

Hacking supply chain: An SBOM survival guide

Dmitry Moiseev

Introduction

RU Born and raised in Russia

🌐 Now navigating the wireless waves of Chicago

🖱️ Creating cutting-edge telecom networking equipment (mostly wireless)

⚡ Sued by a *multi-billion dollar giant* for... reverse engineering (Oops!)

🔧 Still doing what I love: pushing boundaries in wireless tech



What is a Supply Chain Attack?

- **Definition:** A cyberattack targeting third-party vendors or software components to infiltrate a target organization.
- **Why It's Dangerous:**
 - **Wide-reaching impact:** One compromised vendor can infect many downstream customers.
 - **Exploits Trust:** Attackers manipulate trusted relationships between vendors and clients.
 - **Hidden Entry Points:** Attackers hide in the layers of software dependencies, making detection difficult.



Real-World Consequences: Supply Chain Attacks

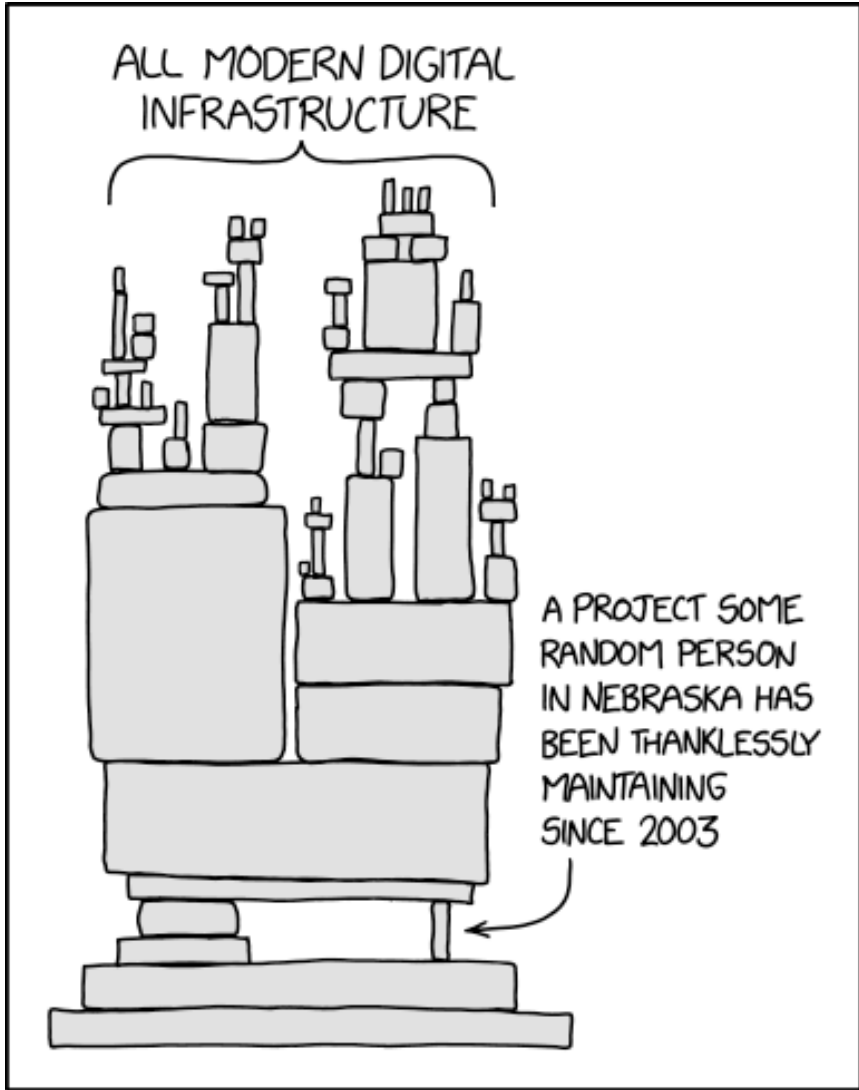
- SolarWinds: Compromise at the supplier level led to widespread breach.
- Log4j: Open-source vulnerabilities impacting millions.
- Dependency Confusion: Exploiting package managers to inject malicious code.



The XZ Utils Backdoor

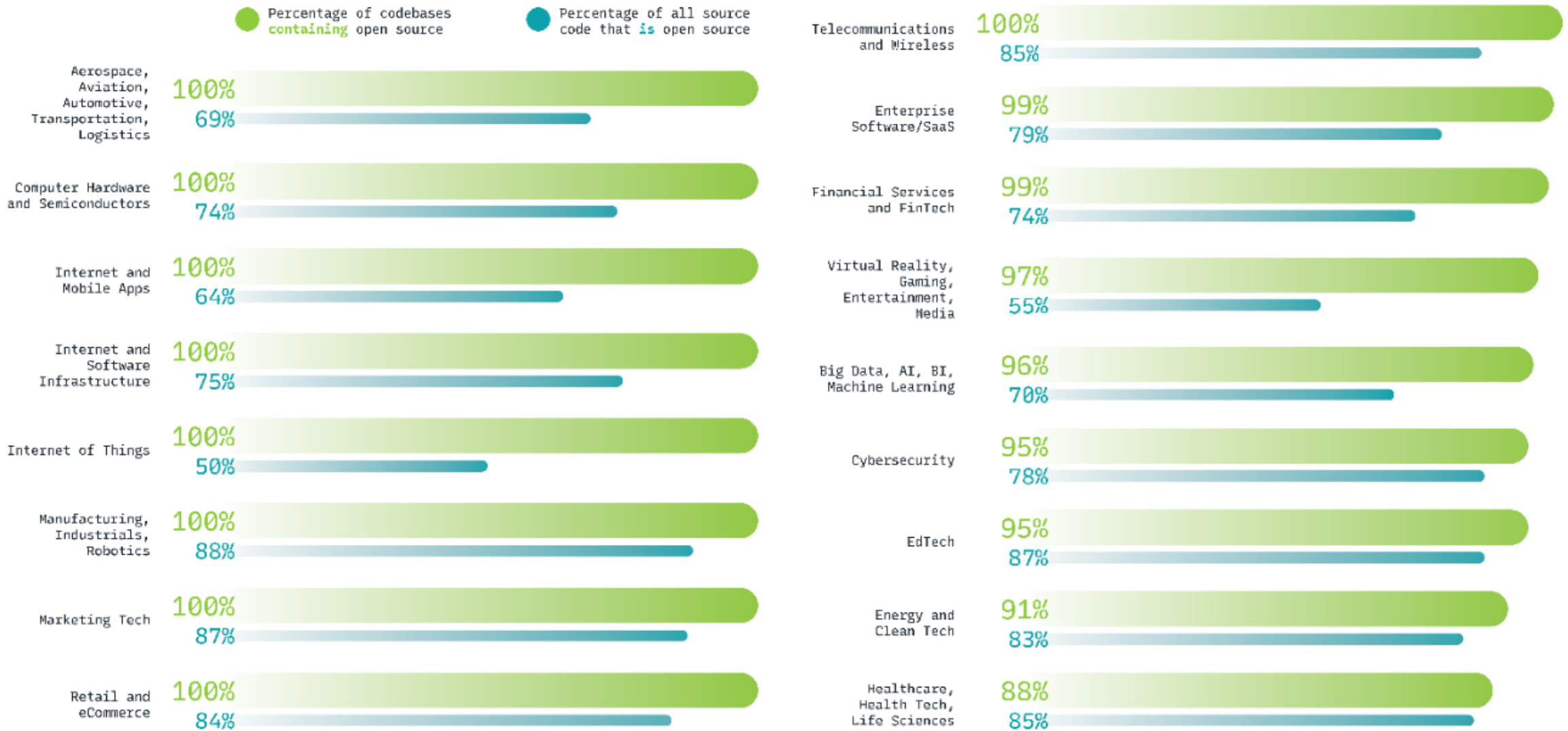
- **Social Engineering the Open-Source Ecosystem:**
Jia Tan, socially engineered their way into a co-maintainer role for the widely used **XZ Utils** (a compression library in nearly all Linux distributions).
- This attack targeted **liblzma**, a dependency for OpenSSH, which runs on over 20 million IPs globally.
- **Years in the Making:** The attack was meticulously planned since at least **2021**. It aimed to backdoor **major Linux distributions**.
- The malicious code was merged as part of binary test input files and executed during the build process
- **CVE-2024-3094:**
The backdoor allowed **unauthenticated remote code execution**, rated as a **CVSS 10** (highest possible severity)
- **Discovered by Chance:**
Microsoft engineer **Andre Freund** stumbled upon the backdoor while troubleshooting CPU spikes in Debian systems.
- It wasn't detected through **rigorous security checks** but by a stroke of luck





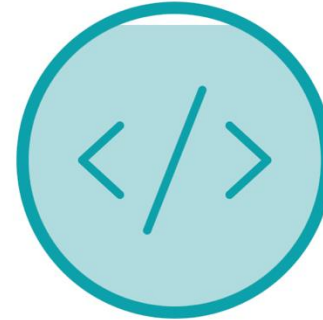
Empowering the World, Exposing the World: The Open Source Paradox

1,067 Codebases Scanned by Industry



How bad is bad?

2024 BlackDuck “Open Source Security and Risk Analysis” (OSSRA) report



96%

of the total codebases
contained open source

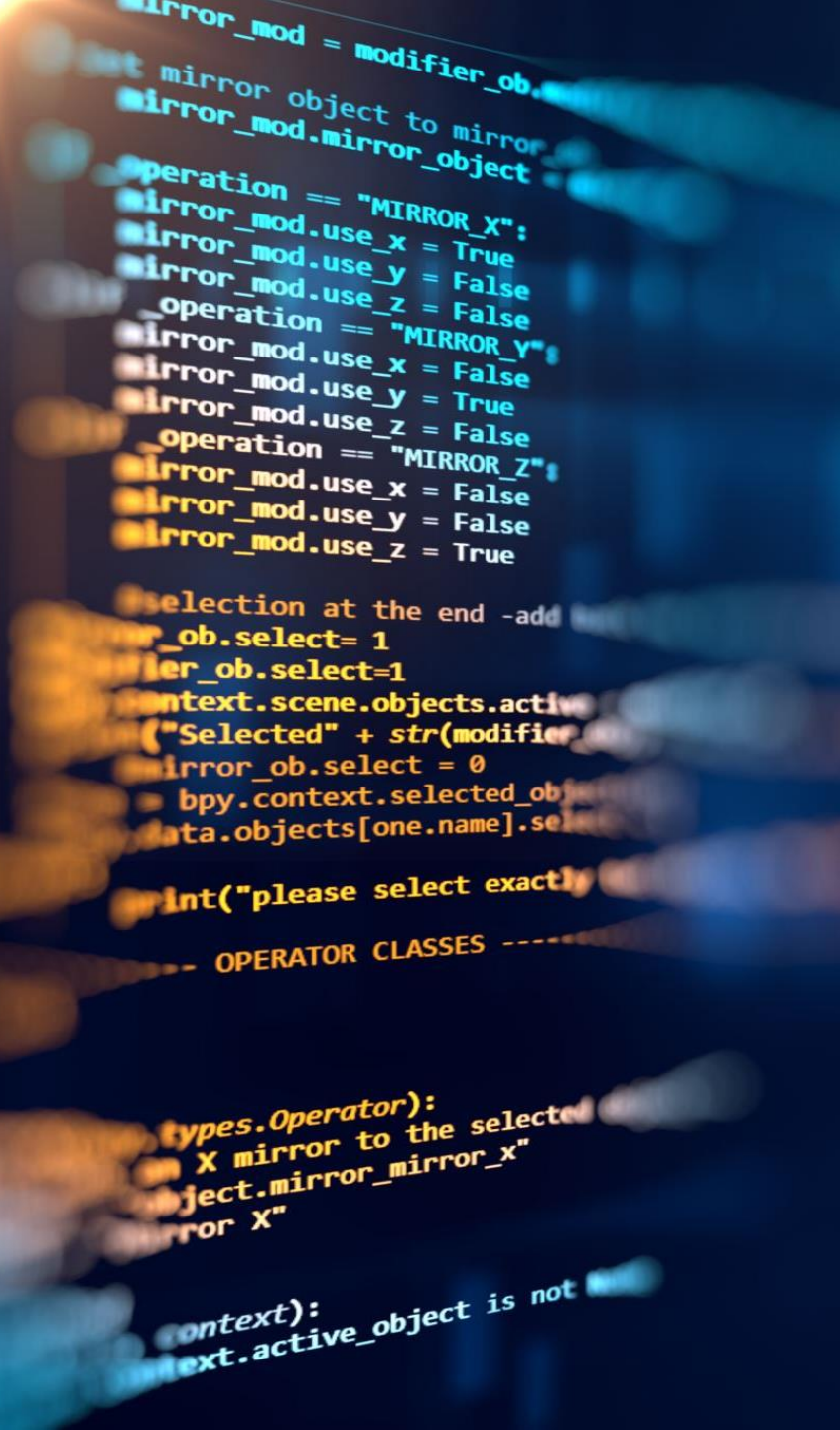


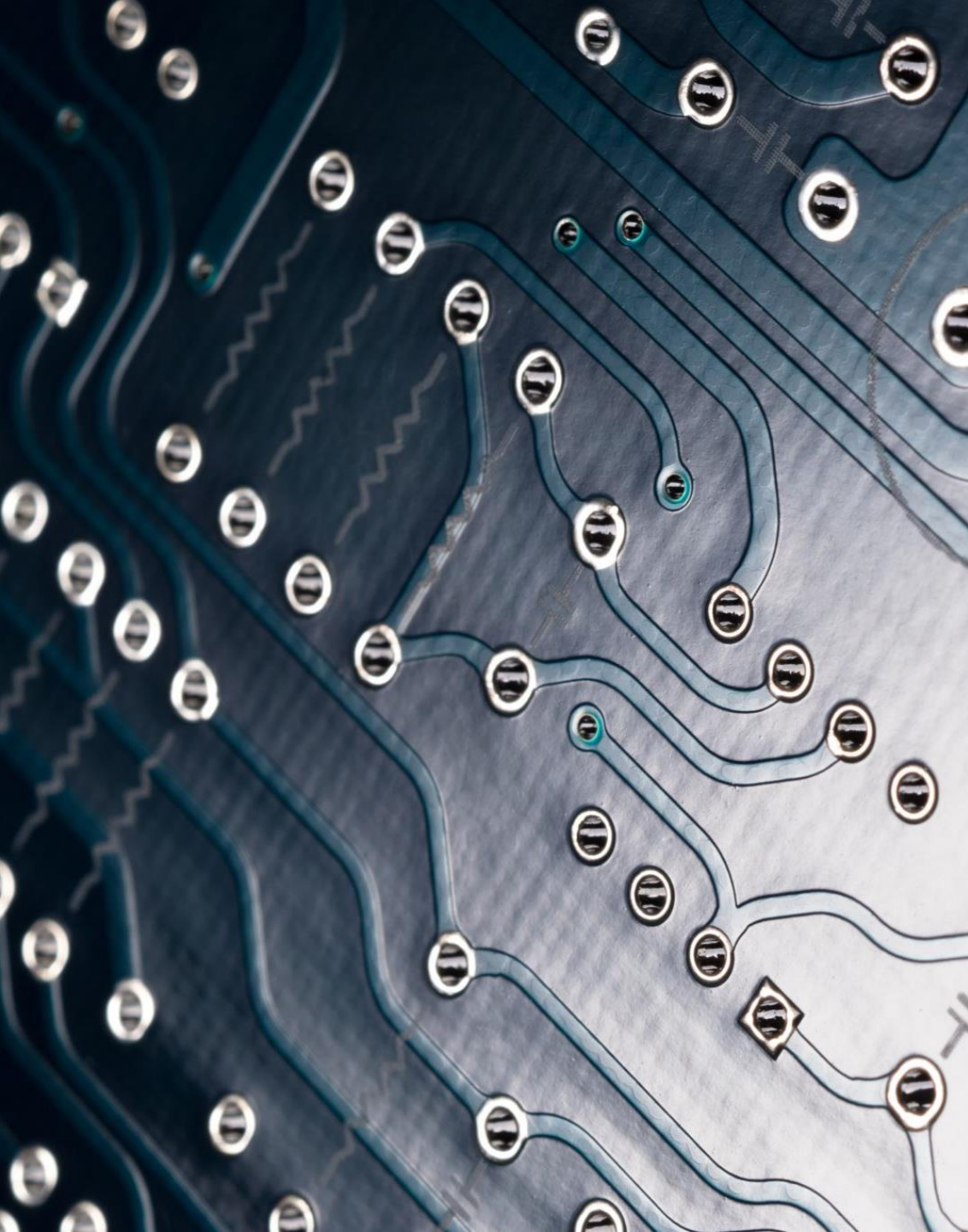
84%

of codebases contained at least
one open source vulnerability

What is an SBOM (Software Bill of Materials)?


- A detailed inventory of all components in a software product.
- Includes open-source and third-party libraries.
- Provides visibility into the entire software supply chain.





Why Does SBOM Matter?

- **Transparency:** Know exactly what's in the software you're attacking or defending.
- **Vulnerability Tracking:** Spot and exploit or patch known weaknesses faster.
- **Compliance Pressure:** Regulatory frameworks are making SBOMs mandatory in many industries.

An American flag is flying on a tall pole in front of a classical building with ornate architectural details, including columns and a balcony. The image is partially cut off by a white curved line on the right side.

Global Push for SBOM Adoption

- **U.S. Executive Order 14028** (Cybersecurity Executive Order): Requires federal agencies and vendors to provide SBOMs.
- **EU Cyber Resilience Act**: Expected to introduce similar requirements for software sold in the EU.
- **NIST's Secure Software Development Framework (SSDF)** also emphasize SBOMs.



NTIA and SBOM Governance

- **What is NTIA?**
 - The **National Telecommunications and Information Administration** (NTIA) is part of the U.S. Department of Commerce, focused on securing communications and internet infrastructure.
- **Why NTIA Cares About SBOMs**
 - To increase **cybersecurity** and transparency in software supply chains.
 - Aims to protect **critical sectors** like telecom, healthcare, and energy from supply chain attacks.
- **Is It a Requirement or a Recommendation?**
 - **Recommendation:** NTIA's SBOM guidelines are currently **recommendations**, but they align with federal efforts to improve cybersecurity standards across industries.



FDA

**U.S. Department of Health and Human Services
Food and Drug Administration**

Regulatory Push for SBOMs

- **FDA Guidelines for Medical Devices**
 - FDA requires **SBOMs** for medical devices.
 - Tracks third-party software and potential vulnerabilities.
- **CISA's Role in Critical Infrastructure**
 - **Advocating SBOM Adoption:** Focuses on securing sectors like **energy, healthcare, finance**.
 - **"SBOM-a-rama" Initiative:** Promoting widespread SBOM use and developing tools for supply chain security.
- **OpenSSF and Linux Foundation Initiatives**
 - **Securing Open-Source Supply Chains:**
 - Focus on improving **open-source software** security.
 - **Best Practices and Tool Development:**
 - Creating SBOM tools and guidelines to manage dependencies.

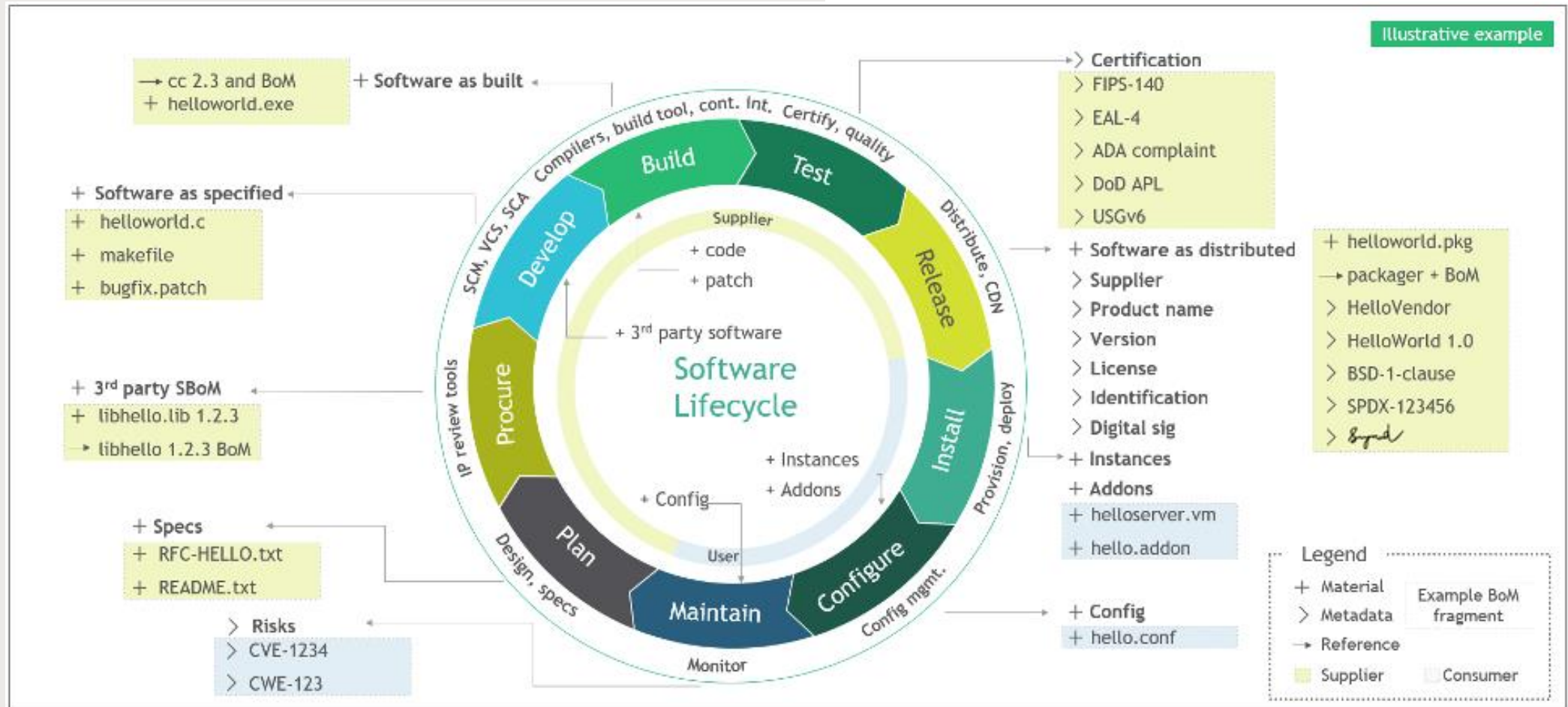
SBOMs: More Than Just Compliance

- SBOMs are moving from niche to mainstream due to growing cyber threats.
- Increasing demand for transparency in software supply chains.



Securing the Entire Software Supply Chain

- SBOMs ensure transparency across the entire supply chain, from open-source dependencies to proprietary code.



SBOM | Formats

- Software package data exchange (SPDX)
- CycloneDX (CDX)
- Software identification tags (SWID)

SBOM Format Comparison

Baseline	SPDX	SWID	CycloneDX
Who & When	Linux Foundation @ 2010	NIST @ 2009	OWASP @ 2017
Audience	Standard-centric (ISO/IEC 5962:2021) Standard , License formats , Deprecated licenses	Standard-centric (ISO/IEC 19770-2:2015) Standard , Guidelins	Developer-centric, Standard , License example
(Original) Focus	Licensing	Deployment Life Cycle	Security
Supported Unique Identifiers	SPDX License ID SWID, PURL, CPE	SWID, CoSWID	SPDX License ID, SWID, PURL, CPE SHA, BLAKE
Main Advantage	Components Relationships	Government-backed, multi-purpose	Verbosity, Security Info
Format	v2.3: RDF/XML, JSON, YAML	XML	V1.5: JSON,XML

of the

eight

B

o

it

94

m

50

r

m

7

t

ium

u

m

4

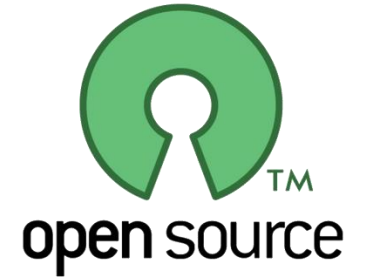
m

um

Software Unique Identifiers

- **CPE (Common Platform Enumeration)**
 - **Usage:** Identifies software, hardware, and operating systems.
 - **Format:** cpe:/a:vendor:product:version
 - **Example:** cpe:/a:microsoft:windows_10:1909
- **PURL (Package URL)**
 - **Usage:** Identifies software packages across different ecosystems (e.g., npm, Maven, PyPI).
 - **Format:** pkg:type/vendor/name@version
 - **Example:** pkg:npm/angular/core@12.0.0
- **SWID (Software Identification Tags)**
 - **Usage:** Tags software with metadata for inventory and compliance.
 - **Standard:** ISO/IEC 19770-2
 - **Example:** <SoftwareIdentity Name="ExampleApp" Version="1.0.0" />
- **GAV (Group, Artifact, Version)**
 - **Usage:** Uniquely identifies artifacts in Maven repositories.
 - **Format:** group:artifact:version
 - **Example:** org.apache.logging.log4j:log4j-core:2.14.1

SBOM and Open Source Licenses



- **Why Licenses Matter in SBOMs**
 - **Open-source software** often includes components governed by a variety of licenses (e.g., MIT, GPL, Apache).
 - Tracking licenses is crucial for **compliance** and **risk management** in software supply chains.
- **SBOMs** help organizations identify which licenses apply to each software component.
- **SPDX License ID**
 - **SPDX (Software Package Data Exchange)** is an open standard for identifying open-source licenses.
 - Each license has a unique **SPDX License ID** (e.g., MIT, GPL-3.0, Apache-2.0), which helps standardize license tracking in SBOMs.
- Simplifies the process of ensuring **legal compliance** by making license identification consistent and automated.

SBOM Tools

1

Generate
SBOM

2

Verify SBOM

3

Modify/Merge
SBOM

4

Diff SBOM

5

Translate
SBOM

6

Visualize
SBOM


```
3 "specVersion": "1.2",
4 "serialNumber": "urn:uuid:371ffb8c-c11e-42b5-b5b9-9280fc62783e",
5 "version": 1,
6 "metadata": {
7   "timestamp": "2020-08-03T08:53:09.834Z",
8   "tools": [
9     {
10      "vendor": "CycloneDX",
11      "name": "Node.js module",
12      "version": "2.0.0"
13    }
14  ],
15  "component": {
16    "type": "library",
17    "bom-ref": "pkg:npm/protonmail-web@4.0.0-beta.20",
18    "name": "protonmail-web",
19    "version": "4.0.0-beta.20",
20    "description": "Angular frontend for protonmail.com",
21    "licenses": [
22      {
23        "license": {
24          "id": "MIT"
25        }
26      }
27    ],
28    "purl": "pkg:npm/protonmail-web@4.0.0-beta.20",
29    "externalReferences": [
30      {
31        "type": "website",
32        "url": "https://github.com/ProtonMail/WebClient#readme"
33      },
34      {
35        "type": "issue-tracker",
36        "url": "https://github.com/ProtonMail/WebClient/issues"
37      }
38    ]
39  }
40 }
```

SBOM Example

SBOM Generation: Automation is Key

1. Syft

- CLI tool to generate SBOMs from **container images** and file systems.
- Supports **CycloneDX** and **SPDX** formats.
- Integrates easily with **CI/CD pipelines** for continuous monitoring.

2. CycloneDX

- Open-source SBOM generation tool with a focus on **security**.
- Supports multiple ecosystems and detailed **component relationship** tracking.
- Popular in **security-first** environments for dependency tracking.

3. FOSSA

- Manages open-source dependencies, licenses, and vulnerabilities.
- Automates **SBOM generation** with **legal compliance** focus.
- Integrates with **CI/CD** for continuous monitoring.

4. Black Duck by Synopsys

- Comprehensive SBOM generation with software composition analysis.
- Tracks **vulnerabilities** and **license compliance** in open-source components.
- Preferred by **large enterprises** for its security insights.

5. Dependency-Track

- Open-source platform focused on tracking vulnerabilities in **dependencies**.
- Real-time **SBOM generation** with vulnerability monitoring.
- Uses **CycloneDX** for detailed component analysis.

6. Tern

- Focused on **container images**, generates SBOMs for containers.
- Provides detailed information on **licenses** and vulnerabilities.
- Ideal for teams focused on **container security**.

7. Anchore

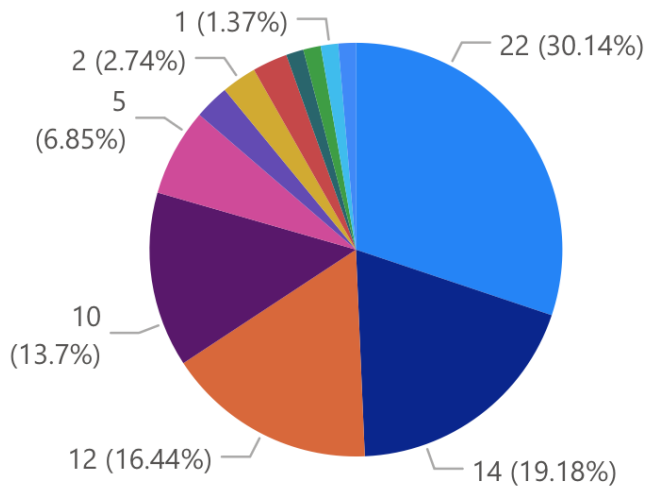
- Platform for **container security** and **software supply chain** integrity.
- Generates SBOMs and integrates with **CI/CD** for continuous security monitoring.
- Strong focus on **containerized environments**.

SBOMs +
Vulnerability
Databases =
Powerful
Defense

SBOMs combined with vulnerability databases (e.g., **CVE**, **NVD**) help identify known vulnerabilities.

Real-time alerts when new vulnerabilities are discovered in used components.

Speed up the patching process by knowing exactly which components are at risk.

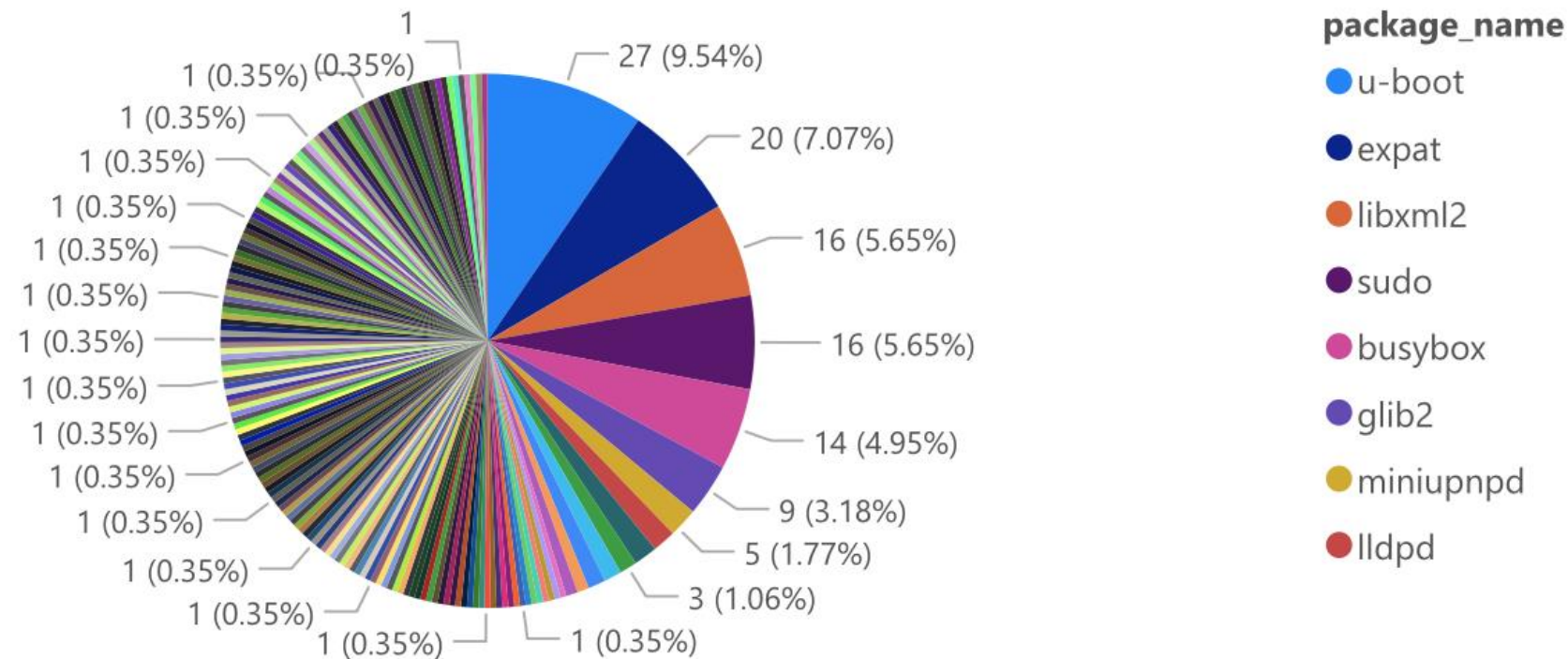


license

- GPL-2.0
- MIT
- Cambium
- BSD-3-Clause
- LGPL-2.1
- Apache-2.0
- BSD-4-Clause
- ISC
- GPL-2.0 BSD
- GPL-2.0 LGPL-2.1
- OpenSSL
- Zlib

syslog-ng	3.0.8	GPL-2.0
uboot-envtools	20081215	GPL-2.0
udevtrigger	106	GPL-2.0
util-linux	2.28	GPL-2.0
wget	1.10.2	GPL-2.0
wireless-tools	29	GPL-2.0
iputils	20071127	GPL-2.0 BSD
uci	12012009.7	GPL-2.0 LGPL-2.1
libubox	2015-09-15	ISC
lldpd	0.7.19	ISC
libiconv	1.11	LGPL-2.1
libiconv-full	1.11.1	LGPL-2.1
libmnl	1.0.3	LGPL-2.1
libwebsockets	v3.0-stable	LGPL-2.1
px5g	1	LGPL-2.1
ajaxfileuploader	2.1	MIT
bootstrap	3.5.3	MIT
bootstrap-slider	0.6.2	MIT
dropbear	2020.81	MIT
expat	2.0.1	MIT
jansson	2.5	MIT
jquery	3.2.1	MIT
jquery-progress	1.0.4	MIT
js-cookie	3.0.0	MIT
lua	5.1.4	MIT
lua-cjson	1	MIT
luajit	2017-01-17-71ff7ef	MIT
ncurses	5.7	MIT
slickgrid	1.7.2	MIT

Count of cve by package_name



package_name	package_version	cve	cvss	Description(cve.mitre.org)
libxml2	2.9.9	CVE-2021-3557	5.90	A vulnerability found in libxml2 in versions before 2.9.11 shows that it did not propagate dereference. If an untrusted XML document was parsed in recovery mode and post-validation, the highest threat from this vulnerability is to system availability.
lighttpd	1.4.54	CVE-2022-22707	5.90	In lighttpd 1.4.46 through 1.4.63, the mod_extforward_Forwarded function of the mod_extforward representing -1), as demonstrated by remote denial of service (daemon crash) in a non-handling of the Forwarded header in a somewhat unusual manner. Also, a 32-bit system
u-boot	2014.10	CVE-2019-11690	5.90	gen_rand_uuid in lib/uuid.c in Das U-Boot v2014.04 through v2019.04 lacks an srand call where CONFIG_RANDOM_UUID is enabled, and Das U-Boot is relied upon for UUID values
dashboard	1	CVE-2018-25063	6.10	A vulnerability classified as problematic was found in Zenoss Dashboard up to 1.3.4. Affected versions are: ZenPacks/zenoss/Dashboard/browser/resources/js/defaultportlets.js. The manipulation of the attack can be launched remotely. Upgrading to version 1.3.5 is able to address this issue

Who's Leading the Charge?

- **Major Tech Companies** like Microsoft, Google, and Red Hat have embraced SBOMs. Yet difficult to find.
- **Sectors like Healthcare and Finance** are pushing for SBOM adoption to protect critical infrastructure. But available either as a part of governance or by request.

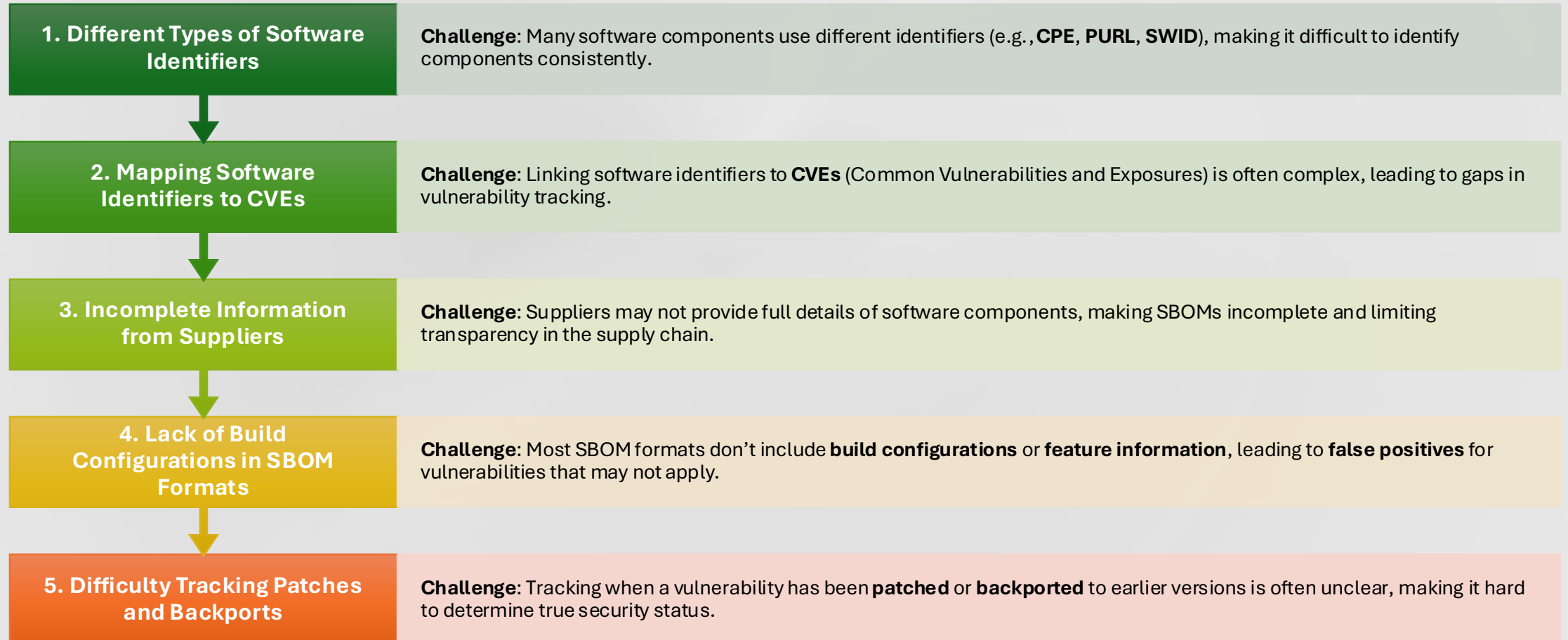


Red Hat



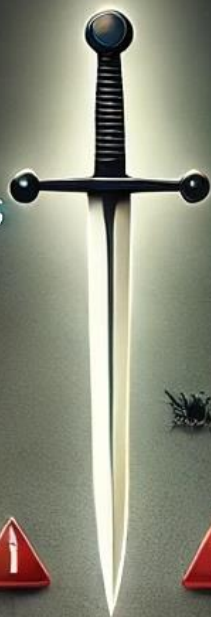
Microsoft

SBOM Challenges



EXPLOIT
SOFTWARE BILL OF MATERIALS

DEFENSE
SOFTWARE BILL OF MATERIALS



SBOM: A Double-Edged Sword

- **For Attackers:** SBOMs can reveal weak links in the software chain.
- **For Defenders:** SBOMs can help harden systems by exposing vulnerabilities before attackers do.