
BLACKTECHSTARTUP EDUCATION

Know What's Inside Your Software: Build an SBOM Before a Customer Demands One

Customers increasingly need visibility into software components and dependencies. A software bill of materials turns dependency inventory into a security, incident-response and procurement capability.

Customers increasingly need visibility into software components and dependencies. A software bill of materials turns dependency inventory into a security, incident-response and procurement capability.

Your application contains far more software than your team wrote

Modern products depend on open-source packages, container images, operating-system libraries, SDKs and vendor components. When a vulnerability is disclosed, the first business question is often not 'how bad is it?' but 'are we using it?' A software bill of materials, or SBOM, creates an inventory of software components and relationships that helps answer that question. CISA published updated minimum elements for SBOMs in July 2026, reinforcing that component transparency is becoming part of mature software supply-chain practice.

Generate inventory from the build, not from memory

Do not ask developers to maintain a spreadsheet of dependencies by hand. Generate SBOMs from package manifests, lock files, container images or build pipelines using tooling appropriate to your stack. Capture supplier, component name, version, identifiers and dependency relationships consistent with the format and customer requirements you need to meet. Store the SBOM with the release it describes. If the application changes every week but the SBOM is produced once a year, the document is historical evidence, not operational visibility.

Use the SBOM to shorten vulnerability response

When a critical vulnerability appears, query the component inventory to determine affected products and versions. Then verify reachability and exploitability rather than assuming every listed dependency creates equal risk. Record remediation, compensating controls and customer communication. The SBOM is a starting map, not a risk score. Pair it with vulnerability-management processes and secure-development practices. The real economic value is reduced investigation time and more credible answers when customers ask whether their deployment is affected.

Make supplier obligations explicit

If you ship third-party or white-labeled software, decide what component information suppliers must provide and how quickly they must notify you of serious vulnerabilities. Include security-update and end-of-support expectations in contracts where appropriate. For proprietary third-party components, you may not have full internal visibility, so document the boundary and supplier assurance. A founder should understand that software supply-chain risk is also contract risk: you can promise a customer response time that your upstream vendor is not obligated to support.

Turn SBOM readiness into enterprise trust

Security questionnaires increasingly ask about secure development, dependency management, vulnerability response and SBOM capability. A small company that can produce a current SBOM, explain how it is generated, show who reviews vulnerabilities and describe how affected customers are notified looks operationally mature. Avoid marketing claims that imply the SBOM guarantees security. Instead, present it as part of a repeatable process. That distinction matters because enterprise buyers value evidence and repeatability more than buzzwords.

The 30-day implementation

Week 1: inventory repositories, package managers, containers and release pipelines. Week 2: choose an SBOM format and generation tool supported by your environment, then generate a baseline for one product. Week 3: integrate generation into the release process and test how quickly you can answer 'do we use component X at version Y?' Week 4: define ownership for vulnerability review, supplier escalation, customer notification and SBOM delivery. Run a tabletop using a real historical vulnerability. If it takes hours to identify affected versions, improve the inventory before a live incident forces the issue.

Practice the question customers will actually ask

Run a monthly component-response drill. Pick a real package in your environment and pretend a severe vulnerability was announced at 9:00 a.m. Ask the team to answer five questions by 10:00: Which released products contain the component? Which versions? Is the vulnerable code reachable or used in the affected configuration? Which customers have those versions? What mitigation or patch path exists? Time the exercise. If the SBOM names the component but cannot connect it to a shipped release, improve release provenance. If you know the release but not which customers run it, improve deployment inventory. If a supplier component has no escalation contact, fix the contract and vendor record. Then create a customer-facing response template with fields for affected versions, exploitation assessment, temporary mitigation, fix availability and next update time. Do not pre-fill claims. The value of SBOM practice is operational compression: turning a half-day scavenger hunt into a defensible answer in minutes. For a small software company trying to win enterprise work, that response speed is a competitive signal because the buyer is evaluating whether your team can manage the next vulnerability after the contract is signed.

Choose one SBOM owner

Assign responsibility for SBOM generation, release attachment and vulnerability queries to a named role. Security can define the process, but engineering and release management usually own the data needed to keep it current. Put the responsibility in the release checklist. An automated file nobody reviews is not a supply-chain capability.

Research behind this guide

Use the primary sources below to verify current rules, eligibility and program details before acting. Program terms can change.

CISA — 2026 Minimum Elements for a Software Bill of Materials



CISA — Software Bill of Materials



CISA — Securing the Software Supply Chain



NIST — Secure Software Development Framework



Important: This article is educational information, not individualized legal, tax, investment or compliance advice. For decisions with legal or financial consequences, use a qualified professional who can review your facts.

SHARE THIS GUIDE ↗

POWERED BY: BLACK TECH STARTUP

blacktechstartup.com