

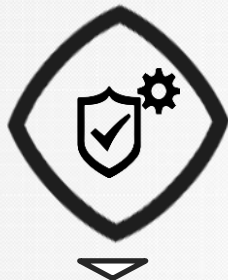
# Industry Insights

## Common Pitfalls and Key Considerations in Using Software Bill of Materials (SBOM)

**SZ Lin (林上智)**

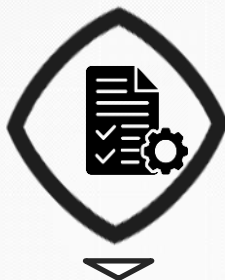
Date: 2024/05/16

# The Benefit of Adopting SBOM



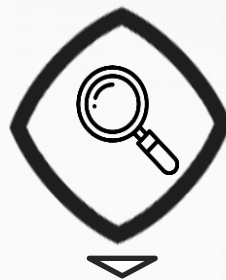
## Security Management

Identifying and avoiding known vulnerabilities



## License Management

Quantifying and managing licenses



## Transparency

Developing and maintaining the software cross the departments.



## Cost effective

Reducing time and human resources in responding to security events and customers' requests

# Regulation

# US EO 14028



BRIEFING ROOM

## Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

- (vii) providing a purchaser a **Software Bill of Materials (SBOM)** for each product directly or by publishing it on a public website;
  - (viii) participating in a vulnerability disclosure program that includes a reporting and disclosure process;
  - (ix) attesting to conformity with secure software development practices; and
  - (x) ensuring and attesting, to the extent practicable, to the integrity and provenance of open source software used within any portion of a product.
- (f) Within 60 days of the date of this order, the Secretary of Commerce, in coordination with the Assistant Secretary for Communications and Information and the Administrator of the National Telecommunications and Information Administration, shall publish minimum elements for an **SBOM**.

# EU - on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020



EUROPEAN COMMISSION

Brussels,  
15.9.2022

COM(2022)  
454 final

2022/0272(COD)

Proposal for a

**REGULATION OF THE EUROPEAN PARLIAMENT AND  
OF THE COUNCIL**

**on horizontal cybersecurity requirements for products with  
digital elements and amending Regulation (EU) 2019/1020**

- (37) In order to facilitate vulnerability analysis, manufacturers should identify and document components contained in the products with digital elements, including by drawing up a **software bill of materials**. A software bill of materials can provide those who manufacture, purchase, and operate software with information that enhances their understanding of the supply chain, which has multiple benefits, most notably it helps manufacturers and users to track known newly emerged vulnerabilities and risks. It is of particular importance for manufacturers to ensure that their products do not contain vulnerable components developed by third parties.



# Medical Regulation – FDA and MDR

## (a) Software Bill of Materials

A Software Bill of Materials (SBOM) can aid in the management of cybersecurity risks that exist throughout the software stack. A robust SBOM includes both the device manufacturer-developed components and third-party components (including purchased/licensed software and open-source software), and the upstream software dependencies that are required/depended upon by proprietary, purchased/licensed, and open-source software. An SBOM helps facilitate risk management processes by providing a mechanism to identify devices that might be affected by vulnerabilities in the software components, both during development (when software is being chosen as a component) and after it has been placed into the market throughout all other phases of a product's life.<sup>29</sup>

Because vulnerability management is a critical part of a device's security risk management processes, an SBOM or an equivalent capability should be maintained as part of the device's configuration management, be regularly updated to reflect any changes to the software in

marketed devices, and should support 21 CFR 820.30(j) (Design History File) and 820.181 (Design Master Record) documentation.

To assist FDA's assessment of the device risks and associated impacts on safety and effectiveness related to cybersecurity, FDA recommends that premarket submissions include SBOM documentation as outlined below. SBOMs can also be an important tool for transparency with users of potential risks as part of labeling as addressed later in Section VI

FDA [1]

Often, specific security information is shared through documentation other than the instructions for use, such as instructions for administrators or security operation manuals. Such information may include the following:

- List of IT security controls included in the medical device
- Depending on the type of product, provisions to ensure integrity/validation of software updates and security patches
- Technical properties of hardware components
- **Software Bill of Materials**
- User roles and respective access privileges/permissions on the device
- Implementation of the logging function, particularly the medical device's log storage capacity and the recommendations for backing up and using the logs

MDR [2]

## **Pitfall #1**

“SBOM can be presented in any format and through various methods.”

# SBOM ... from a real case

Name	Version
XXX product software	1.0



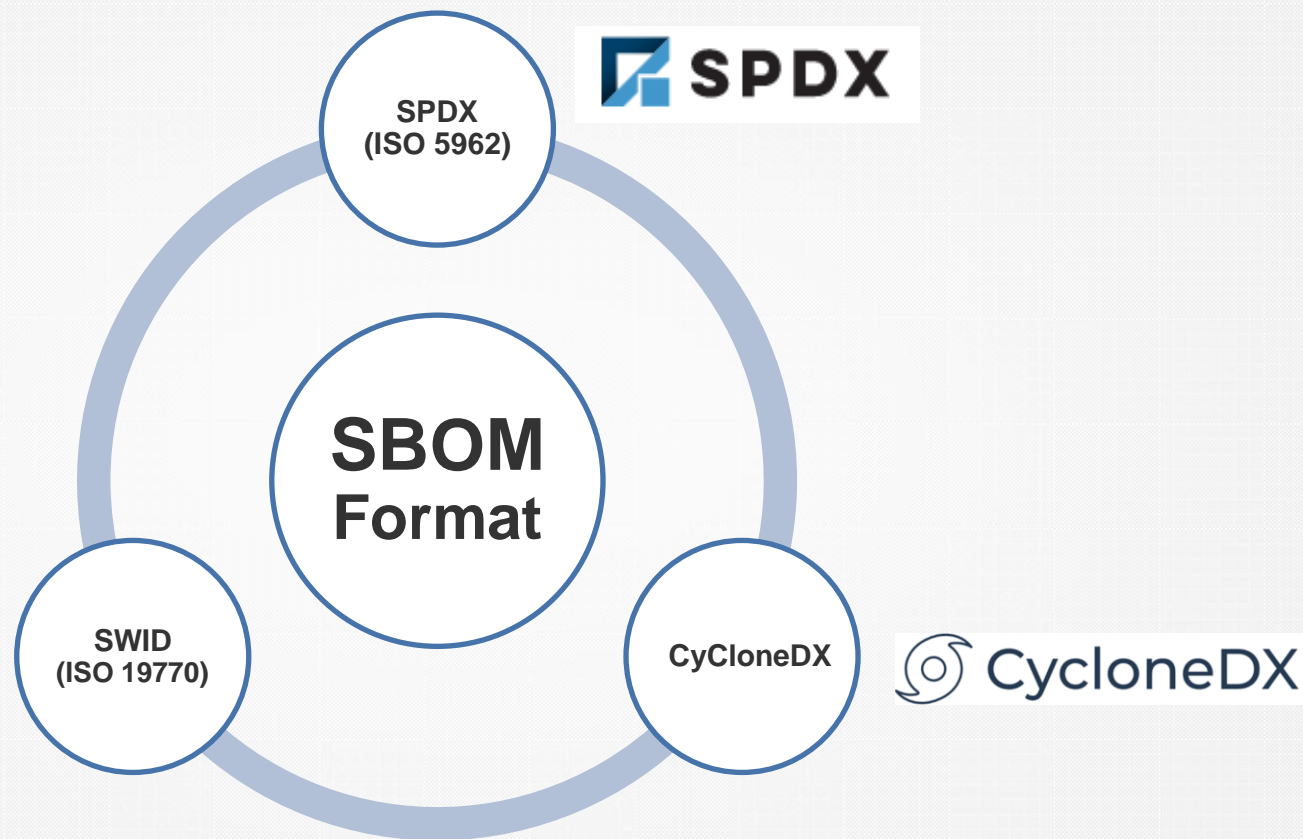


# The SBOM Definition <sup>[3]</sup>

*“is a formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships.”*

# NTIA SBOM Baseline <sup>[4]</sup>

Data Field	Description
Supplier Name	The name of an entity that creates, defines, and identifies components.
Component Name	Designation assigned to a unit of software defined by the original supplier.
Version of the Component	Identifier used by the supplier to specify a change in software from a previously identified version.
Other Unique Identifiers	Other identifiers that are used to identify a component, or serve as a look-up key for relevant databases.
Dependency Relationship	Characterizing the relationship that an upstream component X is included in software Y.
Author of SBOM Data	The name of the entity that creates the SBOM data for this component.
Timestamp	Record of the date and time of the SBOM data assembly.



# An Example of SPDX Format

```
SPDXVersion: SPDX-2.3
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: .
DocumentNamespace: https://anchore.com/syft/dir/
LicenseListVersion: 3.20
Creator: Organization: Anchore, Inc
Creator: Tool: syft-0.79.0
Created: 2023-05-10T12:41:22Z

#### Package: Browser

PackageName: Browser
SPDXID: SPDXRef-Package--Browser-b575d19b437379c1
PackageVersion: 2.1
PackageDownloadLocation: NOASSERTION
FilesAnalyzed: false
PackageSourceInfo: acquired package info from SBOM: backend/src-common/src/test/resources/bom.spdx
PackageLicenseConcluded: NONE
PackageLicenseDeclared: NONE
PackageCopyrightText: NOASSERTION
ExternalRef: SECURITY cpe23Type cpe:2.3:a:Browser:Browser:2.1:*:*:*:*:*:*
ExternalRef: PACKAGE-MANAGER purl pkg:/
```

# Mapping baseline component information to existing formats [5]

Attribute	SPDX	CycloneDX	SWID
Author Name	Creator	metadata/authors/author	<Entity> @role (tagCreator), @name
Timestamp	Created	metadata/timestamp	<Meta>
Supplier Name	PackageSupplier	Supplier publisher	<Entity> @role (softwareCreator/publisher), @name
Component Name	PackageName	name	<softwareIdentity> @name
Version String	PackageVersion	version	<softwareIdentity> @version
Component Hash	PackageChecksum Or VerificationCode	Hash "alg"	<Payload>/../<File> @[hash-algorithm]:hash
Unique Identifier	DocumentNamespace combined with SPDXID	bom/serialNumber component/bom-ref	<softwareIdentity> @tagID
Relationship	Relationship: DESCRIBES; CONTAINS	(Inherent in nested assembly/subassembly and/or dependency graphs)	<Link> @rel, @href

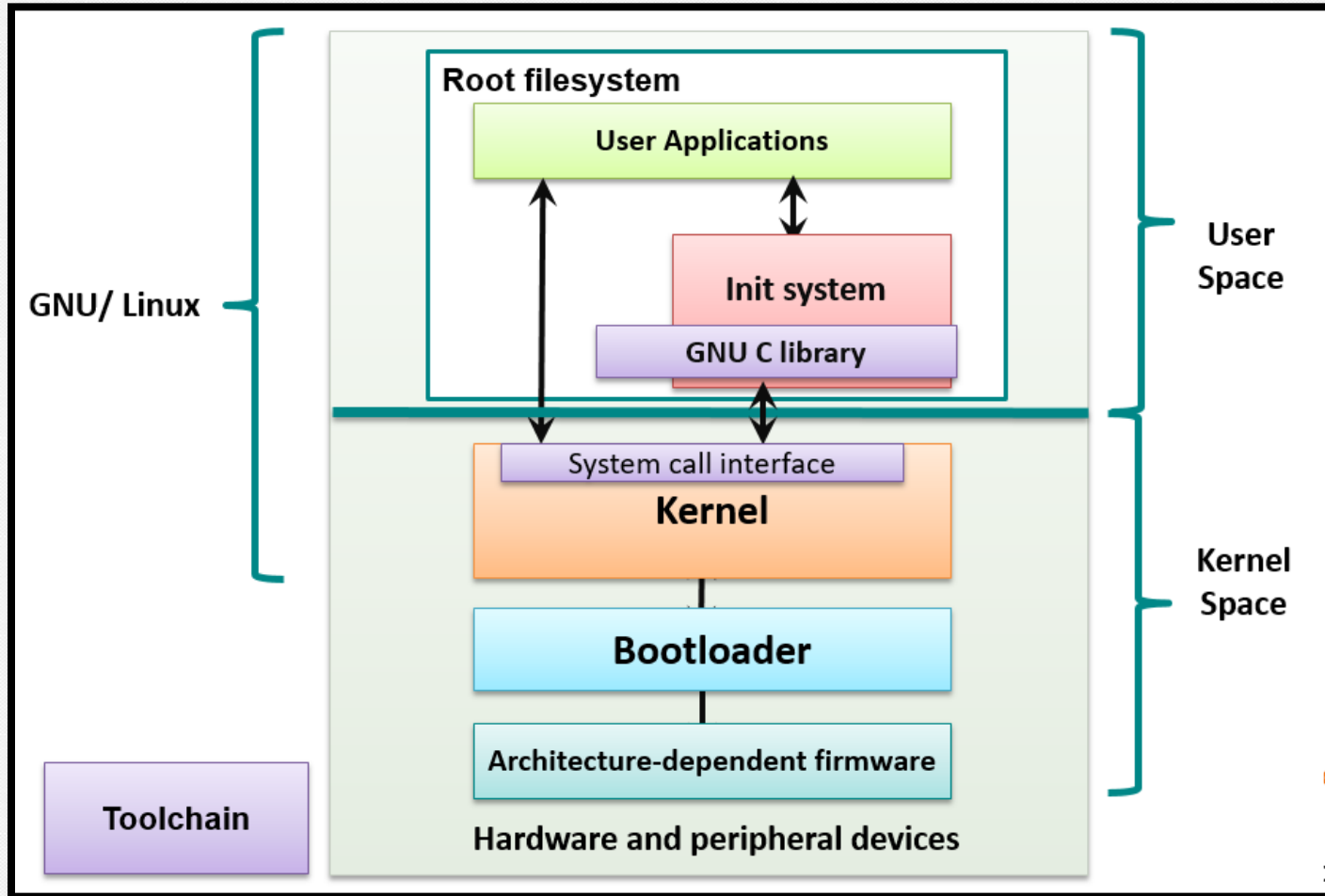


## **Pitfall #2**

“As we did not develop the open-source or commercial software, our focus is solely on listing the software we have authored.”

# SBOM ... from another real case

Name	Version
XXX main application	1.0
XXX web application	2.0
XXX SQL application	2.0
XXX log application	2.1



# The SBOM Definition <sup>[3]</sup>

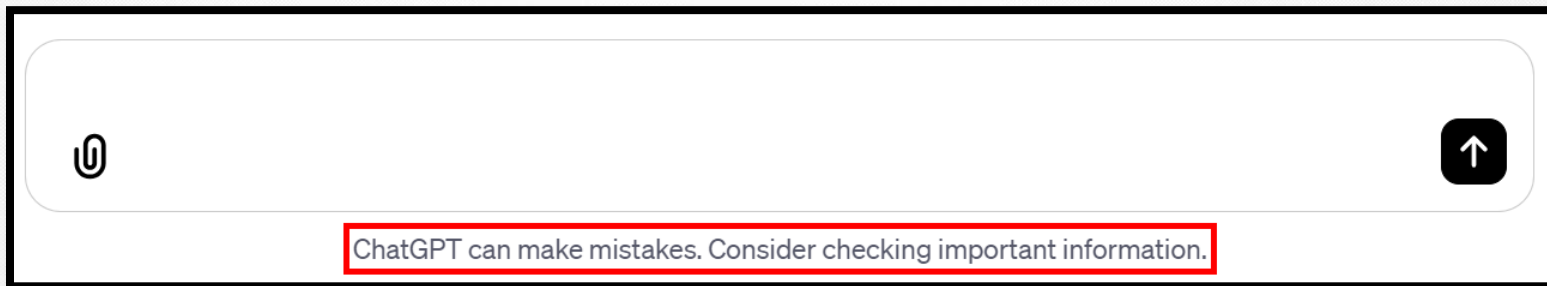
*“may include open source or proprietary software and can be widely available or access-restricted..”*

## **Pitfall #3**

“By purchasing commercial software tools, an SBOM can be generated without the need for human intervention.”



**The tool is meant to assist.  
Please do not rely on it 100%.  
“False Positive”  
“False Negative”**



A screenshot of a chat interface. It features a large white input field with rounded corners, a paperclip icon on the left, and a send button (upward arrow) on the right. Below the input field, a red-bordered box contains the text: "ChatGPT can make mistakes. Consider checking important information."

- Report Generated: 2023-05-10 20:57:31
- Time of last update of CVE Data: 2023-05-10 20:57:20

#### CVE SUMMARY

Severity	Count
CRITICAL	0
HIGH	0
MEDIUM	2
LOW	1
UNKNOWN	0

#### NewFound CVEs

Vendor	Product	Version	CVE Number	Source	Severity	Score (CVSS Version)
apple	mail	1.4.7	CVE-2005-2512	NVD	LOW	2.1 (v2)
apple	mail	1.4.7	CVE-2008-0039	NVD	MEDIUM	6.8 (v2)
apple	mail	1.4.7	CVE-2010-3887	NVD	MEDIUM	4.3 (v2)

# Tool Support for Different SBOM Formats [3]

## SPDX

<b>Format Overview</b>	<b>2</b>
Format Publishing History	2
Tool Classification Taxonomy	2
<b>Open Source Tools</b>	<b>4</b>
Augur	4
FOSSology	4
in-toto	5
kernel-sdpx-ids	5
npm-sdpx	6
Open Source Software Review Toolkit (ORT)	6
OWASP Dependency-Track	6
Quartemaster (QMSTR)	7
REUSE	8
ScanCode Toolkit	8
SPDX Java Libraries and Tools	9
SPDX Python Libraries	10
SPDX GoLang Libraries	10
SPDX JavaScript Libraries	11
SPDX Online Tools	11
SPDX Maven Plugin	12
SPDX Build Tool	12
SPARTS	12
SW360	13
TERN	13
Yocto Project / OpenEmbedded	14
<b>Proprietary Products</b>	<b>15</b>
CyberProtek	15
FOSSID	15
Hub-SPDX (Black Duck Hub Report Utility)	16
MedScan	16
Protecode	17
Protex	17
SourceAuditor	17
TrustSource	18
Vigilant-ops	18

<http://tiny.cc/SPDX>

## SWID

<b>Format Overview</b>	<b>2</b>
Format Publishing History	2
Tool Classification Taxonomy	2
<b>Open Source Tools</b>	<b>3</b>
Swidgen	3
StrongSwan SWID Generator	3
Labels4 SWID Generator	3
Labels4 SWID Maven Plugin	4
libswid	4
SwidTag	4
TagVault SWID Tag Creator	5
RPM 2 SWID Tag	5
NIST SWID for GNU Autotools	6
NIST SWID Tag Validator	6
NIST SWID Builder	6
NIST SWID Maven Plugin	7
NIST SWID Repo Client	7
WOX Toolkit	8
swidq	8
<b>Proprietary Products</b>	<b>9</b>
IT Operations Management	9
Janif Pro	9
CyberProtek	10
MedScan	10
BigFix Inventory	11
Vigilant-ops	12
Microsoft Endpoint Configuration Manager	12

<http://tiny.cc/SWID>

## CycloneDX

<b>Format Overview</b>	<b>2</b>
Format Publishing History	2
Tool Classification Taxonomy	2
<b>Open Source Tools</b>	<b>3</b>
CycloneDX Core for Java	3
CycloneDX for .NET	3
CycloneDX for NPM	3
CycloneDX for Maven	4
CycloneDX for Gradle	4
CycloneDX for PHP Composer	4
CycloneDX for Python	5
CycloneDX for Ruby Gems	5
CycloneDX for Rust Cargo	6
CycloneDX for SBT	6
CycloneDX for Elvix Mix	6
CycloneDX for Erlang Rebar3	6
CycloneDX for Go	7
Eclipse SW360 Antenna	7
HERE Open Source Review Toolkit	7
Retire.js	8
OWASP Dependency-Track	8
OWASP Dependency-Track Jenkins Plugin	8
drtrack-audit	9
<b>Proprietary Products</b>	<b>11</b>
Sonatype Nexus IQ	11
Sonatype Nexus Lifecycle Jenkins Plugin	11
CyberProtek	12
MedScan	12
Reliza Hub	13

<http://tiny.cc/CycloneDX>

## **Pitfall #4**

“Once the SBOM is generated, everything is concluded.”

---

# The Optimal Approach: Identifying Component Vulnerabilities through SBOMs

## **Asset Inventory**

- List of components
- List of licenses
- List of vulnerabilities

## **Configuration Management**

- Change control
- ...



# SBOM Tool Classification Taxonomy

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, etc.). 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

# Consume Tool – SW 360

The screenshot displays the SW360 Consume Tool interface. The top navigation bar includes links for Home, Projects, Components, Licenses, ECC, Vulnerabilities, Requests, Search, Admin, and Preferences. The main content area is divided into a left sidebar and a central panel. The sidebar contains an 'Advanced Search' section with filters for Project name, Project Version, Project type, and Project Responsible (Email). Below this is a 'Project' section with a dropdown menu showing 'Project-A' and 'Project-B'. The central panel shows the 'Project-A' details, including a 'Summary' tab, 'Administration', 'Clearing Status', 'Attachment Usages', 'Obligations' (0/0), 'ECC Status', 'Attachments', 'Vulnerabilities' (2/2), and 'Change Log'. The 'Vulnerabilities' section is highlighted, showing a table of vulnerabilities. The table has columns for Release, External Id, Priority, Matched by, Title, Relevance for project, and Actions. Two vulnerabilities are listed: 'sudo 1.8.27' with External Id 'CVE-2021-3156' and 'OpenSSH 7.9p1' with External Id 'CVE-2021-28041'. Both are marked as 'Not Checked'. The bottom of the interface shows a 'Change rating of selected vulnerabilities to' section with a dropdown set to 'Not Checked' and a 'Change rating' button.

Projects (Widget Page)

SW360

Home Projects Components Licenses ECC Vulnerabilities Requests Search Admin Preferences

Projects

Advanced Search

Project name

Project Version

Project type

Project Responsible (Email)

Project-A

Project-B

Showing 11

SW360

Home Projects Components Licenses ECC Vulnerabilities Requests Search Admin Preferences

Project-A

Summary

Administration

Clearing Status

Attachment Usages

Obligations 0/0

ECC Status

Attachments

Vulnerabilities 2/2

Change Log

Edit Project Link to Projects Tree View List View

SW360

Home Projects Components Licenses ECC Vulnerabilities Requests Search Admin Preferences

Project-A

Summary

Administration

Clearing Status

Attachment Usages

Obligations 0/0

ECC Status

Attachments

Vulnerabilities 2/2

Change Log

Edit Project Link to Projects Show All

Total vulnerabilities: 2

VULNERABILITY STATE INFORMATION

Security Vulnerability Monitoring: Enabled

Security Vulnerabilities Display: Enabled

VULNERABILITIES

Show 10 entries

	Release	External Id	Priority	Matched by	Title	Relevance for project	Actions
<input type="checkbox"/>	sudo 1.8.27	CVE-2021-3156		CPE	CVE-2021-3156	Not Checked	
<input type="checkbox"/>	OpenSSH 7.9p1	CVE-2021-28041		CPE	CVE-2021-28041	Not Checked	

Showing 1 to 2 of 2 entries

Change rating of selected vulnerabilities to Not Checked Change rating

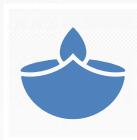
## **Pitfall #5**

“Scanning source code is the only way to generate the SBOM.”

# SBOM Type Definition and Composition <sup>[9]</sup>



Design



Source



Build



Analyzed

“3rd party” SBOM



Deployed



Runtime

“Dynamic” SBOM

## **Pitfall #6**

**“Using SBOM to scan for CVEs is  
always sufficient.”**



---

**Q.** A vulnerability is identified, and possibly assigned a CVE ID, why is it not in the NVD?

**A.** The NVD only contains CVEs that have been published to the CVE List. CVEs that have not been published are in the reserved state and their CVE Descriptions should reflect that by containing **\*\*RESERVED\*\***. The publication to the CVE data feed is controlled by the CVE Assignment Team. Once the CVE is published in the CVE data feeds, it will be available on the NVD website within a few hours.



*"that are not acknowledged by vendors  
but still are serious security issues"*

Date	!CVE ID	Title	CVSS
2023-12-06	<a href="#">NotCVE-2023-0003</a>	RSA signature verification bypass via Arbitrary Code Execution in Sansa Connect bootloader	6.2
2023-11-21	<a href="#">NotCVE-2023-0002</a>	Buffer overflow in NVD Tools	7.5
2023-05-23	<a href="#">NotCVE-2023-0001</a>	Secure Boot Bypass in MSM8916/APQ8016 Mobile SoC	7.6

## **Key Consideration**

The program is being refactored based on modular design principles.

# Avoid putting all your eggs in one basket.

Name	Version
XXX product software	1.0

## **Key Consideration**

Choosing format might be based on supply-chain needs and ecosystem characteristics.

# SBOM Format Comparison Table

	SPDX	CycloneDX	SWID
<b>Latest version</b>	v3.0	v1.6	ISO/IEC 19770-2:2015
<b>Format</b>	Tag-value, RDF/XML, JSON, yml, xls	XML, JSON and Protocol Buffers	XML
<b>Famous Linux project support</b>	Yocto, Linux Kernel, Zephyr	TBD	TBD
<b>Build system support</b>	Buildroot, OpenWRT, Yocto	Buildroot, etc	TBD
<b>Standardization</b>	ISO/IEC 5962:2021	TBD	ISO/IEC 19770-2:2015
<b>Host by</b>	Linux Foundation	OWASP	NIST
<b>“Produce” tool support</b>	11	5	2
<b>“Consume” tool support</b>	5	1	0



# Translating Between SBOM Formats & File Types [6]

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>

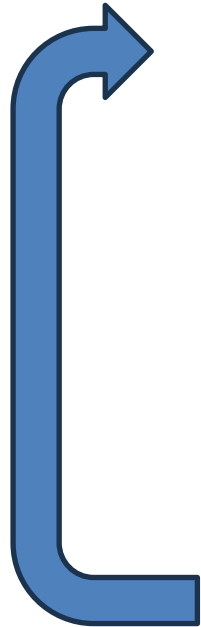
CycloneDX CLI: ( CycloneDX (.xml, .json), SPDX(.spdx))

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

# **Key Consideration**

Protection, Integrity and Authenticity

# SBOM Lifecycle <sup>[10]</sup>



SBOM Delivery



Acceptance/ Validation



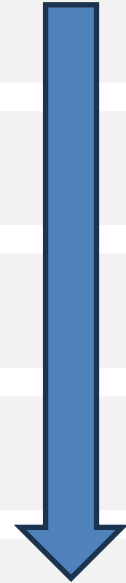
Ingestion and Management



Extraction, Transformation, and Loading



Mapping & Asset Management



---

**Thank you**

# References

- [1] Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions Draft Guidance for Industry and Food and Drug Administration Staff DRAFT GUIDANCE, Document issued on April 8, 2022.
- [2] MDCG 2019-16 Guidance on Cybersecurity for medical devices
- [3] [https://www.ntia.gov/files/ntia/publications/sbom\\_at\\_a\\_glance\\_apr2021.pdf](https://www.ntia.gov/files/ntia/publications/sbom_at_a_glance_apr2021.pdf)
- [4] [https://www.ntia.doc.gov/files/ntia/publications/sbom\\_minimum\\_elements\\_report.pdf](https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf)
- [5] [https://www.ntia.gov/files/ntia/publications/ntia\\_sbom\\_framing\\_2nd\\_edition\\_20211021.pdf](https://www.ntia.gov/files/ntia/publications/ntia_sbom_framing_2nd_edition_20211021.pdf)
- [6] [https://www.ntia.gov/files/ntia/publications/ntia\\_sbom\\_tooling\\_2021-q2-checkpoint.pdf](https://www.ntia.gov/files/ntia/publications/ntia_sbom_tooling_2021-q2-checkpoint.pdf)
- [7] Authoritative Guide to SBOM Implement and optimize use of Software Bill of Materials, Second Edition.
- [8] Software Bill of Materials (SBOM) Sharing Lifecycle Report, April 2023.
- [9] Types of Software Bill of Material (SBOM) Documents.
- [10] Securing the Software Supply Chain: Recommended Practices for Software Bill of Materials Consumption, November 2023