presented by

UEFI Unveiled: Ensuring Transparency in Your Firmware

UEFI 2024 Virtual Plugfest

Presented by Tim Lewis, CTO, Insyde Software

Meet the Presenter

Tim Lewis
CTO
Insyde Software

Agenda

- Introduction
- Use Cases
- Remaining Issues
- Questions

Introduction

SBOM – Software Bill of Materials

- “Software Bill of Materials” (SBOM) is a “formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships”*
  - formal machine-readable inventory – List with required elements.
  - software components – source code, data files and executables.
  - hierarchical relationships – Where did the components come from and how were they added to the software?

* Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM), CISA (September 2024) https://www.cisa.gov/sites/default/files/2024-10/SBOM%20Framing%20Software%20Component%20Transparency%202024.pdf

Why Use SBOMs?

License Management

- Which components are under which licenses?

Component Identity

- Which components make up the software?

Component Dependencies

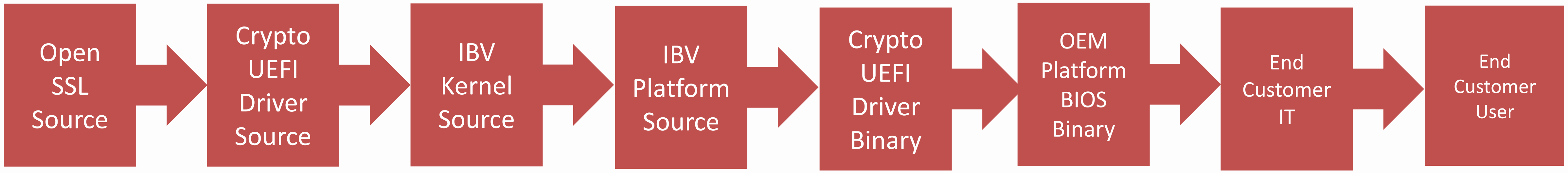
- Which components are created from other components?

Vulnerability Management

- Which components are affected by which vulnerabilities?

Why Customers Use SBOMs?

- Firmware is made from multiple components created and assembled by many parties (the "supply chain") before it reaches the end customer



- Each step in the supply chain is a possible place for vulnerabilities to be found and exploited.
- SBOMs give people at each step visibility into the components so that they know if they are affected by reported vulnerabilities.
- Governments are actively creating regulations for OEMs to secure the supply chain using SBOMs.
  - NIST800-218

What Does A SBOM Contain?

- **Supplier Name** - The name of an entity that creates, defines, and identifies components.
- **Component Name** - Designation assigned to a unit of software defined by the original supplier.
- **Version of the Component** - Identifier used by the supplier to specify a change in software from a previously identified version.
- **Other Unique Identifiers** - Other identifiers that are used to identify a component or serve as a look-up key for relevant databases.
- **Dependency** - Relationship characterizing the relationship that an upstream component X is included in software Y.
- **Author of SBOM Data** - The name of the entity that creates the SBOM data for this component.
- **Timestamp** - Record of the date and time of the SBOM data assembly.
- **Hash*** – Hash of the component.

https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf

SBOM Tradeoffs

Ease-of-generation - How easy is it to create the final SBOM?

Size - How many bytes does it take to store the SBOM?

Readability - How easy is it to find the relevant information? SPDX, SWID (JSON, INI), CycloneDX

SBOM Tradeoffs

Updatability - When components are updated, how difficult is it to update the SBOM?

IP Exposure – Are any details of the component IP and design exposed by in the product’s SBOM?

Paperwork – How difficult is it to maintain the SBOM for initial and subsequent versions of the SBOM?

SBOM Tradeoffs

Vulnerability Exposure – Could the information in the SBOM be used to attack the product?

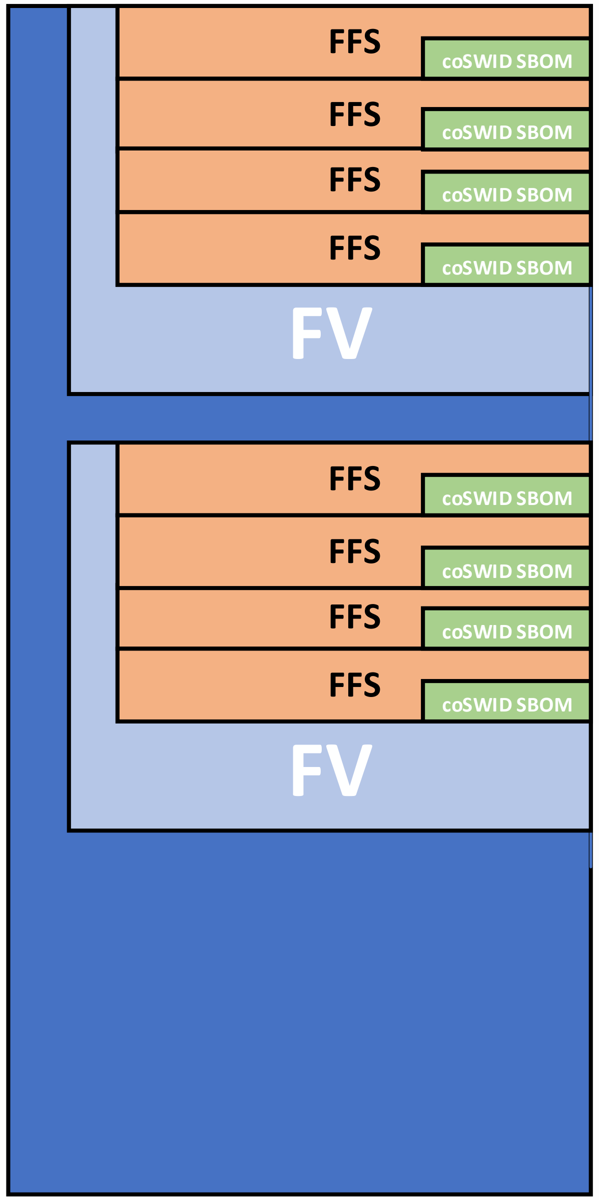
End User Accessibility – Can the end-user find whether the product’s components possibly contain a vulnerability?

How SBOMs Are Stored In The Platform?

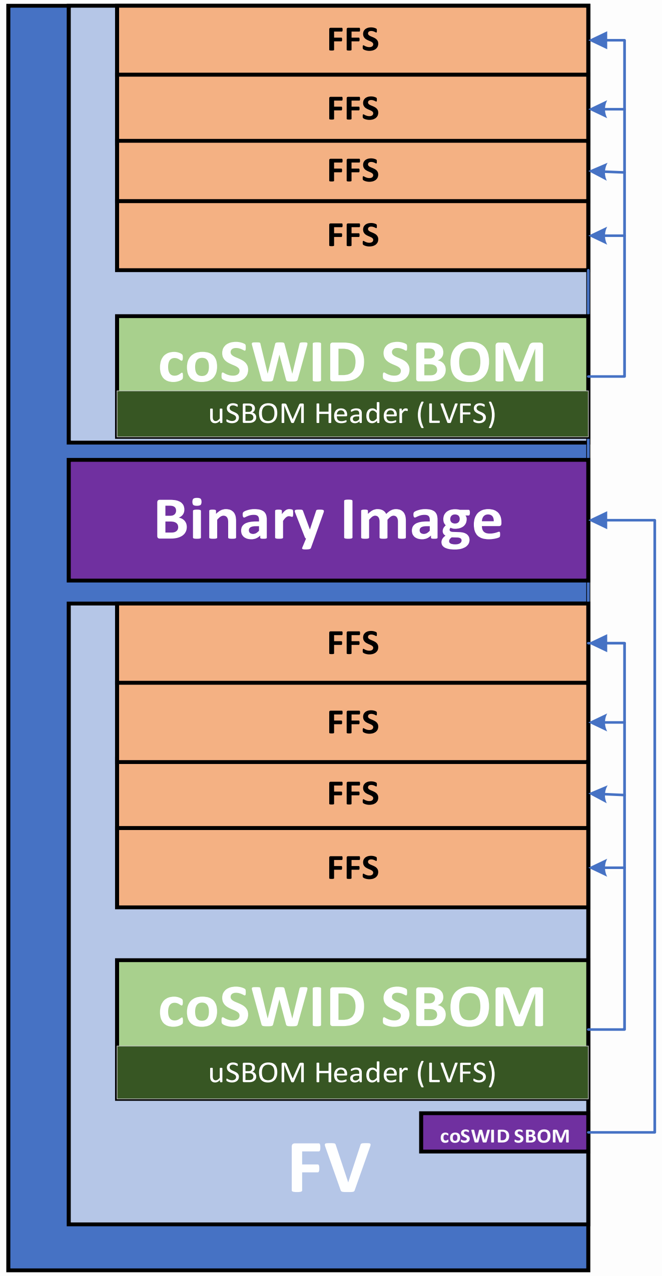
- SBOM information stored in each module. For example: hash and version per EFI EXE file.

- Separate SBOM for other regions (like FSP, option ROMs).

- Advantages: Self-contained, independent updatability.



How SBOMs Are Stored In The Platform?



- SBOM information stored In each region. For example: hash and version per firmware volume (FV) or per region.

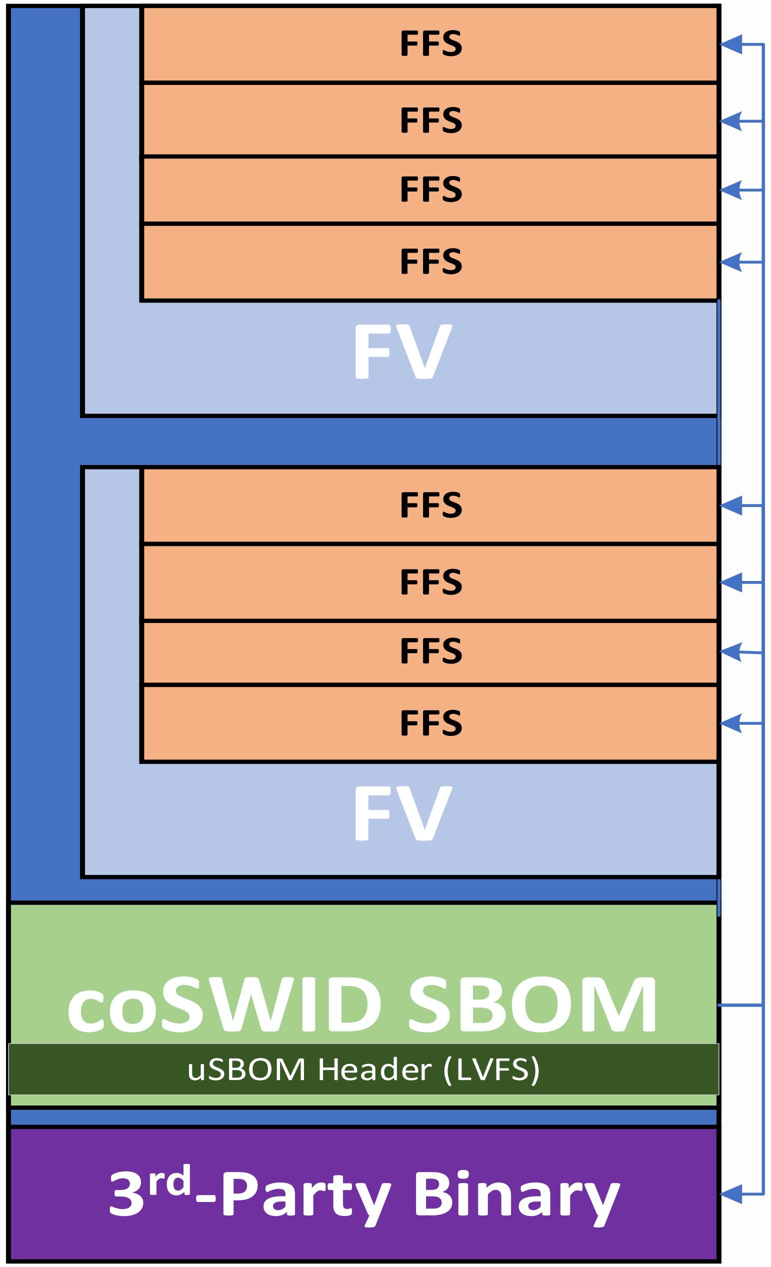
- Separate SBOM for other regions (like FSP, option ROMs).

- Advantages: Updatable by region, self-contained

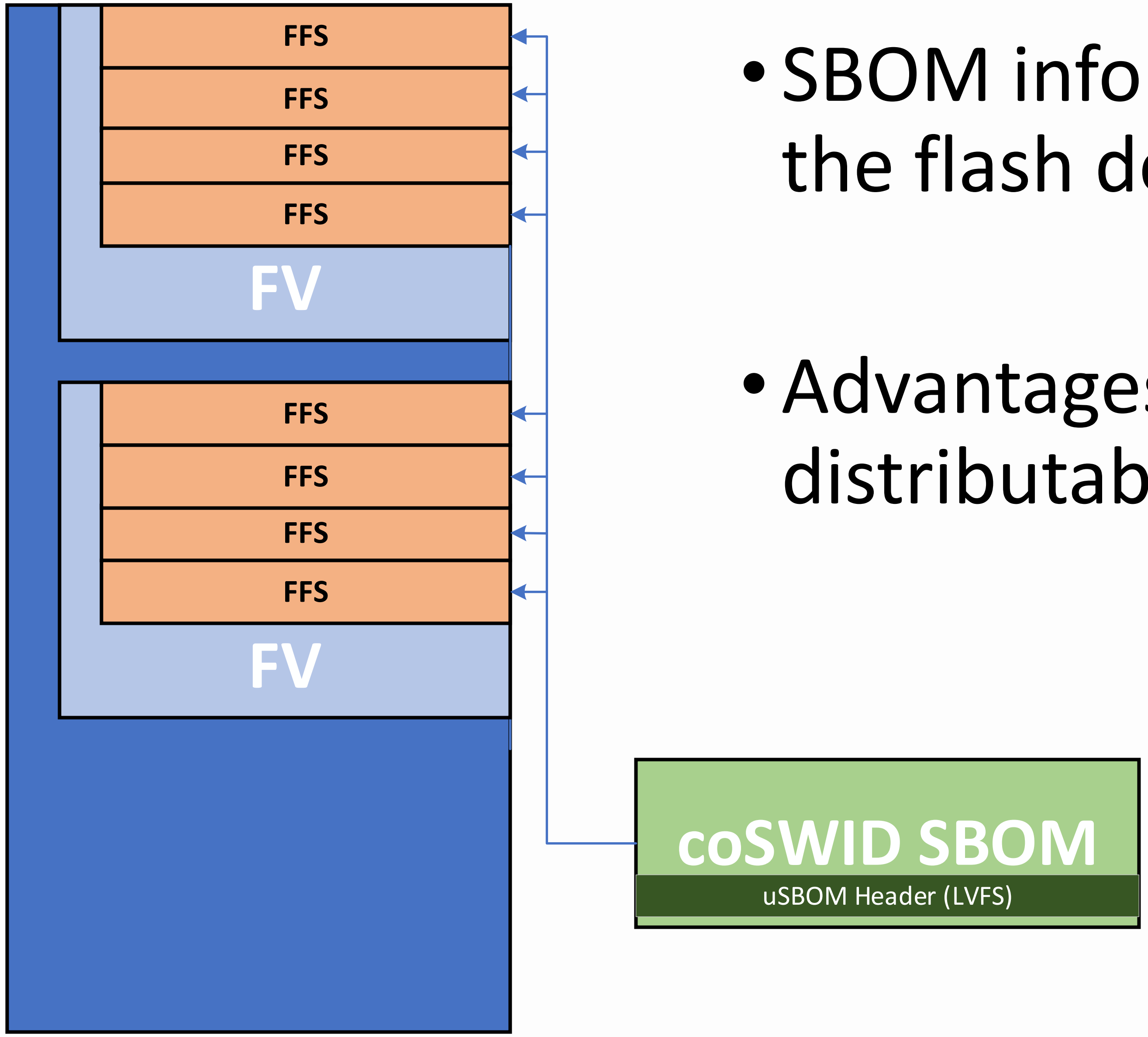
How SBOMs Are Stored In The Platform?

- SBOM information stored In a separate flash region.

- Advantages: Entire product’s SBOM.



How SBOMs Are Stored In The Platform?



- SBOM information stored outside the flash device.

- Advantages: size, separately distributable.

Use Cases

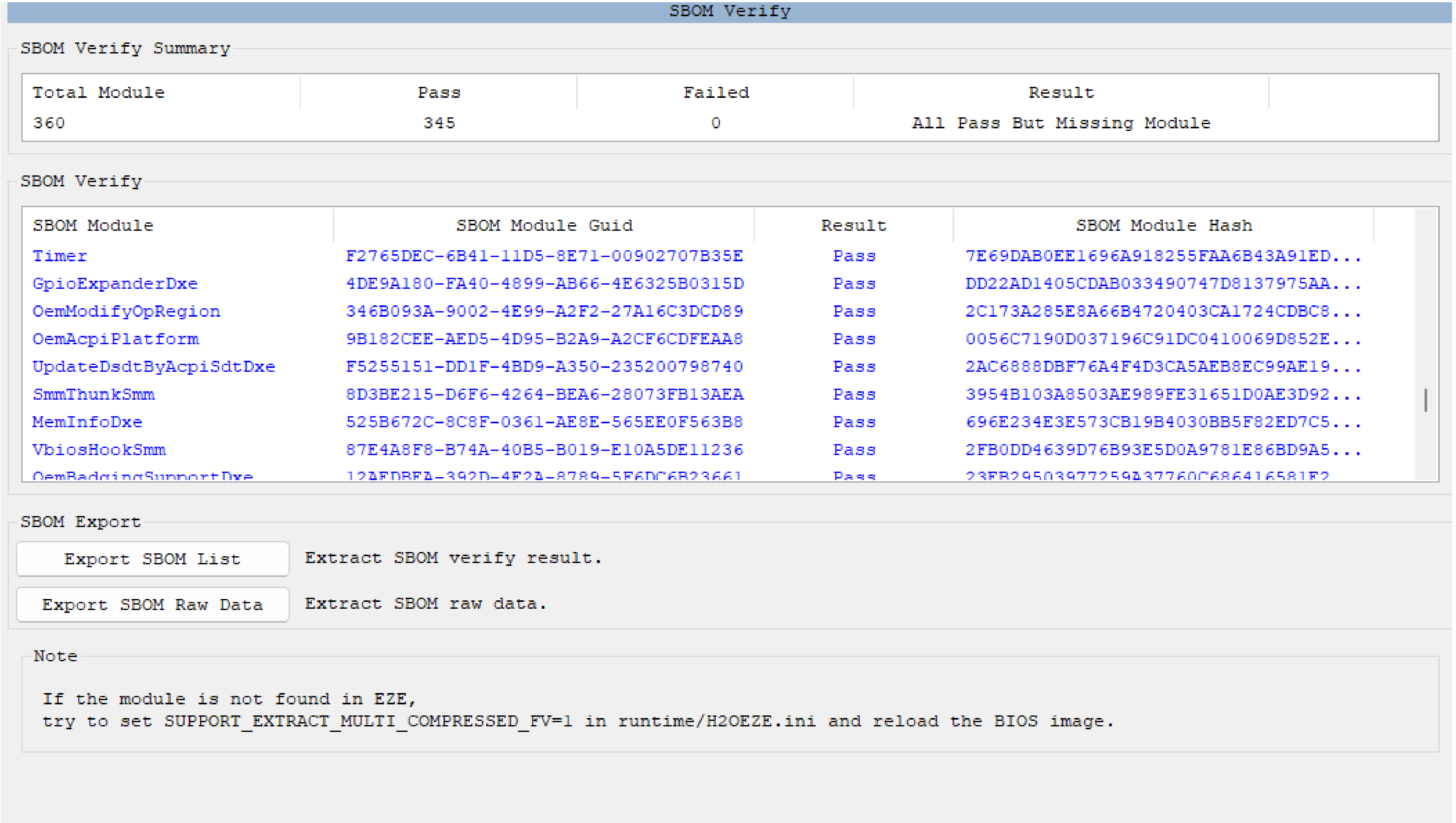
Use Cases

- **Verify SBOM** - Compare SBOM contents vs firmware image actual contents, report module with no entry.
- **Export SBOM** - Export SBOM for firmware in SPDX, CycloneDX or SWID format.
- **List Licenses** – List the licenses for all firmware modules.
- **Security Checks** – Check a SBOM in a VEX to see if there has been a CVE reported against it or any component within it.
- **External Viewing** – Government regulators are requiring uploading SBOMs to a public database.

SBOM Open-Source Tools

- uswid - https://github.com/hughsie/python-uswid

SBOM Tools – Insyde’s H2OEZE



SBOM Verify Summary

Total Module	Pass	Failed	Result
360	345	0	All Pass But Missing Module

SBOM Verify

SBOM Module	SBOM Module Guid	Result	SBOM Module Hash
Timer	F2765DEC-6B41-11D5-8E71-00902707B35E	Pass	7E69DAB0EE1696A918255FAA6B43A91ED...
GpioExpanderDxe	4DE9A180-FA40-4899-AB66-4E6325B0315D	Pass	DD22AD1405CDAB033490747D8137975AA...
OemModifyOpRegion	346B093A-9002-4E99-A2F2-27A16C3DCD89	Pass	2C173A285E8A66B4720403CA1724CDBC8...
OemAcpiPlatform	9B182CEE-AED5-4D95-B2A9-A2CF6CDFEAA8	Pass	0056C7190D037196C91DC0410069D852E...
UpdateDsdtByAcpiSdtDxe	F5255151-DD1F-4BD9-A350-235200798740	Pass	2AC6888DBF76A4F4D3CA5AEB8EC99AE19...
SmmThunkSmm	8D3BE215-D6F6-4264-BEA6-28073FB13AEA	Pass	3954B103A8503AE989FE31651D0AE3D92...
MemInfoDxe	525B672C-8C8F-0361-AE8E-565EE0F563B8	Pass	696E234E3E573CB19B4030BB5F82ED7C5...
VbiosHookSmm	87E4A8F8-B74A-40B5-B019-E10A5DE11236	Pass	2FB0DD4639D76B93E5D0A9781E86BD9A5...
OemBadgingSupportDxe	12AFDBFA-392D-4F2A-8789-5F6DC6B23661	Pass	23FB29503977259A37760C686416581F2

SBOM Export

Export SBOM List — Extract SBOM verify result.

Export SBOM Raw Data — Extract SBOM raw data.

Note

If the module is not found in EZE,
try to set SUPPORT_EXTRACT_MULTI_COMPRESSED_FV=1 in runtime/H2OEZE.ini and reload the BIOS image.

Remaining Issues

Remaining Issues – CVE Matching

- EO 14028 requires unique component names and meaningful versions so that CVEs can be matched against shipping software.
- SDK-based firmware (such as EDK2) means there is heavy customization in the source and libraries.
  - Many components will not share the same hash nor will variant versions share the same version numbers, even if derived from the same source.
- How to meaningfully track security issues back from the binaries to the source?

Remaining Issues - Dependencies

- How to integrate SBOMs from library dependencies into firmware SBOMs?

- Should all library dependencies be included or some (1 layer deep or 3rd party) or none? Or configurable?

Remaining Issues – Partial Updates

- In cases where it is possible to update just part of the firmware (a single FV or a single region or a single component), how to handle SBOM updates?
  - Some storage solutions are chosen to split out those SBOMs for which separate updates are possible (i.e. FSP, IBB/OBB, etc.)

Call to Action

Call to Action

- Make sure you can produce and verify a complete SBOM for 100% of your firmware’s code in SWID, SPDX or CycloneDX format.

- Join the UEFI SBOM Sub-Team to help define the industry path forward.

Questions?

More Questions?

Following today’s webinar, join the live, interactive Microsoft Teams Q&A for the opportunity to chat with the presenters

Visit this link to attend: https://bit.ly/405uaaU
Meeting ID: 221 456 489 213
Password: 4eAm6X

References

- *SBOM at a glance*, NTIA (April 2021), https://www.ntia.gov/sites/default/files/publications/sbom_at_a_glance_apr2021_0.pdf
- Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM), CISA (September 3, 2024), https://www.cisa.gov/sites/default/files/2024-10/SBOM%20Framing%20Software%20Component%20Transparency%202024.pdf
- *The Minimum Elements For a Software Bill of Materials (SBOM) Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity*, NTIA (July 2021), https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf
- *Firmware Embedded SBoM Specification*, https://lvfs.readthedocs.io/en/latest/sbom.html

Thanks for attending a UEFI Forum 2024 Webinar

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

presented by

insyde®