



Framing

NTIA Software Supply Chain
Transparency

January 13, 2021

Framing Working Group

Managed with love and patience by co-chairs Michelle Jump and Art Manion

Meeting almost weekly since July 2018

- Fridays at 1400 EDT
- <https://lists.sei.cmu.edu/mailman/listinfo/ntia-sbom-framing>

Framing concepts that apply to the entire multi-stakeholder process



Michelle Jump

MedSec LLC

Global Regulatory Advisor - Medical Device Cybersecurity

MichelleJump@medsec.com



Art Manion

Software Engineering Institute
CERT Coordination Center

Principle Engineer “/” Technical Manager

amanion@cert.org

Agenda

1. SBOM refresher
2. New draft documents – finalizing comment resolution
 1. Sharing and Exchanging SBOMs
 2. Software Identification Challenge and Guidance
3. Expected upcoming work
 1. “VEX”
 2. Glossary
4. Considerations for Further Work
 1. Beyond baseline
 2. Integrity/Authenticity/Provenance

Refresher: What is an SBOM?

Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)

<https://tinyurl.com/y7s8ab3t>

“An SBOM is effectively a nested inventory, a list of ingredients that make up software components.”

Partial Table of Contents

2 What is an SBOM?

2.2 Baseline Component Information

2.4 Component Relationships

4 SBOM Processes

4.1 SBOM Creation: How

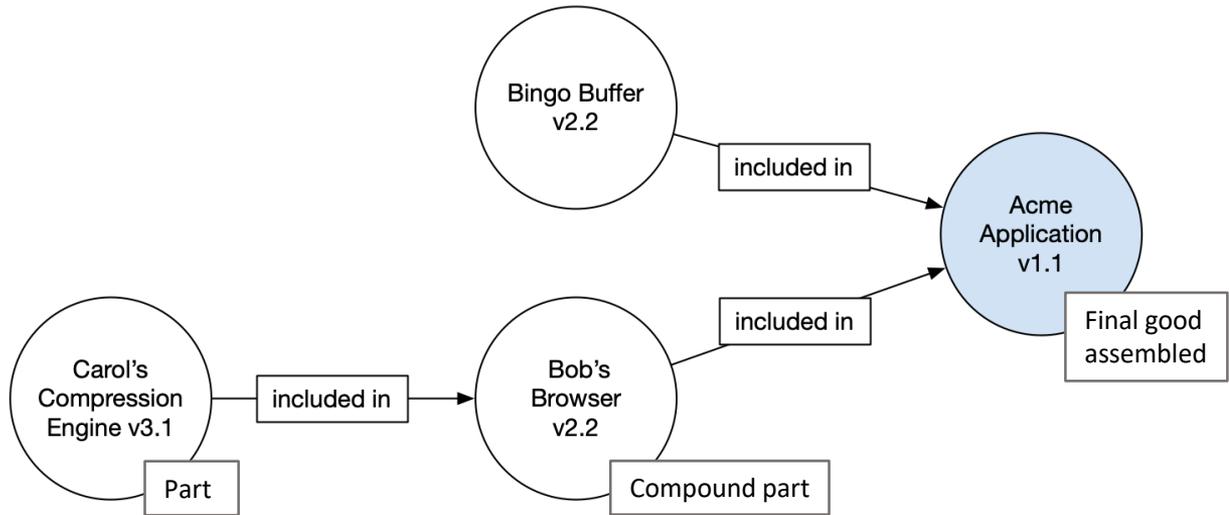
4.2 SBOM Creation: When

4.3 SBOM Exchange

4.4 Network Rules

4.6 Applications of SBOMs

5 Terminology



| Component Name | Supplier Name | Version String | Author | Hash | UID | Relationship |
|------------------------|---------------|----------------|--------|-------|-----|--------------|
| Application | Acme | 1.1 | Acme | 0x123 | 234 | Self |
| --- Browser | Bob | 2.1 | Bob | 0x223 | 334 | Included in |
| --- Compression Engine | Carol | 3.1 | Acme | 0x323 | 434 | Included in |
| --- Buffer | Bingo | 2.2 | Acme | 0x423 | 534 | Included in |



Two Discussion Papers –
Now Released
**Sharing
Identity**

Two New Papers Released

Available at: [SOFTWARE BILL OF MATERIALS | National Telecommunications and Information Administration \(ntia.gov\)](https://www.ntia.gov/SBOM).

1. Sharing and Exchanging SBOMs
2. Software Identification Challenge and Guidance

https://www.ntia.gov/SBOM

National Telecommunications and Information Administration
United States Department of Commerce

Newsroom Publications Blog Offices About Contact

Home

SOFTWARE BILL OF MATERIALS

A "Software Bill of Materials" (SBOM) is effectively a nested inventory, a list of ingredients that make up software components. The following documents were drafted by stakeholders in an **open and transparent process** to address transparency around software components, and were approved by a consensus of participating stakeholders.

Introduction to SBOM

- SBOM at a Glance** is an introduction to the practice of SBOM, supporting literature, and the pivotal role SBOMs play in providing much-needed transparency for the software supply chain.
- The **FAQ document** outlines detailed information, benefits, and commonly asked questions.
- The **Two-Page overview** provides high-level information on SBOM's background and eco-wide solution, the NTIA process, and an example of an SBOM.

[SBOM Explainer Videos on YouTube](#) @

New Documents

- [Software Identity: Challenges and Guidance](#)
This resource reviews the challenges of identifying software components for SBOM implementation with sufficient discoverability and uniqueness. It offers guidance to functionally identify software components in the short term and converge multiple existing identification systems in the near future.
- [SBOM Tool Classification Taxonomy](#)
This resource offers a categorization of different types of SBOM tools. It can help tool creators and vendors to easily classify their work, and can help those who need SBOM tools understand what is available.

Sharing and Exchanging SBOMs

“Transparency in the supply chain enables better risk decision-making for producers and consumers of software. This means that information about the underlying software components in a piece of software—a Software Bill of Material (SBOM)—should be accessible to the right entities at the right time.”

Programmatic end system SBOM sharing

- Goal
 - An IETF standard
 - Discover SBOM, determine its format and an appropriate retrieval mechanism
- Method
 - Provide a simple model
 - Use of existing network discovery functions to access it (MUD, DHCP, certificate attributes)
 - Be content neutral: support SPDX, CycloneDX, and anything else
- Status
 - Adopted by IETF Ops Area Working Group
 - Early code developed
 - More work needed to add “VEX” capability
- <https://tools.ietf.org/html/draft-ietf-opsawg-sbom-access-00>

YANG model:

```
+--rw sboms* [version-info]
  +--rw version-info    string
  +--rw (sbom-type)?
    +--:(url)
      | +--rw sbom-url?  inet:uri
    +--:(local-uri)
      | +--rw sbom-local* enumeration
    +--:(contact-info)
      +--rw contact-uri? inet:uri
```

Software Identification Challenges and Guidance

“Possibly the biggest single challenge to supply-chain transparency and the SBOM model is the difficulty in identifying software components globally... This paper captures some of the major challenges... and offers some guidance on how to address these challenges.”



Ongoing Efforts

Framing: Focus Areas Within Phase II

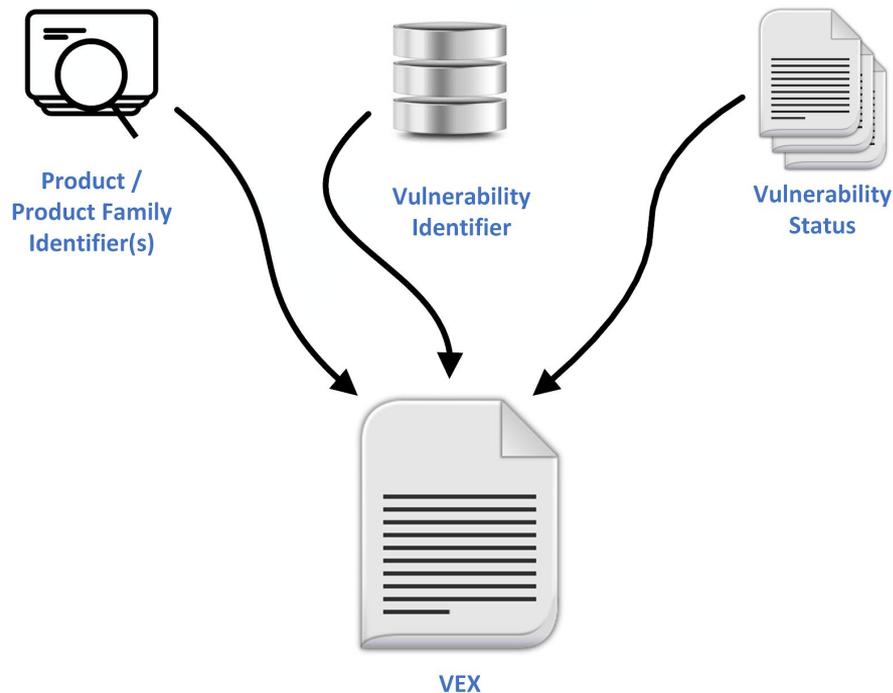
- **Is the baseline accurate and effective?**
 - Reviewing baseline to clarify and detail beyond baseline info
- **VEX: How do we understand the exploitability status of a particular vulnerability?**
 - Vulnerability exploitability status
- **Glossary: Let's all speak the same language**
 - Common set of terms

Framing Group Report: Under Revision

- Baseline:
 - Clarifying from past year of experience
 - NOT significantly changing the baseline
- Beyond the Baseline
 - Providing more context for value of moving beyond the baseline
- Other minor editorial/clarification changes that have been identified since initial release

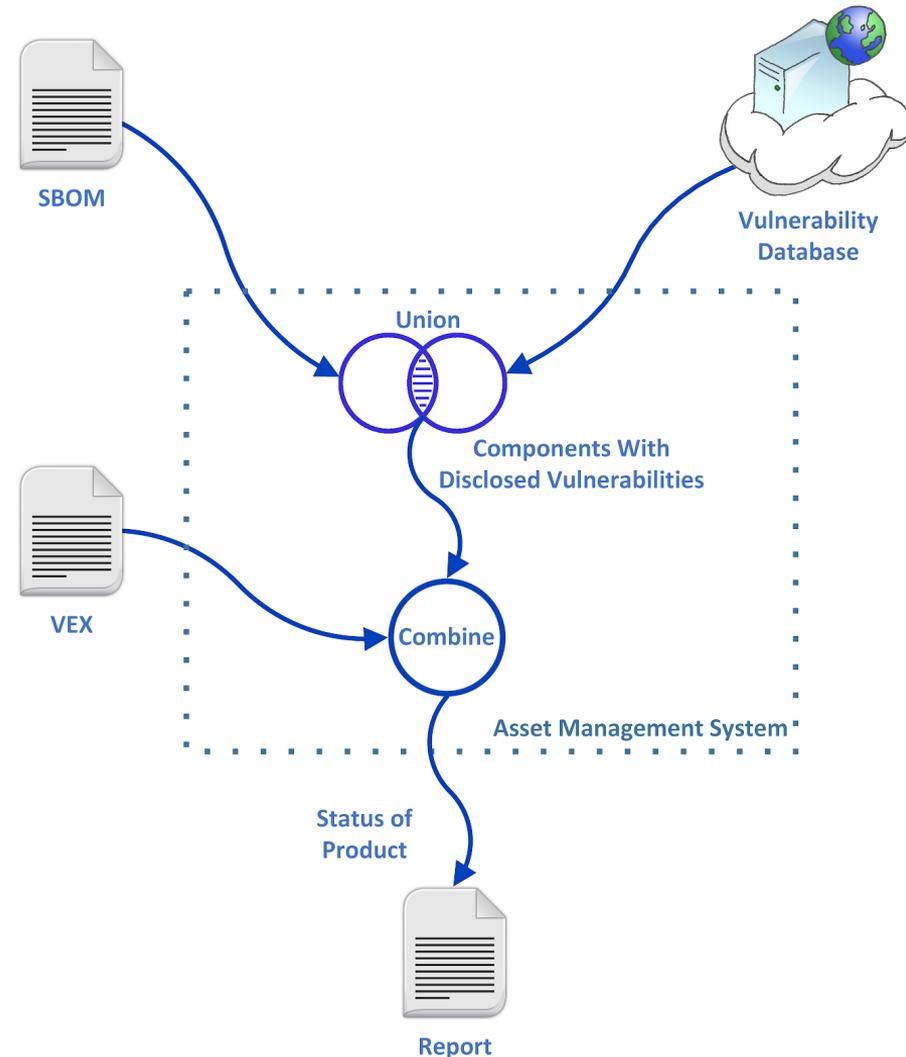
Vulnerability Exploitability eXchange (VEX)

- Is a piece of software (component!) actually affected by a vulnerability?
- Claims about impact from a particular author
 - Can come from the supplier or a third party
- Many but not all use cases associated with SBOM
- Separate from SBOM, but linkable



VEX + SBOM example

- Software includes a vulnerable component
- SW supplier determines that the vuln doesn't affect the built software
 - E.g., relevant code isn't included by compiler
 - E.g., relevant code is present, but not used or exposed
- Supplier issues a VEX with the claim that the component is “not affected” and no action is required
- Consumer integrates SBOM data, vulnerability data, and VEX data to make some risk-based decision



An initial approach to VEX data

| Vex Data Fields | |
|-----------------------------------|---|
| VEX targets | Component identifier(s) or component family identifier |
| VEX metadata | <ul style="list-style-type: none">• Identifier string for the VEX• Author• Author-role• Timestamp• Integrity/Signature/etc• SBOM identifier (optional) |
| For each vulnerability identified | <ul style="list-style-type: none">• Vulnerability identifier• Vulnerability status (Machine readable)• Further vulnerability details• Affected component (optional) |

Link to SBOM

Progress on VEX

- Aligning as a profile for the OASIS Common Security Advisory Framework (CSAF) standard
 - Mapping the minimum fields for CSAF and VEX
 - Hope to include this in the upcoming CSAF version
- Draft definitions of the vulnerability status – based on actions needed
 - `Known_not_affected`
 - No mitigation is required regarding this vulnerability.
 - This could be because the code referenced in the vulnerability is not present, not exposed, compensating controls exist, or other factors. See {{{field x}}} for more details”
 - `Known_affected`
 - Actions are recommended to mitigate or address this vulnerability
 - This could include learning more about the vulnerability and context, and/or making a risk-based decision
 - `Under_Investigation`
 - `Fixed`
- Working with the Healthcare POC to test and align