The NIST Secure Software Development Framework (SSDF)

Karen Scarfone | Scarfone Cybersecurity

Approach Similar to the Cybersecurity Framework

Provides a common language to describe fundamental, sound secure software development practices

Can help an organization document its secure software development practices today and define its future target practices as part of its continuous improvement process

Leverages existing secure software development practices from established standards, guidance, and secure software development practice documents

Do no harm (to organizations who have already adopted established practices)

SSDF Publication Basics

For both *software producers* (e.g., COTS vendors, government software developers, custom software developers) and *software consumers* (federal government agencies and other organizations)

Can be used by organizations in any sector or community, regardless of size or cybersecurity sophistication to manage risks associated with software

Can be applied to software developed to support IT, ICS, IoT, or cyber-physical systems (CPS)

Can be integrated into any existing software development workflow and automated toolchain

Broadly applicable—not specific to technologies, platforms, programming languages, SDLC models, development environments, operating environments, tools, etc.

SSDF Practice Groups

National Institute of Standards and Technology
U.S. Department of Commerce

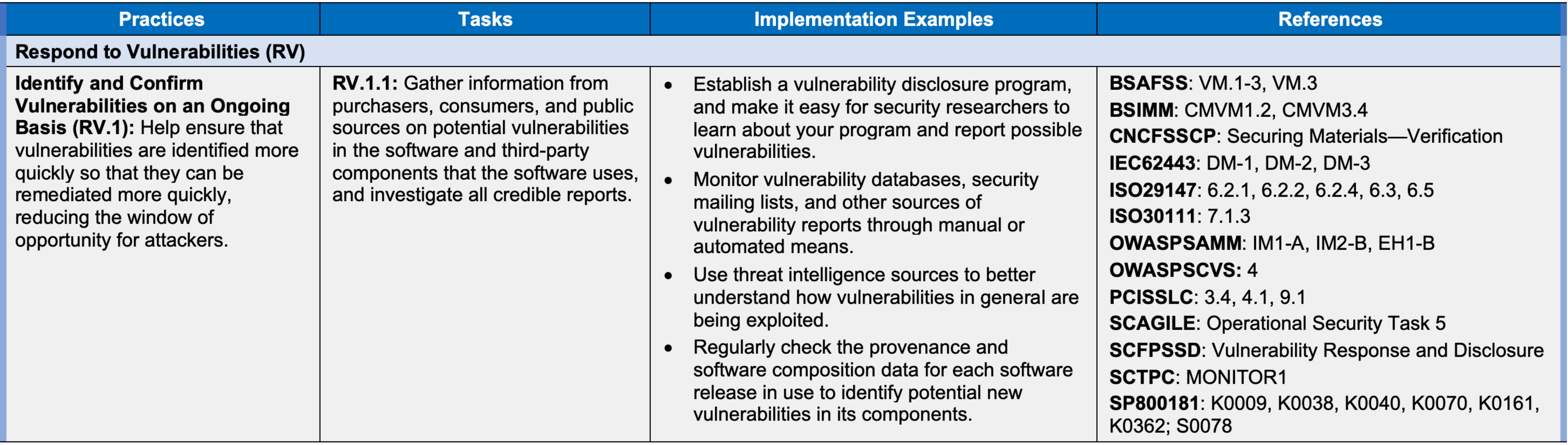
Prepare the Organization (PO): Ensure the organization's people, processes, and technology are prepared to perform secure software development at the organization level and, in some cases, also for each individual project.

Protect the Software (PS): Protect all components of the software from tampering and unauthorized access.

Produce Well-Secured Software (PW): Produce well-secured software that has minimal security vulnerabilities in its releases.

Respond to Vulnerabilities (RV): Identify vulnerabilities in software releases and respond appropriately to address those vulnerabilities and prevent similar vulnerabilities from occurring in the future.

Elements of an SSDF Practice



Practices	Tasks	Implementation Examples	References
Respond to Vulnerabilities (RV)			
Identify and Confirm Vulnerabilities on an Ongoing Basis (RV.1): Help ensure that vulnerabilities are identified more quickly so that they can be remediated more quickly, reducing the window of opportunity for attackers.	**RV.1.1:** Gather information from purchasers, consumers, and public sources on potential vulnerabilities in the software and third-party components that the software uses, and investigate all credible reports.	• Establish a vulnerability disclosure program, and make it easy for security researchers to learn about your program and report possible vulnerabilities. • Monitor vulnerability databases, security mailing lists, and other sources of vulnerability reports through manual or automated means. • Use threat intelligence sources to better understand how vulnerabilities in general are being exploited. • Regularly check the provenance and software composition data for each software release in use to identify potential new vulnerabilities in its components.	**BSAFSS**: VM.1-3, VM.3 **BSIMM**: CMVM1.2, CMVM3.4 **CNCFSSCP**: Securing Materials—Verification **IEC62443**: DM-1, DM-2, DM-3 **ISO29147**: 6.2.1, 6.2.2, 6.2.4, 6.3, 6.5 **ISO30111**: 7.1.3 **OWASPSAMM**: IM1-A, IM2-B, EH1-B **OWASPSCVS:** 4 **PCISSLC**: 3.4, 4.1, 9.1 **SCAGILE**: Operational Security Task 5 **SCFPSSD**: Vulnerability Response and Disclosure **SCTPC**: MONITOR1 **SP800181**: K0009, K0038, K0040, K0070, K0161, K0362; S0078

Task: An individual action (or actions) needed to accomplish a practice

Implementation Example: An example of a type of tool, process, or other method that could be used to implement this task

Reference: An established secure development practice document and its mappings to this task

Draft SP 800-218, SSDF V1.1

- Replaces original SSDF paper from April 2020
- Added references based on public input, including
  - IEC 62443 (ICS/OT)
  - OWASP MASVS (mobile)
  - OWASP SCVS and CNCF SSCP (supply chain)
  - ISO/IEC 29147 and 30111 (vuln disclosure)
- Made other changes to address EO 14028 and the latest threats
- Created Appendix A to map EO clauses to the SSDF practices and tasks that help address each clause

Addressing 4(e) with SSDF V1.1

4(e) directs NIST to "issue guidance identifying practices that enhance the security of the software supply chain" for:

- (vi) maintaining provenance of software code or components, and controls on software components, tools, and services for software development processes
  - Added PS.3.2: *Collect, maintain, and share provenance data for all components and other dependencies of each software release (e.g., software composition in a software bill of materials [SBOM]).*
  - PO.1.3, PO.5.1, PS.3.1, PW.4.1, PW.4.5, RV.1.2
- (vii) providing an SBOM for each product
  - Added PS.3.2
- (viii) participating in a vulnerability disclosure program
  - RV.1.3: *Have a policy that addresses vulnerability disclosure and remediation, and implement the roles, responsibilities, and processes needed to support that policy.*
  - RV.1.1, RV.1.2, RV.2.1, RV.2.2, RV.3.3

Addressing 4(e) with SSDF V1.1

- (ii) providing artifacts that demonstrate conformance to (i) processes
- (i) securing software development environments

  - Added PO.5: *Implement and Maintain Secure Environments for Software Development*
  - Added PO.5.1: *Separate and protect each environment involved in software development.*
  - Added PO.5.2*: Secure and harden development endpoints (i.e., endpoints for software designers, developers, testers, builders, etc.) to perform development-related tasks using a risk-based approach.*
  - PO.3.2: *Follow recommended security practices to deploy and maintain tools and toolchains.*
  - PO.3.3: *Configure tools to generate evidence and artifacts of their support of secure software development practices as defined by the organization.*

Addressing 4(e) with SSDF V1.1

- (v) providing artifacts of tool and process execution for (iii) and (iv), and making available summary information publicly available
- (iii) maintaining trusted source code supply chains
- (iv) checking for known and potential vulnerabilities and remediating them

  - PO.3.3: *Configure tools to generate evidence and artifacts of their support of secure software development practices as defined by the organization.*
  - PS.1.1: *Store all forms of code, including source code and executable code, based on the principle of least privilege so that only authorized personnel, tools, services, etc. have the necessary forms of access.*
  - PS.2.1: *Make integrity verification information available to software purchasers and consumers.*
  - PW.4.5: *Verify the integrity and check the provenance of all in-house and third-party software components before reusing them for the organization's own software.*

Addressing 4(e) with SSDF V1.1

- (v) providing artifacts of tool and process execution for (iii) and (iv), and making available summary information publicly available
- (iii) maintaining trusted source code supply chains
- (iv) checking for known and potential vulnerabilities and remediating them

  - Numerous PW tasks on finding and addressing vulnerabilities before release
  - RV.1.1, *Gather information from purchasers, consumers, and public sources on potential vulnerabilities in the software and third-party components that the software uses, and investigate all credible reports.*
  - RV.1.2, *Review, analyze, and/or test the software's code to identify or confirm the presence of previously undetected vulnerabilities.*
  - RV.2.2, *Develop and implement a remediation plan for each vulnerability.*
  - RV.3.3, *Review the software for similar vulnerabilities, and proactively fix them rather than waiting for external reports.*

Addressing 4(e) with SSDF V1.1

4(e) also directs NIST to "issue guidance identifying practices that enhance the security of the software supply chain" for:

- (ix) attesting to conformity with secure software development practices
  - All practices and tasks that are applicable using a risk-based approach
- (x) ensuring and attesting to the integrity and provenance of open source software used within a product
  - Added PS.3.2: *Collect, maintain, and share provenance data for all components and other dependencies of each software release (e.g., in an SBOM).*
  - PW.4.5: *Verify the integrity and check the provenance of all in-house and third-party software components before reusing them for the organization's own software.*
  - PO.1.3, PW.4.1, PW.4.4

Public Comment Examples

Reflects comments received as of Thursday, November 4

- Clarify the meaning/scope of certain terms, like "build environment" and "provenance"
- Increase automation, especially for the Prepare the Organization (PO) practices and tasks
- Address the effect of resource limitations on performing practices and tasks at scale
- Add prioritization guidance
- Clarify what "artifacts" and "evidence" involve
- Add practice or task on security guides for software administrators or users

Help Us Improve the SSDF & Address the EO

https://csrc.nist.gov/projects/ssdf
&
ssdf@nist.gov

- Submit your comments on Draft SP 800-218 as soon as you can.
- If your organization has produced a set of secure software development practices and you want to map them to the SSDF, please email us so we can introduce you to the National Online Informative References (OLIR) Program. You can contribute your mapping to our collection of informative references.
- We want your thoughts about what types of artifacts can be captured, documented, and shared publicly as byproducts of implementing the SSDF practices. Share examples with us!