UDK2018 SECURITY FEATURE ROUNDUP

Maggie Jauregui, Erik Bjorge and Brian Richardson
Platform Armoring & Resiliency team
OSFC – September 2018

Who are we?

Platform Armoring & Resiliency

- Part of the firmware team in Intel's Software and Services Group (SSG)
- Includes researching new issues, leading the response to discovery of issues, and finding ways to enhance our capability in the future
- Focused on Resiliency (Protect, Detect, and Recovery) for Intel platforms
- Support for CHIPSEC open source project

Why Attack Firmware?

Persistent Compromise

- Update firmware image with malicious content

Stealthy Compromise

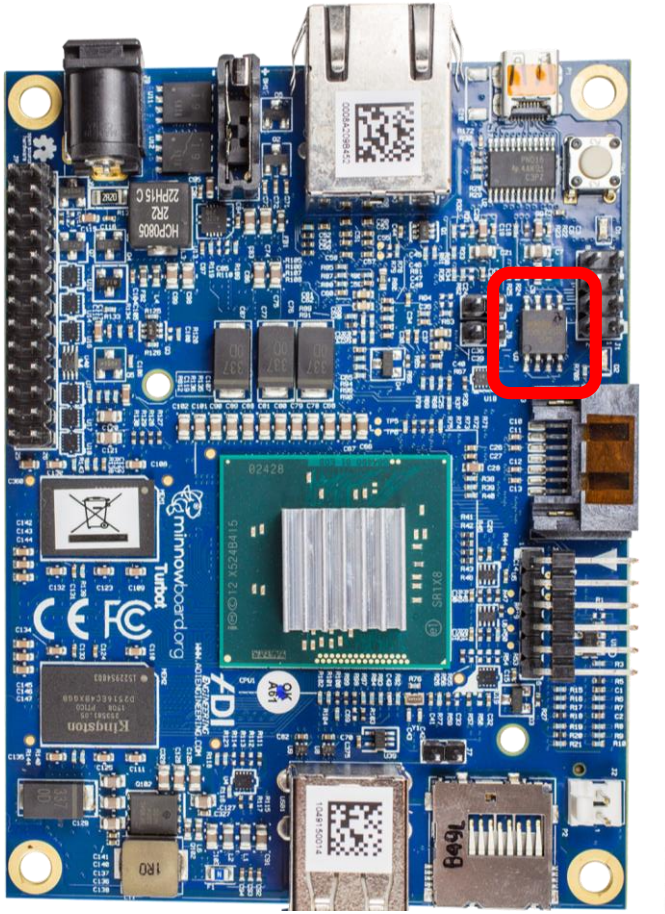
- System Management Mode (SMM) code injection

Bypass of Security Features

- Virtual Machine Manager (VMM) Bypass

Denial of Service

- Corrupt/Delete critical configuration settings



BUILDING A THREAT MODEL…

Note: Contents are meant as examples. It does not represent an exhaustive analysis.

Attacks and Platform Assets

Persistent Compromise

– Update firmware image with malicious content

Stealthy Compromise

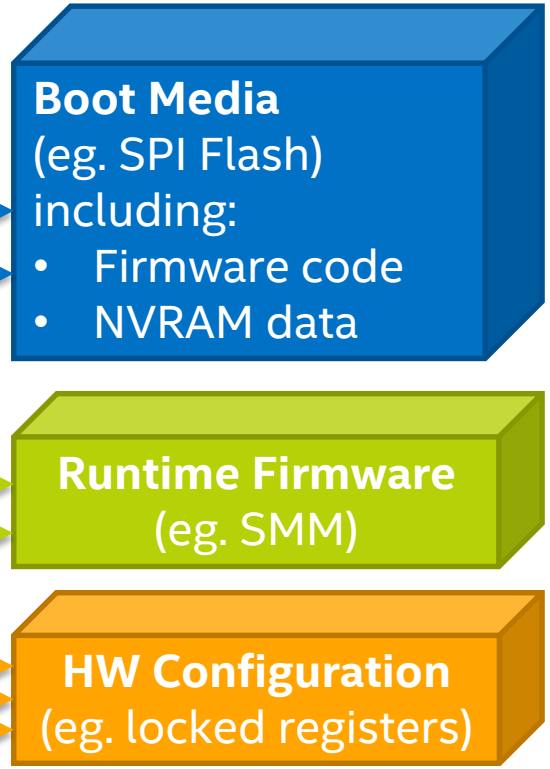
– System Management Moe (SMM) code injection

Bypass of Security Features

– Virtual Machine Manager (VMM) Bypass

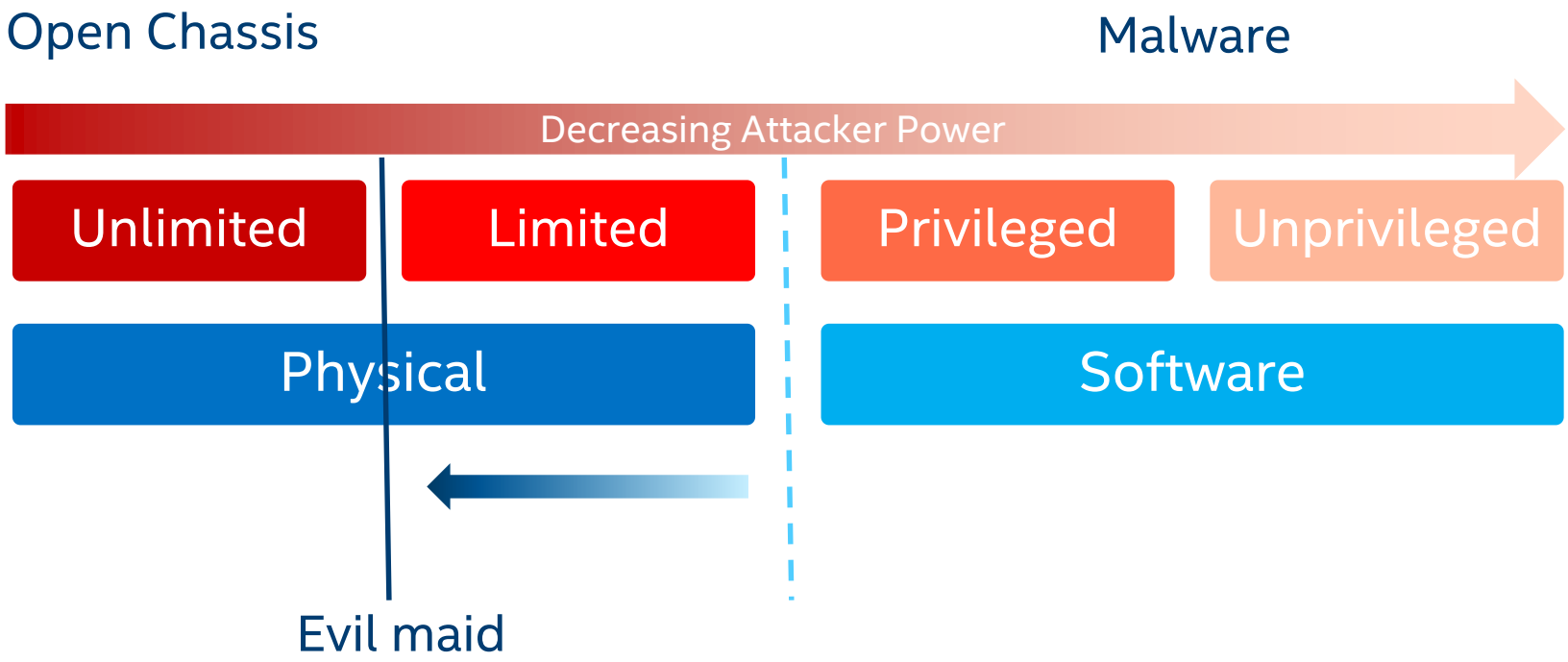
Denial of Service

– Corrupt/Delete critical configuration settings



These are examples. Not an exhaustive list.

Classes of Attacker



Hardware Interfaces as Attack Vectors

Privileged software?

Processor

- General Purpose Registers (RAX, RBX…), Control Registers (CRx), Debug Registers (DRx), …
- CPU Model Specific Registers (MSR)

Processor/Chipset

- I/O Space (ports and BARs)
- PCIe device configuration space
- Memory mapped PCIe configuration access a.k.a Enhanced Configuration Access Mechanism (ECAM)
- Memory mapped I/O ranges
- Intel On-chip System Fabric (IOSF) Message Bus registers

Firmware Interfaces as Attack Vectors

Unprivileged software?

Firmware Code & Data

- Platform Initialization Code
  - Initial Boot Block (IBB), OEM Boot Block (OBB), other firmware
- Environment Setup Code
  - Driver Execution Environment (DXE) drivers, boot loader, etc
- Non Volatile Random Access Memory (NVRAM) Configuration Data

Runtime Code & Data

- Runtime Services
- System Management Mode
  - Software System Management Interrupt (SMI), System Management Random Access Memory (SMRAM)

Resilient Defense

Boot Media

Runtime Firmware (eg. SMM)

Hardware Configuration

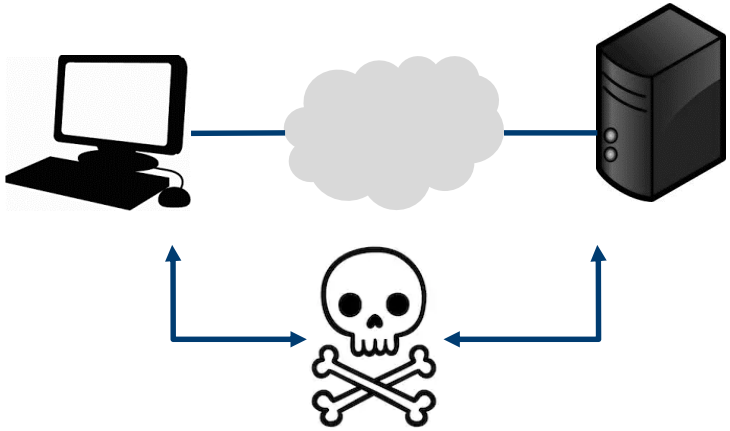
Decreasing Attacker Power

Unlimited | Limited | Privileged | Unprivileged

Physical | Software

Remote Firmware attacks

- Remotely attacking System Firmware, Black Hat USA 2018 https://www.blackhat.com/us-18/briefings/schedule/#remotely-attacking-system-firmware-11588
- UEFI Exploitation for the masses, DEFCON 26 https://www.defcon.org/html/defcon-26/dc-26-speakers.html#Shkatov
- Remote UEFI attacks, Eclypsium https://blog.eclypsium.com/2018/08/27/uefi-remote-attacks/



- Potential attacks -

Spoofing

Man-in-the-Middle (MitM)

Denial-of-Service (DoS)

UEFI HTTP over TLS (HTTPS) boot

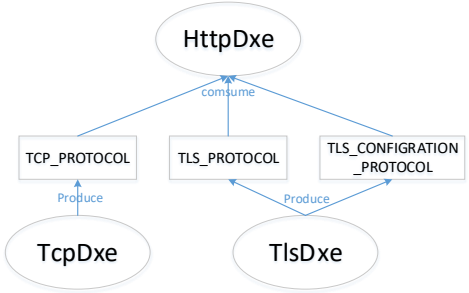
HTTPS Boot Authentication & Verification

Feature usage: Load the specified file from the remote HTTPS server successfully and steadily.

UEFI Arch: IA32 and X64 platform.

TLS version: TLS1.0/1.1/1.2, version negotiation

Protect the certificate variable from malicious modification using an authenticated variable.



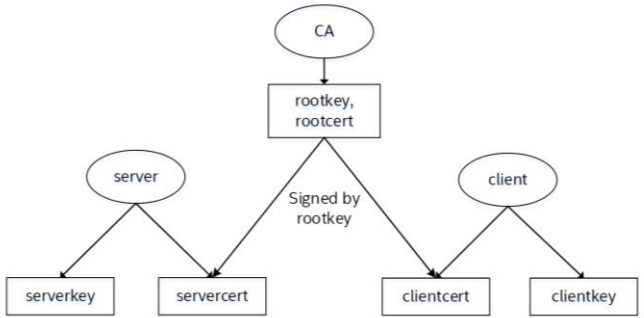
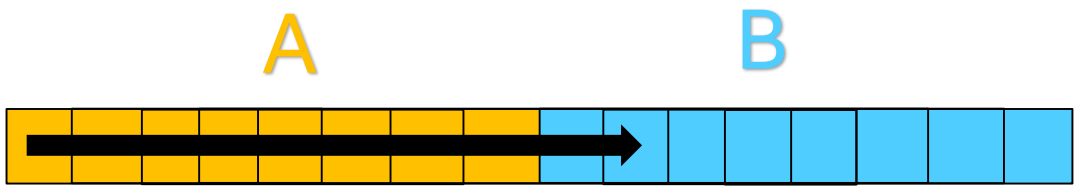


Figure 1: Authentication Mechanism

References:

- HTTPS boot http://www.uefi.org/sites/default/files/resources/UEFI%20Spec%202_6.pdf
- Implementation flow https://github.com/tianocore-docs/Docs/raw/master/White_Papers/EDKIIHttps_TLS_BootGettingStartedGuide_07.pdf

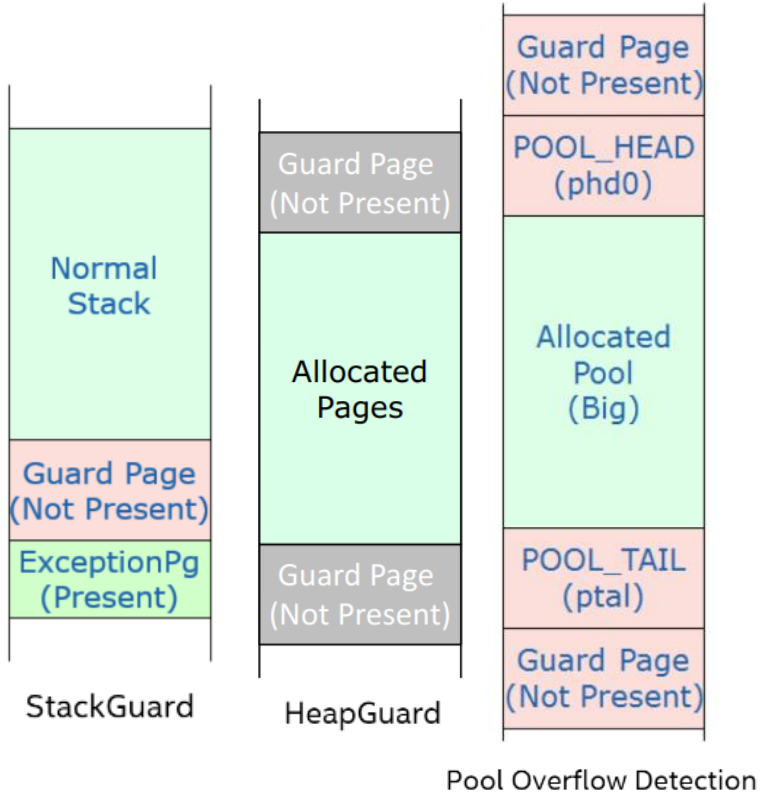
Buffer overflows



- OWASP Top 10
- Stack-based exploitation
- Heap-based exploitation
- Related research
  - **Philips Delays Fix for Cardiograph Cybersecurity Vulnerabilities** https://healthitsecurity.com/news/philips-delays-fix-for-cardiograph-cybersecurity-vulnerabilities
  - **Hacking smart plugs to enter business networks** https://www.helpnetsecurity.com/2018/08/23/hacking-smart-plugs/
  - **Buffer overflow in Unix mailer Exim imperils 400,000 email servers** https://www.theregister.co.uk/2018/03/07/exim_mail_server_bug/
  - **Firefox fixes critical buffer overflow** https://nakedsecurity.sophos.com/2018/06/18/firefox-fixes-critical-buffer-overflow/

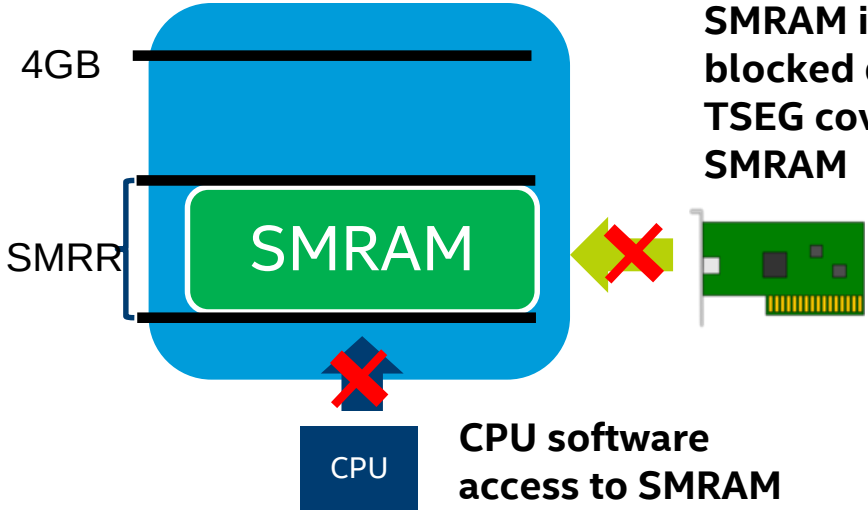
Guard Page

- Catch page overflows when they happen
- Catch pool overflows when they happen
- Guard page set as not present in page table. Upon overflow, a page fault exception is triggered immediately.
- Limitation
  - Memory size overhead
    - Additional 8K for each page allocation.
    - Additional 8K+4K alignment for each pool allocation.



Pre-boot Direct Memory Access (DMA) attacks

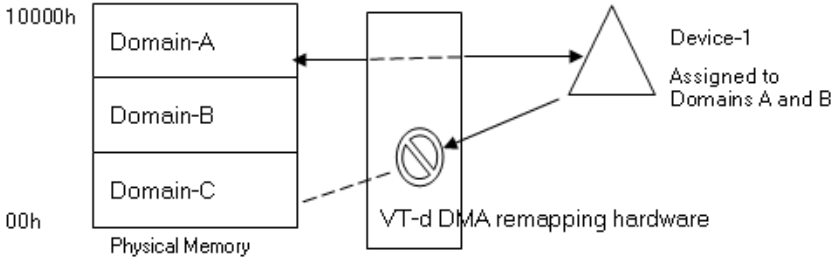
```
DmaBackdoorSimple.c(220) : ****************************
DmaBackdoorSimple.c(221) :
DmaBackdoorSimple.c(222) :   UEFI backdoor loaded
DmaBackdoorSimple.c(223) :
DmaBackdoorSimple.c(224) : ****************************
DmaBackdoorSimple.c(227) : Image address is 0x10000
DmaBackdoorSimple.c(241) : Resident code base address is 0xd6119000
DmaBackdoorSimple.c(148) : BackdoorEntryResident()
-
```



- Dmytro Oleksiuk (Cr4sh) pre-boot DMA backdoors
  https://twitter.com/d_olex/status/916964178035798017
- Ulf Frisk, PCILeech (Attacking UEFI)
  http://blog.frizk.net/2017/08/attacking-uefi.html

Pre-boot DMA Protection

VT-d enables hardware support for isolating and restricting device accesses to the owner of the partition managing the device



WHITE PAPER
Firmware Security
DMA Protection in UEFI

A Tour Beyond BIOS:
Using IOMMU for DMA Protection in UEFI Firmware

This paper presents the idea of using an input–output memory management unit (IOMMU) to resist Direct Memory Access (DMA) attacks in firmware. The example presented uses Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d), and the concept can be applied to other IOMMU engines.

https://firmware.intel.com/sites/default/files/Intel_WhitePaper_Using_IOMMU_for_DMA_Protection_in_UEFI.pdf

VT-d: Virtualization Technology for Directed I/O

Step 0: Install IOMMU Protocol

Step 1: Parse DMAR ACPI Table

Step 2: Setup DMAR Translation Table

Step 3: Get Platform VTD Policy

Step 4: Enable DMA Remapping

Step 5: Grant/Revoke DMA access rights in UEFI Firmware

Step 6: Update DMA Remapping Status when Transferring Control to OS

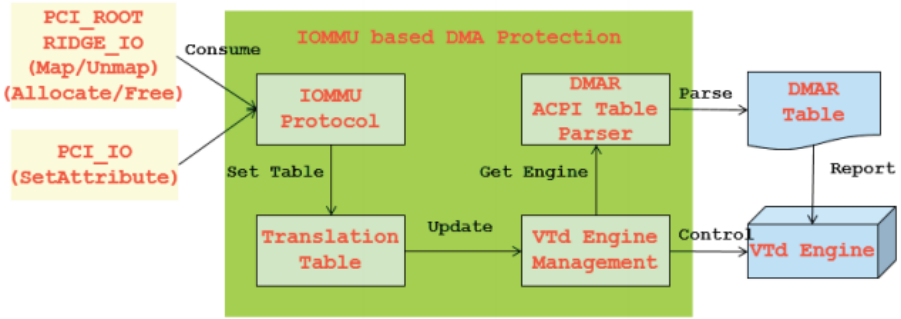
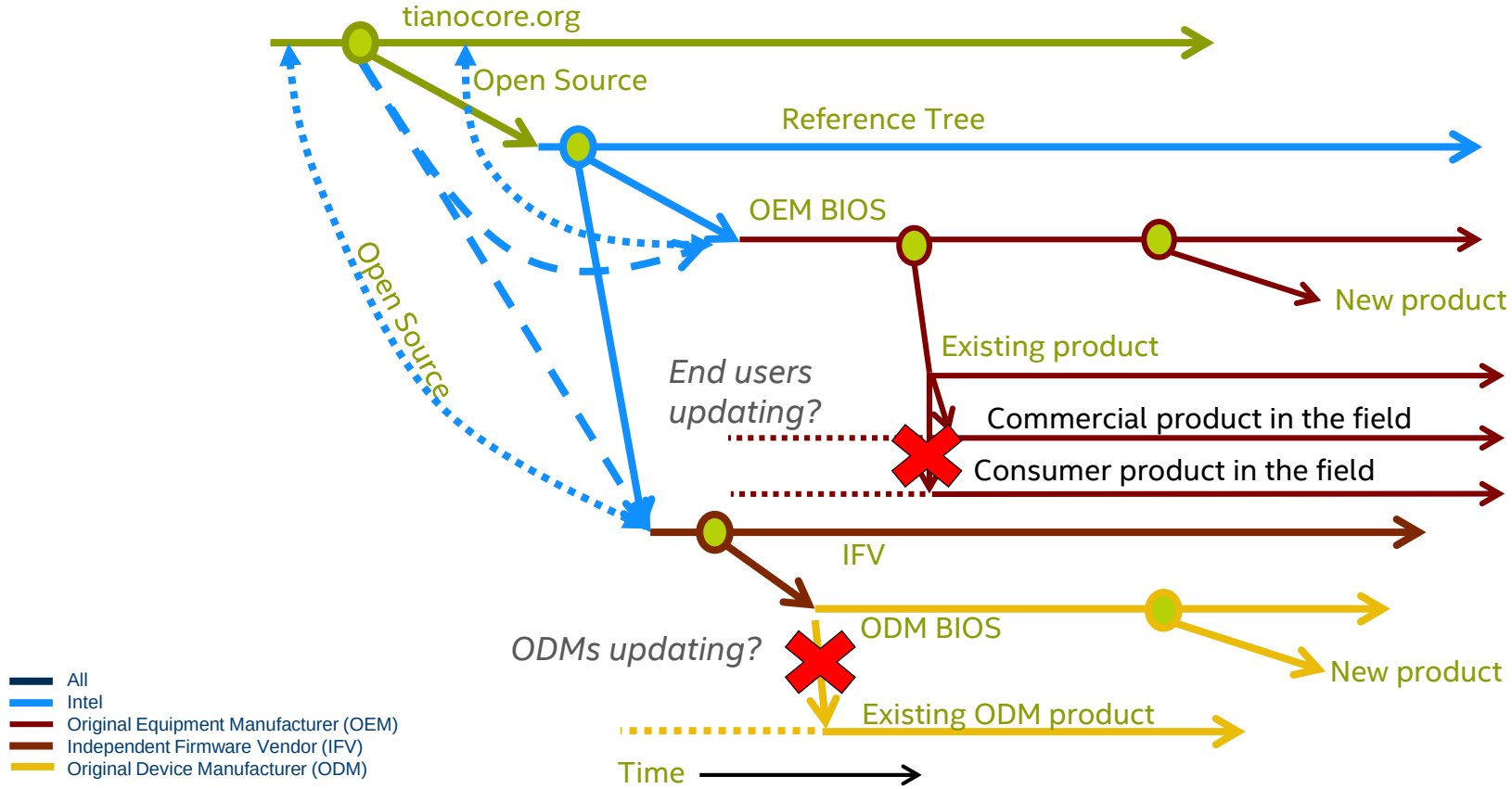


Figure 7 - IOMMU-based DMA Protection Component

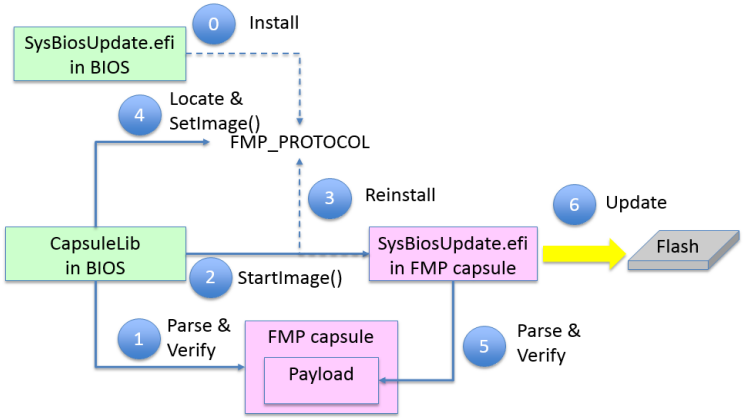
The road from core to platform



Source: http://vzimmer.blogspot.com/2015_05_01_archive.html

EDK II Signed Capsule Update

- EDK II signed capsule update solution to meet NIST guidelines and provide a BIOS authentication check
- OEM choice regarding specific topology of capsule payload
- Signing: RSA 2048
- Digesting: SHA 256
- Anti-Rollback Protection (Security Version Number)
- Capsule sent from OS via UEFI runtime service



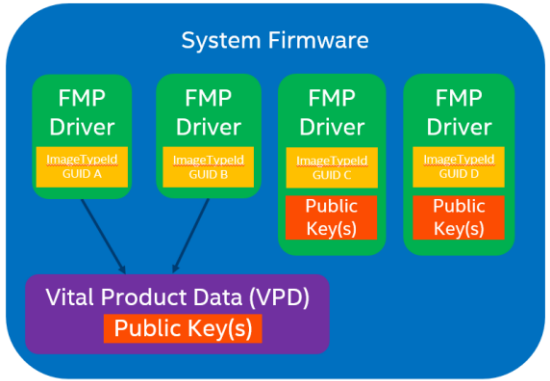
References:

- UEFI Capsule API definition: http://www.uefi.org/sites/default/files/resources/UEFI%20Spec%202_6.pdf
- NIST Guidelines: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-147.pdf, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-147B.pdf

What's new w/FMP & Harmonized Capsule?

- Provide a simple method for platform firmware to produce one or more Firmware Management Protocol (FMP) instances to update firmware images in firmware storage devices from UEFI capsules
  - Platform customizations through libraries and PCDs
  - Firmware storage device customizations through libraries and PCDs
  - Support multiple PKCS7 certificates for authentication (e.g. Development and Production)
- Improve the user experience when firmware updates are being processed
- Provide standalone tools to generate UEFI capsules that contain firmware update images
- Provide standard alone tools to convert a UEFI capsule to a Windows Update driver

Use Case – Multiple FMPs, shared and non-shared keys



ESRT Table
GUID A
GUID B
GUID C
GUID D

* EFI System Resource Table (ESRT)

Join the Capsule Update Hack-a-Thon @ OSFC! (Sep 14-15)

Intel is hosting the first TianoCore hack-a-thon event open to the wider public. (open to OSFC attendees only)

Vulnerabilities found are eligible for Intel Bug Bounty submission

- Bug Bounty guidelines: https://www.intel.com/content/www/us/en/security-center/bug-bounty-program.html
- Participant agreement: https://github.com/tianocore/tianocore.github.io/blob/master/files/TianoCoreHackathonAgreementOSFC.pdf
- EDK II Capsule update Hack-a-Thon (more information) https://github.com/tianocore/tianocore.github.io/wiki/2018-EDK-II-Capsule-Hack-a-thon

Thanks to OSFC organizers for providing the venue.

Legal Notice

No computer system can be absolutely secure.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

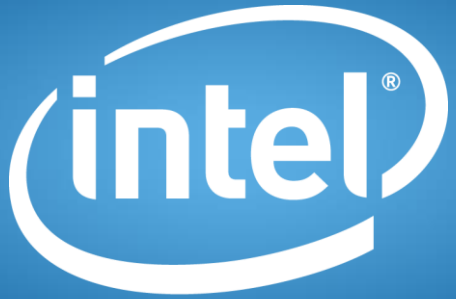
Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel, the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others



Software