lanes
    - TX and RX
  - All others pulled to ground
  - “LowSpeed” lines go to ARM?

<table>
<thead>
<tr>
<th>Pin</th>
<th>Pin out</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>2</td>
<td>HPD</td>
<td>Hot plug detect</td>
</tr>
<tr>
<td>3</td>
<td>HS0TX(P)</td>
<td>HighSpeed transmit 0 (positive)</td>
</tr>
<tr>
<td>4</td>
<td>HS0RX(P)</td>
<td>HighSpeed receive 0 (positive)</td>
</tr>
<tr>
<td>5</td>
<td>HS0TX(N)</td>
<td>HighSpeed transmit 0 (negative)</td>
</tr>
<tr>
<td>6</td>
<td>HS0RX(N)</td>
<td>HighSpeed receive 0 (negative)</td>
</tr>
<tr>
<td>7</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>8</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>9</td>
<td>LSR2P TX</td>
<td>LowSpeed transmit</td>
</tr>
<tr>
<td>10</td>
<td>GND</td>
<td>Ground (reserved)</td>
</tr>
<tr>
<td>11</td>
<td>LSP2R RX</td>
<td>LowSpeed receive</td>
</tr>
<tr>
<td>12</td>
<td>GND</td>
<td>Ground (reserved)</td>
</tr>
<tr>
<td>13</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>15</td>
<td>HS1TX(P)</td>
<td>HighSpeed transmit 1 (positive)</td>
</tr>
<tr>
<td>16</td>
<td>HS1RX(P)</td>
<td>HighSpeed receive 1 (positive)</td>
</tr>
<tr>
<td>17</td>
<td>HS1TX(N)</td>
<td>HighSpeed transmit 1 (negative)</td>
</tr>
<tr>
<td>18</td>
<td>HS1RX(N)</td>
<td>HighSpeed receive 1 (negative)</td>
</tr>
<tr>
<td>19</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>20</td>
<td>DPPWR</td>
<td>Power</td>
</tr>
</tbody>
</table>
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went
How My Adventures Went

- ARM UART Traffic
  - String “EM “
How My Adventures Went

- Thunderbolt Firmware Update
  - Display contents of Application Package
  - Decompress “Payload” file

```
$ ls
Bom  postinstall
PackageInfo postinstall_actions
Payload  preinstall
Scripts  preinstall_actions
ThunderboltFirmwareUpdate.pkg

$ file Payload
Payload: gzip compressed data, from Unix

$ mv Payload Payload.tgz

$ tar zxf Payload.tgz
```
How My Adventures Went

• Two Firmwares for Thunderbolt?
  • One is probably ARM
  • Let’s look for string “EM “

```
$ ls
HPM-em-00020007.bin ThorUtil.efi
MBP-22-01.bin
```
How My Adventures Went

Jackpot!
How My Adventures Went

• Round 2...
  • String \"\x27\x0a\x00\x00\"
How My Adventures Went

Successaroo!
How My Adventures Went
How My Adventures Went

```cpp

379     // hot plug bridge, but use legacy if avail
380     uint8_t line = device->configRead8(kIOPCIConfigInterruptLine);
381     if (tunnelLink)
382         {
383             tunnelLink = (0x15138006 != vendorProd)
384                 && (0x151a8006 != vendorProd)
385                 && (0x151b8006 != vendorProd)
386                 && (0x15498006 != vendorProd)
387                 && ((0x15478086 != vendorProd) || ((revIDClass & 0xff) > 1));
388             DLOG("tunnel bridge 0x%08x, %d, msi %d\n",
389                     vendorProd, (revIDClass & 0xff), tunnelLink);
390         }
391     if (tunnelLink || (line == 0) || (line == 0xFF))
392         {
393             // no legacy ints, need one MSI
394             numVectors = 1;
395         }
```
How My Adventures Went
How My Adventures Went
How My Adventures Went
function InitJTAG {
    eax = InitGPIO();
    if (eax != 0x0) {
       sub_bb8b();
       *(int8_t *)0x16090 = 0x1;
       sub_bbea(0x38);
       sub_bd0f(0x100000);
       sub_bbea(0x3a);
       eax = sub_bd0f(0x0);
       if (eax == 0x151a8086) {
           debug_output(0x1, "InitJTAG: Found EagleRidge (0x151A8086)\n");
           eax = 0x10002c;
       } else if (eax != 0x15138086) {
           debug_output(0x1, "InitJTAG: Found unknown device (0x%08x)\n", eax);
           esi = 0x8000000e;
       } else {
           debug_output(0x1, "InitJTAG: Found LightRidge (0x15138086)\n");
           eax = 0x100028;
       }
    }
    *0x15fe8 = eax;
    esi = 0x0;
}
else {
    eax = debug_output(0x7, "InitJTAG: Couldn't initialize GPIOs (0x%x)\n", esi);
    if (**0x14a84 <= 0x6) {
        *(int16_t *)(**0x14a88 + eax * 0x2) = 0x100;
    }
    *(**0x14a88 + 0x10) = esi;
}
return esi;
How My Adventures Went

• Header rows
  • CN₃ = LED Voltages
  • CN₉ = Power rails
  • CN₄ = JTAG!
How My Adventures Went

• JTAG Reversing
  • Trail and error
  • They make devices for this
  • Pin 1 => TCK
  • Pin 2 => TDI
  • Pin 3 => TMS
  • Pin 4 => TDO

jtag> cable ft2232 vid=0x0403 pid=0x8a98
Connected to libftdi driver.
jtag> idcode
Reading 0 bytes of idcode
Read 00000000(0x00) 00000000(0x00) 00000000(0x00) 00000000(0x00)
jtag> detect
IR length: 6
Chain length: 1
Device Id: not supported (bit 0 was not a 1)
jtag>
How My Adventures Went

JTAG
How My Adventures Went

• JTAG Reversing
  • ARM Cortex Mo uses SWD, not JTAG...
  • Obfuscated with a start sequence
  • We saw this code earlier...

```
jtag> discovery
Detecting IR length ... 6
Detecting DR length for IR 111111 ... 1
Detecting DR length for IR 000000 ... 25
Detecting DR length for IR 000001 ... 25
Detecting DR length for IR 000010 ... warning: TDO seems to be stuck at 1
-1
Detecting DR length for IR 000011 ... warning: TDO seems to be stuck at 1
-1
```
How My Adventures Went

• Thunderbolt Device ROM
  • Vendor Name
  • Device Name
  • Vendor ID
  • Device ID
  • Device Revision
  • UID

HD-PATU3:

Vendor Name: BUFFALO INC.
Device Name: HD-PATU3
Vendor ID: 0x29
Device ID: 0x1
Device Revision: 0x1
UID:
Route String: 3
Firmware Version: 5.1
Port:
  Status: Device connected
  Link Status: 0x2
  Port Micro Firmware Version: 0.2.5
  Cable Firmware Version: 0.1.24
How My Adventures Went

- Thunderbolt Device ROM
  - Vendor Name
  - Device Name
  - Vendor ID
  - Device ID
  - Device Revision
  - UID

HD-PATU3:

- Vendor Name: BUFFALO INC.
- Device Name: HD-PATU3
- Vendor ID: 0x29
- Device ID: 0x1
- Device Revision: 0x1
- UID: 0x002900010005A0E0
- Route String: 3
- Firmware Version: 5.1
- Port:
  - Status: Device connected
  - Link Status: 0x2
  - Port Micro Firmware Version: 0.2.5
  - Cable Firmware Version: 0.1.24
How My Adventures Went

- Thunderbolt Device ROM
  - Vendor Name
  - Device Name
  - Vendor ID
  - Device ID
  - Device Revision
  - UID

```
<table>
<thead>
<tr>
<th>Vendor Name</th>
<th>BUFFALO INC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Name</td>
<td>HD-PATU3</td>
</tr>
<tr>
<td>Vendor ID</td>
<td>0x29</td>
</tr>
<tr>
<td>Device ID</td>
<td>0x1</td>
</tr>
<tr>
<td>Device Revision</td>
<td>0x1</td>
</tr>
<tr>
<td>UID</td>
<td>0x002900010005A0E0</td>
</tr>
<tr>
<td>Route String</td>
<td>3</td>
</tr>
<tr>
<td>Firmware Version</td>
<td>5.1</td>
</tr>
<tr>
<td>Port</td>
<td>Status: Device connected</td>
</tr>
<tr>
<td></td>
<td>Link Status: 0x2</td>
</tr>
<tr>
<td></td>
<td>Port Micro Firmware Version: 0.2.5</td>
</tr>
<tr>
<td></td>
<td>Cable Firmware Version: 0.1.24</td>
</tr>
</tbody>
</table>
```

![Image of Thunderbolt Device ROM and vendor information]
How My Adventures Went

• PCIe Info
  • Thunderbolt is a PCIe Bridge
How My Adventures Went
How My Adventures Went

<table>
<thead>
<tr>
<th>Device ID</th>
<th>Vendor ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 49</td>
<td>80 86</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 10</td>
<td>00 07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class Code</th>
<th>RevID</th>
</tr>
</thead>
<tbody>
<tr>
<td>06 04 00</td>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIST</th>
<th>HdrType</th>
<th>LatTimer</th>
<th>$LineSze</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01</td>
<td>00</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BAR0</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 00 00 00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BAR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 00 00 00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SecLaTmr</th>
<th>SubBus#</th>
<th>SecBus#</th>
<th>PriBus#</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0A</td>
<td>0A</td>
<td>09</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Secondary Status</th>
<th>IO Limit</th>
<th>IO Base</th>
<th>Memory Limit</th>
<th>Memory Base</th>
<th>Prefetch Mem Limit</th>
<th>Prefetch Mem Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 00</td>
<td>41</td>
<td>41</td>
<td>8F B0</td>
<td>8F B0</td>
<td>00 01</td>
<td>00 01</td>
</tr>
</tbody>
</table>
How My Adventures Went

<table>
<thead>
<tr>
<th>Device ID</th>
<th>Vendor ID</th>
<th>Status</th>
<th>Command</th>
<th>Type Code</th>
<th>RevID</th>
<th>BIST</th>
<th>HdrType</th>
<th>LatTimer</th>
<th>$LineSze</th>
<th>BAR0</th>
<th>RAR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>06 12</td>
<td>1B 21</td>
<td>00 10</td>
<td>04 06</td>
<td>01 06 01</td>
<td>01</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Base Address Register 0**

| Address: 4020h | Type: IO |

The BARs are used to determine what type of address space(s) each function may need and requested by that BAR. If a function supports decoding addresses larger than 32 bits, two con
How My Adventures Went

• Gigabit Ethernet Adapter
  • Researching the product
  • Taking it apart
  • Attack vectors
How My Adventures Went
How My Adventures Went

Conformal Coating

ARM Cortex
How My Adventures Went
How My Adventures Went
How My Adventures Went

DSL2210 & LPC

PCIe wires!!
How My Adventures Went
How My Adventures Went
How My Adventures Went

- Altera Cyclone IV GX Transceiver Starter Kit
  - Hard IP for PCIe
  - PCIe x1
  - ~$400
How My Adventures Went

I can has bypass
How My Adventures Went
How My Adventures Went

• Tips and Tricks
  • Get A LOT of devices!
  • Heat up everything SLOWLY!
  • Continuity testing WINS
  • Sniff EVERYTHING
  • Read all ROMs/Flashes
Thank You

- Russ Sevinsky
  - Security Consultant at iSEC Partners
  - rsevinsky@isecpartners.com

- Special thanks to:
  - Everyone @ iSEC, specifically
    - Jesse Burns
    - Mike Warner
  - Craig Heffner
1.) “I/O Attacks in Intel-PC Architectures and Countermeasures”
   http://www.syssec-project.eu/media/page-media/23/syssec2011-s1.4-sang.pdf

2.) “Understanding DMA Malware”:

3.) “Adventures with Daisy in Thunderbolt-DMA-land: Hacking Macs through the Thunderbolt interface”:
4.) Inception: http://www.breaknenter.org/projects/inception/
6.) “Protecting yourself against Firewire DMA attacks on 10.7.x”: http://derflounder.wordpress.com/2012/02/05/protecting-yourself-against-firewire-dma-attacks-on-10-7-x/
UK Offices
Manchester - Head Office
Cheltenham
Edinburgh
Leatherhead
London
Thame

European Offices
Amsterdam - Netherlands
Munich – Germany
Zurich - Switzerland

North American Offices
San Francisco
Atlanta
New York
Seattle

Australian Offices
Sydney