sonatype

CHEAT SHEET

SBOM best practices for development teams

Integrate SBOM management into your development lifecycle

- **Make SBOMs part of your standard SDLC process:** Building SBOM creation into the CI/CD pipeline and automatically generating a list of all open source or third-party dependencies and code moves vulnerability detection left and allows DevOps teams to detect vulnerable components before deploying the application.
- **Educate yourself on the relevant regulatory requirements:** Software supply chain security is getting a lot of attention these days, and SBOMs are taking center stage in this effort. Familiarize yourself with current federal and industry-specific regulations and adopt policies that reflect your needs.
- **Understand your toolchain:** Know what build tools and environments your project uses and which ones can generate SBOMs. Not every tool has a ready-made generator and you may need to create custom scripts
- **Check out community and support:** Since SBOM technologies are new, it's worth assessing the related (if any) communities and available vendor support when choosing SBOM tools.

Automate SBOM generation and updates

- **Plan to automate:** Where possible, integrate SBOM generation into your CI/CD pipelines to ensure updates with every build. Your build process will produce many SBOMs, so be prepared for scale and management
- **Choose the right tools:** Select an SBOM generation tool that integrates well with your specific development environment. Standardizing on one SBOM format will reduce complexity but may not always be feasible.
- **Automate policy support:** Equip development teams to know upfront during the design stage what open source or third-party products are approved and which ones to avoid.

Prioritize vulnerability tracking and management at all levels

- **Cross-reference with vulnerability databases:** be proactive in updating or replacing vulnerable components.
- **Plan for comprehensive coverage:** Remember that vulnerabilities can exist within every level of the software stack, so provide full coverage for top-level components as well as all nested dependents.

Treat SBOMs as sensitive information

- **Secure storage:** Many of these SBOMs may be legal or contractual documents. Treat them accordingly and store SBOMs in a secure yet accessible location for compliance and vulnerability management.
- **Sign your SBOMs:** Providing a digital signature to your SBOM provides a way to authenticate that the SBOM originates with you. These signatures provide a unique identifier, so any changes to the SBOM will generate a new signature.

Regularly review SBOMs for licensing and compliance issues

- **Include your legal team in the process:** Work with legal counsel to establish a policy on approved and disallowed licenses. Lawsuits for unintended license violations are a hidden risk, so familiarizing development teams with which licenses are acceptable is essential in the project planning process.
- **Automate reviews and keep a schedule:** Use automated tools to analyze SBOMs, reduce the time required, and eliminate errors.
- **Plan to self-check:** SBOMs that are inaccurate are worthless and can lead to legal or contractual problems. Plan to apply a self-assessment process that can check that software components can be traced through the build.

Use an exchangeable standard format as part of your SDLC

- **Adopt standard formats to simplify the process and minimize mistakes:** These formats allow SBOMs to be automatically generated during the development process.

CYCLONEDX

CycloneDX is an open-source standard developed by the Open Web Application Security Project (OWASP) community. It was designed specifically to bolster security across software supply chains and is known for its lightweight nature. It fosters an environment where adoption and integration into build pipelines are seamless and efficient.

Notably, CycloneDX is engineered for cyber risk mitigation, gaining the trust of critical sectors such as government and defense. It boasts compatibility with over 200 tools and extends its reach across more than 20 programming languages.

SPDX

SPDX is an open-source blueprint for SBOMs that simplifies the conveyance of essential details such as software names, versions, components, licenses, copyrights, and security references. It excels at reducing redundancies and streamlining distribution and compliance processes, backed by its ISO/IEC recognition as an international standard.

Key attributes of SPDX include:

- Precision in documenting software components, licenses, and other critical data, making it indispensable for compliance and legal frameworks.
- Enables linking of artifacts to global reference systems like CPE, Package URL (purl), SWHID, enhancing security and management of software components.

Understand SBOM limitations

The objective of an SBOM is to provide a clear and comprehensive description of the contents of the software you deliver. Given the complexity of modern build processes and the relative newness of SBOM tooling, it's important to recognize that not every SBOM will be as comprehensive as possible and that most SBOM generators focus on one type of content. A considerable number of SBOMs will be required to cover all the content of a software application. The goal of SBOMs being aggregators of SBOMs from previous build steps has yet to be achieved.

sonatype