



June 17, 2021

National Telecommunications and Information Administration  
U.S. Department of Commerce  
1401 Constitution Avenue, NW, Room 4725  
Attn: Evelyn L. Remaley, Acting NTIA Administrator  
Washington, DC 20230

Re: NTIA-2021-0001 | Docket No. 210527-0117

Cisco would like to thank the National Telecommunications and Information Administration (NTIA) for the opportunity to file these comments in response to the Notice and Request for Comments on Software Bill of Materials Elements and Considerations. President Biden's Executive Order 14028, aimed at improving the security of federal agencies, is both ambitious and necessary. Cisco is committed to maintaining strong protections for our customers, partners, products, and company. We strive to earn trust by being trustworthy, transparent, and accountable. These goals are embodied in Cisco's Trust Principles<sup>1</sup>, which map well to the software-related guidelines NTIA is required to develop pursuant to the Executive order.

Software Bills of Materials (SBOMs) provide a means to significantly improve software transparency. Cisco sits in the middle of the software supply chain, both as a consumer and a producer of software. As one of the largest software companies in the world,<sup>2</sup> with the #1 supply chain program 2-years-running,<sup>3</sup> Cisco is in full support of NTIA's software transparency initiative. We believe that SBOMs are a foundational element necessary to provide greater security and trust in all technology.

We respectfully submit the attached paper on Cisco's initial position. The paper begins with a summary of the themes that recur throughout our response, and then addresses each of the 4 questions asked by NTIA:

1. We support the US government's requirement to produce SBOMs along with software, as this will transform software transparency in the technology industry.
2. SBOM technologies and standards are nascent. Much is still unknown. Only through wide-spread adoption can stakeholders learn the lessons necessary to describe the minimum set of elements that an SBOM should contain.

---

<sup>1</sup> [https://www.cisco.com/c/dam/en\\_us/about/doing\\_business/trust-center/docs/cisco-trust-principles.pdf](https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-trust-principles.pdf)

<sup>2</sup> "Cisco now has one of the largest software businesses in the industry with an annual run rate of north of \$14 billion in software revenue, said Cisco Chairman and CEO Chuck Robbins." <https://www.crn.com/news/networking/cisco-security-webex-see-record-breaking-growth-software-run-rate-soars-past-14b>


<sup>3</sup> <https://www.gartner.com/en/newsroom/press-releases/2021-05-19-gartner-announces-rankings-of-the-2021-supply-chain-top-25>



3. SBOM specifications should be created and improved using existing community-driven, consensus-based standards, which are the foundation of the Internet.
4. The longer-term goal should be to enable automated generation of SBOMs as part of the software development process to maximize the efficiency for their production and to foster consumption of their contents in an actionable and machine-readable fashion at scale

Cisco believes that **adoption at scale** is an area that deserves substantial focus. Cisco customers may have countless devices deployed on their networks. It is not uncommon to see hundreds, if not thousands, of **types** of devices deployed in a single large network. Devices are purchased across many different departments within a customer organization, often in uncoordinated ways. Automation will enable operators to use SBOM data at scale to identify vulnerabilities in specific versions of software running on various device types across a heterogeneous network. Automation will also enable SBOM software suppliers to produce this information with minimum effort. Finally, automation will be essential given that modern software is updated with increasing rapidity. We expect the pace of software development to continue speeding up, which will challenge any model that requires human review of updates and patches before they can be applied.

We look forward to continuing to work with NTIA and other agencies on implementing these efforts and would be happy to offer relevant experts to speak to any of the topics outlined below.

DocuSigned by:  
  
04A598590121432... 6/17/2021  
Eric Wenger  
Senior Director, Technology Policy  
[erwenger@cisco.com](mailto:erwenger@cisco.com)

DocuSigned by:  
  
CC7D93CF392347B... 6/17/2021  
Jeff Schutt  
Security and Trust Architect  
[jfschut@cisco.com](mailto:jfschut@cisco.com)

Cisco Systems  
170 West Tasman Dr  
San Jose, CA 95134 USA  
Phone: 408 526-4000  
Fax: 408 526-4100



**1. Are the elements described above, including data fields, operational considerations, and support for automation, sufficient? What other elements should be considered and why?**

We believe there should be a minimal set of SBOM elements to allow for a consistent syntactical representation of an SBOM. Because the field is rapidly evolving, we are hesitant to provide a firm answer as to whether each of the NTIA-defined fields is either necessary or sufficient. In fact, we expect the answer to change, and we do not believe that NTIA should specify mandatory fields. That role should be left to community-driven, consensus-based standards organizations, as is recommended in OMB Circular No A-119<sup>4</sup>, and reaffirmed in the recent G7 Communique<sup>5</sup>.

Cisco is playing a leading role in many of these standards organizations, which are continuing to develop the formats and mechanisms to generate, share, and consume SBOM data (e.g., Software Package Data Exchange (SPDX), CycloneDX, and the Common Security Advisory Framework (CSAF)). These existing standards enable all practitioners to lend their experience to maturing the technology through operational deployment experience. This will reduce the risk of ossification or obsolescence by allowing for evolution of SBOMs in response to a vibrant and dynamic market for their use.

It is clear from ongoing community discussions that some of the SBOM elements are currently not well specified. Discussions include:

- When a hash is required, the process by which it is generated and the process by which it can be used to validate a component must be clear. A specification for SBOM element hashes should be sufficiently flexible to allow for novel approaches to software identity, such as using [GitRefs](#)<sup>6</sup>. Any attempt to codify or lock in the status quo could stunt further development of critical pathways for future SBOM use.
- An SBOM should contain a version element such that when the tooling improves, a more accurate SBOM can be generated.
- The SBOM should be deterministic and certain elements within the SBOM should be immutable.

---

<sup>4</sup> [https://obamawhitehouse.archives.gov/omb/circulars\\_a119\\_a119fr](https://obamawhitehouse.archives.gov/omb/circulars_a119_a119fr)

<sup>5</sup> <https://www.whitehouse.gov/briefing-room/statements-releases/2021/06/13/carbis-bay-g7-summit-communique/>

<sup>6</sup> <https://docs.google.com/presentation/d/1-Mm-E9lqHQAXfDviVuD4Jk5CW6dJobFaFXT1TGRsowY/edit#slide=id.p>



## 2. Are there additional use cases that can further inform the elements of SBOM?

Today, Cisco requires that SBOMs be generated as part of the Cisco Secure Development Lifecycle (Cisco SDL) and utilizes them for cybersecurity and licensing<sup>7</sup>. We anticipate that SBOMs will provide benefit to a substantial number of other use cases. The elements of an SBOM will inform the entire software supply chain, enabling new use cases across all Stages of the Value Chain<sup>8</sup> (elaborated as **Design, Plan, Source, Make, Quality, Deliver, Sustain, End-of-Life**).

The deployment of software impacts how the SBOM will be shared and used. Two divergent deployment patterns will call for different SBOM implementations:

1. Software delivered to and operated by an end-user.
2. Software delivered as-a-service to end users, operated by the software provider.

Consider an example for the first deployment pattern where software is delivered to and through the Defense Industrial Base, in support of government and critical infrastructure systems. In this scenario the operator will desire SBOMs to support their activities, such as software patching, vulnerability management, and incident response. For these high-assurance use cases the consumer will desire comprehensive SBOMs, that provide informative metadata about the software as well as a degree of confidence in the origin (provenance) and quality (pedigree) of how the software was built.

In the second as-a-service deployment pattern the end user may desire certification of the software against industry governance, risk and compliance frameworks. SBOMs should support these activities, and yet there are reasons why many of the SBOM elements necessary for the use case above would be of very limited utility to the end user who does not maintain the software underlying the service directly. The architecture of these dynamic, complex, cloud-delivered environments means that elements of the SBOM will be ever-

---

<sup>7</sup> [https://www.cisco.com/c/dam/en\\_us/about/doing\\_business/trust-center/docs/cisco-secure-development-lifecycle.pdf](https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-secure-development-lifecycle.pdf)

<sup>8</sup> <https://www.cisco.com/c/en/us/about/trust-center/global-value-chain-security.html#~building-trust>



changing. The value derived from SBOM consumption in the cloud operator's use case may require a different set of elements in the SBOM than those shared with the end-user in the first deployment pattern. Future work is needed to determine how to dynamically generate, share and consume SBOMs at the point when third-party dependencies are loaded at runtime or when executing a workload that may reside in a different host or domain.

**3. SBOM creation and use touches on a number of related areas in IT management, cybersecurity, and public policy. We seek comment on how these issues described below should be considered in defining SBOM elements today and in the future.**

**3a. Software Identity: There is no single namespace to easily identify and name every software component. The challenge is not the lack of standards, but multiple standards and practices in different communities.**

More work is needed to standardize on a mechanism for SBOMs to provide accurate software identity information. Careful consideration is required to ensure that software **identifiers**, **locators**, and **provenance** information are always treated distinctly and not commingled. Existing SBOM elements like **Author Name**, **Supplier Name**, **Component Name** and **Version String** are **not** identifiers of actual software, since they are not collision resistant.

**Identifiers:** Name space requirements will hinge on how that space is to be used. Automated validation of the supply chain could be performed with only statistically unique names, but if human-readability is required, then a **structured** name space is necessary to ensure no name collisions. The URI specified in IETF RFC 3986 is an example.<sup>9</sup>

**Locators:** When there are dependencies on other systems on the Internet, a locator to **those** services and their respective SBOMs is required. There is a very mature system on which to base that locator: the URL as a subset of the URI. Other standards, such as Package URLs, already make use of such a standard. It must be possible to secure the binding of any name that is used.

**Provenance:** While it is beneficial to understand software provenance, the identity of software does not change based on the provenance of the software. The identity, therefore, must not be based on the Author Name or Supplier Name SBOM elements. If the software was changed by midstream actors in the supply chain, the software identity element of the SBOM should change correspondingly with this new software.

---

<sup>9</sup> <https://datatracker.ietf.org/doc/html/rfc3986>



**3b. Software-as-a-Service and online services: While current, cloud-based software has the advantage of more modern tool chains, the use cases for SBOM may be different for software that is not running on customer premises or maintained by the customer.**

See answer to question 2 above. Additionally, Software-as-a-Service composition may vary from one user to the next, and from one instance to the next. Hybrid-cloud deployments add complexity to SBOM generation, on-device-storage, and sharing as customer-managed devices are dependent upon ever-changing software in cloud services to function. We recommend that the utility and means by which to produce, share, and consume SBOMs for software-as-a-service and online services be given time to mature through appropriate existing community-driven standards organizations.

**3c. Legacy and binary-only software: Older software often has greater risks, especially if it is not maintained. In some cases, the source may not even be obtainable, with only the object code available for SBOM generation.**

We agree that older software tends to be associated with greater risks. We do not believe that lack of access to source is a “legacy” matter, but rather the current state of affairs for the vast majority of deployments. It is important to establish the linkage from development of source to inclusion of third-party products, generation of binaries, and their distribution for use, so that the entire supply chain is secured. We believe SBOMs can improve this situation by articulating relationships and enabling traceability through the process.

**3d. Integrity and authenticity: An SBOM consumer may be concerned about verifying the source of the SBOM data and confirming that it was not tampered with. Some existing measures for integrity and authenticity of both software and metadata can be leveraged.**

Information, where the source cannot be authenticated, cannot be trusted. Therefore, it is critical that both software and its metadata be signed by an entity that can be easily identified **at Internet Scale**. This means that SBOMs and other relevant system information must be capable of being automatically discovered, retrieved, and authenticated. The trust models for doing all of this must be simple and integrated with one another to avoid transferring complexity to those least able to manage it. We believe further work in the application of



integrity and authentication mechanisms to software and its metadata, e.g., SBOMs, through existing, community-driven, consensus-based standards is necessary to achieve the outcomes desired.

**3e. Threat model: While many anticipated use cases may rely on the SBOM as an authoritative reference when evaluating external information (such as vulnerability reports), other use cases may rely on the SBOM as a foundation in detecting more sophisticated supply chain attacks. These attacks could include compromising the integrity of not only the systems used to build the software component, but also the systems used to create the SBOM or even the SBOM itself. How can SBOM position itself to support the detection of internal compromise? How can these more advanced data collection and management efforts best be integrated into the basic SBOM structure? What further costs and complexities would this impose?**

It is worth considering that there are costs to producing, transmitting and storing SBOMs. And that there are new areas of risk that may be opened up as a result of their existence. On balance, however, we believe there will be significant net security benefits to the production and consumption of SBOMs. Threat models should be developed for various SBOM use cases to ensure the integrity and authenticity of the SBOM data, as described above. The trustworthiness of SBOMs themselves, as well as other artifacts, will depend upon a substantial number of **other** processes which must be put in place to protect the supply chain. We advise caution and application of the NTIA's crawl, walk, run approach when considering how SBOMs can support more advanced data collection and management efforts. From an engineering perspective, the purpose and value of an SBOM should be properly bounded so that it is not "all things to all people," which is the equivalent of "nothing to nobody." It is not yet clear what costs and complexities would be imposed by adding the more advanced elements proposed by NTIA to the SBOM. The goal should be to maximize benefits and reduce costs. Modeling threats from misuse of SBOMs will be a key element in helping to strike the right balance between risk and reward and will potentially inform decisions about what information should be shared—and with whom.

**3f. High assurance use cases: Some SBOM use cases require additional data about aspects of the software development and build environment, including those aspects that are enumerated in Executive Order 14028. How can SBOM data be integrated with this additional data in a modular fashion?**

Any format should be sufficiently extensible to incorporate metadata about the environment used to create a component. However, such information should be optional for the foreseeable future for two reasons. First, substantial tooling automation would be needed to populate such fields. There is very little open-source software available for such integration





today. Cisco is interested in participating in such activities, not only as a producer, but as a consumer of software.

Second, metadata for one use case (build environment artifact creation) can create a different version of an SBOM created for another use case (identifying software components that a CVE may apply to). This will result in multiple distinct SBOMs describing the same software. Even worse, there can potentially be conflicting information in these differing SBOMs, without meaningful distinction as they describe the same software. For example, different build commands **can** and **do** produce identical bytes of output. Orthogonality of metadata is essential in order to prevent these types of issues. To solve for this, the SBOM itself should have a Unique Identifier that is immutable and does not incorporate ephemeral metadata like the filename, what system it is on, the timestamp, etc. These additional sets of metadata for each use case should be delivered as multiple documents. In this way the SBOM can be extensible without needing to change every time a different use case is considered.

**3g. Delivery. As noted above, multiple mechanisms exist to aid in SBOM discovery, as well as to enable access to SBOMs. Further mechanisms and standards may be needed, yet too many options may impose higher costs on either SBOM producers or consumers.**

As we have noted throughout these comments, delivery of SBOM data must occur at Internet scale. It must be possible for end deployments to discover, access, and retrieve SBOMs along with other information about a device. Significant industry adoption is required to learn the lessons necessary to meet these demands at scale. Cisco supports innovation in these areas and anticipates market forces will determine the appropriate number of options.

**3h. Depth. As noted above, while ideal SBOMs have the complete graph of the assembled software, not every software producer will be able or ready to share the entire graph.**

We agree with this sentiment. The goal for SBOM distribution and use should be to enable sharing of actionable information between developers of software and those persons or entities who will be deploying it. There are entire classes of code that cannot nor should not be disclosed. In certain cases, there may even be an adversarial relationship between a device provider and the device user, a classic example being set-top devices. For years, there was a technology race between hackers aiming to steal content and content providers. Only through the maturing of encryption and code obfuscation technology did this race largely come to an end. Revealing of structure of code in such cases harms cybersecurity. The beneficiaries of such obfuscated code are likely to have an incentive to keep it free of





vulnerabilities. The purpose of the NTIA effort is to align incentives in the supply chain. In the above case, incentives are already aligned.

**3i. Vulnerabilities. Many of the use cases around SBOMs focus on known vulnerabilities. Some build on this by including vulnerability data in the SBOM itself. Others note that the existence and status of vulnerabilities can change over time, and there is no general guarantee or signal about whether the SBOM data is up-to-date relative to all relevant and applicable vulnerability data sources.**

Today our end customers are asking us:

- Are there devices in my network that contain code with known vulnerabilities?
- Are the devices on my network vulnerable to threats as a result?
- Are there known mitigations for the risks associated with those vulnerabilities?

These questions cannot be answered without accurate vulnerability information. Cisco has provided that accurate information for three decades. The industry is now evolving the methods to provide and consume this information. Those methods must support Internet scale.

An SBOM should reflect whatever components are used in a specific version of software at a particular moment in time. The SBOM generated for a given version of software should be deterministic. The SBOM should be updated as the software evolves through new features, code refactoring, bug fixes, and other means.

Vulnerability information should be updated to reflect changes in software security. We do not believe a new SBOM should be generated because of a change in the vulnerabilities known to affect an embedded component. Said differently, SBOMs should not be dependent upon ever-changing vulnerability information.

**3j. Risk Management. Not all vulnerabilities in software code put operators or users at real risk from software built using those vulnerable components, as the risk could be mitigated elsewhere or deemed to be negligible. One approach to managing this might be to communicate that software is “not affected” by a specific vulnerability through a Vulnerability Exploitability eXchange (or “VEX”),<sup>14</sup> but other solutions may exist.**

As mentioned in our previous answer, having accurate vulnerability information upon which to act is critical. That's why Cisco is leading the drive for the Common Security Advisory Format (CSAF), a good candidate format for a VEX. Absent this information, there will be many false positives that tooling will have difficulty attempting to sort. An SBOM should



reflect whatever components are used at a particular moment in time. The vulnerability disposition of those components should also accurately reflect the posture of a system using those components. While the SBOM must be able to link to a VEX, the ability to modify a VEX based on a newly published CVE should not require a change to its associated SBOM.

**4. Flexibility of implementation and potential requirements. If there are legitimate reasons why the above elements might be difficult to adopt or use for certain technologies, industries, or communities, how might the goals and use cases described above be fulfilled through alternate means? What accommodations and alternate approaches can deliver benefits while allowing for flexibility?**

Please see our responses to Questions 1 and 3(c).