CROWDSTRIKE

REQUEST FOR COMMENT RESPONSE

Software Bill of Materials Elements and Considerations.

17 June 2021

I. INTRODUCTION

In response to the National Telecommunication and Information Administration's ("NTIA") request for comment on the Software Bill of Materials ("SBOM") Elements and Considerations, CrowdStrike offers the following views.

CrowdStrike approaches this public consultation from the standpoint of a leading international, US-headquartered cloud-native cybersecurity provider that defends enterprises, including critical infrastructure entities, from globally-distributed threats such as intellectual property theft, financially-motivated crime, destructive attacks, and data breaches.

II. COMMENTS

Executive Order 14028 ("EO") requires NTIA to produce the minimum elements for an SBOM,[1] which is defined as "a formal record containing the details and supply chain relationships of various components used in building software."[2] CrowdStrike's response will be guided by this definition, as well as by the questions posed by NTIA. First, we make a few general observations and comments:

A. EO 14028 includes a number and array of cybersecurity projects and initiatives that stand to enhance national cybersecurity posture, depending on implementation and resourcing. We note and appreciate the extent to which the SBOM initiative and others mandated by the EO will consult with industry, which will be critical to achieving its stated goals.[3]

B. Adversaries regularly target software supply chain information. Exposure of full software component information could provide a blueprint for adversaries to further target and exploit software supply chains by advertising attack vectors or by increasing attack surface through the addition of SBOM infrastructure. We think more data on how to weigh this risk against potential security gains from the SBOM

[1] EO 14028 § 4(f).
[2] EO 14028 § 10(j).
[3] https://www.crowdstrike.com/blog/what-the-new-cybersecurity-executive-order-means-for-public-sector/.

initiative would help clarify what level of effort, and urgency, the software development community should attach to SBOM programs.

C. Some other elements of the EO can support--but from a sequencing standpoint, should likely precede--SBOM adoption. For example, to the extent that government entities will come to possess sensitive information through SBOM data calls, planning, or implementation, adoption of the EO's enterprise and endpoint security, log management, and federal breach reporting provisions could enhance trust among contributing stakeholders.

D. Policymakers should not become overly deterministic about the SBOM initiative. The intent should not be to create SBOMs for their own sake, but rather to strengthen cybersecurity by enhancing software supply chain security. To this end, if there are other ways to drive particular cybersecurity outcomes that are more straightforward, less costly, or otherwise more efficient, those should be used and prioritized as appropriate.

E. We note that the description in the RFC document cites SBOM progress and proofs of concept within certain sectors, e.g., the medical technology supply chain. We believe that the operation of an SBOM program in such a regulated sector would help demonstrate viability; demonstrate benefits; and clarify participant responsibilities that would promote broader adoption across the IT ecosystem.

F. We should be clear-eyed about the impacts the SBOM initiative may have on real or perceived incentives within the software development ecosystem. For example, to the extent that complying with an SBOM reporting scheme imposes a cost or barrier to a provider, that naturally may encourage them to avoid triggering a potential obligation. This may result in some developers favoring 'homebrew'-type solutions for common software functions for which more robust options already exist within, for example, the open source community. While this may lower the risk of single vulnerabilities that impact large portions of the ecosystem, it may not strengthen cybersecurity across the ecosystem overall. (For more on incentives, see Section II(2)(b), below.)

G. It is worth considering how emerging security concepts like Zero Trust and new capabilities in the Identity space will alter defenders' sense of risk or dependency in the future, and other environmental changes since the SBOM concept began to gain currency within the cybersecurity community some years ago.

Responses to individual questions follow.

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 note the use of the “minimum elements” concept in the RFC. Along those lines, we encourage the use of a ‘minimum viable product’-type approach for early SBOMs. The “depth” category, describing the use of SBOMs for graphed dependencies, is ambitious for a concept with this level of maturity.

Another *operational consideration* worth weighing is a criticality threshold. For example, limitation of disclosure to only components that reach a certain threshold could both ease SBOM participation and ensure that data captured within the SBOM is germane for users. Thresholds could be adjusted over time based on provider and user feedback, incidents detected or undetected, and so on.

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

To the extent that the SBOM initiative creates positive externalities, e.g., with respect to licence management, those should be noted by users. But to maintain a manageable scope and promote trust among the user community, the core focus of the program should not broaden beyond explicit cybersecurity purposes.

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: 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.

One factor that may lead to conformity in software identification for SBOM reporting is reducing the noise of seemingly-endless software components to focus on *signal*. As described in responses that follow, it is important to institute a criticality threshold to focus SBOM reporting on software that, if vulnerable, could materially affect security of the relevant supply chain.

b. 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.

Modern software, and in particular cloud-based SaaS solutions, is much more likely to use a dynamic rather than static list of components. Notably:

- Components can number in the thousands. This number depends on the complexity of the offering and how the community comes to define 'components.'
- Risk may vary significantly across components. Many may be accessible only via local access or via network enclaves. Others may have several compensating security controls in place.

These factors complicate both the practical ability to maintain an up-to-date list of each software component and the efficacy of using such information, thereby creating a *signal* vs. *noise* problem. Accordingly, incorporation of a criticality threshold into SBOM requirements may better achieve the objective of enhancing supply chain security.

We believe initial SBOM efforts should focus on edge services that do or can make a connection to the Internet. Such criteria could be useful for identifying the most impactful software with vulnerabilities in a pragmatic manner. Without a criticality threshold, factoring in materiality and compensating security controls, there may be a market barrier of entry that favors legacy, static, or incumbent technologies in lieu of the cutting edge cloud technologies that the government currently seeks.

c. 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.

In general, we believe effort expended reducing reliance on legacy software and arcitures; migrating to cloud-based solutions where possible; and generally modernizing IT infrastructure will yield greater security enhancements than effort bringing legacy software into conformance with emerging SBOM requirements.

d. 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.

SBOM vetting is an important matter to consider. For example, in the government case, the FedRAMP program currently provides robust vetting of federal cloud vendors but can take years for completion. A tailored approach to SBOM can be helpful in both vetting government vendors to begin with as well as responding to vulnerabilities once they become known. (For additional feedback on integrity, see response to 2(e), below.)

e. 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?

The set of concerns raised here underscore the need to appropriately design SBOM scope. The creation of additional attack surface, which may be exploited, or complexity, which may conceal risks, could undermine security gains from the SBOM initiative. Integrity-based attacks on an SBOM may also lead to worse rather than better security outcomes. Avoiding these potential pitfalls will require a careful approach to requirements for what information should be collected and represented on SBOMs; clear guidelines for who ought to control that information; reasonable bounds on how and to whom that information can be displayed; and purposeful guidance from the SBOM advocates for how the information ought to be used.

f. 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?

As noted in Section II(C), above, sequencing EO efforts is important. It may be useful for NIST to determine and publish practices pursuant to EO Sec. 4(e)(i). This may mean integration of build information cannot be included in the initial minimum SBOM elements.

g. 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 noted in Section II(E), above, sustainable SBOM adoption within a particular community will help clarify a number of potential operational issues. It will further help illustrate return on investment for ecosystem participants.

h. 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.

(Refer to Section II(3)(b) for the rationale for considering thresholds rather than the aiming to produce an entire graph, and Section II(4) for best practices for accomodations.)

i. 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.

With respect to vulnerabilities, many defenders clearly still struggle to address application-level issues, and component-level vulnerabilities are considerably more complicated. That said, the application-level is probably the most relevant layer of analysis from defenders' perspective. If a piece of software includes a vulnerable component that is not mitigated in some way, then the application itself is vulnerable. The application provider may be able to communicate information about how to mitigate the issue (e.g., disabling a certain feature of the software or otherwise limiting functionality), but making those determinations at the individual user level, even for enterprises, seems onerous and unlikely to succeed.

In the overwhelming majority of cases, CrowdStrike views real-time data as the gold standard for security practices and use cases. This informs concepts we advocate for like the '1-10-60 Rule.'[4] To the extent that SBOM implementation entails a tradeoff between more comprehensive, but more dated information, versus more modest, but more up-to-date information, we believe that the latter approach would drive better security outcomes.

j. 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"), but other solutions may exist.

Context is important, and a list of software components does not necessarily indicate to what extent vulnerabilities may actually be operationalized. Product architectures and mitigating controls may be dispositive as to whether or not a vulnerability may be exploited by an adversary.

[4] https://www.crowdstrike.com/resources/crowdcasts/the-1-10-60-minute-challenge-a-framework-for-stopping-breaches-faster/.

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?

Advocates should position SBOM as something that can help and demonstrate that it actually strengthens security, and thereby drive voluntary adoption. This can be done along the lines referenced in Section II(E), above. Accommodations are more relevant for regulatory approaches.

To the extent SBOMs become required within particular communities, the best practice would be to focus on the aforementioned criticality threshold and additionally allow for flexibility in domains where it may not be practical to remove a particular software dependency, such as in legacy critical infrastructure systems. This could include permitting those dependent upon certain software to demonstrate, such as through an attestation, both a continued need to use the software and the presence of mitigating controls.

III. CONCLUSION

We commend NTIA for its thoughtful and comprehensive approach to an issue with such complex policy and legal implications on its imposed timeline. The ultimate goal of improving software supply chain security will require a multifaceted approach, including leveraging the very technologies and security principles highlighted in the EO. In light of recent and ongoing threats, identifying and mitigating software security vulnerabilities is a key part of achieving success.

IV. ABOUT CROWDSTRIKE

CrowdStrike® Inc. (Nasdaq: CRWD), a global cybersecurity leader, is redefining security for the cloud era with an endpoint protection platform built from the ground up to stop breaches. The CrowdStrike Falcon® platform's single lightweight-agent architecture leverages cloud-scale AI and offers real-time protection and visibility across the enterprise, preventing attacks on endpoints on or off the network. Powered by the proprietary CrowdStrike Threat Graph®, CrowdStrike Falcon correlates over 3 trillion endpoint-related events 4 per week in real time from across the globe, fueling one of the world's most advanced data platforms for security.

With CrowdStrike, customers benefit from better protection, better performance and immediate time-to-value delivered by the cloud-native Falcon platform.

There's only one thing to remember about CrowdStrike: We stop breaches. Learn more: https://www.crowdstrike.com/.

V. CONTACT

We would welcome the opportunity to discuss these matters in more detail. Public policy inquiries should be made to:

Drew Bagley CIPP/E
VP & Counsel, Privacy and Cyber Policy

Robert Sheldon
Director, Public Policy & Strategy

Email: policy@crowdstrike.com

###