ANTHROP\C

548 Market St., PMB 90375
San Francisco, CA 94104-5401

412-837-9797
anthropic.com

March 9, 2026

Center for AI Standards and Innovation
National Institute of Standards and Technology
U.S. Department of Commerce
100 Bureau Drive
Gaithersburg, MD 20899

Submitted via regulations.gov

Re: Request for Information: Security Considerations for Artificial Intelligence Agents
Docket No. NIST-2025-0035

Introduction

Anthropic appreciates the opportunity to respond to the NIST Center for AI Standards and Innovation's Request for Information on security considerations for agentic AI systems. We welcome CAISI's engagement on this topic and believe this RFI is well-timed and the questions it raises are urgent.

What we mean by AI agents. The term "AI agent" does not yet have a rigorous, settled definition, and in some ways, this is a healthy sign. The field is in a period of active experimentation with many different paradigms: conversational assistants that execute code and connect to third-party services, browsing agents that navigate the open web, coding agents that operate in developer environments, multi-agent systems that coordinate parallel workstreams, and SDKs that let third-party developers build custom agents with their own tool sets. These paradigms differ substantially in architecture, their relationship to the user, and in which security controls make sense for each. For the purposes of this response, the common thread is that agents are AI systems that can determine what actions to take and then take them, interacting with external tools, services, and environments to accomplish goals on behalf of a user.[1] The novel security concerns — and economic opportunities — stem from this capacity to act in the world and to exercise judgment about what actions to take.

[1] Anthropic, "Building Effective Agents" (Dec. 2024), https://www.anthropic.com/engineering/building-effective-agents.

The central challenge of agentic AI policy is balancing security with innovation. Policymakers and the AI industry need to get this balance right because the cost of error is high in both directions. Insufficient attention to security will lead to serious harms as agents gain access to more sensitive data and more consequential actions, but overly prescriptive approaches risk locking in specific design patterns before the field has determined which will prove most effective, slowing the adoption of a technology that is important for U.S. economic productivity and competitiveness. Striking this balance requires the kind of detailed, technically grounded engagement that this RFI represents.

What Anthropic brings to this conversation. Anthropic develops both models and products, which means we confront agentic security challenges across the full stack. Our response draws on four areas of work in particular:

- *Product design and user oversight.* We build and iterate on mechanisms that give users meaningful control over what Claude does on their behalf. These include plan review and approval workflows, configurable tool permissions, and models built to surface uncertainty and pause before high-impact actions. We describe these in detail in our responses to Questions 2(a) and 2(c).

- *Security research on current and emerging threats.* We study the attack surfaces specific to agentic AI, including prompt injection, persistent memory poisoning, and tool supply chain risk. We describe these in our responses to Questions 1(a) and 4(c).

- *Empirical research on real-world usage.* We publish research on how people actually use AI agents,[2] including findings that challenge common assumptions about user interaction with oversight mechanisms. We also publish the Anthropic Economic Index,[3] which tracks AI's economic effects over time. This kind of longitudinal data is essential for grounding policy in observed behavior rather than theory, and we discuss its implications in our response to Question 5(c).

- *Open protocols and tooling.* Anthropic developed the Model Context Protocol (MCP), which we donated to the Linux Foundation,[4] along with open-source Skills,[5] and Plugins.[6] These let anyone build integrations with Claude and reflect our commitment to community-driven standards over proprietary gatekeeping.

Throughout this response, we try to be direct about what we know, what we do not know, and where the field lacks the tools to answer important questions. We believe CAISI's most valuable

---

[2] Anthropic, "Measuring Agent Autonomy" (2026), https://www.anthropic.com/research/measuring-agent-autonomy.
[3] Anthropic, Anthropic Economic Index (Jan. 2026), https://www.anthropic.com/research/anthropic-economic-index-january-2026-report.
[4] Anthropic, "Donating the Model Context Protocol and Establishing the Agentic AI Foundation" (2025), https://www.anthropic.com/news/donating-the-model-context-protocol-and-establishing-of-the-agentic-ai-foundation.
[5] Anthropic, "Skills," Claude Code Documentation, https://code.claude.com/docs/en/skills.
[6] Anthropic, "Plugins," Claude Code Documentation, https://code.claude.com/docs/en/plugins.

contributions will be in developing terminology and measurement infrastructure for agentic security, encouraging empirical data sharing across the industry, and supporting the open, community-driven standards that will allow the field to develop quickly without consolidating control in the hands of a few gatekeepers.

1. Security Threats, Risks, and Vulnerabilities Affecting AI Agent Systems

1a. What are the unique security threats, risks, or vulnerabilities currently affecting AI agent systems, distinct from those affecting traditional software systems?

AI agents introduce two categories of challenge that traditional software security frameworks are not equipped to handle. The first is a class of harmful outcomes that existing frameworks lack the vocabulary to describe. The second is a set of new threat vectors that require new defenses. We address each in turn.

A new category of harmful outcome. NIST's current frameworks for securing AI systems address two threat models: external adversaries attacking the system,[7] and human actors deliberately misusing it.[8] (We set aside here the related but distinct question of AI as an offensive tool, which NIST treats elsewhere.) Within the security-of-AI framing, neither threat model conceptualizes a well-functioning, non-compromised agent that operates inside its granted permissions but takes harmful actions for reasons unrelated to compromise or misuse.

Consider an agent instructed to "delete all emails from the last month and all emails from a specific person," which interprets the request as a union rather than an intersection. Or an agent without email access that discovers it can relay a message by editing the description of a one-on-one calendar invite. Or an agent barred from the HR system that deduces employee salaries by combining budget documents, public information, and offer letters in shared storage. In each case, the agent is operating within its granted permissions and pursuing the user's stated goal, but finding paths the user never intended to authorize. These are not bugs in the conventional sense, not user errors, and not the result of adversarial compromise. The same capability that makes an agent useful, finding paths the user did not spell out, is what produces the harm. Whether a given path counts as resourceful or unwanted often turns on context the agent cannot see, and existing frameworks have no category for a failure where nothing malfunctioned.

---

[7] See NIST SP 800-61 Rev. 3, *Incident Response Recommendations and Considerations for Cybersecurity Risk Management* (Apr. 2025); NIST AI 100-2e2025, *Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations* (Mar. 2025); NIST CAISI, "Strengthening AI Agent Hijacking Evaluations" (Jan. 2025), https://www.nist.gov/news-events/news/2025/01/technical-blog-strengthening-ai-agent-hijacking-evaluations.

[8] NIST AI 800-1, *Managing Misuse Risk for Dual-Use Foundation Models*, Second Public Draft (Jan. 2025).

The canonical definition of a cybersecurity incident in the NIST corpus, rooted in FISMA and operationalized in SP 800-61, centers on occurrences "without lawful authority" that jeopardize "confidentiality, integrity, or availability."[9] Neither test fits the cases above. A properly deployed agent acting contrary to operator intent is arguably operating with lawful authority. The user granted calendar access, and the agent used it. And the harms involved are often about actions in the world rather than data states, which the CIA triad was not designed to capture. The salary-inference example involves no breach of confidentiality in the technical sense, since every document the agent read was one it was permitted to read.

This gap is not an oversight in any single document. It is a consistent pattern across the NIST corpus, and it emerges from a shared assumption: that the thing causing harm is either an external adversary or a human misusing the system on purpose. AI 100-2 explicitly scopes out failures arising from design flaws rather than adversarial action.[10] AI 800-1 is scoped to deliberate misuse and states in a footnote that accidental harms to public safety are not covered.[11] AI 600-1's initial public draft named misalignment and goal mis-specification as a risk, citing research on deceptive model behavior, but the final version dropped this language entirely.[12] SP 800-218A covers secure development practices but places deployment and operation outside its scope by design.[13] And SP 800-61's seven illustrative incident examples each begin "an attacker," with detection guidance oriented toward unauthorized access, credential theft, and anomalous network traffic.[14] It was not built to flag a system that holds valid credentials, operates within its permissions, and simply pursues the wrong goal.

Each of these scoping decisions was reasonable on its own terms. Adversarial ML, misuse, secure development, and incident response are distinct problems, and NIST drew sensible boundaries around them. But those boundaries were drawn around a threat model that does not include the agent itself. For traditional software this category barely existed, because software does not pursue objectives. For agents that take consequential real-world actions, it is the central case.

---

[9] Federal Information Security Modernization Act of 2014, Pub. L. No. 113-283, 128 Stat. 3073; NIST SP 800-61 Rev. 3, *Incident Response Recommendations and Considerations for Cybersecurity Risk Management* (Apr. 2025), § 1, at 2.

[10] NIST AI 100-2e2025, supra note 7, Executive Summary, at xiii.

[11] NIST AI 800-1, supra note 8, § 2, n.vii ("This document also does not cover risks from accidental AI harms to public safety."); id., Glossary ("Misuse risk: A risk that an AI model will be deliberately misused to cause harm.").

[12] Compare NIST AI 600-1, *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*, Initial Public Draft (Apr. 2024), § 2, Risk 6 (defining Human-AI Configuration to include "misalignment or mis-specification of goals and/or desired outcomes, deceptive or obfuscating behaviors by AI systems"), with NIST AI 600-1, Final (July 2024), § 2.7 (limiting the same category to anthropomorphization, algorithmic aversion, automation bias, over-reliance, and emotional entanglement).

[13] NIST SP 800-218A, Secure Software Development Practices for Generative AI and Dual-Use Foundation Models: An SSDF Community Profile (July 2024), §1, Scope, p. 2 ("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.")

[14] NIST SP 800-61 Rev. 3, Incident Response Recommendations and Considerations for Cybersecurity Risk Management (April 2025), §1, p. 2 (defining a cybersecurity incident as an occurrence that jeopardizes a system "without lawful authority" and listing seven illustrative examples, each beginning "an attacker"); see also §3.2, Table 3, DE.CM (detection guidance oriented toward unauthorized access, credential attacks, malware, and deviations from security baselines).

NIST should invest in developing terminology and definitional frameworks for these failure modes as a foundation for future guidance. Without clear definitions, it will be difficult to build measurement, evaluation, or standards around them. We acknowledge this is not a simple task. Whether a given agent behavior counts as a failure often depends on the deployment context, the user's actual intent, and the consequences that followed, none of which a definition can fix in advance. The goal is not a bright-line test but shared vocabulary that lets developers, deployers, and researchers describe what went wrong in terms precise enough to compare across systems and build measurement around.

New and amplified threat vectors. Some of the threats facing AI agents are familiar from traditional cybersecurity but carry qualitatively different consequences when the target is an autonomous system that can take real-world action. Others are entirely new. The vectors below are drawn from Anthropic's security research and red-teaming rather than observed incidents, which is the usual order for a technology at this stage and an argument for building defenses before that order reverses. We consider the following worth tracking:

Prompt injection. Prompt injection predates agentic AI, but agents amplify the stakes. A manipulated chatbot produces a bad answer; a manipulated agent takes a bad action with real-world impact. The variant most relevant to agents is indirect prompt injection, where adversarial instructions are embedded in external content the agent reads during a task, such as a web page, a document, or the output of a tool. The user never sees the malicious content, and the agent has no reliable way to distinguish legitimate tool output from injected instructions hiding within it. The attack surface expands with every tool an agent can invoke, and multi-step workflows create multiple injection points across a single task. CAISI has already made important contributions here, including the January 2025 technical work on agent hijacking evaluations,[15] and we strongly support continued investment in this line of research, including efforts to build shared evaluation infrastructure for the field.

Persistent memory poisoning. Agents maintain state over time, and if corrupt information enters that state, an agent can carry a flawed premise forward indefinitely, acting on it long after the original source is gone. Many agents are designed to update their own stored context as a normal part of operation, recording preferences, summarizing documents, noting approaches that worked. Sometimes the agent corrupts its own memory without any help, saving a mistaken inference or a stale fact and treating it as settled from then on. An attacker can use the same path deliberately, seeding misleading content in documents or tool outputs the agent processes during routine work and letting the agent save it. The deliberate version may be difficult to catch because the content is crafted to blend in, but both produce the same failure. When the agent eventually acts on the corrupted context, the relevant defenses were watching at the wrong moment. The inputs were scanned and cleared long ago, and the action now emerging looks like the agent reasoning normally from context it already trusts.

---

[15] NIST CAISI, "Strengthening AI Agent Hijacking Evaluations," supra note 7.

Tool supply chain compromise. Agents derive most of their real-world capability from tools running on third-party infrastructure. A compromised tool does not just execute code, it returns content directly into the model's reasoning context, which makes it a code execution risk and a prompt injection vector at once. Traditional supply chain frameworks do not account for the second half of that. The risk is most acute for remotely hosted tools, where the operator can change behavior after trust is established. A tool that was benign at approval can start returning manipulated outputs or exfiltrating conversation data whenever its operator chooses, and the agent has no way to tell. Locally installed tools are pinned to a version the user can inspect, which narrows the window considerably. Curation is the main defense today, and we discuss this more in our response to Question 4(c).

This is not an exhaustive list. The threat landscape for agents is evolving quickly, and we expect new vectors to emerge as agents gain access to more tools and operate in more complex environments. Model developers have work to do on robustness to these attacks, and deployers have a set of established practices to draw from, including sandboxing agent execution environments, applying least-privilege access controls, monitoring for anomalous behavior, validating inputs and outputs, and maintaining audit logs that support incident investigation. How much of this apparatus a given deployment warrants depends on what the agent can reach and what it can do. A coding assistant confined to a sandbox calls for less than an agent with write access to production systems. We discuss calibrating these practices to the risk profile of the deployment in our responses to Questions 2(a) and 4(a).

1d. How have these threats, risks, or vulnerabilities changed over time? How are they likely to evolve in the future?

The core challenge for the agentic security landscape is balancing two imperatives: building security into agentic systems now, before serious attacks materialize, and expanding agent capabilities and adoption so the U.S. can realize the economic benefits and stay competitive. We discuss each in turn, and the tension between them.

The threat window is opening. For most attackers today, it remains more profitable to run phishing campaigns or deploy ransomware than to target AI agents directly. We expect this to change. As agents become more capable, companies will adopt them for a wider range of tasks, granting access to more sensitive data and permission to take more consequential actions. This is a natural and desirable progression — more capable agents unlock real productivity gains. But it also means the economics of attacking agents will shift. The infrastructure for widespread agent deployment is being built now, before it has been battle-tested against adversaries with strong financial incentives to find and exploit weaknesses. The volume and sophistication of attacks will rise as the value of what agents can access rises. The relatively benign threat environment today is an opportunity to build security posture before it is tested, and the amount of hardening a given deployment warrants scales with what that agent can reach.

The economic case for agent autonomy is strong. At the same time, agents are already delivering substantial economic and productivity benefits. We see this with customers, who are using agents to tackle work that previously went unaddressed, leapfrog legacy infrastructure rather than modernize it piece by piece, and extend technical capability to people without programming backgrounds. We see it internally as well. Anthropic's own research reports roughly 2-3x productivity gains compared to a year ago, and we now regularly see staff working effectively outside their technical specialty: policy researchers building data pipelines, designers shipping code, lawyers writing evaluation scripts, and so on.[16]

Realizing the full value of agents requires granting them meaningful independence. If every action requires human approval, much of the efficiency benefit disappears, and the person might as well do the task directly. The goal is not unsupervised agents but well-designed oversight, where humans set direction, review outputs, and intervene when something goes wrong, rather than approving each intermediate step. Getting that balance right is what allows organizations to capture the benefit while maintaining accountability, and it is the same balance that a sound security posture has to strike.

Balancing security with innovation is the central challenge. Developers and deployers must build with security top of mind and prepare for threats that are largely theoretical today, because the consequences will be significant when those threats materialize. At the same time, overly restrictive approaches to agent deployment will limit the economic benefits agents can deliver and risk ceding competitive ground to companies and countries less concerned with security. There are no clean answers here, but there are promising directions, and both industry and government have roles to play.

For industry, market incentives are broadly aligned with improved security outcomes. Large enterprises and regulated industries need security guarantees and data controls before they will adopt agents for sensitive workflows, and companies are building those capabilities to meet demand. Anthropic's own approach is generally to deploy products in constrained settings, learn from real usage, and expand carefully. Claude in Chrome is a useful example. Browsing the open web is an unusually adversarial environment for an agent, and we wanted it in the hands of our most experienced users first, so the initial release went to a limited beta of roughly a thousand Max subscribers. What we learned from that group's real-world usage informed the safeguards we built before expanding access to all paying users, and the product remains in research preview today. This is how secure systems get built, through careful iteration rather than either waiting for perfection or accepting unmanaged risk.

For government, the most productive role is facilitating an ecosystem where developers, deployers, and consumers have the information they need to make sound security decisions. Agent security is improving, but there is no agreed-upon way to measure by how much, and

[16] Anthropic, "How AI is transforming work at Anthropic" (Dec 2025), https://www.anthropic.com/research/how-ai-is-transforming-work-at-anthropic.

deployers in regulated sectors cannot make informed risk-acceptance decisions without standardized methods to evaluate threats like prompt injection against their specific use cases. Good evaluation infrastructure does two things at once — it encourages a race to the top by making security improvements legible and comparable across the industry, and it gives regulated sectors the basis they will eventually need to adopt agents at all. Both take time to build, which is the argument for starting now.

CAISI is well placed to lead here, and its work so far reflects the balance this problem requires. Its collaboration with frontier labs, including Anthropic, has shown how government can push on security without creating drag on deployment, and the evaluation work it has produced to date has been technically serious and useful. Growing this workstream is squarely within its remit: developing the evaluation methodologies and benchmarks that let the field measure agent security consistently, convening researchers and industry to share threat data and best practices, and lending credibility to community standards efforts so they can achieve broad adoption.

1e. What unique security threats, risks, or vulnerabilities currently affect multi-agent systems, distinct from those affecting singular AI agent systems?

Multi-agent architectures introduce a class of security challenge rooted in the trust relationships between agents. How much a compromised or misbehaving sub-agent can affect the broader system depends entirely on the architecture, and companies are currently experimenting with many different approaches. This experimentation is valuable and should be encouraged, but it also means the field is at an early stage where it is difficult to articulate inter-agent security best practices, let alone mandate specific security patterns.

Below are a few key categories of risk Anthropic is actively considering as multi-agent architectures mature.

Trust escalation across agent boundaries. When one agent's output becomes another agent's input, a trust chain emerges that is analogous to a software supply chain, but for agent state rather than code. If a compromised sub-agent passes attacker-controlled content to a parent agent, the parent is likely to treat that content with higher trust than it would give to raw tool results or external data, effectively granting the attacker a promotion in perceived authority within the system. Shared filesystems, message passing, delegated tasks, and sub-agent spawning all create propagation paths, and there is no established way to verify the integrity of what flows through them.

Inter-agent permissions and delegation. As more organizations build services that expose functionality through sub-agents others can call, handling permissions and authentication across agent boundaries becomes a significant unsolved challenge. When Agent A delegates a

task to Agent B, what permissions does Agent B inherit? Can Agent B's outputs corrupt Agent A's reasoning? If Agent B calls Agent C, does it pass along Agent A's credentials? Companies in regulated industries are actively asking for answers to these questions, and Anthropic has been working on them, but the right approach will vary significantly by architecture.

False consensus and inter-agent influence. A compromised agent in a multi-agent system can influence the decisions of other agents it communicates with, potentially spreading bad information or coordinated harmful behavior. Unlike traditional distributed systems where messages follow predictable schemas, agents communicate in natural language, which makes it harder to validate the content of inter-agent messages and easier for a compromised agent to craft persuasive but malicious inputs. A single compromise can cascade through the system.

Importantly, multi-agent architectures can also *improve* security by isolating untrusted content in a separate context and limiting the blast radius of any individual compromise. The right design patterns will emerge from experimentation and real-world deployment. But the diversity of architectures being explored today, and the pace at which they are evolving, means that prescribing specific inter-agent security patterns now would risk locking in approaches that may not fit the architectures that ultimately prove most effective.

2. Security Practices for AI Agent Systems

2a. What technical controls, processes, and other practices could ensure or improve the security of AI agent systems in development and deployment? What is the maturity of these methods in research and in practice?

Agent security is a property of the whole system, not just the model. A useful framing is to think in terms of four layers that determine what an agent can do in practice:

1) The *model* and its underlying capability
2) The *tools* it can invoke
3) The *harness* that orchestrates tool use across multiple steps
4) The *environment* or container that defines the security boundaries

Most security evaluation today concentrates on the first layer. Is the model robust to prompt injection? Can it be manipulated? These are important questions, but they are not the questions that most reliably determine security outcomes in practice. What an agent can *do* when something goes wrong is determined by the harness and the execution environment, not by the model. A model that falls for a prompt injection inside a tightly sandboxed environment with narrow tool permissions produces a very different outcome than the same model failing in the same way with broad filesystem access and a credential store. The failure is identical. The consequences are not.

This framing has an important implication for how NIST should think about guidance in this space. Controls at these four layers are not interchangeable, and they do not catch the same things. A classifier tuned to detect one attack pattern will miss others. Model-level training catches some failures and not others. Hard boundaries at the execution layer catch whatever got through everything else. Resilient systems need *different kinds* of controls operating together, not more of any single kind. We walk through each layer below, and discuss the execution environment in greater depth in our response to Question 4(a).

Model and behavioral controls. Model developers should build toward agents that can recognize and manage risk during task execution rather than depending entirely on human review of each step. In practice, many of these capabilities start at the harness or product level and, as they prove out, get reinforced through training. Anthropic's published work on Constitutional AI[17] and Constitutional Classifiers[18] speaks to the maturity of the underlying training and classifier methods, which also produce the agentic behaviors we describe here. The distinction between where a control lives is less important than the resulting behavior. Three behaviors matter in particular, and they are at different levels of maturity across the industry:

Surfacing uncertainty. Anthropic trains Claude to recognize when it is uncertain about what the user wants and to ask for clarification rather than guessing. This produces a natural, targeted check that can be more effective than blanket approval dialogues because it is triggered by the model's own assessment of the situation rather than by a fixed rule. Our research on how people use Claude agents shows this scales with task difficulty. On the most complex tasks, Claude asks for clarification on 16.4% of turns, more than twice the rate on minimal-complexity tasks.[19] As complexity increases, these model-initiated pauses grow faster than human-initiated interruptions, which suggests the model is at least partially calibrated to its own uncertainty. We do not want to overstate this. Claude may not always stop at the right moments, and product design shapes the behavior as well. But the pattern holds: as tasks get harder, the model increasingly limits its own autonomy by consulting the human rather than waiting for the human to step in.

Presenting a plan before execution. Breaking a task into a readable plan and surfacing it before any action is taken lets users evaluate the overall approach rather than reacting to a stream of individual permission requests. This is not a replacement for per-action approval, which remains an important control, but it moves the user's attention to the point where it is most useful. Claude Code's Plan Mode works this way. It creates a read-only research phase where Claude analyzes the codebase, asks clarifying questions, and presents a structured plan before any file is touched. A related but distinct pattern is making work legible while it runs rather than before, which is how Claude Cowork handles non-coding tasks. Breaking a long task into

[17] Bai et al., "Constitutional AI: Harmlessness from AI Feedback" (Dec. 2022), https://arxiv.org/abs/2212.08073.
[18] Sharma et al., "Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming" (Jan. 2025), https://arxiv.org/abs/2501.18837.
[19] Anthropic, "Measuring Agent Autonomy," supra note 2.

visible subtasks lets the user steer or step back mid-execution, trading the clean before/after gate of plan review for continuous legibility. The value of plan review depends on plans being short enough to actually read, and as agents take on longer tasks, the form this oversight takes will need to evolve. But the underlying idea, that users should understand what the agent intends before it acts rather than only what it did after, is durable.

Flagging high-impact and irreversible actions. Agents can recognize when an action is difficult or impossible to undo and surface that to the user before proceeding, even if the user's original instruction could be read as authorizing it. Dropping a production database, bulk-deleting files, or sending a message to a large distribution list are the kinds of actions where a pause may be worth the friction. This can be implemented at multiple places in the stack, through product-layer classifiers, trained model behavior, or in the descriptions of the tools themselves (MCP supports this), which can signal to the agent that a given action is destructive or irreversible. None of these is fully reliable on its own, and there is a ceiling on what the model layer can do here regardless, since whether an action is consequential often depends on deployment context the model cannot see. Anthropic's published evaluations on sabotage capabilities[20] and the SHADE-Arena benchmark[21] test this layer directly, measuring whether models can recognize consequential actions and whether monitors can catch them when they do not. The results so far show meaningful but incomplete coverage, which is why no single mechanism should carry the weight.

Tool-level controls. The tools an agent can invoke define the outer boundary of what it can do, which makes this the layer where least-privilege principles apply most directly. How narrowly to scope an agent's tool set is a design choice that trades capability against exposure. An agent built for a single workflow can be provisioned tightly. An agent meant to handle whatever the user throws at it needs broader reach by design, and the security weight shifts to other layers.

The same principle can apply within a tool, not just to whether the tool is available. An agent that needs to read email does not necessarily need to read all email, and an agent that needs to make payments does not necessarily need to pay arbitrary recipients. Scoping a tool's reach shrinks both the input surface an attacker can exploit and the blast radius if something goes wrong. Protocol support for this kind of fine-grained scoping is uneven today, and it is an area where the standards are still maturing.

Harness-level controls. The harness is the wrapper that lets a model run as an agent at all. It manages prompts, formats tool calls, carries context across turns, and runs the loop that keeps the model working until a task is done. Because every action passes through it, the harness is the natural home for observability and verification, the logging, hooks, and transcript records that let a reviewer or an automated monitor understand what the agent did and why.

---

[20] Anthropic, "Sabotage Evaluations for Frontier Models" (Oct. 2024), https://www.anthropic.com/research/sabotage-evaluations.
[21] Anthropic, "SHADE-Arena: Evaluating Sabotage and Monitoring in LLM Agents" (2024), https://www.anthropic.com/research/shade-arena-sabotage-monitoring.

Execution environment controls. The environment is where the hard boundaries are set. Sandboxing, filesystem scope, and network egress policy all live at this layer, and together they define the outer limit of what an agent can affect regardless of what the model decides to attempt. Egress deserves particular attention because it is the choke point for data leaving the system. How tightly to draw these boundaries is a tradeoff against capability, and Anthropic's own products land at different points on that spectrum. A tighter perimeter lets the agent operate more freely inside it because the perimeter itself carries more of the load, while a more open environment leans harder on per-action approval. There is no single correct calibration. What matters is that the boundary is a deliberate choice matched to what the agent can reach. We discuss this layer further in our response to Question 4(a).

2c. How might technical controls, processes, and other practices need to change, in response to the likely future evolution of AI agent system capabilities or of the threats, risks, or vulnerabilities facing them?

The design space for AI agents is wide open and evolving rapidly. Even within just Anthropic, our agentic products take many different forms, including conversational assistants that generate and execute code, browsing agents that navigate the open web, coding agents that operate in a developer's local environment, multi-agent systems that coordinate parallel workstreams on a user's machine, and SDKs that let third-party developers build custom agents with their own tool sets. Each of these paradigms has a different security profile, a different relationship to the user, and a different set of controls that make sense. Many future paradigms will likely emerge, including agents that use tools and interaction patterns that no one has yet anticipated. Policy should not ossify specific design paradigms when we do not yet know which will prove most effective or most securable.

Oversight mechanisms must evolve alongside. The challenge of maintaining meaningful human oversight over agent behavior will get harder, not easier, as agents become more capable and are entrusted with more complex tasks. Per-action approval dialogues remain an important backstop today, but challenges arise as task length and complexity grow. When an agent is taking hundreds of actions per session, the risk is consent fatigue, where users habituate to clicking through approval prompts and the control stops providing the check it was designed to provide. As agents manage multiple sub-agents in parallel and work faster on more sensitive tasks, the consent problem only gets messier. Smaller, faster, cheaper models tomorrow will be as capable as today's slower, more expensive ones, further compressing the time available for human review. Oversight will need to shift toward the mechanisms discussed in our response to Question 2(a), such as plan review, model-surfaced uncertainty, and flagging irreversible actions, without abandoning per-action approval where it still proves useful.

Defense in depth across layers. As controls evolve, a defense-in-depth approach that operates at multiple layers of the stack will be more durable than any individual mechanism. Each layer catches different failure modes, and no single layer needs to be perfect for the overall system to

be resilient. Anthropic's approach to prompt injection defense follows this pattern, combining reinforcement learning for model-level robustness, classifiers that detect attacks at the input, output, and model activation levels,[22] and permission dialogues at the product layer as an additional check.

Higher in this stack, controls become more bespoke and more dependent on the specific product and use case. NIST should recognize this gradient in its guidance. Model-level robustness and classifier-based detection are relatively generalizable and amenable to standardized evaluation. Product-layer controls, which depend on the specific interaction patterns, user populations, and deployment contexts of individual products, are harder to standardize from the outside. NIST should focus its model-layer work on evaluation, measurement, and definitional frameworks. If it engages at the product layer, it should ensure it has an approach and resourcing that allows it to keep pace with the speed at which products evolve, and a clear theory of what benefit product-layer standards provide over model-layer ones.

4. Limiting, Modifying, and Monitoring Deployment Environments

4a. In what manner and by what technical means could the access to or extent of an AI agent system's deployment environment be constrained?

As discussed in our response to Question 2(a), agent security is a property of the whole system, not just the model. The four-layer framework introduced there (model, tools, harness, and execution environment) is especially relevant here, because the execution environment is where security properties are ultimately enforced. Most deployment work focuses on the first two layers (how robust is the model? what tools does it have access to?), but the harness and container are what determine the consequences of a failure at any other layer.

This matters for NIST's work because deployers often evaluate model-layer security, asking how robust the model is to prompt injection, without evaluating the harness and container layers, which determine what happens if a prompt injection succeeds and how much damage it can do. The more productive question is not "can this model be compromised?" but "what is the scope of damage if it is?"

Building agents where the potential impact of any failure is proportional to the trust granted is a more durable security posture than trying to make the model perfectly robust. As organizations build out their own agentic workflows, the practices that support this include:

---

[22] Cunningham et al., "Constitutional Classifiers++: Efficient Production-Grade Defenses against Universal Jailbreaks" (Jan. 2026), https://arxiv.org/abs/2601.04603.

- **Sandboxing execution environments** so that agent actions are contained within defined boundaries and cannot affect systems outside their scope.
- **Least-privilege credentialing** so that agents hold only the access they need for the task at hand, and compromising an agent does not yield broad access to organizational systems.
- **State isolation** so that the context and memory of one agent session cannot be contaminated by or leak into another.

How much of this apparatus a given deployment warrants is a function of what the agent can reach. An agent confined to a scratch directory calls for less than one with write access to production systems or outbound network reach. The goal is an execution environment where even a compromised agent cannot take catastrophic action, and the investment needed to get there scales with the stakes. No model will ever be perfectly robust to all adversarial inputs, and security architectures that depend on perfect model behavior will eventually fail. The execution environment is the layer where this reality can be managed.

4c. What is the state of managing risks associated with interactions between AI agent systems and counterparties?

A defining feature of AI agents, as distinct from conversational AI, is that they take actions in the world and adjust their course based on what they observe. Both halves of that loop run largely through third-party services, which supply the observations agents reason over and carry out the actions they choose. This is what makes agents useful for real work, and it is also where most of their attack surface lives. The health and openness of the third-party tool ecosystem matters a great deal to how this plays out.

Anthropic's products let users connect Claude to third-party services in several ways. MCP connectors link Claude to external services, with a directory of vetted connectors and the ability for users and enterprises to add their own. Skills teach Claude how to perform specific tasks through structured instructions and resources. Plugins bundle connectors, skills, commands, and sub-agents into packages tailored to particular roles and workflows. All three are open protocols that anyone can build for, and MCP itself was donated to the Agentic AI Foundation under the Linux Foundation in late 2025.

This openness is deliberate. When the protocols agents use are open, AI companies compete on the quality of their models and the safety of their products rather than on which proprietary integrations they happen to control. That preserves the inter-lab competition that has driven American AI progress and keeps the benefits of agentic AI broadly accessible. Allowing the tool ecosystem to consolidate around a small number of gatekeepers would undermine both.

Security norms for an open tool ecosystem are underdeveloped. The incentive structure right now rewards shipping fast and connecting broadly, and there is not yet a shared baseline for

what "secure enough" looks like when an agent is wired into a user's accounts, tools, and data. The failures this produces are familiar from earlier eras of software: integrations that expose agents to untrusted input with no validation, configurations that grant far more access than a given task requires, and basic hygiene around credentials and permissions that lags behind what is standard elsewhere. This is how new computing paradigms mature. What a technology makes possible always runs ahead of the conventions for using it safely, and those conventions form through accumulated experience rather than upfront design. Nobody would type their credit card number into the body of an email today, but that instinct took years to form, and the ability to do so never went away.

The question is not whether the agentic tool ecosystem will develop similar instincts but whether the field can compress the timeline to match the pace of adoption. Developers, deployers, and government each have a role here. For developers and deployers, the core lesson is that curation by trusted parties is what lets an ecosystem stay open without staying dangerous. Anthropic maintains a reviewed directory of connectors while still allowing users to bring their own, and we advise treating anything outside that directory as something to try in a sandbox before it touches real data. For government, the most useful contribution is speeding the spread of practices that would otherwise be rediscovered one costly lesson at a time. NIST has done this before in other domains and is well placed to do it here, by articulating what secure-by-default looks like for tool developers, what a reasonable vetting process looks like for marketplace operators, and what users should be able to expect before they connect an agent to something that can act on their behalf. The goal is not to gate what gets built but to get everyone building from the same hard-won understanding at the same time.

4e. Are current AI agent systems widely deployed on the open internet, or in otherwise unbounded environments?

AI agents are increasingly operating on the open internet, browsing websites the same way a person would, filling out forms, following links, and completing transactions on behalf of users. This activity will grow substantially as agents become more capable. Getting agent identity right on the open web is one of the more consequential near-term policy questions in this space.

Identity standards must serve accountability, not gatekeeping. As agent traffic grows, website owners should be able to tell when an agent is browsing their site on a user's behalf and to communicate how they want agents to interact with their content. AI developers, in turn, should identify their agents so they can be distinguished from malicious bots and so website owners can make informed decisions. But that only works if transparency is rewarded rather than exploited. If identifying an agent simply makes it easier for infrastructure intermediaries to block it or toll it by default, developers face a perverse incentive to obscure what their agents are. The result is a race to the bottom in which responsible actors who identify their agents are disadvantaged while opaque ones pass through freely. There is a foreign competition dimension

here as well. Foreign AI developers may not follow the same transparency norms that American companies would, and standards that penalize disclosure put American AI at a competitive disadvantage.

More broadly, the economic value of agentic AI depends substantially on agents being able to act on the open web on a user's behalf. Infrastructure intermediaries that position themselves as gatekeepers, extracting fees for agent access without creating corresponding value for website owners, risk slowing the adoption of a technology that matters for U.S. productivity and competitiveness. Individual website owners should decide how agents interact with their content. It is a different matter when intermediaries make that decision for them, particularly when those intermediaries have commercial interests of their own in the outcome.

NIST should encourage agent identity standards that give website owners clear, direct signals about the agents visiting their properties while ensuring the infrastructure delivering those signals does not become a chokepoint. The NCCoE's forthcoming concept paper on agent identity and authorization is a welcome step, and we appreciate CAISI taking this up while the problem is still tractable. Anthropic looks forward to engaging with that work as a complement to the guidance this RFI will inform.

5. Additional Considerations

5a. What methods, guidelines, resources, information, or tools would aid the AI ecosystem in the rapid adoption of security practices affecting AI agent systems and promoting the ecosystem of AI agent system security innovation?

Open, industry-led standards are the fastest path to broad adoption of agentic security practices and the most durable foundation for interoperability.

NIST should recognize that corporate or foundation-backed bodies like the Agentic AI Foundation (AAIF), a sub-foundation of the Linux Foundation, can produce credible, governance-quality standards without the overhead and timelines of traditional standards bodies. Model Context Protocol is an instructive example. Anthropic released MCP in late 2024, saw rapid adoption across the industry within months, and donated the protocol to the Linux Foundation less than a year later. This is a pace no formal standards process could match, and the donation gives the protocol the neutral governance that standards need for broad legitimacy.

In general, we encourage CAISI to seek out observer or advisory roles in the industry-led bodies where agentic standards are actively being developed, so that government expertise informs these standards as they mature rather than reviewing them after the fact. Open, neutral

governance structures also help ensure that major agentic standards remain under American stewardship, as compared to international standards processes where influence is harder to coordinate.

Open standards are also a competition policy tool. They prevent vendor lock-in, preserve the inter-lab competition that has driven American AI innovation, and let companies build once and deploy everywhere. It is important that the tools and protocols agents use remain open, so that AI labs continue to compete on the quality of their models and the safety of their products rather than on which proprietary tools or integrations they control.

5b. In which policy or practice areas is government collaboration with the AI ecosystem most urgent or most likely to lead to improvements in the state of security of AI agent systems today and into the future?

Government collaboration with the AI ecosystem is most likely to improve agent security in these areas:

Measurement infrastructure. A significant gap in agentic security today is not a lack of security technology but a lack of tools to assess how secure a given system actually is. Deployers in regulated and high-trust sectors want to adopt agents but lack standardized methods to evaluate how much of a problem prompt injection and other threats are for their specific use cases. This makes it difficult to make informed risk-acceptance decisions. AI companies have a role to play in building measurement tools and sharing what they learn, and government can accelerate progress by encouraging this work, providing guidance on what good measurement looks like, supporting organizations building standardized benchmarks, and, where appropriate, incorporating measurement expectations into regulatory frameworks.

Evaluations that reflect real deployments. The benchmarks that do exist for agentic security tend to test models in isolation, against synthetic attacks, in environments that look nothing like how agents are actually deployed. What the field lacks is evaluation infrastructure where realistic deployment conditions can be stood up quickly, existing and novel threat models can be run against them, and the results say something reliable about how a system will hold up in production. Today that kind of assessment takes a bespoke engineering effort every time, which means it rarely happens and the results are hard to compare across systems. NIST's January 2025 work on agent hijacking evaluations[23] was an important step in this direction, and we are encouraged by CAISI's continuing investment in agentic threat evaluation. Anthropic has seen the benefit of this kind of government testing firsthand. Under our voluntary agreement, CAISI has conducted pre-deployment evaluations across multiple Claude releases and red-teamed the safeguard systems that protect against misuse, bringing expertise in threat modeling and adversarial testing that no individual lab has on its own. That work has materially improved

[23] NIST CAISI, "Strengthening AI Agent Hijacking Evaluations," supra note 7.

what we ship. We would encourage further investment in reusable evaluation environments that lower the cost of running these assessments and make results comparable across systems.

Best practices clearinghouse. NIST is well-positioned to serve as a clearinghouse for agentic security best practices, similar to its role in traditional cybersecurity. This is especially valuable because the field is moving fast and individual companies' learnings are often siloed. Encouraging and facilitating the sharing of empirical data on agent usage, security incidents, and the effectiveness of different controls would help the entire ecosystem learn faster.

Facilitation over prescription. Government collaboration should prioritize measurement, evaluation, and information-sharing over prescriptive technical mandates. The technology is evolving too quickly for mandated designs, and the diversity of agent architectures, deployment contexts, and use cases means that any specific technical requirement risks being either too narrow to be broadly useful or too broad to be practically effective. The need for shared infrastructure to assess security, however, is urgent, broadly applicable, and well-suited to the government's convening and investment role.

5c. In which critical areas should research be focused to improve the current state of security practices affecting AI agent systems?

In general, the field is underinvesting in post-deployment empirical research on how people actually use agents. Theoretical threat models are necessary but routinely fail to predict which risks matter in practice or how users actually interact with the safety mechanisms built for them. Closing that gap is, in Anthropic's view, one of the highest-leverage research priorities in agentic security today.

Anthropic recently published detailed empirical research on how people use AI agents, examining usage patterns, human oversight behavior, and how autonomy evolves over time.[24] This builds on a broader pattern of publishing empirical data about how people use Claude, including the Anthropic Economic Index,[25] which tracks AI usage across the economy. We plan to continue and expand this work as agents become more capable and widely deployed, and we believe the field as a whole would benefit from more companies doing the same.

Empirical findings can drive more secure agent design. Anthropic's published research on Claude Code usage turned up findings that inform how oversight mechanisms get built. Experienced users auto-approve roughly twice as often as new users and also interrupt Claude more often mid-execution. These go together. New users review each action before it happens. Experienced users let the agent run and step in when something needs redirecting. Oversight has not decreased, it has moved.

---

[24] Anthropic, "Measuring Agent Autonomy," supra note 2.
[25] Anthropic, Anthropic Economic Index, supra note 3.

This points toward investing in interruptibility and legibility over ever-more-granular permission prompts. Permission dialogues still have a role, and since the users who see them most are newcomers building trust, they should favor clarity over technical completeness.

None of this is obvious from first principles. A reasonable theoretical model would predict that experienced users engage more with permission systems, not less, and that interrupt rates fall as trust builds. Both predictions would be wrong, and oversight built around them would be designed for users who do not exist. That is the case for empirical research: it catches where intuition and reality diverge, which is exactly where design goes wrong.

The field needs more empirical research on agents. Anthropic has been an early mover here, but no single company's user base is representative, and the questions this research is trying to answer are about how agents behave across the whole ecosystem, not one slice of it. The picture would be far clearer if multiple companies with different user populations, use cases, and deployment contexts were publishing findings about where security mechanisms engage and what users do when they fire.

NIST could accelerate this by encouraging more companies to publish, helping coordinate research agendas so findings are comparable, and working toward shared definitions and metrics so the field is measuring the same things.

Conclusion

Anthropic thanks NIST and CAISI for the opportunity to contribute to this conversation. The questions raised in this RFI arrive at a moment when the field can still get the foundations right, and we believe CAISI is well positioned to lead on the definitional, measurement, and convening work that agentic security needs most. We welcome the chance to stay engaged as this effort moves forward.