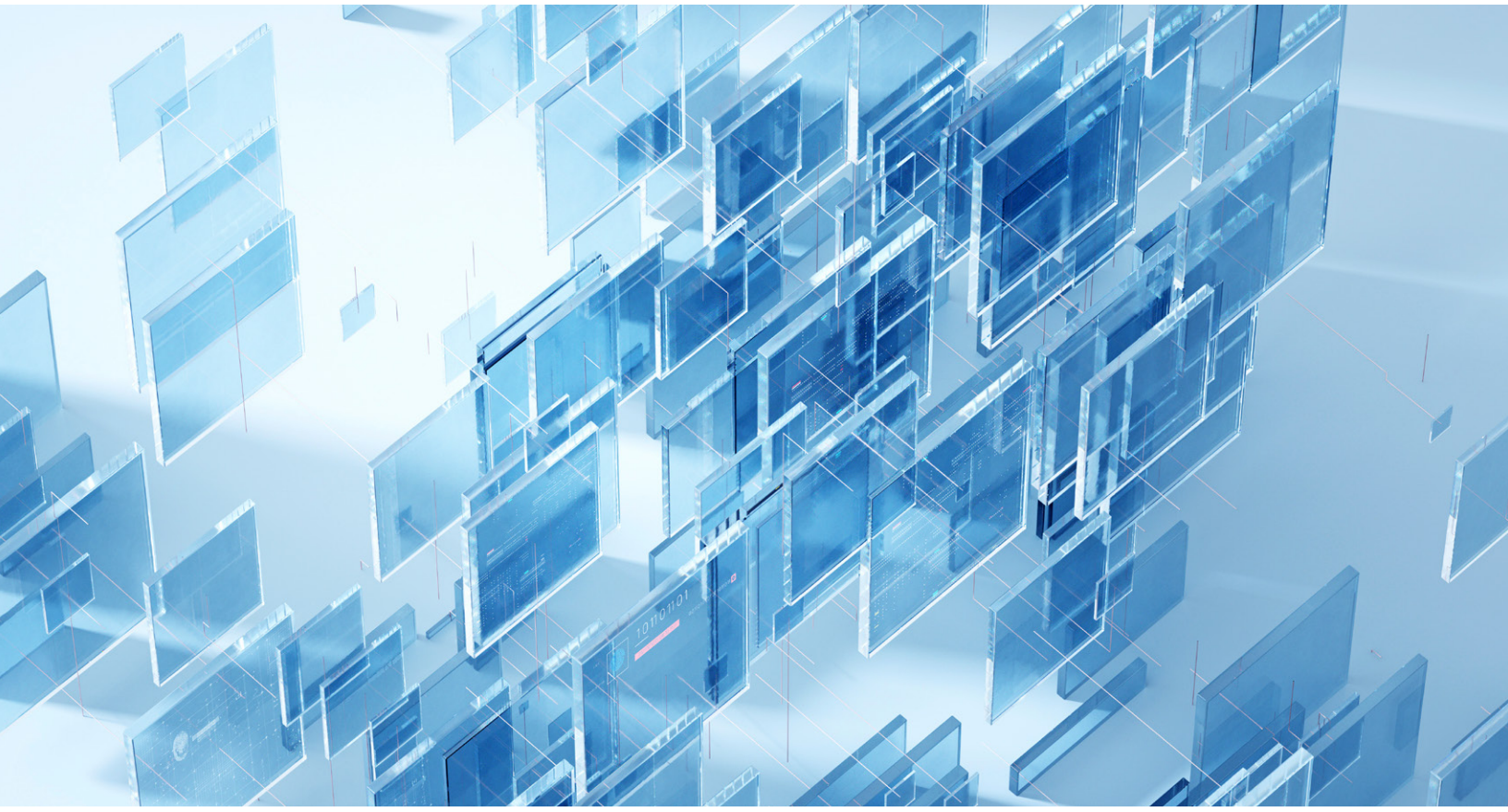
McKinsey & Company

Risk & Resilience Practice

Software bill of materials: Managing software cybersecurity risks

As software-related vulnerabilities continue to grow, companies must manage their software cyberrisks to innovate faster and create safer, more secure digital products.

by Tucker Bailey, Justin Greis, Matt Watters, and Josh Welle



September 2022

Until recently, most companies were unaware of the "ingredients" or code that make up the software that powers their products and enterprise software. This is an issue because third-party code usage is increasing, and the consumption of open-source software (OSS) will accelerate in the years to come.

Companies leverage OSS because it reduces costs and increases the pace of software development. By layering in code that someone else has already built, developers can decrease their time to market and accelerate the feature sets most desired by their business partners. The challenge is that the code in the OSS repository may have embedded malware, bugs, or other vulnerabilities unbeknownst to the developer. Without a robust vetting process for the code in the OSS repository from which their developers are pulling, companies will remain unaware of threats lurking in their products.

To keep pace with the evolving security dynamic, companies need new programs and management techniques to understand the origin and security of the code that underpins their digital products and business operations. Without such a system, companies will remain vulnerable to unwittingly inserting malware that could cause cybersecurity incidents or damage critical applications. While some cyberthreats can be avoided by developing custom code in-house, that process is resource intensive and time-consuming. Moreover, using OSS has benefits such as being able to build quickly in the cloud or increasing developer productivity. In reality, most applications are built using a combination of custom code and open-source components. That is when a delicate balancing act falls on chief technology officers (CTOs), chief information officers (CIOs), and chief information security officers (CISOs) who are sensitive to OSS's inherent risks.

A software bill of materials (SBOM) program can help companies protect themselves and their customers by creating a system that vets all incoming code before it can be adopted by developers.

What an SBOM is, and why it helps

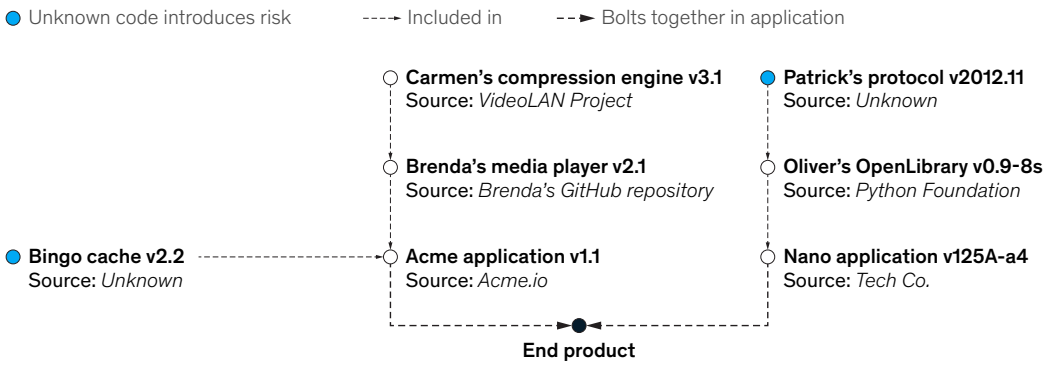
An SBOM is like the nutritional label on a cereal box—it lists the ingredients inside, highlighting content that may be harmful to some, such as gluten and peanuts. Specifically, an SBOM is a formal, machine-readable inventory of software components and dependencies (which result from combining various OSS components, third-party code, and code developed in-house), technical information about those components, and the code's hierarchical relationships (Exhibit 1).

To keep pace with the evolving security dynamic, companies need new programs and management techniques to understand the origin and security of the code that underpins their digital products and business operations.

Exhibit 1

Developers often include libraries or packages in software without knowing their origin; a software bill of materials lists a software's components.

Software bill of materials, illustrative



An SBOM helps developers know the origin of the "code behind the code" so they can make a determination on whether it is secure. It also helps them understand and mitigate known vulnerabilities in code, saving time and costs. SBOMs also help legal and compliance departments identify license history and allowable use cases for a piece of third-party code, which reduces any potential misuse. Furthermore, SBOMs help security and forensic teams identify the impact on software after the discovery of newly identified common vulnerabilities and exposures (CVEs), which improve organizations' ability to respond and remediate.

Why an SBOM program is necessary

During the past five years, OSS development rose from approximately 35 percent to about 75 percent of organizations' audited codebase.[1] As a case in point, by 2021, according to OpenUK, nine out of ten United Kingdom–based companies reported using OSS.[2] Organizations use OSS because it helps with cost savings, developer flexibility, and coding speed.

OSS usage creates opportunity for greater developer collaboration and allows coders to move quickly, because code libraries contain limitless amounts of prebuilt functionality and tooling resources. Having software components available on demand allows developers to leverage other developers' work. Components are accessible and available from any location, and these elements are independent of individual manufacturers, making them attractive to start-ups and emerging technology players—or essentially to anyone who wants to build software quickly.

Although the benefit of incorporating third-party code is clear, OSS contributes to organizational risk. OSS is not regulated or overseen by a central authority and is publicly available. It can contain potential vulnerabilities, out-of-date code, and cyber exploits, which can expose organizations to cyberattacks. Most organizations seek to better understand and reduce their cyber and technology risks, but organizations recognize building and maintaining secure code is a vital cornerstone of any cybersecurity strategy.

[1] "Forrester Wave: Software composition analysis, 2021," Forrester, August 18, 2021.
[2] "State of Open: The UK in 2021 phase three: The values of Open," OpenUK, October 2021.

Organizations no longer question if they will be impacted by a cyberattack, but instead ask when, how, and what the impact will be after the cyberattack occurs. In 2014, 62 percent of organizations reported experiencing a cyberattack. By 2021, that number grew to 86 percent. The cost of cyberattacks is growing rapidly as well. In 2015, the global impact of cybercrime was estimated at $3 trillion, according to Verizon's *Data Breach Investigations Report*. By 2021, that number increased to approximately $6 trillion and is expected to grow to about $10 trillion by 2025. Of the thousands of daily cyberattacks, supply chain vulnerabilities are increasingly the cause.

Given the benefits of OSS as well as its inherent risks, companies are finding it increasingly challenging to manage their software supply chains. For example, in December 2020, the Sunburst malware exploit proved to have supply chain origins. The incident impacted about 33,000 users, of which 18,000 had the product infected with the malicious code.

SBOMs enable companies to preserve the benefits of OSS while managing the risks. They help companies ward off issues, as in the Sunburst case, where one company that had SBOM capability was able to determine that it was vulnerable to the tainted code within two hours. Others affected were less fortunate, such as an energy company that determined it had the bad software two and a half months later. For that two and a half months, the company was vulnerable to even more attacks and enterprise compromise.

Cybercriminals and other bad actors often employ various methods to launch software supply chain attacks, often targeting the open-source or common third-party code. To that end, in 2018, researchers uncovered 12 malicious Python libraries uploaded on the official Python Package Index (PyPI).[3] Taken together, incidents such as the Sunburst malware case and the Python Package Index compromise have spurred companies to consider quickly developing SBOMs (see sidebar, "The federal government wants to protect the nation from malicious code").

[3] Catalin Cimpanu, "Twelve malicious Python libraries found and removed from PyPI," ZDNET, October 27, 2018.

The federal government wants to protect the nation from malicious code

Software bill of materials' (SBOM) importance has reached the US government, and it wants to get greater levels of security.

In the wake of incidents that occurred in May 2021, White House Executive Order 14028 outlined the importance of organizations having an SBOM program.[1] The Department of Commerce issued a statement identifying minimum elements of an SBOM for software sales to federal agencies.[2] The National Institute of Standards and Technology (NIST) provides guidance on Cybersecurity Supply Chain Risk Management (C-SCRM) practices, including information on SBOM standards.

The increase in government policy initiatives reflects a multiyear process whereby NIST worked with stakeholders to pilot SBOM concepts, executed proof of concepts in the healthcare industry, and assembled resources to help organizations build SBOM programs.[3]

[1] "Executive order on improving the nation's cybersecurity," White House Briefing Room, May 12, 2021.
[2] "The minimum elements for a software bill of materials (SBOM)," National Telecommunications and Information Administration, July 12, 2021; minimum components include data fields, automation support, practices, and processes.
[3] "Marking the conclusion of NTIA's SBOM process," National Telecommunications and Information Administration, February 9, 2022; "Software bill of materials," National Telecommunications and Information Administration.

How SBOM programs mitigate risks with increased code transparency

Most organizations have applications made up of different subcomponents and pieces, some taken from OSS libraries, others purchased from third parties, and some created and customized by developers. Each source has different limitations associated with it. For example, OSS and third-party software have licenses that change over time or may have usage limitations. A typical SBOM system surveys, identifies, and characterizes these coding elements (Exhibit 2).

Organizations that struggle to understand vulnerabilities within their code open themselves to security or financial risk. Here are several benefits an organization can experience from creating an SBOM program:

— *Vulnerability identification and mitigation.* SBOM systems inventory code subcomponents and allow analysts to identify code with known vulnerabilities. When a critical vulnerability is announced, security analysts reference SBOM artifacts and determine if there is an impact on code. This helps incident-response personnel identify where vulnerabilities exist and prioritize mitigations. It also provides leaders with the ability to respond quickly and confidently when dealing with board members, customers, investors, and regulators.

— *Software license management.* SBOM programs help organizations manage their third-party software and OSS licenses, which can be quite complex and change over time. This ensures developers are consistently able to determine the permissible use cases for OSS and third-party software, ultimately protecting organizations from financial risk stemming from inappropriate or unauthorized use of third-party software. It also helps compliance teams respond to license claims or audits.

— *Software development life cycle (SDLC) improvement.* SBOM programs can help organizations improve their SDLC. Generating an SBOM at preset times during the SDLC can identify problems like known vulnerabilities or license issues. Teams mitigate these issues early, reducing costs and preventing delays. SBOM programs allow organizations to become more efficient and accelerate how they build, deploy, and run software.

Exhibit 2

Developing clean code involves considering software bill of materials best practices at each stage of the process.

Integration of software bill of materials (SBOM) best practices into development process

1. Design *The design for the code recognizes areas that may involve dependencies*

2. Code *Open-source dependencies are integrated into code*

3. Build *The initial SBOM is created with a list of dependencies*

4. Test *Testing identifies code vulnerabilities in the program*

5. Package *Dependencies are collected to prepare for deployment, and SBOM is created*

6. Deploy *Code is put into production with a reference to SBOM*

7. Operate *SBOM updated as the software is patched; if a vulnerability emerges, then the SBOM will create an alert*

For example, as code is built, developers create an initial SBOM, which lists the dependencies inherent in the software. During testing and packaging, vulnerabilities and additional dependencies are identified and the SBOM is updated as the code is deployed. As vulnerabilities emerge during operations, the SBOM program will alert developers when a patch is required. After patching, the SBOM is updated again.

How to develop an SBOM program: Invest, create, integrate, and design

Realizing the benefits and challenges associated with OSS development, it is clear that SBOM programs are integral to safer coding and a resilient enterprise. To that end, organizations can focus on four key practices when establishing SBOM programs:

1. *Invest in SBOM capabilities by building on existing software composition analysis (SCA) tools.* Many organizations have SCA capabilities in place. Depending on how robust the capabilities are, teams can use this as a foundation to build an SBOM program. Leadership can utilize resources efficiently by determining which SBOM tools can be purchased versus developed in-house. Product leaders should find a trusted vendor and build a program using tools that fit with SDLC processes, including smooth and automated integration with a continuous integration and continuous deployment (CI/CD) pipeline. Automated SBOM capabilities can be developed in-house or purchased from a third-party vendor.

2. *Create an SBOM program with a cross-functional team.* SBOM programs involve multiple parts of the organization—each with a different role to play. It is important to launch an SBOM program using a cross-functional team. This team should include personnel from various backgrounds—developers, information security, procurement, legal, risk, privacy, and compliance, because SBOM programs touch multiple parts of the organization. Each participating team should have representation at decision-making levels, and their needs should be accounted for throughout the process (Exhibit 3).

Exhibit 3

Successful software bill of materials programs work cross-functionally across multiple parts of an organization.

Software bill of materials (SBOM) program integration, not exhaustive

Developers
Integrate SBOM into software development life cycle to identify and remediate vulnerabilities more rapidly

Legal
Develop and implement standardized contract clauses requiring SBOM use and service-level agreements

Risk
Analyze SBOM to identify known vulnerabilities and balance cost vs known risk

Compliance
Implement governance structure and policies to ensure SBOM use throughout organization

Privacy
Create processes to ensure confidentiality of, and access to, certain information about an entity is protected

Procurement
Work to ensure vendors supply SBOM for software they provide

Infosec[1]
Identify code that may be impacted when vulnerability is identified

[1]Information security.

3. *Integrate SBOM into the SDLC.* SBOM programs provide benefits to developers by identifying vulnerabilities and licensing issues early in the process. Organizations can benefit most when they examine their development processes and integrate SBOM generation and review at multiple points in the development life cycle. Building automated SBOM generation and review capabilities reduces workload and encourages developers to adopt new ways of working.

4. *Design SBOM governance.* SBOM programs require cross-functional collaboration and communication and may require additional structure to incentivize adoption, compliance, and cooperation. Organizations that have a decentralized or nonhierarchical culture may find it difficult to have developers comply with SBOM requirements. Security benefits and SDLC efficiency help generate increased developer buy-in. Also, organizations can consider a governance structure that assigns roles and responsibilities for SBOM-related tasks.

Launching an SBOM program

Companies are racing to use third-party code, including OSS, which means creating an SBOM program must follow suit to avoid the challenges befalling many in the business world today. In addition to this article, guidance on creating a program is available at the Cybersecurity and Infrastructure Security Agency (CISA) and the National Institute of Standards and Technology (NIST) developer forums.[4] SBOM regulations primarily affect companies delivering services to the government, like aerospace and defense, but they can be helpful guides for all companies looking to mitigate the same issues. Highly regulated industries, like utilities and financial services, often have the same onus to monitor OSS software usage.

Looking ahead, organizations that proactively examine and assess code that enters their ecosystem via a robust SBOM program will be better prepared for impending government regulations on software supply chain and keep their IT infrastructure and customers safer. Strong SBOM programs allow organizations to build secure code in a cost-effective and efficient manner in the face of increasing cyberthreats. The time to act is now.

Tucker Bailey is a partner in McKinsey's Washington, DC, office; **Justin Greis** is a partner in the Chicago office; and **Matt Watters** is an associate partner in the New Jersey office, where **Josh Welle** is a consultant.

The authors wish to thank Jim Boehm, Dennis Dias, Michael Glynn, Alex Keegan, Charlie Lewis, Lucy Shenton, and Daniel Wallance for their contributions to this article.

Designed by McKinsey Global Publishing

[4] See "Software bill of materials," Cybersecurity and Infrastructure Security Agency, accessed September 2022.