presented by

Secure and Scalable Firmware Updates via Capsules

UEFI 2025 Developers Conference & Plugfest

October 9, 2025

Mohamad Saleh & Sean Loe

Meet the Presenters

Mohamad Saleh

VP of Engineering

Mohamad Saleh is Vice President of Engineering at Insyde Software, Inc., with over 25 years in firmware and embedded systems. He leads server, data center, and embedded engineering for Intel and AMD platforms, driving innovation and reliability. He also champions platform security solutions, including secure boot and hardware root-of-trust integrations.

Meet the Presenters

Sean Loe

Engineering Manager

Firmware Engineering Manager at Insyde Software, with over 15 years of experience in the BIOS industry. He specializes in OEM/ODM customization and resolving complex field issues, with a strong focus on firmware hardening.

Agenda

- Why Capsule Update
- UEFI Capsule Basics
- Capsule Structure
- Linux Capsule Flow
- Security Aspects
- Targets & Scenarios
- Q&A (+ Optional Deep Dive: CPU Microcode DMR vs Capsule)

Why Capsule Update?

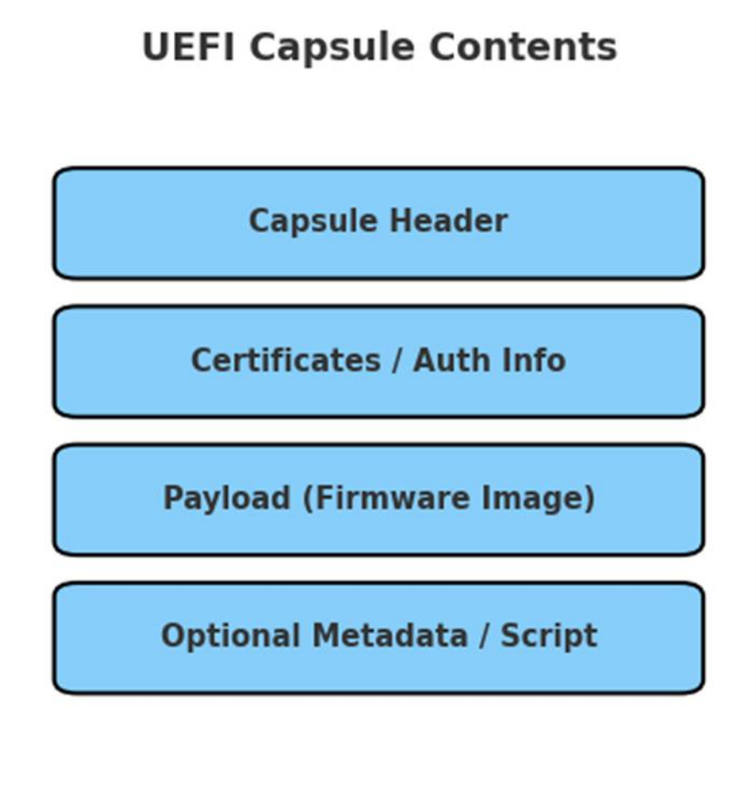
- Standardized firmware update method across OSes
- OS-initiated updates (Linux fwupd, Windows Update)
- Secure and authenticated firmware delivery
- Replaces vendor-specific update hacks

UEFI Specification Basics

- Defined in UEFI Spec, Section 8 – Capsule Services
- Capsule = header + payload (payload may contain, driver(s) + firmware)
- Delivered by OS → processed by firmware after reset
- Works with BIOS, TPM, NVMe, Microcode, and can also encompass embedded controllers, device firmware (network RAID), and sometimes configuration or platform-specific data.

Capsule Structure

- **Capsule Header**: [GUID(Globally Unique Identifier), Flags, Size]
- **Certificates**: Authentication Info
- **Payload**: Firmware image / package
- **Optional**: Metadata, Scripts



Capsule GUID vs Resource GUID

- **Capsule GUID** → identifies capsule format (e.g., FMP Capsule)
- **Resource GUID** (ImageTypeId) → identifies what is updated
  - **Examples**: BIOS, TPM, Microcode, NVMe, etc.
- **Exposed via ESRT (EFI System Resource Table)**

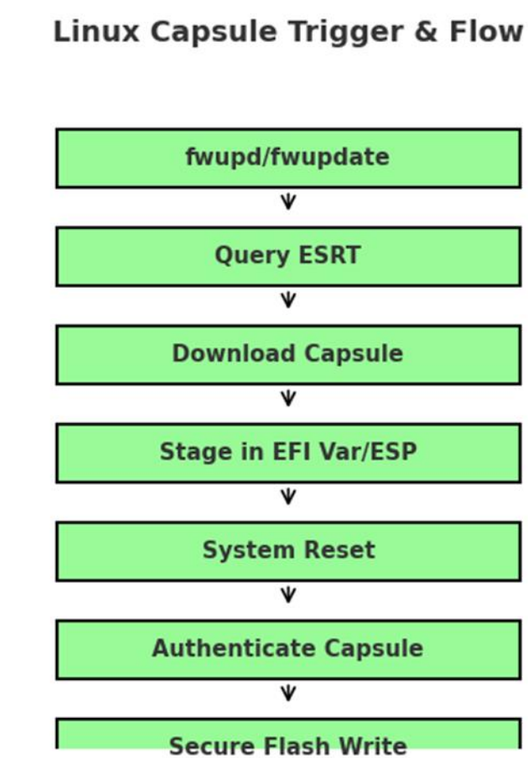
ESRT Importance

- EFI System Resource Table (ESRT) = UEFI configuration table
- Lists updatable firmware resources.
- Each entry: GUID, version, rollback, status
- OS uses ESRT to discover and manage updates

Linux Capsule Flow

- **Tools**: fwupd/fwupdate
- **Flow**:

1. OS queries ESRT
2. Downloads capsule
3. Calls UpdateCapsule()
4. Stage in UEFI variable or ESP (EFI System Partition)
5. System reset

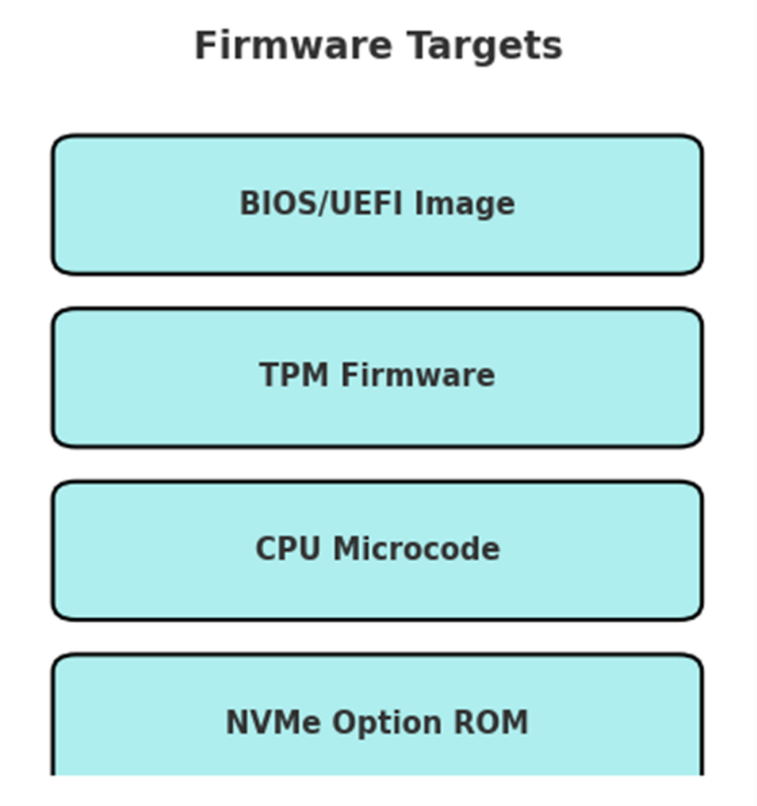


Security Aspects

- Small capsules staged in EFI variables, Large capsules in ESP
- Secure Boot validates capsule signature
- Rollback prevention via the lowest supported version
- Secure Flash protection via Boot Guard and SPI (Serial Peripheral Interface) lock

Targets & Scenarios

- After reset firmware scans storage for capsule
- Authenticates capsule
- Applies update to:
  - BIOS/UEFI image
  - TPM firmware
  - CPU Microcode
  - Device device firmware (NVMe/Network/RAID)



Summary

- UEFI Capsule = Secure, Standardized update mechanism
- Capsule GUID = format identifier
- Resource GUID via ESRT = Firmware Target
- Linux flow: fwupd + ESRT + UpdateCapsule()
- Supports BIOS, TPM, Microcode, Device firmware (network/storage/RAID) updates
- Secure Boot + Secure Flash ensure trusted updateS

References

• *Unified Extensible Firmware Interface (UEFI) Specification*, Version 2.11 (latest)
https://uefi.org/specifications

• Linux Vendor Firmware Service (LVFS) and fwupd client.
https://fwupd.org

• TCG PC Client Firmware Profile Specification (for TPM firmware update guidance).
https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification

• Intel® Boot Guard and Intel® BIOS Guard documentation for secure flash enforcement.
https://www.intel.com/content/www/us/en/developer/articles/technical/boot-guard-and-bios-guard.html

• Windows Update delivery of UEFI capsules.
https://learn.microsoft.com/en-us/windows-hardware/drivers/bringup/using-the-uefi-updatecapsule-function

CPU Microcode Updates – Capsule vs DMR

- Capsule (Traditional)

  vs

  Dynamic Microcode Replacement (Runtime)

- When Applied: Reset vs Runtime
- Persistence: Volatile (both)
- Delivery: Capsule vs OS/DMR interface
- Downtime: Requires reboot vs No reboot
- Use Case: Routine vs Emergency
- Security: Capsule signature vs Vendor-signed DMR

TPM Firmware Update via Capsule + FMP Driver

- Capsule Header (GUID)
- FMP (Firmware Management Protocol) DXE Driver
- TPM (Trusted Platform Module) Firmware Payload

Flow:

1. Capsule delivered
2. Driver loads in DXE (Driver Execution Environment)
3. Registers FMP protocol
4. Applies TPM firmware payload

Questions?