Hewlett Packard Enterprise

June 17, 2021

Ms. Evelyn L. Remaley
Acting Administrator
National Telecommunications and Information Administration
U.S. Department of Commerce
1401 Constitution Avenue NW, Room 4725
Washington, DC 20230
Via email: SBOM_RFC@ntia.gov

Re: Software Bill of Materials Elements and Considerations (RIN 0660-XC051)

Dear Acting NTIA Administrator Remaley:

Hewlett Packard Enterprise is pleased to provide input on the questions set forth in NTIA's request for comments on *Software Bill of Materials Elements and Considerations* as part of the *Executive Order on Improving the Nation's Cybersecurity*.

Hewlett Packard Enterprise (HPE) is an independent, publicly traded U.S. technology company headquartered in Houston, Texas. We maintain a comprehensive technology portfolio that includes cloud computing solutions, data center infrastructure, IT for data and analytics, high performance computing, and networking equipment. HPE helps customers across government and in a range of industries unlock the potential of their data assets with software solutions such as architecture for data analytics, service identity platforms, and software platforms to support the entire machine learning lifecycle.

Overall, HPE applauds NTIA's work to date on the development of SBOM standards, especially the approach recognizing distinct technologies that can be brought to bear on the problem. As we look forward, it is important to note that efforts related to the establishment of a standard for a SBOM will need to be reconciled with the changes that have occurred in the software ecosystem for the packaging and running of applications. Most enterprises have or are in the process of transitioning to container-based runtime models and cloud native type platforms. These ecosystem evolutions bring with them additional artifact types and levels of complexity in the construct of overarching "manifests" that contain hundreds of individual artifacts. Further, given the broad adoption of Cloud Native Computing Foundation (CNCF) software and associated deployment models, the need for a standard that contemplates the CNCF ecosystem alongside the more traditional software deployment architectures and models is acute.

Any reasonably sized cloud platform implementation will contain artifacts that consist of:

- Container images
- Configurations (either file-based or in git repositories)
- Open Policy Agent bundles
- Web Assembly bundles
- Helm charts
- Repository maps and definitions
- Falco security scanner configurations
- Kubernetes platform plugins

This is by no means an exhaustive list as it keeps growing, and it is well worth reviewing the broad scope of Open Container Initiative (OCI) artifacts that are published here: https://artifacthub.io/.

As the cloud native ecosystem has evolved quickly, standards around artifacts are relatively recent. The OCI (https://opencontainers.org/) has produced working standards that are, for the most part, broadly accepted across the major participants in the Cloud Native Computing Foundation and the Linux Foundation and the ultimate software consumers. The OCI specs that should inform the further development of artifact level SBOM standards are:

- The OCI Registry spec, which defines the functionality of any registry (repository) of OCI artifacts: https://github.com/opencontainers/distribution-spec . (Important note: OCI Registries are artifact format agnostic)
- The OCI Image Format Specification: https://github.com/opencontainers/image-spec
- The OCI Artifact Project: https://github.com/opencontainers/artifacts

While the OCI specifications for artifacts provide a pathway forward for atomic SBOM components (with the https://Sigstore.dev project providing a clear pathway to artifact signature/provenance), the CNCF ecosystem does not currently have a broadly agreed approach to a platform level SBOM.

HPE recommends that NIST engage with the CNCF and Linux Foundation regarding a workable approach to these complex manifests. The manifest design will necessarily have to accommodate ongoing operational activities (update at the atomic artifact level, configuration change, etc.) and be a reliable and accurate measure of the running platform at any point in time.

HPE responses to specific questions in the RFC:

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

HPE recommends the development of a reproducible standard for naming "component" elements. A reproducible standard will ensure that industry is not relying on an incomplete dictionary (NIST's Common Platform Enumeration (CPE)). The standard should also address naming collisions.

HPE recommends that the following data fields be added to the "baseline component information":

- Source URL: This will help to ensure provenance (that component was downloaded from)
- Component's License: This has wide-spread usefulness to downhill consumers of SBOMs
- Description: This will help remove ambiguity that makes CPE less useful
- Author of the Component: If different from value in "Author of SBOM Data" data field
- Metadata describing the purpose of the SBOM (top of tree)

Additionally, we recommend that the standard define the minimum level depth for an SBOM (i.e. 1 level for completely original code pieces):

- SBOMs must provide for external references (to other SBOMs)
- The SBOM Builder should provide hash tree for all components of their Product (and SBOM Standard should provide for this). This allows verification of provenance for all Components listed in the SBOM

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

An SBOM standard should provide for naming of Aggregations [collections] of common components for known and familiar Security, Efficiency, and Interoperability. There are very few standards in this area, but the industry would benefit from standards such as: LAMP (Linux, Apache, MySQL, php),

official Containers, (Components in an OS instance that is FIPS Validated). The SBOM standard should publish an input format for private users to suggest/rate such Aggregations.

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.*

a. Software Identity: HPE strongly encourages NIST to work with industry to improve the CPE standard for SW Identity. We recommend developing more predictable rules for creating new names in a reproducible manner and self-registration for Product names in CPE. An alternative approach would be for software developers to provide (in the SBOM) the hash for each Component: as the most unique and reproducible identifier for a Component version (instance) (must publish which hash algorithm). The standard should support multiple formats for SW Identity in a single SBOM, including NIST CPE and non-CPE Identifiers that are self-identifying. This naming standard, reconciled with CPE, will support ease of analysis between SBOM components and CVEs.

b. Software-as-a-Service and online services: The SBOM should list APIs available in cloud-based and online services, and versions of APIs. These APIs should be tracked as if they are SW Components. Off-Premise software should be required to adhere to the same SBOM discovery and publication processes and standards as On-Premise solutions. The customer has no responsibility for maintenance, but still deserves the same secure practices through transparency of SBOM and SBOM practices.

c. Legacy and binary-only software: The SBOM standard should also recommend retiring legacy software that is binary-only and has no obtainable source code.

d. Integrity and authenticity: The standard should document which repositories automatically check for digital signatures or hashes from known/good sources, and create a standard which all other repositories/ component managers can follow. Additionally, it should:

- Require all packages to be validated for source and lack of tampering
- Store the Hash with the record of component in SBOM
- State whether the Component was digitally signed, and if the creator of SBOM validated the digital signature

e. Threat model: A standard should be developed for including SBOM creation during a Product's build process, certify build process is on managed environment running on locked-down configuration, and sign each build with the SBOM included (build environment is managed, and has its own signed SBOM).

f. High assurance use cases: The certificate used by SBOM, if not part of PKI, must make the public key available publicly for verification. At a minimum, it should attest to the standards to which the Dev/ Build environment adheres to assure the Build Environment, and Announce and Validate that the SBOM for the Dev/ Build environment is filed and available for forensic purposes/future audits.

g. Delivery: The SBOM should be delivered as part of package/ product instead of delivering in an SBOM library, etc. For large, recursive SBOMs, there will be difficulties navigating it. When an SBOM is delivered, it needs to provide a way to view/navigate the SBOM and what it contains. Naming a Standard Format would be better than providing a viewer tool.

h. Depth: Standards should be developed for components to publish and certify their SBOMs. When more Components adhere to standards for making their SBOM available, then full graphs of products' SBOMs will be easier to complete. Standards should also be developed for Code Repositories to offer this for the common packages they include/ manage/ pull into builds.

i. Vulnerabilities: SBOMs are static, and cannot have up-to-date data. An external (live, self-updating) database of vulnerabilities must be integrated with, or used with SBOMs.

Exceptions can be documented: if an impacted Component is used, there can be an explanation of why the product in question isn't impacted. Environment details and Build Editions can help shrink the number of false flags for Impacted SW, and decrease the need for exception documentation.

An SBOM may contain Vendor Security Impact that makes statements about vulnerabilities known at the time of SBOM creation, concerning the impact and criticality of those vulnerabilities.

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?*

Minimum requirement should be a simple SBOM that includes only Vendor, Component name, Version data fields for each component, so that minimum compliance is not burdensome.

HPE would be pleased to work with NTIA and NIST on the above recommendations and to help advance the Administration's goal of enhancing software supply chain security.

Sincerely,

CJ Coppersmith
Director, Product Cybersecurity and Compliance