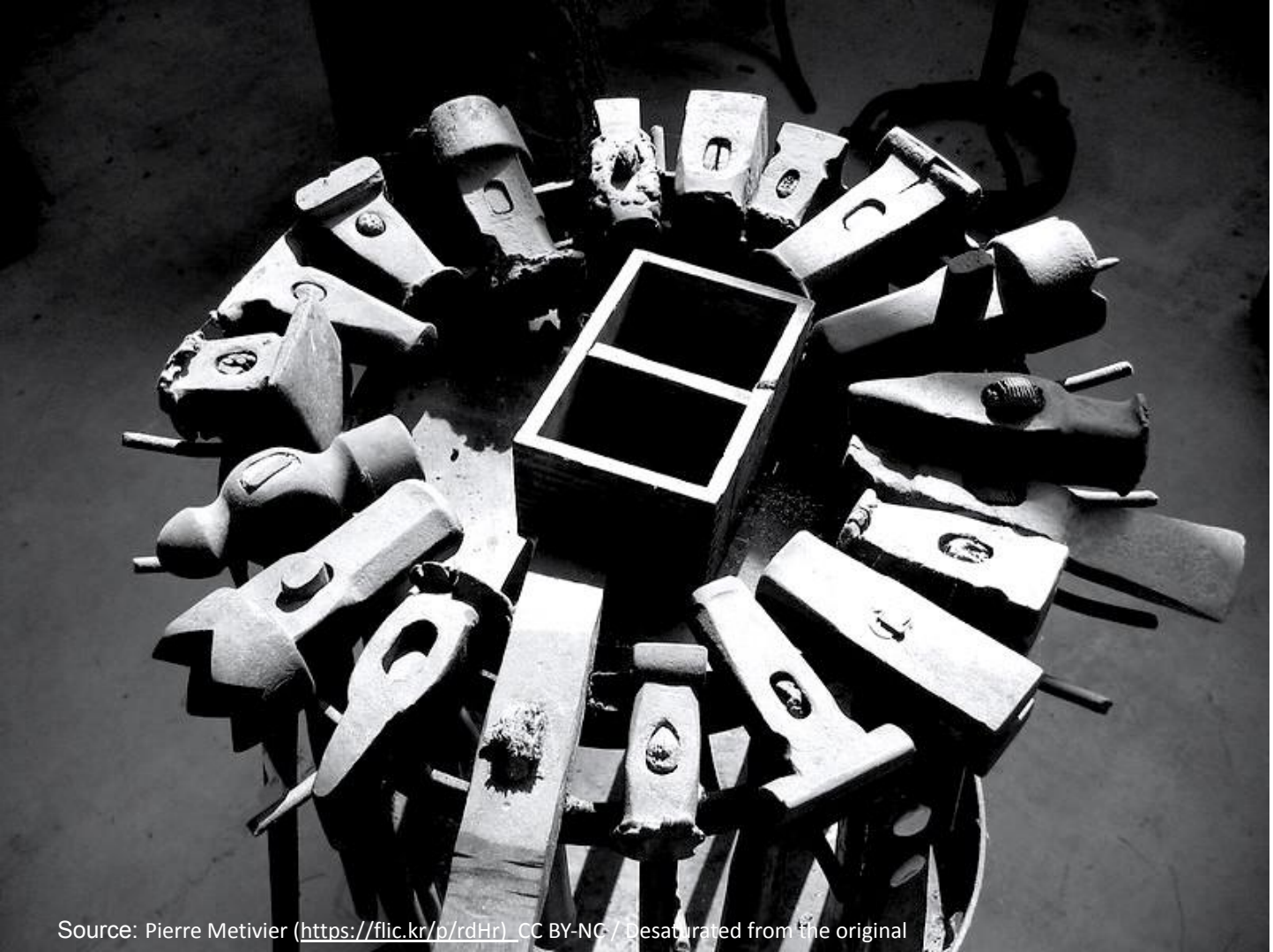

Source: Pierre Metivier (https://flic.kr/p/rdHr) CC BY-NC / Desaturated from the original

NTIA Software Component Transparency
April 29, 2021

Formats & Tooling

Workgroup

Formats & Tooling Working Group

Co-chairs: JC Herz & Kate Stewart

Meeting biweekly since July 2018

- Fridays at 1100 EDT
- https://lists.linuxfoundation.org/mailman/listinfo/ntia-sbom-formats

Adoption of concepts from framing into existing formats and tools that work with those formats.

JC Herz

Ion Channel

COO

jc.herz@ionchannel.io

Kate Stewart

Linux Foundation

VP of Dependable Embedded Systems

stewart@linux.com

Agenda

- Workgroup Goals
- Recap of Formats in Use
  - Updated Formats White Paper
- Taxonomy update
- Playbooks
  - Consumer Playbook Overview
  - Supplier Playbook Overview
- First Plugfest feedback
- Future Directions
- Feedback Requests

Formats and Tooling Workgroup Goal

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

- Tooling
  - Documenting tooling
  - Identifying tooling gaps ← Plugfest starting to highlight
  - Documenting processes ← Playbooks starting to address
  - Turnkey universal translation tools

Formats and Tooling workgroup is focusing on addressing these items.

Formats and Tooling: Objectives

Identify SBOM Formats in Commercial Use

- SPDX - https://spdx.github.io/spdx-spec/
- CycloneDX - https://cyclonedx.org/docs/latest
- SWID - ISO/IEC 19770-2:2015

Identify Software Identifiers in Commercial Use and Emerging Identifiers

- Common Platform Enumeration - CPE
- Package URLs - PURL
- Software ID tags - SWID tag
- Software Heritage persistant ID - SWHID

Formats and Tooling: Objectives

- Define and categorize criteria for the minimum required information in an SBOM from Framing Definitions
  - Field definitions
  - Data extensions for provision of additional/external/deeper information
- Enable translation between SBOM formats
  - See emerging tooling & taxonomy
- Create Playbooks for Generation and Consumption of SBOM
  - Supplier Playbook - draft release: https://docs.google.com/document/d/16FwpPX3P0Pd1D82Dp2VmpRnaMWUA-wvfXbL7AIXDthM/edit
  - Consumer Playbook - draft release: https://docs.google.com/document/d/1Ae0l1MDS8m1on58e8mdVIA9NujzPD0k5j352VlDZr9I/edit

++ Create a reference Corpus of example SBOMs

- Plugfest to identify Gaps
- Agree on reference examples to use in each format for others to consult.

What should a minimum viable SBOM contain?

NTIA SBOM Minimum Fields	SPDX	CycloneDX	SWID
Supplier Name	(3.5) PackageSupplier:	publisher	<Entity> @role (softwareCreator/publisher), @name
Component Name	(3.1) PackageName:	name	<softwareIdentity> @name
Unique Identifier	(3.2) SPDXID:	bom/serialNumber and component/bom-ref	<softwareIdentity> @tagID
Version String	(3.3) PackageVersion:	version	<softwareIdentity> @version
Component Hash	(3.10) PackageChecksum:	hash	<Payload>/../<File> @[hash-algorithm]:hash
Relationship	(7.1) Relationship: CONTAINS	(Nested assembly/subassembly and/or dependency graphs)	<Link> @rel, @href
Author Name	(2.8) Creator:	bom-descriptor:metadata/manufacture/contact	<Entity> @role (tagCreator), @name

Source: NTIA's **Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)**

Translating between SBOM Formats & File Types

SwiftBOM: (SPDX(.spdx), SWID(.xml), CycloneDX(.xml,.json))

- Demo at: https://democert.org/sbom/
- Source code at: https://github.com/CERTCC/SBOM/tree/master/sbom-demo

DecoderRing: (SPDX (.spdx), SWID(.xml))

- Source code at: https://github.com/DanBeard/DecoderRIng

SPDX tools: (SPDX (.spdx, .json, .yaml, .rdf, .xml, .xls))

- Demo at: https://tools.spdx.org/app/
- Source code at: https://github.com/spdx/spdx-online-tools

CycloneDX CLI: (CycloneDX (.xml, .json), SPDX(.spdx))

- Source code at: https://github.com/CycloneDX/cyclonedx-cli

Updated: Taxonomy for Classifying SBOM Tools

Category	Type	Description
Produce	Build	SBOM is automatically created as part of building a software artifact and contains information about the build
	Analyze	Analysis of source or binary files will generate the SBOM by inspection of the artifacts and any associated sources
	Edit	A tool to assist a person manually entering or editing SBOM data
Consume	View	Be able to understand the contents in human readable form (e.g. picture, figures, tables, text.). Use to support decision making & business processes
	Diff	Be able to compare multiple SBOMs and clearly see the differences (e.g. comparing two versions of a piece of software)
	Import	Be able to discover, retrieve, and import an SBOM into your system for further processing and analysis
Transform	Translate	Change from one file type to another file type while preserving the same information
	Merge	Multiple sources of SBOM and other data can be combined together for analysis and audit purposes
	Tool support	Support use in other tools by APIs, object models, libraries, transport, or other reference sources

More details in: https://www.ntia.gov/files/ntia/publications/ntia_sbom_tooling_taxonomy-2021mar30.pdf

Next steps - tool collections

- Google docs today already list tools in the three formats
  - **SWID:** http://tiny.cc/SWID
  - **SPDX:** http://tiny.cc/SPDX
  - **CycloneDX:** http://tiny.cc/CycloneDX
- Plan to shift this over to GitHub to allow a more open process.
- Working Group will develop some transparent governance rules
  - Allow anyone to submit tools
  - Have some clear assertions for tools to get on the list

Playbooks for using “Tools in Operation”

- 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)

SBOM Playbook: Consumer Playbook

- Acquisition of SBOM from supplier
- SBOM Ingestion and Parsing
- Software Entity Resolution
- Data Flows into Third Party Processes and Platforms
  - Configuration Management Database
  - Security Operations Center
  - Software Asset Management System
  - Supply Chain Risk and Vendor Management
- Ongoing Monitoring
- IP and Confidentiality Status of SBOMs and Underlying Data

SBOM Playbooks: Supplier Playbook

- Supplier definition includes: commercial vendor, contract developer, open source software supplier developing and maintaining OSS code.
- SBOM production workflow: development pipeline vs. legacy processes
- SBOM scope: What's in the Box
  - Areas of consensus: single application and its compiled dependencies
  - Still in discussion: external services (SBOM formats can do this)
  - Need for clarity about SBOM coverage: runtime dependencies, container contents
  - As long as extent of coverage is clear (i.e. fields present with "no attestation"), level of detail will ultimately be negotiated between supplier and consumer
- Validation of SBOMs (formats)
- Verification of Components
- Provision of SBOMs to recipients
  - Reference to NTIA Framing Group report: https://www.ntia.doc.gov/files/ntia/publications/ntia_sbom_framing_sharing_july9.pdf
  - IP Status of SBOMs

Plugfest

Plugfest used common set of projects to generate source and binary SBOMs for in the different formats for the same example from different tools, to aid compare & contrast. “Sweat equity” from 11 producers and 6 consumers.

- First Plugfest:
  - Time v1.9 (a small package being used in most OSes)
  - node-express-realworld-example-app (Small example node.js application)
  - zephyr-v2.5.0/samples/hello_world (hello world for embedded)
  - blinky.ex (blinky for embedded)
- Second Plugfest:
  - Same set + a couple of binaries (possibly a container)

Organization	Produce /Consume	Formats	Examples Tried
aDolus - FACT	Produce	SPDX, SWID	Time, Node, Zephyr, Blinky
Copado/New Context	Produce	SPDX, CycloneDX	Node
CyBeats	Produce + Consume	SPDX, CycloneDX	Time, Node, Zephyr
Fortress	Produce	CycloneDX	Time, Node, Zephyr, Blinky
FOSSology	Produce	SPDX	Time
LLNL longclaw	Produce	SPDX	Time, Zephyr
ORT + Scancode	Produce	SPDX, CycloneDX	Time, Node, Zephyr
sFractal	Produce	SPDX, CycloneDX, SWID	Blinky
Source Auditor	Produce	SPDX	Time, Node
Synopsys Black Duck	Produce	SPDX	Time, Node, Zephyr, Blinky
Wind River	Produce	SPDX	Time

Organization	Produce /Consume	Formats	Examples Tried
Ion Channel	Consume	SPDX	Node
NYP	Consume	SPDX	Node, Time, Blinky, Zephyr
NSA	Consume	SPDX	-- (did Schema analysis)
SW 360	Consume	SPDX	Time, Node, Zephyr, Blinky
binaire.io	Consume	SPDX, CycloneDX	Time, Node, Zephyr, Blinky

Summary:

- Participants found it useful and wanted to have follow on, with same samples plus binaries.
- Flagged the importance of having some set of naming conventions or practices for SBOMs themselves.
- Participants also discussed challenges around packages vs. files.
- The community seemed to gravitate to JSON for encoding.
- Bugs got fixed in the tools as a result of interactions, and pairwise partnering between producers and consumers were discussed.
- Next Plugfest planned for June, aligning with OASIS OpenC2 event.

Next Steps

- Update 2019 White Paper
- Continue to collect tools. 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
- Continue population of examples via Plugfests and Comparison
  - Reference corpus of examples illustrated with each format and encodings
  - Planning underway for next Plugfest in June, aiming to add more binaries as starting points.
- Finalize Playbooks
  - **Consumer Playbook Draft:** https://docs.google.com/document/d/1Ae0l1MDS8m1on58e8mdVIA9NujzPD0k5j352VlDZr9I/edit
  - **Supplier Playbook Draft:** https://docs.google.com/document/d/1Ae0l1MDS8m1on58e8mdVIA9NujzPD0k5j352VlDZr9I/edit
- Collaboration with medical and any new PoCs, provide feedback of gaps to framing
- Document formats and tooling for VEX

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

More Info...

Meetings: Weekly, next meeting scheduled for **May 7 at 11am EST.** Contact leads to be added to meeting invite

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

Backup Material

SBOMs Examples (Work in Progress)

SPDX

- https://github.com/lfscanning - LF projects source packages.
- https://github.com/swinslow/spdx-examples - source & binary examples

CycloneDX

- https://github.com/CycloneDX/sbom-examples - binary examples

SWID

- Time 1.9 from Red Hat distro - binary example

Current SBOM Options Available

SPDX


```
SPDXVersion: SPDX-2.1
DataLicense: CC0-1.0
DocumentNamespace: http://www.spdx.org/spdxdocs/8f141b09-1138-4fc5-aecb-
fc10d9ac1eed
DocumentName: SpdxDoc for GNU Time
SPDXID: SPDXRef-DOCUMENT

## Creation Information
Creator: Person: Gary O'Neall
Creator: Tool: Source Auditor Open Source Console
Created: 2018-08-17T11:29:46Z
LicenseListVersion: 3.2
## Relationships
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-1

## Package Information
PackageName: GNU Time
SPDXID: SPDXRef-1
PackageVersion: 1.9
PackageFileName: time-1.9.tar.gz
PackageSupplier: Organization: GNU
PackageOriginator: Organization: GNU
PackageDownloadLocation: https://ftp.gnu.org/gnu/time/time-1.9.tar.gz
PackageVerificationCode: 4ee8abb8bc16eaa2a44446bb6354fef171bb5543
PackageChecksum: SHA1: 75068c26abbed3ad3980685bae21d7202d288317
PackageHomePage: https://www.gnu.org/software/time/
PackageLicenseConcluded: (GFDL-1.3 AND GPL-3.0-or-later AND LicenseRef-1)
## License information from files
PackageLicenseInfoFromFiles: X11
PackageLicenseInfoFromFiles: GPL-2.0-or-later WITH Libtool-exception
PackageLicenseInfoFromFiles: GPL-3.0-or-later
PackageLicenseInfoFromFiles: LicenseRef-1
PackageLicenseInfoFromFiles: GFDL-1.3
PackageLicenseDeclared: GPL-3.0-or-later
PackageLicenseComments: <text>Several files contained a GPL 2.0 or later
license.  Since they were linked to a GPL 3.0 package, GPL 3.0 was used.</
text>
PackageCopyrightText: <text>Copyright (C) 1990-2018 Free Software Foundation,
Inc.</text>
PackageSummary: <text>The `time' command runs another program, then displays
information about the resources used by that program.</text>
PackageDescription: <text>The `time' command runs another program, then
displays information about the resources used by that program.</text>

## File Information
FileName: ./tests/help-version.sh
SPDXID: SPDXRef-164
FileType: SOURCE
FileChecksum: SHA1: 30b3973b22ddbcd9e8982a06c5a2440fcb315013
LicenseConcluded: GPL-3.0-or-later
LicenseInfoInFile: GPL-3.0
LicenseComments: Seen licenses generated by Source Auditor Scanner.  Results
should be manually verified.
FileCopyrightText: <text>Copyright Free Software Foundation, Inc</text>
FileNotice: <text>NOASSERTION</text>
```


File formats: .xls, .spdx, .rdf, .json, .yml, .xml

CycloneDX


```
<?xml version="1.0"?><bom serialNumber="9e253f92-4e1c-497e-8f87-50730d24f18a" xmlns="http://cyclonedx.org/schema/bom/1.1">
<components><component type="library"><description>Nerves System BR - Buildroot based build platform for Nerves
Systems</description><hashes><hash alg="SHA-256">e3fda6bc49f8e3662d37355aad88c0839296597c0b6f6653d21967db1890b038</hash></hashes>
<licenses><license><id>Apache-2.0</id></license><license><name>GPLv2</name></license></licenses><name>nerves_system_br</name>
<purl>pkg:hex/nerves_system_br@1.9.5</purl><version>1.9.5</version></component><component type="library"><description>Nerves -
Create firmware for embedded devices like Raspberry Pi, BeagleBone Black, and more</description><hashes><hash alg="SHA-
256">07079342db3a03d19694118a93f220359fbd94b6e174b98d1ea2709db9e81da9</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>nerves</name><purl>pkg:hex/nerves@1.5.1</purl><version>1.5.1</version></component><component
type="library"><description>Socket handling library for Elixir</description><hashes><hash alg="SHA-
256">98a2ab20ce17f95fb512c5cadddba32b57273e0d2dba2d2e5f976c5969d0c632</hash></hashes><licenses><license><id>WTFPL</id></license>
</licenses><name>socket</name><purl>pkg:hex/socket@0.3.13</purl><version>0.3.13</version></component><component type="library">
<description>Read and write to U-Boot environment blocks</description><hashes><hash alg="SHA-
256">b01e3ec0973e99473234f27839e29e63b5b81eba6a136a18a78d049d4813d6c5</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>uboot_env</name><purl>pkg:hex/uboot_env@0.1.1</purl><version>0.1.1</version></component><component
type="library"><description>Nerves Toolchain CTNG - Toolchain Platform</description><hashes><hash alg="SHA-
256">452f8589c1a58ac787477caab20a8cfc6671e345837ccc19beefe49ae35ba983</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>nerves_toolchain_ctng</name><purl>pkg:hex/nerves_toolchain_ctng@1.6.0</purl><version>1.6.0</version>
</component><component type="library"><description>A ring buffer backend for Elixir Logger with IO streaming.</description><hashes>
<hash alg="SHA-256">b1baddc269099b2afe2ea3a87b8e2b71e57331c0000038ae55090068aac679db</hash></hashes><licenses><license><id>Apache-
2.0</id></license></licenses><name>ring_logger</name><purl>pkg:hex/ring_logger@0.8.0</purl><version>0.8.0</version></component>
<component type="library"><description>Nerves System Linter - Lint Nerves System Defconfigs.</description><hashes><hash alg="SHA-
256">84e0f63c8ac196b16b77608bbe7df66dcf352845c4e4fb394bffd2b572025413</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>nerves_system_linter</name><purl>pkg:hex/nerves_system_linter@0.3.0</purl><version>0.3.0</version>
</component><component type="library"><description>DNS library for Elixir using `inet_dns` module.

Note: The `inet_dns` module is considered internal to Erlang and subject to
change. If this happened this library will be updated.</description><hashes><hash alg="SHA-
256">81c46d39f7934f0e73368355126e4266762cf227ba61d5889635d83b2d64a493</hash></hashes><licenses><license><name>BSD-3-Clauses</name>
</license></licenses><name>dns</name><purl>pkg:hex/dns@2.1.2</purl><version>2.1.2</version></component><component type="library">
<description>A simple mDNS (zeroconf, bonjour) server and client for device discovery on your local network.</description><hashes>
<hash alg="SHA-256">f08414daf5636bf5cd364611e838818e9250c91a3282a817ad9174b03e757401</hash></hashes><licenses><license><id>Apache-
2.0</id></license></licenses><name>mdns</name><purl>pkg:hex/mdns@1.0.3</purl><version>1.0.3</version></component><component
type="library"><description>Nerves Toolchain - arm-unknown-linux-gnueabihf</description><hashes><hash alg="SHA-
256">ba48ce7c846ee12dfca8148dc7240988d96a3f2eb9c234bf08bffe4f0f7a3c62</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>nerves_toolchain_arm_unknown_linux_gnueabihf</name>
<purl>pkg:hex/nerves_toolchain_arm_unknown_linux_gnueabihf@1.2.0</purl><version>1.2.0</version></component><component
type="library"><description>Nerves System - Raspberry Pi 3 A+</description><hashes><hash alg="SHA-
256">c75c5a4c42a2aed5a4ed8116c143993500d1546edb5550978f528e980a7d5f03</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>nerves_system_rpi3a</name><purl>pkg:hex/nerves_system_rpi3a@1.9.2</purl><version>1.9.2</version>
</component><component type="library"><description>Atomic nested term storage and dispatch registry</description><hashes><hash
alg="SHA-256">df791dc276652fcfb53be4dab823e05f8269b96ac57c26f86a67838dbc0eefe7</hash></hashes><licenses><license><id>Apache-
2.0</id></license></licenses><name>system_registry</name><purl>pkg:hex/system_registry@0.8.2</purl><version>0.8.2</version>
</component><component type="library"><description>Nerves Toolchain - armv6-rpi-linux-gnueabi</description><hashes><hash alg="SHA-
256">007668c7ad1f73bad8fd54ad1a27a3b0fb91bca51b4af6bb3bbdac968ccae0ba</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>nerves_toolchain_armv6_rpi_linux_gnueabi</name>
<purl>pkg:hex/nerves_toolchain_armv6_rpi_linux_gnueabi@1.2.0</purl><version>1.2.0</version></component><component type="library">
<description>Nerves System - Raspberry Pi A+ / B+ / B</description><hashes><hash alg="SHA-
256">a9b68559cbb4585d7b050394f2e2395cc7517f9a0816d9ea9115f1164db731e9</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>nerves_system_rpi</name><purl>pkg:hex/nerves_system_rpi@1.9.2</purl><version>1.9.2</version></component
<component type="library"><description>Get your boot on.</description><hashes><hash alg="SHA-
256">f9a1b7d6212cf18ba91c4f71c26076059df33cea4db2eb3c098bfa6673349412</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>shoehorn</name><purl>pkg:hex/shoehorn@0.6.0</purl><version>0.6.0</version></component><component
type="library"><description>Nerves System - BeagleBone Black, BeagleBone Green, PocketBeagle and more</description><hashes><hash
alg="SHA-256">e7ac32898cc4fb259a308116df2745fff12ea360cdb91c74906baef49b223ada</hash></hashes><licenses><license><id>Apache-
2.0</id></license></licenses><name>nerves_system_bbb</name><purl>pkg:hex/nerves_system_bbb@2.4.2</purl><version>2.4.2</version>
</component><component type="library"><description>A Make compiler for Mix</description><hashes><hash alg="SHA-
256">38349f3e29aff4864352084fc736fa7fa0f2995a819a737554f7ebd28b85aaab</hash></hashes><licenses><license><id>Apache-2.0</id>
</license></licenses><name>elixir_make</name><purl>pkg:hex/elixir_make@0.6.0</purl><version>0.6.0</version></component><component
```


File formats: .json, .xml

SWID


```
<?xml version="1.0" encoding="utf-8"?>
<SoftwareIdentity xmlns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd" xmlns:sha256="http:/
/www.w3.org/2001/04/xmlenc#sha256" xmlns:n8060="http://csrc.nist.gov/ns/swid/2015-extensions/1.0" xm
lns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://standards.iso.org/iso
/19770/-2/2015/schema.xsd http://standards.iso.org/iso/19770/-2/2015-current/schema.xsd     http://c
src.nist.gov/ns/swid/2015-extensions/1.0 https://csrc.nist.gov/schema/swid/2015-extensions/swid-2015
-extensions-1.0.xsd" xml:lang="en-US" name="zip" tagId="unavailable.invalid.zip-3.0-26.fc32.x86_64"
version="3.0-26.fc32.x86_64" versionScheme="rpm">
  <Entity name="" regid="invalid.unavailable" role="tagCreator"/>
  <Link rel="required" use="required" type="swid+xml" ownership="shared" href="swid:unavailable.inva
lid.bzip2-libs-1.0.8-2.fc32.x86_64-rpm-72c50b49853aa8ce60896262a85734085f7836d7553e18e68037f3e913724
25d.swidtag"/>
  <Link rel="required" use="required" type="swid+xml" ownership="shared" href="swid:unavailable.inva
lid.glibc-2.31-2.fc32.x86_64-rpm-e8641adf7969deaa30846bac77c7accf70f3588da3ee4668c73090bfa2e97507.sw
idtag"/>
  <Link rel="required" use="required" type="swid+xml" ownership="shared" href="swid:unavailable.inva
lid.unzip-6.0-47.fc32.x86_64-rpm-8a274fd9aafd2a6d435dd0f923b73be1870097e3bd8e989ad5809f59de14ba78.sw
idtag"/>
  <Meta product="zip" colloquialVersion="3.0" revision="26.fc32" arch="x86_64" summary="A file compr
ession and packaging utility companiestible with PKZIP"/>
  <Evidence date="2020-06-12T19:08:27Z" deviceId="localhost.localdomain" n8060:pathSeparator="/" n80
60:envVarPrefix="$" n8060:envVarSuffix="">
    <File size="213648" name="zip" location="/usr/bin" sha256:hash="8abb7885954cd7cd8a2f9dbecf96a965
c4a837329b9a9bf1eb4e586b8f7e22f5" key="true"/>
    <File size="106416" name="zipcloak" location="/usr/bin" sha256:hash="fa902ca689f188350642284ba56
306660d574755ddb63fcf27f9b333e8ec7f80" key="true"/>
    <File size="97816" name="zipnote" location="/usr/bin" sha256:hash="1418608f5d7675b39e681eec8cc6f
30e0bb418ffd92a73a28575514c38abbac2" key="true"/>
    <File size="97864" name="zipsplit" location="/usr/bin" sha256:hash="0d13183bb15a20ad76012b83b26b
a5b8def1e37195caf48c34a54e09557ef2f0" key="true"/>
    <Directory name=".build-id" location="/usr/lib">
      <Directory name="4c">
        <File size="28" name="224381b5ef923772bf5e1742f00af581b848da" key="true"/>
      </Directory>
      <Directory name="54">
        <File size="27" name="65529f3700a5309915077e5c55cf4db21ad84a" key="true"/>
      </Directory>
```


File formats: .xml

Information Collected per Tool

Tool Template

Support	Produce?, Consume?, Transform?
Functionality	
Location	Website: Source:
Installation instructions	
How to use	
Versions Supported	

Example: FOSSology

Support	Produce (Analyze, Edit), Consume(View,Diff,Import), Transform(Translate, Merge, Tool Support)
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

Tool Support for Different SBOM Formats

SPDX

http://tiny.cc/SPDX

CycloneDX

http://tiny.cc/CycloneDX

SWID

http://tiny.cc/SWID

Areas to Learn: Generalized vs. Industry-Specific Requirements

- Generalized requirements for code: software, firmware, embedded
- Where do SBOM requirements of firmware/embedded diverge from IT?
  - Ex: Auto industry, Energy, Medical devices with firmware and embedded
- Where do SBOM requirements for licensed/proprietary third party components diverge from third party open source components?
- Lessons Learned and Best Practices for SBOM IP
  - Open Formats
  - Content may be delivered under NDA
  - Content must be capable of transfer to final-goods-assembler without copyright restriction
    - Assumption: NDAs carry the weight of confidentiality terms
- Why this matters: SBOM is an intermediary phase of the data
  - Operational requirement for data to be ingested by enterprise processes and platforms
  - Ex: CMDB, SAM, SOC
  - Configuration management can’t become a “derivative work” and function as intended.