Source: Pierre Metivier (https://flic.kr/p/rdHr) CC BY-NC / Desaturated from the original

NTIA Software Component Transparency
July 9, 2020

Formats & Tooling Workgroup

JC Herz
Kate Stewart

Agenda

- Workgroup Goals
- Examples of Formats in Use
  - Populating the Example repos
- Tooling Taxonomy
  - Sufficient? Concept of Operation? Map to PoC
- Tooling Information being collected per Format
  - Guidelines for tools to be added to Ecosystem Page
  - Moving from docs → github
- Future Directions
  - Evolving to a Playbook for different contexts (use cases).
    - Legacy system, modernize, new devops, ….
- Feedback Requests

Formats and Tooling Workgroup Goal

Wrapping up from phase I, we identified for the need for:

- Tooling
  - Documenting tooling
  - Identifying tooling gaps
  - Documenting processes
  - Turnkey universal translation tools

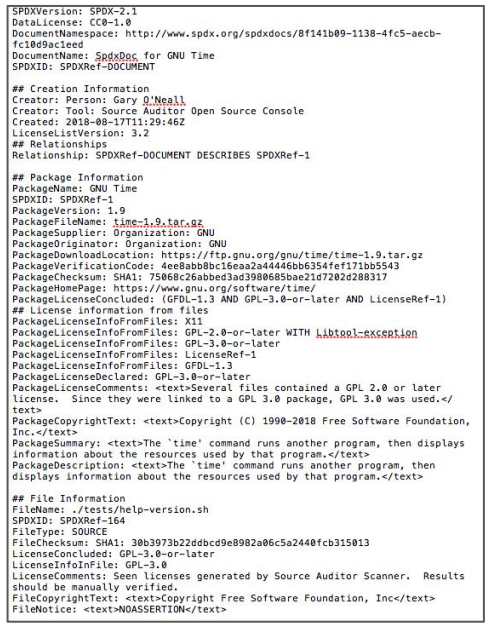
Formats and Tooling workgroup is focusing on addressing these items.

Examples of Formats

SWID

CycloneDX

SPDX



Phase II - Test Corpus

Are there examples that community needs?
How much rigor?

Taxonomy used for Classifying Tools

Category	Type	Description
Author during Build	Build	Document is automatically created as part of building an artifact and contains information about the build.
Author after Creation	Manual	A person will manually fill in the information
	Audit Tool	A source code analysis or audit tool will generate the document by inspection of the artifact and any associated sources.
Consume	View	Be able to understand the contents in human readable form (picture, figures, tables, text.). Use to support decision making & business processes.
	Diff	Be able to compare two documents of a given formation and clearly see the differences. For instance, comparing between two versions of a piece of software.
	Analyze	Be able to import a document into your system
Transform	Translate	Change from one file type to another file type while preserving the same information.
	Merge	Multiple sources of documents can be merged together for analysis and audit puproses
	Tool integration	Support use in other tools by APIs, libraries.

Information to Collect per Tool

Tool Template

Support	Author during Build, Author after Creation, Consume, Transform
Functionality	
Location	Website: Source:
Installation instructions	
How to use	
Versions Supported	

Example: FOSSology

Support	Author after Creation (Audit tool, Manual), Consume(View,Diff,Analyze), Transform(Translate, Merge, Tool Integration)
Functionality	FOSSology is an open source license compliance software system and toolkit allowing users to run license, copyright and export control scans from a REST API. As a system, a database and web UI are provided to provide a compliance workflow. As part of the toolkit multiple license scanners, copyright and export scanners are tools available to help with compliance activities.
Location	Website: https://www.fossology.org/ Source: https://github.com/fossology
Installation instructions	https://www.fossology.org/get-started/
How to use	https://www.fossology.org/get-started/basic-workflow/
Versions Supported:	SPDX 2.1, SPDX 2.2 (WIP)

Tooling Surveys to be Reviewed:

SWID

SPDX

CycloneDX

Proposed Criteria for adding Tools to Ecosystem Documents

- Documents are open to all for comments.
- At least one of the formats needs to be supported
- Produce, Consume or both. Need to provide evidence, such as:
  - Open source - show options in code (including intention)
  - Tools that produce - show valid format documents in format
  - Tools that consume - show evidence of a test ingestion (from suite)
  - There will be a lightweight test suite of representative files for demonstration & discussion, that can be used to compare against.
- Fill in the template and when evidence is shown to document owner they approve.
- Contact document curators if questions, follow up, etc.
  - SWID: Charles Schmidt <cmschmidt@mitre.org>
  - SPDX: Kate Stewart <kstewart@linuxfoundation.org>
  - Cyclone DX: Steve Springett <steve.springett@owasp.org>

Status of Translation Tools

Decoder Ring

- The end goal of the decoder ring project is to be able to easily and automatically translate between all SBOM formats, keeping as much nuance and precision as possible between formats
- Code: https://github.com/DanBeard/DecoderRing
- What's working: Basic N-level SBOM in: SPDX v2.1, SWID (with modification)

CERTCC/SBOM - aka "Vijay's Tool"

- The SBOM demo tool was created to pursue use case of building multi-lingual basic SBOM from manual entry to support effort such as the SBOM PoC effort.
- Code: https://github.com/CERTCC/SBOM/tree/master/sbom-demo
- What's working: Demo available at: https://sbom.democert.org/sbom, SPDX v2.1, SWID (with mods), CycloneDX 1.2

Next Area: Tools in Context, Playbook/CONOPs

- Next: Concepts of Operation (CONOPS) for how they can be used
  - Generation and Consumption
  - Different Use Cases
    - Software Lifecycle Management
    - Entitlements
    - Vulnerability Management
  - Different Roles in the Supply Chain
    - Third Party Supplier (OSS, Commercial Software)
    - Integrator
    - First-party Developer (Internal Enterprise DevOps)
    - Procurement
    - Compliance (interface with external certifiers, regulators, insurers)

Next Steps

Priorities for next steps?

- Continue to collect tools
  - Know a tool to be added to each ecosystem document? Put a comment in the document, so it can be added.
  - SWID: http://tiny.cc/SWID
  - SPDX: http://tiny.cc/SPDX
  - CycloneDX: http://tiny.cc/CycloneDX
- Population of Examples in Phase II - Test Corpus
- Working on Playbooks
- Collaboration with other health care PoC, other use cases & framing

Volunteers interested on working on above areas?
Feedback on proposed approach?

More Info...

Mailing List: ntia-sbom-formats@linuxfoundation.org

Subscribe at: https://lists.linuxfoundation.org/mailman/listinfo/ntia-sbom-formats

Shared Drive:
https://drive.google.com/drive/folders/1KAQ7AWlWMKcSFnRc_S-7XB76xFRRWLmT

Meetings: Every 2 weeks, next meeting scheduled for July 17 at 11am EST. Contact leads to be added to meeting invite.