To The National Institute of Standards and Technology (NIST),

Re: Comment on Request for Information Regarding Security Considerations for Artificial Intelligence AI Agents

This comment is submitted on behalf of the Messaging, Malware and Mobile Anti-Abuse Working Group (M^3AAWG) in response to the RFI on Security Considerations for Artificial Intelligence Agents (Docket No. 241204-0281). M^3AAWG is a technology-neutral global industry organization that brings over two decades of operational experience combating abuse at scale across email, messaging, and mobile ecosystems. Our members include major internet service providers, communications service providers, email service providers, hosting and cloud services providers, and security vendors who collectively handle billions of transactions daily. As AI agents increasingly operate in production environments with real-world consequences, we have identified security gaps that require attention.

I. Agent Architecture and Isolation

We are seeing deployments that grant AI agents far too much access with insufficient containment. Based on our members' experience with compromised systems, our recommendations are as follows:

Principle of Least Privilege for Access Control[1]
Enforcement mechanisms are needed:

- Every agent should operate in a constrained execution environment with limited capabilities, using containerization built for AI workloads.
- Agents should not have filesystem access unless specifically needed for tasks. Even then, access should be scoped to specific paths.
- API credentials should be limited to capability-specific tokens, rather than full access.
- Each agent should have its own scoped identity (service account or role) with short-lived, automatically rotated credentials stored in a secrets manager. To reduce the risk of reuse and lateral movement in the event of a compromise, credentials should not be hard-coded in prompts, code, or environment variables.
- Architect agents and their sandboxes should be designed under a "presume breach" model, using hardened runtimes and microsegmentation to prevent a single compromised agent from accessing other agents, shared memory, or high-value systems.
- Security rules should be built into the core operating system or virtual machine (kernel or hypervisor).
- Accountability and responsibility should be defined between provisioners and deployers, as well as granters and developers, for systems that use autonomous agents.

[1] In response to the RFI questions 1(d), 2(a), and 4(a).

Deployment and Update Security[2]

Incorporating machine learning (ML) models and training data into an already complex software supply chain increases deployment risk. Model artifacts should be cryptographically signed with hardware-backed verification prior to execution. Further guidelines on signing model artifacts can be found in the M[3]AAWG AI Model Lifecycle Security Best Practices.

The rollback of a compromised model should be treated with the same rigor as the rollback of a bad code deployment; however, state management is more complex. Clear guidance on state rollback and impact analysis is essential.

Provenance and Chain of Trust[3]

Lineage for every component should be documented, as outlined in the NIST Proposed Zero Draft for a Standard on Documentation of AI Datasets and AI Models.[4]

II. Data Integrity and Model Security

Data Poisoning: The Quiet Threat[5]

Data poisoning attacks are insidious because they are hard to detect and their effects may not surface until much later. For example, a few carefully crafted examples in a training set can create persistent backdoors or long-term vulnerabilities. In spam filtering, adversaries may succeed in poisoning feedback loops, potentially undermining the detection of malicious material. With AI agents, the scale and scope of poisoning may be greater.

We already have ways to defend against data poisoning, but they are not being used widely enough. For instance, identifying statistical oddities in the training data is a good start. However, to avoid creating a single point of failure in the data pipeline, pulling data from multiple sources is essential. During fine-tuning, training data should be subjected to the same careful scrutiny as production code. This "code review for datasets" approach is detailed further in the M[3]AAWG AI Model Lifecycle Security Best Practices document.

While model training attacks are still a concern, we recommend that NIST also address the increased risk of runtime memory poisoning. Adversaries can inject malicious data into Retrieval-Augmented Generation (RAG) databases, long-term memory stores, or even the immediate context window to poison the agent's persistent operational memory and alter its behavior in real time.

---

[2] In response to the RFI question 2(e) artifacts.
[3] In response to the RFI question 2(e).
[4] National Institute of Standards and Technology, “Extended Outline: Proposed Zero Draft for a Standard on Documentation of AI Datasets and AI Models,” 09 2025. https://www.nist.gov/document/extended-outline-proposed-zero-draft-standard-documentation-ai-datasets-and-ai-models

[5] In response to the RFI question 1(a).

Model Extraction and Inversion

Model extraction attacks, in which an adversary queries a model to build a functionally equivalent copy, represent a known and ongoing threat. Rate limiting and query monitoring may help mitigate the risk, but sophisticated attackers can work around these practices.

Model inversion attacks can leak training data. Training agents on proprietary business data or personally identifying information (PII) could introduce significant regulatory ramifications. Privacy-enhancing techniques and technologies, including governance and risk management strategies, may be critical defenses against model inversion, particularly for data-sensitive applications. While confidential computing and trusted execution environments (TEEs) can limit the degree of vulnerability by protecting the model during operation, they are not primary or standalone solutions for preventing data leakage resulting from model inversion.

Input/Output Sanitization[6]

Guidance is needed around large, unconstrained input/output models and systems. For example, prompt injection is the new SQL injection, following the same pattern of untrusted input being interpreted as commands. The difference is that natural language makes the attack surface enormous. Every text input is potentially malicious; therefore, additional tooling and safeguards are recommended.

Output validation is equally critical. Agent-generated code needs analysis before execution. Likewise, agent-generated system commands need to be validated against a whitelist. Tasks such as formatting text for display require proper escaping for common markup languages such as HTML. The same fundamentals still apply, but they become harder to implement when the output is probabilistic.

III. Runtime Monitoring and Behavioral Constraints

Guardrails[7]

A multilayered approach is necessary. AI safety during model training is vital but it should not be the sole safeguard against harmful behavior. For example, separate, model-agnostic policy enforcement should be implemented, with independent verification required before any consequential action is taken. While this is vital for systems security, risk assessments for AI agents should explicitly consider potential impacts on individuals—such as discriminatory outcomes, erroneous automated decisions, or reputational and economic harms—and include mechanisms for detection, escalation, and remediation.

Real-time monitoring should track resource consumption, access patterns, and behavioral drift, and compare them to dynamic baselines of activity. Severe automated actions should be avoided in the case of unusual but benign behavior.

---

[6] In response to the RFI question 1(d).
[7] In response to the RFI question 4(d).

Beyond resource drift, risk assessments should account for agent-specific behavioral risks. This includes the potential for "rogue agents" that may self-replicate, spawn unauthorized child agents, or intentionally diverge from their original goals to optimize unintended or malicious outcomes.

Adversarial Testing and Red Teaming at Scale[8]

Guidelines should be established for adversarial test suites, including practical attack scenarios grounded in real-world threat models. Standardization should focus on the testing methodology and reporting criteria, while the specific attack scenarios should be continuously updated and protected to prevent adversaries from weaponizing them. M^{3}AAWG members run continuous red team exercises; AI agents should face the same scrutiny.

The challenge is the emergent behavior. While traditional software has defined code paths, AI agents can find creative solutions, including new ways to circumvent security controls. Automated adversarial testing should explore the full behavior space, not just obvious attack vectors.

Logging and Forensics

When an incident or events of interest occur, reconstruction should be possible. Every agent decision needs clear logging, including (to the extent possible), the input that triggered it, the context it considered, the action it took, and the outcome. All logging should be handled as if it contains personally identifiable or sensitive information, in accordance with local privacy regulations. For investigative purposes, these logs should be tamper-evident (through cryptographic hashing and off-system storage) and retained in accordance with regulatory retention periods.

Decision Traceability for Audit and Compliance

In addition to supporting forensic investigation, logging for AI agents should, where possible, enable decision traceability for governance and compliance purposes. For actions involving sensitive data or high-impact outcomes, logs should support reconstruction of the policies evaluated, categories of data involved, and the criteria used to select a particular action over alternatives. Logging should also provide clear documentation of how and why decisions were made by the AI system.

This level of traceability and explainability is recommended for demonstrating due diligence, supporting regulatory inquiries, and enabling meaningful post-incident governance reviews. NIST can help guide how logging systems should be designed to ensure traceability of sensitive data, extending beyond operational debugging.

Defensive AI Red Teaming

[8] In response to the RFI question 3(a).

Attackers are already using AI to automate abuse at scale.[9] We recommend NIST include voluntary guidance on the use of AI for controlled red-team testing that defenders can leverage, helping prevent unnecessary regulatory burden. Organizations should simulate both normal customer behavior and AI-enabled attack patterns to avoid false positives and missed threats. For example, a customer may download a year of statements, while an automated attack may mimic the same workflow at an abnormal scale. Without AI-based simulation, systems cannot reliably distinguish between legitimate behavior and abuse.

Standards and guidelines should therefore promote defensive AI testing, workflow-based scenario evaluation, and strong governance that separates simulation from malicious use.

IV. Human Oversight and Control

Emergency Shutoff and Alerting[10]

Human oversight and control, including emergency shutoff protocols, should take precedence in cases where AI systems can cause significant risk or damage.

Alerting should trigger automatically when an agent exceeds predefined risk thresholds, including excessive failed actions, unexpected resource consumption, or access to restricted data. As such, guidelines for monitoring and alerting metrics should be established.

Protocols may include technical circuit breakers that immediately sever agent access to critical Application Programming Interfaces. Furthermore, we recommend that NIST encourage cross-agent validation requirements, where high-impact actions from one agent are verified by a secondary, independent security agent to prevent automated errors or malicious instructions from propagating through a network.

Transparency for Security[11]

Perfect interpretability may be unachievable, but sufficient transparency to audit security-relevant decisions is necessary. If an agent accesses sensitive data or takes a high-risk action, logging should include the steps taken and the inputs that triggered those actions. Where possible, a concrete chain of reasoning that a security analyst can evaluate should also be documented to support compliance requirements and forensic investigation.

Data Minimization and Privacy by Design

AI agents should be designed and governed to adhere to strict privacy-by-design principles, which begin with explicit requirements for data minimization. These include purpose limitation, ensuring that agents only collect,

---

[9] FBI San Francisco, "FBI Warns of Increasing Threat of Cyber Criminals Utilizing Artificial Intelligence," 05 2024. https://www.fbi.gov/contact-us/field-offices/sanfrancisco/news/fbi-warns-of-increasing-threat-of-cyber-criminals-utilizing-artificial-intelligence

[10] In response to the RFI question 4(b).

[11] In response to the RFI's question 3(b).

access, and use the minimum amount of personal or sensitive data necessary for a defined task. Furthermore, explicit, time-based retention limits should be established, tied to the agent's purpose and lifecycle, to ensure that data is not retained indefinitely after its utility has expired. These controls are foundational to building trust and reducing the overall privacy risk associated with autonomous systems.

Implementing these stringent data lifecycle controls is essential for aligning agent governance with leading global and national privacy frameworks. This proactive approach helps organizations harmonize globally, adhere to FTC guidelines, and align with common risk-based frameworks and requirements, e.g. NIST.

Defined Accountability for AI Agents

AI agents operating in production environments should have clearly identified human accountability (the human-in-the-loop principle). Each agent should have a designated owner responsible for approving its deployment, defining its operating scope, and overseeing its ongoing behavior. This accountability includes conducting initial risk assessment, authorizing permissions and integrations, reviewing monitoring signals and alerts, and exercising decision-making authority when circuit breakers or kill switches are triggered.

Absent clear accountability, even well-designed technical controls risk becoming ineffective in practice. Governance structures should ensure that decision rights and escalation paths are defined before AI agents are allowed to operate autonomously in environments with real-world consequences.

Accountability should also address the psychological risk of human manipulation. We recommend specialized training for human reviewers to recognize "persuasive but fraudulent" AI justifications, such as fake audit rationales or fabricated security logs, designed to trick humans into authorizing a malicious configuration change.

Separation of Duties for High-Risk AI Agents

For AI agents operating at elevated risk levels, such as those with broad system access, privileged credentials, or authority to execute irreversible actions, organizations should implement clear separation of duties, similar to segregated controls used in other high-risk technical lifecycle management processes. The teams responsible for developing or configuring an agent should not be solely responsible for approving its deployment or monitoring its behavior in production. Separating these functions reduces the risk of governance blind spots, ensures independent review of agent capabilities, and aligns AI agent oversight with established internal control practices used in other high-impact automated systems.

This principle also highlights the need for separate guidance for agent developers and deployers, as their roles involve different security concerns. For example, deployers may introduce security issues by caching predictions, while developers inherently have access to sensitive assets like model weights.

Agent Lifecycle Governance

Governance controls for AI agents should extend across the full system lifecycle, not just initial deployment, to prevent model drift or undetected performance deterioration. Organizations should establish governance checkpoints that include periodic reauthorization of agent permissions, reassessment of access scopes as tasks evolve, and formal review of behavioral drift relative to established benchmarks of the agent's original design intent.

Decommissioning procedures should be explicitly defined and enforced. These should include revocation of credentials, disabling of integrations, and secure retention of tamper-evident logs for an appropriate period to support audit and investigation needs.

Recommendations for NIST

We respectfully urge NIST to:

1. **Develop measurable security metrics for AI agents.** Move beyond aspirational principles to define concrete, testable criteria to address acceptable false positive rates for guardrails, minimum logging granularity, and the required isolation strength for different risk levels. These metrics should also include formal security analysis of current protocols, including the Model Context Protocol (MCP) and Agent2Agent (A2A).

2. **Prioritize the harmonization and consolidation of NIST publications** (e.g., AI RMF, AI CSF Profile, SSDF AI Profile, CAISI documents). Given the existing volume of AI content, consolidation would improve clarity and facilitate industry adoption.

3. **Reduce compliance fragmentation and promote international interoperability.** NIST guidance on AI agents should explicitly build upon and align with existing standards and risk-based frameworks, including the NIST AI Risk Management Framework, as well as broader cybersecurity frameworks and applicable global standards, including ISO/IEC/IEEE. It should also incorporate emerging industry-specific resources such as the OWASP Top 10 for Agentic Applications[12] to ensure practitioners have access to a unified set of threat models.

4. **Create risk-based security profiles for specific agentic use cases** similar to existing NIST frameworks. Requirements vary per deployment and system. For example, a simple chatbot has different requirements than an agent managing infrastructure, but each profile should have clearly defined controls.

[12] OWASP Gen AI Security Project, "OWASP Top 10 for Agentic Applications for 2026," 12 2025. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

5. **Support testing and certification guidelines**. Third-party validation would help, especially for smaller organizations that lack internal red team capabilities.

6. **Address the supply chain problem**[13]. Model repositories, training data sources, and fine-tuning services all introduce risk. Guidance for vendor assessment and ongoing monitoring is needed.

7. **Develop a Risk-Tiered Agent Governance framework.**[14] This risk-tiered classification framework for AI agents should align with other global standards, e.g., ISO/IEC, using the risk-based approach. Risk tiers should account for factors such as autonomy level, data sensitivity, action impact, and external exposure, as well as the likelihood and severity of harm to individuals resulting from agent outputs. Governance requirements, including approval thresholds, monitoring rigor, logging granularity, and human-in-the-loop controls, should scale with these risk tiers. Security profiles alone are insufficient without corresponding governance requirements that clarify oversight expectations and accountability as agent capabilities increase.

8. **Create a framework for AI Agent Incident Response and Reporting.**[15] AI agents should be explicitly integrated into organizational incident response frameworks. NIST guidance should emphasize the need for predefined severity levels for agent-related incidents, clear internal notification thresholds, and structured post-incident reviews that evaluate the adequacy of both technical controls and governance decisions. For AI agents operating in regulated or safety-critical contexts, material incidents involving AI agents should trigger formal reporting and corrective action processes comparable to those used for other critical system failures.

9. **Continue to develop and maintain user-friendly tools and solutions for testing AI security**. There is a need for open-source testing tools developed in collaboration between government and the broader AI ecosystem. These tools should provide baseline testing attestation of AI agents. Currently, the testing ecosystem is closed and largely proprietary.

10. **Provide explicit guidance on defensive AI simulation.** Adversaries already use AI to scale attacks, and defenders should test against realistic AI-generated behavior to detect abuse without disrupting legitimate users. In industries such as finance, this means distinguishing between normal automated activity and adversarial automation, which requires controlled AI-based modeling of both. We recommend that NIST guidance incorporate defensive AI testing, workflow-based scenario evaluation, and governance that clearly separate simulation from harmful activity. Defending against AI-enabled threats requires AI-enabled defenses because attackers do not operate within the same constraints.

[13] In response to the RFI question 3(b).
[14] In response to the RFI question 4(d).
[15] In response to the RFI question 3(a).

11. **Develop explicit guidance to address inter-agent communication**. We recommend that NIST include protocols for message authentication, identity verification, and encryption for all inter-agent traffic to mitigate the confused deputy problem. This ensures that low-privilege agents cannot spoof their identities or inject unauthorized instructions into the workflow of high-privilege agents, maintaining strict privilege separation across the agent ecosystem.

Research Approaches for Security Prioritization

In response to the RFI question, "Which research approaches should be prioritized to advance the scientific understanding and mitigation of security threats, risks, and vulnerabilities affecting AI agent systems?", M^{3}AAWG recommends the following areas of focus:

Model and Data Practices: Prioritize research into model, input data practices (redaction, restricting input size) and access control mechanisms.

Constraint Management Testing: Focus on developing methods to test constraint management—how well AI agents can abide by enforced constraints in their reasoning—and how to measure this performance in real time.

Tool-Chaining Sequence Analysis: Investigate an agent's tool-chaining sequence to determine whether available resources and capabilities could be combined in unanticipated ways and create security vulnerabilities.

Deployment Modifications: Research the security implications of modifications that deployers make to software acquired out of the box.

Testing and Evaluation: Develop trustworthy, reliable, repeatable, and transparent testing and evaluation methods and ensure documentation and data availability are aligned to their intended purpose.

M^{3}AAWG members have spent years learning hard lessons about securing complex, automated systems at scale. We have seen what happens when security is an afterthought. AI agents are powerful, but with that power comes responsibility.

We appreciate the opportunity to submit feedback and welcome further engagement as needed to answer any questions during this process. Please address any inquiries to M^{3}AAWG Executive Director Amy Cadagin at comments@m3aawg.org.

Sincerely,

Amy Cadagin
Executive Director
Messaging, Malware and Mobile Anti-Abuse Working Group (M3AAWG)
comments@m3aawg.org
P.O. Box 9125, Brea, CA 92822

Submission Date: March 6, 2026