sonatype

CHEAT SHEET

Best practices for SBOMs in DevSecOps

Automated SBOM generation

- **Automate for precision:** Leverage automation tools for each software build, ensuring your SBOM is always accurate and current.
- **Separate build and release:** Incorporate SBOMs within your software development life cycle (SDLC) to enable monitoring. Also ensure SBOM data is meticulously captured and securely retained for versions that are released, deployed, or shipped.

Integration with DevSecOps workflow

- **CI/CD pipeline embedding:** Incorporate SBOM generation and management tools within CI/CD workflows for automatic security assessments.
- **In-depth component scanning:** Ensure SBOMs are created with accurate identification tied to deep, timely, accurate data to ensure a proper view of risk.

Strategic utilization

- **Rapid vulnerability response:** Quickly identify and remediate vulnerabilities identified via SBOMs to ensure compliance and secure software components.
- **Assurance:** Maintain audit-ready compliance by importing and retaining every SBOM unlocking rapid response to incidents, audits, and compliance requests.

Collaboration and accessibility

- **Universal access:** Grant all relevant teams access to an SBOM application or interface to foster a collaborative security culture.
- **Targeted training:** Provide education on the advantages and interpretations of SBOMs, emphasizing security implementations.

Tools and services

- **Focus on integration and automation:** Opt for tools that offer seamless workflow integration, automate SBOM generation, and provide comprehensive scanning for security and compliance.
- **Choose dual-purpose tools:** Ensure your tools support both integrated SBOM generation during the SDLC and efficient management of 1st- and 3rd-party applications, enabling risk and compliance oversight across your software ecosystem.

Continuous monitoring and feedback

- **Alert system:** Implement an alert mechanism for newly discovered vulnerabilities in existing SBOMs that could be affecting your 1st- and 3rd-party software components.
- **Iterative improvement:** Establish feedback loops for continuous refinement of your SBOM strategy, adapting to emerging security challenges and tech advancements.

Implementation steps

1. **Evaluate current processes:** Assess how existing development and security workflows align with SBOM capabilities.
2. **Select appropriate tools:** Choose SBOM and DevSecOps tools that offer easy integration, scanning capabilities, and support for SBOM management.
3. **Automate SBOM integration:** Automate SBOM generation within your CI/CD pipeline for consistent updates and scanning.
4. **Procurement and 3rd-party SBOM management:** Implement processes for ingestion and management of SBOMs from 3rd-party vendors, ensuring they meet your security and compliance standards.
5. **Establish Governance, Risk and Compliance [GRC] protocols:** Integrate SBOM insights into your governance, risk management, and compliance (GRC) framework to enhance decision-making and regulatory adherence.
6. **Enable teams:** Educate development, security, and operations teams on utilizing SBOMs for enhanced security.
7. **Make available to customers:** Develop and implement a process to share verified SBOMs with customers, enhancing transparency and trust in your software's security and compliance.
8. **Implement monitoring and risk assessment:** Establish continuous monitoring of SBOM data for real-time threat and vulnerability assessment, ensuring immediate response and mitigation.
9. **Monitor performance:** Regularly assess the effectiveness of SBOM integration on security posture and make necessary optimizations.