presented by

Microsoft

General Firmware Overview of Recommendations for Window OS

Spring 2017 UEFI Seminar and Plugfest

March 27 - 31, 2017

Presented by Fei Zhou (Microsoft, Inc.)

Agenda

- Introduction
- Creators Update
  - Device Guard & Credential Guard
  - Firmware(FW) update through Windows Update (WU)
  - Secure Boot
  - HSTI
- Call to Action, Reference and Links

Introduction

- Fei – who am I?
- Why is Microsoft, a software company, interested in UEFI?
  - Firmware is software
- Security starts in the firmware
  - Ensure Boot Firmware is protected
  - Control is passed to OS boot files
  - The OS is now able to build on the secure foundation

Device Guard, Credential Guard, Firmware Updates in WU, SMBIOS, UEFI CA

Device Guard & Credential Guard

- DG/CG Readiness tool
- DG/CG New Requirements
  - Virtualization-based security (VBS) enables NX protection for UEFI RUNTIME SERVICES
  - Firmware support for SMM Protection (WSMT table)
- Firmware (FW) updates through Windows Update (WU)
- SMBIOS fields
- UEFI Certificate Authority (CA)

Device Guard & Credential Guard

- Latest release of Device Guard and Credential Guard Readiness Tool v3.0 link to Microsoft Download page
  - Fixes HSTITest.dll, clears LSALSO's UEFI var, better text strings & messages, speed optimized, now also works on Windows 10 Pro editions
- PC OEM requirements for Device Guard and Credential Guard published on MSDN

DG/CG: First New Requirement

- **VBS enablement of NX protection for UEFI RUNTIME SERVICES**
  - VBS will enable No-Execute (NX) protection on UEFI runtime service code and data memory regions. UEFI runtime service code must support read-only page protections, and UEFI runtime service data must not be executable implement the UEFI 2.6 EFI_MEMORY_ATTRIBUTES_TABLE. All UEFI runtime service memory (code and data) must be described by this table
    - PE sections need to be page-aligned in memory (not required in non-volatile storage)
    - The Memory Attributes Table needs to correctly mark code and data as RO/NX for configuration by the OS
      - All entries must include attributes EFI_MEMORY_RO, EFI_MEMORY_XP, or both.
      - No entries may be left without either of the above attributes, indicating memory that is both executable and writable. Memory must be either readable and executable or writeable and non-executable (Documented on MSDN)

  Notes:
    - Only applies to UEFI runtime service memory and not UEFI boot service memory. Protection is applied by VBS on OS page tables
    - Do not use sections that are both writable and executable
    - Do not attempt to directly modify executable system memory
    - Do not use dynamic code
- Microsoft has open sourced our test tool for Memory Map and EMAT verification on GitHub

DG/CG: Second New Requirement

- Firmware Support for SMM Protection (WSMT table)
  - Firmware vendors must **re-design** System Management Interrupts (SMIs) to only read or write to an "allowed" list of regions that include MMIO and EFI-allocated memory. It is not enough to check that pointers are outside of SMM, they must only be within these "safe" regions. This prevents SMM from becoming a confused deputy which can bypass Windows flagship "Guard" features
    - NOTE: Page protections that "enforce" pointers being in only "safe" regions are not yet required
  - **If, and only if,** you do the above re-design, then you may implement WSMT to assert you have completed this work. **If you implement WSMT without doing this work, you are leaving an open attack portal**
  - WSMT, once implemented, confirms that the commbuffer has been fixed for SMI processing. Details on WSMT are on MSDN

Firmware Updates Delivered Through Windows Update (WU)

- Reliable field-update of firmware is a critical security feature and necessary to sunset legacy application transport types
- Enables most expedient update mechanism for system and device firmware
- In order to have the ability to target a system for firmware update in the field, the OEM/ODM/IHV will need to prepare the system while on the factory floor
  1. Enable EFI UpdateCapsule() & QueryCapsuleCapabilities in firmware
  2. Create a unique ID for each model that is destined to receive the same firmware package
     1. Online documentation will refer to the Unique ID as a GUID or UUID it is not globally or universally unique. It functions as a target designator)
  3. Populate ACPI table 'EFI System Resource Table (ESRT)' with unique ID
  4. Package uploaded to Microsoft will require a Computer Hardware ID (CHID)
  5. CHID is generated using a tool provided in Microsoft SDK (ComputerHardwareIDs.exe)
  6. CHID generation relies on populated SMBIOS fields described later
- **Review documentation on MSDN before implementing**

SMBIOS

- Microsoft is driving initiatives to enable use of SMBIOS for management of systems as described in DMTF documentation
- Needed SMBIOS fields: **manufacturer, family, product name, base board product, SKU number, and enclosure type**
- The ComputerHardwareIDs.exe tool will generate Computer Hardware IDs (CHIDs)
  - Empty fields result in no CHID generation that use that empty field
  - A non-unique SMBIOS field can generate a non-unique ID
  - A CHID is used along with Unique ID in ESRT for firmware targeting
- Microsoft has documentation describing SMBIOS fields and usage/ Downloadable document on MSDN

UEFI CA updates

- **Microsoft UEFI CA Signing policy updates**
  - Microsoft will not sign EFI submissions that use EFI_IMAGE_SUBSYSTEM_EFI_RUNTIME_DRIVER
    - Use EFI_IMAGE_SUBSYSTEM_EFI_BOOT_SERVICE_DRIVER to prevent unnecessary use of runtime EFI drivers
  - Use of EFI Byte Code: Microsoft will not sign EFI submissions that are EBC-based submissions
  - For Linux shims, there will be a new review process hosted by Linux Shim Review Board Shim-review@lists.freedesktop.org

Enforcement Date: 01/01/2017

UEFI CA Update (cont.)

- If your submission is a DISK encryption or a File/Volume based encryption, then you MUST make sure that you either *don't* encrypt the EFI system partition or if you *do* encrypt, be sure to decrypt it and make it available by the time Windows is ready to boot
- Microsoft UEFI CA Signing Policy Updates located on the Microsoft Hardware Certification Blog

Enforcement Date: 01/01/2017

Updates on Secure Boot

Windows Hardware Compatibility Requirements for Creators Update

Changes coming to
System.Fundamentals.Firmware.UEFISecureBoot

- Microsoft encourages all partners to follow the “recommended” options below for reserving NVRAM, with longer term OS requirement to move to a 128KB model. We will be updating the required specifics at a later date
- **Sub-Bullet 30:** Reserved Memory for Windows Secure Boot UEFI Variables. A total of at least 64KB (Recommended 128KB)of non-volatile NVRAM storage memory must be available for NV UEFI variables (authenticated and unauthenticated, BS and RT) used by UEFI Secure Boot and Windows. The maximum supported variable size must be at least 32KB (Recommended 64Kb). There is no maximum NVRAM storage limit.
- Below are the current attribute options we are expecting use cases for.

```
#define EFI_VARIABLE_NON_VOLATILE                          0x00000001
#define EFI_VARIABLE_BOOTSERVICE_ACCESS                    0x00000002
#define EFI_VARIABLE_RUNTIME_ACCESS                        0x00000004
#define
EFI_VARIABLE_HARDWARE_ERROR_RECORD                         0x00000008
#define
EFI_VARIABLE_AUTHENTICATED_WRITE_ACCESS                    0x00000010
#define
EFI_VARIABLE_TIME_BASED_AUTHENTICATED_WRITE_ACCESS   0x0000
0020
#define EFI_VARIABLE_APPEND_WRITE                          0x00000040
```

Customized Deployment of Secure Boot

- re Secure Boot options programmatically
  - dmins can set and deploy PK/[future Secure Boot variables]
  - Uses new t modes from UEFI 2.5 Section 30.3
    - New modes: Setup, loyed, Audit
    - Ship in Deployed Mode
- Requires PCR[7] with TPM 2.0

Yes we want it, but the implementation is not ready for it at this time

Hardware Security Test Interface (HSTI)

Hardware Security Test Interface

- Features like Device Guard, Credential Guard and Device Encryption require HSTI
- HSTI provides a self-test for the HW & FW security configuration that is *not* represented by the UEFI “SecureBoot” global variable
  - It fills the Pre-UEFI & outside-UEFI security testability gap
  - Tests provided by IBV, ODM and OEM
  - Helps ensure correct security configuration
  - Questions about implementation? Start with your silicon provider and BIOS/Firmware Vendor
- HSTI 1.1.a, details on MSDN

Call to action, Reference, Links

Call To Action

- Compatibility readiness for Creators Update
- Check out the DG/CG Requirements and Validation Tool
- Review Firmware updates on WU
- Review UEFI CA process
- Populate SMBIOS fields
- Make a note for updates to Secure Boot
- Review and implement HSTI 1.1.a

Reference

If you have questions, please do one of the following

- Connect with us at the Plugfest
- Follow up in email using the following alias (for Security and UEFI related Questions): SAUEFI@Microsoft.com

Links

Title	Resource link
Hardware Compatibility Specification for Systems for Windows 10, version 1607 System.Fundamentals.Firmware.UEFISecureBoot	https://msdn.microsoft.com/en-us/windows/hardware/commercialize/design/compatibility/systems
Device Guard and Credential Guard readiness tool	https://www.microsoft.com/en-us/download/details.aspx?id=53337
PC OEM requirements for Device Guard and Credential Guard	https://msdn.microsoft.com/library/windows/hardware/mt767514(v=vs.85).aspx
ACPI system description tables	https://msdn.microsoft.com/en-us/library/windows/hardware/dn495660(v=vs.85).aspx#wsmt
Windows SMM Security Mitigations Table (WSMT)	http://go.microsoft.com/fwlink/p/?LinkId=786943
Windows UEFI firmware update platform	https://msdn.microsoft.com/en-us/windows/hardware/drivers/bringup/windows-uefi-firmware-update-platform
Driver Publishing Workflow for Windows 10	http://download.microsoft.com/download/B/A/8/BA89DCE0-DB25-4425-9EFF-1037E0BA06F9/windows10_driver_publishing_workflow.docx
Microsoft UEFI CA Signing policy updates	https://blogs.msdn.microsoft.com/windows_hardware_certification/2013/12/03/microsoft-uefi-ca-signing-policy-updates/
UEFI CA test we ask submitters to perform before submitting -Pre-submission testing for UEFI submissions	https://blogs.msdn.microsoft.com/windows_hardware_certification/2013/12/03/pre-submission-testing-for-uefi-submissions/
Unified Extensible Firmware Interface Specification 2.6	http://www.uefi.org/sites/default/files/resources/UEFI%20Spec%202_6.pdf
Hardware Security Testability Specification	https://msdn.microsoft.com/en-us/library/windows/hardware/mt712332(v=vs.85).aspx

Thanks for attending the Spring 2017 UEFI Seminar and Plugfest

For more information on the UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

Microsoft