CYBER STATECRAFT INITIATIVE

#ACcyber

DFRLab

ISSUE BRIEF

NOVEMBER 2022

The Cases for Using the SBOMs We Build

AMELIE KORAN, WENDY NATHER, STEWART SCOTT, AND SARA ANN BRACKETT

In the beginning, developers created package manifests and header files. Code was formless and required documentation. Tabs and spaces hovered on the surfaces of the editors, and the spirit of Dennis Ritchie hovered over the code.

And then a developer typed, "git commit" and behold, there was a commit, and the developer saw that the commit was good, so they separated BRANCH from MAIN. They called the BRANCH a version and MAIN the source, and there were pulls and pushes and the first release. And yet lo, users often had little idea what was in any of it. This went on to cause many problems, but that did not make it a bad idea.

INTRODUCTION: SBOMS, PUBLIC POLICY, AND YOU

Anyone in tech, cyber policy, or security circles has probably heard about software bills of materials (SBOMs) by now and considered how they or their organization might use SBOM data. Many recent efforts strive to answer this question—one good example is Microsoft's Open-Source Software Secure Supply Chain framework.[1] Asking about SBOM use is nonetheless a reasonable act of self-examination given their relatively recent appearance on the policy scene, mostly in the wake of major software supply chain incidents.[2]

SBOMs themselves are not new. One widely accepted SBOM format, the Software Package Data Exchange (SPDX), dates back to 2011.[3] Notably, that original SBOM concept has its roots in complex physical manufacturing processes in industries like the automotive sector to under-

1 "Open Source Software (OSS) Secure Supply Chain (SSC) Framework" (2022; repr., GitHub: Microsoft, August 4, 2022), https://github.com/microsoft/oss-ssc-framework/blob/165ba893f2080e75bc69acaa6ea3fc8550315738/specification/Open_Source_Software_(OSS)_Secure_Supply_Chain_(SSC)_Framework.pdf.

2 Incidents, rather than attacks, as several also included valid use cases and functionality leading to cascading failures or vulnerabilities—all important to recognize.

3 Adrian Bridgwater, "Linux Foundation Eases Open Source Licensing Woes," Computer Weekly, August 19, 2011, https://web.archive.org/web/20210820144000/https://www.computerweekly.com/blog/Open-Source-Insider/Linux-Foundation-eases-open-source-licensing-woes.

The **Cyber Statecraft Initiative** works at the nexus of geopolitics and cybersecurity to craft strategies to help shape the conduct of statecraft and to better inform and secure users of technology. This work extends through the competition of state and non-state actors, the security of the internet and computing systems, the safety of operational technology and physical systems, and the communities of cyberspace. The Initiative convenes a diverse network of passionate and knowledgeable contributors, bridging the gap among technical, policy, and user communities.

The mission of the **Digital Forensic Research Lab (DFRLab)** is to identify, expose, and explain disinformation where and when it occurs using open-source research; to promote objective truth as a foundation of government for and by people; to protect democratic institutions and norms from those who would seek to undermine them in the digital engagement space; to create a new model of expertise adapted for impact and real-world results; and to forge digital resilience at a time when humans are more interconnected than at any point in history, by building the world's leading hub of digital forensic analysts tracking events in governance, technology, and security.

stand intricate supply chains, as well as in legal practices for recording the inheritance of licenses through a business.[4] Meanwhile, those who have compiled software from source code are likely familiar with build manifests that indicate all the packages, libraries, and other bits of code needed to properly construct a final piece of software. The bigger a project, from a simple application to an entire operating system, the longer and more complex that manifest becomes. An SBOM is similar—a snapshot in time of each component making up a piece of software, with additional metadata tracking provenance (information about component authors and affiliations) and versioning.[5]

While SBOMs are intuitively useful and have received some notable policy attention of late—from the National Telecommunications and Information Administration's (NTIA) minimum-viable elements project to mentions in executive orders (EOs) and Office of Management and Budget (OMB) memoranda—they are just one tool (more precisely, one class of data) in the wider arsenal for managing risk in software systems.[6, 7] Although conversation about SBOMs has largely (and understandably) focused on their generation, requirements, and format, their growing maturity demands wider consider consideration of next steps: developing clear use cases for SBOMs. An absence of mature, well-understood use cases for SBOMs threatens their future as an effective risk management tool.

Though SBOMs and their widespread adoption face other, arguably more dire, challenges—for example, the risks of mistimed regulation and disconnects between SBOM designers and consumers—policymakers and the security community can directly address use cases now. Letting the challenges of SBOM generation drown out demand signals from the user side of the pipeline risks inundating purchasers, developers, and acquisition officers alike with a torrent of useless spreadsheets and effete compliance certifications.

Indeed, these uses extend beyond just technology-consuming firms to include governments and other central risk assessment bodies. An absence of well-articulated SBOM use cases and illustrated relevance to communities of SBOM consumers holds twin challenges. First, it risks mission creep, where policymakers might begin to frame SBOMs as a silver bullet for all supply-chain woes without clear demarcation of the problems they are designed to address. Second, it undersells SBOMs to those who would consume them, leading to slower adoption, poor tooling, and the malformation of a potentially powerful data standard into yet more bloated security theater.

To address the opportunity for further usage conversations, this paper offers several grounded applications for SBOMs, focusing particularly on the benefits they offer their consumers, from chief information security officers (CISOs) to acquisition officers and from software consumers to the Cybersecurity and Infrastructure Security Agency (CISA). Incident response may be the most intuitive role for SBOMs—a way to determine impacted software when a widespread component is compromised or found vulnerable—but it is far from the only one. SBOMs can help development teams determine what packages they will be managing. They can feed software composition analysis (SCA), acting as an ingredient and source list. They can help compliance officers streamline licensing acquisition and manage the adoption of components produced by sanctioned or entity-listed companies. At the largest scale, they can map out portions of the software ecosystem, highlighting little-known relationships and concentrations of dependence, while shedding light on the benefits of using extant code and the risks of relying on external repositories. First though, this paper considers the state of contemporary SBOM policy conversations.

STILL FIGHTING YESTERDAY'S BATTLES

The year 2014 saw one of the first truly widespread, dire software supply-chain events: the OpenSSL "Heartbleed" vulnerability.[8] Heartbleed put the many systems that relied on OpenSSL at significant risk, allowing malicious actors to extract sensitive information due to a relatively simple software flaw. The incident catalyzed a small surge in private-sector funding to open-source projects to support security efforts and raised questions about ways to effectively track the use of critical, community-developed software in systems spread around the world, as well as ways to coordinate responses to flaws found in such code. The US government immediately asked all federal agencies, as part of alerting the public through the Department of Homeland Security (DHS), to emphasize where websites and other internet services used OpenSSL libraries.[9] However, that was only the tip of the iceberg—in fact, OpenSSL also lived on many mobile devices, embedded hardware systems, and phone and conference-call systems,[10] as well as much networking infrastructure.

4 The Linux Foundation, "The Linux Foundation's SPDXTM Workgroup Releases New Version of Software Package Data ExchangeTM Standard - Linux Foundation," August 30, 2012, https://www.linuxfoundation.org/press/press-release/the-linux-foundations-spdx-workgroup-releases-new-version-of-software-package-data-exchange-standard-2.

5 In practice, many real SBOM-generation processes are more complex—build processes might resolve placeholder dependencies, with only the end result reflected in an SBOM, for example.

6 National Telecommunications and Information Administration (NTIA), "The Minimum Elements For a Software Bill of Materials (SBOM)" (Washington, DC: United States Department of Commerce, July 12, 2021), https://www.ntia.doc.gov/report/2021/minimum-elements-software-bill-materials-sbom.

7 Exec. Order. No. 14028 on Improving the Nation's Cybersecurity, Federal Register, 86 FR 26633 (May 12, 2021), https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity.

8 Timothy B. Lee, "The Heartbleed Bug, Explained," Vox, May 14, 2015, https://www.vox.com/2014/6/19/18076318/heartbleed.

9 Larry Zelvin, "Reaction on 'Heartbleed': Working Together to Mitigate Cybersecurity Vulnerabilities | Homeland Security," Department of Homeland Security, April 11, 2014 [Updated September 20, 2018], https://www.dhs.gov/blog/2014/04/11/reaction-%E2%80%9Cheartbleed%E2%80%9D-working-together-mitigate-cybersecurity-vulnerabilities-0.

10 Cisco Security, "Cisco Security Advisory: OpenSSL Heartbeat Extension Vulnerability in Multiple Cisco Products," Cisco, April 9, 2014, http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20140409-heartbleed.

Collecting data on the usage of OpenSSL protocols among websites to understand Heartbleed exposure was a useful first step to an unwieldy triage process. Wider SBOM adoption at the time would have aided long-tail remediation of false negatives and subtle implementations. Further, had a CISA-style entity been able to ingest and use SBOM information on OpenSSL, the true sprawl of the library would have been more immediately apparent and accessible—perhaps even before the vulnerability was found, leading to a better, more targeted response and, crucially, enabling proactive investment and security before the incident.

Discussions of SBOMs and their development have the opportunity now to match the technical solutions enabled by SBOM data to the policy challenges around transparency, processes, and due diligence they can address, and use case refinement will drive that matching. SBOMs offer a mechanical view into the minutiae of documentation for software, summarizing all the pieces of code that make up modern applications and services. If the end goal is for the digital ecosystem to widely adopt SBOMs—both their production and practical use by recipients—much of the necessary intermediary work in ingesting and interpreting SBOM data remains unfinished. This is understandable: in the early SBOM days, deliberate decisions to limit scope—in the NTIA Minimum Elements for an SBOM process, for instance—helped reduce a sprawling problem set to a tractable project.[11] Now that SBOMs are moving toward the mainstream, beginning to address broader use scenarios will help drive their adoption and maturity, and industry, in particular, can play a key role in pushing an aggressive development cycle with clearly defined uses for SBOMs, each contributing to different facets of cybersecurity.

The potential role for SBOMs in long-term remediation of Heartbleed-style events to provide a snapshot of the composition of software packages is clear. However, this security model requires that, upon build and deployment, developers and consumers update and transparently publish SBOMs for consumption. SBOMs only work well if they are common, standardized, and quickly updated—all a considerable way off from the current situation,[12] as a February 2022 Linux Foundation (LF) study found that less than half of surveyed organizations were "using" SBOMs.[13] This survey likely represents an optimistic upper-bound as well—64 percent of respondents were LF member companies, likely skewing toward SBOM maturity; only 74 percent of the organizations classed as using SBOMs were producing and consuming them; and even partial or marginal organizational use would have counted for the survey (and is helpfully broken down within the analysis, which acknowledges and strives to address potential sample bias well).

This is not a critique of adoption speed and progress to date, but rather an acknowledgment that the next steps for SBOMs will require a gear shift that well-articulated use cases and a clear policy demand signal can help accomplish. The same survey queried about adoption plans, forecasting a promising 66 percent increase in the rate of SBOM production and consumption amongst respondents. Anecdotally, some industries at the forefront of SBOM development are already innovating these use cases. For instance, the healthcare sector—which acted as one of the testbeds for NTIA's SBOM proof-of-concept studies[14]—use SBOM processes to highlight relationships with suppliers and OSS communities that merit increased support, as well as produce human-readable risk analysis information.[15]

The most common, general communications to policymakers about SBOMs are that they are ingredient lists most useful for assessing the scale and supporting the recall of tainted, defective components. This describes the minimum viable SBOM: a list of component software, only referred to upon the discovery of a defective part—and in the case of NTIA's minimum viable SBOM, only one layer of dependencies is tracked.[16]

This paper is not a call to reinvent SBOM standards. Like so much of government cybersecurity policy, the extreme visibility of SBOMs is a reaction to crisis. Rather, it argues that use cases can and should shape the production and adoption of SBOM and the tools accompanying them. As mentioned earlier, some of this work is underway,[17] but policy conversations can continue focusing on what SBOM data can enable and what tooling and production/adoption incentives will best drive development there at a sufficient pace. Policies can also help match the different methods of SBOM production to the most applicable usage. Use cases strengthen the SBOM value proposition with both code maintainers and consumers, as well as help overcome obdurate resistance from technology vendors with little desire to have their behavior "shaped."[18]

11 NTIA, "The Minimum Elements For a Software Bill of Materials (SBOM)."

12 The Cybersecurity Coalition, "Comments on NTIA's Request for Information (RFI) on 'Software Bill of Materials Elements and Considerations,'" June 17, 2021, https://assets.website-files.com/60cd84aeadd2475c6229482f/60ec9f0a15e85933daa3b5ca_Coalition%20SBOM%20Response-Final%206-17-21.pdf.

13 Stephen Hendrick, "The State of Software Bill of Materials (SBOM) and Cybersecurity Readiness" (The Linux Foundation | Research, January 2022), https://8112310.fs1.hubspotusercontent-na1.net/hubfs/8112310/LF%20Research/State%20of%20Software%20Bill%20of%20Materials%20-%20Report.pdf. The survey does well acknowledging and striving to address the above-mentioned sources of possible bias explicitly, too.

14 National Telecommunications and Information Administration, "Healthcare SBOM Proof of Concept" (NTIA, April 29, 2021), https://www.ntia.doc.gov/files/ntia/publications/ntia_sbom_healthcare_update-2021-04-29.pdf.

15 Sourced from conversations with New York Presbyterian.

16 NTIA, "The Minimum Elements For a Software Bill of Materials (SBOM)," 12.

17 Velichka Atanasova, "Let's Get SBOM Ready - Open Source Blog," VMWare, April 14, 2022, https://blogs.vmware.com/opensource/2022/04/14/sbom-ready/.

18 Alliance for Digital Innovation et al., "Cautionary Notes on Codifying Use of SBOMs," September 14, 2022, https://fcw.com/media/multi_association_letter_on_sbom_final_9.14.2022.pdf.

USE CASES

The policy challenge behind SBOMs is the question of adoption—compelling use-cases can motivate that while sensibly shaping the circumstances and specifics of regulation. Below are four foundational use cases for SBOMs, each with their respective audiences, outcomes, and positions in the product and incident lifecycles. This is by no means an exclusive list, but it represents diverse and important usage. Each asks for different levels of SBOM completeness, from a minimum-viable components list to a thorough accounting of support, funding, versioning, and deployment context that no current SBOM standard mandates.

1. Procurement—for reducing compliance burdens and preventing duplicative purchases.

2. Vulnerability Management and Threat Intelligence—for tracking compromised components and remediation planning.

3. Incident Response—for validating liability claims and guiding patch efforts.[19]

4. Ecosystem Mapping—for providing a bird's-eye view of dependencies in an enterprise's ecosystem and beyond.

Discussing the role for SBOMs in these cases and the larger impacts of their use, offers clarity for the government on how to incentivize and structure SBOM adoption and, for industry, on what tooling to focus development.

1. Guiding Software Procurement and Adoption Decisions

SBOMs can prove useful during the procurement process for any third-party software, beyond obvious security functions. Large organizations often make individual purchases rather than coordinating licensing centrally. So creating an inventory and consolidating duplicate purchases or capabilities for cost savings can make a chief financial officer's (CFO) day. Licensing checks can also surface instances where entities adopt open-source software but cannot legally incorporate it into other products. These are quick wins because the acceptance or rejection of that software can be binary: if licensing prevents use or an existing contract covers a need, the procurement goes no further. Software asset registers and intellectual property scanners already strive to serve these functions, but, given the overlap in their data and that of SBOMs, there is room for tooling to support quick decisions making for all, as well as for the different data sources to support rather than supplant each other.

Binary decisions can form part of a standard procurement process in pre-negotiations with suppliers, but decisions involving judgment calls (such as the relative criticality of a known bug) tend to slow workflows and create angry calls from executives who want to know the reasons behind a derailed purchase. Again, deciding ahead of time which data points in an SBOM are deal-breakers will streamline the process of software procurement. Simple, written policies such as "never adopt or acquire software with components from X supplier"—which can refer to competitors, companies operating in sanctioned nations, entity-listed organizations, known risky projects, or anything else unambiguously identified—can work well here, especially with automation.

When it comes to adopting and integrating open-source software, many of these policies should already exist at most firms, but SBOM use with a standardized format can streamline validation. One must check on the status of a project referenced in an SBOM: how healthy, deep, and thorough its community support is, how much investment it enjoys, or, if tied to a proprietary offering, how dedicated to support the parent company is—none included in the SBOM per se, but retrievable from tools like OpenSSF Scorecard, SLSA levels, and more once upon identifying dependencies. Many CISOs already struggle with the need to collect supply-chain data at a granular level for risk management. While insufficient for high-security organizations, SBOMs are workable substitutes for medium or small enterprises that lack the in-house expertise to analyze all their software in depth, and their unaltered data can serve to inform and define procurement standards and policies alongside risk-management posture.

2. Adding Smarts to Vulnerability Management and Threat Intelligence

One of the main use cases for SBOMs is identifying components affected by vulnerabilities. SBOMs provide visibility into software a level or two deeper than is common today, particularly provenance. They allow for better triage, cross-referencing dependencies, and remediation planning for identified vulnerabilities. SBOMs provide the roadmap through software relationships that enable this degree of dedicated care.

One constructive application of SBOMs in this context is improving the usefulness of vulnerability risk ratings to impacted organizations. One organization's "critical" is not necessarily so for a different environment, use case, or business model. Application security professionals already know this, but wide adoption of SBOMs may change how

19 To differentiate vulnerability management and incident response, consider the former tracking vulnerabilities, relevant threat intelligence around dependencies, preemptive response planning, and determining whether a vulnerability impacts an enterprise. The latter comes into play after that determination—guiding patch efforts, outreach to third-party maintainers, mitigation, and tailoring general remediation plans to specific incidents.

they design a remediation strategy by clarifying what entity is ultimately responsible for fixing a vulnerability and how those outside an organization's control might handle that request. Some dependencies may have quite capable maintainers that can be relied on while others might require significant external support. SBOM data highlighting dependencies can help teams identify what external parties they rely on for code support and adjust accordingly and ahead of incidents.

Developers may need to confirm whether a vulnerable component of a package is actually in use. If not, the organization can declare the risk "low" and simply note that policy will change if it incorporates that component in the future. There is a possible resource squeeze in the future for enterprises that need more application development and security staff to investigate the origins of disclosed vulnerabilities, determine remediation responsibilities, pass on notifications and updates to affected parties within the ecosystem, and sign off on version changes to internally generated SBOMs. SBOMs are part of enabling that level of decision-making, allowing better tracking of dependencies and changes to them to provide better insight into actual vulnerability exposure. Again, the data SBOMs provide are just part of the foundation on which to build these processes, complemented by other tools and data like GitBOM and Vulnerability Exploitability eXchange (VEX), highlight the importance of sharpened demand signals from SBOM consumers.

One of the main questions to ask with any SBOM is whether its source and contents are trustworthy. One useful method involves scanning the binary of the software to validate the accuracy of the SBOM—essentially checking that what is under the hood matches the parts list. Binary scanners are imperfect, and if the same scanners help generate an SBOM in the first place,[20] they may not produce reliable SBOMs for consumers using them in their own vulnerability scanning.[21] SBOMs and scanning can help each other, mutually improving the accuracy of package component determination.

The overall risk rating of a software vulnerability informs the risk of a partial or phased remediation. Whether waiting for a third party to deliver a patch or allocating limited internal resources against dependencies that take longer to resolve and downstream requirements from partners, organizations will be able to monitor SBOM-sourced vulnerability data as part of their infrastructure risk-management practices (in conjunction with centralized data like VEX).[22] This monitoring can also help threat intelligence analysts better understand organizational exposure. Better dependency knowledge from an SBOM can help clarify where dependencies might be under-resourced, frequently targeted by adversaries, or otherwise deserving of extra scrutiny and resourcing. The frequency of versioning changes can even provide insight into changes that support critical components. Even simply improving organizational visibility into the attack surface of its dependencies will help prioritize resourcing, direct remediation planning, and expand overall cybersecurity for an organization making full use of its SBOMs.

3. Incident Response and Building a Better Packing Slip

While the above uses focus on using SBOMs for response planning prior to an incident, SBOMs also have utility right of "boom," or after the fact. In many cases, initially, SBOMs can act as verification for incident reports and recommendations—a pointer to where things went wrong in a compromise. As corroborating evidence, a verified SBOM from an environment, system, or other package can help in the review of an incident and determine the impact on parallel systems or previous system versions. The core value within incident response and forensics is accurately comparing versions, changes, and their respective release times. An SBOM may provide some simple insight—after all, if an organization cannot confirm or deny whether a system was affected, does it have to declare a breach anyway? Having an SBOM that raises unanswerable questions is a business risk to examine with the leadership—business risks that otherwise might not have surfaced.

SBOMs can also aid in crisis communication among partners, affected organizations, and customers during and following an incident. Most product-security organizations already have a workflow to add SBOM information to, but they may require some additional information, such as a timeline matching SBOM versions to the systems under investigation. A challenge with this level of forensics is that organizations rarely have the right level of logging and sufficient log retention to be able to confirm authoritatively which versions of components were in use at the time of an incident. Suppliers may need to help customers determine whether an incident affected them, and sometimes that information may simply be unavailable.

One more use of SBOMs in incident response is to validate that an assertion about the contents listed by an SBOM were reasonably accurate at the time of release and that no known and unaddressed vulnerabilities existed. Organizations can reference attestations later if events or evidence indicate something different. While SBOMs are often compared to the ingredients list on a food-product label for software, another analogy could consider them a packing slip, describing what a supplier claimed was in a box at the time of its sealing. If a checksum to verify the absence of tampering fails, an SBOM can help guide responders to tracking down the discrepancies between shipped and delivered software.

20 One of several ways to generate an SBOM.

21 Ariadne Conill, "Not All SBOMs Are Created Equal," Chainguard, April 22, 2022, https://www.chainguard.dev/unchained/not-all-sboms-are-created-equal.

22 National Telecommunications and Information Administration (NTIA), "Vulnerability-Exploitability eXchange (VEX) – An Overview," September 27, 2021, https://www.ntia.gov/files/ntia/publications/vex_one-page_summary.pdf.

4. A Systemic View of Software Risk

In addition to using SBOMs between and within companies, SBOMs can also serve government agencies and other third parties in mapping dependency chains and concentration risk across the software ecosystem. Recent, widespread vulnerabilities, including log4shell, emphasize the degree to which single dependencies can underpin vast quantities of software. Without a systemic view into dependency patterns, government agencies and others will struggle immensely to assess risk across and within sectors. Given access to SBOMs from multiple sources, government could use that aggregated data to assemble a rough map of dependencies across slices of the digital ecosystem—a picture not just the dependencies of one application, but of many, and more importantly, where they overlap. While contemporary software composition analysis (SCA) can provide similar insight into widely-depended-on software,[23] running SCA tools across the far larger set of software considered here would likely prove far less feasible or replicable. To protect both intellectual property and the critical nodes such a map might highlight, government would need to take extra care in protecting this data, but it would prove useful in identifying under-secured or under-resourced dependencies ripe for proactive investment and support. Vulnerabilities in one company's codebase or within a popular open-source repository can have global impact. Widespread ignorance about software dependencies hampers proactive support that might include security auditing, maintainer funding, development of alternate dependencies, or any other number of methods to reduce the risk of high-leverage dependency.

Governments and private-sector companies currently lack measures that describe the scale of use of different pieces of software. Metrics such as download counts, license purchases, or userbase size do not provide information about deployment or reliance, either upstream or downstream. A package with only a single user could still be critically important if all kinds of different software depend on it. However, without relationship mapping, the entire ecosystem remains blind to that package's position as an essential link in the supply chain. SBOMs can reduce this problem by providing data, when aggregated from many sources, for an ecosystem-wide view of software dependencies to CISA and other entities, even if only for part of an enterprise. CISA is likely to be tasked with some of this work should the Securing Open Source Software Act of 2022 (S.4913), pass into law, or under III.B.2 of OMB M-22-18 on Enhancing the Security of the Software Supply Chain through Secure Software Development Practices.[24, 25]

As more workloads move into the cloud, understanding and assessing the risk present in those systems is vital. One important use of SBOMs for software-as-a-service (SaaS) consumers is encouraging greater transparency in vulnerability reporting and mitigation inside cloud services. Over time, this information will help support more precise decision-making about the security practices of different vendors. While some companies have made policy choices about what to reveal to customers and what to withhold,[26] SBOMs are useful tools for other companies to define their own policies, and for customers to push for what they (or regulators) find most comfortable. As part of this effort, good questions will need clear answers regarding how SBOMs can be most useful amidst widely varying configurations and associated products present in different SaaS deployments. Wider generation, use, and consumption provide incentives to determine and sharpen answers to these.

SBOMs can help, though differences between cloud and on-premises software create challenges. One is the speed at which the cloud changes. If SBOMs change minute-to-minute with cloud configurations, they might produce too much information and impede meaningful use by recipients. However, operating off out-of-date information is also risky. Additionally, cloud instances often utilize many different third-party services, so tracking the versioning of each service for each instance or configuration within an SBOM is difficult. Building SBOMs with this aggregate use case in mind will be important to managing this deluge of data, and a key to that is a clearer demand signal from consumers of cloud SBOMs, in and outside of the public sector, about how they aim to incorporate that data into their risk-management practices.

A standardized method for companies (and other entities) to inform each other of dependencies, used and combined at scale, would ease the task of assessing risk across sectors. For SBOMs to fulfill this role, the information contained within them must be consistently organized, filled, and updated, which might pose a challenge to organizational resources. Such data would be most useful when combined with assessments of the context surrounding any piece of software. Even so, SBOMs, as currently imagined, still provide a valuable piece of the puzzle not otherwise measurable. Better data on the arrangement of and relationships with the larger software ecosystem would allow CISA and other agencies to target resources more effectively toward shoring up mission-critical software.

23 Frank Nagle et al., "Census II of Free and Open Source Software — Application Libraries" (Linux Foundation Research; OpenSSF; Laboratory for Innovation Sciences at Harvard: Harvard Laboratory for Innovation Science (LISH) and Open Source Security Foundation (OpenSSF), March 2, 2022), https://lish.harvard.edu/publications/census-ii-free-and-open-source-software-%E2%80%94-application-libraries.

24 "Securing Open Source Software Act of 2022," S.4913, 117th Cong. (2022), https://www.congress.gov/bill/117th-congress/senate-bill/4913.

25 Shalanda Young, United States, Office of Management and Budget, OMB Memo to the Heads of Executive Departments and Agencies, M-22-18, "Enhancing the Security of the Software Supply Chain through Secure Software Development Practices," September 14, 2022, https://www.whitehouse.gov/wp-content/uploads/2022/09/M-22-18.pdf.

26 Kevin Beaumont [@GossiTheDog], "For Anybody Who Doesn't Know, August 2022's Windows Patches Included Fixes for NSA and GCHQ Reported Cryptographic Bugs. but MS Didn't Tell You and Didn't Issue a CVE.," Tweet, Twitter, October 12, 2022, https://twitter.com/GossiTheDog/status/1580244775638212608.

WHY DEFINE USE CASES AT ALL?

Clearly defining the use cases will help guide and preserve the inertia of SBOM adoption and development, from shaping the automated tools for SBOM ingestion to pointing toward new product offerings and molding federal procurement policy. Only considering the challenges of SBOM generation while disregarding the other end of the pipe risks drowning purchasers, developers, and acquisition officers alike in a sea of useless spreadsheets and symbolic compliance certifications.

Though SBOMs and this paper's considered uses of them are as important to proprietary software components as open-source ones, for the latter, they provide the beginnings of a more fundamental guidance, too. Unlike in traditional supply chains for physical goods or in the exchange of proprietary code, OSS dependence rarely sees an exchange of money or a contractual agreement.[27] Rather, there is simply a quick "pip install XX –user" and "import YY as ZZ," often from the public repository. SBOM adoption can eventually change the nature of that informal incorporation, and policymakers still have a chance to sculpt, for better or for worse, the roles and responsibilities that will redefine the ecosystem.

A key policy challenge is determining exactly which entities are ultimately responsible for producing and publishing SBOMs. Suppliers to finished goods manufacturers, due to various global and national regulations, often must detail the source of their materials—whether from forced or child labor, farmed or created under sustainable practices, acquired legally, and so on. The answers have implications for marketing as well as compliance and legal departments. Someone in the chain of the software development lifecycle must be responsible for the creation of SBOMs, but the trust framework for the completeness and veracity of their claims has yet to be developed, and debate over who, precisely, is responsible for making them and what levers are appropriate for achieving compliance persists. Burdening open-source developers and maintainers with that task, though, is an overreach in the absence of ubiquitous tooling to generate SBOMs automatically.

At the regulatory level, all this is challenging, as countries take multiple approaches to what entity is responsible for providing compliance and conformance assurances. This also complicates how governments support the security of open-source software supply chains, as each may have a different goal or preferred method despite aligned motivations. In the United States, CISA wants to assist, even lead, efforts to help support the securing of critical open-source software. However, the culture of open-source communities, the history of their development, and the very tenets that make open source a vital font of innovation all buck against direct government regulation in such stewardship, especially given that open-source code, legally in the United States, is a form of free speech.[28] Importantly, governments supporting the open-source ecosystem will not be able to rely on blanket requirements, and their assistance in identifying critical projects, supporting tooling development, and investing in developers and communities will provide more fruitful results.

SBOMs, sufficiently standardized and adopted, offer data that can serve critical policy challenges when combined with appropriate tooling and processes, allowing a better understanding of and investment in dependencies before incidents occur, as well as more complete vulnerability remediation fixes afterward. Applied and used correctly, SBOMs can make the ecosystem's most capable actors responsible for its coherence. Incorrectly executed, burdensome requirements for SBOM generation could sterilize the open-source world's thriving innovation.

So, What Could You Do About It?

SBOM generators have an outsized say in the use cases of SBOMs because they determine what each bill of materials contains. In developing tools for aggregation, analysis, and production of SBOMs, generators could do the following to speed adoption and provide a more complete, practical set of capabilities to SBOM consumers:

- **Develop tooling to convert** from raw SBOM data to actionable information more intuitively. CISOs, CTOs, and CIOs will not have the time or resourcing to parse through vast, rapidly changing informal tracking of dependency information, but automated checks with customizable, risk-tolerance leveling and other policies can make SBOMs a practical tool during acquisition and incorporation decision-making processes. Adding context, alongside SBOMs, that clearly declares what they do and do not contain and what purposes they serve can help here.

- **Develop tooling to provide** more practical and varied information based on SBOM contents. Many of the use cases discussed above require a touch more detail than conveyed by current SBOM formats. This next layer of tooling, in tandem with products that coordinate SBOM consumption, will provide value both to their manufacturers and users.

27 Iliana Etaoin, "There Is No 'Software Supply Chain,'" iliana.fyi, September 19, 2022, https://iliana.fyi/blog/software-supply-chain/.

28 Alison Dame-Boyle, "EFF at 25: Remembering the Case That Established Code as Speech," Electronic Frontier Foundation, April 16, 2015, https://www.eff.org/deeplinks/2015/04/remembering-case-established-code-speech.

The OMB and CISA have recently begun moving towards SBOM requirements at the federal level, likely in tandem with updates to government procurement processes and working with critical infrastructure sectors. They face a key challenge:

- **Provide better support for smaller enterprises** that cannot easily adopt and produce SBOMs in a compliant manner. CISA might pursue this through added tooling and support in their small-to-medium business (SMB) programs and by tailoring any legal requirements to the unique needs and exposures of different sectors. These need not be new tools adding more complexity and variation to the SBOM landscape, but rather increased funding and guidance for SMBs to access tools normally available only to larger enterprises. Large IT vendors can also act as an intermediary in this provision by offering tooling and support for SMBs with government subsidies.

CISA could model practices to gather SBOM data beyond that used by a single enterprise. Wider collection of SBOM data is necessary for the envisioned aggregate use case. Although this process is more straightforward for open-source systems, there are valid concerns about SBOMs revealing proprietary information and providing attackers with the tools to identify vulnerable targets, particularly among software-as-a-service vendors, whose products are otherwise difficult to scrutinize. Industry could work with government to identify solutions to this information problem; doing so would increase the supply-chain insight SBOMs could provide. Aggregating and analyzing in-house collections of SBOMs first would be a good starting point and force government and industry to directly address the tradeoffs between identifying nodes of systemic risk to better secure them and pointing attackers to those nodes—some of which will be under-supported—through their identification.

SBOM users will need to provide the demand signals to producers that shape the future utility of software bills of materials. Often, users and consumers will be the same party, or at least departments within the same company, but they may also be small firms less focused on tech development, non-profits, or companies without the resources to do more than implement well-documented tooling. This responsibility is also a chance to extract significant value from SBOMs.

- **Accept the imperfect SBOM and iterate:** If a complete SBOM must trace dependencies all the way down to another complete SBOM, they will rarely exist except for the simplest of components. Imperfect is not impractical. The processes that develop around SBOM use must not assume or depend on complete information. Industry and government could explicitly discuss how to navigate imperfect SBOMs and thresholds for acceptable inaccuracy while ensuring users can adopt and iterate on necessarily imperfect standards.

- **Innovate your use cases:** Depending on the depth of information contained in or pointed to by an SBOM, consuming organizations might highlight the use of memory-unsafe languages, insecure calls, unmaintained libraries, or methods highlighted in Open Web Application Security Project (OWASP) and other "top of" lists to block these technologies from their environment. Risk managers can even develop tools converting detailed SBOMs into tolerable-risk metrics.

- **Build with ease for the user in mind:** Part of strengthening the utility and longevity of SBOMs is enabling the use of this rich source of data in a wide range of possible ways. Tooling should reflect the expectation that many users are non-expert and/or lack considerable resources for IT administration and security, prioritizing simplicity and intelligibility over maximal functionality. Enterprise support for SBOM-tool users can help here too.

CONCLUSION

Businesses and developers hold mixed sentiments toward requiring SBOM production in regulations. Keen observers will find working groups with names like "SBOMs Everywhere" with employees from the very same companies funding letters (thinly veiled by trade associations), denouncing some efforts to promulgate SBOM requirements through policy.[29] Part of this fractured view of SBOMs reflects the early stages of SBOM maturity, and part, the variety of opinions and incentives within large organizations too often treated as monolithic entities. More importantly, it reflects a disconnect among available government levers, SBOM functionality, and industry incentives. Procurement requirements are one of government's most effective levers for shaping cybersecurity practices, and industry insistence that government wait for trivial or even default compliance before regulation is circular—if SBOMs were standard practice already, there would be no need to specifically request them to begin with, and government requiring higher security standards from its vendors is far from aberrant. The mismatch between federal security needs and the state of SBOM adoption and maturity is a significant opportunity for industry to continue to deepen its partnership with government and other would-be SBOM users to keep up the pace on SBOM development while shaping the tools serving SBOMs and the challenges that they can address.

A key question persists: what do SBOM producers stand to gain, short of compliance, from their considerable toil? Many prior requirements of large suppliers and component suppliers—self-attestations or FedRAMP requirements, for example—might have necessitated great expenditure in return for relatively small benefits to individual entities. Without making a clear case for SBOM use and the resultant tools that provide return on investment, policymakers advancing SBOMs risk mortgaging their future as a marketing tool— another sticker slapped on the proverbial product denoting begrudging compliance with federal requirements. Successful policy supporting SBOMs must put them on a sustainable path, tying hard and fast requirements to clear benefits for the ecosystem and the entities within it. Part of this must translate to better articulating how SBOMs can be consumed and used toward a variety of ends and by a diversity of organizational types.

Lacking a clear, tangible value proposition, particularly to considerations like the bottom line, future contracts, operations, or other more immediately recognizable benefits will create friction between parties that desire to use SBOMs and those that will not willfully provide them, even while governments and other organizations push to have SBOMs a standard part of their procurements. It is worth noting that some of the best analogs to SBOMs share a similarly fraught origin. Nutrition labels, ingredient lists, and food-goods advertising regulations span a century-long tug-of-war between government, industry, and consumer.[30] The transition from prepared-from-scratch meals to off-the-shelf purchasing helped spur Food and Drug Administration (FDA) regulation, as consumers required better visibility into their purchases.[31] Notably, some companies already use SBOMs or similar data internally, of their own accord, and presumably, for some of the benefits enumerated here—Google and Microsoft are easy enough examples to find public records of this.[32, 33]

This paper aims to remove some of the friction against SBOM adoption and strengthen their long-term utility as a source of data for important risk management decisions, showing potential consumers clear benefits from using SBOMs, nudging producers and tool developers towards new offerings, and making clear to policymakers the importance of decisions they are already considering. SBOMs, initially marketed in cybersecurity as a solution to the fact that one cannot secure dependencies one does not know about, can enable so much more along the way. It is time they were sold as such.

ACKNOWLEDGMENTS:

The authors of this paper would like to thank external reviewers John Speed Meyers, Aeva Black, William Bartholomew, and Allan Friedman, who all took significant time to provide input during this paper's development, as well as Anais Gonzalez and Donald Partyka for designing the final document and others who contributed invaluable feedback.

29 Alliance for Digital Innovation et al., "Cautionary Notes on Codifying Use of SBOMs," September 14, 2022.

30 Institute of Medicine (US) Committee on Examination of Front-of-Package Nutrition Rating Systems and Symbols, "Front-of-Package Nutrition Rating Systems and Symbols: Phase I Report," in History of Nutrition Labeling, ed. Ellen A. Wartella, Alice H. Lichtenstein, and Caitlin S. Boon (Washington, DC: National Academies Press (US), 2010), https://www.ncbi.nlm.nih.gov/books/NBK209859/.

31 Department of Nutritional Sciences, University of Texas at Austin, "Factual Food Labels: A Closer Look at the History," April 6, 2018, https://he.utexas.edu/ntr-news-list/food-labels-history.

32 Jessica Lyons Hardcastle, "Google SLSA, Linux Foundation Drops SBOM for Supply Chain Security Boost," SDxCentral, June 18, 2021, https://www.sdxcentral.com/articles/news/google-slsa-linux-foundation-drops-sbom-for-supply-chain-security-boost/2021/06/.

33 Simon Bisson, "How Microsoft Will Publish Info to Comply with Executive Order on Software Bill of Materials," TechRepublic, May 6, 2022, https://www.techrepublic.com/article/microsoft-publish-info-comply-executive-order-software-bill-materials/.

Atlantic Council

Board of Directors

CHAIRMAN
*John F.W. Rogers

EXECUTIVE CHAIRMAN EMERITUS
*James L. Jones

PRESIDENT AND CEO
*Frederick Kempe

EXECUTIVE VICE CHAIRS
*Adrienne Arsht
*Stephen J. Hadley

VICE CHAIRS
*Robert J. Abernethy
*C. Boyden Gray
*Alexander V. Mirtchev

TREASURER
*George Lund

DIRECTORS
Todd Achilles
Timothy D. Adams
*Michael Andersson
David D. Aufhauser
Barbara Barrett
Colleen Bell
Stephen Biegun
Linden P. Blue
Adam Boehler
John Bonsell
Philip M. Breedlove
Myron Brilliant
*Esther Brimmer
Richard R. Burt
*Teresa Carlson
*James E. Cartwright
John E. Chapoton
Ahmed Charai
Melanie Chen
Michael Chertoff
*George Chopivsky
Wesley K. Clark
*Helima Croft
*Ankit N. Desai
Dario Deste
*Paula J. Dobriansky
Joseph F. Dunford, Jr.
Richard Edelman
Thomas J. Egan, Jr.
Stuart E. Eizenstat
Mark T. Esper
*Michael Fisch
Alan H. Fleischmann
Jendayi E. Frazer
Meg Gentle
Thomas H. Glocer
John B. Goodman
*Sherri W. Goodman
Jarosław Grzesiak
Murathan Günal
Frank Haun
Michael V. Hayden
Tim Holt
*Karl V. Hopkins
Kay Bailey Hutchison
Ian Ihnatowycz
Mark Isakowitz
Wolfgang F. Ischinger
Deborah Lee James
*Joia M. Johnson
*Safi Kalo
Andre Kelleners
Brian L. Kelly
Henry A. Kissinger
John E. Klein
*C. Jeffrey Knittel
Joseph Konzelmann
Franklin D. Kramer
Laura Lane
Almar Latour
Yann Le Pallec
Jan M. Lodal
Douglas Lute
Jane Holl Lute
William J. Lynn
Mark Machin
Umer Mansha
Marco Margheri
Michael Margolis
Chris Marlin
William Marron
Christian Marrone
Gerardo Mato
Erin McGrain
John M. McHugh
*Judith A. Miller
Dariusz Mioduski
Michael J. Morell
*Richard Morningstar
Georgette Mosbacher
Majida Mourad
Virginia A. Mulberger
Mary Claire Murphy
Edward J. Newberry
Franco Nuschese
Joseph S. Nye
Ahmet M. Ören
Sally A. Painter
Ana I. Palacio
*Kostas Pantazopoulos
Alan Pellegrini
David H. Petraeus
*Lisa Pollina
Daniel B. Poneman
*Dina H. Powell McCormick
Michael Punke
Ashraf Qazi
Thomas J. Ridge
Gary Rieschel
Lawrence Di Rita
Michael J. Rogers
Charles O. Rossotti
Harry Sachinis
C. Michael Scaparrotti
Ivan A. Schlager
Rajiv Shah
Gregg Sherrill
Jeff Shockey
Ali Jehangir Siddiqui
Kris Singh
Walter Slocombe
Christopher Smith
Clifford M. Sobel
James G. Stavridis
Michael S. Steele
Richard J.A. Steele
Mary Streett
*Gil Tenzer
*Frances M. Townsend
Clyde C. Tuggle
Melanne Verveer
Charles F. Wald
Michael F. Walsh
Ronald Weiser
*Al Williams
Maciej Witucki
Neal S. Wolin
*Jenny Wood
Guang Yang
Mary C. Yates
Dov S. Zakheim

HONORARY DIRECTORS
James A. Baker, III
Ashton B. Carter
Robert M. Gates
James N. Mattis
Michael G. Mullen
Leon E. Panetta
William J. Perry
Condoleezza Rice
Horst Teltschik
William H. Webster

**Executive Committee Members*

List as of October 20, 2022