Response of Red Hat, Inc. (“Red Hat”) to NTIA’s Request for Comment ("RFC") on Software Bill of Materials Elements and Considerations [Docket No. 210527–0117] [RIN 0660–XC051] June 17, 2021

Red Hat appreciates the opportunity to comment on the above-referenced matter in anticipation of the guidance on software supply chain being prepared by NIST. As the leading provider of open source software solutions (using a community-powered approach to deliver resilient and high-performing cloud, Linux, middleware, storage and virtualization technologies), Red Hat welcomes the release of the Executive Order on Improving the Nation’s Cybersecurity (“EO”).

As the Department of Commerce undertakes its various directives under the Executive Order, its work should focus on how to enhance transparency, repairability, and resiliency, all essential to the trustworthiness of US government software assets. The strategic objective of the EO -- and the broader longer-term goal -- is to mature software supply chain assurance practices. An SBOM will not solve the issues of software supply chain security by itself. It is just one piece of the puzzle, perhaps in some respects a useful one, but simply one piece. The continuation of already on-going work on SBOMs where there are agreed upon use cases will potentially allow this concept to become more valuable over time.

With this context, Red Hat offers the following insights based on our experience in routinely responding to our customers on questions touched on in the RFC.

Red Hat has for many years disclosed and has been transparent[1] about our software products, including many of the elements identified in the RFC.[2] These efforts are integral to the critical role of Red Hat in developing and supporting the *full life cycle of our open source software offerings* for our customers in the US government, critical infrastructures, and many mission-critical environments. Our lifecycle is one of curation and management.

First and foremost, **NTIA should keep its recommended requirements simple and minimal**. At this time, when SBOMs are still very much a work-in-progress, priority should be

[1] See the landing page for our Customer Portal where we provide security data on an on-going basis, at https://www.redhat.com/security/data/metrics/.

[2] The kind of information outlined in the RFC (e.g., supplier name, component name, version of the component, any other unique identifier) is typically readily obtainable by our customers from running systems. If they need to verify a component, they are directed to the fact that almost every deliverable is cryptographically signed and can be verified on the running systems directly. Much (but not all) of this information is access-controlled for customers and users.

on what would typically be found in a common 'bill of materials'. The purpose of an SBOM is to accurately and correctly convey the components that have gone into a particular product offering. Using SBOMs beyond that purpose makes them unwieldy and diminishes their utility. They are not a substitute for good software management security practices.

The recommended minimum requirements should not require a specific format or tooling,[3] nor solve the questions of SBOM delivery and communication, but rather provide the data elements bulleted in the RFC utilizing a machine readable open data format.[4] An SBOM provided by a software vendor should appropriately indicate the third-party component(s) incorporated into the product. But tracking all dependencies for the sake of tracking all dependencies at all levels should not be included as a minimum requirement in each and every SBOM.[5]

Developing and building software components and products is an inherently iterative process which can result in an extensive source code historical record or blueprint for new products. This full history of code components will be overinclusive to the user. This is especially true given the apparent intended audience of the anticipated guidance as articulated in the EO: the purchaser.

Discussions on SBOMs are continuing (and will be on-going) after June 17 (most likely driven by sector-specific environments and requirements). NTIA should avoid making recommendations that seek to solve all the issues around or attempt to outline a universal SBOM in the limited time it has under the EO.[6] **We strongly agree with NTIA, as stated in the RFC, that SBOMs can ride on 'existing mechanisms' and such delivery can 'reflect the nature of the software as well'.**[7]

Over the years, we have found that customers have a variety of perspectives on what kind of product component information is important to them. Their requests will derive from the vantage the requester has inside the particular customer's operations -- as a developer, procurer, security practitioner, business leader, IT support, or other. Integrating the myriad of

---

[3] In this regard, the definition of SBOM found in section 10 of the EO is not a real world definition of on-the-market-today solutions beyond research, proof-of-concept and demonstration projects. There are a variety of reasons for the stark reality that SBOMs, as such, are not widely utilized in the software industry today. A great deal of work remains to be done to scope, define, document, and operationalize (especially with other management tools) SBOMs and to make commercial software SBOMs provided to customers commonplace.

[4] This is along the lines of the approach taken in the OASIS Common Security Advisory Framework (CSAF) Common Vulnerability Reporting Framework (CVRF).

[5] If warranted, dependency information could be requested by the government purchaser to the product vendor.

[6] SBOMs as such are not yet widely employed across all software use environments, and the Department of Commerce should be cautious in stating 'best practices' in this area. The state of work currently underway is still in very much at a 'beta' stage, and is largely sector or technology specific (e.g., energy sector, medical devices). Real-world deployments are distinctly lacking. Rushing headlong into promoting a 'system' and bypassing these efforts would cause confusion in the marketplace, producing poor quality results for the US government, and even identifying the _value_ of what the system is (or who it's valuable to).

[7] Given the key definition of 'critical software' awaits action and feedback, this is especially the case. The definition of "critical software" will determine the scope of the guidance NIST develops. Unfortunately, section 4(g) merely states that NIST "shall publish a definition of the term 'critical software' for inclusion in the guidance" without any requirement to get input from the private sector. It is essential that this definition be narrowly tailored to address confidentiality, integrity, and availability, as directed in the Executive Order. **We strongly recommend that the Secretary of Commerce, acting through NIST, seek public input on this fundamental element *before* the Secretary of Homeland Security, acting through the Director of CISA, provides agencies a list of categories of software and software products in use or in the acquisition process that meet this definition.**

perspectives of an enterprise into the minimum requirements will be impractical, overly inclusive, cost-ineffective, and very likely promote confusion for an already complex federal government procurement process.

Moreover, the market has not shown that customers have gravitated to a particular format or technical implementation regarding how the requested data should be provided or displayed. Customers are often satisfied with existing information or generated manifests that answer their specific questions, *if they cannot otherwise locate the information they are seeking*. In this regard, NTIA should provide to NIST the basic, relevant common data fields to satisfy its responsibilities under the EO. The Department of Commerce, through NIST, should continue to collaborate with Industry to further industry efforts to develop automation to deliver an SBOM that a customer can consume as they require, remaining technology neutral for future iterations and evolving data.

The NTIA minimum SBOM requirements should not seek to be a one stop shop for all security questions. That would be neither practical, appropriate, nor cost effective. Different elements of a software product build rightly have differing security (or security assurance) focus of attention. For example, a cryptographic library may have a FIPS-140 certification, but may not need to have that information as part of an SBOM, *especially when the information is readily available elsewhere*.

We strongly discourage the inclusion of known or potential vulnerabilities in the SBOM minimum requirements. Disclosure of all known or potential vulnerabilities could in effect offer bad actors a roadmap to customers' systems who are using older versions where the issue is not yet fixed (especially where the customer has not updated its software) or cannot be fixed at the moment. In cloud services, where there may be even faster fixing, the inclusion of "vulnerability information" will quickly become out of date as the remediation is pushed out within days or even hours.

From a practical operational perspective, **the CVE process is distinct from any software build process,[8] and it is ill advised and simply not realistic to attempt to remediate via an SBOM**. We refer NTIA to the Red Hat Risk Report for an understanding of how quickly vulnerabilities affecting our products are addressed. To include this kind of information as a minimum SBOM requirement will promote confusion for existing customers, disrupt existing channels of communicating critical information, and pose greater risk to software resiliency. We note, again, the recognition in the RFC that there is a vital role to utilize 'existing mechanisms' and making sure that the recommendations 'reflect the nature of the software as well'**.**

An SBOM should be a factual list of ingredients and not an assessment as to whether the ingredients are "good" or "bad". An SBOM is not intended to draw conclusions on its suitability or fitness at any given point in time. The minimum requirements should be data field specific, not format specific.

[8] Known vulnerabilities are typically discovered through mechanisms such as vulnerability scans. An SBOM *could* possibly augment or assist in scanning but it would certainly not be considered a replacement for a vulnerability scanner that would have more timely and complete information. From a functional perspective, an SBOM is issued at the point of product release and should not change based on the discovery of vulnerabilities, whereas other metadata such as found in OVAL or CVRF could need to be updated.

Fundamentally, **SBOMs do not provide (and should not be relied on to be) the mechanism for software assurance, supply chain security, incident management, etc**. They are one potential ingredient of supply chain 'risk management', and merely an element for consideration in the development of the guidance that NIST is directed to provide. Many of the questions posted in the RFC can (and likely will) be addressed in that guidance (e.g., automation, software integrity, etc.). Indeed, while an SBOM may provide input data, it does not substitute for focus on the end goal[9]: to bolster software vendor supply chain assurance practices and mature a continuous, dynamic software life cycle management approach that puts trust and resiliency as a critical customer focus.

To achieve the greatest degree of success, NTIA should:

- Keep it simple and minimal -- along the lines of a traditional bill of materials -- rather than trying to address a broader, universal approach to SBOM, given the current state of market acceptance and development. The required fields should be provided in a machine readable open data format.
- Recognize that SBOMs are not, in and of themselves, a solution to the cyber security challenge that the EO seeks to address. They are one element of the broader guidance that the EO directs NIST to produce which will likely engage many of the issues and questions posed in the RFC.
- Acknowledge that software vendors like Red Hat already provide transparent and on-going information about our products. To the greatest degree possible the NTIA recommendations should provide that SBOMs can ride on 'existing mechanisms' and delivery can 'reflect the nature of the software as well'.

Red Hat appreciates this opportunity to share our experiences, observations, and recommendations via this RFC. We offer to NTIA (and the Department of Commerce) that if there are any questions or desire to delve more deeply into any aspect of our submission and other topics related to the RFC, please do not hesitate to contact us.

Contact:

Mark Bohannon
Vice President, Global Public Policy
& Associate General Counsel
Red Hat, Inc.
markb@redhat.com

Vince Danen
Senior Director, Product Security
Red Hat, Inc.
vdanen@redhat.com

[9] An SBOM should be related to a base or default install. Customers can and do add or subtract from that install, so any data that a vendor like Red Hat assembles is only as good as the recommended or default install.