presented by

Best Practices for UEFI Secure Boot Guidelines

UEFI 2021 Virtual Plugfest

April 15, 2021

Presented by Tim Lewis, Insyde Software and Manoj Khandelwal, HPE

Meet the Presenters

Tim Lewis, CTO
Insyde Software

Manoj Khandelwal, Software Engineer
HPE

Introduction

- The National Security Agency (NSA) of the United States released its “UEFI Secure Boot Customization” guidelines for configuring platform firmware to take advantage of the security promise provided by Secure Boot.

- Malware targets firmware because of its unique role in setting up and maintaining the platform’s hardware security capabilities. Also, anti-virus solutions today are limited in their ability to detect and remove it.

- UEFI’s Secure Boot prevents malware that targets platform firmware from getting started by verifying that each component is trusted before using it.

- The NSA’s six guidelines help IT administrators and end users correctly configure the UEFI Secure Boot and related settings in their BIOS.
  - UEFI (and related specifications) enable these capabilities, but they must be configured based on the actual use case of the platform.

Agenda

1. Turn On UEFI Boot
2. Turn On UEFI Secure Boot
3. Customize UEFI Secure Boot
4. Set Strong Administrator Passwords
5. Update BIOS Regularly
6. Verify BIOS Integrity With a TPM

1. Turn On UEFI Boot

- “Machines running legacy BIOS or Compatibility Support Module (CSM) should be migrated to UEFI native mode.”
- Current platforms support UEFI boot, but the security advantages are negated if not enabled.
- The previous “legacy” boot standard is inherently insecure, leaving opportunities for malware to insert itself into the boot process
  - Some BIOS vendors allow a “dual” mode or “combo” mode that tries to intelligently switch between legacy and UEFI. While convenient, this mode is not secure.

Why Do Companies Turn Off UEFI Boot?

- "It is working today, why change it?"
- "My OS/application only works in legacy mode"
- "My plug-in video card doesn't work."
- Internal processes tied to an OS or application only available in "legacy" mode.

"Leveraging legacy mode or CSM reintroduces security, access control, and memory vulnerabilities addressed by the UEFI standard and prohibits the use of UEFI Secure Boot."

2. Turn On UEFI Secure Boot

- “Secure Boot should be enabled on all endpoints and configured to audit firmware modules, expansion devices, and bootable OS images.”

- Some systems trade security for speed by allowing ‘fast boot’ to skip measuring or verifying certain parts of the firmware

3. Customize UEFI Secure Boot

- “Secure Boot should be customized, if necessary, to meet the needs of organizations and their supporting hardware and software.”

- Most BIOS setup utilities and many BMC management interfaces allow the Secure Boot keys and certificates to replaced.

Why Customize UEFI Secure Boot?

- Customizing Secure Boot allows administrators to:
  - Respond to certain vulnerabilities without waiting for a BIOS update.
    - For example: fixes to the UEFI security revocation list can be updated manually for the grub2/shim vulnerabilities like BootHole and the 8 additional CVEs disclosed.
    - Do this with care, with guidance from your OS provider, to make sure you don’t prevent your platform from booting your current OS version.

Why Customize UEFI Secure Boot?

- Customizing Secure Boot allows administrators to:
  - Further reduce the attack vectors by:
    - Removing the standard UEFI CA certificate from Microsoft from the Secure Boot database and
    - Installing signatures or hashes of the specific OS boot loader (and plug-in card option ROMs) in use.
    - This prevents any other OS (including all Microsoft OS' and other versions of Linux) from being booted.
  - Disable booting to OS environments which might give the user enhanced access by
    - Disabling external ports (SATA, USB, etc.)
    - Preventing the changing of the boot order

Real World Example

Secure Boot at Hewlett Packard Enterprise

Why Customize UEFI Secure Boot?

- No BIOS Setup access, so secure boot needs to be enabled by default and cannot be disabled at all.

- If there is no user interface and no bash shell command prompt, traditional approaches to update the Secure Boot key don’t work. So Secure Capsule update is used.

- If HW root of trust is missing (e.g. Intel Denvertron Soc), You may need to apply some customized solutions on top of UEFI Secure Boot to enable more comprehensive security coverage.

Generate Capsule Image for BIOS Upgrade

- There are 3 components while creating the capsule image.
  - The SPI flash BIOS image.
  - The UEFI flash update application.
  - A configuration file for controlling the update options.
- These 3 components are combined and then signed to generate the signed capsule image.
- The UEFI flash update application must be signed also because if secure boot is on, then the application is an external executable to the system which also needs to be verified.
- In HPE's case, we used our own signing tool, which mimics Microsoft's signtool but sends the image to HPE signing server and gets the sign image.

BIOS Upgrade in Secure Boot

- Copy the signed capsule image file to the UEFI spec-defined location for capsule storage (e.g /EFI/UpdateCapsules) . Trigger the BIOS upgrade by setting the OsIndications UEFI variable followed by a reboot.
- During BIOS boot, check if OsIndications is set and whether the capsule file is present.
- Verify the capsule signature using the HPE key (integrated within the BIOS). If verification succeeds then call the flash application in the capsule to update the BIOS.
- When the BIOS update is done, the capsule image is deleted and OsIndications variable is cleared, and perform a hard reset.

BIOS Integrity Check

- Performing a BIOS integrity check is not trivial because:
  - BIOS flash image doesn't contain any HPE signature/certificate (Signature and Certificate are stripped off before capsule contents are written into the flash)
  - When flashing the BIOS, only a few regions’ contents change across reboot (e.g. GBE region, ME stats, BIOS stats, NVRAM area etc.), so the entire 16MB image isn’t considered for integrity check.

BIOS Integrity Check by BIOS

1. HPE introduced a new “signature” region at a fixed location in the final bios image (16MB).

2. After the BIOS build, the SHA256 hash of specific BIOS regions is calculated. These regions include static code, the BIOS version information and microcode.

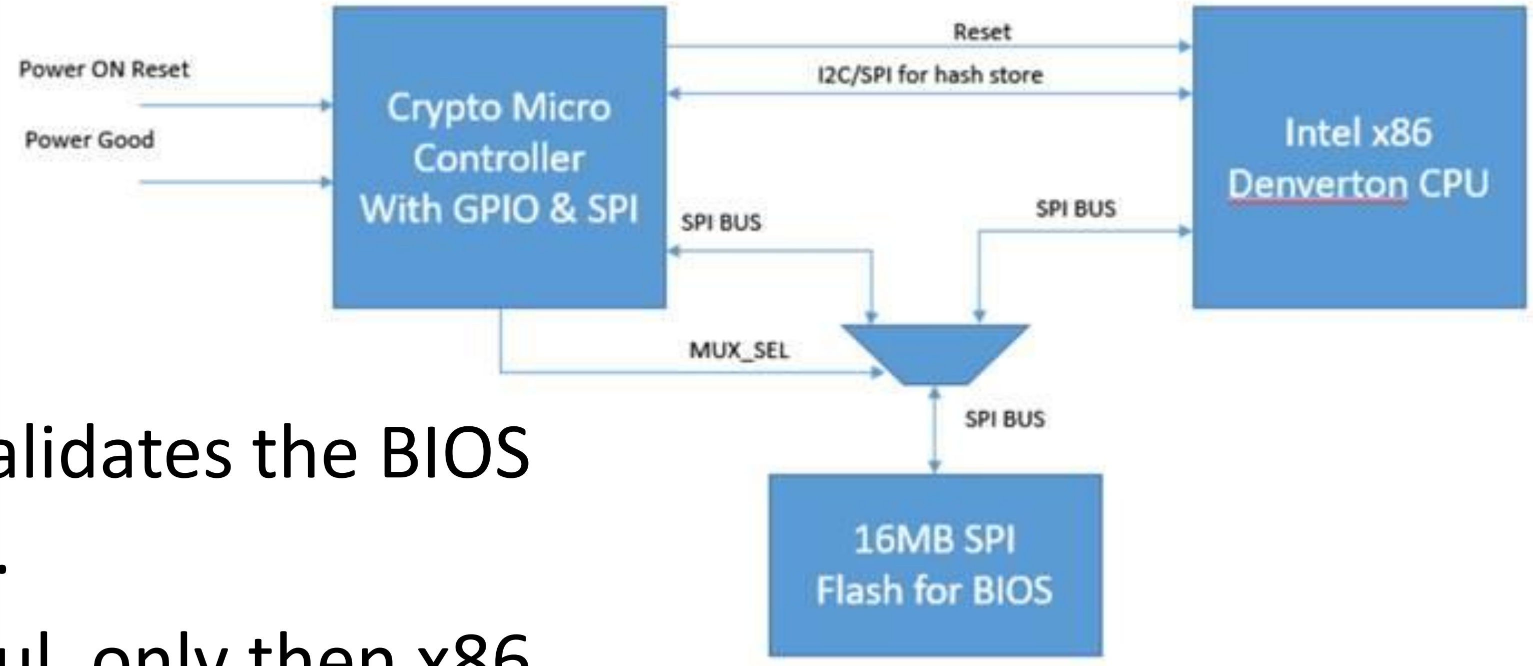
3. The SHA256 hash is sent to the HPE signing server. The server returns a signed hash image from the HPE signing server (32 bytes + signature and certificate size), which is copied into the “signature” region

4. After power on, the early BIOS code calculates the combined hash of each of the specific valid regions within the BIOS image.

5. Then, after verifying the “signature” region contents are valid, the BIOS compares the stored hash and the calculated hash. If both are same, boot continues. Otherwise halt the system.

BIOS Integrity Check by External Chip

- This is a real HW root of trust example:



- This external crypto chip validates the BIOS in SPI flash post power ON.
- Once validation is successful, only then x86 CPU will be out of reset, or else keep in reset.
- This chip has a fusing option so that we can fuse one public key hash which will be used to verify the signature of hash file stored in the signature region.
- Validation flow is similar to BIOS integrity check by BIOS, except that that the external chip is doing the validation which makes it the real HW root of trust.

4. Set Strong Administrator Passwords

- “Firmware should be secured using a set of administrator passwords appropriate for a device's capabilities and use case.”
- UEFI specification provides password support via the User Identity and HII (see chapters 33-36).
- Administrator passwords in most platforms restrict access to BIOS configuration options that control UEFI Secure Boot and other platform security features.
- Therefore, if the password is not secure, the BIOS is not secure and the platform is not secure.

Administrator Password Guidance

- Firmware passwords should meet the same industry-wide requirements as OS passwords (complexity, length, re-use, etc.)
  - Possibly add 2nd-factor such as a secure dongle, etc.

- Weak or re-used passwords are still a problem that requires each organization to establish a policy for creating and tracking passwords.

Administrator Password Guidance

- Default passwords are dangerous passwords because it is effectively the same as no password
  - Hacker can lock you out of your own system or introduce unnoticed changes to security
- Governments are addressing this by regulation:
  - California's SB-327 requires a unique default password for each device, or forcing users to set their own password the first time they connect

5. Update BIOS Regularly

- “Firmware should be updated regularly and treated as importantly as operating system and application updates.”
- UEFI provides standard mechanisms for passing firmware updates (called capsules)to the BIOS.
- Like other software in the system, firmware may need regular updates as security issues are discovered and security fixes released.

Computer User BIOS Update Guidance

- Are you notified when a firmware security update is available?
- Are these updates reliably distributed and applied within your company?
- Do your BIOS settings allow these firmware updates to be applied without human intervention?
  - NSA guidelines point out that hard disk or user passwords that halt the firmware update reboot cycle are not recommended.

Computer Manufacturer Update Guidance

- Are firmware updates published in such a way that they will be detected and applied by the computers you manufacture? (LVFS, Windows Update, proprietary utility, etc.)

6. Verify BIOS Integrity with a TPM

- “A Trusted Platform Module (TPM) should be leveraged to check the integrity of firmware and the Secure Boot configuration.”
- The Trusted Computing Group (TCG) provides specifications that layer additional security capabilities on top of UEFI Secure Boot

6. Verify BIOS Integrity with a TPM

- The BIOS uses the TPM to measure the firmware and the Secure Boot configuration and passes this to the operating system.

- TPM-aware OS boot loaders check that the measurements at boot match recorded “golden” measurements.
  - Examples include: Microsoft boot loaders, Trusted SHIM, Trusted GRUB, Tboot, and TPM-rEFInd

Call To Action

- Deliver/buy platforms that can be configured more securely, based on the NSA guidelines and your environment and use cases.
- Enable UEFI Secure Boot. Consider customization for more control. UEFI has great security features, if you use them.
- Develop a specific UEFI configuration for each make and model device. Write them down.
- Enable and activate the Trusted Platform Module (TPM).

*Based on the NSA's "UEFI Lockdown Quick Guidance"

Questions?

More Questions?

Following today’s webinar, join the live, interactive WebEx Q&A for the opportunity to chat with the presenter

Visit this link to attend: https://bit.ly/2QKqwl5
Meeting number (access code): 182 223 2926
Meeting password: UEFIForum (83343678 from phones and video systems)

Thanks for attending the UEFI 2021 Virtual Plugfest

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

presented by

insyde®

Hewlett Packard Enterprise

Thank You!

Resources

- UEFI Defensive Practices Guidance
  - https://www.nsa.gov/Portals/70/documents/what-we-do/cybersecurity/professional-resources/ctr-uefi-defensive-practices-guidance.pdf
- UEFI Lockdown Quick Guidance
  - https://www.nsa.gov/Portals/70/documents/what-we-do/cybersecurity/professional-resources/csi-uefi-lockdown.pdf
- Boot Security Modes and Recommendations
  - https://www.nsa.gov/Portals/70/documents/what-we-do/cybersecurity/professional-resources/csi-boot-security-modes-and-recommendations.pdf
- Insyde’s white paper.
  - https://www.insyde.com/products/insydeh2o/whitepaper/insyde-nsa-uefi-guidelines

Resources

- USRT UEFI Revocation List
  - https://uefi.org/revocationlistfile
- TCG
  - EFI Protocol Specification - https://trustedcomputinggroup.org/resource/tcg-efi-protocol-specification/
  - EFI Platform Specification - https://trustedcomputinggroup.org/resource/tcg-efi-platform-specification/
- California SB-327
  - https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB327