Tool-toTool Software Bill of Materials Specification

Robert A. Martin
Sr. Secure Software & Technology Prin. Eng.
Trust & Assurance Cyber Technologies Dept.
Cyber Solutions Innovation Center

13 October 2020
Enabling Adoption of SBOMs

SBOMs

- Agriculture and Food
- Energy
- Transportation
- Chemical Industry
- Postal and Shipping

SBOMs

- Medical Devices
- Merchandise
- Automobiles
- Trains
- Vessels/Boats
- Building Mgmt Sys
- Software

SBOMs

- Water
- Public Health
- Telecommunications
- Banking and Finance
- Key Assets

End Users in Industry, Government, and Commerce

Product & Service Suppliers

Integrated Development Environments (IDEs)

Source Code & Package Repositories

Frameworks

Build Choreography

Tools & Capabilities for Software

Software Composition Analysis

Cloud Tools

Lic Mgmt

Tool-to-Tool SBOM Exchange Standard effort (3T-SBOM-ES)

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-22
Ecosystem of SW Development, Integration, and Management Tools
Source Code & Package Repositories
Amazon ECR, Assembla, Azure Container Registry, Beanstalk, Bitbucket, Codebase, Docker, GitHub, GitLab, Glitch, Google Container Registry, JFrog Artifactory, JFrog Xray, inedo, Kubernetes, Launchpad, Maven, Nexus (Sonatype), Phabricator, ProjectLocker, Repository Hosting, Savannah, SourceForge, SourceRepo, Subversion, and Unfuddle

Build & Build Choreography Capabilities
Ansible, Autotab, Bamboo, Bitrise, Buildkite, Buildroot, CircleCI, CMakel, CruiseControl, Final builder, GCC, Gitlab CI, GoCD, Integrity, Jenkins, Strider CD, TeamCity, Terraform, Travis CI, Urbancode, and Vagrant

Developer Desktops (Embedded, Web, Cloud, Desktops/Servers)

Frameworks: .NET, Angular, Ansible, Apache Spark, ASP.NET, Bootstrap, Chef, Cordova, CryEngine, Django, Drupal, Express, Flask, Flutter, Hadoop, HTML5 Builder, Laravel, Node.js, Pandas, Puppet, React Native, React.js, Ruby on Rails, Spring, TensorFlow, Torch/PyTorch, Unity D, Unreal Engine, Visual Online, Vue.js, and Xamarin

Cloud Tools: Azure, AWS CodeBuild, Cloud Foundry, Google Cloud Build, Kwater, Pivotal, and Red Hat

Software Composition Analysis:
Black Duck Software Composition Analysis (Synopsys), CAST Highlight (CAST Software), Finate State, FlexNet Code Insite (Flexera), Ion Channel, Insignary, GrammaTech, SourceClear, Sonatype, Snyk, and WhiteSource
Provenance and Pedigree

**DEFINITIONS**

- **Provenance**
  1. The origin, or source of something
  2. The history of ownership of a valued object, work of art, or literature

- **Pedigree**
  1. A register recording a line of ancestors
  2. An ancestral line: lineage
     - The origin and the history of something; broadly: background, history

**CONFUSION**

Many use “Provenance” for both meanings.

The provenance of a piece of data is both the custody chain of an Artifact, Document or Record. Pedigree captures the history of how an Artifact or Document was produced or derived.

*Definitions (from Merriam-Webster.com)*

Provenance
Captures *chain of custody* of an Artifact, Document or Record

Pedigree
Captures the *history* of how an Artifact or Document was produced or derived
Ensuring we have a **verifiable attestation** of the **origin of all code** running in production so that we can have a **root of trust** as we move forward to defining and enforcing a collection of **policies** throughout the different stages of the **software development process**.
Existing Formats in Use

CycloneDX

A framework to secure the integrity of software supply chains

What are SWID Tags?

There are two styles of software tags:

Software Identification (SWID) Tags
Software identification tags (SWID tags) record unique information, including the name, edition, version, whether it is legacy support software inventory and asset management initiative. They in international standard ISO/IEC 19770-2:2015.

Find out more:
- Software identification tag FAQ

Software Entitlement (SWEN) Tags
Software entitlement specifies how license consumption maps...
Usage Scenarios for Tool-to-Tool SBoM

- **Refer, Transfer or Purchase** (definition of what it is)
- **Pedigree** (history of how it was produced)
- **Provenance** (chain of custody of it)
- **Integrity** (cryptographic basis of unalteredness)
- **Proper and Legal** (conditions about its use)
- **Known Sw Vulns** (known fixes are applied to it)
- **Assurance** (safe-secure-resilient)
- **SBoM of a SW Service** (SBoM of sw delivering service)

Supply Chain Sequence Integrity

Between Tiers

Spanning Tiers

Supplier

Contract/Agreement

Customer

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-22
## Usage Scenarios and Tool-to-Tool SBOM candidate elements

### Usages

1. **Refer, Transfer or Purchase** (definition of what it is)
2. **Pedigree** (history of how it was produced)
3. **Provenance** (chain of custody of it)
4. **Integrity** (cryptographic basis of unalteredness)
5. **Intellectual Property Constraints**
6. **Known SW Vulns** (known fixes are applied to it)
7. **Assurance** (secure-safe-resilient)
8. **SBOM of a SW Service** (SBOM of sw delivering service)
9. **Supply Chain Sequence Integrity**

### SBoM elements

- Author of SBoM
- SBoM population method
- SBoM Time-Stamp
- Supplier
- **Components** (sources, executables, patches)
  - Version
  - Notes
  - Licenses
  - Created Using
  - Created By
  - Item Hash/Signature

### Correlated Info
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**Usages**

1. Refer, Transfer or Purchase
   (definition of what it is)
2. Pedigree
   (history of how it was produced)
3. Provenance
   (chain of custody of it)
4. Integrity
   (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns
   (known fixes are applied to it)
7. Assurance
   (secure-safe-resilient)
8. SBOM of a SW Service
   (SBOM of sw delivering service)
9. Supply Chain Sequence
   Integrity

**SBoM elements**

- Author of SBOM
- SBoM population method
- SBoM Time-Stamap
- Supplier
- Components
  (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

### Usages

1. **Refer, Transfer or Purchase**
   - (definition of what it is)

2. **Pedigree**
   - (history of how it was produced)

3. **Provenance**
   - (chain of custody of it)

4. **Integrity**
   - (cryptographic basis of unalteredness)

5. **Intellectual Property Constraints**

6. **Known SW Vulns**
   - (known fixes are applied to it)

7. **Assurance**
   - (secure-safe-resilient)

8. **SBOM of a SW Service**
   - (SBOM of sw delivering service)

9. **Supply Chain Sequence Integrity**

### SBoM elements

- **Author of SBOM**
- **SBOM population method**
- **SBOM Time-Stamp**
- **Supplier**
- **Components**
  - (sources, executables, patches)
- **Version**
- **Notes**
- **Licenses**
- **Created Using**
- **Created By**
- **Item Hash/Signature**

### Correlated Info

- **None**
Usage Scenarios and Tool-to-Tool SBOM candidate elements

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBOM of a SW Service (SBOM of sw delivering service)
9. Supply Chain Sequence Integrity

- Author of SBOM
- SBOM population method
- SBOM Time-Stemp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**Usages**

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBOM of a SW Service (SBOM of SW delivering service)
9. Supply Chain Sequence Integrity

**SBOM elements**

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**Usages**

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBOM of a SW Service (SBOM of sw delivering service)
9. Supply Chain Sequence Integrity

**SBoM elements**

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

None
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**Usages**

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBOM of a SW Service (SBOM of sw delivering service)
9. Supply Chain Sequence Integrity

**SBOM elements**

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

Vulnerability Knowledge Bases
Vulnerability Management Systems
Usage Scenarios and Tool-to-Tool SBOM candidate elements

**Usages**

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBOM of a SW Service (SBOM of sw delivering service)
9. Supply Chain Sequence Integrity

**SBoM elements**

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

**Correlated Info**

- Notes on exploitability of vulns
- Vulnerability Knowledge Bases
- Weakness Knowledge Bases
- Assessment Results
- Design Review
- Code Review
- Attack Surface Analysis
- Static Analysis
- Dynamic Analysis
- Fuzz Testing
- Pen Testing
- Blue Teaming
- Red Teaming
- Organized as an Assurance Case
Usage Scenarios and Tool-to-Tool SBOM candidate elements

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBOM of a SW Service (SBOM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements

- Author of SBoM
- SBoM population method
- SBoM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

Correlated Info

Logging SBOMs of Services Used

© 2020 The MITRE Corporation. All rights reserved. Approved for Public Release; Distribution Unlimited. Case No: 20-01465-22
Usage Scenarios and Tool-to-Tool SBOM candidate elements

Usages

1. Refer, Transfer or Purchase (definition of what it is)
2. Pedigree (history of how it was produced)
3. Provenance (chain of custody of it)
4. Integrity (cryptographic basis of unalteredness)
5. Intellectual Property Constraints
6. Known SW Vulns (known fixes are applied to it)
7. Assurance (secure-safe-resilient)
8. SBOM of a SW Service (SBOM of sw delivering service)
9. Supply Chain Sequence Integrity

SBoM elements

- Author of SBOM
- SBOM population method
- SBOM Time-Stamp
- Supplier
- Components (sources, executables, patches)
- Version
- Notes
- Licenses
- Created Using
- Created By
- Item Hash/Signature

Correlated Info

Desired sequence of ordered software supply chain steps, and requirements for each step for a specific project of interest
Launched 24 Sep 2019

CHAIRS

- Bob Martin (MITRE)
- Dr. Bill Curtis (CISQ)
- Kay Williams (Microsoft - Azure - CD Foundation)

PARTICIPANTS

- Ken Moste (UL)
- Philippe-Emmanuel Douziech (CAST)
- Santiago Torres-Arias (Illinois NYU)
- Diakhone Godeen (Blackberry)
- William Cox (Black Duck by Synopsys)
- Steve Lasker (Microsoft - Artifact Store, Open Container Initiative)
- Brian Russell (Google - CD Foundation)
- Nitish Bakliwal (Microsoft - Windows)
- Kate Stewart (Linux Foundation - SPDX)
- William Bartholomew (GitHub)
- David Edelsohn (IBM - GCC)
- Jason Shaver (Microsoft - Developer Division)
- Adam Baldwin (npm, GitHub)
- Thomas Stemmerger (HERE - OSS Review Toolkit)
- Sean Bornum (MITRE)
- Charles Schmitt (MITRE)
- Kris Pandit (Microsoft - Windows)
- Michael Richardson (Sandelman Software Works Inc.)
- Srivath Poduri (Microsoft - Supply Chains)
- Alexander Stein (SSA)
- Deep Datta (Frog)

- Vijay Chari (Rockmeter/IDC - Medical Devices)
- Allan Friedman (NTIA)
- Gerald Heidebreich (Microsoft - CloudBuild)
- Dan Lorenz (Google - CD Foundation)
- Jeffrey Martin (White Source Software)
- Michael Muehr (CAST)
- Bryan Sullivan (Microsoft - Security and Compliance)
- Brian Fox (Sonatype)
- Fred Blais (CloudBees - Jenkins)
- Mark Gabin (Frog)
- Gary O’Neall (Source Auditor - SPDX)
- Anna Deberitam (Snyk)
- Duncan Sporrell (SFractl Consulting)
- Philippe Ombredanne (nexB - ScanCode)
- Solomon Rubin (FOSSA)
- Adrian Digio (Microsoft - Engineering Systems)
- Steve Wilson (Linux Foundation)
- Henk Birkholz (Fraunhofer ST - IETF PATS and RIIM)
- Paul Anderson (GrammaTech)
- Gareth Rushgrove (Snyk)
- Shelly Waite-Bey (Waite SLTS - Cybersecurity, Compliance and AI)
Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)

NTIA Multistakeholder Process on Software Component Transparency
Framing Working Group
2019-11-12

Table of Contents

Table of Contents 2

1 Problem Statement 4
  1.1 Goals 4

2 What is an SBOM? 6
  2.1 SBOM Elements 7
  2.2 Baseline Component Information 7
    2.2.1 Author Name 7
    2.2.2 Supplier Name 7
    2.2.3 Component Name 8
    2.2.4 Version String 8
    2.2.5 Component Hash 8
    2.2.6 Unique Identifier 8
    2.2.7 Relationship 8

3 Merging Existing Exports 9
<table>
<thead>
<tr>
<th>Name</th>
<th>Owner</th>
<th>Last modified</th>
<th>File size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subgroups</td>
<td>William Bartholomew</td>
<td>May 8, 2020</td>
<td></td>
</tr>
<tr>
<td>Misc</td>
<td>Kay Williams</td>
<td>Feb 3, 2020</td>
<td></td>
</tr>
<tr>
<td>Face to Face Meetings</td>
<td>Kay Williams</td>
<td>Dec 6, 2019</td>
<td></td>
</tr>
<tr>
<td>Modeling</td>
<td>Philippe Emmanuel</td>
<td>Nov 25, 2019</td>
<td></td>
</tr>
<tr>
<td>Meeting Recordings</td>
<td>William Bartholomew</td>
<td>Oct 30, 2019</td>
<td></td>
</tr>
<tr>
<td>User Scenarios</td>
<td>William Bartholomew</td>
<td>Oct 21, 2019</td>
<td></td>
</tr>
<tr>
<td>Specification Drafts</td>
<td>me</td>
<td>Oct 15, 2019</td>
<td></td>
</tr>
<tr>
<td>Whitepaper Versions</td>
<td>me</td>
<td>Sep 29, 2019</td>
<td></td>
</tr>
<tr>
<td>Weekly Meeting Agenda and</td>
<td>Kay Williams</td>
<td>Sep 16, 2020</td>
<td></td>
</tr>
<tr>
<td>Working Group Members</td>
<td>Kay Williams</td>
<td>Sep 15, 2020</td>
<td></td>
</tr>
</tbody>
</table>
3T SBOM Exchange Standard Metamodel
3T SBOM Exchange Standard Model

Seeded from CycloneDX + SPDX

HashAlgorithm: String
SHA1
SHA224
SHA256
SHA384
SHA512
SHA3-224
SHA3-256
SHA3-384
SHA3-512
MD2
MD4
MD5
MD6
SPDX-PVC
BLAKE2b-256
BLAKE2b-384
BLAKE2b-512
BLAKE3

Seeded from CycloneDX (minus FILE plus SOURCE)

PackageType: String
APPLICATION
FRAMEWORK
LIBRARY
CONTAINER
OPERATING-SYSTEM
DEVICE
FIRMWARE
SOURCE

Seeded from SPDX (SPDX->BOM)

FileType: String
SOURCE
ARCHIVE
APPLICATION
AUDIO
IMAGE
TEXT
VIDEO
DOCUMENTATION
BOM
OTHER

OnlineService
+ field: type
+ field: type
+ field: type

API
+ field: type
+ field: type
+ field: type

Dataset
+ field: type
+ field: type
+ field: type
Lowering Adoption Hurdles for SBOMs and more

Tool-to-Tool SBOM Exchange Standard (3T-SBOM-ES) effort
Whitepaper ➔ CISQ ➔ OMG RFC ➔ ISO Std

• Socialize at March 2019 OMG meeting
• Draft SBoM as a Whitepaper in 3-day CISQ SBoM working session at September 2019 OMG meeting
• Prototype draft 3T-SBOM format in tool ecosystem, revise and draft RFC based on prototype results
• Co-submit draft RFC to OMG at Dec 2020 or Mar 2021 meeting
• Mar or Jun 2021 OMG meeting – charter FTF
• Jun or Sep 2021 OMG meeting – approve as OMG Standard
• Sep or Dec 2021 Fast Track to ISO
3T-SBOM effort contact: ramartin@mitre.org