Enterprise Cloud Coalition ECC

November 4, 2021

National Institute of Standards and Technology
Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930)
Gaithersburg, MD 20899-8930
Email: ssdf@nist.gov

RE: ECC's Response to NIST's Request for Comments on the Secure Software Development Framework (SSDF) v1.1

Dear NIST CSD Team:

The Enterprise Cloud Coalition (ECC) appreciates the opportunity to submit comments to the National Institute of Standards and Technology (NIST) Computer Security Division (CSD) team on the draft version of Special Publication (SP) 800-218, *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating Risk of Software Vulnerabilities*.[1]

ECC is comprised of enterprise-focused and cloud-based U.S. companies with shared business models based on processing data on behalf of other companies without monetizing user data for advertising. Taking advantage of the cloud-native features and functionality available through our solutions, ECC companies are directly involved in building and advancing technology that promotes innovation while providing ethical and consumer protection-related benefits to enterprises and their customers or users. As a result, ECC promotes those policies and standards that emphasize the privacy and security of data and information.

With the SSDF representing one of the requirements in Executive Order 14028, *Improving the Nation's Cybersecurity* (Cyber EO), that has the potential to impact the software supply chain beyond the federal ecosystem, ECC commends NIST for seeking feedback from members of industry and the public.

This response addresses both the explicit questions asked by NIST as well as additional considerations that should be helpful in finalizing SP 800-218 and any other efforts related to better securing the development of software and the associated software development life cycle (SDLC).

General Comments and Feedback

Explicitly adopting a risk-based approach will lead to wider adoption.

ECC appreciates the numerous explicit explicit mentions made by NIST to account for and consider risk when determining how and to what degree the practices will be implemented. As was mentioned in the introduction, "each software producer may have unique security assumptions, and each software consumer may have unique security needs and requirements."[2] These unique circumstances can result in the organizations using different tasks or implementation examples while still meeting the ultimate objective of the practice itself. Recognizing and emphasizing that fact in the

[1] National Institute of Standards and Technology (2021) Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. (U.S. Department of Commerce, Washington, D.C.), NIST Special Publication (SP) 800-218, Draft, 30 September 2021. https://doi.org/10.6028/NIST.SP.800-218-draft
[2] *Id.* page 3, lines 313-15.

final draft and future work is important to not only encourage faster and wider adoption of the SSDF considering the substantial but also to ensuring that the guidance does not end up becoming a prescriptive checklist.

Maintaining a tool- and technology-agnostic focus is essential.

With the variety of approaches taken by ECC companies in developing their solutions, these collective experiences are an authoritative example for how important a tool- and technology-agnostic focus will be to developing, maintaining, and enhancing a secure and resilient software supply chain. As NIST pointed out, there are several different SDLC models,[3] many with unique products and techniques. With the rapid pace of development in cloud-based services partly attributable to the similarly fast pace in developing new approaches to SDLC models, updates to the SSDF should be informed by consistent collaboration and outreach with industry to ensure adequate exposure to these advances. This will be essential to ensure the final draft and future work does not result in recommendations that would dissuade developers from using otherwise safe approaches simply due to their novelty in comparison to the provided tasks or implementation examples.

"Shifting left" will require a baseline of measurable practices.

ECC companies have found great success when SDLC-associated requirements are based on achievable objectives that are measurable. Far from being prescriptive, a measurable objective simply means having the ability to know the goal has been accomplished. By including enough specificity in the descriptions of the practices to allow for the development of metrics, organizations can validate compliance through their own tasks or implementation examples. This can be especially helpful to those involved with non-technical roles, particularly those involved in the procurement and third-party contract management processes, when attempting to use language from the SSDF for their own uses. Two practices that could use some more clarity are:

- PO.5 – What is an acceptable metric that can be used to measure how "strongly protected" the software development environment is and will be after updates?
- PW.1 – Are there goals that should be achieved other than ensuring software complies with the SSDF-based practices of the organization?

A similar concept should be applied to tasks and implementation examples that are provided in the final draft and future work, where appropriate. However, insisting on the measurability of current and future practices will help to ensure the SSDF can be applied as broadly as possible.

Mandating an owner of all code used will be necessary for success.

Ensuring any code used in the released software should have an identified owner who can answer questions, make updates, and otherwise provide necessary information is a necessary function of a SSDF. Whether as part of the work required to Prepare the Organization (PO) or otherwise, this individual or group should have in place succession planning, absence coverage, documentation responsibilities, as well as monitoring for and incorporating into the code base they own any publicly available security patches. While this will be important for internally developed code, it will be especially useful for code obtained from third parties. Similarly, despite the advantages and benefits to society brought by open source code, some degree of formalized secure software development practices in that component must be verified before its use in released software, especially where it serves a critical or essential function.

Answers to Specific Questions

Do the SSDF practices, tasks, and implementation examples fit well into your current software development practices? Are there any conflicts or gaps that the SSDF should address?

Yes, the current practices, tasks, and implementation examples fit well within the current software development practices used by ECC companies.

[3] *Id.* page 1, lines 242-45.

While there are no conflicts with the practices, there are unique ways in which some companies are implementing them that verifiably meet or exceed the objectives despite not being mentioned in the examples provided. It is partly for this reason that the general feedback emphasized the adherence to a tool- and technology-agnostic focus, based on a risk-based approach, using a measurable baseline of secure practices.

As for gaps, beyond the one mentioned above in terms of mandating an owner of any code that makes up part of the final release, there is also a need for future work in this space to address the shared responsibility model that is inherent in modern cloud-based and cloud-native software, especially when obtained through the “as-a-Service” delivery model. Whether that guidance to the user of the software can be enhanced through updated SSDF practices is dependent on the guidance ultimately produced. ECC companies have the experience necessary to contribute to such future work and look forward to providing our expertise to NIST and other stakeholders working to address this problem.

Should the SSDF practices and tasks involving software integration, building, and delivery be split so that integration is separate from building and delivery?

No, integration should not be separated from building and delivery. By maintaining the three together, the SSDF will remain a high-level, generally applicable approach. Beginning to carve out and diverge from the more unified approach increases the risk of conflict and loss of utility. This will be especially exacerbated if non-technical influences use the separated SSDF to create artificial phases or organizational silos. With the SSDF targeted at a broad audience as well as a broad number of SDLC models and developer environments, maintaining a unified approach will be important.

What types of artifacts and evidence can be captured, documented, and shared publicly as byproducts of implementing the secure software development practices? Are there examples you can share?

While there are several artifacts and evidence that can be provided to verify certain practices (e.g., the results and coverage of static and automated testing, the environment or tech stack used during development, etc.), they will differ and be dependent on the metric being validated. Beyond reiterating the need for measurable practices as was recommended in the General Feedback provided above, there must be an understanding of the value of artifacts being sought. This is especially true if it is ever required to share this information publicly, as publication can create a supply chain vulnerability simply through exposure.

Establishing the criticality of the information as part of the validation process as well as the cost associated with generating and sharing those records will be important to consider. In keeping with the recommendations associated with using automated process, the use of machine-readable formats should also be considered, especially for their ability to allow for validation without over-sharing. This has the added benefit of providing the capability for multiple methods of verification, validating verification through redundancy.

***

Again, ECC thanks NIST for the opportunity to provide feedback on SP 800-218 and contributing to NIST’s important work in improving the security of our nation’s software. We welcome questions on our feedback and look forward to continuing to be a part of this discussion as it develops.

Most Sincerely,

Omid Ghaffari-Tabrizi
Enterprise Cloud Coalition
omid@enterprisecloudcoalition.org
https://www.EnterpriseCloudCoalition.org/