BETRAYING THE BIOS: WHERE THE GUARDIANS OF THE BIOS ARE FAILING

Alex Matrosov
@matrosov
Have a lot of fun with UEFI Security and RE at

Former Security Researcher @Intel

Reverse Engineering since 1997

Book co-author nostarch.com/rootkits

@matrosov
➢ Intro

➢ Attacks on BIOS Updates
  ✓ Unsigned Updates
  ✓ BIOS protection bits
  ✓ SmiFlash and SecSmiFlash

➢ Intel Boot Guard
  ✓ AMI implementation details
  ✓ Discover ACM secrets
  ✓ Vulns
  ✓ Boot Guard Bypass!

➢ Intel BIOS Guard
  ✓ AMI implementation details
All rootkits want to get into Ring 0

Before OS loads

MBR/VBR (bootcode) → Loader (bootcode) → Rootkit (ring 0)

SMM (ring -2) → Loader (ring -2) → Rootkit (ring 0)

Operating System

Loader (ring 3) → Rootkit (ring 0)
More mitigations, more rootkits complexity
BIOS Update Issues
No more legacy! UEFI is everywhere!!
How many different firmware's inside BIOS update?
All the vulnerabilities mention in this research found inside AMI-based UEFI firmware’s
All Guardians of the BIOS on one slide

- CVE-2017-3197
- CVE-2017-11312
- BIOS_CNTL
- BIOS Lock Bit (BLE)
- BIOS_WE
- SMM_BWP
- FLOCKDN
- SPI Write Protection (PRx)
- ACM/Microcode
- ASUS
- Gigabyte
- MSI
- UEFI
- Firmware
- Image
- BIOS Guard
- Boot Guard
- Signed Image
- CVE-2017-3198
- CVE-2017-11313
- CVE-2017-11314
How different vendors care about security?

<table>
<thead>
<tr>
<th>Vendor Name</th>
<th>BLE</th>
<th>SMM_BWP</th>
<th>PRx</th>
<th>Authenticated Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUS</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MSI</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Gigabyte</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dell</td>
<td>+</td>
<td>+</td>
<td>-+</td>
<td>+</td>
</tr>
<tr>
<td>Lenovo</td>
<td>+</td>
<td>+</td>
<td>RP</td>
<td>+</td>
</tr>
<tr>
<td>HP</td>
<td>+</td>
<td>+</td>
<td>RP/WP</td>
<td>+</td>
</tr>
<tr>
<td>Intel</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Apple</td>
<td>-</td>
<td>-</td>
<td>WP</td>
<td>+</td>
</tr>
</tbody>
</table>
Module: BIOS Interface Lock (including Top Swap Mode)

[*] BiosInterfaceLockDown (BILD) control = 1
[*] BIOS Top Swap mode is disabled (TSS = 0)
[*] RTC TopSwap control (TS) = 0
[+]: PASSED: BIOS Interface is locked (including Top Swap Mode)

[*] running module: chipsec.modules.common.bios_wp
[*] Module path: c:\Chipsec\chipsec\modules\common\bios_wp.pyc

Module: BIOS Region Write Protection

[*] BC = 0x08 << BIOS Control (b:d.f 00:31.0 + 0xDC)

[00] BIOSWE = 0 << BIOS Write Enable
[01] BLE = 0 << BIOS Lock Enable
[02] SRC = 2 << SPI Read Configuration
[04] TSS = 0 << Top Swap Status
[05] SMM BWP = 0 << SMM BIOS Write Protection

[-]: BIOS region write protection is disabled!

[*] BIOS Region: Base = 0x00A00000, Limit = 0x00FFFFFF
SPI Protected Ranges

<table>
<thead>
<tr>
<th>PRx (offset)</th>
<th>Value</th>
<th>Base</th>
<th>Limit</th>
<th>WP?</th>
<th>RP?</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR0 (74)</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PR1 (78)</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PR2 (7C)</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PR3 (80)</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PR4 (84)</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

[!] None of the SPI protected ranges write-protect BIOS region
!! WE NEED ALL YOUR PARTY INVITES !!

blackhat.com/docs/asia-17/materials/asia-17-Matrosov-The-UEFI-Firmware-Rootkits-Myths-And-Reality.pdf
Why so vulnerable?

- BIOS LOCK (BLE) not enabled
  (CLVA-2016-12-001/CVE-2017-3197)
  ✓ Attacker is able to modify BIOSWE bit
  ✓ Attacker can arbitrary write to SPI flash from OS

- FW update process don’t verify signature
  ✓ Attacker is able to abuse BIOS updater with signed driver

- SmiFlash Handler multiple vulns
  (CLVA-2016-12-002/CVE-2017-3198)
  ✓ Attacker can elevate privileges to SMM (ring -2)
How BIOS Update Guardians Fail?

1. OS Update App
2. OS Update Driver
3. SMM SmiFlash
4. SMM SecSmiFlash
5. Memory (DRAM) Update Image
6. SPI Flash Update Image

CVE-2017-3753
CVE-2017-11316
CVE-2017-11315
SMIFlash Handler Issues: Gigabyte, Lenovo, MSI

➢ SmiFlash HANDLERS (SMiFlash.efi) ➔ CVE-2017-3753, CVE-2017-11316

[BC327DBD-B982-4f55-9F79-056AD7E987C5]
✓ ENABLE 0x20
✓ READ 0x21
✓ ERASE 0x22
✓ WRITE 0x23
✓ DISABLE 0x24
✓ GET_INFO 0x25

➢ No checks for the input pointers
   SmmIsBufferOutsideSmmValid()
SecSMIFlash Handler Issues: ASUS

➢ SecSmiFlash HANDLERS (SecSMiFlash.elfi) → CVE-2017-11315
  [3370A4BD-8C23-4565-A2A2-065FEDE6080]
  ✓ LOAD_IMAGE 0x1d
  ✓ GET_POLICY 0x1e
  ✓ SET_POLICY 0x1f

➢ No checks for the input pointers
  SmmIsBufferOutsideSmmValid()

That’s why BIOS Guard created
Responsible Disclosure Fun

✓ Discovery Date: 2017-04-20
✓ Intel PSIRT Notified: 2017-05-22
✓ All the Vendors Notified: 2017-05-26
✓ Disclosure Notification Date: 2017-05-30
✓ Lenovo Released a Patch: 2017-07-11
✓ ASUS Released a Patch: 2017-06-23
✓ MITRE Assign 6 CVE’s: 2017-07-13
✓ Gigabyte Released a Patch: 2017-07-25
✓ Public Disclosure Date: 2017-07-27

https://github.com/CylanceVulnResearch/disclosures
Bravo @ASUS! You silently patch 3 of my SMM issues after a month of detailed disclosure notice. Final reply is brilliant: it's not an issue!

11:39 AM - 7 Jul 2017

Alex Matrosov
@matrosov

Replying to @matrosov @ASUS
It will be a great addition to my #BHUSA talk with details about disclosure process ;)

Replying to @matrosov @ASUS
Finally ASUS agreed they patched my bugs. Good to know but I'm already confirmed this with simple check by BinDiff for patched SMM driver ;)

Tweet your reply
Dear sender,

Thank you for the e-mail. Please don't get us wrong, all of your findings are valuable and we deeply appreciate for the kindness sharing.

We would mention "Fixed UEFI and SMI vulnerability. Special thanks for Cylance" in the update BIOS, or it can be discussed if you have ideas of wording in mind.

Thank you

Best regards,
ASUS Security | (c)ASUSTeK Computer Inc.
Intel Boot Guard
Different shades of Secure Boot

- **Secure Boot -> since 2012**
  - ✓ Root of Trust = Firmware -> BIOS
  - ✓ Attack Surface = Firmware

- **Measured Boot (Boot Guard) -> since 2013**
  - ✓ Root of Trust = Hardware -> Trusted Platform Module (TPM)
  - ✓ Attack Surface = Firmware

- **Verified Boot (Boot Guard) -> since 2013**
  - ✓ Root of Trust = Hardware -> Field Programming Fuse (FPF)-Locked
  - ✓ Attack Surface = Firmware + Hardware
Different shades of Secure Boot

- **Secure Boot** - since 2012
  - ✓ Root of Trust = Firmware -> BIOS
  - ✓ Attack Surface = Firmware

- **Measured Boot (Boot Guard)** - since 2013
  - ✓ Root of Trust = Hardware -> Trusted Platform Module (TPM)
  - ✓ Attack Surface = Firmware

- **Verified Boot (Boot Guard)** - since 2013
  - ✓ Root of Trust = Hardware -> Field Programming Fuse (FPF) -> Locked
  - ✓ Attack Surface = Firmware + Hardware
Why Boot Guard has been created?

- **Secure Boot** starts from DXE phase and impacted with any SMM issues/implants
- No verification on early boot for SEC/PEI boot phases
- **Measured Boot** starts before PEI phase but also impacted with any SMM issues/implants
- The Root of Trust must be locked by hardware (**Verified Boot**)
- The first step of verification should rely on microcode authentication
Intel Boot Guard Technology

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

Locked in BIOS

OS Loader

Locked in Hardware

CPU

Microcode

CPU

Reset

Boot Guard

ACM

Reset

Vector

IBB

(SEC + PEI)

Secure Boot

(DXE + BDS)
Intel Boot Guard operating modes

- Not Enabled
- Measured Boot (root of trust = TPM)
- Verified Boot (root of trust = FPF)
- Measured + Verified Boot (root of trust = FPF + TPM)
Boot Guard: Chain of Trust

Platform Controller Hub (PCH)
Management Engine (ME)
Field Programing Fuse (FPF)
hash of root OEM pub key (SHA-256)

UEFI Firmware Image

Key Manifest (KM)
- key manifest security version number (SVN)
- hash of IBB pub key (SHA-256)
- OEM root pub key (RSA-2048)
- RSA signature on KM SVN + hash of IBBM pub key

Initial Boot Block Manifest (IBBM)
- IBBM security version number (SVN)
- hash of IBB (SHA-256)
- IBBM pub key (RSA-2048)
- RSA signature on IBBM SVN + hash of IBB
Demystifying Intel Boot Guard

FPF → BIOS Update Image

http://en.community.dell.com/techcenter/extras/m/white_papers/20444061
## Guard’s Configuration of Tested Hardware

<table>
<thead>
<tr>
<th>Vendor Name</th>
<th>ME Access</th>
<th>EC Access</th>
<th>CPU Debugging (DCI)</th>
<th>Boot Guard</th>
<th>Forced Boot Guard ACM</th>
<th>Boot Guard FPF</th>
<th>BIOS Guard</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASUS VivoMini</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>MSI Cubi2</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Gigabyte Brix</td>
<td>Read/Write Enabled</td>
<td>Read/Write Enabled</td>
<td>Enabled</td>
<td>Measured Verified</td>
<td>Enabled (FPF not set)</td>
<td>Not Set</td>
<td>Disabled</td>
</tr>
<tr>
<td>Dell</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Enabled</td>
<td>Measured</td>
<td>Verified</td>
<td>Enabled</td>
<td>Enabled</td>
</tr>
<tr>
<td>Lenovo ThinkCentre</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disbled</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>HP EliteDesk</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disbled</td>
<td>Disbled</td>
<td>Disbled</td>
</tr>
<tr>
<td>Intel NUC</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Enabled</td>
<td>Disabled</td>
<td>Disbled</td>
<td>Disabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Apple</td>
<td>Read Enabled</td>
<td>Disabled</td>
<td>Disabled</td>
<td>Not Supported</td>
<td>Not Supported</td>
<td>Not Supported</td>
<td>Not Supported</td>
</tr>
</tbody>
</table>
Safeguarding Rootkits: Intel BootGuard
by Alex Ermolov

ZERONIGHTS 2016

Hash Key Configuration for Bootguard / ISH

Parameter | Value
---|---
OEM Public Key Hash | 00 00 00 00 00 00 00 00 ...

Boot Guard Configuration

Parameter | Value
---|---
Key Manifest ID | 0x0
Boot Guard Profile Configuration | Boot Guard Profile 0 - No_FVME
CPU Debugging | Enabled
BSP Initialization | Enabled
You never attack the standard, you attack the implementation, including the process.
Platform Controller Hub (PCH)
Management Engine (ME)
Field Programming Fuse (FPF)
hash of root OEM pub key (SHA-256)

UEFI Firmware Image

Key Manifest (KM)

- key manifest security version number (SVN)
- hash of IBB pub key (SHA-256)
- OEM root pub key (RSA-2048)
- RSA signature on KM SVN + hash of IBBM pub key

Initial Boot Block Manifest (IBBM)

- IBBM security version number (SVN)
- hash of IBB (SHA-256)
- IBBM pub key (RSA-2048)
- RSA signature on IBBM SVN + hash of IBB

Boot Guard: Chain of Trust
Boot Guard: Key Manifest (KM)

- UBYTE Signature[8]
- UBYTE Unknown
- UBYTE Unknown1
- UBYTE KmSvn
- UBYTE Unknown2
- UBYTE Unknown3
- UINT16 Unknown4[0]
- struct KEY_HASH IbmKeyHash
  - UBYTE Unknown4[1]
  - UINT16 Unknown5
- struct KEY_RSA OemPubKey
  - struct RSA_PUBLIC_KEY Key
    - UBYTE Unknown8
    - UINT16 Size
    - UINT32 Exp
    - UBYTE PubKey[256]
    - UINT16 Unknown16
  - struct RSA_SIGNATURE Signature
    - UINT16 KeySize
    - UINT16 Unknown16
    - UBYTE Signature[256]

- IBBM Hash
- RSA OEM Root Pub Key
- RSA Signature (KM_SVN + hash (IBBM Pub Key))
Boot Guard: Chain of Trust

Platform Controller Hub (PCH)

Management Engine (ME)

Field Programing Fuse (FPF)

hash of root OEM pub key (SHA-256)

UEFI Firmware Image

Key Manifest (KM)

key manifest security version number (SVN)

hash of IBB pub key (SHA-256)

OEM root pub key (RSA-2048)

RSA signature on KM SVN + hash of IBBM pub key

Initial Boot Block Manifest (IBBM)

IBBM security version number (SVN)

hash of IBB (SHA-256)

IBBM pub key (RSA-2048)

RSA signature on IBBM SVN + hash of IBB
Boot Guard: Boot Policy Manifest (BPM)

- struct BOOT_POLICY_MANIFEST BPM
  - struct BOOT_POLICY_MANIFEST_HEADER Hdr
    - UCHAR Signature[8]
    - UCHAR Unknown
    - UCHAR Unknown
    - UCHAR Unknown
    - UCHAR Unknown
    - UCHAR AcmSvn
    - UCHAR Unknown
    - UINT16 Unknown6
  - struct IBB_ELEMENT IIBS
    - UCHAR Signature[8]
    - UCHAR Unknown
    - UCHAR Unknown
    - UCHAR Unknown
    - UCHAR Unknown
    - UINT32 Unknown3
    - UINT64 Unknown4
    - UINT64 VtdBar
    - UINT32 Unknown5
    - UINT32 Unknown6
    - UINT64 Unknown7[2]
    - UINT16 Unknown8
  - struct KEY_HASH IbbHash
  - struct KEY_HASH SigHash
  - UCHAR SegmentNum
  - struct IBB_SEGMENT IbbSegment[4]
    - struct PLATFORM_MANUFACTURER PM
  - struct BOOT_POLICY_MANIFEST_SIGNATURE BPMMS
    - UCHAR Signature[8]
    - UCHAR Version
  - struct RSA_SIGNATURE KeySignature

- IBB Hash
- BB Offsets
- IBBM Pub Key
- RSA Signature ((IBBM_SVN + hash (IBB)))
<table>
<thead>
<tr>
<th>Name</th>
<th>Action</th>
<th>Type</th>
<th>Subtype</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>10C22623-D86F-4721-AA30-4C12AF4230A7</td>
<td></td>
<td>File</td>
<td>PEI module</td>
<td>IdeRecovery</td>
</tr>
<tr>
<td>0002AE8B-F334-4C15-A7F0-E1E897E9F9E1</td>
<td></td>
<td>File</td>
<td>PEI module</td>
<td>NvmeRecovery</td>
</tr>
<tr>
<td>89F06049-F297-4436-8540-E08F9E825B8</td>
<td></td>
<td>File</td>
<td>PEI module</td>
<td>SdioRecovery</td>
</tr>
<tr>
<td>9B3F28D5-A8A6-46C8-BA72-BD40B847A71A</td>
<td></td>
<td>File</td>
<td>PEI module</td>
<td>AmtTcgPlatformPei...</td>
</tr>
<tr>
<td>77D3DC50-D42B-4916-AC00-8F469035D150</td>
<td></td>
<td>Pad</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6520F532-2A27-4195-B331-C0854683E0BA</td>
<td>File</td>
<td>Raw</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8E295B87-D377-4B75-BFDC-9A2F6DBDE22</td>
<td></td>
<td>File</td>
<td>Freeform</td>
<td></td>
</tr>
<tr>
<td>5885965C-45SD-4CC6-9C4C-7F086967D2B0</td>
<td></td>
<td>File</td>
<td>Freeform</td>
<td></td>
</tr>
<tr>
<td>Pad-file</td>
<td></td>
<td>Pad</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C30FF4A-10C6-4C0F-A454-FD319B6F6CE6</td>
<td></td>
<td>Pad</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pad-file</td>
<td></td>
<td>Pad</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7C9A9FB8-2B2B-4027-8F16-F7D277D58025</td>
<td></td>
<td>Pad</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Size</th>
<th>Version</th>
<th>Checksum</th>
<th>Type</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>00000080</td>
<td>00000000</td>
<td>0100h</td>
<td>FIT Header</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0100h</td>
<td>Microcode</td>
<td>LocalOffset 00000000, CPUID 000406E3h, Revision 00000074h, Date 01052016h</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0100h</td>
<td>Microcode</td>
<td>LocalOffset 00017418h, CPUID 000406E3h, Revision 00000028h, Date 04152015h</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0100h</td>
<td>Microcode</td>
<td>LocalOffset 0002C418h, CPUID 000506E3h, Revision 00000074h, Date 01052016h</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0100h</td>
<td>Microcode</td>
<td>LocalOffset 00043818h, CPUID 000506E2h, Revision 0000002Ch, Date 07082015h</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0100h</td>
<td>BIOS ACM</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0100h</td>
<td>BootGuard Key Manifest</td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>0100h</td>
<td>BootGuard Boot Policy</td>
<td></td>
</tr>
</tbody>
</table>
// FIT Entry type definitions

#define FIT_TYPE_00_HEADER
#define FIT_TYPE_01_MICROCODE
#define FIT_TYPE_02_STARTUP_ACM
#define FIT_TYPE_07_BIOS_STARTUP_MODULE
#define FIT_TYPE_08_TPM_POLICY
#define FIT_TYPE_09_BIOS_POLICY
#define FIT_TYPE_0A_TXT_POLICY
#define FIT_TYPE_0B_KEY_MANIFEST
#define FIT_TYPE_0C_BOOT_POLICY_MANIFEST
#define FIT_TYPE_10_CSE_SECURE_BOOT
#define FIT_TYPE_2D_TXTSX_POLICY
#define FIT_TYPE_2F_JMP_DEBUG_POLICY
#define FIT_TYPE_7F_SKIP

0x00
0x01
0x02
0x07
0x08
0x09
0x0A
0x0B
0x0C
0x10
0x2D
0x2F
0x7F

BootGuard Boot Policy
Boot Guard: Initial Boot Block (IBB)
Boot Guard: Initial Boot Block (IBB)
# Boot Guard: Authenticated Code Module (ACM)

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModuleType</td>
<td>30002h</td>
</tr>
<tr>
<td>HeaderType</td>
<td>A1h</td>
</tr>
<tr>
<td>ModuleVendor</td>
<td>8086h</td>
</tr>
<tr>
<td>Date</td>
<td>20150624h</td>
</tr>
<tr>
<td>ModuleSize</td>
<td>2000h</td>
</tr>
<tr>
<td>AcmSvn</td>
<td>2h</td>
</tr>
<tr>
<td>Unknown1</td>
<td>1h</td>
</tr>
<tr>
<td>EntryPoint</td>
<td>3BB1h</td>
</tr>
<tr>
<td>KeySize</td>
<td>40h</td>
</tr>
<tr>
<td>Unknown4</td>
<td>8Fh</td>
</tr>
<tr>
<td>RsaPubKey</td>
<td>256</td>
</tr>
<tr>
<td>RsaPubExp</td>
<td>11h</td>
</tr>
<tr>
<td>RsaSig</td>
<td>256</td>
</tr>
</tbody>
</table>

- **ACM Header**
  - **EntryPoint**: 3BB1h
  - **RSA Pub Key**: [Binary data]
  - **RSA Exp**: [Binary data]
  - **RSA Signature**: [Binary data]
Boot Guard: Authenticated Code Module (ACM)

- ACM is x86 (32-bit) code developed by Intel
- ACM executes in AC-RAM (Cache-as-RAM or NEM)
- ACM has CPU and Chipset specifics
- ACM verifies Key Manifest (KEYM) + IBB (IBBM)
ACM is x86 (32-bit) code developed by Intel.

ACM executes in AC-RAM (Cache-RAM or NEM).

ACM has CPU and chipset specifics.

ACM verifies Key Manifest (KEYM) + IBB (IBBM).

Boot Guard: Authenticated Code Module (ACM)
Boot Guard: Authenticated Code Module (ACM)
Boot Guard: Authenticated Code Module (ACM)
Boot Guard ACM BinDiff: Haswell vs Skylake
### Boot Guard ACM BinDiff: Broadwell vs Skylake

<table>
<thead>
<tr>
<th>Similarity</th>
<th>Conf</th>
<th>Chan</th>
<th>EA Primary</th>
<th>Name Primary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>00000045</td>
<td>sub_45</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>000001CE</td>
<td>sub_1CE</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>00001E7</td>
<td>sub_1E7</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>0000200</td>
<td>sub_200</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>0000229</td>
<td>sub_229</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>0000F88</td>
<td>sub_F88</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>0001603</td>
<td>sub_1603</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>000165A</td>
<td>sub_165A</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>0001863</td>
<td>sub_1863</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>00018D9</td>
<td>sub_18D9</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>0001FC2</td>
<td>sub_1FC2</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>00023A2</td>
<td>sub_23A2</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>0002C36</td>
<td>sub_2C36</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>0002CB2</td>
<td>sub_2CB2</td>
</tr>
<tr>
<td>1.00</td>
<td>0.98</td>
<td>------</td>
<td>000405D</td>
<td>sub_405D</td>
</tr>
<tr>
<td>1.00</td>
<td>0.96</td>
<td>------</td>
<td>0000A04</td>
<td>sub_A04</td>
</tr>
<tr>
<td>1.00</td>
<td>0.96</td>
<td>------</td>
<td>0000A39</td>
<td>sub_A39</td>
</tr>
<tr>
<td>1.00</td>
<td>0.96</td>
<td>------</td>
<td>0000FF3</td>
<td>sub_FF3</td>
</tr>
<tr>
<td>1.00</td>
<td>0.96</td>
<td>------</td>
<td>000255S</td>
<td>sub_255S</td>
</tr>
<tr>
<td>1.00</td>
<td>0.96</td>
<td>------</td>
<td>0004019</td>
<td>sub_4019</td>
</tr>
<tr>
<td>1.00</td>
<td>0.96</td>
<td>------</td>
<td>000409A</td>
<td>sub_409A</td>
</tr>
<tr>
<td>1.00</td>
<td>0.90</td>
<td>------</td>
<td>000410A</td>
<td>sub_410A</td>
</tr>
<tr>
<td>0.99</td>
<td>0.99</td>
<td>------</td>
<td>00002FC</td>
<td>sub_2FC</td>
</tr>
<tr>
<td>0.99</td>
<td>0.99</td>
<td>------</td>
<td>00004892</td>
<td>sub_4892</td>
</tr>
<tr>
<td>0.99</td>
<td>0.99</td>
<td>------</td>
<td>00002708</td>
<td>sub_2708</td>
</tr>
<tr>
<td>0.98</td>
<td>0.99</td>
<td>------</td>
<td>00041C8</td>
<td>sub_41C8</td>
</tr>
<tr>
<td>0.96</td>
<td>0.99</td>
<td>------</td>
<td>0002F60</td>
<td>sub_2F60</td>
</tr>
<tr>
<td>0.96</td>
<td>0.99</td>
<td>------</td>
<td>00002486</td>
<td>sub_2486</td>
</tr>
<tr>
<td>0.96</td>
<td>0.99</td>
<td>------</td>
<td>00028F7</td>
<td>sub_28F7</td>
</tr>
<tr>
<td>0.93</td>
<td>0.99</td>
<td>------</td>
<td>0002DFA</td>
<td>sub_2DFA</td>
</tr>
<tr>
<td>0.93</td>
<td>0.94</td>
<td>------</td>
<td>00033A0</td>
<td>sub_33A0</td>
</tr>
<tr>
<td>0.91</td>
<td>0.99</td>
<td>------</td>
<td>00031E2</td>
<td>sub_31E2</td>
</tr>
<tr>
<td>0.91</td>
<td>0.99</td>
<td>------</td>
<td>0004112</td>
<td>sub_4112</td>
</tr>
<tr>
<td>0.87</td>
<td>0.95</td>
<td>------</td>
<td>0003DCE</td>
<td>sub_3DCE</td>
</tr>
<tr>
<td>0.62</td>
<td>0.75</td>
<td>------</td>
<td>0003D08</td>
<td>sub_3D08</td>
</tr>
<tr>
<td>0.50</td>
<td>0.73</td>
<td>------</td>
<td>000458B</td>
<td>sub_458B</td>
</tr>
<tr>
<td>0.45</td>
<td>0.62</td>
<td>------</td>
<td>0002C7A</td>
<td>sub_2C7A</td>
</tr>
<tr>
<td>0.42</td>
<td>0.57</td>
<td>------</td>
<td>0004411</td>
<td>sub_4411</td>
</tr>
<tr>
<td>0.40</td>
<td>0.50</td>
<td>------</td>
<td>000453C</td>
<td>sub_453C</td>
</tr>
<tr>
<td>0.33</td>
<td>0.47</td>
<td>------</td>
<td>0004699</td>
<td>sub_4699</td>
</tr>
<tr>
<td>0.28</td>
<td>0.41</td>
<td>------</td>
<td>0002024</td>
<td>sub_2024</td>
</tr>
<tr>
<td>0.23</td>
<td>0.29</td>
<td>------</td>
<td>00040922</td>
<td>sub_40922</td>
</tr>
<tr>
<td>0.21</td>
<td>0.35</td>
<td>------</td>
<td>0004984D</td>
<td>sub_4984D</td>
</tr>
<tr>
<td>0.18</td>
<td>0.33</td>
<td>------</td>
<td>0003D66</td>
<td>sub_3D66</td>
</tr>
<tr>
<td>0.12</td>
<td>0.24</td>
<td>------</td>
<td>0003CD3</td>
<td>sub_3CD3</td>
</tr>
<tr>
<td>0.09</td>
<td>0.24</td>
<td>------</td>
<td>000484D</td>
<td>sub_484D</td>
</tr>
<tr>
<td>0.05</td>
<td>0.09</td>
<td>------</td>
<td>00028C2</td>
<td>sub_28C2</td>
</tr>
</tbody>
</table>
Boot Guard ACM BinDiff: Broadwell vs Skylake
Boot Guard BIOS Components (AMI)

- **PEI**
  - **BootGuardP ei** [B41956E1-7CA2-42db-9562-168389F0F066]

- **SMM**
  - **VerifyFwBootGuard** [EE89F590-A816-4ac5-B3A9-1BC759B12439]

- **DXE**
  - **BootGuardDxe** [1DB43EC9-DF5F-4cf5-AAF0-0E85DB4E149A]
BootGuardPei Validation Flow

```c
EFI_STATUS BootGuardPei(EFI_PEI_SERVICES **PepServices, VOID *Ppi)
{
    ...

    Status = GetBootMode();
    if (EFI_ERROR( Status ) ) {
        return Status;
    }

    ...

    if ( (BootMode == BOOT_IN_RECOVERY_MODE) || (BootMode == BOOT_ON_FLASH_UPDATE) || BootMode == BOOT_ON_S3_RESUME ) {
        return Status;
    }

    BootGuardVerifyTransitionPEItoDXEFlag = 0;

    ...

    CalculateSha256(BootGuardHashKeySegment0);
    CalculateSha256(CurrentBootGuardHashKey0);

    if ( !MemCmp(BootGuardHashKeySegment0, CurrentBootGuardHashKey0, 32) ) {
        BootGuardVerifyTransitionPEItoDXEFlag = 1;
    } else {
        BootGuardVerifyTransitionPEItoDXEFlag = 0;
        return EFI_SUCCESS;
    }

    if ( !(BootGuardHashKeySegment1 == 0) ) {
        CalculateSha256 (BootGuardHashKeySegment1);
        CalculateSha256 (CurrentBootGuardHashKey1);

        if ( !MemCmp(BootGuardHashKeySegment1, CurrentBootGuardHashKey1, 32) ) {
            BootGuardVerifyTransitionPEItoDXEFlag = 1;
        } else {
            BootGuardVerifyTransitionPEItoDXEFlag = 0;
            return EFI_SUCCESS;
        }
    }

    return Status;
}
```
Boot Guard:
PEI FV_HASH

➢ FV_HASH_KEY [CBC91F44-A4BC-4A5B-8696-703451D0B053]
Boot Guard:
PEI FV_HASH

FV_HASH_KEY

CBC91F44-A4BC-4A5B-8696-703451D0B053
VerifyFwBootGuard SMM Validation Flow
(Intel ME communications over HECI)

- Find and Verify ACM
  - Verify ACM SVN

- Find and Verify Key Manifest (KM)
  - Verify KM SVN

- Find and Verify Boot Policy Manifest (BPM)
  - Verify BPM SVN

- If something wrong return EFI_SECURITY_VIOLATION
BootGuardDxe Validation Flow

```c
EFI_STATUS BootGuardDxe(EFI_HANDLE ImageHandle, EFI_SYSTEM_TABLE *SystemTable)
{
    ...

    if ( BootGuardSupported() == FALSE ) {
        return EFI_SUCCESS;
    }

    ...

    BootMode = GetBootMode();
    if ( (BootMode == BOOT_IN_RECOVERY_MODE) || (BootMode == BOOT_ON_FLASH_UPDATE) ) {
        return EFI_SUCCESS;
    }

    ...

    }

    return EFI_SUCCESS;
```
EFI_STATUS BootGuardDxe(EFI_HANDLE ImageHandle, EFI_SYSTEM_TABLE *SystemTable)
{
    ...
    if ( BootGuardSupported() == FALSE ) {
        return EFI_SUCCESS;
    }
    ...
    BootMode = GetBootMode();
    if ( (BootMode == BOOT_IN_RECOVERY_MODE) || (BootMode == BOOT_ON_FLASH_UPDATE) ) {
        return EFI_SUCCESS;
    }
    ...
    if ( BootGuardVerifyTransitionPEItoDEXFlag == 0 ) {
        BootGuardRegisterCallBack();
    }
    return EFI_SUCCESS;
}
Target Platform

➢ Gigabyte (GB-BSi7HA-6500)
   ✓ Intel 6th generation Core i7 CPU (Skylake) with vPro
   ✓ Intel Boot Guard – ENABLED
   ✓ Intel BIOS Guard – NOT ENABLED

➢ Vulnerabilities
   ✓ Host Write/Read Access to ME (CVE-2017-11314)
   ✓ Intel Boot Guard Configuration not Locked (CVE-2017-11313)
Target Platform

GB - BSi7HA - 6500

✓ Intel 6th generation Core i7 CPU (Skylake) with vPro

✓ Intel Boot Guard – ENABLED

✓ Intel BIOS Guard – NOT ENABLED

Vulnerabilities

✓ Host Write/Read Access to ME (CVE-2017-11314)

✓ Intel Boot Guard Configuration not Locked (CVE-2017-11313)
### Target Platform

- **Gigabyte (GB-BSi7HA-6500)**
  - **Intel 6th generation Core i7 CPU** (Skylake) with **vPro**
  - **Intel Boot Guard** – **ENABLED**
  - **Intel BIOS Guard** – **NOT ENABLED**

### Vulnerabilities

- **Host Write/Read Access to ME** (**CVE-2017-11314**)
- **Intel Boot Guard Configuration not Locked** (**CVE-2017-11313**)

---

**Boot Guard Configuration**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Manifest ID</td>
<td>0x0001</td>
</tr>
<tr>
<td>Boot Guard Profile Configuration</td>
<td>Boot Guard Profile 5 - FVME</td>
</tr>
<tr>
<td>CPU Debugging</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

---

**OEM Public Key Hash**

EE 7D B6 9F 8B 18 F5 41 F6 46 ...
Vertical Markets
- School
- University computer labs
- Libraries
  - Hospital / Medical equipment
  - Governmental

Powerful Commercial Applications
- Factory testing machine
- Bank ATM system
- Gaming equipment
- Vending machine
- Security system
Five steps to bypass Boot Guard

1) Modify UEFI firmware update image with rootkit/implant or Disable Intel Boot Guard

2) Initial Boot Block (IBB)
   ✓ Recalculate signature on 2048-bit RSA key pair for IBB
   ✓ Modify IBB manifest inside UEFI firmware update file
   ✓ Recalculate signature for IBB manifest with different 2048-bit RSA key pair

3) Modify Root Key manifest
   ✓ Recalculate SHA256 hash of the public key from Root Key Manifest

4) Modify ME region with new key (CVE-2017-11314)
   ✓ Modify Boot Guard configuration with active verified boot policy

5) Lock Boot Guard configuration with by FPF (CVE-2017-11313)
Boot Guard: Chain of Trust

**Platform Controller Hub (PCH)**

**Management Engine (ME)**

**Field Programing Fuse (FPF)**

hash of root OEM pub key (SHA-256)

**UEFI Firmware Image**

**Key Manifest (KM)**

key manifest security version number (SVN)

hash of IBB pub key (SHA-256)

OEM root pub key (RSA-2048)

RSA signature on KM SVN + hash of IBBM pub key

**Initial Boot Block Manifest (IBBM)**

IBBM security version number (SVN)

hash of IBB (SHA-256)

IBBM pub key (RSA-2048)

RSA signature on IBBM SVN + hash of IBB
“Intel provides a 6th and 7th generation Core Platforms Secure Configuration Specification, which covers how to securely configure the platform. Additionally, Intel makes available a utility that our ecosystem partners can use to test and identify potential configuration issues.”
“For FPF issue, we discuss with internal the BIOS don’t need any update but we will add ME Lock tool to our production process soon, the new production ship will include ME Lock.”
Intel BIOS Guard
Intel BIOS Guard

- Armoring SPI Flash access
  - Access controlled by BIOS Guard ACM
  - Attack Surface = Firmware

- BIOS update authentication
  - Root of Trust = Hardware -> Trusted Platform Module (TPM)
  - Attack Surface = Firmware

- Verified Boot -> since 2013
  - Root of Trust = Hardware -> Field Programming Fuse (FPF) -> Locked
  - Attack Surface = Firmware + Hardware
Demystifying Intel BIOS Guard

Boot Guard BIOS Components (AMI)

➢ PEI
   ➢ BiosGuardPeiApRecoveryCapsule [C776AEA2-AA27-446e-975B-E0BEA9078BD9]
   ➢ BiosGuardRecovery [95C894B4-DAEC-46E1-8600-3C4C7FC985D6]
   ➢ BiosGuardCpuPolicyOverride [FAF79E9F-4D40-4F02-8AC9-4B5512708F7F]

➢ SMM
   ➢ BiosGuardSmm [44FE07D3-C312-4ad4-B892-269AB069C8E1]
   ➢ BiosGuardServices [6D4BAA0B-F431-4370-AF19-99D6209239F6]

➢ DXE
   ➢ BiosGuardDxe [6D1D13B3-8874-4e92-AED5-22FC7C4F7391]
   ➢ BiosGuardNvs [17565311-4B71-4340-88AA-DC9F4422E53A]
Boot Guard BIOS Components (AMI)

- **PEI**
  - BiosGuardPeiApRecoveryCapsule - AMI Capsule Update Validation
  - BiosGuardRecovery - Recovery Update Image parser
  - BiosGuardCpuPolicyOverride
    - Find Public Key
    - Find and Load BIOS Guard ACM

- **SMM**
  - BiosGuardSmm - Recovery SMI Handlers

- **DXE**
  - BiosGuardDxe - Recovery helper for update process
    - UEFI variable cleanup
  - BiosGuardNvs - ACPI helper for update process
    - AMI Capsule validation
BIOS Guard Commands (AMI)

- **PEI**
  - BG_READ
  - BG_WRITE
  - BG_ERASE
  - BG_WRITE_ENABLE
  - BG_WRITE_DISABLE

- **SMM**
  - BG_READ
  - BG_WRITE
  - BG_ERASE
All the stuff will be released on public
save the link:

https://github.com/REhints/BlackHat_2015
Thank you for your attention!

Alex Matrosov
@matrosov