OWASP Agentic AI Top 10 Survival Guide

A New Era of Intelligent Systems and Their Emerging Risks

paloalto NETWORKS | PRISMA AIRS™

Foreword

Why the OWASP Agentic Top 10 Matters Now

The Open Worldwide Application Security Project or OWASP® is instrumental in helping organizations secure applications and navigate major shifts in security by translating emerging risks into practical, widely adopted guidance.

The OWASP Top 10 for Agentic Applications list is a go-to resource for securing AI and AI agents in a comprehensive and best practice-oriented fashion. It serves as a mainstay for security practitioners globally as threats continue to evolve and agentic AI attack surfaces expand.

Agentic AI systems fundamentally change how risk manifests in the enterprise. In addition to generating outputs, these systems plan, retrieve information, invoke tools, maintain memory, and execute actions across multiple steps with limited human oversight. A downside is that risk moves beyond isolated vulnerabilities toward behavior-driven failure modes that emerge over time.

The OWASP Agentic Top 10 captures this shift well. Where the LLM Top 10 focuses primarily on prompt-level issues, the Agentic Top 10 addresses the risks tied to autonomous action, such as goal manipulation, unsafe tool invocation, and unsafe multiagent coordination. These systemic risks surface as agents interact with tools, data, identity systems, and each other. They can expose gaps in many existing security architectures.

In agentic environments, understanding intent, constraining autonomy, and enforcing policy at runtime become as important as authentication and authorization, which is where the security ecosystem is beginning to evolve. Platforms that combine AI runtime controls with cloud security and SOC visibility illustrate the type of integrated control layer required to operationalize the principles outlined in the Agentic Top 10.

Lawrence Pingree
Head of Data and AI Research
Software Analyst Cyber Research

Introduction

Agentic AI has moved quietly but quickly from isolated pilots to the connective fabric of modern enterprises. What used to be small experiments are now systems that pull information together, make sense of it, and act without waiting for a human to drive every step. Once that happens, the system responds; more so, it behaves.

For CISOs, that shift is significant. The familiar guardrails—identity controls, API gateways, and static permissions—no longer describe how these systems operate. Agents learn from what they've seen, interpret what they're given, and adjust as they go. This power enables both scale and speed, as well as introduces failure modes that don't resemble traditional application vulnerabilities and can't be contained by configuration alone.

Threat mitigation in this environment takes on a different role. It still patches vulnerabilities or reacts when something breaks, but now, also reduces risk early by shaping what agents are allowed to do. In addition, it watches for attempts to manipulate or redirect them as they operate and quickly contains unsafe behavior before it leads to real-world impact.

This shift mirrors how modern cybersecurity has evolved more broadly through prevention, detection, and response working together as a continuous control loop. While eliminating risk is unrealistic for autonomous systems, the objective now is to keep risk constrained, observable, and governable.

Recognizing these emerging challenges, OWASP released the Top 10 for Agentic Applications for 2026, signaling that autonomous behavior has matured beyond what legacy guardrails can control. The focus moves from securing outputs to governing intent, planning, memory, and tool use—the elements that define how agents behave in production environments.

This guide, which uses the OWASP Agentic Top 10 as its foundation, outlines each class of agentic threat, the mitigation strategies to reduce its exposure, and how Prisma® AIRS™ operationalizes protection in real enterprise environments.

ASI01

Agent Goal Hijack

Attackers manipulate an agent's natural-language input to affect and alter its intended goals, exfiltrating data, manipulating outputs, or hijacking workflows.

ASI02

Tool Misuse and Exploitation

Agents misuse legitimate tools using prompt manipulation or privilege control, resulting in data exfiltration, unsafe operations, output manipulation, or workflow hijacking.

ASI03

Identity and Privilege Abuse

Weak scoping and dynamic delegation allow privilege escalation and cross-agent impersonation through cached credentials, inherited roles, or unintended delegated scopes.

ASI04

Agentic Supply Chain Vulnerabilities

Poisoned or impersonated tools, dynamically loaded prompts, models, or connections to MCPs or external agents propagate malicious logic at runtime, compromising agents through dynamic dependencies and unverified sources.

ASI05

Unexpected Code Execution (RCE)

Crafted prompts or poisoned inputs trigger unsafe code generation, agent deserialization, or shell execution.

ASI06

Memory and Context Poisoning

Adversaries poison RAG stores, memory, or context windows to plant false knowledge, bias logic, or trigger hidden or risky behaviors across sessions or agents.

ASI07

Insecure Inter-Agent Communication

A lack of encryption, authentication, or semantic validation of exchanges between agents enables message tampering, replay, or goal manipulation in multiagent systems.

ASI08

Cascading Failures

A single fault or malicious event propagates across interlinked agents, amplifying harm through chained autonomous actions.

ASI09

Human-Agent Trust Exploitation

Attackers exploit user overtrust in agent outputs through deception, emotional manipulation, or fake explainability, driving unsafe or fraudulent human approvals.

ASI10

Rogue Agents

Compromised or malicious agents deviate from intended goals, collude, self-replicate, or hijack workflows, acting as autonomous insider threats within agent ecosystems.

Figure 1. OWASP Top 10 for Agentic Applications

1 | When the Agent Gets Tricked

OWASP ASI01: Agent Goal Hijack

Agentic systems receive instructions from many sources, including system prompts, user input, tools, workflows, and memory. When those instructions conflict or are subtly manipulated, the agent can drift from its intended purpose. This behavior often looks like confusion at first, until the agent takes an unintended or harmful action.

Attackers exploit this situation by manipulating prompts or context to shift the agent's role, bypass guardrails, or redirect behavior without obvious rule breaking.

Mitigation Strategy

- **Confirm the instructions are consistent and don't compete.** Normalize inputs so the agent isn't pulled in conflicting directions, such as pairing "recommend only" guidance with action-oriented directives.
- **Anchor the agent to a clear task definition.** Maintain a stable description of what the agent is meant to accomplish and prevent unapproved role changes.
- **Inspect inputs for goal manipulation.** Review prompts, retrieved content, and tool outputs for hidden objectives, redirection, or false urgency.
- **Watch for shifts in reasoning or behavior.** Detect signs that the agent is mixing directives or acting outside its intended role.

How Palo Alto Networks Helps

Prisma AIRS maintains clarity of agent intent as instructions flow in from multiple sources and helps prevent agents from being redirected to malicious action. AI Agent Security governs how instructions, goals, and context are interpreted. AI Runtime Security™ scans user inputs, Model Context Protocol (MCP) tool traffic, and Retrieval-Augmented Generation (RAG) context to detect adversarial manipulation and goal redirection attempts. Meanwhile, AI Posture Management surfaces configuration drift that can weaken guardrails.

2 | When the Agent's Tools Are Manipulated

OWASP ASI02: Tool Misuse and Exploitation

Agents rely on tools—APIs, scripts, and systems—to take action. If an attacker convinces the agent to call a tool out of context or with malicious parameters, the consequences can be immediate and severe.

Mitigation Strategy

- **Define which tools the agent is allowed to use and when.** For example, you might limit a patient record lookup tool to read-only data retrieval so an attacker can't coerce the agent into unsafe or unintended tool use.
- **Check tool requests before they run.** Each request should make sense and match the agent's approved responsibilities.
- **Watch for unusual tool activity.** Flag high-frequency tool use, irregular timing, or calls that don't match the task.

How Palo Alto Networks Helps

Prisma AIRS gives organizations control over how AI agents use tools and take action across SaaS environments. AI Agent Security restricts tool access to only what each agent is explicitly authorized to use, limiting opportunities for misuse or abuse. In production, AI Runtime Security helps intervene when tool descriptions and invocations contain threats, such as malicious URLs and malicious code. AI Posture Management continuously watches for unauthorized changes to tool permissions that could expand an agent's reach.

3 | When the Agent Oversteps Access

OWASP ASI03: Identity and Privilege Abuse

Agentic systems act with access. When credentials or permissions are too broad, quietly expanded, or left in place too long, agents can access data or perform actions beyond their intended scope—even when no system has been compromised.

Mitigation Strategy

- **Apply least-privileged access that aligns with the agent's role.** Permissions are limited to what the agent needs to read or do, and nothing more.
- **Review credentials and scopes before and during operation.** Confirm that tokens, roles, and access scopes remain appropriate as the agent runs and evolves.
- **Require additional checks for elevated actions.** Use approvals or validation for high-risk or privileged operations.
- **Watch for permission creep over time.** Detect unauthorized increases in access, new systems, or actions that don't match expected behavior.
- **Remove access that is no longer needed.** Automatically revoke unused, expired, or excessive permissions.

How Palo Alto Networks Helps

Prisma AIRS enforces clear permission boundaries for AI agents across enterprise and SaaS environments. AI Agent Security defines and governs agent access, while AI Posture Management tracks identities, effective permissions, and drift over time. In production, AI Runtime Security helps to block inputs to agents that attempt to override its identity to follow new permissions.

4 | When the Supply Chain Is Poisoned

OWASP ASI04: Agentic Supply Chain Vulnerabilities

Agents dynamically load tools, plugins, models, and prompts at runtime. If a single dependency is tampered with, an attacker can introduce unsafe code or hidden instructions that compromise the entire workflow.

Mitigation Strategy

- **Maintain a live inventory.** Know exactly which agents, tools, and plugins are running in your environment at all times.
- **Validate tool definitions.** Don't blindly trust external tools. Scan schemas and descriptors for malicious instructions before the agent loads them.
- **Detect shadow usage.** Watch for agents that connect to unapproved or "shadow" servers, like unauthorized MCP servers, that bypass standard vetting.
- **Monitor dependencies.** Check for risky code libraries or compromised models that could propagate malicious logic.

How Palo Alto Networks Helps

Prisma AIRS helps secure the agentic supply chain by ensuring that every component an agent loads is visible and verified. AI Posture Management maintains a continuous inventory of all agents, tools, plugins, code dependence analysis, and models, enabling teams to quickly identify unauthorized additions or configuration drift. AI Runtime Security complements this by actively scanning MCP tool schemas for malicious instructions hidden in tool descriptors and detecting shadow usage of unauthorized MCP servers through App-ID™.

5 | When Outputs Cause Real-World Harm

OWASP ASI05: Unexpected Code Execution (RCE)

While agentic systems generate text, their outputs can trigger commands, system changes, or real-world actions. If those outputs aren't checked before execution, a single response can cause irreversible harm, even when the system didn't intend to do anything malicious.

Mitigation Strategy

- **Check high-impact actions before they run.** Confirm commands and system changes are safe and expected.
- **Require approvals for risky or irreversible actions.** Insert human judgment where consequences matter.
- **Watch for outputs that could cause harm.** Detect actions that fall outside allowed behavior.

How Palo Alto Networks Helps

When AI outputs are connected directly to systems and workflows, even a well-intentioned response can create real-world consequences. Prisma AIRS helps reduce that risk by adding safeguards before AI-generated actions are executed.

AI Runtime Security evaluates agent outputs prior to execution to help detect malicious or unsafe behavior. Over time, AI Posture Management enforces least-privileged policies so agents have only the access required to perform approved actions.

6 | When Memory Gets Tampered With

OWASP ASI06: Memory and Context Poisoning

Agentic systems both respond and remember. They store conversation history, retrieved documents, intermediate results, and long-term context that shapes future decisions. If attackers can influence what is written into that memory, they can quietly steer the agent's behavior over time, even when later interactions seem normal or trustworthy.

Mitigation Strategy

- **Check everything before it becomes a memory.** Only allow information into memory after it's been reviewed for hidden instructions, misleading content, or policy violations.
- **Control who and what can change memory.** Limit memory updates to approved processes. Don't allow tools, users, or external sources to write freely to long-term context.
- **Watch for unexpected memory changes.** Alert when memory is edited, overwritten, or expanded in ways that don't match the task at hand.
- **Validate behavior against trusted context.** Compare outputs to approved sources to catch scenarios where the system starts "believing" something it shouldn't.

How Palo Alto Networks Helps

Prisma AIRS protects an AI agent's memory to remain a trusted asset rather than an attack surface. AI Runtime Security can inspect RAG context before it is sent to LLMs by helping identify and mitigate poisoned content or hidden instructions that could influence agent behavior over time. It can also prevent an agent from accessing a third-party URL that might contain poisoned data or instructions.

AI Agent Security governs how agentic systems are permitted to interact with applications, tools, and data—based on intent, context, and policy. AI Posture Management continuously discovers RAG data sources—including data sources teams might not know the agent is using—to mitigate the risk of shadow poisoning and flag unauthorized changes.

7 | When Agents Team up the Wrong Way

OWASP ASI07: Insecure Inter-Agent Communication

Agents often work together by sharing messages and decisions. If those communications aren't controlled, agents can reinforce bad instructions, coordinate unsafe actions, or influence each other in ways no one intended.

Mitigation Strategy

- **Limit what agents can share with each other.** Prevent unintended reinforcement of unsafe decisions.
- **Check agent-to-agent messages.** Confirm communications don't contain risky instructions or hidden collaboration.
- **Watch for unexpected coordinated behavior.** Detect when agents start acting together outside the intended design.

How Palo Alto Networks Helps

When AI agents begin interacting with one another, a loss of control often happens gradually. Prisma AIRS helps teams stay ahead of that risk by providing visibility and guardrails as agents exchange messages and influence decisions.

AI Runtime Security scans a tool description schema to verify whether third-party tools and agents contain evidence of injected threats. AI Runtime Security Network Intercept enforces protocol-level controls on interagent traffic, helping to prevent attack vectors, such as a cryptographic protocol downgrade, by requiring specific TLS protocols. AI Posture Management gives visibility into agent interactions and prevents permission creep that can lead to harmful activity.

8 | When the Agent Loops Out of Control

OWASP ASI08: Cascading Failures

Agentic systems can plan and act across many steps without stopping. If no clear limits are set, a single mistake can turn into a chain of actions that grows, repeats, or escalates until the system does real harm.

Mitigation Strategy

- **Set limits on how far the agent can plan or chain actions.** Prevent long or escalating sequences.
- **Detect repeated or escalating patterns.** Identify loops or self-reinforcing plans early.
- **Use automatic stop conditions.** Shut down or pause the agent when activity becomes unsafe.

How Palo Alto Networks Helps

Prisma AIRS helps keep agent autonomy in check by placing explicit limits on how plans are formed and carried out. As agents begin executing those plans, AI Runtime Security focuses on reducing the risk of cascading failures at the source. It detects and blocks agent outputs that conflict with context, contain malicious instructions or jailbreak attempts, or introduce unsafe behavior before it can persist or propagate across agents. AI Posture Management enforces least-privileged access and policy controls so safeguards remain intact over time, even as agent tools, permissions, and environments evolve.

9 | When the Agent Manipulates the Human

OWASP ASI09: Human-Agent Trust Exploitation

AI agents operate pervasively across workflows and systems, optimizing for responsiveness and task completion. Their natural language fluency and perceived expertise can foster user trust, increasing the risk of overreliance.

If an agent is compromised or misaligned, it can exploit this trust to subtly influence human decisions—guiding users to approve harmful actions that appear legitimate. In these scenarios, the agent becomes an invisible intermediary, manipulating outcomes while leaving the final, auditable action to the human, complicating detection and forensic attribution.

Mitigation Strategy

- **Detect manipulative or persuasive agent outputs.** Identify emotional pressure, misleading explanations, or fabricated rationales aimed at influencing users.
- **Require validation for sensitive actions.** High-impact actions cannot be approved based on a single agent response.
- **Provide visibility and accountability.** Maintain logs of agent recommendations and explanations to support review and investigation.
- **Watch for repeated persuasion patterns.** Detect attempts to steer users toward risky outcomes over time.

How Palo Alto Networks Helps

Prisma AIRS helps prevent attackers from taking advantage of the trust between an agent and its users. AI Runtime Security filters instructions designed to exploit trust, reducing the likelihood that a single jailbreak prompt could lead to high-risk actions. It can mitigate the downstream effects of human-agent trust exploitation by preventing sensitive data leaks, malicious code generation, and other risks. AI Posture Management can enable your organization to discover and regulate the scope of an agent's privileges, reducing the possible attack surface of human-agent trust.

10 | When the Agent Goes Rogue

OWASP ASI10: Rogue Agents

Sometimes the threat isn't an external attacker, but the agent itself. Whether compromised by a bad actor or misaligned through cumulative errors, a "rogue agent" deviates from its intended function. It acts like an autonomous insider threat that can deceive humans or execute harmful actions.

Mitigation Strategy

- **Watch for behavioral deviation.** Define what "normal" looks like for each agent and flag high-risk actions that don't match its role.
- **Block data exfiltration.** Agents cannot send sensitive data to unauthorized parties or external destinations.
- **Prevent C2 communication.** Detect and block attempts by agents to contact command and control (C2) servers.
- **Isolate compromised agents.** Have the ability to cut off an agent's access immediately if it shows signs of acting autonomously against your interests.

How Palo Alto Networks Helps

Prisma AIRS mitigates the risk of insider threats by monitoring agent intent and restricting execution paths to limit the potential impact of unsafe or malicious behavior. AI Runtime Security detects when a rogue agent attempts to invoke malicious C2 servers or execute malicious code. It helps disrupt attack activity as it occurs, while enforcing data controls that reduce the risk of exfiltrating sensitive information to unauthorized parties. AI Posture Management further reduces the attack surface by controlling agent privileges, limiting the potential impact of a compromised or misaligned agent.

Conclusion: Build Agentic AI You Can Trust and Control

Agentic AI is everywhere in the enterprise now—in workflows, SaaS apps, internal tools, and decision systems. The OWASP Agentic Top 10 helps define the risks, but security leaders need more than a framework. You need a way to enforce guardrails continuously, as agents plan, adapt, and act in real time.

These guardrails are essential because agentic systems rarely fail in obvious ways, but rather drift. Small deviations—like a misleading prompt, an unsafe tool invocation, or a subtle permission change—compound quietly until they produce real-world impact. In this environment, a reliable defense provides sustained visibility and active control over how agents behave and what they output.

That's where Prisma AIRS helps.

Prisma AIRS provides this control layer by watching what agents receive and retrieve, which tools they use, and what actions they attempt. It can block unsafe steps, limit permissions, filter harmful outputs, and trace behavior across SaaS agents, MCP workflows, and custom builds.

Prisma AIRS gives enterprises something essential—a unified way to help agentic AI stay fast, safe, and predictable.

You don't need to slow innovation to secure agents. You need to guide it with the right guardrails in place so you can deploy agentic AI bravely.

The Agentic AI Defense Scorecard

Objective: Assess your organization's readiness to detect, block, and mitigate the 10 critical threats facing autonomous AI agents.

Instructions: For each risk domain, evaluate your current controls using one of the following responses:

- **No (N):** We have no visibility or control over this.
- **Partial (P):** We have manual reviews or limited coverage.
- **Yes (Y):** We have automated controls in place.

Part 1: Input and Context Security

Preventing attackers from manipulating an agent’s purpose or perception

Assessment Question	Related Threat	Risk Context	Response (N, P, or Y)
1. Do you check everything that’s sent to an AI agent to make sure no one is trying to trick it into ignoring rules or acting in an unsafe way?	**Agent Goal Hijack**	An attacker can slowly change what an AI agent believes its role or purpose is so the agent starts taking unsafe risks without realizing it’s breaking rules.	
2. Do you detect manipulative or persuasive language designed to either pressure the agent into bypassing safeguards, or to influence human users into approving risky actions (e.g., “do this immediately and don’t question it”)?	**Human-Agent Trust Exploitation**	Adversaries or misaligned agents can exploit the trust between humans and agents to bypass oversight, extract sensitive data, or persuade users to approve harmful actions.	
3. Do you validate that an AI agent’s declared role, scope, and authorized context remain aligned with its intended purpose, and prevent it from operating outside those bounds?	**Rogue Agents**	A compromised or misconfigured agent can continue operating outside its approved role or scope, while appearing legitimate.	

Part 2: Tool and Data Governance

Controlling how the agent interacts with external systems and information

Assessment Question	Related Threat	Risk Context	Response (N, P, or Y)
1. Do you clearly decide in advance which tools an AI agent is allowed to use—and block access to everything else?	**Tool Misuse and Exploitation**	Attackers can trick an AI agent into using tools the wrong way or with harmful inputs.	
2. Do you check documents or data before an AI agent uses them to make sure they don't contain malicious, misleading, or sensitive content?	**Memory and Context Poisoning**	Misleading data pulled from files or the web can distort the agent's reasoning and influence future decisions.	
3. Does a human need to review and approve important or high-risk actions before an AI system is allowed to execute them?	**Cascading Failures**	Irreversible actions (e.g., deleting files or transferring funds) require judgment beyond the agent's capability.	

Part 3: Behavioral and Runtime Safety

Monitoring the agent's autonomy and logic in real time

Assessment Question	Related Threat	Risk Context	Response (N, P, or Y)
1. Do you check new information for safety and accuracy *before* an AI agent is allowed to store it and rely on it in the future?	**Memory and Context Poisoning**	Malicious content inserted into memory can influence decisions long term.	
2. Do you have predefined limits and kill switches that halt an AI agent without human intervention when behavior becomes unsafe, unbounded, or wasteful?	**Cascading Failures**	AI agents can get stuck repeating themselves, wasting resources, or taking big and bigger actions.	
3. Are agent-to-agent messages logged, visible, and inspected—rather than treated as an internal black box?	**Insecure Inter-Agent Communication**	Agents working together can reinforce unsafe decisions or share prohibited data.	
4. Can you detect and stop AI systems (or their credentials) from quietly gaining more access over time than they were originally approved for?	**Identity and Privilege Abuse**	AI agents should only have the access they need. Any attempt to gain more access must be caught right away.	

Scorecard Analysis:
Interpreting Your Results

This scorecard is designed to surface how well your organization can govern agentic behavior and autonomy over time, as well as whether basic safeguards exist. The outcomes signal where risk is most likely to accumulate as autonomy increases.

As you review your responses, look for patterns rather than individual answers. Agentic systems rarely fail because of a single missing control. They fail when small gaps compound across inputs, tools, memory, and autonomy.

The Blind Zone: Mostly "No" Answers

What This Result Means

A majority of "No" responses indicates that AI agents in your environment are effectively operating as black boxes. You have limited visibility into how agents interpret instructions, what data they rely on, or how their actions are constrained. In this state, agents are highly susceptible to goal hijacking, human-agent trust exploitation, and unsafe tool use—often without triggering traditional security alerts.

Why It's Risky

In the Blind Zone, failures tend to surface late, after damage has already occurred. Because intent and behavior aren't monitored, attacks manifest as "normal" agent activity until outcomes become irreversible.

Immediate Priorities

Focus first on both "Input and Context Security" and "Tool and Data Governance":

- Inspect and filter inputs, retrieved context, and role-defining instructions that influence agent behavior.
- Explicitly restrict which tools and data sources agents can access.
- Reduce risk quickly by shrinking the attack surface before autonomy expands further.

The Drift Zone: Mostly “Partial” Answers

What This Result Means

“Partial” responses signal that some guardrails exist, but that they are uneven, manual, or static, which is common in organizations that rely on API gateways, fixed permissions, or periodic reviews. These controls provide baseline protection, but they are not designed to detect behavioral drift in agents that plan, loop, and adapt dynamically.

Why It’s Risky

In the Drift Zone, problems accumulate quietly. An agent might gradually expand its scope or privileges, reuse unsafe memory, or chain actions in unintended ways—none of which look anomalous in isolation. Over time, this creates a systemic risk that is difficult to unwind.

Immediate Priorities

Shift focus to “Behavioral and Runtime Safety”:

- Monitor for looping, escalation, and privilege creep.
- Enforce limits and kill switches that operate automatically.
- Move from static permissioning to continuous behavioral oversight.

The Governor Zone: If Your Answers Are Mostly “Yes”

What This Result Means

A majority of “Yes” responses indicates that you have established a balance of prevention, detection, and response for agentic systems. You are actively shaping behavior, observing agents at runtime, and intervening before unsafe actions are executed or allowed to persist.

What Comes Next

At this stage, the challenge is scale. Manual reviews and point solutions do not hold as agentic AI spreads across SaaS platforms, internal tools, and multiagent workflows.

Next Step

Operationalize these controls using automation and centralized visibility. Platforms, such as Prisma AIRS, help enforce agent-level guardrails consistently.

Turn Guardrails into Real-World Protection

If you're evaluating how to operationalize the OWASP Agentic Top 10 in your environment, our experts can help you.

Speak with an AI security expert →

Authors

Jane Leung

Sahil Mehta

Munjal Munshi

Hitendar Sethi

paloalto NETWORKS

3000 Tannery Way
Santa Clara, CA 95054

Main: +1.408.753.4000
Sales: +1.866.320.4788
Support: +1.866.898.9087

www.paloaltonetworks.com

prisma_eb_owasp-agentic-ai-top-10-survival-guide_010625