Openpolicy

June 2, 2024
To: NIST
Via SSDF@nist.gov

OpenPolicy comments on
National Institute of Standards and Technology (NIST) SP 800-218A: Secure Software Development Practices for Generative AI and Dual-Use Foundation Models

Overview

OpenPolicy appreciates the opportunity to provide feedback on the NIST SP 800-218A: Secure Software Development Practices for Generative AI and Dual-Use Foundation Models. As a technology company dedicated to democratizing the ability of innovative enterprises of all sizes to engage with policymakers, OpenPolicy is committed to fostering open policymaking through the use of AI, innovation, and technology. We aim to support broader access and participation in the regulatory development process by connecting innovators with policymakers.

We look forward to active engagement with NIST as this framework evolves and stand ready to offer additional input as needed. We are dedicated to supporting the implementation of the AI Executive Order and its objectives, promoting trustworthy and responsible development and use of AI systems, and actively collaborating with NIST and other implementing agencies. We proudly participate in the NISA AI Safety Institute Consortium, reinforcing our commitment to guiding organizations in managing AI risks and advancing secure software development practices. We provided prior feedback and engaged on the following deliverables related to the AI Executive Order, and AI policy more broadly. [1]

We appreciate NIST's active engagement with government, industry, academia, and the public on the framework development and believe that an open and collaborative policymaking dialogue is essential, and encourage NIST to continue to expand its engagement with innovative companies. This approach supports the implementation of NIST's best practices and ongoing efforts to ensure secure practices for AI model development within the framework of the SSDF, emphasizing cybersecurity and risk-based approaches tailored to the AI lifecycle, which is critical to fostering an appropriate, updated,

[1] OpenPolicy's OMB AI RFI Comments https://docs.google.com/document/d/1xUFSi-8_klqlJeAjQS0UO9QFk6Xm9o7Sk6kv5uOQy18/edit?usp=sharing ; OpenPolicy's NIST AI RFI Comments https://docs.google.com/document/d/1P2LZTMgrRzzNNXkEYRpHs7eVhN_X6bMIaAB5aOQn7gw/edit?usp=sharing ;

and scalable secure software development practices for generative AI and dual-use foundation models, because it allows NIST to adapt the guidelines to the most current risks, and required controls and methods to address them The participation of innovative companies, particularly startups specializing in cutting-edge AI security, trust and safety solutions, significantly contributes to this effort. Indeed, many technologies used to support the requirements of the AI EO, OMB AI memo, and the NIST RFIs and guidelines referenced in this draft are developed by these innovative companies and startups. These technologies evaluate the measurements, considerations, and development of AI software, including data sourcing, designing, training, fine-tuning, and evaluation perspectives that guide and lead the industry, which OpenPolicy actively collaborates with these communities: Armis, HiddenLayer, Lasso, Kiteworks, HUMAN, Finite State, Cranium, InfoSec Global, Merlin Cyber, and ThriveDX.

AI risks and enhancing AI security

We commend NIST for its leadership in implementing the AI EO requirements, especially with an emphasis on managing AI risks and enhancing AI security. However, there are areas where the approach outlined in NIST SP 800-218A can be further refined to address the broader spectrum of AI-related threats and to align with the dynamic nature of AI technology and its integration with other technological paradigms, as well as governance approaches outlined in other NIST frameworks such as the CSF. While NIST SP 800-218A provides valuable, secure software development practices tailored for generative AI and dual-use foundation models, it could benefit from incorporating broader risk management strategies, including for risks introduced by such models to other software, and technology environments but extend beyond the model's development life cycle.

As AI systems become more sophisticated, so do the tactics and techniques employed by malicious actors to exploit vulnerabilities within these models, or threat vectors that have been amplified due to these models' introduction into larger software development life cycles or technology use cases. Traditional cyber security and risk methods of threat monitoring, detection, and response are often too slow and manual to effectively counteract these rapidly emerging threats as these methods were not formed for the model types involved in AI and are thus blind to the multifaceted AI attack techniques and vectors. Given the rapidly evolving threat landscape and the increasing complexity of AI systems, risk measurement and controls must be significantly enhanced. It is crucial that technology and automation are scaled up to effectively measure security outcomes and manage risks in governance and compliance.

A technology-scaled mindset to enhance governance and measurements should underpin the revision to SSDF to address threats introduced by AI

OpenPolicy encourages NIST to identify further opportunities to leverage technology and automation tools to ensure the suggested practices align with a proactive, secure, and safe approach to AI. Establishing AI security and risk measurement controls alone is insufficient; a scalable strategy is necessary to implement these controls effectively. In other words, laying down controls for AI security and risk measurement is inadequate without a scaled approach to guarantee and measure their adoption. Thus, It is imperative to leverage automated tools for real-time detection, monitoring, and response to AI software threats. These tools not only provide the necessary scale, speed, and agility to counteract dynamic threats but also support ongoing regulatory efforts to enhance AI security and compliance. By integrating governance automation tools like OSCAL, organizations can ensure robust security and effective compliance management, thereby fostering safer AI software development and a more trustworthy AI ecosystem. Ongoing regulatory and policy efforts, such as those led by NIST and the AI EO 14110, emphasize the need for robust and adaptive security measures. By leveraging automated tools to enhance governance, risk identification, and mitigation, organizations can more effectively align with these regulations, ensuring that their AI systems are both secure and compliant while enhancing the accuracy and consistency of compliance efforts.

AI management and testing

In the pursuit of managing AI software risk, the integration of automated tools with the nuanced understanding of human testers in AI red teaming is not just beneficial, but crucial. This approach, which involves both general red teamers for broad flaw detection and specialists for pinpointing more nuanced issues, fosters a collaborative environment that draws on the community's collective wisdom. By encouraging practices like bias bounties and vulnerability disclosures, NIST can tap into a wealth of knowledge and uncover otherwise unnoticed flaws.

Further, NIST should consider developing a unified approach for AI testing, building on methodologies such as MITRE-ATLAS, clearly distinguishing between cybersecurity and AI-specific terms to support the practical evaluation of AI systems. This involves developing and differentiating testing methodologies for security and “non-security harms,” such as bias and discrimination, and establishing terminology for algorithmic flaws. Distinct definitions and testing methodologies for traditional cybersecurity threats versus AI algorithmic flaws will ensure a comprehensive evaluation of AI systems. Establishing clear standards and language for "AI red-teaming" can further distinguish between adversarial and non-adversarial testing approaches, enhancing the efficacy and trustworthiness of AI technologies.

Given the dynamic nature of AI threats and risks necessitates a proactive approach to security, where continuous monitoring enables the detection of anomalies and potential threats as they occur, allowing immediate action to mitigate risks before they can cause significant harm- NIST should underscore the need for continuous monitoring and evaluation of AI systems. This includes real-time detection and alerts, monitoring AI data flows to prevent unauthorized disclosures, and enforcing comprehensive software protection policies to ensure a robust software security framework for AI models. Real-time detection and alert systems are vital for promptly identifying and responding to security breaches, thereby reducing the window of opportunity for malicious actors.

By emphasizing continuous monitoring and evaluation of AI systems, integrating AI red teaming and continuous testing, and enhancing model validation and human rights impact assessments through automated tools, NIST can significantly strengthen the security and trustworthiness of AI technologies. Red teaming involves simulating attacks to uncover weaknesses in AI systems, while continuous testing ensures that these systems remain secure over time. These proactive measures help strengthen AI systems against emerging threats and ensure their resilience, thereby being essential practices for addressing the complex and evolving threats faced by AI systems and ensuring their safe and ethical deployment. Additionally, incorporating non-invasive, software-based measures supports transparency and adaptability with minimal disruption, safeguarding AI models without compromising their functionality or data integrity. These measures are essential for addressing the complex and evolving threats faced by AI models, ensuring organizations that their AI software and systems are protected against threats while maintaining operational efficiency.

Layered defense strategy

We appreciate NIST's efforts to establish secure software development practices for AI and LLMs and recommend emphasizing advanced security measures critical for addressing unique vulnerabilities associated with these technologies. Particularly, there is a vital need to prevent data leakage, not only by protecting direct data exposures but also by securing subtle leakages through model outputs, which could be addressed and overcome through real-time monitoring and anomaly detection, as mentioned above, to ensure vulnerabilities are managed before escalating into more significant issues.

Adaptive policy enforcement and regulatory compliance are not just important, they are essential in today's continuously evolving threat landscape. Security policies tailored to the specific risks and operational contexts of LLM applications should be flexible enough to adapt to new threats and regulatory changes, ensuring ongoing compliance and robust security. Therefore, NIST should consider advocating for integrating security from the early

stages of AI application development, making security foundational rather than an afterthought, enhancing the overall security posture of AI systems, and strengthening supply chain security.

A comprehensive approach to verifying and monitoring all components involved in LLM deployment can mitigate risks from third-party sources. Additionally, insights from ActiveFence's[2] technology review suggest the importance of addressing specific vulnerabilities that LLMs exhibit in handling sensitive content across various languages and cultural contexts. Their findings emphasize the need for dynamic content analysis and adaptation to better manage emergent risks, particularly in non-English languages and in the nuanced understanding of cultural contexts, which could profoundly impact model safety and effectiveness.

Therefore, NIST's guidelines could also recommend technologies, solutions, and procedures that support continuous assessment and adjustment of security measures to keep pace with the rapid evolution of AI technologies and threat vectors. Integrating standards for advanced data protection, including techniques such as real-time data masking and encryption, is essential to maintain data integrity and confidentiality throughout the AI model's lifecycle. This underscores the aim of SP 800-218A to ensure the secure deployment and operation of AI technologies, supporting the safe and secure adoption of AI across various sectors. By incorporating a layered defense strategy and advanced threat detection mechanisms, these guidelines can provide a more comprehensive framework for the secure deployment and operation of AI technologies, effectively addressing the complex and evolving threat environment.

Under the Prepare the Organization (PO) section, which recommends these suggestions

- **Define Security Requirements for Software Development (PO.1)**

Task

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. 1.1: Identify, document, and continuously update all security requirements for the organization's software development infrastructures and processes, and maintain the requirements over time.

[2] ActiveFence's The LLM Safety Review https://www.activefence.com/LLMSafety

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Include AI model development in the security requirements for software development infrastructure and processes.

Task

PO.1.2: Identify and document all security requirements for organization-developed software to meet, and maintain the requirements over time

PO. 1.2: Identify, document, and continuously update all security requirements for organization-developed software to meet, and maintain the requirements over time.

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

R1: Organizational policies should support all current requirements specific to AI model development security for organization-developed software. These requirements should include areas such as AI model development, AI model operations, and data science, continuously updated through automated and manly processes. Requirements may come from many sources, including laws, regulations, contracts, and standards.

Task

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.

PO. 1.3: Communicate and continuously review requirements to all third parties who will provide commercial software components to the organization for reuse by the organization's own software using governance tools.

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Include AI model development security in the requirements being communicated for third-party software components.

- **Implement Roles and Responsibilities (PO.2)**

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Include AI model development security in SDLC-related roles and responsibilities throughout the SDLC. The roles and responsibilities should include AI model development, AI model operations, and data science.

R1: Include AI model development security in SDLC-related roles and responsibilities throughout the SDLC. The roles and responsibilities should include AI model development, AI model operations, and data science, with continuous updates and training.

N1: Roles and responsibilities involving AI system producers, AI model producers, and other third-party providers can be documented in agreements.

Task

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.2.2: 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, incorporating real-time threat updates and simulations.

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Role-based training should include understanding cybersecurity vulnerabilities and threats to AI models and their possible mitigations.

R1: Role-based training should include understanding cybersecurity vulnerabilities and threats to AI models and their possible mitigations, supported by technological tools for real-time updates and simulations.

Task

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.

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, ensuring alignment with security and governance practices.

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Leadership should commit to secure development practices involving AI models.

R1: Leadership should commit to secure development practices involving AI models, ensuring that these practices are continuously monitored and updated through technological systems.

- **Implement Supporting Toolchains (PO.3)**

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

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, supporting automated compliance checks.

N1: Ideally, automated toolchains will perform the vast majority of the work related to securing AI model development.

N1: Ideally, automated toolchains will perform the vast majority of the work related to securing AI model development, including continuous testing and real-time threat detection.

N2: See PO.4, PO.5, PS, and PW for information on tool types.

Task

PO.3.2: Follow recommended security practices to deploy, operate, and maintain tools and toolchains

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

R2: Verify the security of toolchains at a frequency commensurate with risk, using automated tools to ensure comprehensive coverage.

Task

PO.3.3: Configure tools to generate artifacts of their support of secure software development practices as defined by the organization.

PO. 3.3: Configure tools to generate artifacts of their support of secure software development practices as defined by the organization, ensuring these artifacts are generated and verified.

Recommendations, Considerations, and Notes Specific to AI Model Development

N1: An artifact is "a piece of evidence". Evidence is "grounds for belief or disbelief; data on which to base proof or to establish truth or falsehood". Artifacts provide records of secure software development practices. Examples of artifacts specific to AI model development include attestations of training dataset integrity and provenance.

- **Practice: Define and Use Criteria for Software Security Checks (PO.4)**

Task

PO.4.1: Define criteria for software security checks and track throughout the SDLC.

Recommendations, Considerations, and Notes Specific to AI Model Development

C1: Consider requiring review and approval from a human-in-the-loop for software security checks beyond risk-based thresholds.

C1: Consider requiring review and approval from a human-in-the-loop for software security checks beyond risk-based thresholds, including flagging potential issues and streamlining the review process.

Task

PO.4.2: Implement processes, mechanisms, etc. to gather and safeguard the necessary information in support of the criteria.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

- **Implement and Maintain Secure Environments for Software Development (PO.5)**

Task

PO.5.1: Separate and protect each environment involved in software development.

Recommendations, Considerations, and Notes Specific to AI Model Development

C1: Consider separating execution environments from each other to the extent feasible, such as by using sandboxing or containers.

R1: Monitor, track, and limit resource usage and rates for AI model users.

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

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]

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Perform continuous security and performance 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.

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, ensuring timely and efficient responses to potential threats.

- **Protect All Forms of Code and Data from Unauthorized Access and Tampering (PS.1)**

Task

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.

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, utilizing automated access controls and monitoring tools.

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

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.

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, supported by access management systems.

R3: Store reward models separately from AI models and data.

R3: Store reward models separately from AI models and data, ensuring they are monitored and managed through automated systems.

C1: Consider preventing all human access to model weights.

C2: Consider requiring all AI model development to be performed within organization-approved environments only

C2: Consider requiring all AI model development to be performed within organization-approved environments only, utilizing compliance and monitoring tools.

Task

PS.1.2: Protect all training, testing, finetuning, and aligning data from unauthorized access and modification. [Not part of SSDF 1.1]

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Continuously monitor the confidentiality 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.

C1: Consider securely storing training, testing, fine-tuning, and aligning data for future use and reference if feasible, using archival and recovery systems.

Task

PS.1.3: Protect all model weights and configuration parameter data from unauthorized access and modification. [Not part of SSDF 1.1]

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Keep model weights and configuration parameters separate from training, testing, fine-tuning, and aligning data.

R1: Keep model weights and configuration parameters separate from training, testing, fine-tuning, and aligning data, utilizing segregation and monitoring tools.

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.

R3: Follow the principle of least privilege to restrict access control 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, multiparty authorization, and air-gapped environments.

- **Provide a Mechanism for Verifying Software Release Integrity (PS.2)**

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Perform versioning and tracking for infrastructure tools (e.g., pre-processing, transforms, collection) that support dataset creation and model training.

Task

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]

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Track the provenance of an AI model and its components, including the training libraries and frameworks used to build the model.

C1: Consider disclosing the provenance of the training, testing, fine-tuning, and aligning data used for an AI model.

- ***Design Software to Meet Security Requirements and Mitigate Security Risks (PW.1)***

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

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, supply chain attacks, unauthorized information disclosure, and theft of AI model weights.

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, automated agents, supply chain attacks, unauthorized information disclosure, and theft of AI model weights.

Task

PW.1.2: Track and maintain the software's security requirements, risks, and design decisions.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

- ***Review the Software Design to Verify Compliance with Security Requirements and Risk Information (PW.2)***

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

- ***Confirm the Integrity of Training, Testing, Fine-Tuning, and Aligning Data Before Use (PW.3)***

Task

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 purposes, and mitigate the risks as necessary. [Not part of SSDF 1.1]

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Verify the provenance and integrity of all training, testing, fine-tuning, and aligning data before use.

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.

Task

PW.3.2: Track the provenance of all training, testing, fine-tuning, and aligning data used for an AI model. [Not part of SSDF 1.1].

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

PW.3.3: Include adversarial samples in the training and testing data to improve attack detection. [Not part of SSDF 1.1].

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

- **Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality (PW.4)**

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

C1: Consider using an existing AI model instead of creating a new one.

C1: Consider using an existing AI model instead of creating a new one, supported by verification compatibility and security.

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

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.

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 using verification and validation tools.

Recommendations, Considerations, and Notes Specific to AI Model Development

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; reward models; adaptation layers; and configuration parameters — before using them.

R2: Scan and thoroughly test acquired AI models and their components for vulnerabilities before use.

- **Create Source Code by Adhering to Secure Coding Practices (PW.5)**

Task

PW.5.1: Follow all secure coding practices that are appropriate to the development languages and environment to meet the organization's requirements.

PW.5.1: Follow all secure coding practices that are appropriate to the development languages and environment to meet the organization's requirements, incorporating tools to assist in adherence and validation.

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

R3: Encode inputs and outputs to prevent the execution of unauthorized code, incorporating encryption and validation tools.

- **Configure the Compilation, Interpreter, and Build Processes to Improve Executable Security (PW.6)**

Task

PW.6.1: Use compiler, interpreter, and build tools that offer features to improve executable security.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

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.

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, leveraging tools for configuration management and compliance.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

- **Review and/or Analyze Human-Readable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.7)**

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

C1: Consider performing scans of AI model code in addition to testing the AI models, leveraging scanning tools.

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

- **Test Executable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements (PW.8)**

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

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, and adversarial testing.

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, and adversarial testing, supported by other testing tools.

Task

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.

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, using tools for tracking and analysis.

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Test all AI models for vulnerabilities in accordance with the organization's code testing policies or guidelines.

R1: Test all AI models for vulnerabilities in accordance with the organization's code testing policies or guidelines using automated testing tools for efficiency and accuracy.

- **Configure Software to Have Secure Settings by Default (PW.9)**

Task

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.9.1: 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 to ensure consistent application of secure settings.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

PW.9.2: Implement the default settings (or groups of default settings, if applicable), and document each setting for software administrators.

PW.9.2: Implement the default settings (or groups of default settings, if applicable), and document each setting for software administrators, supported by configuration management tools.

Recommendations, Considerations, and Notes Specific to AI Model Development

N1: Documenting settings can be performed earlier in the process, such as when defining a secure baseline (see PW.9.1).

N1: Documenting settings can be performed earlier in the process, such as when defining a secure baseline, using tools for documentation and compliance.

- **Identify and Confirm Vulnerabilities on an Ongoing Basis (RV.1)**

Task

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.1.1: 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 using data collection and analysis.

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Log, monitor, and analyze all inputs and outputs for AI models to detect possible security and performance issues (see PO.5.3).

R2: Make the users of AI models aware of mechanisms for reporting potential security and performance issues.

R2: Make the users of AI models aware of streamlined mechanisms for reporting potential security and performance issues.

R3: Monitor vulnerability and incident databases for information on AI-related concerns, including the machine learning frameworks and libraries used to build AI models.

R3: Monitor vulnerability and incident databases for information on AI-related concerns, including the machine learning frameworks and libraries used to build AI models for continuous monitoring.

Task

RV.1.2: Review, analyze, and/or test the software's code to identify or confirm the presence of previously undetected vulnerabilities.

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

R3: Conduct continuous audits of AI models.

Task

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.1.3: Have a policy that addresses vulnerability disclosure and remediation, and implement the roles, responsibilities, and processes needed to support that policy, supported by tracking and management systems.

Recommendations, Considerations, and Notes Specific to AI Model Development

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.

R2: Make users of AI models aware of their inherent limitations and how to report any cybersecurity problems that they encounter using reporting tools.

- **Assess, Prioritize, and Remediate Vulnerabilities (RV.2)**

Task

RV.2.1: Analyze each vulnerability to gather sufficient information about risk to plan its remediation or other risk response.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

RV.2.2: Plan and implement risk responses for vulnerabilities.

Recommendations, Considerations, and Notes Specific to AI Model Development

R1: Risk responses for AI models should consider the time and expenses that may be associated with rebuilding them.

R2: Be prepared to roll back to a previous AI model, since that may be the most feasible response in some cases.

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.

Task

RV.3.1: Analyze identified vulnerabilities to determine their root causes.

Recommendations, Considerations, and Notes Specific to AI Model Development

N1: The ability to review training, testing, fine-tuning, and aligning data after the fact can help identify some root causes.

Task

RV.3.2: Analyze the root causes over time to identify patterns, such as a particular secure coding practice not being followed consistently.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.

Task

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.

Recommendations, Considerations, and Notes Specific to AI Model Development

None.