

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

POSITIVE TECHNOLOGIES



Intel DCI Secrets



POSITIVE TEC



POSITIVE TECHNOLOGIES

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

- CVSSv3: **AV:L**/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H (8.2 High)
- 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



SPI-Flash Region Access Permissions

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

Access Method

Type: MEM Register (Size: 32 bits)	Device: Function:
--	------------------------------------

Default: 404h

Fast SPI



Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	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 CSxE region 2 overriding the permissions in the Flash Descriptor. Bits for unassigned masters are reserved.
23:16	0h RW/L	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.
15:8	4h RO/V	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 OR 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
7:0	4h RO/V	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 OR 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

SPI-Flash Access Control: Good Case

FROM \ TO	DESC	BIOS	ME	GBE
DESC	NA	NA	NA	NA
BIOS	R	R/W	-/-	-/-
ME	R	R	R/W	R
GBE	-/-	-/-	-/-	R/W

SPI-Flash Access Control: Bad Case

FROM \ TO	DESC	BIOS	ME	GBE
DESC	R/W	NA	NA	NA
BIOS	R/W	R/W	R/W	R/W
ME	R/W	R/W	R/W	R/W
GBE	R/W	R/W	R/W	R/W

ME-Region Permissions: Wild World

Model	Read	Write
ASUS Z170-A	-	-
Gigabyte Brix 3350C	+	+
Gigabyte Brix 6300	+	+
Gigabyte Z97M	+	+
Gigabyte B360	+	+
Lenovo Yoga	-	-
Lenovo ThinkPad x260	-	-
Apple	+	-
Intel NUC	-	-

“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.



PCH Strap Table

Pin Name	Strap description	Sampled	Configuration	Circuit									
SPKR <small>Different from Calpella</small>	No reboot mode setting	PWROK	0 = Default (weak pull-down 20K) 1 = Setting to No-Reboot mode										
GNT3# / GPIO55	Top-Block Swap Override	PWROK	0 = "top-block swap" mode 1 = Default (weak pull-up 20K)										
INTVRMEN	Integrated 1.05V VRM enable	ALWAYS	Should be always pull-up										
HDA_SDO	Flash Descriptor Security <small>Only for Interposer</small>	PWROK	0 = effective(Default: weak pull down) 1 = Override										
GNT1# / GPIO51	Boot BIOS Selection 1 [bit-1]	PWROK	<table border="1"> <thead> <tr> <th>GNT1#</th> <th>GNT0#</th> <th>Boot Location</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>SPI</td> </tr> <tr> <td>0</td> <td>0</td> <td>LPC</td> </tr> </tbody> </table>	GNT1#	GNT0#	Boot Location	1	1	SPI	0	0	LPC	
GNT1#	GNT0#	Boot Location											
1	1	SPI											
0	0	LPC											
GPIO19 <small>Different from Calpella</small>	Boot BIOS Selection 0 [bit-0]	PWROK											

ME FW Overwrite



Communication Protocol

Management Engine Interface (MEI)

- Formerly called HECI (**H**ost-**E**MBEDDED **C**ommunication **I**nterface)
- 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

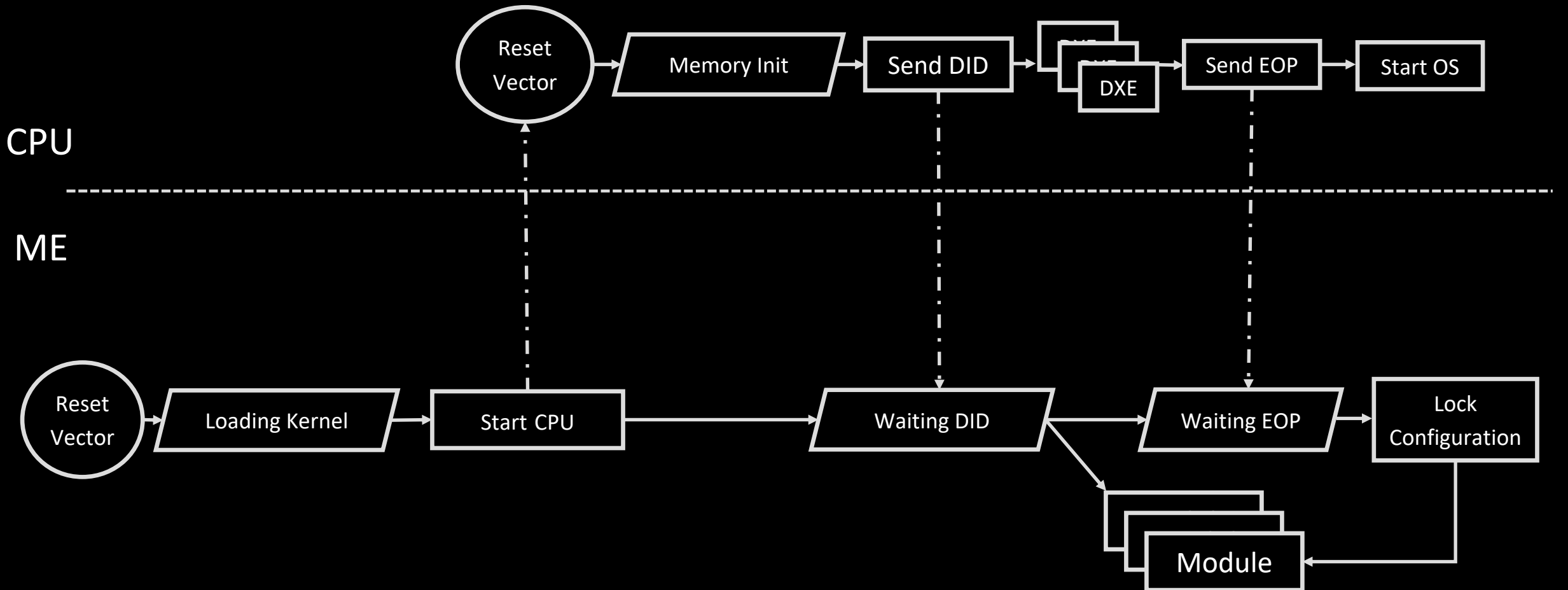
HMR FPO Enable MKHI Command

- HMR FPO - **H**ost **ME** **R**egion **F**lash **P**rotection **O**verride
- 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**

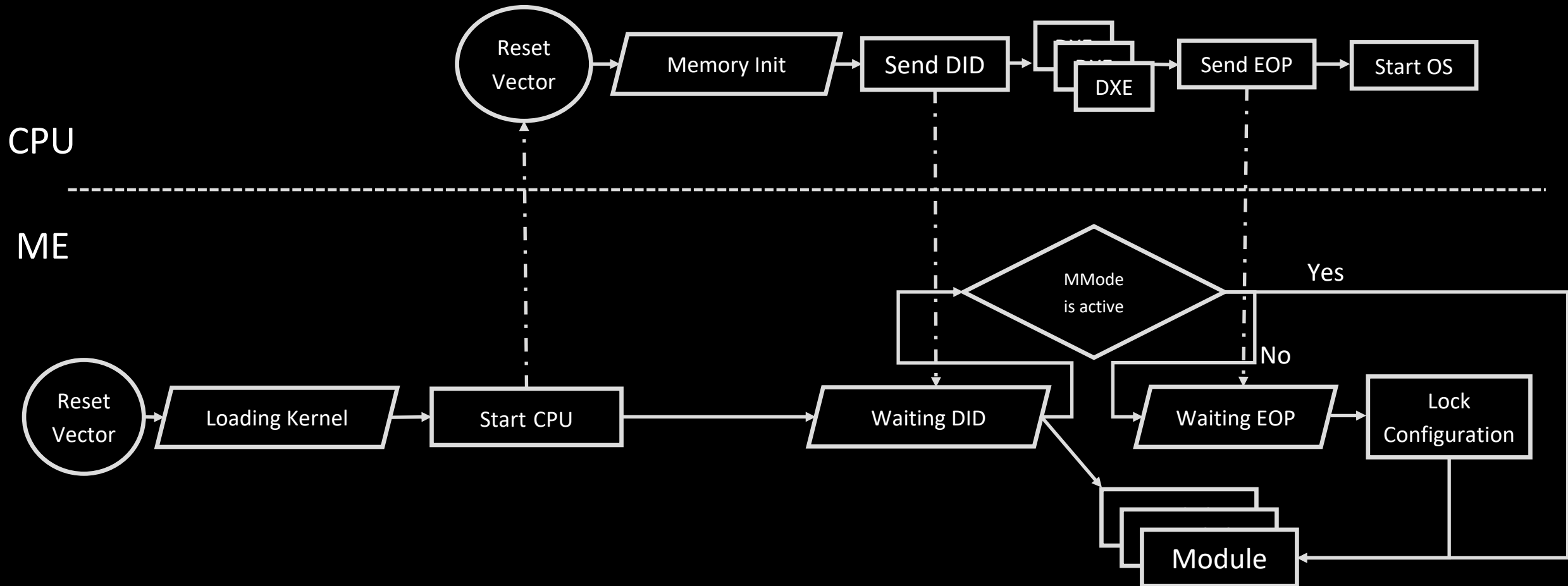
HMR FPO Enable MKHI Command

- 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



System Loading: Real Way



Manufacturing Mode

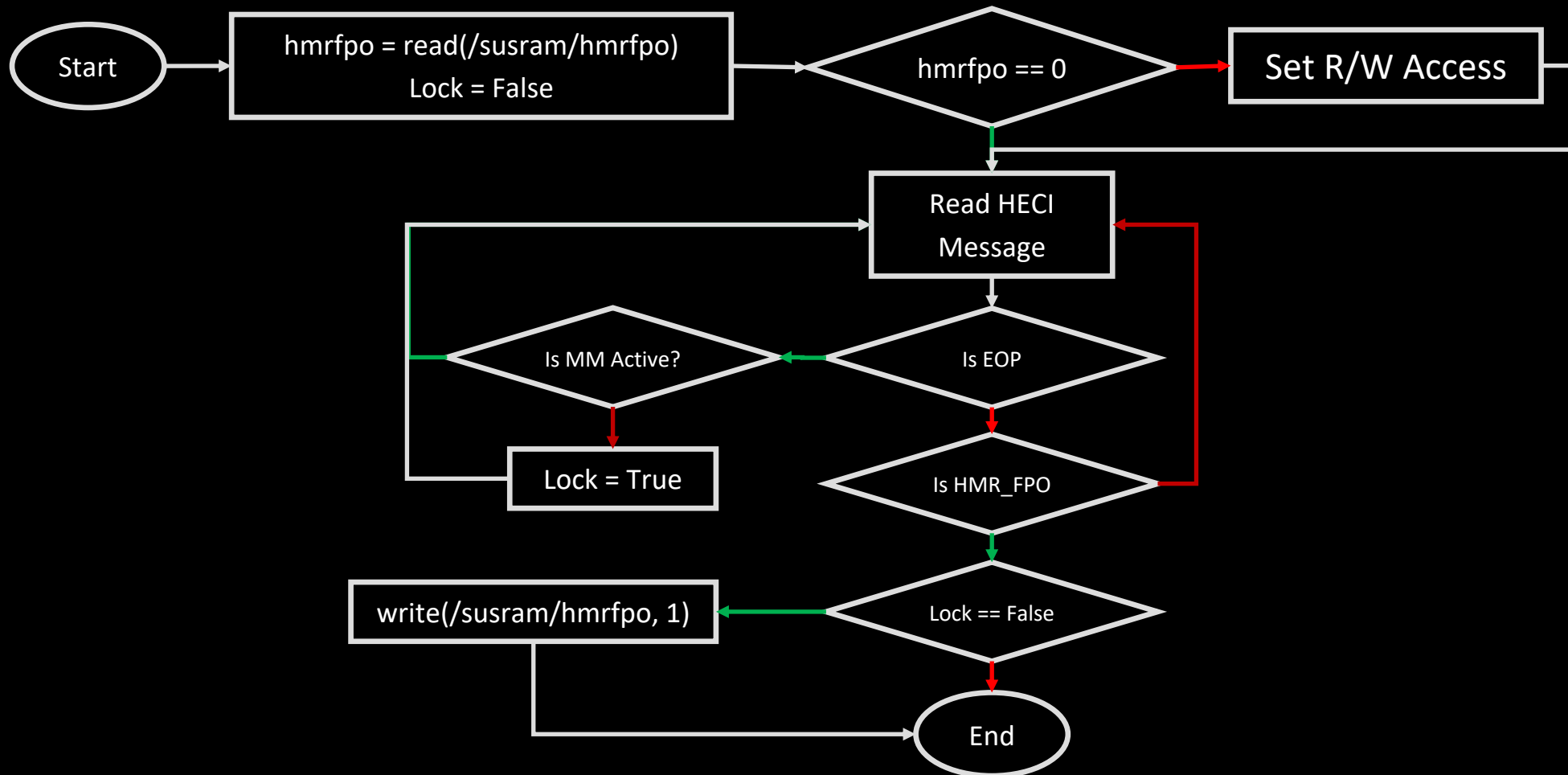
- HMR
- FPO - Host ME Region Flash Protection Override
- It has a flash command ID 0x01, from the group MEI_GROUP_ID_HMRFPO (0x05)
- The binary sequence sent to MEI is: 0x800c0007 0x00000105 0x00000000 0x00000000
- ~~It can be sent only if **End of Post** command has not been sent yet~~
- It takes effect 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)

```
home/mca/ [5]
1: iF=070 m=dN---Irwxr-x--- u=0046 g=00EE s=1DEF2070.4A32.D29AC77F . <dir> Non-I
2: iF=008 m=dN---Ir-xr-xr-x u=0000 g=0000 s=10000008.0000.00000000 .. <dir> Non-I
3: iF=071 m= N---Irw-r----- u=0000 g=00EE s=137D3071.06F6.9D4401C2 eom 1 Non-I
4: iF=072 m= N---Irw-r---r-- u=0046 g=00EE s=1E234072.6FE5.A11D54CB manuf_lock 1 Non-I
5: iF=03B m= N---Irwxr----- u=0000 g=00EE s=10B0E03B.1A3B.17744312 deploy 32 Non-I
```

Flash Descriptor: Unlock




We found that **Apple laptops** on Intel
chipsets are
running in Manufacturing Mode

Restriction

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: CFG Register (Size: 32 bits) **Device:** 31
Function: 2

Default: 0h

3	2	2	2	1	1	8	4	0															
1	8	4	0	6	2																		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CF9LOCK	RSVD				PB_DIS_LOCK	RSVD		CF9GR	RSVD														

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V/L	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 RSMRST# 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'.
30:25	0h RO	Reserved.
24	0h RW/L	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_CFG*.PB_DIS bit can no longer be changed.
23:21	0h RO	Reserved.
20	0h RW/L	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 RSMRST# assertion.
19:0	0h RO	Reserved.

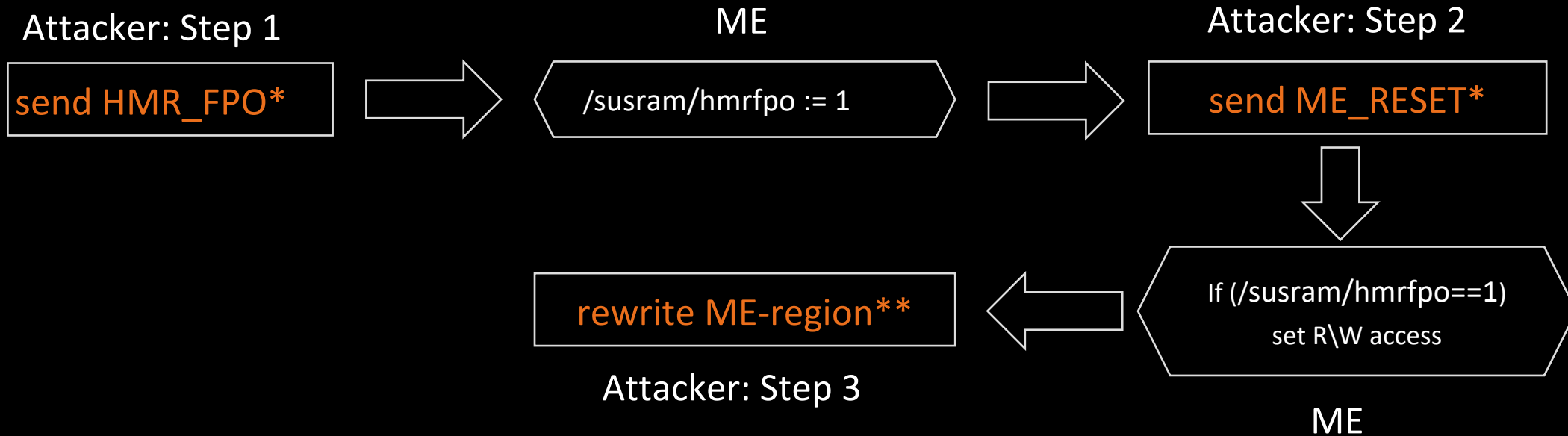
Platform Reset Type

	ME	CPU
Global Reset	+	+
Soft Reset	-	+
???	+	-

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

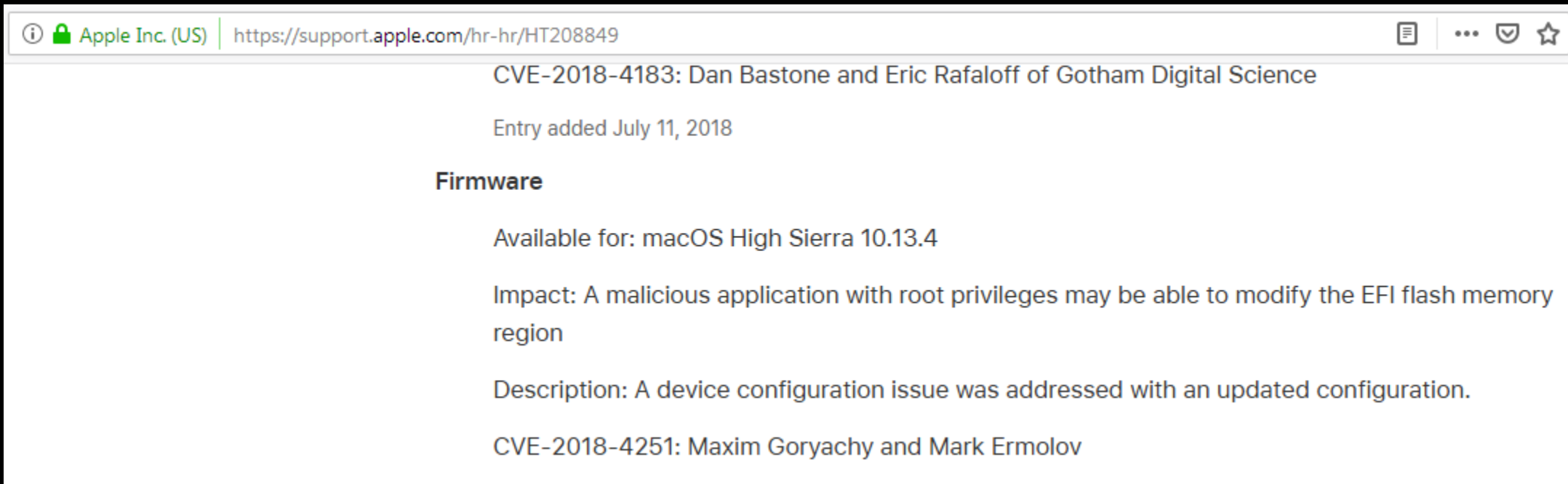


*Need access to HECI device

**Need access to SPI device

Demo

CVE-2018-4251



The screenshot shows a web browser window with the address bar displaying "Apple Inc. (US) | https://support.apple.com/hr-hr/HT208849". The page content includes the following text:

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

```
Administrator: Command Prompt
D:\fpt>TXEInfoWin64.exe -fwsts

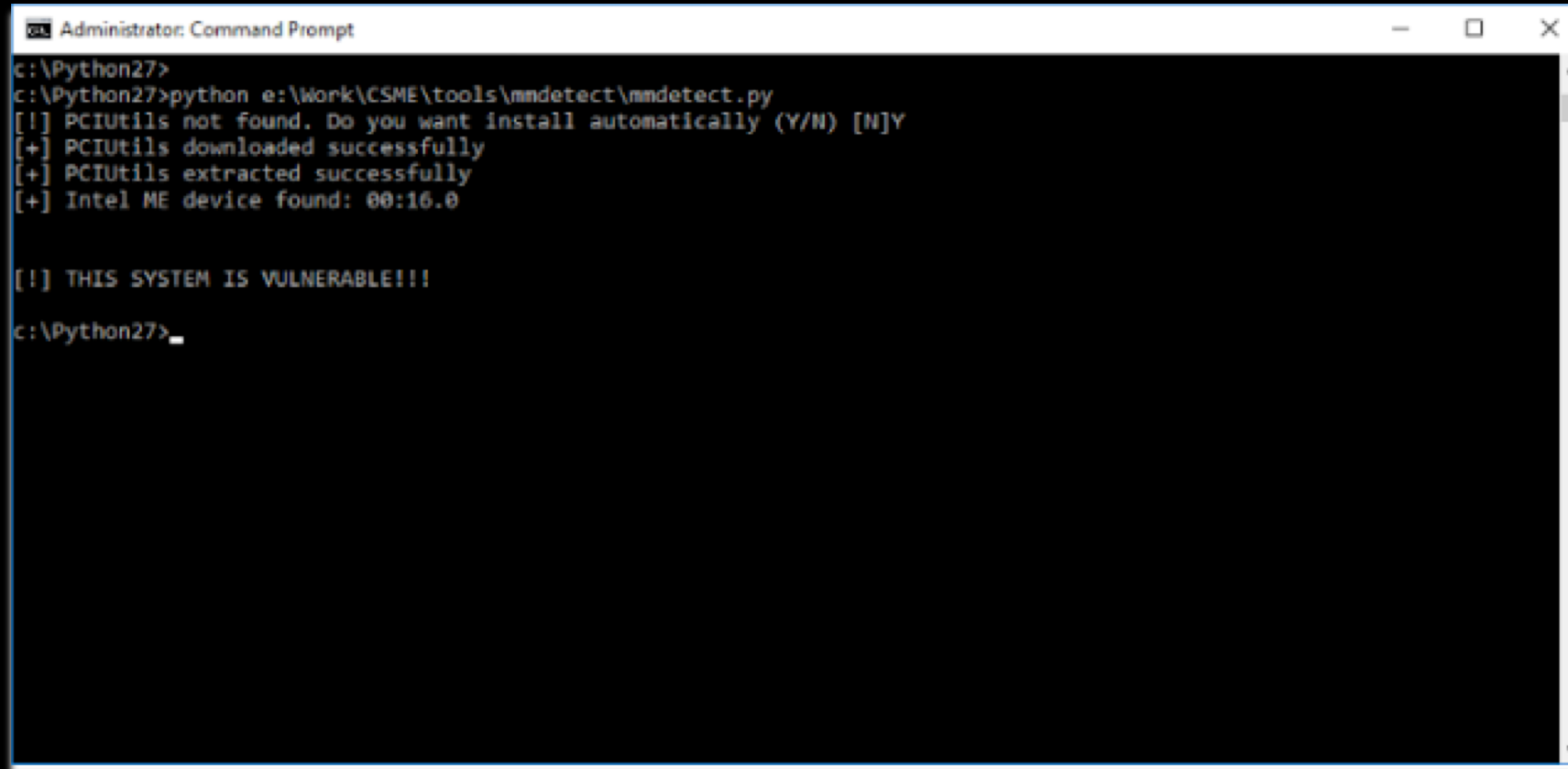
Intel(R) TXEInfo Version: 3.1.50.2222
Copyright(C) 2005 - 2017, Intel Corporation. All rights reserved.

FW Status Register1: 0x80000255
FW Status Register2: 0x09030400
FW Status Register3: 0x30B50608
FW Status Register4: 0x00000000
FW Status Register5: 0x00000000
FW Status Register6: 0x00000000

CurrentState: Normal
ManufacturingMode: Enabled
FlashPartition: Valid
OperationalState: CM0 with UMA
InitComplete: Complete
BUPLoadState: Success
ErrorCode: No Error
ModeOfOperation: Normal
SPI Flash Log: Not Present
Phase: ROM/Preboot
TXE File System Corrupted: No
PhaseStatus: INIT_SUSRAM
FPF and TXE Config Status: Not committed

D:\fpt>
```

Detection: Mmdetect



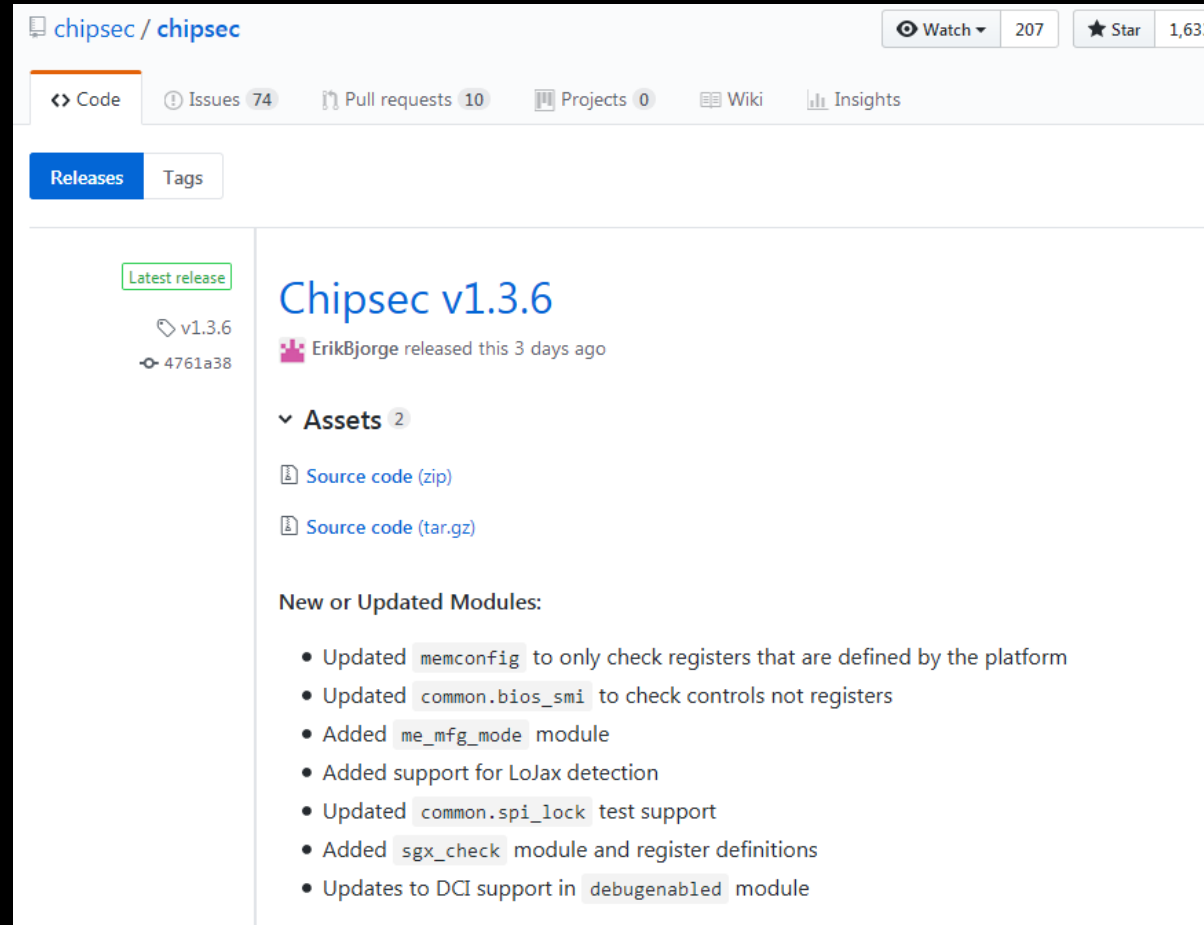
```
Administrator: Command Prompt
c:\Python27>
c:\Python27>python e:\Work\CSME\tools\mmdetect\mmdetect.py
[!] PCIUtils not found. Do you want install automatically (Y/N) [N]Y
[+] PCIUtils downloaded successfully
[+] PCIUtils extracted successfully
[+] Intel ME device found: 00:16.0

[!] THIS SYSTEM IS VULNERABLE!!!

c:\Python27>
```

<https://github.com/ptresearch/mmdetect>

Detection: CHIPSEC

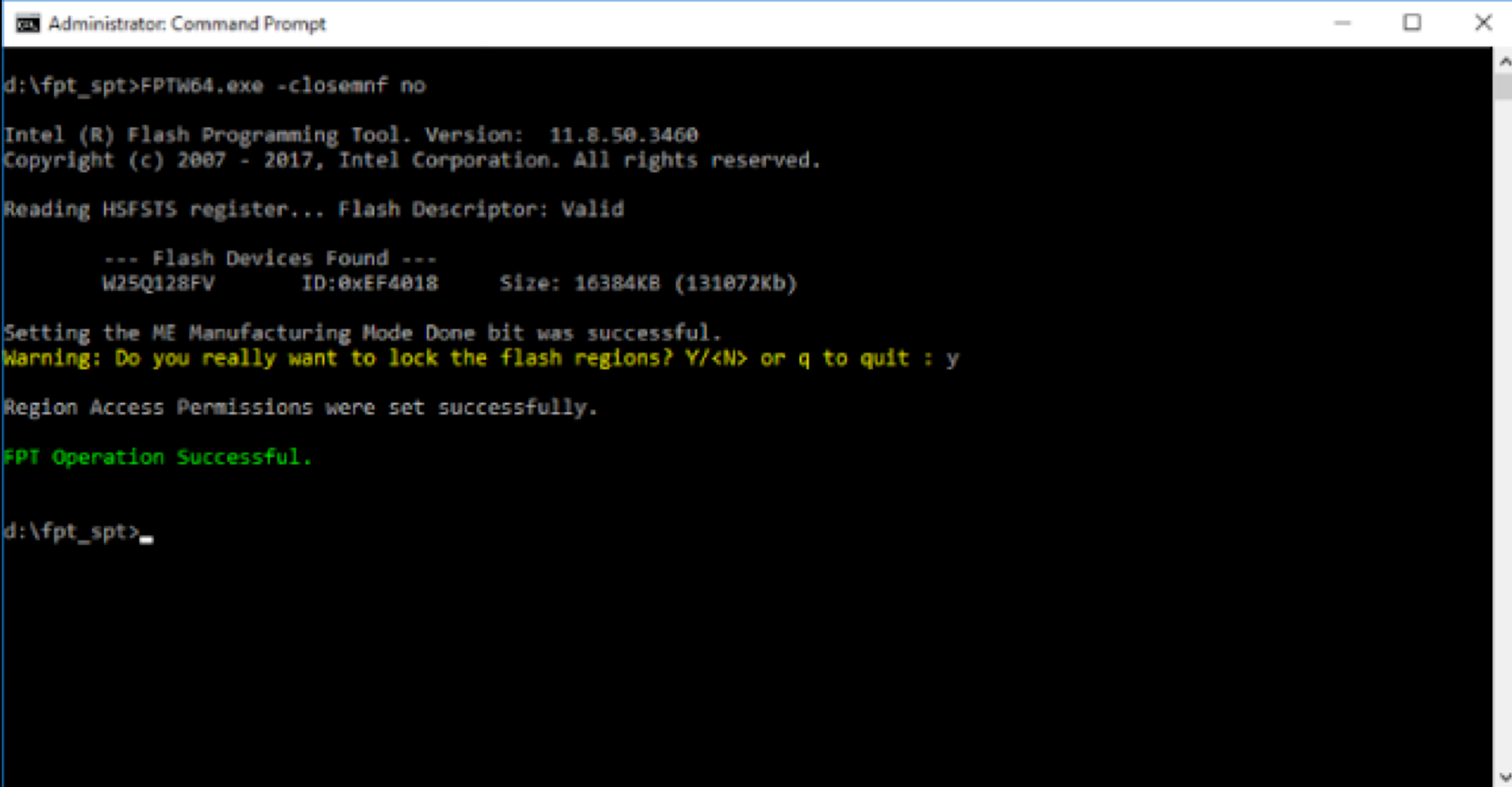


The screenshot shows the GitHub release page for the project 'chipsec / chipsec'. The page is titled 'Chipsec v1.3.6' and was released by ErikBjorge 3 days ago. The release includes two assets: 'Source code (zip)' and 'Source code (tar.gz)'. A section titled 'New or Updated Modules:' lists several updates and additions:

- Updated `memconfig` to only check registers that are defined by the platform
- Updated `common.bios_smi` to check controls not registers
- Added `me_mfg_mode` module
- Added support for LoJax detection
- Updated `common.spi_lock` test support
- Added `sgx_check` module and register definitions
- Updates to DCI support in `debugenabled` module

<https://github.com/chipsec/chipsec/releases/tag/v1.3.6>

Disabling Manufacturing Mode



```
Administrator: Command Prompt

d:\fpt_spt>FPTW64.exe -closemnf no

Intel (R) Flash Programming Tool. Version: 11.8.50.3460
Copyright (c) 2007 - 2017, Intel Corporation. All rights reserved.

Reading HSFSTS register... Flash Descriptor: Valid

    --- Flash Devices Found ---
    W25Q128FV      ID:0xEF4018      Size: 16384KB (131072Kb)

Setting the ME Manufacturing Mode Done bit was successful.
Warning: Do you really want to lock the flash regions? Y/<N> or q to quit : y

Region Access Permissions were set successfully.

FPT Operation Successful.

d:\fpt_spt>
```

fitw64.exe -closemnf no

Q & A

<https://github.com/ptresearch>

<http://blog.ptsecurity.com>