1. Introduction

Section 4.1.a of Executive Order (EO) 14110, *Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence* [1], tasked NIST with "developing a companion resource to the Secure Software Development Framework to incorporate secure development practices for generative AI and for dual-use foundation models." This document is that companion resource.

The software development and use of AI models and AI systems inherit much of the same risk as any other digital system. A unique challenge for this community is the blurring of traditional boundaries between system code and system data, as well as the use of plain human language as the means of interaction with the systems. AI models and systems, their configuration parameters (e.g., model weights), and the data they interact with (e.g., training data, user queries, etc.) can form closed loops that can be manipulated for unintended functionality.

AI model and system development is still much more of an art than an exact science, requiring developers to interact with model code, training data, and other parameters over multiple iterations. Training datasets may be acquired from unknown, untrusted sources. Model weights and other training parameters can be susceptible to malicious tampering. Some models may be complex to the point that they cannot easily be thoroughly inspected, potentially allowing for undetectable execution of arbitrary code. User queries can be crafted to produce undesirable or objectionable output and — if not sanitized properly — can be leveraged for injection-style attacks. The goal of this document is to identify the practices and tasks needed to address these novel risks.

1.1. Purpose

The SSDF provides a common language for describing secure software development practices throughout the software development life cycle. This document augments the practices and tasks defined in SSDF version 1.1 by adding recommendations, considerations, notes, and informative references that are specific to generative AI and dual-use foundation model development. These additions are documented in the form of an *SSDF Community Profile* ("Profile"), which is a baseline of SSDF practices and tasks that have been enhanced to address a particular use case. An example of an addition is, "Secure code storage should include AI models, model weights, pipelines, reward models, and any other AI model elements that need their confidentiality, integrity, and/or availability protected."

This Profile supplements what SSDF version 1.1 already includes. The Profile is intended to be used in conjunction with NIST Special Publication (SP) 800-218, *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities* [6] and should not be used without SP 800-218. Readers should also utilize the implementation examples and informative references defined in SP 800-218 for additional information on how to perform each SSDF practice and task for all types of software development, as they are also generally applicable to AI model and AI system development.

1.2. Scope

This Profile's scope is *AI model development*, which includes data sourcing for, designing, training, fine-tuning, and evaluating AI models, as well as incorporating and integrating AI models into other software. Consistent with SSDF version 1.1 and EO 14110, **practices for the deployment and operation of AI systems with AI models are out of scope**. Similarly, while cybersecurity practices for training data and other forms of data being used for AI model development are in scope, the rest of the data governance and management life cycle is out of scope.

Practices and tasks in this Profile do not distinguish between human-written and AI-generated source code, because it is assumed that all source code should be evaluated for vulnerabilities and other issues before use.

1.3. Sources of Expertise

This document leverages and integrates numerous sources of expertise, including:

- NIST research and publications on trustworthy and responsible AI, including the *Artificial Intelligence Risk Management Framework (AI RMF 1.0)* [2], *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations* [3], *Towards a Standard for Identifying and Managing Bias in Artificial Intelligence* [4], and the Dioptra experimentation testbed for security evaluations of machine learning algorithms [5].
- NIST's *Secure Software Development Framework (SSDF) Version 1.1* [6], which is a set of fundamental, sound, and secure software development practices. It provides a common language to help facilitate communications among stakeholders, including software producers and software acquirers. The SSDF has also been used in support of EO 14028, *Improving the Nation's Cybersecurity* [7], to enhance software supply chain security.
- NIST general cybersecurity resources, including *The NIST Cybersecurity Framework (CSF) 2.0* [8], *Security and Privacy Controls for Information Systems and Organizations* [9], and *Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations* [10].
- AI model developers, AI researchers, AI system developers, and secure software practitioners from industry and government with expertise in the unique security challenges of AI models and the practices for addressing those challenges. This expertise was primarily captured through NIST's January 2024 workshop, where speakers and attendees shared suggestions for adapting secure software development practices and tasks to accommodate the unique aspects of AI model development and the software leveraging them.

1.4. Document Structure

This document is structured as follows:

- Section 2 provides additional background on the SSDF and explains what an SSDF Community Profile is and how it can be used.
- Section 3 defines the SSDF Community Profile for AI Model Development.
- The References section lists all references cited in this document.
- Appendix A provides a glossary of selected terms used within this document.

2. Using the SSDF Community Profile

AI model producers, AI system producers, AI system acquirers, and others can use the SSDF to foster their communications regarding secure AI model development throughout the software development life cycle.[2] Following SSDF practices should help AI model producers reduce the number of vulnerabilities in their AI models, reduce the potential impacts of the exploitation of undetected or unaddressed vulnerabilities, and address the root causes of vulnerabilities to prevent recurrences. AI system producers can use the SSDF's common vocabulary when communicating with AI model producers regarding their security practices for AI model development and when integrating AI models into the software they are developing. AI system acquirers can also use SSDF terms to better communicate their cybersecurity requirements and needs to AI model producers and AI system producers, such as during acquisition processes.

The SSDF Community Profile is not a checklist to follow, but rather a starting point for planning and implementing a risk-based approach to adopting secure software development practices involving AI models. The contents of the Profile are meant to be adapted and customized, as not all practices and tasks are applicable to all use cases. Organizations should adopt a risk-based approach to determine what practices and tasks are relevant, appropriate, and effective to mitigate the threats to software development practices from the organization's perspective as an AI model producer, AI system producer, or AI system acquirer. Factors such as risk, cost, feasibility, and applicability should be considered when deciding which practices and tasks to use and how much time and resources to devote to each one. Cost models may need to be updated to effectively consider the costs inherent to AI model development. A risk-based approach to secure software development may change over time as an organization responds to new or elevated capabilities and risks associated with an AI model or system.

Generative AI and dual-use foundation models present additional challenges in tracking model versioning and lineage. Source code for defining the model architecture and building model binaries is amenable to secure software engineering practices for versioning, lineage, and reproducibility. However, the final model weights are defined only after the model is trained and fine-tuned; this is where limitations in tracking all aspects of collection, processing, and training arise. Organizations should follow secure software development practices for the parts of a model that can be covered fully and strive to introduce secure practices to the extent possible for the stages and corresponding artifacts where obtaining such security guarantees is hard to achieve. Organizations should document the parts and artifacts that are not covered by the secure software development practices.

The Profile's practices, tasks, recommendations, and considerations can be integrated into machine learning operations (MLOps) along with other software assets within a continuous integration/continuous delivery (CI/CD) pipeline.

The responsibility for implementing SSDF practices in the Profile may be shared among multiple organizations. For example, an AI model could be produced by one organization and executed within an AI system hosted by a second organization, which is then used by other organizations.

[2] For consistency with SSDF 1.1, this document uses a general software development life cycle. Organizations using this document are encouraged to adapt it to any machine learning-specific life cycle they are using.

In these situations, there is likely a shared responsibility model involving the AI model producer, AI system producer, and AI system acquirer. An AI system acquirer can establish an agreement with an AI system producer and/or AI model producer that specifies which party is responsible for each practice and task and how each party will attest to its conformance with the agreement.

A limitation of the SSDF and this Profile is that they only address cybersecurity risk management. There are many other types of risks to AI systems (e.g., data privacy, intellectual property, and bias) that organizations should manage along with cybersecurity risk as part of a mature enterprise risk management program. NIST resources on identifying and managing other types of risk include:

- *AI Risk Management Framework (AI RMF)* [2] and the *NIST AI RMF Playbook* [11]
- *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations* [3]
- *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile* [12]
- *Towards a Standard for Identifying and Managing Bias in Artificial Intelligence* [4]
- *Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations* [10]
- *NIST Privacy Framework: A Tool for Improving Privacy Through Enterprise Risk Management, Version 1.0* [13]
- *Integrating Cybersecurity and Enterprise Risk Management (ERM)* [14]

3. SSDF Community Profile for AI Model Development

Table 1 defines the SSDF Community Profile for AI Model Development. The meanings of each column are as follows:

- **Practice** contains the name of the practice and a unique identifier, followed by a brief explanation of what the practice is and why it is beneficial.

  Task specifies one or more actions that may be needed to perform a practice. Each task includes a unique identifier and a brief explanation.

  All practices and tasks are unchanged from SSDF version 1.1 unless they are explicitly tagged as "Modified from SSDF 1.1" or "Not part of SSDF 1.1." An example is the PW.3 practice, **"**Confirm the Integrity of Training, Testing, Fine-Tuning, and Aligning Data Before Use" and all of its tasks.

- **Priority** reflects the suggested relative importance of each task *within the context of the profile* and is intended to be a starting point for organizations to assign their own priorities:
  - **High:** Critically important for AI model development security compared to other tasks
  - **Medium:** Directly supports AI model development security
  - **Low:** Beneficial for secure software development but is generally not more important than most other tasks
- **Recommendations, Considerations, and Notes Specific to AI Model Development** may contain one or more items that recommend what to do or describe additional considerations for a particular task. Organizations are expected to adapt, customize, and omit items as necessary as part of the risk-based approach described in Section 2.

  Each item has an ID starting with one of the following:
  - "R" (recommendation: something the organization should do)
  - "C" (consideration: something the organization should consider doing)
  - "N" (note: additional information besides recommendations and considerations)

  An R, C, or N designation and its number can be appended to the task ID to create a unique identifier (e.g., "PO.1.2.R1" is the first recommendation for task PO.1.2).

  Note that a value of "No additions to SSDF 1.1" in this column indicates that the Profile does not contain recommendations, considerations, or notes specific to AI model development for the task. Refer to SSDF version 1.1 [6] for baseline guidance on the secure development task in question and to the other references in this document for additional information related to the task.
- **Informative References** point (map) to parts of standards, guidance, and other content containing requirements, recommendations, considerations, or other supporting

information on performing a particular task. The Informative References come from the following sources:

- *AI Risk Management Framework 1.0* [2]. Several crosswalks have already been defined between the AI RMF and other guidance and standards; see https://airc.nist.gov/AI_RMF_Knowledge_Base/Crosswalks for the current set.
- *OWASP Top 10 for LLM Applications Version 1.1* [15]. Each identifier indicates one of the top 10 vulnerability types and might also refer to an individual prevention and mitigation strategy for that vulnerability type.
- *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations* [3]. This report outlines key types of machine learning attack stages and attacker goals, objectives, and capabilities, as well as corresponding methods for mitigating and managing the consequences of attacks.

NIST is also considering adding a column for Implementation Examples in a future version of the Profile. An **Implementation Example** is a single sentence that suggests a way to accomplish part or all of a task. While the Recommendations and Considerations column describes the "what," Implementation Examples would describe options for the "how." Such examples added to this Profile would supplement those already defined in SSDF version 1.1. See the Note to Readers for more information on providing input on additional Informative References and Implementation Examples.

Note: This Profile supplements what SSDF version 1.1 [6] already includes and is intended to be used in conjunction with it, not on its own. As a reminder, the deployment and operation of AI systems with AI models are out of the Profile's scope, as are most parts of the data governance and management life cycle.

There are gaps in the numbering of some SSDF practices and tasks. For example, the PW.4 practice has three tasks: PW.4.1, PW.4.2, and PW.4.4. PW.4.3 was a task in SSDF version 1.0 that was moved elsewhere for version 1.1, so its ID was not reused.

Table 1. SSDF Community Profile for AI Model Development

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
Prepare the Organization (PO)				
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 software development life cycle (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).	**PO.1.1:** Identify and document all security requirements for the organization's software development infrastructures and processes, and maintain the requirements over time.	High	**R1:** Include AI model development in the security requirements for software development infrastructure and processes. **R2:** Identify and select appropriate AI model architectures and training techniques in accordance with recommended practices for cybersecurity, privacy, and reproducibility.	**AI RMF:** Map 1.3, 1.5, 1.6
	PO.1.2: Identify and document all security requirements for organization-developed software to meet, and maintain the requirements over time.	High	**R1:** Organizational policies should support all current requirements specific to AI model development security for organization-developed software. These requirements should include the areas of AI model development, AI model operations, and data science. Requirements may come from many sources, including laws, regulations, contracts, and standards. **C1:** Consider reusing or expanding the organization's existing data classification policy and processes. **N1:** Possible forms of AI model documentation include data, model, and system cards.	**AI RMF:** Govern 1.1, 1.2, 3.2, 4.1, 5.1, 6.1; Map 1.1
	PO.1.3: Communicate requirements to all third parties who will provide commercial software components to the organization for use by the organization's own software. [Modified from SSDF 1.1]	Medium	**R1:** Include AI model development security in the requirements being communicated for third-party software components.	**AI RMF:** Map 4.1, 4.2 **OWASP:** LLM05-1

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
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.	**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.	High	**R1:** Include AI model development security in SDLC-related roles and responsibilities throughout the SDLC. The roles and responsibilities should include, but are not limited to, AI model development, AI model operations, and data science. **N1:** Roles and responsibilities involving AI system producers, AI model producers, and other third-party providers can be documented in agreements.	**AI RMF:** Govern 2.1
	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.	High	**R1:** Role-based training should include understanding cybersecurity vulnerabilities and threats to AI models and their possible mitigations.	**AI RMF:** Govern 2.2 **OWASP:** LLM04-7
	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.	Medium	**R1:** Leadership should commit to secure development practices involving AI models.	**AI RMF:** Govern 2.3
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.	**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.	High	**R1:** Plan to develop and implement automated toolchains that secure AI model development and reduce human effort, especially at the scale often used by AI models. **N1:** Ideally, automated toolchains will perform the vast majority of the work related to securing AI model development. **N2:** See PO.4, PO.5, PS, and PW for information on tool types.	**AI RMF:** Measure 2.1 **OWASP:** LLM08
	PO.3.2: Follow recommended security practices to deploy, operate, and maintain tools and toolchains.	High	**R1:** Execute the plan to develop and implement automated toolchains that secure AI model development and reduce human effort, especially at the scale often used by AI models. **R2:** Verify the security of toolchains at a frequency commensurate with risk.	**AI RMF:** Measure 2.1 **OWASP:** LLM05-3, LLM05-9, LLM08, LLM09

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
	PO.3.3: Configure tools to generate artifacts of their support of secure software development practices as defined by the organization.	Medium	**N1:** An *artifact* is "a piece of evidence" [16]. *Evidence* is "grounds for belief or disbelief; data on which to base proof or to establish truth or falsehood" [17]. Artifacts provide records of secure software development practices. Examples of artifacts specific to AI model development include attestations of the integrity and provenance of training datasets.	**AI RMF:** Measure 2.1
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.	**PO.4.1:** Define criteria for software security checks and track throughout the SDLC.	Medium	**R1:** Implement guardrails and other controls throughout the AI development life cycle, extending beyond the traditional SDLC. **C1:** Consider requiring review and approval from a human-in-the-loop for software security checks beyond risk-based thresholds.	**AI RMF:** Measure 2.3, 2.7; Manage 1.1 **OWASP:** LLM01-2
	PO.4.2: Implement processes, mechanisms, etc. to gather and safeguard the necessary information in support of the criteria.	Low	No additions to SSDF 1.1	**AI RMF:** Measure 2.3, 2.7; Manage 1.1 **OWASP:** LLM01-2
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, AI model training, build, test, and distribution environments. [Modified from SSDF 1.1]	**PO.5.1:** Separate and protect each environment involved in software development.	High	**C1:** Consider separating execution environments from each other to the extent feasible, such as through isolation, segmentation, containment, access via APIs, or other means. **R1:** Monitor, track, and limit resource usage and rates for AI model users during model development. **R2:** Only store sensitive data used during AI model development, including production data, within organization-approved environments and locations within those environments. **R3:** Protect all training pipelines, model registries, and other components within the environments according to the principle of least privilege.	**OWASP:** LLM01-1, LLM01-4, LLM04, LLM08, LLM10

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
			R4: Continuously monitor training-related activity in pipelines and model modifications in the model registry. **R5:** Follow recommended practices for securely configuring each environment. **R6:** Continuously monitor each environment for plaintext secrets.	
	PO.5.2: Secure and harden development endpoints (endpoints for software designers, developers, testers, builders, etc.) to perform development tasks using a risk-based approach.	Medium	No additions to SSDF 1.1	**OWASP:** LLM01-1, LLM05-3, LLM05-9, LLM08
	PO.5.3: Continuously monitor software execution performance and behavior in software development environments to identify potential suspicious activity and other issues. [Not part of SSDF 1.1]	High	**R1:** Perform continuous security monitoring for all development environment components that host an AI model or related resources (e.g., model APIs, weights, configuration parameters, training datasets). **R2:** Continuous monitoring and analysis tools should generate alerts when detected activity involving an AI model passes a risk threshold or otherwise merits additional investigation.	**AI RMF:** Measure 2.4 **OWASP:** LLM03-7, LLM04, LLM05-8, LLM09, LLM10
Protect Software (PS)				
Protect All Forms of Code and Data from Unauthorized Access and Tampering (PS.1): Help prevent unauthorized changes to code and data, both inadvertent and intentional, which could circumvent or negate the intended security characteristics of the software. For code and data that are 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. [Modified from SSDF 1.1]	**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.	High	**R1:** Secure code storage should include AI models, model weights, pipelines, reward models, and any other AI model elements that need their confidentiality, integrity, and/or availability protected. These elements do not all have to be stored in the same place or through the same type of mechanism. **R2:** Follow the principle of least privilege to minimize direct access to AI models and model elements regardless of where they are stored or executed. **R3:** Store reward models separately from AI models and data.	**OWASP:** LLM10

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
			R4: Permit indirect access only to model weights. **C1:** Consider preventing all human access to model weights. **C2:** Consider requiring all AI model development to be performed within organization-approved environments only.	
	PS.1.2: Protect all training, testing, fine-tuning, and aligning data from unauthorized access and modification. [Not part of SSDF 1.1]	High	**R1:** Continuously monitor the confidentiality (for non-public data only) and integrity of training, testing, fine-tuning, and aligning data. **C1:** Consider securely storing training, testing, fine-tuning, and aligning data for future use and reference if feasible.	**OWASP:** LLM03, LLM06, LLM10
	PS.1.3: Protect all model weights and configuration parameter data from unauthorized access and modification. [Not part of SSDF 1.1]	High	**R1:** Keep model weights and configuration parameters separate from training, testing, fine-tuning, and aligning data. **R2:** Continuously monitor the confidentiality (for closed models only) and integrity of model weights and configuration parameters. **R3:** Follow the principle of least privilege to restrict access to AI model weights, configuration parameters, and services during development. **R4**: Specify and implement additional risk-proportionate cybersecurity practices around model weights, such as encryption, cryptographic hashes, digital signatures, multi-party authorization, and air-gapped environments.	**OWASP:** LLM10
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.	**PS.2.1:** Make software integrity verification information available to software acquirers.	Medium	**R1:** Generate and provide cryptographic hashes or digital signatures for an AI model and its components, artifacts, and documentation. **R2:** Provide digital signatures for AI model changes.	**OWASP:** LLM05-6
Archive and Protect Each Software Release (PS.3): Preserve software releases in order to	**PS.3.1:** Securely archive the necessary files and supporting data (e.g., integrity	Low	**R1:** Perform versioning and tracking for infrastructure tools (e.g., pre-processing,	**OWASP:** LLM10

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
help identify, analyze, and eliminate vulnerabilities discovered in the software after release.	verification information, provenance data) to be retained for each software release.		transforms, collection) that support dataset creation and model training. **R2:** Include documentation of the justification for AI model selection in the retained information. **R3:** Include documentation of the entire training process, such as data preprocessing and model architecture. **N1:** AI models and their components may need to be added at this time to an organization's asset inventories.	
	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], through Supply-chain Levels for Software Artifacts [SLSA]). [Modified from SSDF 1.1]	Medium	**R1:** Track the provenance of an AI model and its components and derivatives, including the training libraries, frameworks, and pipelines used to build the model. **R2:** Track AI models that were trained on sensitive data (e.g., payment card data, protected health information, other types of personally identifiable information), and determine if access to the models should be restricted to individuals who already have access to the sensitive data used for training. **C1:** Consider disclosing the provenance of the training, testing, fine-tuning, and aligning data used for an AI model.	**OWASP:** LLM03-1, LLM05-4, LLM05-5, LLM10
Produce Well-Secured Software (PW)				
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	**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.	High	**R1:** Incorporate relevant AI model-specific vulnerability and threat types in risk modeling. Examples of these vulnerability and threat types include poisoning of training data, malicious code or other unwanted content in inputs and outputs, denial-of-service conditions arising from adversarial prompts, supply chain attacks, unauthorized information disclosure, theft of AI model weights, and misconfiguration of data pipelines. [3]	**AI RMF:** Govern 4.1, 4.2; Map 5.1; Measure 1.1; Manage 1.2, 1.3 **OWASP:** LLM01, LLM02, LLM03, LLM04, LLM05, LLM06,

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
security requirements and risks during software design (secure by design) is key for improving software security and also helps improve development efficiency.			**C1:** Consider periodic risk modeling updates for future AI model versions and derivatives after AI model release. **C2:** During risk modeling, consider checking that the AI model is not in a critical path to make significant security decisions without a human in the loop.	LLM07, LLM08, LLM09, LLM10
	PW.1.2: Track and maintain the software's security requirements, risks, and design decisions.	Medium	No additions to SSDF 1.1	**AI RMF:** Govern 4.1, 4.2; Map 2.1, 2.2, 2.3, 3.2, 3.3, 4.1, 4.2, 5.2; Manage 1.2, 1.3, 1.4
	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.	Medium	No additions to SSDF 1.1	
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.	**PW.2.1:** Have 1) a qualified person (or people) who were not involved with the design and 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. [Modified from SSDF 1.1]	High	No additions to SSDF 1.1	**AI RMF:** Measure 2.7; Manage 1.1
Confirm the Integrity of Training, Testing, Fine-Tuning, and Aligning Data Before Use (PW.3): Prevent data that is likely to negatively impact the cybersecurity of the AI model from	**PW.3.1:** Analyze data for signs of data poisoning, bias, homogeneity, and tampering before using it for AI model training, testing, fine-tuning, or aligning	High	**R1:** Verify the provenance (when known) and integrity of training, testing, fine-tuning, and aligning data before use.	**AI RMF:** Measure 2.1; Manage 1.2, 1.3

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
being consumed as part of AI model training, testing, fine-tuning, and aligning. [Not part of SSDF 1.1]	purposes, and mitigate the risks as necessary. [Not part of SSDF 1.1]		**R2:** Select and apply appropriate methods for analyzing and altering the training, testing, fine-tuning, and aligning data for an AI model. Examples of methods include anomaly detection, bias detection, data cleaning, data curation, data filtering, data sanitization, fact-checking, and noise reduction. **C1:** Consider using a human-in-the-loop to examine data, such as with exploratory data analysis techniques [18].	**OWASP:** LLM03, LLM06
	PW.3.2: Track the provenance, when known, of all training, testing, fine-tuning, and aligning data used for an AI model, and document which data do not have known provenance. [Not part of SSDF 1.1]	Medium	**N1:** Provenance verification is not possible in all cases because provenance is not always known. However, it is still beneficial for security purposes to track and verify provenance whenever possible, and to track when provenance is unknown.	**AI RMF:** Measure 2.1 **OWASP:** LLM03-1 **Adv ML**
	PW.3.3: Include adversarial samples in the training and testing data to improve attack prevention. [Not part of SSDF 1.1]	Medium	**R1:** Use a process and corresponding controls to test the adversarial samples and put appropriate guardrails on training and testing use.	**OWASP:** LLM03-6, LLM05-7 **Adv ML**
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.	**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.	Medium	**C1:** Consider using an existing AI model instead of creating a new one.	**OWASP:** LLM05
	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.	Low	No additions to SSDF 1.1	
	PW.4.4: Verify that acquired commercial, open-source, and all other third-party software components comply with the	High	**R1:** Verify the integrity, provenance, and security of an existing AI model or any other acquired AI components — including training, testing, fine-tuning, and aligning datasets;	**OWASP:** LLM05-2, LLM05-6 **Adv ML**

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
	requirements, as defined by the organization, throughout their life cycles.		reward models; adaptation layers; and configuration parameters — before using them. **R2:** Scan and thoroughly test acquired AI models and their components for vulnerabilities and malicious content before use.	
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.	**PW.5.1:** Follow all secure coding practices that are appropriate to the development languages and environment to meet the organization's requirements.	High	**R1:** Expand secure coding practices to include AI technology-specific considerations. **R2:** Code the handling of inputs (including prompts and user data) and outputs carefully. All inputs and outputs should be logged, analyzed, and validated within the context of the AI model, and those with issues should be sanitized or dropped. **R3:** Encode inputs and outputs to prevent the execution of unauthorized code.	**AI RMF:** Manage 1.2, 1.3, 1.4 **OWASP:** LLM01, LLM02, LLM04-1, LLM06, LLM07, LLM09-9, LLM10
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.	**PW.6.1:** Use compiler, interpreter, and build tools that offer features to improve executable security.	Low	**C1:** Consider using secure model serialization mechanisms that reduce or eliminate vectors for the introduction of malicious content.	
	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.	Low	**C1:** Consider capturing compiler, interpreter, and build tool versions and features as part of the provenance tracking.	
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.	**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.	Medium	**R1:** Code review and analysis policies or guidelines should include code for AI models and other related components. **C1:** Consider performing scans of AI model code in addition to testing the AI models.	
	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	High	**R1:** Scan all AI models for malware, vulnerabilities, backdoors, and other security issues in accordance with the organization's code review and analysis policies or guidelines.	**AI RMF:** Measure 2.3, 2.7; Manage 1.1, 1.2, 1.3, 1.4

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
	development team's workflow or issue tracking system.			**OWASP:** LLM03-7d, LLM07-4
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.	**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.	High	**R1:** Include AI models in code testing policies and guidelines. Several forms of code testing can be used for AI models, including unit testing, integration testing, penetration testing, red teaming, use case testing, and adversarial testing. **C1:** Consider automating tests within a development pipeline as part of regression testing where possible.	
	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.	High	**R1:** Test all AI models for vulnerabilities in accordance with the organization's code testing policies or guidelines. **R2:** Retest AI models when they are retrained or new data sources are added.	**AI RMF:** Measure 2.2, 2.3, 2.7; Manage 1.1, 1.2, 1.3, 1.4 **OWASP:** LLM03-7d, LLM05-7, LLM07-4
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.	**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.	Medium	No additions to SSDF 1.1	**AI RMF:** Measure 2.7
	PW.9.2: Implement the default settings (or groups of default settings, if applicable), and document each setting for software administrators.	Medium	**N1:** Documenting settings can be performed earlier in the process, such as when defining a secure baseline (see PW.9.1).	**AI RMF:** Measure 2.7; Manage 1.2, 1.3, 1.4
Respond to Vulnerabilities (RV)				
Identify and Confirm Vulnerabilities on an Ongoing Basis (RV.1): Help ensure that vulnerabilities are identified more quickly so	**RV.1.1:** Gather information from software acquirers, users, and public sources on potential vulnerabilities in the software	High	**R1:** Log, monitor, and analyze all inputs and outputs for AI models to detect possible security and performance issues (see PO.5.3).	**AI RMF:** Govern 4.3, 5.1, 6.1, 6.2;

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
that they can be remediated more quickly in accordance with risk, reducing the window of opportunity for attackers.	and third-party components that the software uses, and investigate all credible reports.		**R2:** Make the users of AI models aware of mechanisms for reporting potential security and performance issues. **N1:** In this context, "users" refers to AI system producers and acquirers who are using an AI model. **R3:** Monitor vulnerability and incident databases for information on AI-related concerns, including the machine learning frameworks and libraries used to build AI models.	Measure 1.2, 2.4, 2.5, 2.7, 3.1, 3.2, 3.3; Manage 4.1 **OWASP:** LLM03-7a, LLM09, LLM10
	RV.1.2: Review, analyze, and/or test the software's code to identify or confirm the presence of previously undetected vulnerabilities.	Medium	**R1:** Scan and test AI models frequently to identify previously undetected vulnerabilities. **R2:** Rely mainly on automation for ongoing scanning and testing, and involve a human-in-the-loop as needed. **R3:** Conduct periodic audits of AI models.	**AI RMF:** Govern 4.3; Measure 1.3, 2.4, 2.7, 3.1; Manage 4.1 **OWASP:** LLM03-7b, LLM03-7d
	RV.1.3: Have a policy that addresses vulnerability disclosure and remediation, and implement the roles, responsibilities, and processes needed to support that policy.	Medium	**R1:** Include AI model vulnerabilities in organization vulnerability disclosure and remediation policies. **R2:** Make users of AI models aware of their inherent limitations and how to report any cybersecurity problems that they encounter.	**AI RMF:** Govern 4.3, 5.1, 6.1; Measure 3.1, 3.3; Manage 4.3
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.	**RV.2.1:** Analyze each vulnerability to gather sufficient information about risk to plan its remediation or other risk response.	Medium	**N1:** This may include deep analysis of generative AI and dual-use foundation model input and output to detect deviations from normal behavior.	**AI RMF:** Govern 4.3, 5.1, 6.1; Measure 2.7, 3.1; Manage 1.2, 2.3, 4.1 **Adv ML**
	RV.2.2: Plan and implement risk responses for vulnerabilities.	High	**R1:** Risk responses for AI models should consider the time and expenses that may be associated with rebuilding them.	**AI RMF:** Govern 5.1, 5.2, 6.1; Measure 3.3;

Practice	Task	Priority	Recommendations [R], Considerations [C], and Notes [N] Specific to AI Model Development	Informative References
			R2: Establish and implement criteria and processes for when to stop using an AI model and when to roll back to a previous version and its components. **C1:** Consider being prepared to stop using an AI model at any time and to continue operations through other means until the AI model's risks are sufficiently addressed.	Manage 1.3, 2.1, 2.3, 2.4, 4.1
Analyze Vulnerabilities to Identify Their Root Causes (RV.3): Help reduce the frequency of vulnerabilities in the future.	**RV.3.1:** Analyze identified vulnerabilities to determine their root causes.	Medium	**N1:** The ability to review training, testing, fine-tuning, and aligning data after the fact can help identify some root causes.	**AI RMF:** Govern 5.1, 6.1; Measure 2.7, 3.1; Manage 2.3, 4.1
	RV.3.2: Analyze the root causes over time to identify patterns, such as a particular secure coding practice not being followed consistently.	Medium	No additions to SSDF 1.1	**AI RMF:** Govern 5.1, 6.1; Measure 2.7, 3.1; Manage 4.1, 4.3
	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.	Medium	No additions to SSDF 1.1	**AI RMF:** Govern 5.1, 5.2, 6.1; Measure 2.7, 3.1; Manage 4.1, 4.2, 4.3
	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.	Medium	No additions to SSDF 1.1	**AI RMF:** Govern 5.2, 6.1; Measure 2.7, 3.1; Manage 4.2, 4.3