NIST Special Publication 800-218 Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities

Executive Order on Improving the Nation’s Cybersecurity

Issued May 12, 2021 by President Biden

One Key Areas Covered by this Executive Order: Enhancing Software Supply Chain Security

- Includes the requirement that NIST shall develop guidance that shall include standards, procedures, or criteria regarding:

  Secure software development environments, including such actions as:

  - using administratively separate build environments;
  - auditing trust relationships;
  - establishing multi-factor, risk-based authentication and conditional access across the enterprise;
  - documenting and minimizing dependencies on enterprise products that are part of the environments used to develop, build, and edit software;
  - employing encryption for data; and
  - monitoring operations and alerts and responding to attempted and actual cyber incidents
- To respond to this requirement NIST created NIST Special Publication 800-218 Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities

NIST SP800-218 Secure Software Development Framework (SSDF)

- Describes a set of high-level practices based on established standards, guidance, and secure software development practice documents. These practices, collectively called the Secure Software Development Framework (SSDF)
- Focus is on the outcomes of the practices rather than on the tools, techniques, and mechanisms to do so
- Defines only a high-level subset of what organizations may need to do, so organizations should consult the references and other resources for additional information on implementing the practices
  - Identifies secure software development practices but does not prescribe how to implement them
- Not all practices are applicable to all use cases; organizations should adopt a risk-based approach to determine what practices are relevant, appropriate, and effective to mitigate the threats to their software development practices

NIST SP800-218 Secure Software Development Framework (SSDF)

- Recommended practices are based on established secure software development practice documents
- Practices are organized into four groups:
  - **Prepare the Organization (PO):** Organizations should ensure that their people, processes, and technology are prepared to perform secure software development at the organization level.
  - **Protect the Software (PS):** Organizations should protect all components of their software from tampering and unauthorized access.
  - **Produce Well-Secured Software (PW):** Organizations should produce well-secured software with minimal security vulnerabilities in its releases.
  - **Respond to Vulnerabilities (RV):** Organizations should identify residual vulnerabilities in their software releases and respond appropriately to address those vulnerabilities and prevent similar ones from occurring in the future.

NIST SP800-218 Secure Software Development Framework (SSDF)

- Each practice definition includes the following elements:
  - **Practice:** The name of the practice and a unique identifier, followed by a brief explanation of what the practice is and why it is beneficial
  - **Tasks:** One or more actions that may be needed to perform a practice
  - **Notional Implementation Examples:** One or more notional examples of types of tools, processes, or other methods that could be used to help implement a task. No examples or combination of examples are required, and the stated examples are not the only feasible options. Some examples may not be applicable to certain organizations and situations.
  - **References:** Pointers to one or more established secure development practice documents and their mappings to a particular task. Not all references will apply to all instances of software development.

NIST SP800-218 Secure Software Development Framework (SSDF)

Prepare the Organization (PO) Practices:

- **Define Security Requirements for Software Development (PO.1)**: Ensure that security requirements for software development are known at all times so that they can be taken into account throughout the SDLC and duplication of effort can be minimized because the requirements information can be collected once and shared. This includes requirements from internal sources (e.g., the organization’s policies, business objectives, and risk management strategy) and external sources (e.g., applicable laws and regulations)
- **Implement Roles and Responsibilities (PO.2)**: Ensure that everyone inside and outside of the organization involved in the SDLC is prepared to perform their SDLC-related roles and responsibilities throughout the SDLC

NIST SP800-218 Secure Software Development Framework (SSDF)

Prepare the Organization (PO) Practices:

- **Implement Supporting Toolchains (PO.3)**: Use automation to reduce human effort and improve the accuracy, reproducibility, usability, and comprehensiveness of security practices throughout the SDLC, as well as provide a way to document and demonstrate the use of these practices. Toolchains and tools may be used at different levels of the organization, such as organization-wide or project-specific, and may address a particular part of the SDLC, like a build pipeline
- **Define and Use Criteria for Software Security Checks (PO.4)**: Help ensure that the software resulting from the SDLC meets the organization’s expectations by defining and using criteria for checking the software’s security during development
- **Implement and Maintain Secure Environments for Software Development (PO.5)**: Ensure that all components of the environments for software development are strongly protected from internal and external threats to prevent compromises of the environments or the software being developed or maintained within them. Examples of environments for software development include development, build, test, and distribution environments

NIST SP800-218 Secure Software Development Framework (SSDF)

Protect Software (PS) Practices:

- **Protect All Forms of Code from Unauthorized Access and Tampering (PS.1)**: Help prevent unauthorized changes to code, both inadvertent and intentional, which could circumvent or negate the intended security characteristics of the software. For code that is not intended to be publicly accessible, this helps prevent theft of the software and may make it more difficult or time-consuming for attackers to find vulnerabilities in the software
- **Provide a Mechanism for Verifying Software Release Integrity (PS.2)**: Help software acquirers ensure that the software they acquire is legitimate and has not been tampered with
- **Archive and Protect Each Software Release (PS.3)**: Preserve software releases in order to help identify, analyze, and eliminate vulnerabilities discovered in the software after release

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW) Practices:

- **Design Software to Meet Security Requirements and Mitigate Security Risks (PW.1)**: Identify and evaluate the security requirements for the software; determine what security risks the software is likely to face during operation and how the software's design and architecture should mitigate those risks; and justify any cases where risk-based analysis indicates that security requirements should be relaxed or waived. Addressing security requirements and risks during software design (secure by design) is key for improving software security and also helps improve development efficiency
- **Review the Software Design to Verify Compliance with Security Requirements and Risk Information (PW.2)**: Help ensure that the software will meet the security requirements and satisfactorily address the identified risk information

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW) Practices:

- **Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality (PW.4)**: Lower the costs of software development, expedite software development, and decrease the likelihood of introducing additional security vulnerabilities into the software by reusing software modules and services that have already had their security posture checked. This is particularly important for software that implements security functionality, such as cryptographic modules and protocols
- **Create Source Code by Adhering to Secure Coding Practices (PW.5)**: Decrease the number of security vulnerabilities in the software, and reduce costs by minimizing vulnerabilities introduced during source code creation that meet or exceed organization-defined vulnerability severity criteria
- **Configure the Compilation, Interpreter, and Build Processes to Improve Executable Security (PW.6)**: Decrease the number of security vulnerabilities in the software and reduce costs by eliminating vulnerabilities before testing occurs

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW) Practices:

- **Review and/or Analyze Human-Readable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.7)**: Help identify vulnerabilities so that they can be corrected before the software is released to prevent exploitation. Using automated methods lowers the effort and resources needed to detect vulnerabilities. Human-readable code includes source code, scripts, and any other form of code that an organization deems human readable
- **Test Executable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.8)**: Help identify vulnerabilities so that they can be corrected before the software is released in order to prevent exploitation. Using automated methods lowers the effort and resources needed to detect vulnerabilities and improves traceability and repeatability. Executable code includes binaries, directly executed bytecode and source code,
  and any other form of code that an organization deems executable
- **Configure Software to Have Secure Settings by Default (PW.9)**: Help improve the security of the software at the time of installation to reduce the likelihood of the software being deployed with weak security settings, putting it at greater risk of compromise

Respond to Vulnerabilities (RV) Practices:

- **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 in accordance with risk, reducing the window of opportunity for attackers
- **Assess, Prioritize, and Remediate Vulnerabilities (RV.2)**: Help ensure that vulnerabilities are remediated in accordance with risk to reduce the window of opportunity for attackers
- **Analyze Vulnerabilities to Identify Their Root Causes (RV.3)**: Help reduce the frequency of vulnerabilities in the future

Correspondence Between SSDF Subsections and Cybersecurity Executive Order Subsections

EO 14028 Subsection	SSDF Practices and Tasks
4e(i)(A)	PO.5.1
4e(i)(B)	PO.5.1
4e(i)(C)	PO.5.1, PO.5.2
4e(i)(D)	PO.5.1
4e(i)(E)	PO.5.2
4e(i)(F)	PO.3.2, PO.3.3, PO.5.1, PO.5.2
4e(ii)	PO.3.2, PO.3.3, PO.5.1, PO.5.2
4e(iii)	PO.3.1, PO.3.2, PO.5.1, PO.5.2, PS.1.1, PS.2.1, PS.3.1, PW.4.1, PW.4.4
4e(iv)	PO.4.1, PO.4.2, PS.1.1, PW.2.1, PW.4.4, PW.5.1, PW.6.1, PW.6.2, PW.7.1, PW.7.2, PW.8.2, PW.9.1, PW.9.2, RV.1.1, RV.1.2, RV.2.1, RV.2.2, RV.3.3
4e(v)	PO.3.2, PO.3.3, PO.4.1, PO.4.2, PO.5.1, PO.5.2, PW.1.2, PW.2.1, PW.7.2, PW.8.2, RV.2.2
4e(vi)	PO.1.3, PO.3.2, PO.5.1, PO.5.2, PS.3.1, PS.3.2, PW.4.1, PW.4.4, RV.1.1, RV.1.2
4e(vii)	PS.3.2
4e(viii)	RV.1.1, RV.1.2, RV.1.3, RV.2.1, RV.2.2, RV.3.3
4e(ix)	All practices and tasks consistent with a risk-based approach
4e(x)	PS.2.1, PS.3.1, PS.3.2, PW.4.1, PW.4.4

Subsection 4e is the subsection on Supply Chain Security I mentioned back on Slide 2

NIST SP800-218
Secure Software Development Framework (SSDF)

TASKS

NIST SP800-218 Secure Software Development Framework (SSDF)

Prepare the Organization (PO):

- **Define Security Requirements for Software Development (PO.1):**
  - **PO.1.1**: Identify and document all security requirements for the organization’s software development infrastructures and processes, and maintain the requirements over time
  - **PO.1.2**: Identify and document all security requirements for organization-developed software to meet, and maintain the requirements over time
  - **PO.1.3**: Communicate requirements to all third parties who will provide commercial software components to the organization for reuse by the organization’s own software

NIST SP800-218 Secure Software Development Framework (SSDF)

Prepare the Organization (PO):

- **Implement Roles and Responsibilities (PO.2)**:
  - **PO.2.1**: Create new roles and alter responsibilities for existing roles as needed to encompass all parts of the SDLC. Periodically review and maintain the defined roles and responsibilities, updating them as needed
  - **PO.2.2**: Provide role-based training for all personnel with responsibilities that contribute to secure development. Periodically review personnel proficiency and role-based training, and update the training as needed
  - **PO.2.3**: Obtain upper management or authorizing official commitment to secure development, and convey that commitment to all with development related roles and responsibilities

NIST SP800-218 Secure Software Development Framework (SSDF)

Prepare the Organization (PO):

- **Implement Supporting Toolchains (PO.3)**:
  - **PO.3.1**: Specify which tools or tool types must or should be included in each toolchain to mitigate identified risks, as well as how the toolchain components are to be integrated with each other
  - **PO.3.2**: Follow recommended security practices to deploy, operate, and maintain tools and toolchains
  - **PO.3.3**: Configure tools to generate artifacts of their support of secure software development practices as defined by the organization

NIST SP800-218 Secure Software Development Framework (SSDF)

Prepare the Organization (PO):

- **Define and Use Criteria for Software Security Checks (PO.4)**:
  - **PO.4.1**: Define criteria for software security checks and track throughout the SDLC
  - **PO.4.2**: Implement processes, mechanisms, etc. to gather and safeguard the necessary information in support of the criteria

NIST SP800-218 Secure Software Development Framework (SSDF)

Prepare the Organization (PO):

- **Implement and Maintain Secure Environments for Software Development (PO.5)**:
  - **PO.5.1**: Separate and protect each environment involved in software development
  - **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

NIST SP800-218 Secure Software Development Framework (SSDF)

Protect Software (PS):

- **Define Security Requirements for Software Development (PO.1):**
  - **PS.1.1**: Store all forms of code – including source code, executable code, and configuration-as-code – based on the principle of least privilege so that only authorized personnel, tools, services, etc. have access

NIST SP800-218 Secure Software Development Framework (SSDF)

Protect Software (PS):

- **Provide a Mechanism for Verifying Software Release Integrity (PS.2)**:
  - **PS.2.1**: Make software integrity verification information available to software acquirers

NIST SP800-218 Secure Software Development Framework (SSDF)

Protect Software (PS):

- **Archive and Protect Each Software Release (PS.3)**:
  - **PS.3.1**: Securely archive the necessary files and supporting data (e.g., integrity verification information, provenance data) to be retained for each software release
  - **PS.3.2**: Collect, safeguard, maintain, and share provenance data for all components of each software release (e.g., in a software bill of materials [SBOM])

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW):

- **Design Software to Meet Security Requirements and Mitigate Security Risks (PW.1)**:
  - **PW.1.1**: Use forms of risk modeling – such as threat modeling, attack modeling, or attack surface mapping – to help assess the security risk for the software
  - **PW.1.2**: Track and maintain the software’s security requirements, risks, and design decisions
  - **PW.1.3**: Where appropriate, build in support for using standardized security features and services (e.g., enabling software to integrate with existing log management, identity management, access control, and vulnerability management systems) instead of creating proprietary implementations of security features and services

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW):

- **Review the Software Design to Verify Compliance with Security Requirements and Risk Information (PW.2)**:
  - **PW.2.1**: Have 1) a qualified person (or people) who were not involved with the design and/or 2) automated processes instantiated in the toolchain review the software design to confirm and enforce that it meets all of the security requirements and satisfactorily addresses the identified risk information

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW):

- **Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality (PW.4)**:
  - **PW.4.1**: Acquire and maintain well-secured software components (e.g., software libraries, modules, middleware, frameworks) from commercial, open source, and other third-party developers for use by the organization's software
  - **PW.4.2**: Create and maintain well-secured software components in-house following SDLC processes to meet common internal software development needs that cannot be better met by third-party software components
  - **PW.4.4**: Verify that acquired commercial, open-source, and all other third-party software components comply with the requirements, as defined by the organization, throughout their life cycles

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW):

- **Create Source Code by Adhering to Secure Coding Practices (PW.5)**:
  - **PW.5.1**: Follow all secure coding practices that are appropriate to the development languages and environment to meet the organization’s requirements

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW):

- **Configure the Compilation, Interpreter, and Build Processes to Improve Executable Security (PW.6)**:
  - **PW.6.1**: Use compiler, interpreter, and build tools that offer features to improve executable security
  - **PW.6.2**: Determine which compiler, interpreter, and build tool features should be used and how each should be configured, then implement and use the
    approved configurations

Produce Well-Secured Software (PW):

- **Review and/or Analyze Human-Readable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.7)**:
  - **PW.7.1**: Determine whether code *review* (a person looks directly at the code to find issues) and/or code *analysis* (tools are used to find issues in code, either in a fully automated way or in conjunction with a person) should be used, as defined by the organization
  - **PW.7.2**: Perform the code review and/or code analysis based on the organization's secure coding standards, and record and triage all discovered issues and recommended remediations in the development team's workflow or issue tracking system

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW):

- **Test Executable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.8)**:
  - **PW.8.1**: Determine whether executable code testing should be performed to find vulnerabilities not identified by previous reviews, analysis, or testing and, if so, which types of testing should be used
  - **PW.8.2**: Scope the testing, design the tests, perform the testing, and document the results, including recording and triaging all discovered issues and recommended remediations in the development team's workflow or issue tracking system

NIST SP800-218 Secure Software Development Framework (SSDF)

Produce Well-Secured Software (PW):

- **Configure Software to Have Secure Settings by Default (PW.9)**:
  - **PW.9.1**: Define a secure baseline by determining how to configure each setting that has an effect on security or a security-related setting so that the default settings are secure and do not weaken the security functions provided by the platform, network infrastructure, or services
  - **PW.9.2**: Implement the default settings (or groups of default settings, if applicable), and document each setting for software administrators

NIST SP800-218 Secure Software Development Framework (SSDF)

Respond to Vulnerabilities (RV):

- **Identify and Confirm Vulnerabilities on an Ongoing Basis (RV.1)**:
  - **RV.1.1**: Gather information from software acquirers, users, 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.1.3**: Have a policy that addresses vulnerability disclosure and remediation, and implement the roles, responsibilities, and processes needed to support that policy

NIST SP800-218 Secure Software Development Framework (SSDF)

Respond to Vulnerabilities (RV):

- **Analyze Vulnerabilities to Identify Their Root Causes (RV.3)**:
  - **RV.3.1**: Analyze identified vulnerabilities to determine their root causes
  - **RV.3.2**: Analyze the root causes over time to identify patterns, such as a particular secure coding practice not being followed consistently
  - **RV.3.3**: Review the software for similar vulnerabilities to eradicate a class of vulnerabilities, and proactively fix them rather than waiting for external reports
  - **RV.3.4**: Review the SDLC process, and update it if appropriate to prevent (or reduce the likelihood of) the root cause recurring in updates to the software or in new software that is created