LinuxCon Europe

UEFI Mini-Summit

7 October 2015

Session 5 – “UEFI Development in an Open Source Ecosystem”

Michael Krau. Intel

Vincent Zimmer, Intel

Abstract data

- **UEFI Development in an Open Source Ecosystem**
  - UEFI technology Open source development progression session will address:
    - The status of today's prevalent UEFI 'chain of trust' security tools, e.g. Secure Boot and Intel® Platform Trust Technology (PTT).
    - Improved community hosting, communications and source control methodologies, creating valuable opportunities to integrate firmware functions into distros.
  - **Attendees will have the opportunity to share and recommendations for future open UEFI development resources and processes.**

History

- It’s easier to study history…
  - 1998
    - Intel Boot Initiative underway
    - PCI 2.2 specification
    - 92.9 Million PCs sold
    - USB 1.1
    - 802.11
    - IA-64 in development
- Industrial Ecosystem
  - Conservative mindset
  - Evolutionary not revolutionary (in the classic sense of ‘evolution’)

Road to Open Source

- UEFI formed in 2005
  - UEFI is a specification organization
  - Implementation specifics are open to membership
- UEFI Open Community Website (URL: Tianocore.org) late 2005
  - Reference implementations
  - Originally defined to provide the community with working UEFI implementations for evaluation experimentation, and development
    - NT32
    - EDK
    - EDK II (UDK2010, UDK 2014)
    - Duet
    - OVMF (circa 2010)
  - Becomes de facto UEFI standard implementation
- Platform implementations:
  - ARM implementation (BeagleBoard: 2011)
  - Intel implementation (MinnowBoard: 2013)
  - Intel implementation (MinnowBoard MAX: 2014)

Standards

- Technologies utilize firmware resource, only to become obsolete or problematic:
  - Int 15 extensions (Joystick, OEM specifics)
  - Video standards: CGA, VGA, EGA, XGA, Vesa
  - Media/Drives: Audio Cassette, MFM, ESDI, SCSI, IDE, Floppy, SATA
- Firmware standard that allowed transition of technologies as “state-of-the-art” improved
- Defined and refined standards; eliminate ambiguity in interface behavior

Community

- Creation of the UEFI Forum and associated community (2005)
- In August 2015
  - 11 Promoter : 44 Contributor: 208 Adopters: 25 Individual Adopters - 288 total members
- "*You may be assuming that the traditional competitiveness between companies persists in the UEFI Forum and the spec work groups it oversees. However, there is actually very little of that, especially compared with other industry-standards bodies. The general attitude within UEFI is that the firmware layer should be unified, interoperable, well-specified and secure. There is no room for competition or company-specific advantage in the firmware layer.*" **Gary Simpson** (Firmware Architect at AMD, 03-August-2015)

Lessons on the way

- “Open Source” is more than available source code
  - Correct tools
  - Correct process
  - Open communication
  - Community and consensus
- Open Source is not necessarily for everyone
  - Some small scale manufacturers do not have the resources or time to invest in developing expertise in code base

UEFI Opening Up

- Update to site layout
  - New layout to make space for more open communication and process
  - Improved organization of information
  - Creating addition channels to discuss specification issues and implementations
  - Sharing approved specification changes before publication of new specification
  - Discussion allowed for refinement before publication

Opening Further

- Tianocore.org ‘facelift’
  - Site user interface upgrades
  - Conversion from SVN to GIT on UEFI Open Source Community Website
  - Improved communication and Email lists
  - Adjustments in hosting to accommodate the unique aspects of firmware
- Industry providing additional options

The Future is Now

- In 10 years (LinuxCon 2025)
  - UEFI sees tremendous opportunity to:
    - Learn
    - Grow
    - Change

We WILL be better!

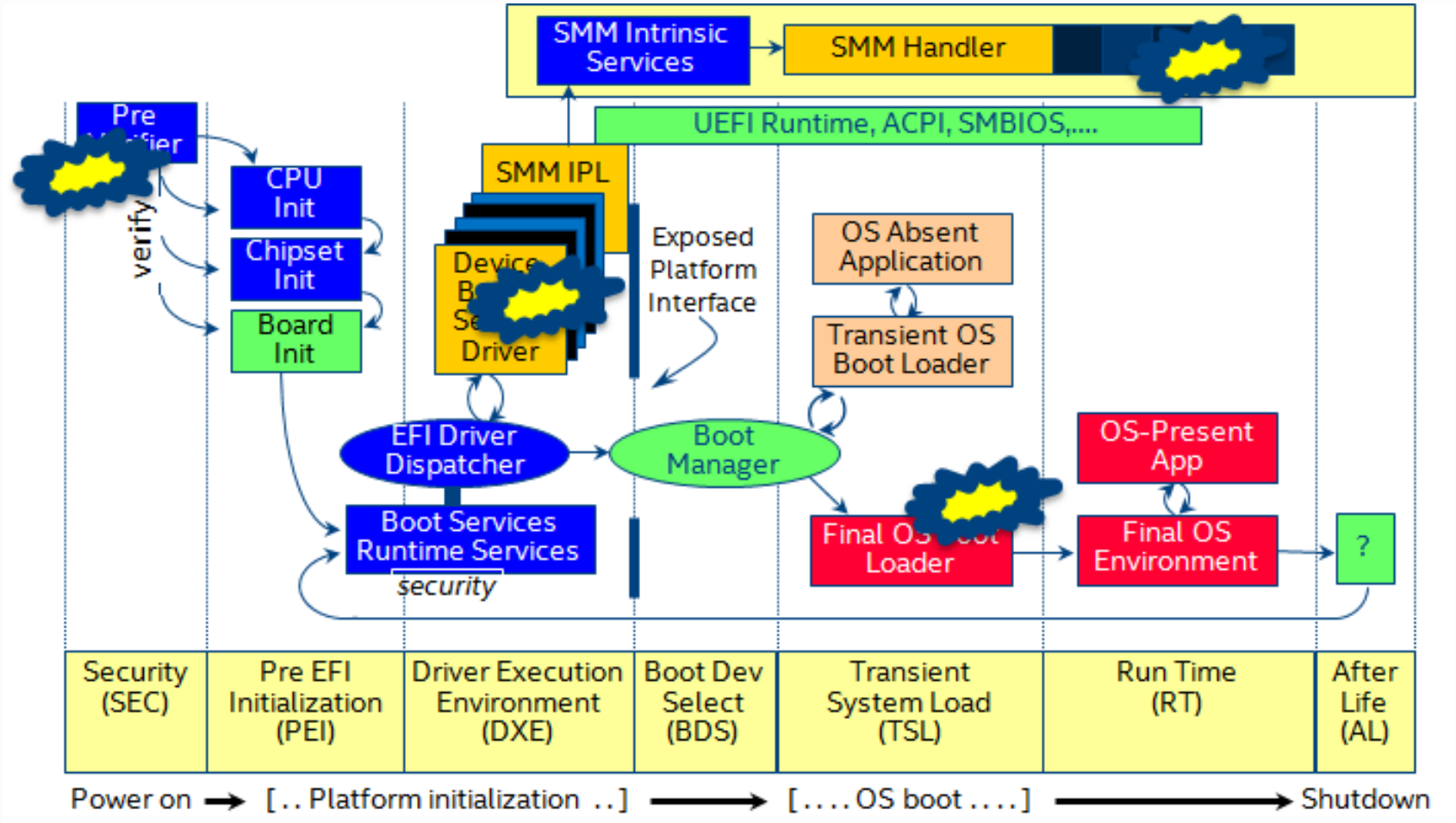
A case of learning & growth

Security in the Firmware space

- Platform Threats:
  - BIOs Malware
  - Bootkits
  - Device FW Malware
  - Option Rom Malware
  - HVM Rootkits (Blue Pill)
  - UEFI RootKits
  - SMM Rootkits
  - ACPI Rootkits
  - Evil Maid
  - HW Trojans

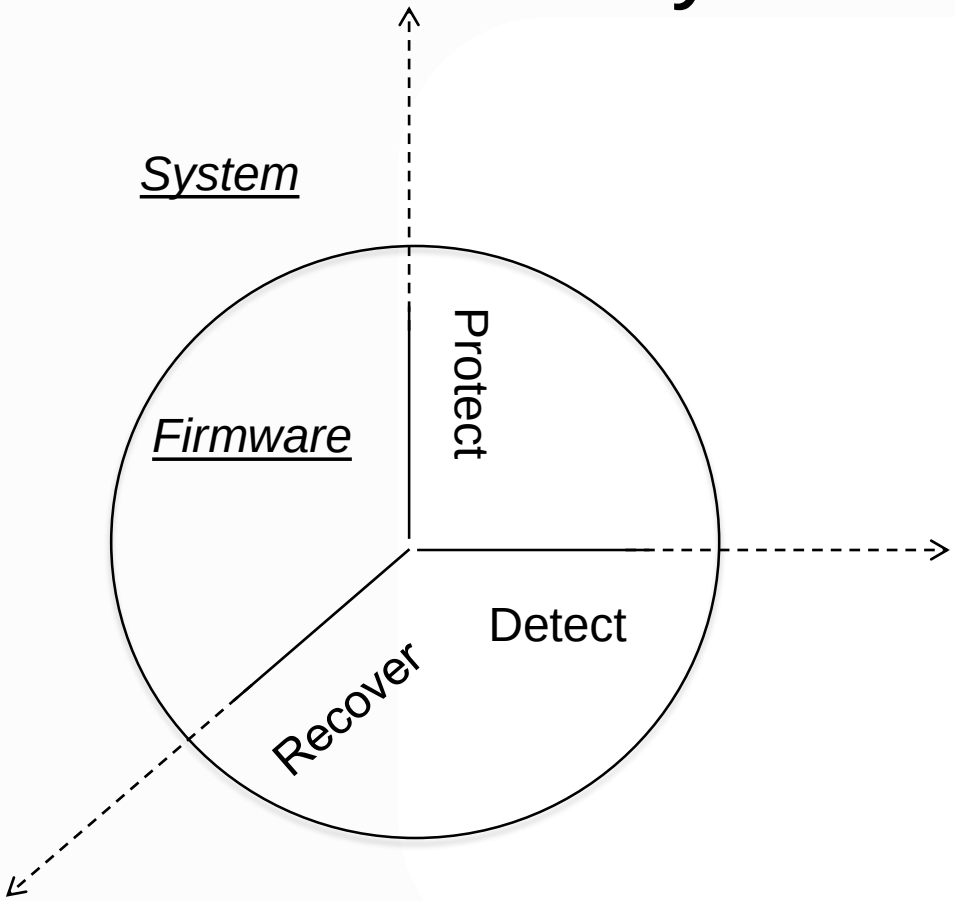
Security Fundamentals

What could go Wrong???



Security Ingredients

- Three vectors of security

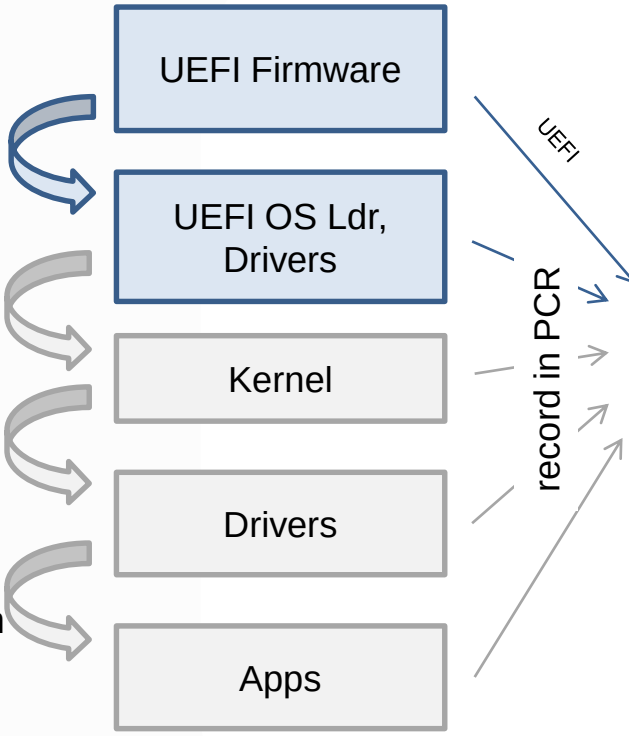


UEFI Secure Boot vs. TCG Trusted Boot

UEFI authenticate OS loader
(pub key and policy)

Check signature of before loading

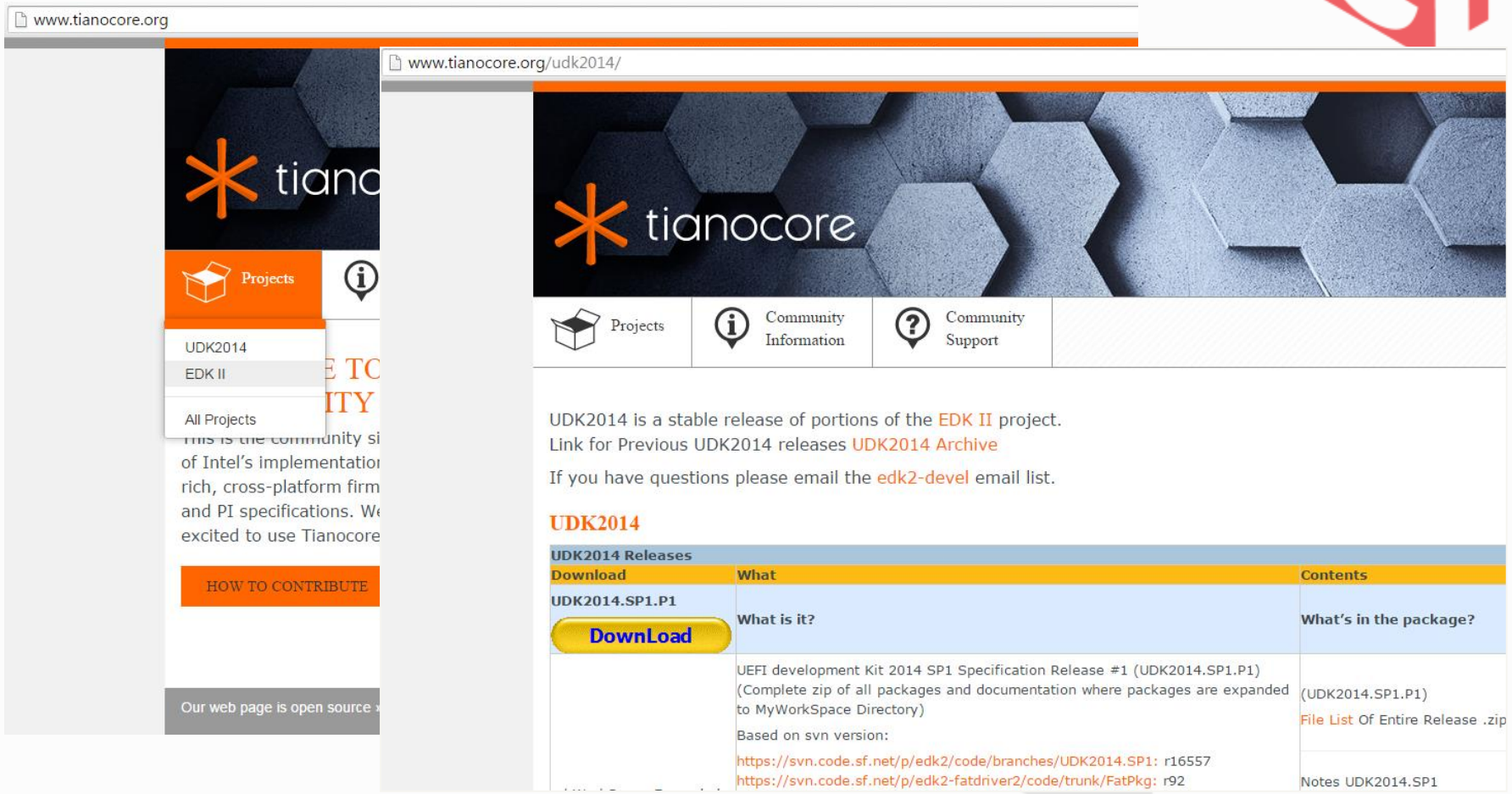
- UEFI Secure boot will stop platform boot if signature not valid (OEM to provide remediation capability)
- UEFI will require remediation mechanisms if boot fails



UEFI PI will measure OS loader & UEFI drivers into TPM (1.2 or 2.0) PCR (Platform Configuration Register)

- TCG Trusted boot will never fail
- Incumbent upon other software to make security decision using attestation

Open Source Content



UDK2014 Available on Tianocore.org

UDK2015 Coming Soon

UEFI Developerment Kit 2014 Security Package

RandomNumberGenerator

- UEFI driver implementing the EFI_RNG_PROTOCOL from the UEFI2.4 specification

Trusted Computing Group (TCG)

- PEI Modules & DXE drivers implementing Trusted Computing Group measured boot
- EFI_TCG_PROTOCOL and EFI_TREE_PROTOCOL from the TCG and Microsoft* MSDN websites, respectively

UserIdentification

- DXE drivers that support multi-factor user authentication
- Chapter 31 of the UEFI 2.4 specification

Library

- DxeVerificationLib for “UEFI Secure Boot”, chapter 27.2 of the UEFI 2.4 specification + other support libs

VariableAuthenticated

- SMM and runtime DXE authenticated variable driver, chapter 7 of the UEFI2.4 specification

https://svn.code.sf.net/p/edk2/code/trunk/edk2/SecurityPkg

Additional Capabilities in Open Source

Variable Lock Protocol

Make variables read-only

https://github.com/tianocore/edk2/blob/master/MdeModulePkg/Include/Protocol/VariableLock.h

Lock Box

Protect content across re-starts

https://github.com/tianocore/edk2-MdeModulePkg/blob/master/Include/Protocol/LockBox.h

Capsule Update

Generic capsule update driver support

http://comments.gmane.org/gmane.comp.bios.tianocore.devel/8402

Recovery

Device support for recovery from PEI

https://svn.code.sf.net/p/edk2/code/trunk/edk2/MdeModulePkg/Include/Guid/RecoveryDevice.h

https://svn.code.sf.net/p/edk2/code/trunk/edk2/

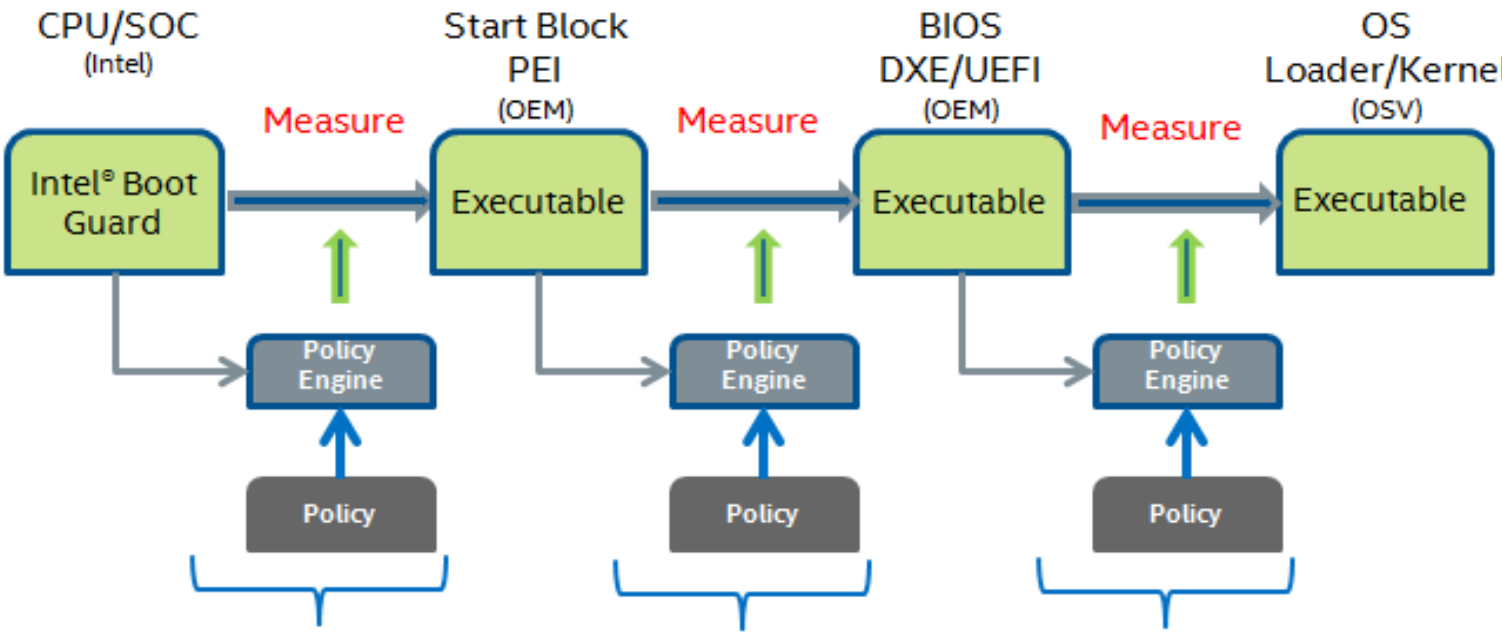
Code Management

Analyze and Mark external Interfaces where input can be attacker controlled data, comment headers

```
/**  Install child handles if the Handle supports GPT partition structure.

   Caution: This function may receive untrusted input.
   The GPT partition table is external input, so this routine will do basic
   validation
   for GPT partition table before install child handle for each GPT partition.

   @param[in]  This       Calling context.
   @param[in]  Handle     Parent Handle.
   @param[in]  DevicePath Parent Device Path.

 **/
 EFI_STATUS
 PartitionInstallGptChildHandle
```

UEFI Development Kit 2010 example:
http://edk2.svn.sourceforge.net/svnroot/edk2/trunk/edk2/MdeModulePkg/Universal/Disk/PartitionDxe/Gpt.c

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

Conclusions

- UEFI is about:
  - Standards
  - Evolution
    - Security
    - Technology
    - Practices and interactions
  - Cooperation
  - Community
- Protecting the interests of the entire ecosystem in the Firmware space

Questions?

Interested in Joining?

www.uefi.org/membership

UEFI FW/OS Forum:
uefi.org/FWOSForum
A free public forum focused on firmware and O/S integration

USRT Security Issue Reporting:
uefi.org/security
A safe reporting site to inform the UEFI of any security issue or vulnerability based on firmware