NTIA Supply Chain Transparency

Framing Working Group

2019-04-11
Stable Topics 1

Required elements, without which there is no SBOM

- Core component identity, sufficient uniqueness
  - Minimum viable identification (MVI)
  - Recommended constructs
    - namespace:name
    - supplier:component:version:hash

- Relationships between components
  - Minimum: “includes”, “alias”?

Optional elements, meta-information needed for most use cases/applications

- Nearly anything you want
  - Yes, really, see “Open Topics 3”
Problem statement [ref]

Terms, partially stable [ref]

- Supplier
- Component
- SBOM
  - overall collection of data and processes
- Inventory
  - core identification, subset of SBOM

When in to produce SBOM, type of SBOM

- Time of build, packaging, delivery, “as-built”
  - Yes, these are technically different
  - Binary/object, not source
  - Change means new component and new SBOM
Depth, one-/multi-hop

- Not either-or, but both
- Supplier creates SBOM for their components
  - Defines components and sufficiently unique names
  - Records dependencies
- Supplier ideally obtains SBOMs for included components from their upstream suppliers
  - One hop upstream required
  - Additional hops optional, but fragile
  - Supplier has first-hand knowledge of what they include and relationship with upstream supplier
  - When not possible, supplier obtains or creates component identifiers
- Supplier delivers collected SBOMs to customer
  - One hop downstream
How are SBOMs shared, exchanged? What does transparency look like?

- Multiple techniques, different types of software and systems
  - Files included with distribution
  - URL, unique ID lookup
  - Atom/ROLIE (RFC 8322), SParts?

SBOM history

- Supplier or consumer can maintain records
- Not relying on any central repository, but not preventing archival
Open Topics

2

Opaqueness, transition

- What happens when SBOM is not available?
- Record differences between components
  - Knowledge that there are no further upstream components/dependencies
  - Lack of such knowledge (opaqueness), component may be a terminal/root node or not

Awareness, adoption, how-to, tools
Common use cases/applications include

- Intellectual property management
  - License, entitlement, copyright, attribution, other
  - Clear terms for “license” and “entitlement”

- High assurance
  - Provenance, pedigree, formulation, integrity

- Vulnerability management
  - Requires a catalog of vulnerabilities, like CVE
  - Requires mapping between vulnerabilities and components
  - Means to convey exposure/exploitability of vulnerabilities

- What else should we call out?
Open Topics

4

Awareness, adoption, how-to, tool support

- Easy start guide for “crawl” stage
- Examples
  - Health care proof of concept
  - SWID, SPDX
  - Existing tools?

Services, not-on-premises components and systems

- Provider/operator wants SBOM like any other user?
- Service user man not care, SBOM may change rapidly (daily/hourly)
- Not prevented, but not primary focus?
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Supplier</strong></td>
<td><strong>Component</strong></td>
<td>Version</td>
<td>Hash</td>
<td>Includes</td>
</tr>
<tr>
<td>2</td>
<td>OpenSSL</td>
<td>OpenSSL</td>
<td>0.9.8a</td>
<td>0x113a8...</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>Apache</td>
<td>httpd</td>
<td>1.3.26</td>
<td>0x33af2...</td>
<td>OpenSSL 0.9.8a</td>
</tr>
<tr>
<td>4</td>
<td>MDM1</td>
<td>FooPump</td>
<td>4.0</td>
<td>0x44a83...</td>
<td>Apache httpd 1.3.26</td>
</tr>
</tbody>
</table>
namespace:

org.openssl:”OpenSSL 0.9.8a”
org.apache:”httpd 1.3.26”
com.mdm1:”FooPump 4.0 0x44a83...”
<SoftwareIdentity name="openssl"
tagId="openssl/openssl@0.9.8a" version="0.9.8a"/>

<SoftwareIdentity name="apache_httpd"
tagId="apache/httpd@1.3.26" version="1.3.26"/>
<Link href="swid:openssl/openssl@0.9.8a" rel="requires"/>

<SoftwareIdentity name="apache_httpd"
tagId="apache/httpd@1.3.26" version="1.3.26"/>
<Link href="swid:openssl/openssl@0.9.8a" rel="requires"/>
PackageName: openssl
SPDXID: openssl/openssl@0.9.8a
PackageVersion: 0.9.8a

PackageName: apache_httpd
SPDXID: apache/httpd@1.3.26
PackageVersion: 1.3.26
Relationship: openssl/openssl@0.9.8a
PREREQUISITE_OF apache/httpd@1.3.26

PackageName: "MDM1 FooPump"
SPDXID: mdm1/foopump@4.0
PackageVersion: 4.0
Relationship: apache/httpd@1.3.26
PREREQUISITE_OF mdm1/foopump@4.0
Graph
Sample of other data

<table>
<thead>
<tr>
<th></th>
<th>SWID</th>
<th>SPDX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hash</strong></td>
<td>hash-entry</td>
<td>PackageVerificationCode</td>
</tr>
<tr>
<td></td>
<td>hash-alg-id</td>
<td>PackageChecksum</td>
</tr>
<tr>
<td></td>
<td>hash-value</td>
<td>FileChecksum</td>
</tr>
<tr>
<td><strong>License</strong></td>
<td></td>
<td>LicenseConcluded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PackageLicenseDeclared</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LicenseName</td>
</tr>
<tr>
<td><strong>Entitlement</strong></td>
<td>@entitlementKey</td>
<td></td>
</tr>
</tbody>
</table>
Required

Component Identity (MVI)
Provenance
Formulation
Expected Usage
Vulnerability Management
Inclusion

Diagram:
- **Component Identity (MVI)**
  - **Created By**
  - **Modified By**
  - **Applies To**
  - **Affects**
  - **Exposes**
  - **Includes**
  - **Built Using**
  - **Normally Operates Like**
- **User**
- **License**
- **Vulnerability**
- **Formulation**
- **Usage Description**
Feature Support
License
Vulnerability Management
High Assurance
Framing WG Logistics

Co-chairs

- Michelle Jump <michelle.jump@novaleah.com>
- Art Manion <amanion@cert.org>

Current meeting schedule

- Weekly Fridays at 2 PM EST

Mailing list

- https://lists.sei.cmu.edu/mailman/listinfo/ntia-sbom-framing

Google Drive

- https://drive.google.com/drive/folders/1vOypGE1gWuKwfnmvLApHJY10NI62cUxH