ncta

The Internet & Television Association
25 Massachusetts Avenue, NW | Suite 100
Washington, DC 20001

(202) 222-2300

Rick Chessen
Chief Legal Officer
Senior Vice President, Legal & Regulatory Affairs

o (202) 222-2445 e rchessen@ncta.com

June 17, 2021

Evelyn L. Remaley
Acting Administrator
National Telecommunications and Information Administration
1401 Constitution Ave., NW
Room 4725
Washington, D.C. 20230

Dear Ms. Remaley:

NCTA – The Internet & Television Association (NCTA)[1/] submits these comments in response to the Request for Comments (RFC) issued by the National Telecommunications and Information Administration (NTIA) seeking public input on the minimum elements for a Software Bill of Materials (SBOM) and what other factors should be considered in the request, production, distribution, and consumption of SBOMs.[2/] NTIA was directed by Executive Order 14028, "Improving the Nation's Cybersecurity", to publish the minimum elements for the SBOM. NTIA has invited comments on the "full range of issues", welcomed input on "elements that may be difficult to adopt or use", and invited commenters to propose "alternate means" to fulfill the goals described in the RFC.[3/]

Overview. NCTA supports the conceptual objectives underlying the NTIA effort to develop an SBOM. There is value in examining and assessing the degree to which improved transparency regarding software components can improve cybersecurity outcomes in connection with products and services reliant upon such software. A workable SBOM has the potential to benefit the broader internet ecosystem, including purchasers of software-driven network gear and critical software, as well as end users of software applications, IoT devices, personal computing devices, and home networking equipment.

Despite these laudable objectives, there are numerous key operational challenges in

[1/] NCTA is the principal trade association of the cable television industry in the United States, which is a leading provider of residential broadband service to U.S. households. Its members include owners and operators of cable television systems serving nearly 80% of the nation's cable television customers, as well as more than 200 cable program networks

[2/] Department of Commerce, National Telecommunications and Information Administration, *Software Bill of Materials Elements and Considerations*, Docket No. 210527–0117, 86 Fed. Reg. 29568 (June 2, 2021) (RFC).

[3/] *Id.* at 29570.

designing and implementing an SBOM that preclude its immediate use across all industries. As explained in greater detail below, these concerns are the lack of (i) mature tools to support an SBOM; (ii) infrastructure for managing the naming space; and (iii) an identified approach to securing the confidentiality, integrity and availability of the SBOM. Based on these shortcomings, NCTA recommends that NTIA conduct limited Federal agency trials and voluntary industry pilots to identify and address risks and gaps.

There Is Value in Exploring the Workability and Security Effects of the SBOM Concept. The Executive Order defines an SBOM as "a formal record containing the details and supply chain relationships of various components used in building software."[4/] A key objective of the SBOM concept is to provide more visibility into software componentry in order to give organizations more information about the building blocks of packaged software they are considering, which can aid in the overall risk assessments of their software-based products and services and better inform organizations of known vulnerabilities associated with components in the software.

This emerging technology's potential should be explored, while recognizing that significant operational challenges remain unaddressed. Software typically is not built from scratch, but is developed by combining components, development frameworks, libraries, operating system features, and other elements. A software application may be comprised of hundreds or even thousands of components and sub-components contributed by an array of developers and licensors. There is, conceptually, a "bill of materials" describing the elements and ingredients that comprise a software offering, just as there is for hardware products.[5/] When vulnerabilities emerge, identifying the affected ingredients and elements at risk and assessing their role and functions in software applications utilized by end users in products and services could be helpful in mitigating the harms and risks of such vulnerabilities.[6/]

An SBOM Should Not Be Viewed as Primary Mechanism for Addressing Software Security Concerns. SBOMs are one of the many tools that can be used to improve software security. As the NTIA has noted, one size does not fit all. Designing an SBOM that could provide incremental security benefits for software developers, vendors, product and service integrators, and end users is an especially daunting task that is not without risk. SBOM's ultimate value will depend upon the degree to which its operational challenges can be surmounted (as described below). But there are a variety of widely accepted practices already

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

[5/] Software applications operate on top of a stack of software components that sit between the application and the physical hardware. The stack can be very different for different types of hardware, particularly for IoT devices, and will also vary depending on whether the device is operated by a processor, micro-controller, network controller etc. Some hardware may run an operating system while other hardware will allow software to make direct calls to the controllers for vulnerabilities (intentional or accidental) in the stack which may not be tracked in a SBOM of the application running on top.

[6/] For example, the Poodle Attack that exploited a component vulnerability in the Secure Socket Layer (SSL) used in many operating systems and applications like web browsers to secure e-commerce transactions. https://us-cert.cisa.gov/ncas/alerts/TA14-290A.

employed that, if followed rigorously, help to generate confidence in the security of software, such as threat modeling static analysis, dynamic analysis, fuzzing, pen testing, and bug bounties.[7/] SBOM has the potential to be an additional tool, but it should be used to supplement – and not supplant – these practices. NTIA should carefully examine the costs and benefits of an SBOM, as well as the degree to which the security outcome objectives associated with one or more SBOM elements outcomes can be achieved through other methods. Further, like any tool, there are certain problems that an SBOM cannot address adequately, such as prevention of zero day attacks. At best, the SBOM may help reduce the spread of, and raise awareness around, such vulnerabilities.

The Potential Security Risks Associated with an SBOM Should Be Addressed and Mitigated. An improperly executed SBOM could provide information to attackers that would make the software supply chain more vulnerable, not less. It could provide a "road map" to all of the vulnerabilities of an application and increase the "threat surface" through which the application can be exploited, and to a threat actor it could be an especially useful list of software to readily exploit to the extent a zero-day vulnerability is later found in components. While it may be true that many of the individual elements of an SBOM are available to malicious actors in other forms and from other sources,[8/] it is nonetheless worthwhile to consider whether compiling such a list in a single format facilitates the work of attackers by providing them with a readily available means to map the prevalence of common element vulnerabilities and whether there are measures to countervail that risk.

Operational Challenges. There are key operational challenges associated with designing and implementing an SBOM, including the following.

Maturity of tools to support SBOM. Given that the concept of an SBOM is relatively new, the tools that do exist are relatively new and there are still gaps in the tool chain to produce, consume, and transform SBOMS. In addition to the categorization of tools noted by the SBOM working group, additional tools need to be developed to support the automated cross-referencing SBOMs with published Common Vulnerabilities and Exposures (CVEs) as part of any risk management and/or vulnerability management program.

It may not be easy to automate the process of producing SBOMs or developing tools to map software components to known vulnerabilities. Many software vendors already apprise their customers of known vulnerabilities they may be addressing when releasing updates or patches. When a subcomponent of a software application has a vulnerability or bug, the end user often may need to only patch product elements using that subcomponent, rather than all the

[7/] Static threat analysis involves reading a file or code and looking for vulnerabilities or malware. Dynamic threat analysis entails examining the behavior of software (sometimes in a safe environment) to determine if it has vulnerabilities. Fuzzing or fuzz testing consists of providing random invalid or unexpected input data to a computer program to test the manner in which the software handles such data. Pen testing employs authorized cyber attacking to evaluate system or software security. Bug bounties typically involve contests with compensation offered by software makers for users that find software bugs or vulnerabilities.

[8/] SBOM FAQs, Nov. 16, 2020, at 5-6, https://www.ntia.gov/files/ntia/publications/sbom_faq_-_20201116.pdf.

functionalities dependent upon that application. But the process of isolating vulnerabilities and aligning them with subcomponents is challenging, and in many instances takes manual validation.[9] The absence of automated tools for reliably aligning vulnerabilities with their source and origin limits the utility of any particular SBOM.[10]

Infrastructure for managing the naming space. As the RFC notes, there "is not single namespace to easily identify and name every software component," with the key challenge being less a lack of standards and more the presence of multiple standards and practices in different industries and communities.[11] An SBOM that conveys a software application's utilization of a particular open source software component may be highly incomplete with respect to components with millions of lines of code, unless it goes further and identifies the specific coding sequences being employed and their purpose and functionality in relation to the application as a whole. Without consistency and uniformity across SBOMs, the component name could end up signifying a variety of different software processes and functions, which undermines the utility of the SBOM. Further, while naming and delineation at the module or sub-component level may provide information that is more actionable and targeted, it risks both producing information overload and still overlooking relevant and commonly used ingredients that may reside below layers of sub-components.[12]

While the depth of the naming issues can extend to all layers of the component and sub-component stack, the breadth of those issues is also challenging. SBOM use will not be limited to cybersecurity or IT operational teams but will extend to software purchasers and product and service integrators, who will need to be trained on how to review SBOMs and interpret SBOM data and analysis. This is likely to be a significant undertaking, but one that is critical to realizing value from the SBOM. To the extent that SBOMs become a key part of the purchasing decision process, accuracy in both the SBOM and the purchaser's interpretation of the SBOM information have potentially large commercial impacts.

Securing the SBOM to ensure its confidentiality, integrity, and availability. It is critical that security considerations be front-loaded into the design of SBOMs. Bolting on security measures after the design and development of SBOMs risks replicating mistakes in other parts of the internet ecosystem, such as DNS and Internet routing – which have been plagued by security issues and sub-optimal solutions due to the lack of focus on security in their initial design. Identifying and incorporating the appropriate range and complexity of security tools and

[9] These challenges are amplified with, for example, IoT devices using field programmable gate arrays (FPGAs), since such devices exist in between hardware/software domains and may elude coverage under security-related frameworks such as an SBOM or export control requirements.

[10] The RFC does not envision integrating the Common Vulnerabilities and Exposures reference system for publicly known information-security vulnerabilities and exposure with the SBOM. While such integration might make sense, it could also prolong and complicate SBOM design and development, and may be a value-add gap that could be filled by the private sector in appropriate circumstances and perhaps eventually integrated as tools mature.

[11] RFC, 86 Fed. Reg. 29570.

[12] Most software components are compiled into machine readable instructions. In theory, compilers themselves can contain vulnerabilities, but including compiler versions in an SBOM would make its implementation and update process an even more cumbersome and time-consuming exercise for a low-likelihood vulnerability.

measures to incorporate into an SBOM will require careful consideration of the trade-offs at stake. For example, cryptographic hashing of software components included in an SBOM could enhance the integrity, but also impose a significant level of effort to the software building process. Mismatched hashes could be the result of very benign changes in code or changes in compiler versions and could be difficult to trace back.

Access and authentication protocols will be critical design elements for a variety of reasons. As noted above,[13/] malicious actors may seek access to SBOMs in order to find common ingredients across applications that offer a vulnerability exploit that could be used to launch an attack at scale and reduce the costs and complexity of such attacks. In addition, software componentry may be highly valuable from an intellectual property standpoint, and SBOMs thus may be an attractive target for corporate espionage and data exfiltration.

Lastly, there remain important questions about the degree to which developer incentives under an SBOM align with the initiative's overall security objectives. SBOMs must be designed to avoid incentivizing developers to focus only on disclosure of relevant ingredients and vulnerabilities at the expense of secure design and rapid remediation of vulnerabilities. They should operate in a manner that averts shifting ultimate responsibility for preventing and mitigating harm from software vulnerabilities from the developers themselves to entities further down the stack.

Voluntary Approaches and Limited Agency Trials. Given the nascency of this undertaking, and lack of mature tools and proven processes, we strongly support NTIA's view that use of SBOMs should be voluntary.[14/] They should also be rooted in risk assessments that ascertain where and how they can accrue the most benefits for software developers, vendors, product and service integrators, and end users. The use of SBOMs should be market-driven and sector-based**,** as it will be impossible to deploy SBOM internet-wide or sector-wide overnight. SBOMs applied to a PC or IoT devices will differ markedly from those pertaining to network applications software, critical software, and cloud-based services.

NTIA's work to date has focused on developing a broadly applicable concept, but in practice SBOMs will need to be tailored to accommodate variances and unique characteristics and circumstances among different types of sectors, enterprises, and entities. By taking a risk-based market driven approach, the deployment of SBOM can follow the "crawl, walk, run model" that is often used when developing and deploying new technology and standards such as an SBOM.

NTIA should continue to support private sector efforts to develop industry/sector pilots that are underway to identify gaps and resolve kinks. NTIA also should gain real-world experience with utilizing SBOMs by administering trials at select government agencies or sub-departments, working initially with the largest software vendors supplying the selected

[13/] *See supra* at text accompanying n. 8.

[14/] *See David J. Redl, NTIA Launches Initiative to Improve Software Component Transparency, Nat'l Telecomm. & Info. Admin. (June 6, 2018),* https://www.ntia.doc.gov/blog/2018/ntia-launches-initiative-improve-software-component-transparency.

government entities. Such trials will help to generate a compendium of lessons learned, best practices, and recommended tools and processes that could perhaps eventually be applied across a larger swath of government agencies.

NTIA should work with industry to develop an achievable roadmap based on incremental progress only after further study and analysis, and before making recommendations that could bind design and use of SBOMs for Federal agency software purchases. And SBOMs should remain a voluntary, sector and sub-sector-based mechanism for private sector commercial transactions. Best effort SBOM creation supported by well-defined and road-tested processes may still result in inaccuracies due to the complexity of software components that may span multiple industries, third parties, and technology stacks – all of which weighs in favor of prudence and deliberation in moving forward with this initiative.

Minimum Elements of an SBOM. With respect to the minimum elements or components of an SBOM, the baseline component information or data fields listed in the RFC are likely adequate for the limited and voluntary agency-based trials suggested here. However, to fully address security concerns associated with an SBOM, NTIA should consider whether to add a digital signature field to support digitally signing the SBOM to help ensure the authenticity of the SBOM. While it could add some additional complexity and cost, it is a common feature of digital data exchanges and therefore should at least be considered as an optional data element. Additionally, not only is a digital signature important to validate the integrity and authenticity of an SBOM itself, NTIA also should consider approaches to ensuring the integrity and authenticity around delivering and sharing the SBOM with its intended recipients. This too could be considered as an optional data field designed to bolster SBOM security and mitigate the prospect of proprietary or multiple SBOM access and delivery standards from emerging as a result of avoiding digital rights management (DRM) issues on the SBOM.

Conclusion. For the reasons set forth above, NTIA should continue its effort to craft the minimum elements of a workable SBOM, and carefully work through the significant operational challenges posed by this initiative. To that end, NTIA should assess the viability of the minimum elements it opts to initially adopt through a set of limited Federal agency trials and voluntary industry pilots. These trials and pilots will help identify gaps and generate lessons learned and best practices that can be further adapted as the SBOM iteration process moves forward.

Respectfully submitted,

/s/ Rick Chessen

Rick Chessen
Loretta Polk
NCTA – The Internet & Television Association
25 Massachusetts Avenue, N.W. – Suite 100
Washington, DC 20001-1431
(202) 222-2445

Matt Tooley
Vice President, Broadband Technology

June 17, 2021