A Software Bill of Materials Is Critical for Comprehensive Risk Management

By Dr. Georgianna Shea

Executive Summary

In today's world, very little software is entirely original. Software developers use existing, open-source, and commercially available software components to create new products. Programmers are not trying to reinvent the wheel; they leverage blocks of already developed code for time and cost efficiency. Collaboration on code development and reuse of software is a standard practice that is enabled and encouraged. On average, 75 percent of a software product is open-source code, according to the *2021 Open-Source Security and Risk Analysis Report*.[1]

This presents a cyber-risk management problem. The problem is not the use of open-source software per se, but that customers generally receive software products without understanding the nested software contained within them. Customers are, in effect, purchasing a box of cereal without knowing if it contains nuts, wheat, soy, or other standard ingredients, even though those customers may have a severe allergic reaction to nuts. The customer cannot effectively manage assets and risk without knowing the software's contents, origins, and history of changes and who made those changes.

A solution to this problem is to provide customers with a Software Bill of Materials (SBOM). An SBOM is a list of nested software components, designed to enable supply chain transparency.[2] The SBOM identifies the component software and facilitates analysis and auditing of the components to determine risk and compliance. SBOMs have always been a good idea but not a requirement, and buyers often do not know to ask for them.

Luckily, that may be changing. President Joe Biden's May 2021 executive order (E.O.) on cybersecurity, E.O. 14028, explains that "[b]uyers can use an SBOM to perform vulnerability or license analysis, both of which can be used to evaluate risk in a product. Those who operate software can use SBOMs to quickly and easily determine whether they are at potential risk of a newly discovered vulnerability."[3]

Without an SBOM, companies cannot take the first steps to secure themselves. The National Institute of Standards and Technology (NIST) Cybersecurity Framework explains that a foundational step to cybersecurity and risk management requires identifying data, personnel, and systems.[4] Before an organization can *protect* itself, before it can *detect* anomalies on its network and devices, the organization must identify its software and the software's components before responding to indicators of a breach. If an organization does not know what its software contains, it should assume that the software is compromised and develop an appropriate risk management plan.

To aid the public and private sectors' understanding of the utility of SBOMs, FDD's Transformative Cyber Innovation Lab (TCIL) walked through the paces of developing and analyzing an SBOM. This first-hand perspective enables TCIL to provide concrete lessons learned rather than general recommendations. In this effort, TCIL collaborated with Virgil Systems, a company specializing in trusted data communications in a zero-trust world, and ION Channel, which specializes in the software supply chain. This report outlines the process used in, and the lessons learned and best practices revealed by, TCIL's pilot project.

1. For comparison by sector, see Appendix A.

2. U.S. Department of Commerce, National Telecommunications and Information Administration, Multistakeholder Process on Software Component Transparency Framing Working Group, "Framing Software Component Transparency: Establishing a Common Software Bill of Material (SBOM)," November 12, 2019. (https://www.ntia.gov/files/ntia/publications/framingsbom_20191112.pdf)

3. The relevant text of the executive order and the responses from agencies charged with implementing provisions related to SBOMs can be found in Appendix B.

4. For more information on the five functions in the NIST Framework and on the importance of an SBOM to the Identify Function, see Appendix C.

A Software Bill of Materials Is Critical for Comprehensive Risk Management

An important finding of the pilot is that having an SBOM is only the first step. Having a list of ingredients enables further analysis, but without that analysis, an SBOM is just a list. Critical next steps include understanding the software's dependencies and vulnerabilities, ensuring continuous monitoring so that new risk information is ingested, and creating an immutable auditing capability to ensure the integrity of the data.

The Real-World Effects of Software Vulnerability

The security of software starts with the software development lifecycle. During a software build, a community of developers will work on portions of the code and then integrate them with open-source code and other dependency software. As such, software development requires configuration management, which records who added what and when they added it. Software developers need configuration management, for example, to ensure they have proper licensing to reuse existing code. A secure configuration-management process should have a chain-of-custody-like process that includes a hash value (a unique identifier similar to a fingerprint) to ensure integrity as the code traverses the software development lifecycle.

Conducting risk assessments of the software throughout its entire lifecycle becomes much more complex when open-source software uses closed-source binaries in the form of dynamically loadable libraries. To address this challenge, software developers could use advanced artificial intelligence capabilities (such as deep learning of binary patterns and vulnerabilities) to analyze binaries and provide a list of leading indicators of potential vulnerabilities that may result in Common Vulnerabilities and Exposures (CVEs) or publicly known security vulnerabilities. In this way, the SBOM would include closed- and open-source measures of risk and leading indicators of a future CVE.

The developer's and supplier's lack of integrity and security controls has real-world implications for the consumer. For example, the SolarWinds cyber breach revealed in December 2020 left tens of thousands of organizations at risk and vulnerable. At some point during the build process for the company's Orion software, hackers believed to be operating at the direction of Russia's Foreign Intelligence Service secretly inserted unauthorized third-party software into the build process. When SolarWinds pushed a software update to its customers, the patch contained hidden malicious code. Had the developers used an integrity-focused tool such as a blockchain-based SBOM with an immutable, auditable capability (which automatically fingerprints and records authorized code additions) integrated into the configuration-management process of the development pipeline, they would likely have detected the injection of malicious software. Had they created and monitored the hash values on the software segments, the injection of unauthorized software would have changed the fingerprint (or hash value) within a block or created an unauthorized new block in the chain. Such a change would have set off alarms, leading SolarWinds to investigate the discrepancy. The company could have halted the distribution of the updates instead of pushing a compromised product to its customers.

Using nested code in a company's website may also expose that company to security vulnerabilities, legal liability, and reputational harm. Websites contain about 70 percent third-party code[5] and often use this code to perform core functions of the website.[6] For example, a retail website will have a purchasing function that allows for credit card transactions. Who will the customer hold accountable if his or her credit card information gets stolen because of vulnerabilities in the third-party code? Target paid millions of dollars to settle claims after hackers breached a third-party vendor to steal millions of credit card numbers from the company in 2013.[7] Customers will blame the retail company, not the third-party software provider.

5. "There's A Lack Of Awareness About Malicious Third Party Code," *MacTech*, October 11, 2019. (https://www.mactech.com/2019/10/11/theres-a-lack-of-awareness-about-malicious-third-party-code)

6. The software, referred to as scripts, employs multiple types of code in the digital environment to perform various functions. Some code depends directly on the functions of the website.

7. "Target Settles 2013 Hacked Customer Data Breach For $18.5 Million," *Reuters*, May 24, 2017. (https://www.nbcnews.com/business/business-news/target-settles-2013-hacked-customer-data-breach-18-5-million-n764031)

A retail website might also use another piece of third-party code to collect customers' browsing and purchasing history to sell to advertisers. The company accepts unknown risk if it does not know whether that same component is, for example, collecting and selling data that also identifies people's likely political affiliation based on their browsing and purchasing history.

Piloting the Solution: Developing and Analyzing an SBOM

Recognizing a systemic problem and an existing but underutilized solution, TCIL set out to demonstrate the development and utility of an SBOM. Initially, the pilot started with one objective: to create an SBOM so that TCIL could offer first-hand lessons learned from that process. After the team — consisting of software supply chain experts, data integrity experts, and a threat expert — collaborated on the objective, it became clear that the team needed to analyze the SBOM and build an immutable, auditable capability to ensure the integrity of the SBOM and the underlying software.

Step 1: Identify the Software: The first step in developing the SBOM was identifying the software package for the pilot. The team chose a third-party software program publicly available on GitHub and currently used by the Department of Defense (DoD). The team also chose software with no documented CVEs to show that using an SBOM can reveal potential risks contained in a software package with no known vulnerabilities.[8]

Step 2: Download the Data and Create an SBOM: After downloading the publicly available files, ION Channel used an automated process to create the SBOM. Downloading the information from GitHub and creating the SBOM with the available information took a few minutes.[9] However, generating an SBOM by hand would have been significantly more time-consuming and would have added a layer of questionable integrity. This is itself an important lesson and reminder for buyers. Indeed, in July, the National Telecommunications and Information Administration (NTIA) issued guidance on the minimum elements of an SBOM in accordance with Biden's May 2021 executive order on cybersecurity, noting that SBOMs should be automatically generated and machine-readable.[10]

Although TCIL conducted the pilot before NTIA publicized the baseline attributes of an SBOM (see Table 1), the pilot's SBOM meets NTIA's recommendations regarding fields, automation, and process.[11] To determine the data fields TCIL would include in its SBOM, the team first identified what type of analysis was needed to develop a risk management plan following the NIST Cybersecurity Framework. Knowing the information requirements of the framework, the team determined the attributes needed to conduct the analysis and ensured that the SBOM contained those elements. Experts at NTIA no doubt conducted a similar process.

8. GitHub is an online platform that hosts remote software repositories and collaboration amongst the development team and strangers worldwide. Sites such as GitHub also allow developers to download and use open-source software.

9. All configuration-management repositories should have the information necessary to build an SBOM. GitHub's software repositories are configured to easily extract the relevant information. If an organization does not have the tools or knowledge to develop its SBOM, companies such as ION Channel can provide a metered service to create SBOMs, analyze the results, and conduct continuous monitoring.

10. U.S. Department of Commerce, National Telecommunications and Information Administration, "The Minimum Elements for a Software Bill of Materials (SBOM) Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity," July 12, 2021. (https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf)

11. Specifically, after downloading the publicly available files from GitHub, ION Channel used an automated process to ingest a CSV file format into ION Channel's custom-developed templates, which mapped the various fields to ION Channel's analysis engine. This resulted in a metadata tagged JSON formatted document that a developer or consumer could use as an SBOM.

A Software Bill of Materials Is Critical for Comprehensive Risk Management

Table 1: Minimum Elements of an SBOM[12]

Minimum Elements	
Data Fields	Document baseline information about each component that should be tracked: Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, and Timestamp.
Automation Support	Support automation, including via automatic generation and machine-readability, to allow for scaling across the software ecosystem. Data formats used to generate and consume SBOMs include SPDX, CycloneDX, and SWID tags.
Practice and Processes	Define the operations of SBOM requests, generation, and use, including Frequency, Depth, Known Unknowns, Distribution and Delivery, Access Control, and Accommodation of Mistakes.

The attributes of the SBOM dictate the type of analysis that can be done.[13] A U.S. government contractor, for example, cannot verify that it uses no Huawei products if the contractor does not know the supplier of the software nested within the package it uses. Therefore, a company that wants to receive an SBOM from a software vendor should ensure that the contract specifies the minimum elements the SBOM must include for the company to conduct the analysis it needs.[14]

During the initial receipt of an SBOM, it is imperative for the customer to require validation of the SBOM's format and authoritative naming of components as a condition of the company's acceptance of the SBOM. Doing so is necessary to prevent invalid formats and low data quality from thwarting the intent of statutory mandates or contractual obligations for software transparency. If the format is invalid or the names of components are not authoritative (for example, external IDs that can map to known vulnerabilities and points of origin), the SBOM should be rejected, and the supplier should correct and resubmit.

Step 3: Analyze the SBOM: After creating an SBOM with the minimum attributes recommended by NTIA, the team scanned the data to thoroughly analyze the complete software package and the nested component software.[15] This analysis revealed that even a CVE-free software can have:

- Nine direct component dependencies, which then had 700 component dependencies;
- 19 dependencies with no version identified;
- Four critical software development vulnerabilities (based on the Common Vulnerability Scoring System);
- Seven high software development vulnerabilities (based on the Common Vulnerability Scoring System); and
- Two low software development vulnerabilities (based on the Common Vulnerability Scoring System).

A timeline showed the software being non-compliant (to predetermined criteria), then coming back into compliance, then going out again. This represented regularly updated and maintained software, proving that security is usually only a snapshot in time.

The results of an SBOM analysis will inevitably include vulnerabilities. Customers should not expect an SBOM analysis report to be vulnerability- and risk-free. Rather, the purpose of the results is to enable the decision-maker to understand what he or she is receiving and to make an informed risk management plan. For example, customers using a certain software in weapons systems will have a different risk tolerance than those using the same software in accounting systems.

12. U.S. Department of Commerce, National Telecommunications and Information Administration, "The Minimum Elements for a Software Bill of Materials (SBOM) Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity," July 12, 2021, page 3. (https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf)

13. Appendix D explains the connection between the SBOM's attributes and the analysis that the acquirer can conduct.

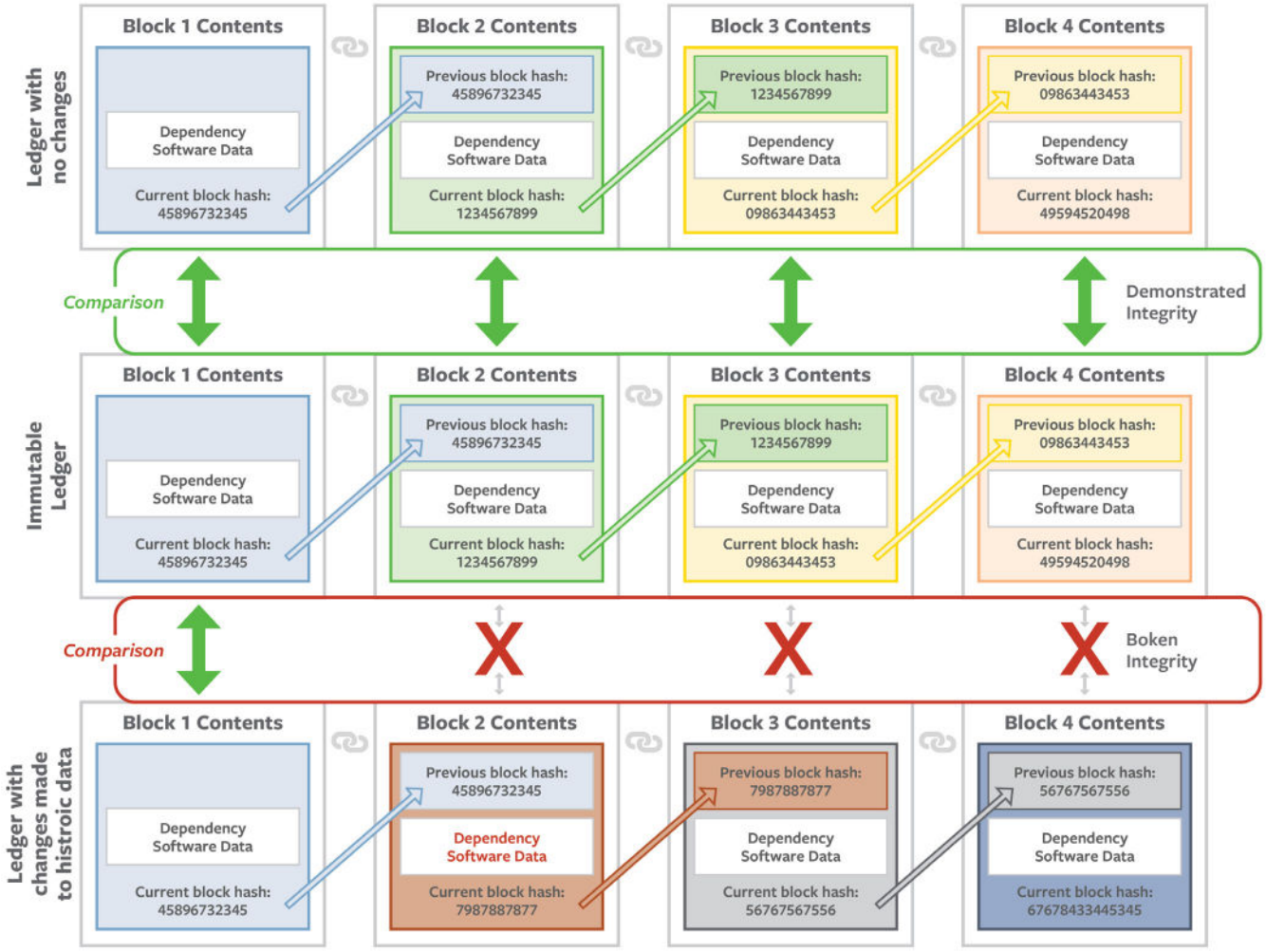
14. A company's business needs, along with governance requirements, will inform the analysis that the company should conduct.

15. A user can import the data into a preferred format to manually conduct the analysis or use a provider to perform an automated, robust analysis. ION Channel has the latter capabilities and can graphically and logically display the results based on a pre-established ruleset.

A Software Bill of Materials Is Critical for Comprehensive Risk Management

Step 4: Ensure the Integrity of the SBOM: The team then used blockchain to create an immutable and auditable record of the SBOM's contents, history of changes, and provenance. Doing so enabled the team to ensure the integrity of the SBOM. Blockchain is a series of data records that are linked together through a process that establishes integrity. Each record, or "block," contains the hash (the numeric value) of the previous record in the series. With blockchain technology applied to an SBOM, the data of each software dependency is recorded, hashed, and stored. Each change or addition to the software creates a new block in the chain without altering existing records.[16] If an actor tries to alter the existing record, the change will appear as an altered hash value for all subsequent blocks in the chain. Comparison to the original ledger will reveal that something is amiss. (See Figure 1)

Figure 1: Blockchain Concept



16. An SBOM will contain a variable number of blocks and can quickly become very large. As noted, the pilot discovered over 700 dependent component software packages. Not all dependencies were linear. When implementing blockchain for integrity, a scalable solution that can handle complex graph-based ledgering due to the interrelated nature of software dependencies is needed.

For example, if an original software package had a component acquired from Huawei Technologies, the SBOM recorded through blockchain would include that information. If the supplier attempted to remove the data indicating the provenance of that software component, the blockchain would detect a change to the hash value. The consumer could then see that the SBOM received does not match the publicly available ledger, meaning there is a problem.

The use of blockchain as an integrity solution must be paired with a restoration process. The blockchain can detect a change — be it the addition of a period to a record or the removal of Huawei identifiers — but it cannot identify the source of the change. Auditors will also need a restoration process to understand the significance of the change.

Recommendations

On July 12, 2021, NTIA issued its guidelines for the minimum elements for an SBOM.[17] These guidelines are an essential first step. Software developers should adopt the minimum standard, and customers should require vendors to provide this information.

However, from the federal government's perspective, more can be done to ensure the adoption and effective use of SBOMs. The most efficient way to ensure SBOM adoption would be to update the Federal Acquisition Regulation and the Defense Federal Acquisition Regulation Supplment to include SBOM contracting language. As part of Cyber Test and Evaluation, the contract would require software assurance beyond the static code analysis for critical software. Instead, contracts would require machine-readable SBOMs, automated continuous monitoring of SBOMs, and evidence of continuous monitoring to ensure everyone adopts it. These requirements align with Section 4 of the president's May 2021 executive order.

Additionally, the contracts would require validation of SBOM format and authoritative naming of components as a condition of acceptance of an SBOM, to prevent invalid formats and low data quality from thwarting the intent of statutory mandates for software transparency. The government would then check the validity of the format and the authoritativeness of the underlying software and reject and require resubmission of SBOMs that fail the automatic check at the procurement point.

However, a rapid change to contract requirements risks a scenario in which suppliers provide invalid and unauthoritative SBOMs as a box-checking exercise, creating a mountain of garbage data worthless for vulnerability management. With that risk in mind, a gradual approach may be more effective in the long run. Therefore, the federal government should first promote and incentivize SBOMs rather than mandating them on a broad basis.

1. **Continue to refine SBOM guidance:** NTIA or NIST should issue guidelines to ensure that SBOMs have the flexibility to incorporate new data sets and have immutable auditability and scalability. The guidelines should also recommend that SBOMs be machine-readable and continuously monitored. NTIA or NIST should explore Zero-Trust concepts in the architecture and design principles around the system structures supporting SBOMs.
2. **Help the private sector understand and adopt SBOMs:** While this pilot showcased the relative ease and critical importance of using SBOMs, the concept is new to most private-sector actors. The U.S. government should help establish private and public working groups to continue to shape policies regarding SBOMs. The U.S. government should also help private-sector entities, associations, and information sharing councils to include SBOM requirements in contracts during procurement and to include SBOM analysis within risk assessment procedures. These working groups should explore strategies to incentivize rapid adoption by communities of interest and should engage non-traditional stakeholders, including city, county, and state leaders. The U.S. government should also establish public-private partnerships capable of conducting continuous monitoring of SBOMs.

17. These are included in Appendix B. U.S. Department of Commerce, National Telecommunications and Information Administration, "The Minimum Elements for a Software Bill of Materials (SBOM) Pursuant to Executive Order 14028 on Improving the Nation's Cybersecurity," July 12, 2021, page 3. (https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf)

3. **Move toward requiring SBOMs in all relevant government contracts:** As noted above, while the quickest means to achieve SBOM adoption would be to update the Federal Acquisition Regulation and the Defense Federal Acquisition Regulation Supplement with SBOM contracting language. Adding new requirements can be burdensome for private industry when those requirements are not well understood and communicated. While SBOMs are not new, a phased approach rather than an immediate regulatory update may be more likely to gain industry acceptance. Departments and agencies across the federal government should begin testing the use of SBOM requirements wherever possible. Within DoD, this could take the form of the following types of steps:
   - DoD Chief Information Security Officer: Include SBOM criteria in Cyber Maturity Model Certification requirements.
   - DoD Cyberspace Division, Joint Staff/J-6: Update the System Survivability Key Performance Parameter for cyber to include "Identify" alongside the pillars Prevent, Mitigate, and Recover, and then add the Cyber Survivability Attribute: SBOM analysis.
   - DoD Acquisition and Sustainment (A&S) Policy: Link SBOM requirements and analysis criteria to Cyber Dependency as defined in the Cyber Survivable Endorsement Guide. Update Department of Defense Instruction (DoDI) 5000.90, "Cybersecurity for Acquisition Decision Authorities and Program Managers."
   - DoD A&S Policy: Require an SBOM in the Software Acquisition Adaptive Acquisition Pathway. Require an SBOM for Urgent Acquisition, Middle Tier of Acquisition, Major Capability Acquisition, and Defense Business Systems, based on the Cyber Dependency Level. Update DoDI 5000.90, "Cybersecurity for Acquisition Decision Authorities and Program Managers."
   - DoD Research and Engineering, Developmental Test and Evaluation, and Assessments: Include SBOM criteria and recommendations in the DoD Cybersecurity Test and Evaluation Guidebook.
   - DoD Joint Federated Assurance Center: Establish a repository of all DoD-acquired SBOMs. Continuously monitor and share vulnerability information.
   - DoD Defense Information Systems Agency: Update Cybersecurity Service Provider Standards and Evaluator Scoring Metrics to include SBOM as a part of the Identity function and SBOM analysis under the vulnerability analysis task.
   - Defense Intelligence Agency, Threat Analysis Center: Conduct an intelligence review of SBOM.
   - Office of the Secretary of Defense: Incentivize rapid technology transition, rapid artificial intelligence and machine learning technology, distributed ledger technologies, and Zero-Trust as core enabling concepts to reduce burdens, costs, and systemic bottlenecks or overloads.

Conclusion

Software in any form is a complex blend of multiple components. Without knowing the contents of the software, it is impossible to conduct an accurate risk assessment or to develop a solid risk management plan. This TCIL pilot execution demonstrated the development of an SBOM, the analysis that can be achieved from an SBOM's fields, and the recommended attributes of an SBOM. The solution to software transparency and integrity exists. Now government and industry must use it.

SBOM analysis will always identify vulnerabilities in component software. Sometimes that may lead to intellectual property and licensing breaches that will result in costly legal battles for older software development companies that did not start with secure configuration and development practices. Other times, time-critical development programs will receive risk information that will inform the purchaser's risk management plan.

But in all cases, not knowing does not make the vulnerability or liability go away. Not knowing a software's components and functions only makes organizations and consumers more vulnerable to cyberattack, exploitation, and liability. By understanding the vulnerabilities and risks, companies can make informed risk-acceptance decisions and develop mitigation and recovery plans that make enterprises resilient in the face of escalating cyberattacks.

Appendix A: Industry Sectors and Percentage of Open-Source Software

Table 2: Percentage of Open-Source Software in Codebases, by Industry[18]

Industry Sector	Percentage of Open-Source Software in Codebases
Aerospace, Aviation, Auto, Transportation, Logistics	70%
Big Data, AI, BI, Machine Learning	76%
Computer Hardware and Semiconductors	74%
Cybersecurity	84%
Education Technology	82%
Energy and Clean Tech	81%
Enterprise Software/SaaS	72%
Financial Services and FinTech	69%
Healthcare, Health Tech, Life Sciences	82%
Internet and Mobile Apps	82%
Internet and Software Infrastructure	79%
Internet of Things	89%
Manufacturing, Industrials, Robotics	84%
Marketing Tech	82%
Retail and E-Commerce	48%
Telecommunications and Wireless	57%
Virtual Reality, Gaming, Entertainment, Media	76%

18. "Open-Source Security and Risk Analysis Report," *Synopsys*, 2021. (https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html)

Appendix B: E.O. 14028 on Improving the Nation's Cybersecurity; SBOM-Related Directives and Responses From the Department of Commerce, NTIA, and NIST

DIRECTIVE PROMULGATED IN E.O. 14028, SECTION 4(F):[19]

> Within 60 days of the date of this order, the Secretary of Commerce, in coordination with the Assistant Secretary for Communications and Information and the Administrator of the National Telecommunications and Information Administration, shall publish minimum elements for an SBOM.

RESPONSE FROM DEPARTMENT OF COMMERCE AND NTIA:[20]

> The Executive Order (14028) on Improving the Nation's Cybersecurity directs the Department of Commerce, in coordination with the National Telecommunications and Information Administration (NTIA), to publish the 'minimum elements' for a Software Bill of Materials (SBOM).

Minimum Elements	
Data Fields	Document baseline information about each component that should be tracked: Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, and Timestamp.
Automation Support	Support automation, including via automatic generation and machine-readability to allow for scaling across the software ecosystem. Data formats used to generate and consume SBOMs include SPDX, CycloneDX, and SWID tags.
Practice and Processes	Define the operations of SBOM requests, generation and use including Frequency, Depth, Known Unknowns, Distribution and Delivery, Access Control, and Accommodation of Mistakes.

DIRECTIVE PROMULGATED IN E.O. 14028, SECTION 4(G):

> Within 45 days of the date of this order, the Secretary of Commerce, acting through the Director of NIST, in consultation with the Secretary of Defense acting through the Director of the NSA, the Secretary of Homeland Security acting through the Director of CISA, the Director of OMB, and the Director of National Intelligence, shall publish a definition of the term "critical software" for inclusion in the guidance issued pursuant to subsection (e) of this section. That definition shall reflect the level of privilege or access required to function, integration and dependencies with other software, direct access to networking and computing resources, the performance of a function critical to trust, and potential harm if compromised.

NIST DIRECTOR RESPONSE:[21]

> EO-critical software is defined as any software that has, or has direct software dependencies upon, one or more components with at least one of these attributes:
>
> - is designed to run with elevated privilege or manage privileges;
> - has direct or privileged access to networking or computing resources;
> - is designed to control access to data or operational technology;
> - performs a function critical to trust; or,
> - operates outside of normal trust boundaries with privileged access.

19. Executive Order 14028, "Executive Order on Improving the Nation's Cybersecurity," May 12, 2021. (https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/)

20. U.S. Department of Commerce, National Telecommunications and Information Administration, "The Minimum Elements for a Software Bill of Materials (SBOM)," July 12, 2021. (https://www.ntia.doc.gov/report/2021/minimum-elements-software-bill-materials-sbom)

21. U.S. Department of Commerce, National Institute of Standards and Technology, "Critical Software – Definition & Explanation," July 9, 2021. (https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/critical-software-definition-explanatory)

A Software Bill of Materials Is Critical for Comprehensive Risk Management

DIRECTIVE PROMULGATED IN E.O. 14028, SECTION 4(H):

> Within 30 days of the publication of the definition required by subsection (g) of this section, the Secretary of Homeland Security acting through the Director of CISA, in consultation with the Secretary of Commerce acting through the Director of NIST, shall identify and make available to agencies a list of categories of software and software products in use or in the acquisition process meeting the definition of critical software issued pursuant to subsection (g) of this section.

NIST DIRECTOR RESPONSE:[22]

> The table below provides a preliminary list of software categories considered to be EO-critical. This table illustrates the application of EO-critical software's definition to the scope of the recommended initial implementation phase described above. As noted previously, CISA will provide the authoritative list of software categories at a later date.

Category of Software	Description	Types of Products	Rationale for Inclusion
Identity, credential, and access management (ICAM)	Software that centrally identifies, authenticates, manages access rights for, or enforces access decisions for organizational users, systems, and devices	• Identity management systems • Identity provider and federation services • Certificate issuers • Access brokers • Privileged access management software • Public key infrastructure	Foundational for ensuring that only authorized users, systems, and devices can obtain access to sensitive information and functions
Operating systems, hypervisors, container environments	Software that establishes or manages access and control of hardware resources (bare metal or virtualized/containerized) and provides common services such as access control, memory management, and runtime execution environments to software applications and/or interactive users	• Operating systems for servers, desktops, and mobile devices • Hypervisors and container runtime systems that support virtualized execution of operating systems and similar environments	Highly privileged software with direct access and control of underlying hardware resources and that provides the most basic and critical trust and security functions
Web browsers	Software that processes content delivered by web servers over a network, and is often used as the user interface to device and service configuration functions	Standalone and embedded browsers	• Performs multiple access management functions • Supports browser plug-ins and extensions such as password managers for storing credentials for web server resources • Provides execution environments for code downloaded from remote sources • Provides access management for stored content, such as an access token which is provided to web servers upon request

22. Ibid.

A Software Bill of Materials Is Critical for Comprehensive Risk Management

Category of Software	Description	Types of Products	Rationale for Inclusion
Endpoint security	Software installed on an endpoint, usually with elevated privileges which enable or contribute to the secure operation of the endpoint or enable the detailed collection of information about the endpoint	• Full disk encryption • Password managers • Software that searches for, removes, or quarantines malicious software • Software that reports the security state of the endpoint (vulnerabilities and configurations) • Software that collects detailed information about the state of the firmware, operating system, applications, user and service accounts, and runtime environment	• Has privileged access to data, security information, and services to enable deep inspection of both user and system data • Provides functions critical to trust
Network control	Software that implements protocols, algorithms, and functions to configure, control, monitor, and secure the flow of data across a network	• Routing protocols • DNS resolvers and servers • Software-defined network control protocols • Virtual private network (VPN) software • Host configuration protocols	Privileged access to critical network control functions Often subverted by malware as the first step in more sophisticated attacks to exfiltrate data
Network protection	Products that prevent malicious network traffic from entering or leaving a network segment or system boundary	• Firewalls, intrusion detection/avoidance systems • Network-based policy enforcement points • Application firewalls and inspection systems	Provides a function critical to trust, often with elevated privileges
Network monitoring and configuration	Network-based monitoring and management software with the ability to change the state of—or with installed agents or special privileges on—a wide range of systems	• Network management systems • Network configuration management tools • Network traffic monitoring systems	Capable of monitoring and/or configuring enterprise IT systems using elevated privileges and/or remote installed agents
Operational monitoring and analysis	Software deployed to report operational status and security information about remote systems and the software used to process, analyze, and respond to that information	Security information and event management (SIEM) systems	• Software agents widely deployed with elevated privilege on remote systems • Analysis systems critical to incident detection and response and to forensic root cause analysis of security events • Often targeted by malware trying to deactivate or evade it

A Software Bill of Materials Is Critical for Comprehensive Risk Management

Category of Software	Description	Types of Products	Rationale for Inclusion
Remote scanning	Software that determines the state of endpoints on a network by performing network scanning of exposed services	Vulnerability detection and management software	Typically has privileged access to network services and collects sensitive information about the vulnerabilities of other systems
Remote access and configuration management	Software for remote system administration and configuration of endpoints or remote control of other systems	• Policy management • Update/patch management • Application configuration management systems • Remote access/ sharing software • Asset discovery and inventory systems • Mobile device management systems	Operates with significant access and elevated privileges, usually with little visibility or control for the endpoint user
Backup/recovery and remote storage	Software deployed to create copies and transfer data stored on endpoints or other networked devices	• Backup service systems • Recovery managers • Network-attached storage (NAS) and storage area network (SAN) software	• Privileged access to user and system data • Essential for performing response and recovery functions after a cyber incident (e.g., ransomware)

Appendix C: SBOMs Enable Functions of the NIST Cybersecurity Framework

The NIST Cybersecurity Framework (CSF) is a flexible tool used to address and manage cybersecurity risk through a repeatable and performance-based approach.[23] E.O. 13636 of 2013 promoted the CSF's adoption for critical infrastructure.[24] E.O. 13800 of 2017 required all agency heads to use the framework.[25]

The CSF consists of five functions: Identify, Protect, Detect, Respond, and Recover. Identify is the first and foundational function in the framework. Before an enterprise can protect itself, before it can detect anomalies on the network, before it can respond to indicators of a breach, it must identify its data, personnel, systems, and compliance requirements. The CSF Identify function consists of the following categories:[26]

- Asset Management
- Business Environment
- Governance
- Risk Assessment
- Risk Management Strategy
- Supply Chain Risk Management
- The composition of a company's software affects all six of the categories within the Identify function.

Function	Category	Enabling Technology
IDENTIFY (ID)	**Asset Management (ID.AM):** The data, personnel, devices, systems, and facilities that enable the organization to achieve business purposes are identified and managed consistent with their relative importance to organizational objectives and the organization's risk strategy.	SBOM
	Business Environment (ID.BE): The organization's mission, objectives, stakeholders, and activities are understood and prioritized; this information is used to inform cybersecurity roles, responsibilities, and risk management decisions.	SBOM
	Governance (ID.GV): The policies, procedures, and processes to manage and monitor the organization's regulatory, legal, risk, environmental, and operational requirements are understood and inform the management of cybersecurity risk.	SBOM
	Risk Assessment (ID.RA): The organization understands the cybersecurity risk to organizational operations (including mission, functions, image, or reputation), organizational assets, and individuals.	SBOM
	Risk Management Strategy (ID.RM): The organization's priorities, constraints, risk tolerances, and assumptions are established and used to support operational risk decisions.	SBOM
	Supply Chain Risk Management (ID.SC): The organization's priorities, constraints, risk tolerances, and assumptions are established and used to support risk decisions associated with managing supply chain risk. The organization has established and implemented the processes to identify, assess, and manage supply chain risks.	SBOM

Source: NIST (Chart modified slightly from the NIST original.)[27]

23. U.S. Department of Commerce, National Institute of Standards and Technology, "Cybersecurity Framework," accessed August 20, 2021. (https://www.nist.gov/cyberframework)

24. Executive Order 13636, "Improving Critical Infrastructure Cybersecurity," February 12, 2013. (https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/executive-order-improving-critical-infrastructure-cybersecurity)

25. Executive Order 13800, "Strengthening the Cybersecurity of Federal Networks and Critical Infrastructure," May 11, 2017. (https://www.federalregister.gov/documents/2017/05/16/2017-10004/strengthening-the-cybersecurity-of-federal-networks-and-critical-infrastructure); The White House, National Security Council, "Strengthening the Cybersecurity of Federal Networks and Critical Infrastructure," June 1, 2018. (https://trumpwhitehouse.archives.gov/articles/strengthening-the-cybersecurity-of-federal-networks-and-critical-infrastructure)

26. U.S. Department of Commerce, National Institute of Standards and Technology, "Cybersecurity Framework: Identify," accessed August 20, 2021. (https://www.nist.gov/cyberframework/identify)

27. U.S. Department of Commerce, National Institute of Standards and Technology, "Cybersecurity Framework Version 1.1," April 2018. (https://www.nist.gov/cyberframework/framework)

A Software Bill of Materials Is Critical for Comprehensive Risk Management

The *asset management* category involves identifying and managing "the data, personnel, devices, systems, and facilities that enable the organization to achieve business purposes."[28] The task of identifying assets has traditionally included identifying software, and it should also include identifying software components that make up the complete software package. If an organization does not know what its software comprises, the asset management process will be incomplete, and the organization may be blind to the vulnerabilities and risks within its software packages.

The *business environment* category involves understanding and prioritizing an enterprise's "mission, objectives, stakeholders, and activities" to inform "cybersecurity roles, responsibilities, and risk management decisions."[29] If an organization does not know what its software comprises, the organization may unknowingly introduce high-risk software into systems that have a low risk tolerance and require a high degree of security, assurance, and dependability, such as critical infrastructure and weapons systems.

The *governance* category involves understanding the policies and procedures needed to comply with an organization's regulatory, legal, and operational requirements and then using those policies to inform cyber-risk management. If an organization does not know what its software comprises, the organization cannot effectively audit the software to ensure compliance with regulatory, legal, risk, environmental, and operational requirements. These requirements might include, for example, directions from the U.S. Treasury Department's Office of Foreign Assets Control not to use components originating from Crimea, Cuba, Iran, North Korea, or Syria. Without knowing the software components nested within the complete software package, an analysis of the package's origin, compliance, and security is impossible.

Companies may need to adhere to regulations such as the General Data Protection Regulation or the California Consumer Privacy Act when purchasing software to use within the digital environment. The regulations apply to all layers of software that make up the digital environment. Without knowing the third (plus)-party components, ensuring compliance with relevant laws is impossible.

The *risk assessment* category involves an enterprise's understanding of the cybersecurity risks to its operations, mission, and reputation. The results of a risk assessment reveal threats and vulnerabilities and determine the organization's risk level. If an organization does not know what its software comprises, its network operators cannot monitor the ongoing security of, and risks presented by, those underlying components. Organizations use resources such as the National Vulnerability Database and the list of CVEs to identify known vulnerabilities within their software.

When programmers reuse software, they may conduct security scans on its components. Those components might have no identified CVEs not because there are no vulnerabilities, but because no one has researched the software and publicly identified its vulnerabilities. For this reason, a point-in-time security check is never adequate. Instead, software components should be identified, assessed, and monitored on an ongoing basis to determine an organization's risk exposure.

In addition to security vulnerabilities, the nested software may have license, compatibility, or maintenance issues. Without knowing the nth-party (plus) software components, an accurate assessment of the potential risks introduced by the software is impossible.

The risk assessment's results then feed into the next category, a *risk management strategy*. Operational risk decisions and the overall risk strategy are only as good as the risk assessment data upon which they are built.

28. U.S. Department of Commerce, National Institute of Standards and Technology, "Cybersecurity Framework: Identify," accessed August 20, 2021. (https://www.nist.gov/cyberframework/identify)

29. Ibid.

A Software Bill of Materials Is Critical for Comprehensive Risk Management

If an organization does not know what its software comprises, the organization's risk management strategy will not address the inherent vulnerabilities of the underlying software components. Software licensing, version, development compliance, dependencies, associated CVEs, and code origin are just a few of the component software attributes that organizations should include when developing their risk management plans.

Having identified its "priorities, constraints, risk tolerances, and assumptions,"[30] a company can then assess and manage supply chain risks. The final category, *effective supply chain risk management*, requires that software and the software components be identified, prioritized, and routinely evaluated. If an organization does not know what its software comprises, it should assume the software is compromised.

30. Ibid.

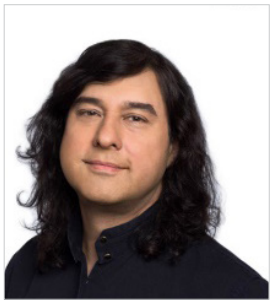
Appendix D: SBOM Attributes and Resulting Analysis

SBOM Attribute	Analysis	Possible Resulting Information	CSF Category Informed
Author of the SBOM	Service provider analysis	• Format used • Generation process used • Attributes identified • Analysis performed • Ability to share or distribute • SBOM integrity verification practice	• Asset Management • Business Environment • Governance • Risk Assessment • Risk Management Strategy • Supply Chain Risk Management
Name of supplier of the components	Open-source analysis	• Supplier risk assessment	
SBOM unique identifier	Integrity analysis	• A unique hash value is used to ensure immutable, auditable, and provenance	
Relationship of components	Dependency hierarchy	• Relationship	
• Complete software name • Component software (direct dependency) • Transitive dependencies (the direct component dependencies)	Dependency scan	• Count of all dependencies (direct and transitive) • Name of all dependencies	
	Vulnerability scan	• Known vulnerabilities identified in the software	
	Technical Debt Analysis	• Number and degree of outdated dependencies whose vulnerabilities (known and potential) have long exposure times, and which create operational risk by making the capability challenging to update • Dependencies with no version number	
	Language scan	• Number of programming languages detected within the software	
	Ecosystem Risk	• Level of developer support • Supplier risk indicators such as end-of-life and change-of-control detection • Days since the last commit • Unique committers • Evidence geographical origin	
	License scan	• Licenses for the project	
	Monitoring software (difference scan)	• Changes in the code detected since the previous five scans • Patterns of maintenance	
	Container analysis	• Container dependencies • Container images	
Component hash	Integrity analysis	• Assessment of integrity provenance	

A Software Bill of Materials Is Critical for Comprehensive Risk Management

Acknowledgments

FDD's TCIL is a nonprofit organization that relies on volunteers with a passion for advancing cybersecurity practices. Thank you, Arun, Yaron, Clayton, JC, and John for your contributions and expertise. You offered this pilot truly unparalleled experience that will help government agencies and private-sector companies on the front lines of the cyber battlespace to adopt secure, accountable practices that will mitigate supply chain vulnerabilities.



Arun Majumdar
CEO & Co-Founder
Virgil Systems

JC Herz
CEO
ION Channel

Yaron Vorona
COO & Co-Founder
Virgil Systems

John Scott
COO
ION Channel

Clayton Jones
CSO
Virgil Systems

A Software Bill of Materials Is Critical for Comprehensive Risk Management

About the Author

Dr. Georgianna "George" Shea serves as chief technologist for FDD's Center on Cyber and Technology Innovation and TCIL. In that role, she identifies cyber vulnerabilities in the U.S. government and private sector, devising pilot projects to demonstrate feasible technology and non-tech solutions that, if scaled, could move the needle in defending U.S. prosperity, security, and innovation.

About the Foundation for Defense of Democracies

FDD is a Washington, DC-based, nonpartisan 501(c)(3) research institute focusing on national security and foreign policy.

About FDD's Transformative Cyber Innovation Lab

TCIL finds and nurtures technologically feasible, testable pilot projects which begin to solve some of the hardest cyber problems afflicting the national security industrial base and the United States. TCIL's mission is to help shorten the lag between idea and piloting and between piloting and the adoption of potential solutions to the thorniest of cyber problems. TCIL seeks to drive revolutionary, society-wide improvement in cyber resilience through the innovative synthesis of technology, policy, and governance.

For more information, visit: https://www.fdd.org/projects/transformative-cyber-innovation-lab