The Phantom Menace: Intel ME Manufacturing Mode

Mark Ermolov & Maxim Goryachy
About us

Mark Ermolov  
Security Researcher at Positive Technologies  
Twitter: @markel___  
e-mail: mermolov[at]ptsecurity[dot]com

Maxim Goryachy  
Security Researcher at Positive Technologies  
Twitter: h0t_max  
e-mail: mgoryachy[at]ptsecurity[dot]com
Motivation & Retrospective
We found the unsigned code execution in Intel ME 11 (INTEL-SA-00086)*

*How to Hack a Turned-Off Computer, or Running Unsigned Code in Intel Management Engine
Need write access to ME-region for exploitation
We were looking for a solution...
Found several undocumented HECI commands that allow rewriting ME-Region
Agenda

- Intel-SA-00086 Overview
- Write Protection Bypass
- Communication Protocol
- Manufacturing Mode
- Platform Restart
- What Can Users Do?
Intel-SA-00086 Overview
Intel ME Overview

- **Undocumented** Intel technology with proprietary firmware
- **Root of trust** for almost all modern Intel security features
- Has **full access** to all platform hardware
- Has hardware capabilities for **interception** of all user activity
- Controls **all stages** of platform operating cycle
Intel-SA-00086 Vulnerability

- Attacker needs **write access to MFS partition** of ME SPI region
- Affected Intel® Management Engine (ME), Intel® Server Platform Services (SPS), and Intel® Trusted Execution Engine (TXE)
Affected Products

• 6th, 7th & 8th Generation Intel® Core™ Processor Family
• Intel® Xeon® Processor E3-1200 v5 & v6 Product Family
• Intel® Xeon® Processor Scalable Family
• Intel® Xeon® Processor W Family
• Intel® Atom® C3000 Processor Family
• Apollo Lake Intel® Atom Processor E3900 series
• Apollo Lake Intel® Pentium™
• Celeron™ N and J series Processors
Intel-SA-00086: PoC

- JTAG PoC for the Gigabyte Brix GP-BPCE-3350C platform
- https://github.com/ptresearch/IntelTXE-PoC
Write Protection Bypass
Ways to Rewrite ME SPI Region

• Mistakes of SPI flash regions settings in SPI flash descriptor
• Via HMR-FPO HECI message
  ✓ Manufacture mode
  ✓ Attack on UEFI setup variable
  ✓ DMA attack
• Security Descriptor Override jumper
• SPI programmer
SPI-Flash Layout

DESC
BIOS
ME
GBE
SPI-Flash Region Access Permissions

24.3.21 Flash Region Access Permissions (CSXE_FRACC)—Offset 50h

Access Method

<table>
<thead>
<tr>
<th>Type: HEC Register (Size: 32 bits)</th>
<th>Device: Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default: 434h</td>
<td></td>
</tr>
</tbody>
</table>

Post SPI

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Default &amp; Access</th>
<th>Field Name (ID): Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td>0h RWI/S</td>
<td>CSME Master Write Access Grant (MEMWAG): Each bit [31:24] corresponds to Master[7:0]. CSME can grant one or more masters write access to the CSME region 2 overriding the permissions in the Flash Descriptor. Bits for unassigned masters are reserved.</td>
</tr>
<tr>
<td>23:16</td>
<td>0h RWI/S</td>
<td>CSME Master Read Access Grant (MEMRAG): Each bit [23:16] corresponds to Master[7:0]. CSME can grant one or more masters read access to the CSME region 2 overriding the read permissions in the Flash Descriptor. Bits for unassigned masters are reserved.</td>
</tr>
<tr>
<td>15:8</td>
<td>4h KOV</td>
<td>CSME Region Write Access (MERWA): Each bit [15:8] corresponds to Regions [7:0]. If the bit is set, this master can erase and write that particular region through register accesses. The contents of this register are that of the Flash Descriptor: Flash Master 2 Master Region Write Access. A particular master has granted CSME write permissions in their Master Write Access Grant register OR the Flash Descriptor Security Override strap is set. CSME always have the write access to its own Region 2 by default. See also CM_WAP.</td>
</tr>
<tr>
<td>7:0</td>
<td>4h KOV</td>
<td>CSME Region Read Access (MERRA): Each bit [7:0] corresponds to Regions [7:0]. If the bit is set, this master can read that particular region through register accesses. The contents of this register are that of the Flash Descriptor: Flash Master 2 Master Region Read Access. A particular master has granted CSME read permissions in their Master Read Access Grant register OR the Flash Descriptor Security Override strap is set. CSME always have the read access to its own Region 2 by default. See also CM_RAP.</td>
</tr>
</tbody>
</table>
## SPI-Flash Access Control: Good Case

<table>
<thead>
<tr>
<th>FROM</th>
<th>DESC</th>
<th>BIOS</th>
<th>ME</th>
<th>GBE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>BIOS</td>
<td>R</td>
<td>R/W</td>
<td>-/-</td>
<td>-/-</td>
</tr>
<tr>
<td>ME</td>
<td>R</td>
<td>R</td>
<td>R/W</td>
<td>R</td>
</tr>
<tr>
<td>GBE</td>
<td>-/-</td>
<td>-/-</td>
<td>-/-</td>
<td>R/W</td>
</tr>
</tbody>
</table>
## SPI-Flash Access Control: Bad Case

<table>
<thead>
<tr>
<th>FROM</th>
<th>DESC</th>
<th>BIOS</th>
<th>ME</th>
<th>GBE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>R/W</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>BIOS</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>ME</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>GBE</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
</tbody>
</table>
## ME-Region Permissions: Wild World

<table>
<thead>
<tr>
<th>Model</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUS Z170-A</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Gigabyte Brix 3350C</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Gigabyte Brix 6300</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Gigabyte Z97M</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Gigabyte B360</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Lenovo Yoga</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Lenovo ThinkPad x260</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Apple</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Intel NUC</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
“Magic” Jumper

FAQ and Troubleshooting

Q: How can I overwrite the descriptor when FPT does not have write access? How can I overwrite a region that is locked down by descriptor protections? How do I write to flash space that is not defined by the descriptor?

A: By asserting HDA_SDO (flash descriptor override strap) low on the rising edge of PWROK, you can read, write and erase all of SPI flash space regardless of descriptor protections. Any protections imposed by BIOS or directly to the SPI flash part still apply. This should only be used in debug or manufacturing environments. End customers should NOT receive systems with this strap engaged.

<table>
<thead>
<tr>
<th>PCH Strap Table</th>
<th>Strap description</th>
<th>Sampled</th>
<th>Configuration</th>
<th>Circuit</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPKR</td>
<td>Different from Capella</td>
<td>No reboot mode setting</td>
<td>PWROK</td>
<td>0 = Default (weak pull-down 20K)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = “top-block swap” mode</td>
<td></td>
</tr>
<tr>
<td>GNT3#/GPIO55</td>
<td>Top-Block Swap Override</td>
<td>PWROK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Default (weak pull-up 20K)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Override</td>
<td></td>
</tr>
<tr>
<td>INTRMEN</td>
<td>Integrated 1.05V VRM enable</td>
<td>ALWAYS</td>
<td>Should be always pull-up</td>
<td></td>
</tr>
<tr>
<td>HDA_SDO</td>
<td>Flash Descriptor Security</td>
<td>PWROK</td>
<td>0 = effective (Default: weak pull down)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Override</td>
<td></td>
</tr>
</tbody>
</table>
ME FW Overwrite
Communication Protocol
Management Engine Interface (MEI)

- Formerly called HECI (Host-Embedded Communication Interface)
- From host’s view it is **internal PCI device** with BDF 0:22:0(1)
- Communication performed using **ring buffers** accessed by MMIO registers of MEI
- ME applications communicate with host applications through MEI using unique **client IDs hardcoded in firmware**
- Each client ID defines the structure of messages passing through MEI
HMR FPO Enable MKHI Command

- HMR FPO - Host ME Region Flash Protection Override
- It has MKHI command ID 0x01, from the group MKHI_GROUP_ID_HMRFPO (0x05)
- The binary sequence sent to MEI is: 0x800c0007 0x00000105 0x00000000 0x00000000
- It can be sent only if End of Post command has not been sent yet
- It takes effect after next reboot and works only before subsequent reboot
- If the command is in effect, ME region on SPI flash can be written from host ignoring flash descriptor master access settings
HMR FPO Enable MKHI Command

- HMR FPO - **Host ME Region Flash Protection Override**
- It has MKHI command ID **0x01**, from the group MKHI_GROUP_ID_HMRFPO (0x05)
- The binary
  
  - y sequence sent to MEI is: 0x800c0007 0x00000105 0x00000000 0x00000000
  
  - It can be sent only if **End of Post** command has not been sent yet
  - It takes effect after next reboot and **works only before subsequent reboot**
  - If the command is in effect, **ME region on SPI flash can be written from host ignoring flash descriptor master access settings**
System Loading: “Normal” Way

CPU

ME

Reset
Vector

Reset
Vector

Loading Kernel

Start CPU

Wait for DID

Waiting DID

Send DID

Send EOP

Start OS

DXE

Loading Kernel

Start CPU

Waiting DID

Waiting EOP

Lock Configuration

Module

Module
System Loading: Real Way

CPU

ME

Reset Vector → Loading Kernel → Start CPU → Memory Init → Send DID → DXE → Send EOP → Start OS

MMode is active

Yes

No

Waiting DID → Waiting EOP → Lock Configuration → Module
Manufacturing Mode

- HMR

- FPO - Host ME Region Flash Protection Override
  - It has MKHI command ID 0x01, from the group MKHI_GROUP_ID_HMRFPO (0x05)
  - The binary sequence sent to MEI is: 0x800c0007 0x00000105 0x00000000 0x00000000
  - It can be sent only if **End of Post** command has not been sent yet
  - It takes effect after next reboot and works only before subsequent reboot
  - If the command is in effect, ME region on SPI flash can be written from host ignoring flash descriptor master access settings
ME FW Manufacture mode

- A special initial mode of ME Firmware designed for platform testing by vendors *
- Allows set-up BootGuard, ISH and other important PCH settings
- Indicated by bit #4 of HFS MEI register (0x40 MEI config space offset)
- Intel an added auto-disabling feature for ME 11+ (if access mask is set)
- On same platform stored in one-time-programmable memory (FUSEs)

* Firmware Bring Up guide from Intel ME system tools
Flash Descriptor: Unlock

Start

hmrpfo = read(/susram/hmrpfo)
Lock = False

hmrpfo == 0

Set R/W Access

Read HECI Message

Is MM Active?
Lock = True

Is EOP
Is HMR_FPO
Lock == False

write(/susram/hmrpfo, 1)

End
We found that Apple laptops on Intel chipsets are running in Manufacturing Mode.
Apple's computers contain an additional check in the UEFI, which runs when the UEFI is launched and blocks startup of the system if the ME region has been opened with HMRFPO
Platform Restart
Platform Reset: CPU Side

PMC Controller (D31:F2)

Type: CPU Register
(Device: 32 bits)
Function: 2

Default: 0h

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Access</th>
<th>Field Name (19): Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>RW/W/R</td>
<td>CF9h Lockdown (CF9LOCK): 0 = CF9h Global Reset bit R/W. 1 = CF9h Global Reset bit RO. When set, this bit becomes RO and is reset by a CF9h reset or RSRST# assertion (other reset types are not applicable). In manufacturing/debug environments this bit should be left as default '0'. In all other environments, BIOS must program this bit to '1'.</td>
</tr>
<tr>
<td>24</td>
<td>RW/L</td>
<td>Power Button Disable Lock (PB_DIS_LOCK): Once set, this bit cannot be changed until the next global reset. When this bit is set to 1, the PM(Constant PB_DIS bit can no longer be changed.</td>
</tr>
<tr>
<td>26</td>
<td>RW/L</td>
<td>CF9h Global Reset (CF9GR): 0 = A CF9h write of 6h or Eh will only reset the Host partition. 1 = A CF9h write of 6h or Eh will cause a Global Reset of both the Host and the ME partitions. It is recommended that BIOS should set this bit early on in the boot sequence, and then clear it and set the CF9LOCK bit prior to loading the OS in both an ME Enabled and a ME Disabled system. This register is locked by the CF9 Lockdown (CF9LOCK) bit. This register is not reset by a CF9h reset. It is reset by RSRST# assertion.</td>
</tr>
</tbody>
</table>

35
## Platform Reset Type

<table>
<thead>
<tr>
<th></th>
<th>ME</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Reset</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Soft Reset</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>???</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>
ME Rest HECI Command

- It has MKHI command ID 0x0b, from the group 00
- The binary sequence sent to MEI is: 0x80060007 0x00000b00 0x00000300
- Command can be sent at any time, even after EOP
Write Protection Bypass

Attacker: Step 1
send HMR_FPO*

ME
/susram/hmrfpo := 1

Attacker: Step 2
send ME_RESET*

If (/susram/hmrfpo==1)
send HMR_FPO*

rewrite ME-region**

Attacker: Step 3
set R\W access

ME

*Need access to HECI device
**Need access to SPI device
Demo
CVE-2018-4251

CVE-2018-4183: Dan Bastone and Eric Rafaloff of Gotham Digital Science
Entry added July 11, 2018

Firmware

Available for: macOS High Sierra 10.13.4

Impact: A malicious application with root privileges may be able to modify the EFI flash memory region

Description: A device configuration issue was addressed with an updated configuration.

CVE-2018-4251: Maxim Goryachy and Mark Ermolov
INTEL-SA-00086 + CVE-2018-4251

**Local vector** for exploitation of INTEL-SA-00086, which enables **running arbitrary code in Intel ME**
What Can Users Do?
Detection: MEInfo
Detection: Mmdetect

https://github.com/ptresearch/mmdetect
Detection: CHIPSEC

https://github.com/chipsec/chipsec/releases/tag/v1.3.6
Disabling Manufacturing Mode

```
fitw64.exe -closemnf no
```
Q & A

https://github.com/ptresearch
http://blog.ptsecurity.com