**Getting Your Ducks in a Row with SBOMs**

**The What and the Why Behind SBOMs**

A Software Bill of Materials (SBOM) serves as a list of ingredients for software, outlining its components. It makes the project transparent by listing all the libraries, dependencies, and third-party modules used. NTIA, NIST, and CISA advocate for extensive SBOM usage, while OpenSSF and platforms like GitHub simplify its application.

While progress is being made, creating SBOMs remains a complex challenge. Two common approaches, SPDX, and CycloneDX, make choosing between them more challenging. Hopefully, this article adds clarity on SBOMs and their relevance.

**SBOM and Transparent Security**

SBOMs are vital for software transparency and security, providing a detailed inventory of components like libraries, dependencies, and third-party code. They offer developers and security teams the visibility needed to identify vulnerabilities, manage risks, and ensure compliance. As software leans more heavily on open-source and third-party components, SBOMs are essential for understanding what’s running in an application and how secure those components are.

The adoption of SBOMs is gaining momentum, but approaches differ across the US and the EU. In the States, the President’s *Executive Order* (*EO*) 14028 mandates that all software used by federal agencies must include an SBOM. This move aims to strengthen supply chain security by improving transparency and addressing vulnerabilities in third-party code. In contrast, the EU has yet to introduce mandatory SBOM requirements. While it encourages best practices for transparency and security, the EU’s focus remains on industry-driven adoption rather than regulation, creating a slower and more flexible approach.

Beyond security, SBOMs benefit industries like healthcare, finance, and critical infrastructure by providing transparency into components that could jeopardise safety or compliance. By promoting stronger security practices, SBOMs can lower the costs of cyber incidents and build trust in digital products, contributing to greater economic stability. As the digital economy expands, SBOMs will continue to play a key role in securing software supply chains. For instance, the SPDX 3.0 standard, introduced in late 2022, signifies the progress, and its final version is going to make SBOM generation and interpretation more efficient, facilitating adoption across unique ecosystems.

**Fitting the Bill with SBOM Tools**

These tools streamline software development and ensure transparency by capturing vital information about software components, such as component names, versions, dependencies, and known vulnerabilities. They automate the creation of SBOMs, allowing for easy tracking and management of software components across the development lifecycle. The right tool enables developers to pinpoint and rectify security vulnerabilities, thus bolstering the integrity of the software supply chain.

Regarding maturity of the tools, some of them may not fully integrate with existing CI/CD pipelines and thus not capture all the relevant data, which leaves gaps in visibility. Seeking advice in places like [https://sbombenchmark.dev](https://sbombenchmark.dev/) could be beneficial. Alternatively, organisations may adhere to such a strategy:

- **Choose the Right Too**l: Evaluate SBOM tools based on the specific requirements of your project, such as compatibility with existing software frameworks and compliance needs. Tools like *Syft* are great for containerised applications, while *Dependency-Check* excels at identifying vulnerabilities.

- **Standardisation:** Choose a tool that aligns with common standards, such as SPDX or CycloneDX, for better compatibility and simpler integration across different platforms and teams.

- **Automation Integration**: Make sure the tool integrates with your development environment. Automating the SBOM generation during the build or deployment process helps maintain consistency and reduces manual errors.

- **Comprehensive Coverage**: Ensure that the tool you select provides full visibility of all relevant components, including open-source libraries, dependencies, and vulnerabilities, while leaving no gaps in the software inventory.

 - **Privacy Considerations**: SBOMs contain detailed information about software components, which can raise privacy issues when dealing with sensitive or proprietary code. Developers need to balance transparency with safeguarding intellectual property.

**Lend a Hand to Secure SBOM Adoption**

A key barrier to adoption of an SBOM is the learning curve. Training and resources must equip teams to generate SBOMs and to understand their value for security and compliance. It is understandable that organisations might be cautious about adopting SBOMs, given their concerns over time and resource constraints. However, the long-term benefits such as improved security, faster vulnerability response and better compliance will outweigh the costs. A clear cost-benefit analysis can help decision-makers.

*TODO: add an outer link to something cost-benefit related, or provide anything like that related to a product, better off, with supporting docs, ready examples, etc. Here, a screenshot/diagram will suffice.*

No policy is more effective than fostering a culture of collaboration between developers, security teams, and compliance officers. To begin the process, organisations should involve all stakeholders in both the planning and execution stages of tool integration.

To offer a quick example, one might consider looking at this project, available at <https://github.com/cossacklabs/acra>. An insightful example of SBOM is ready for download by the link <https://github.com/cossacklabs/acra/dependency-graph/sbom>.

**Key Takeaways**

1. **SBOMs for Better Security**: SBOMs shed light on your software’s components, helping to spot vulnerabilities and manage risks before they escalate.

2. **Regulatory Push**: The U.S. has made SBOMs mandatory for government software, while the EU adopts a more relaxed, recommendation-based approach. This creates both challenges and opportunities for businesses operating across both regions.

3. **Choose the Right Tools**: Selecting the right SBOM tool is crucial. Ensure it fits in with your workflow and supports standards like SPDX or CycloneDX.

4. **Privacy and Accuracy Hiccups**: While SBOMs enhance transparency, they can raise privacy concerns and suffer from inaccuracies if the organisation relies on secondary sources a lot.

5. **Industry-Wide Impact**: SBOMs are not only vital for security—they are a significant advantage for compliance across sectors like healthcare and finance, boosting trust and reliability.

6. **Standards in Progress**: With **SPDX 3.0** still under development, SBOM standardisation is on the horizon, promising greater consistency and easier integration.

In summary, incorporating SBOMs is the smart move for securing software, simplifying compliance, and making your development processes future-proof.