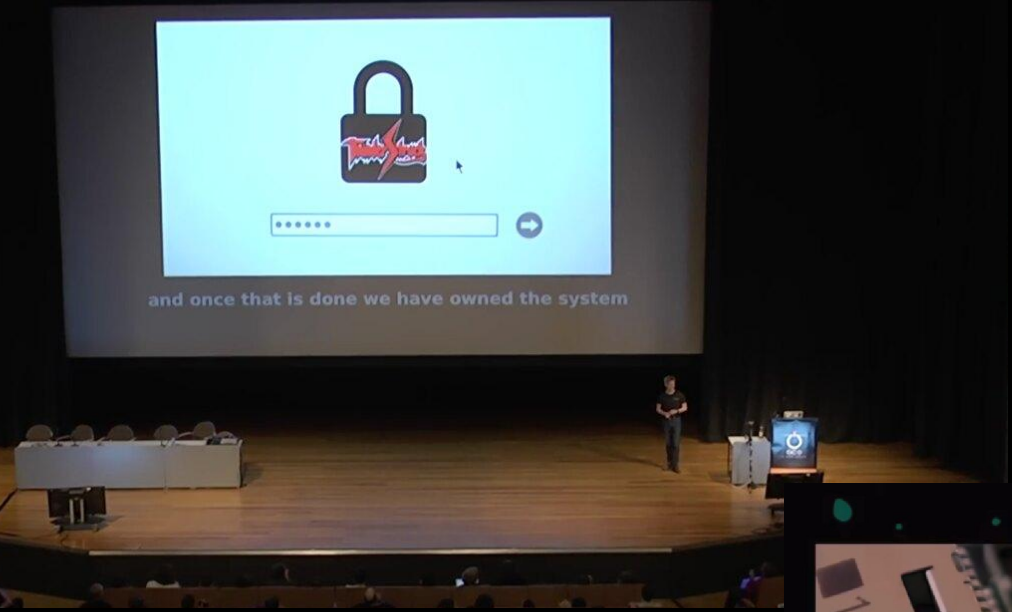
Now You See It...

TOCTOU Attacks Against BootGuard

Peter Bosch & Trammell Hudson


FabCreate



Boot strapping slightly more secure systems

Trammell Hudson @grs

A6C7 4E34 1054 A169 CE52
BE5F B65B FE54 0DEF 86C0

Hardware & Firmware Threats

Platform Threats

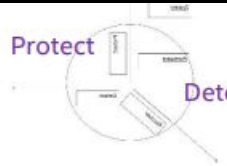
- BIOS Malware
- UEFI Rootkits
- Bootkits
- SMM Rootkits
- Device FW Malware
- ACPI Rootkits
- Option ROM Malware
- Evil Maid
- HVM Rootkits (Blue Pill)
- HW Trojans

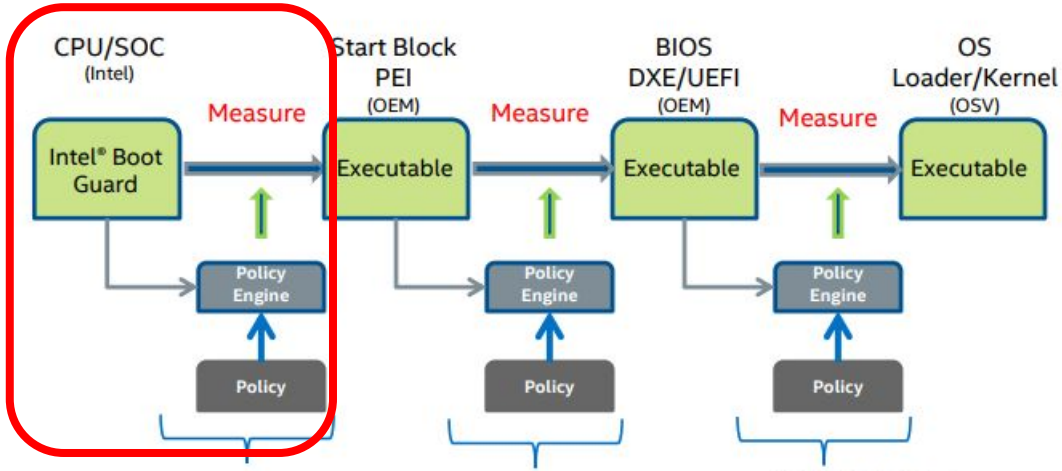
IDF15
INTEL DEVELOPER FORUM

1

http://vzimmer.blogspot.com/2013/09/where-do-i-sign-up.html

Full Verified Boot Sequence





Intel® Device Protection Technology with Boot Guard

http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/4th-gen-core-family-mobile-brief.pdf

OEM PI Verification Using PI Signed Firmware Volumes

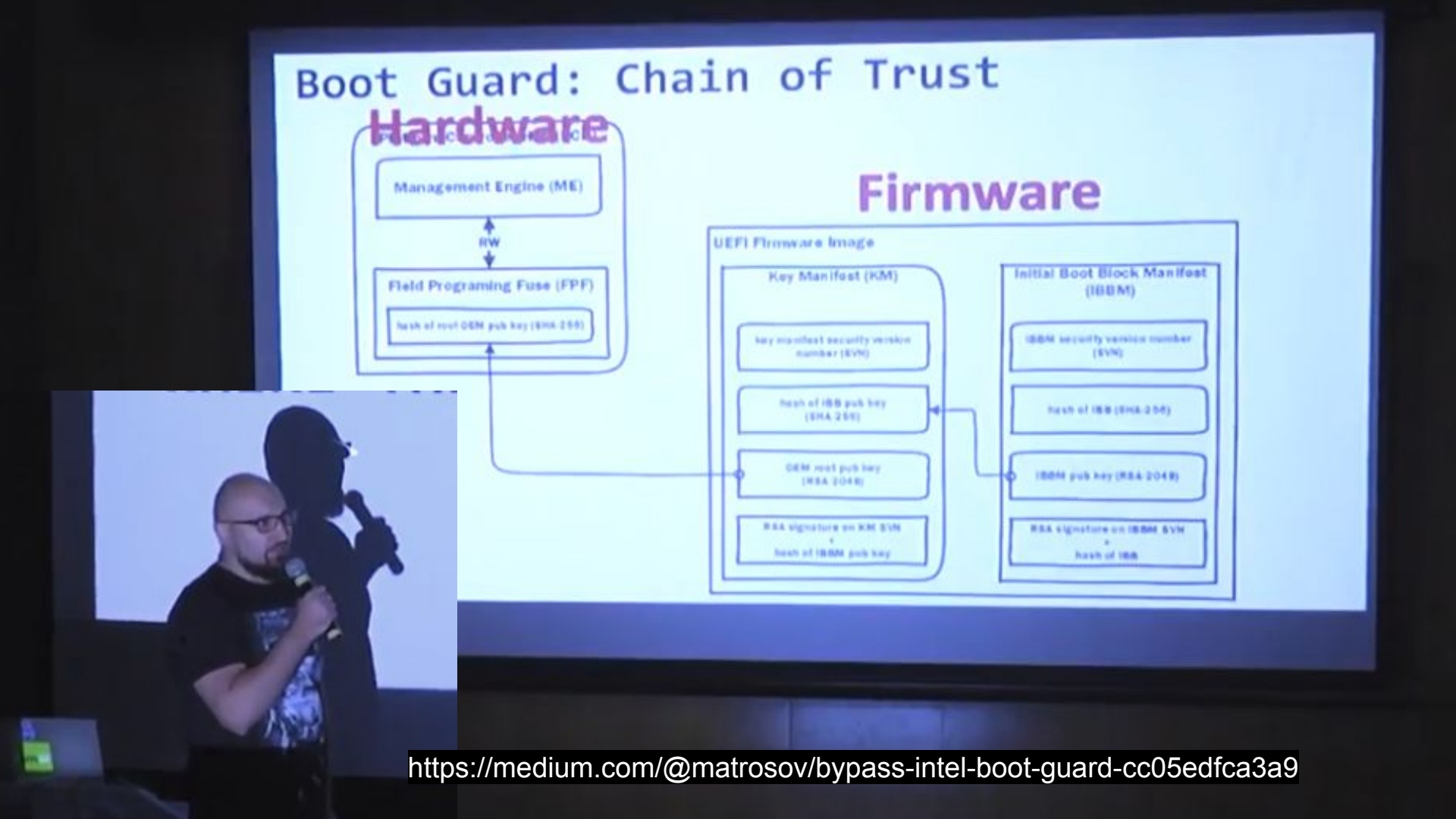
Vol 3, section 3.2.1.1 of PI 1.3 Specification

OEM UEFI 2.4 Secure Boot

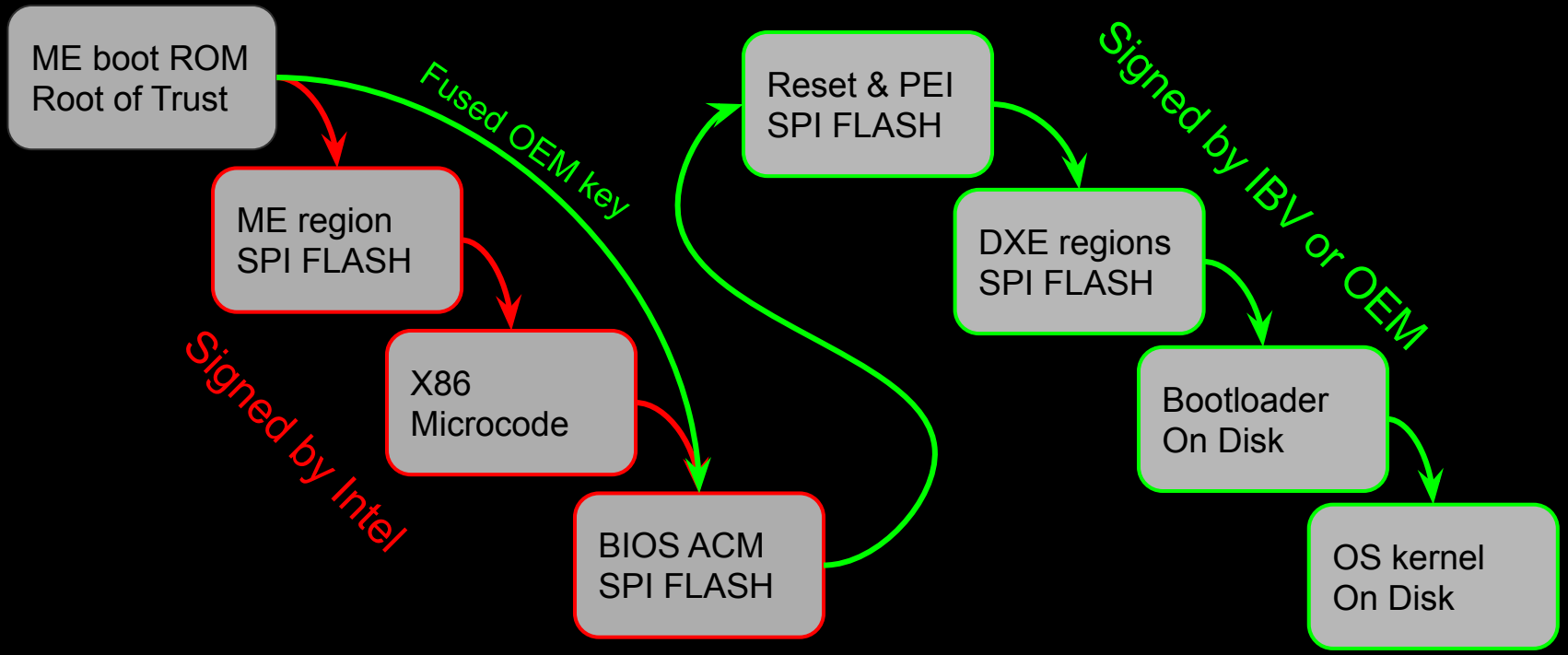
Chapter 27.2 of The UEFI 2.4 Specification

IDF15 INTEL DEVELOPER FORUM

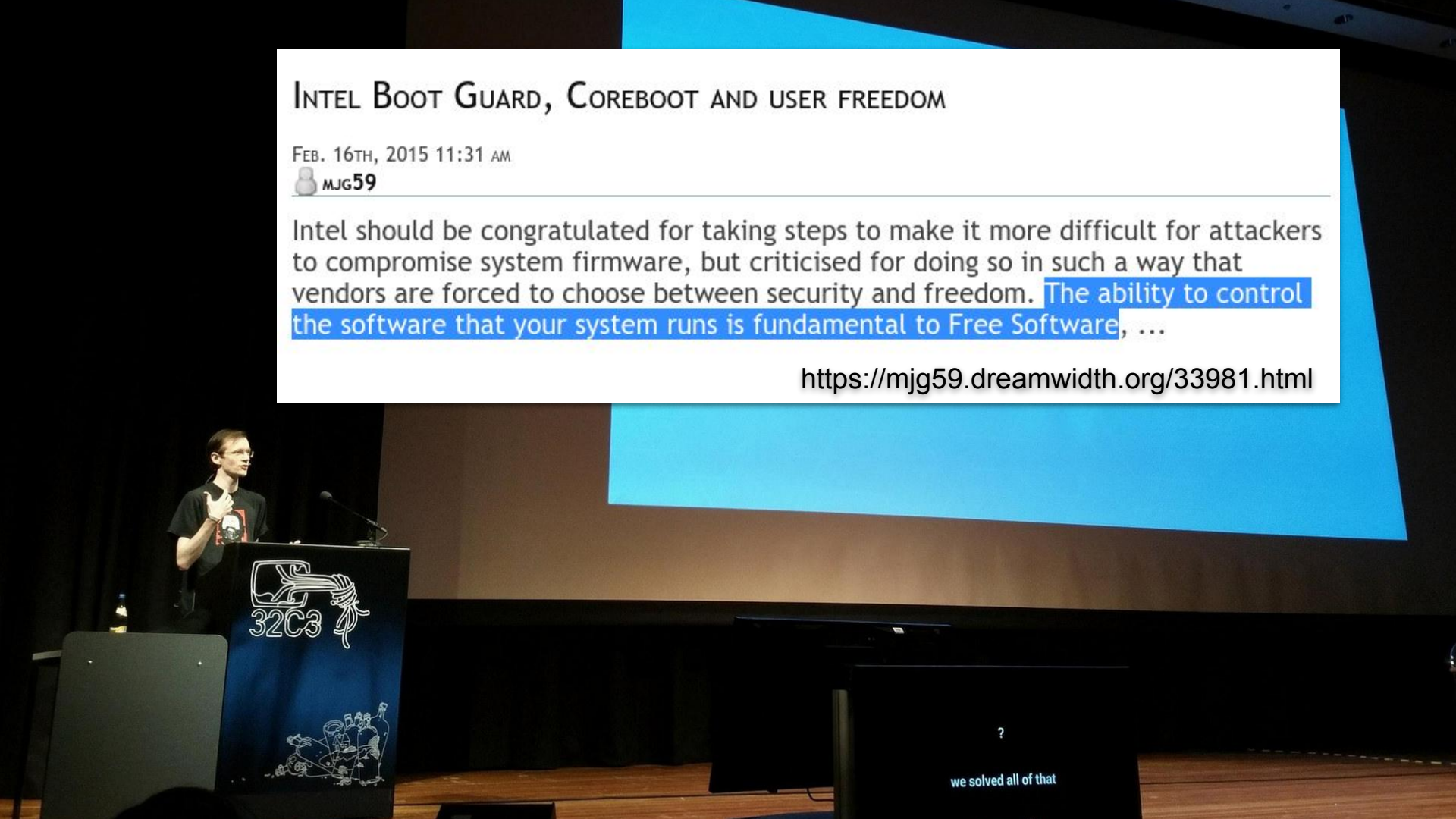
http://vzimmer.blogspot.com/2013/09/where-do-i-sign-up.html

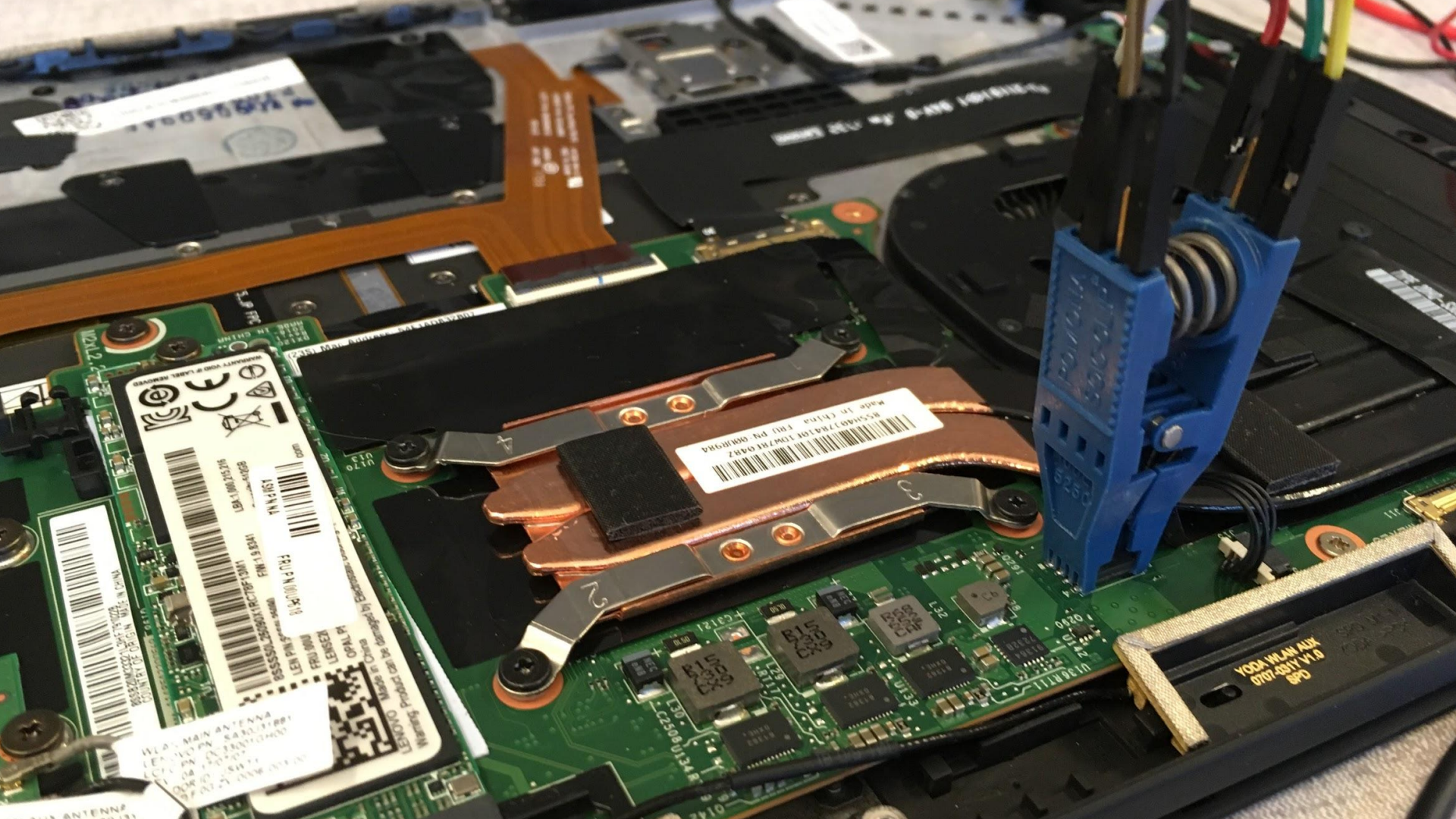

Boot Guard: Chain of Trust
Hardware
Management Engine (ME)
RW
Field Programing Fuse (FPF)
Firmware
UEFI Firmware Image
Key Manifest (KM)
Initial Boot Block Manifest (IBBM)
https://medium.com/@matrosov/bypass-intel-boot-guard-cc05edfca3a9

Chain of Trust (simplified)



Coreboot slide


Intel Boot Guard, Coreboot and user freedom
Feb. 16th, 2015 11:31 am
mjg59
Intel should be congratulated for taking steps to make it more difficult for attackers to compromise system firmware, but criticised for doing so in such a way that vendors are forced to choose between security and freedom. The ability to control the software that your system runs is fundamental to Free Software, ...
https://mjg59.dreamwidth.org/33981.html
32C3
?
we solved all of that


YODA WLAN AUX
0707-001Y V1.0
SPD

Don’t attack the algorithm,
Attack the implementation

https://2016.zeronights.ru/wp-content/uploads/2017/03/Intel-BootGuard.pdf

Safeguarding rootkits: Intel BootGuard

Alexander Ermolov

The issue

One day I found out that some systems have the SPI flash regions unlocked and the BootGuard configuration not set (nor enabled, nor disabled):

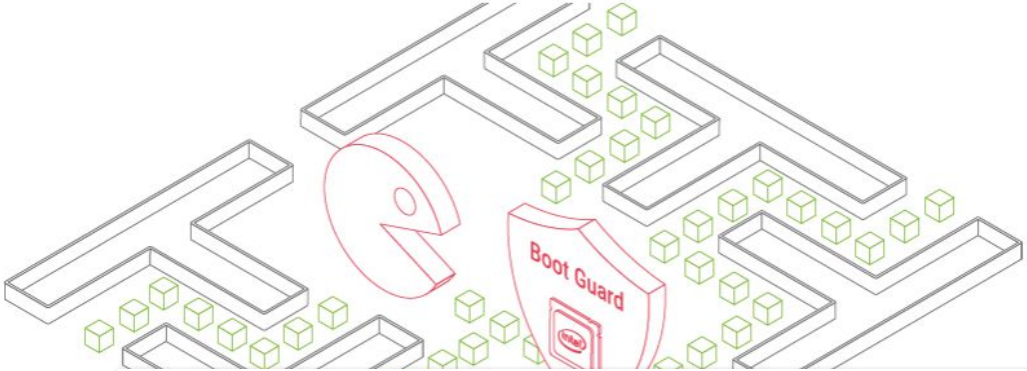
- All Gigabyte systems
- All MSI systems
- 21 Lenovo branded notebook machine types and 4 ThinkServer machine types
- other few vendors I cannot mention at the moment

That's because of the close manufacturing fuse was not set at the end of the manufacturing line.

EMBEDI

5 October, 2017

Bypassing Intel Boot Guard



Category: Resear
Tags: #intel, #vul

... if an attacker manages to delete the BootGuardDxe from the DXE volume, the protection of the DXE part will not work at all (there will be no code to check the results of the verification done by the IBB).

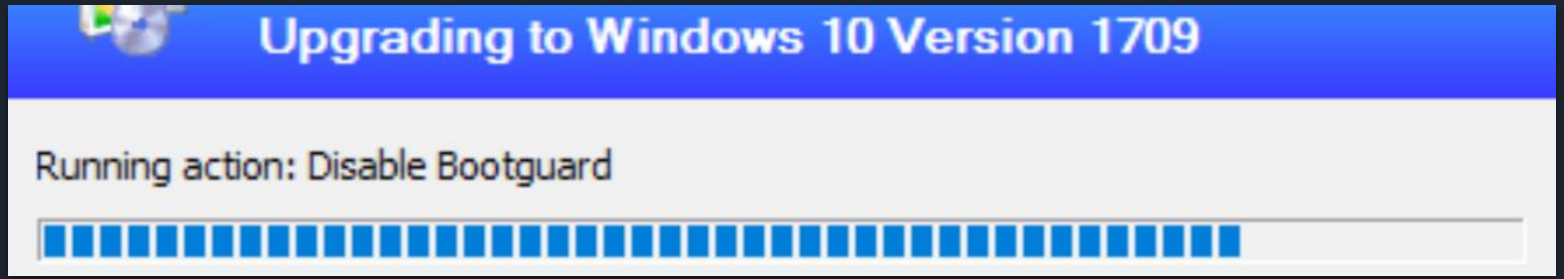
https://embedi.org/blog/bypassing-intel-boot-guard/

Alex Matrosov
@matrosov

Following

Slides "Modern Secure Boot Attacks: Bypassing Hardware Root of Trust from Software" from #BHASIA and #OPCDE2019 released! Lenovo keeps manufacturing mode Boot Guard "backdoor" to unlock DXE volume for arbitrary modifications. It fully breaks Secure Boot!

github.com/REhints/Public …



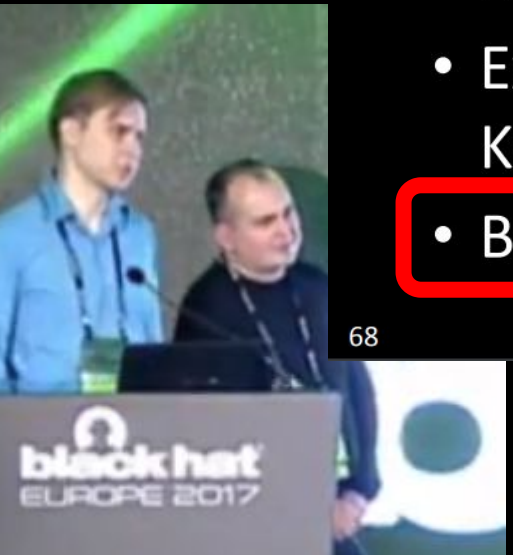
3:30 AM - 20 Apr 2019 https://twitter.com/matrosov/status/1119518664649211904

233 Retweets **451** Likes

“Chain of Trust” is only as secure as *every* link in the chain.

Our Achievements

- Switched-on AMT on non-vPro systems
- Activated JTAG for Intel ME via the vulnerability
- Dumped starter code (aka ROM)
- Recovered complete Huffman code for ME 11
- Extracted Integrity and Confidentiality Platform Keys [FFS17]
- Bypassed Intel Boot Guard



https://twitter.com/_markel___/status/984933953110007808

Common Vulnerabilities and Exposures

CVE-ID

CVE-2018-9062

Description

In some Lenovo ThinkPad products, one BIOS region is not properly included in the checks, allowing injection of arbitrary code.

References

- URL:http://www.securityfocus.com/bid/105387
- CONFIRM:https://support.lenovo.com/us/en/so

Assigning CNA

Lenovo Group Ltd.

CVE List CNAs Board About
News & Blog
Go to for:
CVSS Scores
CPE Info
Advanced Search

Common Vulnerabilities and Exposures

CVE-ID

CVE-2018-12169

Description

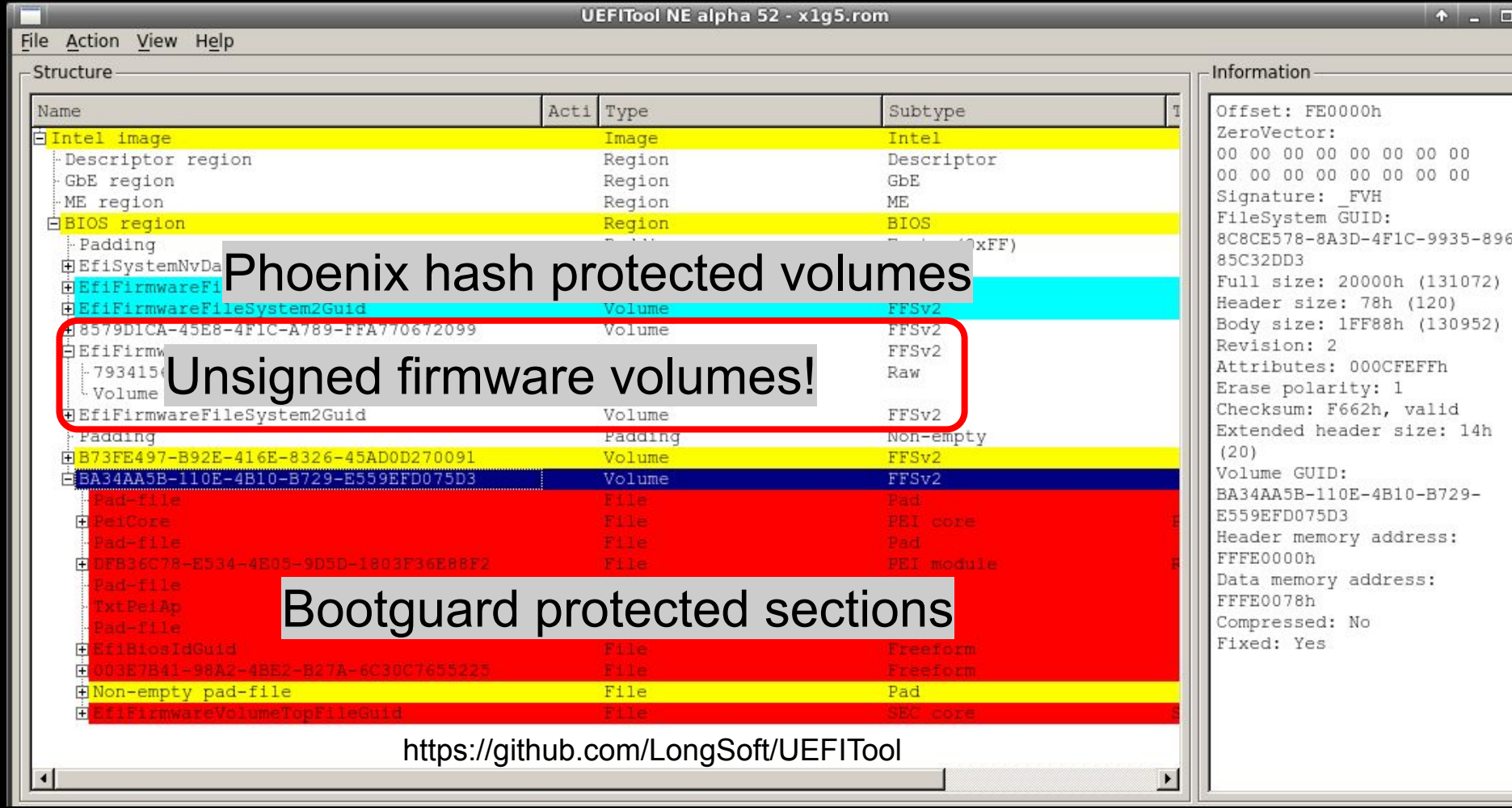
Platform sample code firmware in 4th Generation Intel Core Processor, 5th Generation Intel Core Processor, 6th Generation Intel Core Processor, 7th Generation Intel Core Processor and 8th Generation Intel Core Processor contains a logic error which may allow physical attacker to potentially bypass firmware authentication.

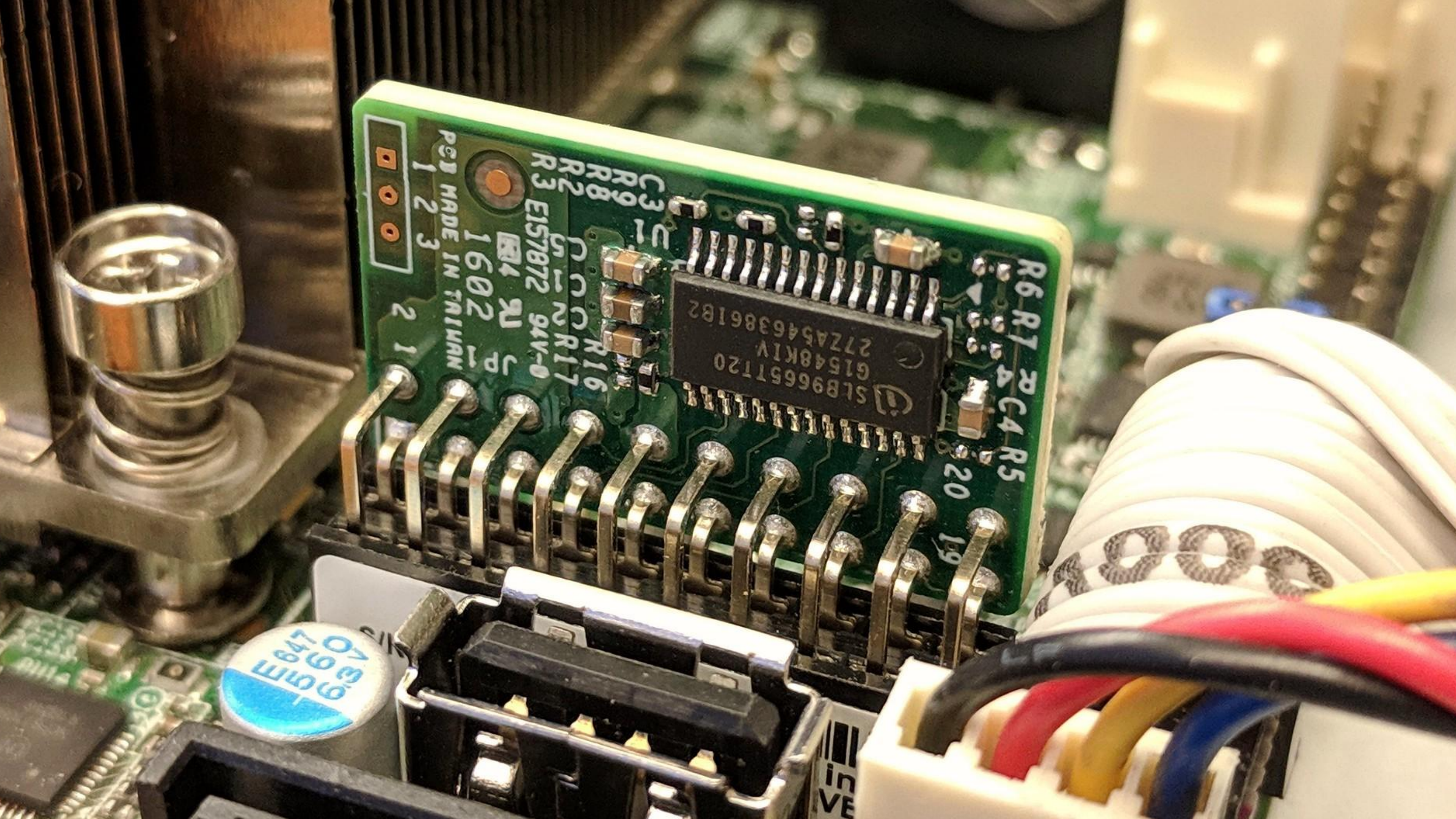
References

- URL:http://www.securityfocus.com/bid/105387
- CONFIRM:https://edk2-docs.gitbooks.io/security-advisory/content/unauthenticated-firmware-chain-of-trust-bypass.html
- CONFIRM:https://support.lenovo.com/us/en/solutions/LEN-20527

Assigning CNA

Intel Corporation

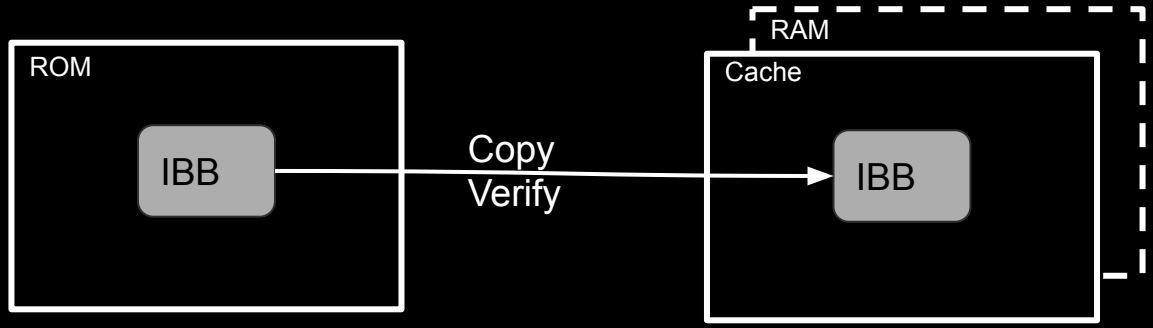

UEFITool NE alpha 52 - x1g5.rom
File Action View Help
Structure
Information
Name
Acti
Type
Subtype
Intel image
Image
Intel
Descriptor region
Region
Descriptor
GbE region
Region
GbE
ME region
Region
ME
BIOS region
Region
BIOS
Padding
EfiSystemNvDa
EfiFirmwareFi
EfiFirmwareFileSystem2Guid
Volume
FFSv2
8579D1CA-45E8-4F1C-A789-FFA770672099
Volume
FFSv2
EfiFirmw
FFSv2
793415
Raw
Volume
EfiFirmwareFileSystem2Guid
Volume
FFSv2
Padding
Padding
Non-empty
B73FE497-B92E-416E-8326-45AD0D270091
Volume
FFSv2
BA34AA5B-110E-4B10-B729-E559EFD075D3
Volume
FFSv2
Pad-file
File
Pad
PeiCore
File
PEI core
Pad-file
File
Pad
DFB36C78-E534-4E05-9D5D-1803F36E88F2
File
PEI module
Pad-file
TxtPeiAp
Pad-file
EfiBiosIdGuid
File
Freeform
003E7B41-98A2-4BE2-B27A-6C30C7655225
File
Freeform
Non-empty pad-file
File
Pad
EfiFirmwareVolumeTopFileGuid
File
SEC core
Offset: FE0000h
ZeroVector:
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
Signature: _FVH
FileSystem GUID:
8C8CE578-8A3D-4F1C-9935-8961
85C32DD3
Full size: 20000h (131072)
Header size: 78h (120)
Body size: 1FF88h (130952)
Revision: 2
Attributes: 000CFEFFh
Erase polarity: 1
Checksum: F662h, valid
Extended header size: 14h
(20)
Volume GUID:
BA34AA5B-110E-4B10-B729-
E559EFD075D3
Header memory address:
FFFE0000h
Data memory address:
FFFE0078h
Compressed: No
Fixed: Yes
Phoenix hash protected volumes
Unsigned firmware volumes!
Bootguard protected sections
https://github.com/LongSoft/UEFITool


MADE IN TAIWAN
E157872 94V-0
1602
SLB9665TT20

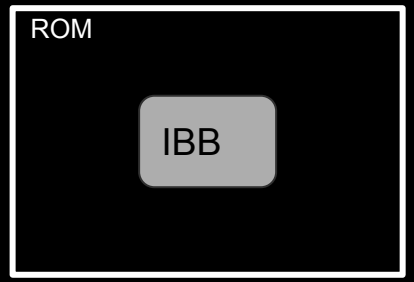
C.3 OEM Profile Parameters

Table C-3. Profile Parameters Description

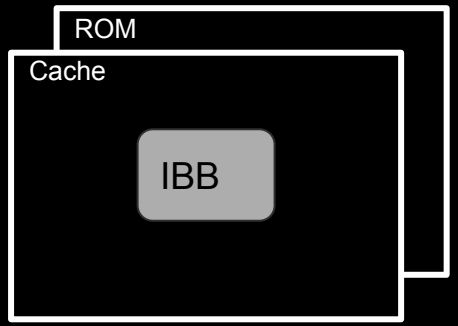
Parameter	Description	Settings
Protect Bios Environment Enabled (PBE)	Platform manufacturer may want Initial boot block to be protected between verification/ measurement and execution from attacks on buses and non-CPU components. Boot Guard accomplishes this by allowing the initial boot block to be verified and executed in LLC in NEM if PBE is enabled.	**false** - Take no actions to control the environment during execution of the BIOS components (default) **true** - Takes actions to control the environment during the execution of the BIOS components.


RAM
ROM
Cache
IBB
Copy
Verify
IBB

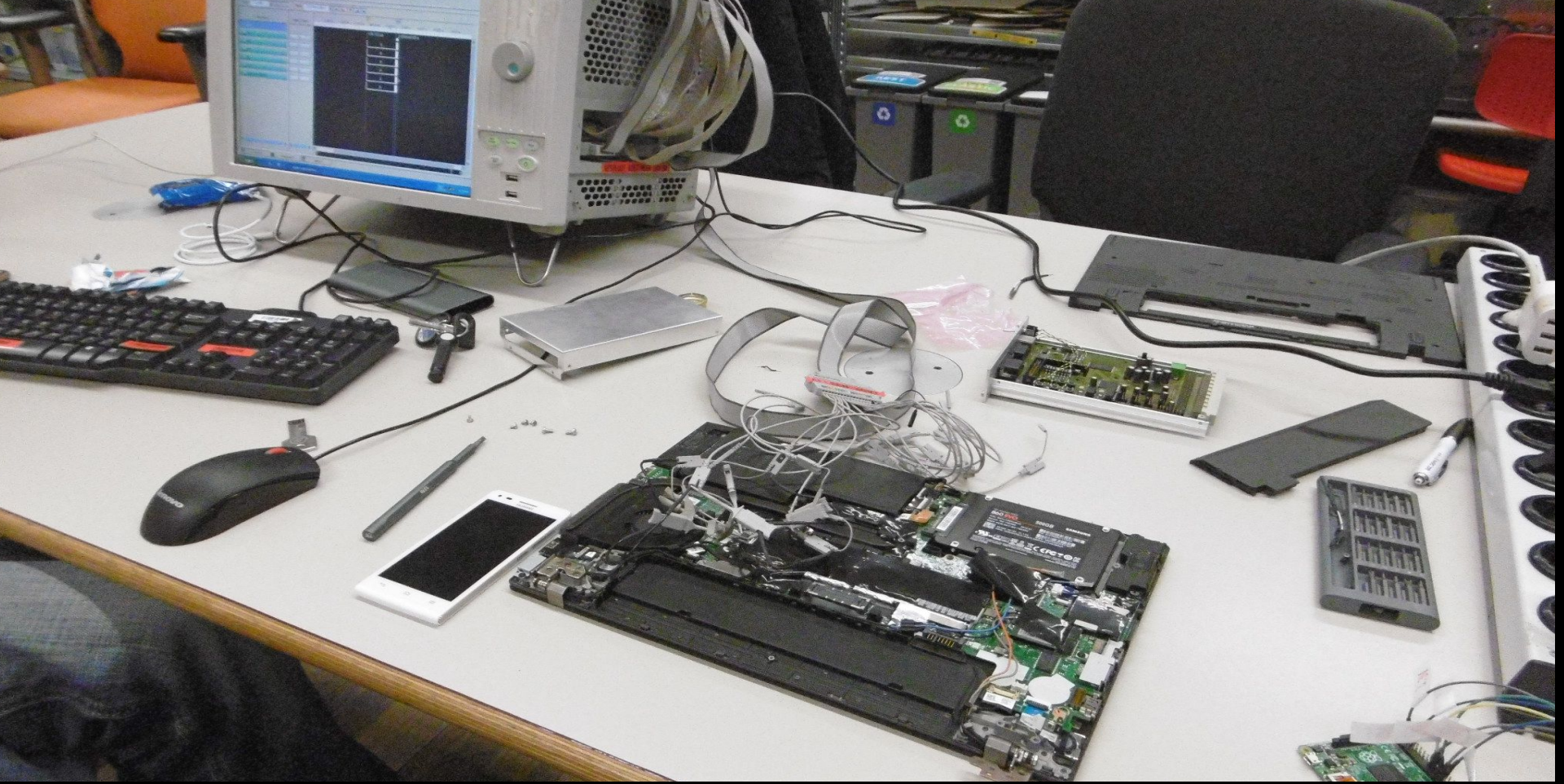
1. Start ACM

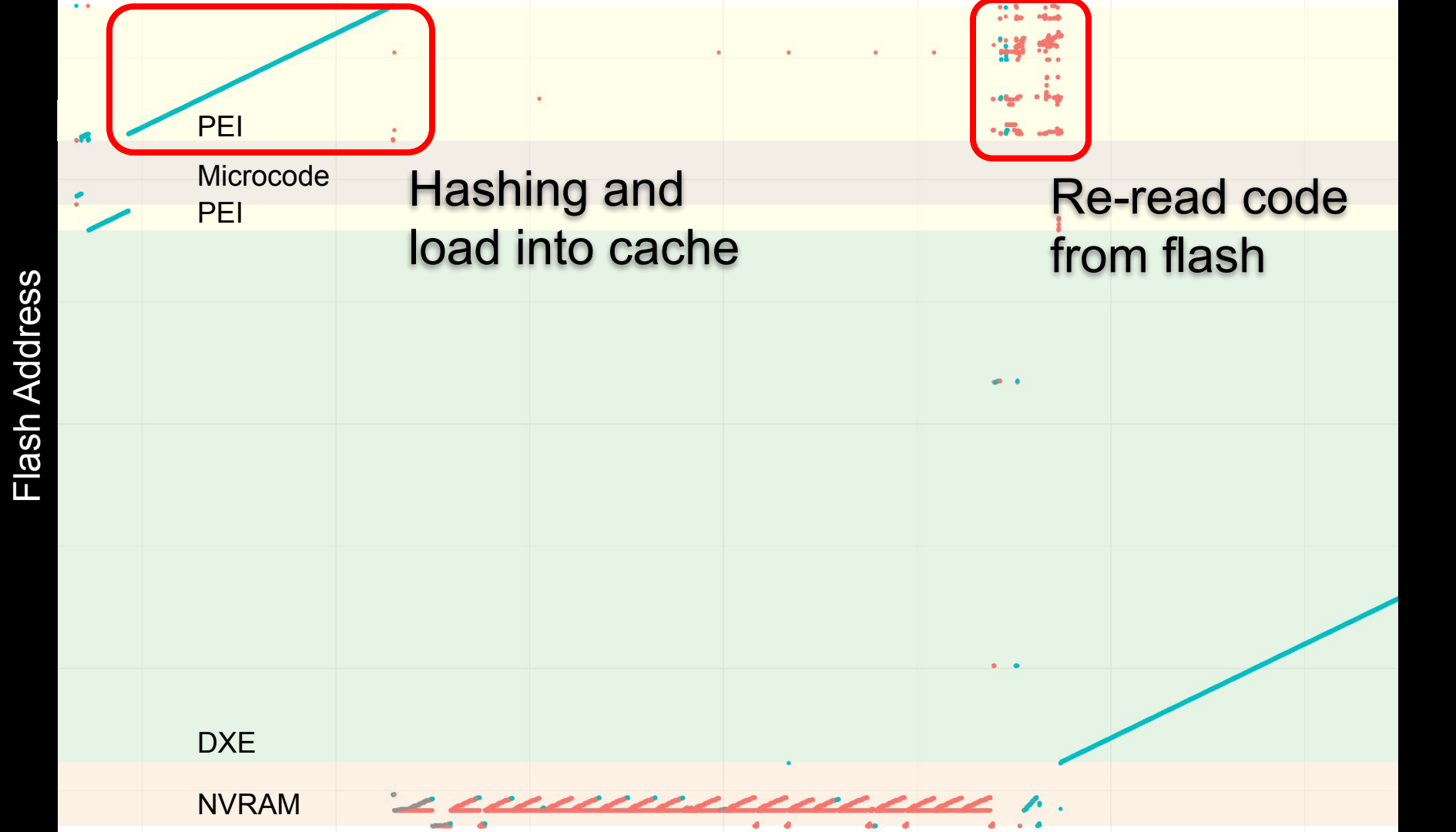


1. Start ACM
2. Verify IBB



What could go wrong?




Flash Address
PEI
Microcode
PEI
Hashing and
load into cache
Re-read code
from flash
DXE
NVRAM

```
0xec9ac0 SiInitPreMem-pe32 +4324 15
0xec9a80 SiInitPreMem-pe32 +42e4 18
0xec9ac0 SiInitPreMem-pe32 +4324 16
0xec9a80 SiInitPreMem-pe32 +42e4 19
0xf16cc0 SiInitPreMem-pe32 +51524 2
0xf9e200 TraceHubStatusCodeHandlerPei-pe32 +2044 1
0xffcc40 SecCore-pe32 +1434 1
0xffcc80 SecCore-pe32 +1474 1
0xfe0570 PeiCore-pe32 +454 0
0xfe0340 PeiCore-pe32 +224 1
0xfe057c PeiCore-pe32 +460 0
0xfe0440 PeiCore-pe32 +324 1
0xfa9940 SystemErrorLogPei-pe32 +2a4 1
0xfa9900 SystemErrorLogPei-pe32 +264 1
```

SecCore::PeiTemporaryRamDone

```
FFFFCC42 mov    ecx, IA32_MTRR_DEF_TYPE
FFFFCC47 rdmsr
FFFFCC49 and    eax, ~IA32_MTRR_ENABLE
FFFFCC5A mov    ecx, IA32_MTRR_DEF_TYPE
FFFFCC5F wrmsr
```

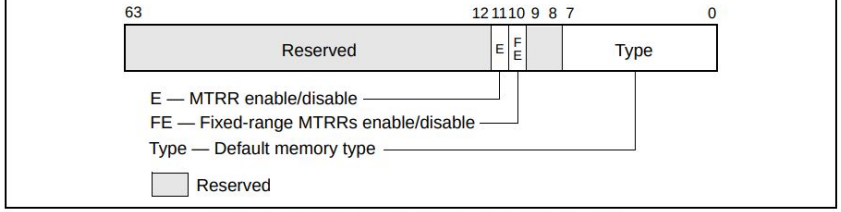
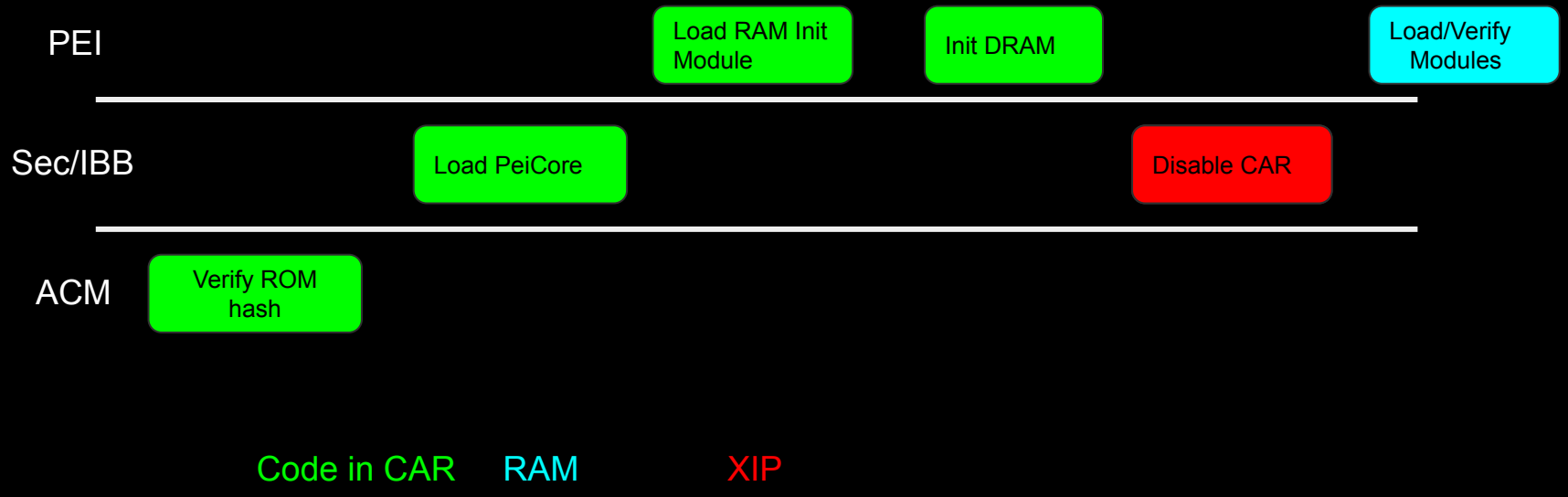
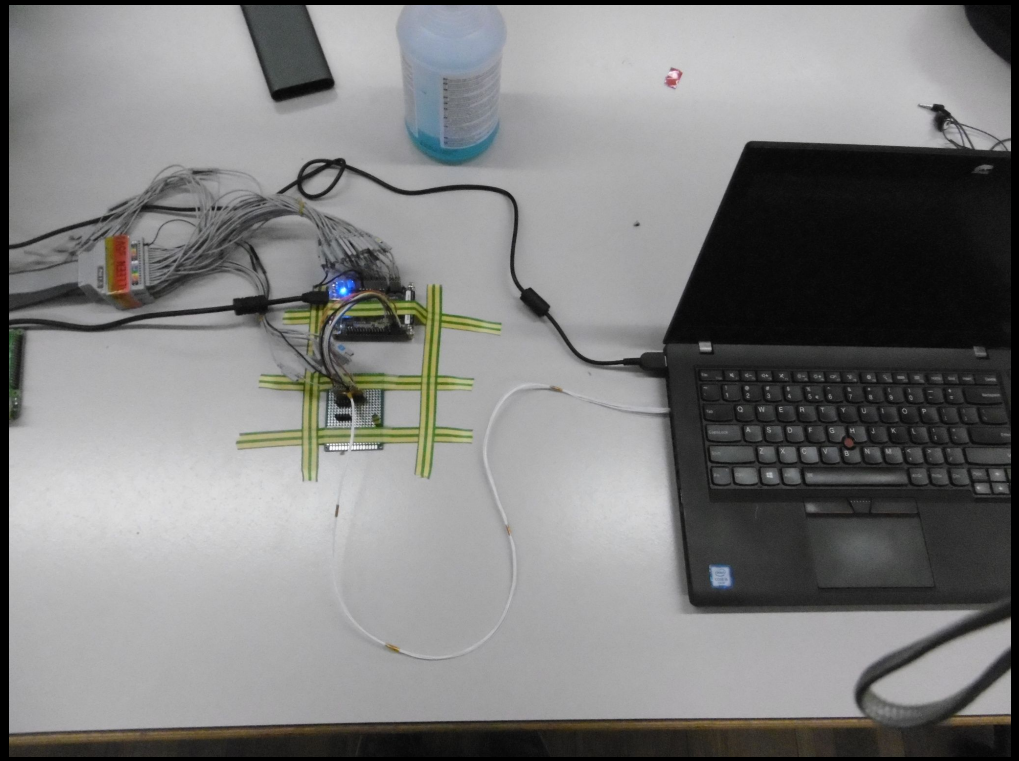


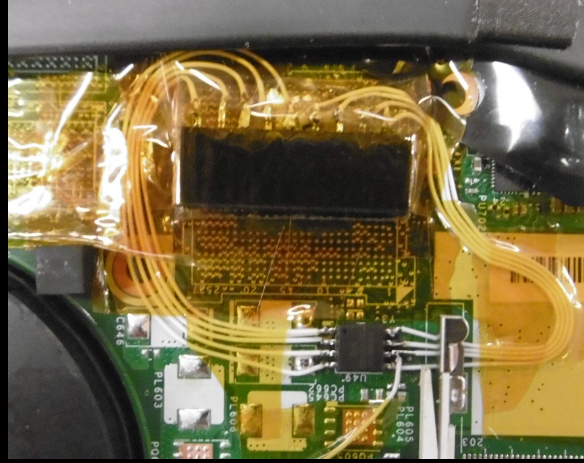
Figure 11-6. IA32_MTRR_DEF_TYPE MSR

- **E (MTRRs enabled) flag, bit 11** — MTRRs are enabled when set; all MTRRs are disabled when clear, and the UC memory type is applied to all of physical memory. When this flag is set, the FE flag can disable the fixed-range MTRRs; when the flag is clear, the FE flag has no affect. When the E flag is set, the type specified in the default memory type field is used for areas of memory not already mapped by either a fixed or variable MTRR.

Early Boot: ACM, Sec and PEI Phases





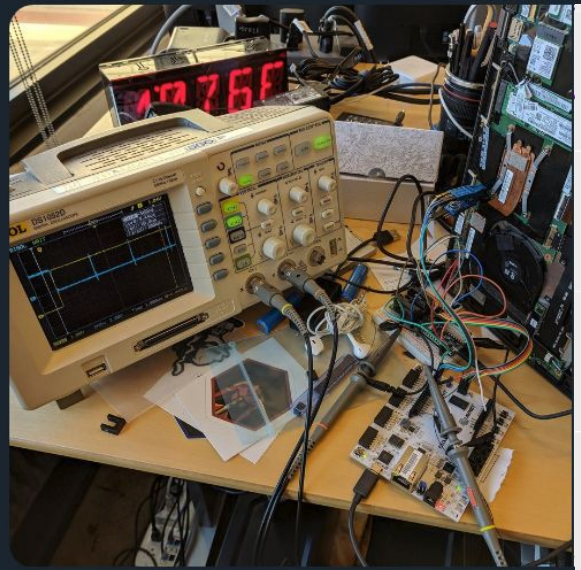


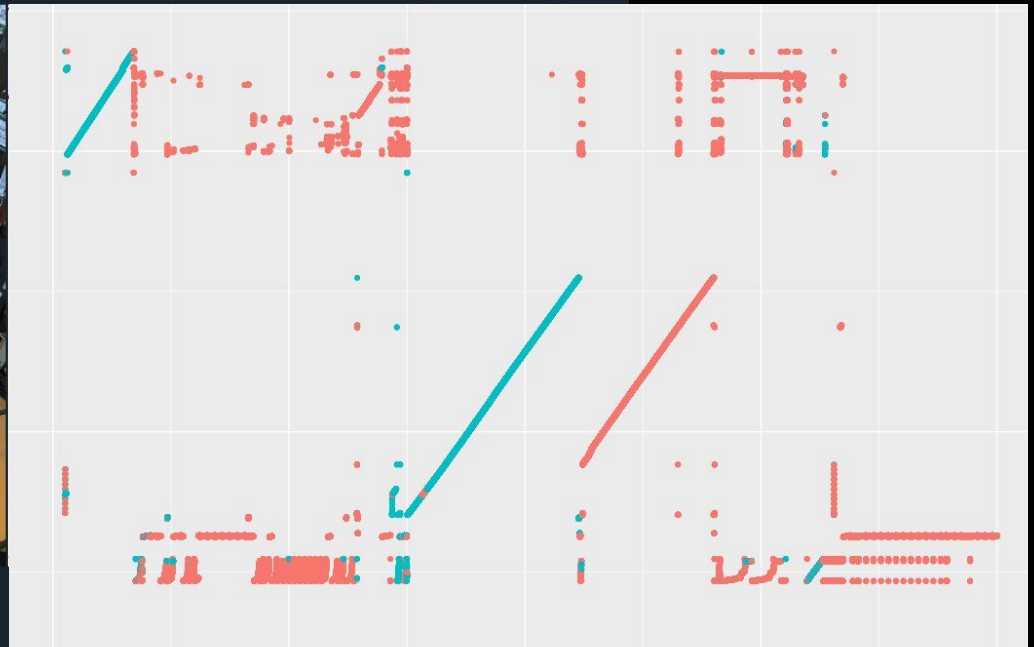
```
C:\Documents and Settings\labuser\My Documents\Agilent Technologies\Logi
er\Export Files>python ttyout.py cpu_rdmsr2.csv
Hello World F
1234ABCD
```


OPEN SOURCE
FIRMWARE
Get rid of your BIOS
9ELEMENTS
CYBER Security
coreboot
tianocore
U-Boot
LinuxBoot

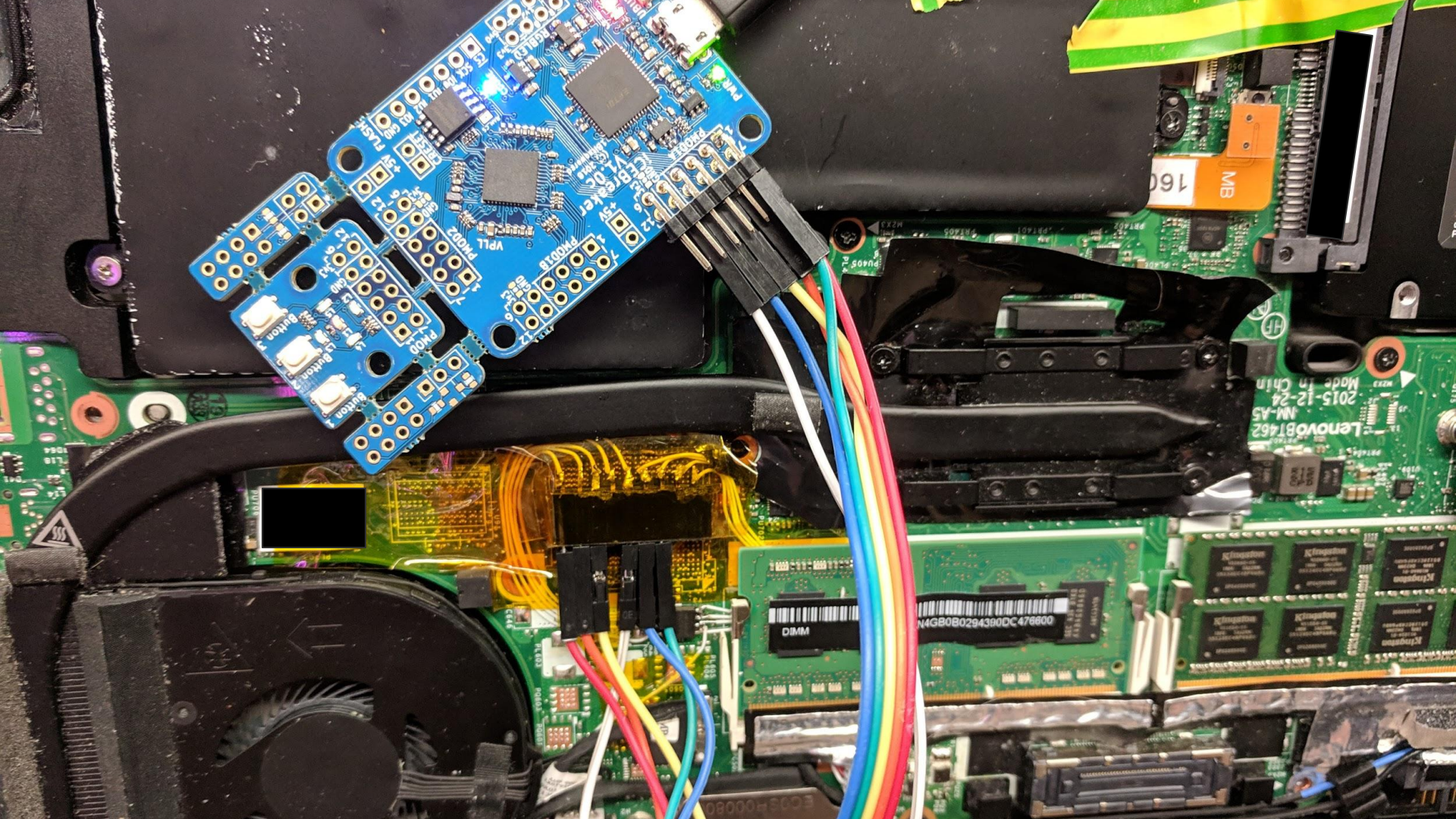
Trammell Hudson
@qrs

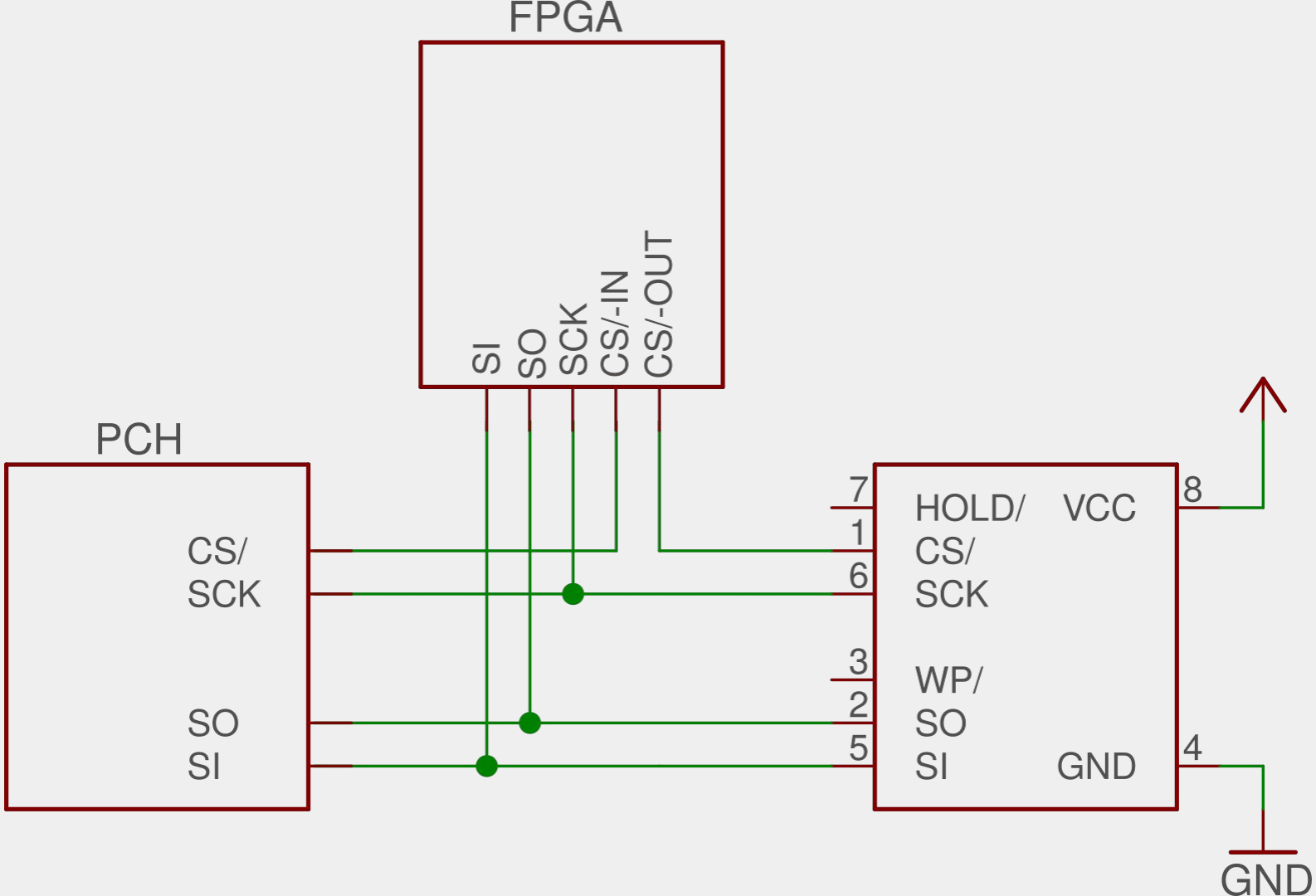
Using an FPGA to record the flash memory accesses during boot shows some interesting patterns of re-reading the same data from the UEFI BIOS region multiple times.

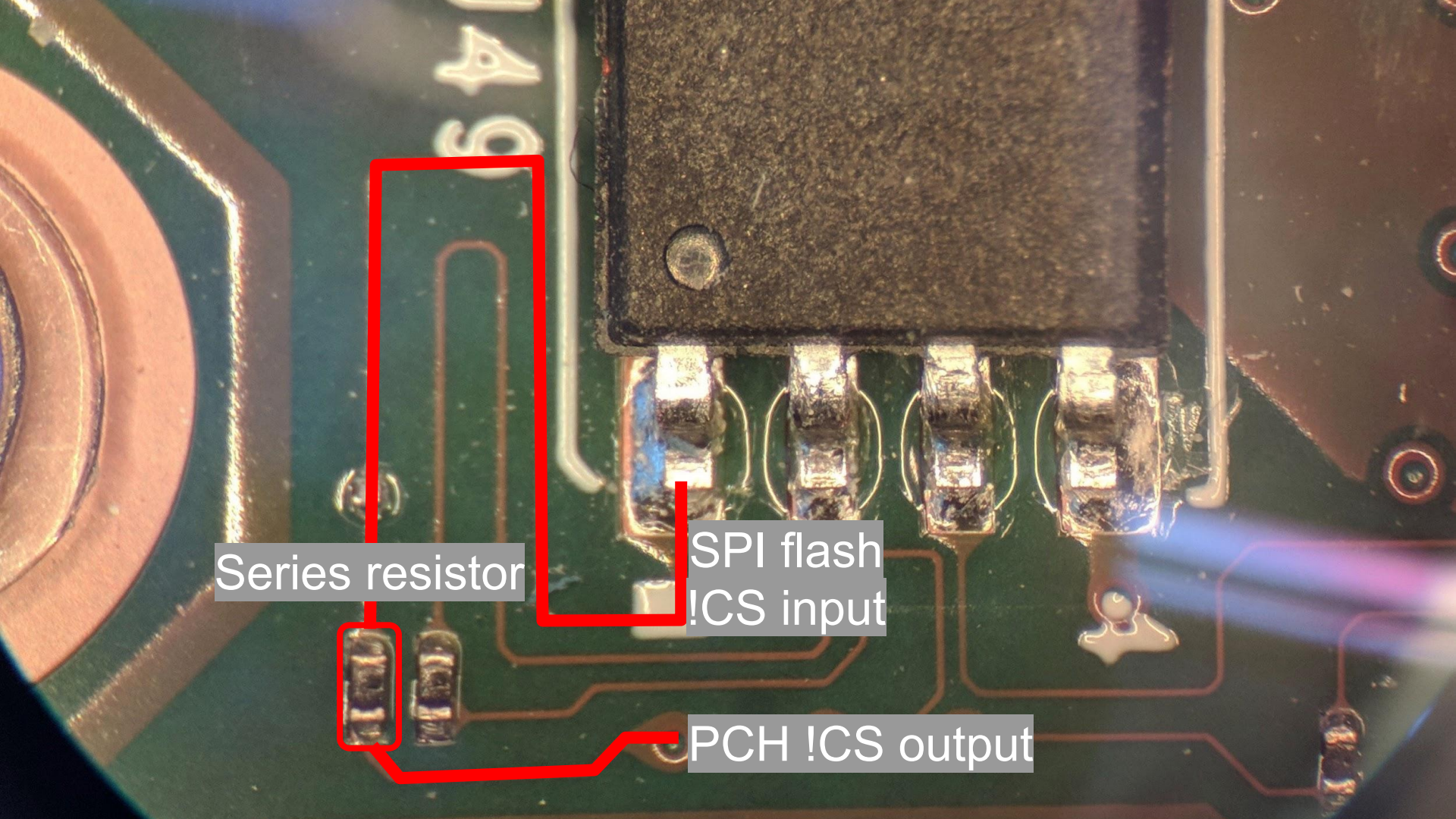


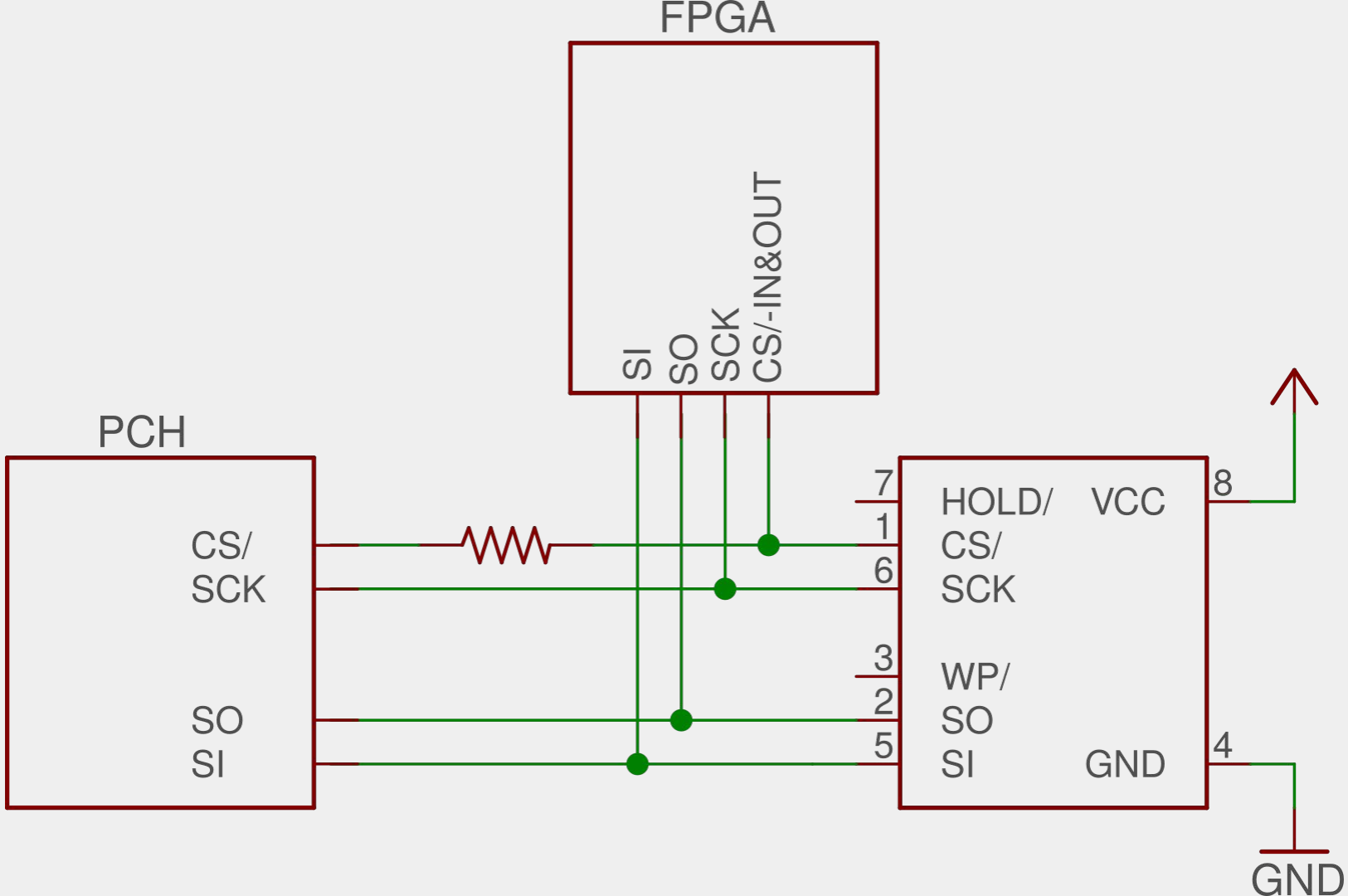


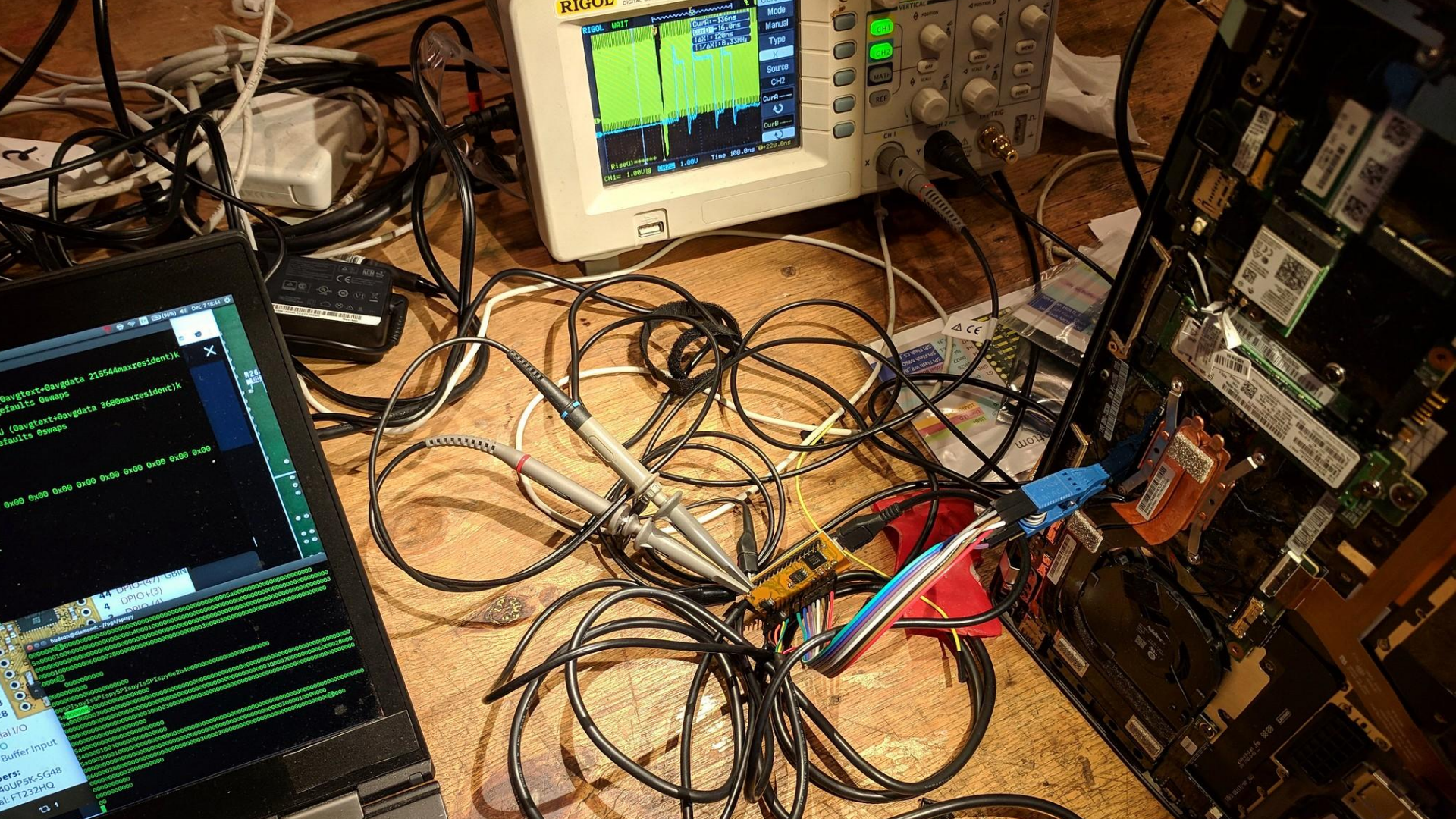
5:49 PM - 11 Jun 2018

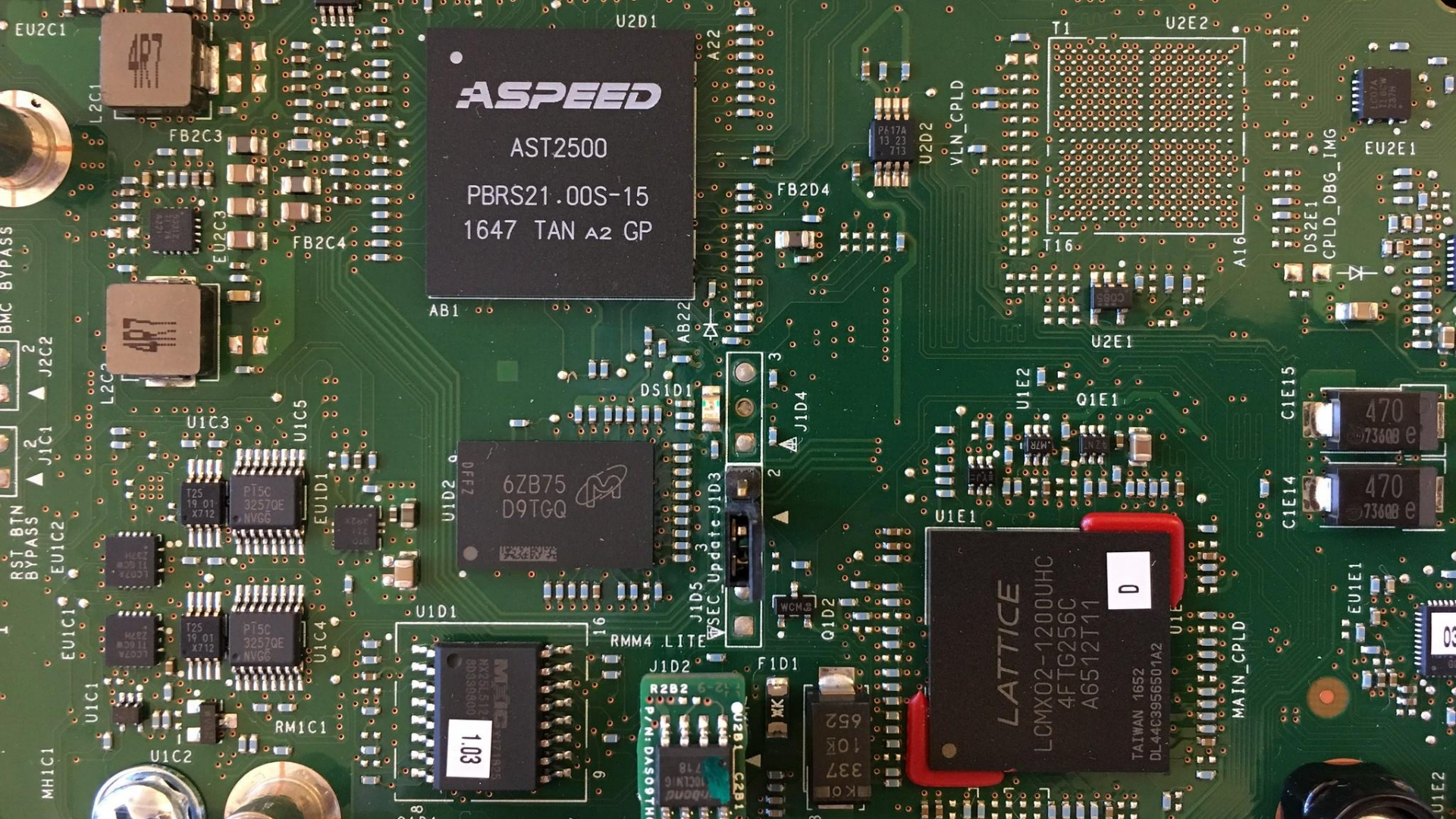

DIMM

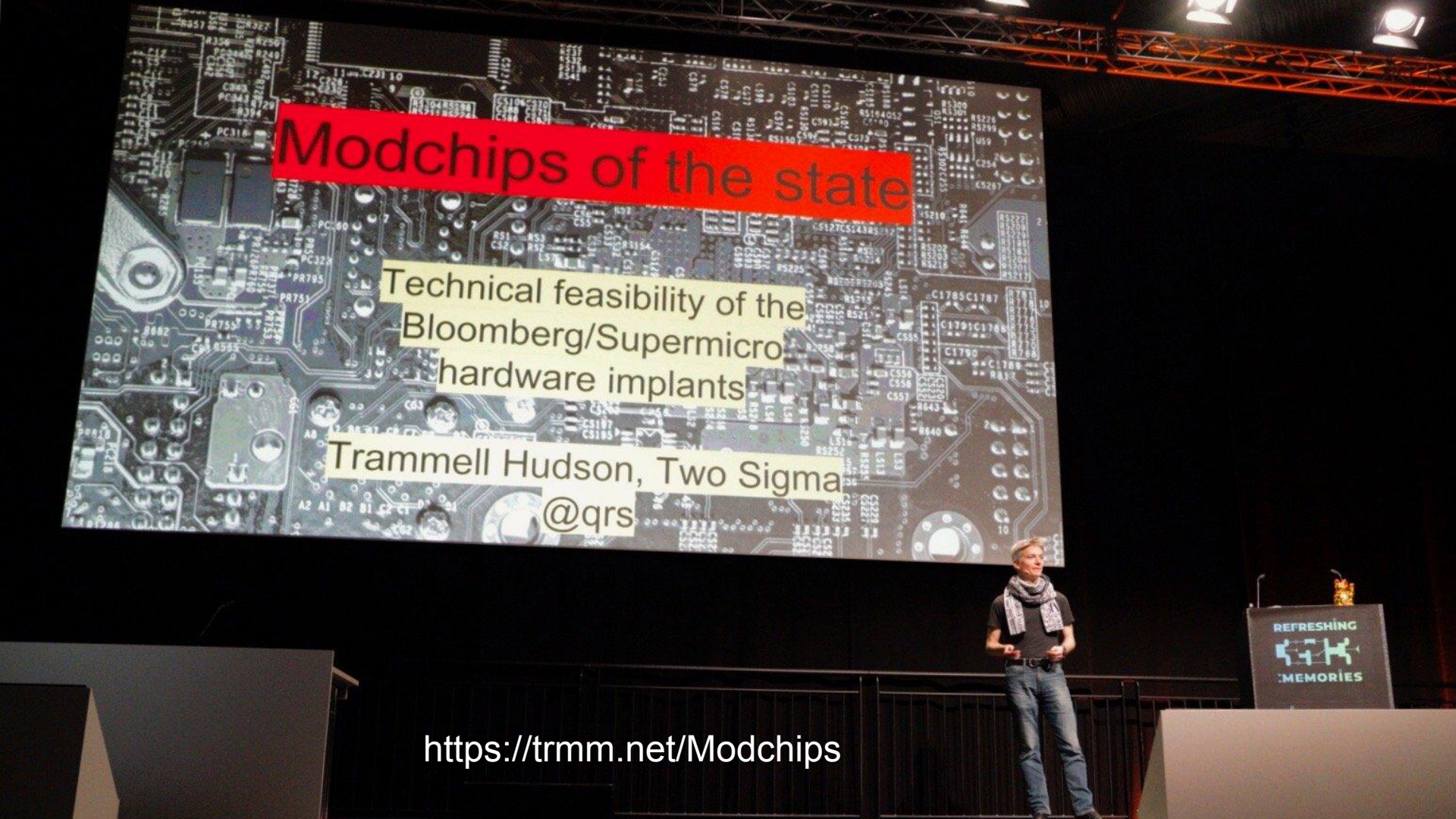

FPGA
SI
SO
SCK
CS/-IN
CS/-OUT
PCH
CS/
SCK
SO
SI
7
1
6
3
2
5
HOLD/
CS/
SCK
WP/
SO
SI
VCC
GND
8
4
GND


Series resistor
SPI flash
!CS input
PCH !CS output

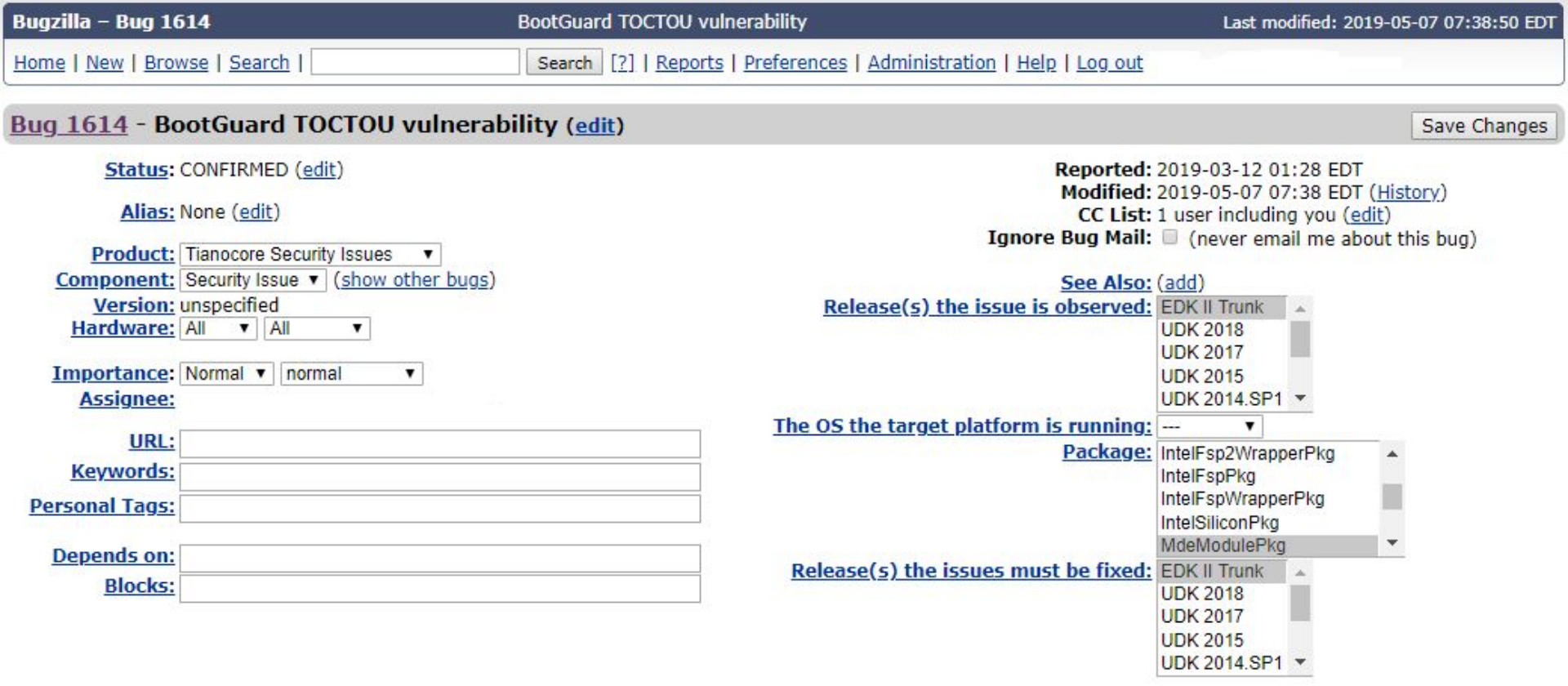

FPGA
SI
SO
SCK
CS/-IN&OUT
PCH
CS/
SCK
SO
SI
7
1
6
3
2
5
HOLD/
CS/
SCK
WP/
SO
SI
VCC
GND
8
4
GND


RIGOL
RIGOL WAIT
Mode
Manual
Type
Source
CH2
CurA
CurB
CH1
CH2
MATH
REF
VERTICAL
HORIZONTAL
Rise(1)*****
CH1 1.00V
MEAS 1.00V
Time 100.0ns @-220.0ns


ASPEED
AST2500
PBRS21.00S-15
1647 TAN A2 GP
U2D1
AB1
A22
AB22
U2D2
VLN_CPLD
T1
U2E2
T16
A16
U2E1
EU2E1
DS2E1
CPLD_DBG_IMG
FB2D4
EU2C1
L2C1
FB2C3
FB2C4
EU2C3
L2C3
BMC BYPASS
J2C2
J1C1
RST BTN BYPASS
U1C3
U1C5
EU1D1
EU1C2
EU1C1
U1C4
U1C1
RM1C1
U1C2
MH1C1
DS1D1
6ZB75
D9TGQ
U1D2
J1D4
J1D3
J1D5
SEC_Update
U1D1
RMM4 .LITE
J1D2
R2B2
F1D1
Q1D2
U1E1
LATTICE
LCMX02-1200UHC
4FTG256C
A6512T11
TAIWAN 1652
DL44C03956501A2
MAIN_CPLD
Q1E1
U1E2
C1E15
C1E14
470
EU1E1
1.03


Modchips of the state
Technical feasibility of the Bloomberg/Supermicro hardware implants
Trammell Hudson, Two Sigma
@qrs
REFRESHING
MEMORIES
https://trmm.net/Modchips

Response and Mitigations



Intel CVE-2019-11098

In addition to the call stack, the PEI Foundation will copy the following from temporary to permanent memory:

- PEI Foundation private data
- PEI Foundation heap
- HOB list
- Installed Firmware Volumes

https://uefi.org/sites/default/files/resources/PI_Spec_1_7_final_Jan_2019.pdf

Any permanent memory consumed in this fashion by the PEI Foundation will be described in a HOB, which the PEI Foundation will create.

The PEI Foundation will copy any installed firmware volumes from the temporary memory location to a permanent memory location with the alignment specified in the firmware volume header. Any *uncompressed* PE32 or TE sections within PEIMs in these firmware volumes will be fixed up. This ensures any static **EFI_PEI_PPI_DESCRIPTOR**s or PPI interface pointers in these PEIMs point to the permanent memory addresses.

In addition, if there were any **EFI_PEI_PPI_DESCRIPTOR**s created in the temporary memory heap or declared statically in PEIMs, their respective locations have been translated by an offset equal to the difference between the original location in temporary memory and the destination location in permanent memory. In addition to this heap copy, the PEI Foundation will traverse the PEI PPI database. Any references to **EFI_PEI_PPI_DESCRIPTOR**s that are in temporary

Why open source firmware is important

Jessie Frazelle - @jessfraz

https://blog.jessfraz.com/post/why-open-source-firmware-is-important-for-security/

https://coreboot.org/
https://www.linuxboot.org/
http://osresearch.net/


SPISpy coming soon!
Peter Bosch
@peterbjornx
Trammell Hudson
@qrs