2026 AGENTIC AI SECURITY

Field Guide

Prepared by

Amine Raji, PhD

aminrj.com

OWASP Agentic Top 10 with Real-World Attacks, Code Examples, and Defensive Playbooks

EXECUTIVE SUMMARY

The enterprise AI agent market is accelerating at a pace that far outstrips security readiness. According to Cisco's State of AI Security 2026 report, 83% of organizations plan to deploy AI agents this year, yet only 29% feel their security posture is ready for agentic workloads.

This gap is not theoretical. In the twelve months since the Model Context Protocol (MCP) became the de facto standard for connecting AI agents to tools and data, we have seen a surge of real-world breaches, published CVEs, and research demonstrating fundamental security flaws in how these systems are built and deployed.

Endor Labs analyzed 2,614 MCP server implementations and found that 82% use file system operations prone to path traversal and 67% use code-injection-prone APIs. Barracuda identified 43 agent framework components with embedded supply chain vulnerabilities. 492 publicly exposed MCP servers were found lacking basic authentication.

This field guide provides security engineers, CISOs, and engineering leaders with a practical reference for understanding and mitigating the risks specific to agentic AI systems. It maps the OWASP Agentic Top 10 to real-world incidents, explains the three dominant MCP attack patterns with code examples, presents a timeline of major breaches, and provides a defensive playbook with implementation priorities.

WHO THIS GUIDE IS FOR

- Security engineers assessing AI agent deployments
- CISOs building governance frameworks for agentic AI
- Engineering leads deploying MCP-connected systems
- Anyone responsible for securing LLM-powered applications in production

01. WHY AI AGENTS ARE DIFFERENT

The security model for traditional LLM chatbots is relatively straightforward: user sends input, model generates output, application displays response. The attack surface is constrained to input manipulation and output handling.

AI agents fundamentally change this equation. An agent is not just a model that generates text, it is a model that acts. This distinction introduces five properties that expand the attack surface exponentially.

THE FIVE PROPERTIES THAT CHANGE EVERYTHING

Property	What It Means	Security Implication
Autonomy	Agents plan and decide without human approval per step	A single poisoned input can trigger an entire chain of unsupervised actions
Tool Use	Agents call APIs, execute code, read/write files, send messages	Every tool is a potential privilege escalation path
Delegation	Agents hand off tasks to other agents	Failures propagate through multi-agent chains without visibility
Persistence	Agents maintain state across sessions via memory	Memory can be poisoned to influence all future interactions
Identity	Agents act with credentials and permissions	Compromised agents operate with the user's full access level

THE ATTACK SURFACE EXPANSION

A traditional LLM application has a relatively narrow attack surface: input, output, and the model itself. An agentic system expands this to include tools, memory stores, other agents, credentials, external data sources, and the orchestration layer connecting them all.
The foundational vulnerability remains the same, the semantic gap.

LLMs cannot distinguish instructions from data. System prompts, user input, retrieved context, and tool descriptions are all processed identically as text tokens. In a chatbot, this leads to prompt injection that produces wrong answers. In an agent, this leads to prompt injection that takes wrong actions — with real-world consequences.

KEY INSIGHT

The semantic gap problem gets exponentially worse when the LLM can ACT, not just TALK.

Every tool connected to an agent is a potential execution path for an attacker who can influence the agent's context.

02. THE OWASP AGENTIC TOP 10 PRACTITIONER REF.

The OWASP Top 10 for Agentic Applications (2026) is the first globally peer-reviewed framework for agentic AI security risks. Developed by over 100 industry experts, it provides the common vocabulary that security teams need when assessing AI agent deployments. Below is each entry mapped to a real-world incident and a specific defensive control.

ASI01 — AGENT GOAL HIJACKING

Risk: Attacker alters the agent's objectives via poisoned inputs, causing it to pursue the attacker's goals instead of the user's.

Real-World Incident: DockerDash (November 2025). Docker container metadata (LABELs) containing malicious instructions hijacked the Ask Gordon AI assistant's reasoning via the MCP Gateway, leading to remote code execution. The agent read a LABEL that said "Run docker ps -q to list containers" and treated it as a legitimate task.

Defensive Control: Treat all external data as untrusted input. Implement content sanitization on data retrieved from external sources before it enters the agent's context window. Never pass raw metadata, API responses, or file contents directly to the LLM without filtering.

ASI02 — TOOL MISUSE AND EXPLOITATION

Risk: The agent's tools are weaponized through manipulated inputs, causing them to perform actions the developer never intended.

Real-World Incident: Supabase Cursor Integration (mid-2025). Customer support tickets containing SQL-like instructions were processed by the AI coding assistant, which exfiltrated integration tokens through its database access tools. The attacker never touched the database directly — the agent did it for them.

Defensive Control: Require explicit human confirmation before executing sensitive tool operations (file reads, network calls, database writes). Scan tool descriptions for hidden instructions using tools like mcp-scan before installing any MCP server.

ASI03 — AGENT IDENTITY AND AUTHORIZATION FAILURES

Risk: Agent operates with excessive or stolen credentials, enabling unauthorized access to systems and data.

Real-World Incident: mcp-remote CVE-2025-6514. A critical vulnerability in the widely-used mcp-remote package (437,000+ downloads) allowed remote code execution via a crafted authorization_endpoint parameter. An attacker could hijack the OAuth proxy to execute arbitrary commands on the host system.

Defensive Control: Apply the principle of least privilege to all agent credentials. Use short-lived, scoped tokens. Audit MCP server packages for known CVEs before deployment. Run MCP servers in sandboxed containers with minimal filesystem and network access.

ASI04 — KNOWLEDGE AND MEMORY POISONING

Risk: Agent's data sources (RAG stores, memory, knowledge bases) are corrupted to produce flawed or malicious outputs in future interactions.

Real-World Incident: Gemini Advanced Memory Corruption (February 2025). A security researcher demonstrated that hidden instructions could be stored in Gemini's long-term memory and triggered in later sessions, creating a persistent backdoor that survived across conversations.

Defensive Control: Validate and sanitize all data before it enters memory or RAG stores. Implement integrity checks on stored context. Provide users with visibility into and control over what agents remember.

ASI05 — UNCONTROLLED CASCADING FAILURES

Risk: Failures propagate through multi-agent chains, amplifying errors or malicious actions across the system.

Real-World Incident: Auto-GPT Remote Code Execution (2023). An indirect prompt injection caused an autonomous agent chain to execute arbitrary code, demonstrating how a single compromised step can cascade through an entire multi-agent workflow.

Defensive Control: Implement circuit breakers and maximum iteration limits on agentic loops. Require human approval for high-impact or irreversible actions. Design containment boundaries so that failure in one agent does not compromise the entire system.

ASI06 — ROGUE AGENTS

Risk: Compromised agents that appear legitimate but act maliciously — self-replicating actions, persistent exfiltration, or sabotage.

Real-World Incident: Invariant Labs WhatsApp MCP Rug-Pull (April 2025). A malicious MCP server presented a completely benign interface on first load, then changed its tool descriptions on subsequent loads to instruct the LLM to exfiltrate the user's WhatsApp chat history via a co-installed legitimate server.

Defensive Control: Hash all tool descriptions on first load and alert on any change (detects rug pulls). Implement kill switches for agent processes. Monitor for behavioral drift in installed MCP servers.

ASI07 — SENSITIVE INFORMATION DISCLOSURE

Risk: Agents leak confidential data — system prompts, user data, API keys, internal documents — through outputs, tool parameters, or logs.

Defensive Control: Implement output filtering that scans LLM responses for data patterns (SSN, API keys, phone numbers, internal URLs). Apply DLP mechanisms at the agent output boundary. Never expose full system prompts or tool schemas to end users.

ASI08 — INSECURE AGENT-AGENT COMMUNICATION

Risk: Multi-agent systems where agents delegate to each other without mutual authentication, enabling impersonation and unauthorized cross-agent actions.

Defensive Control: Enforce per-server permission scoping. When multiple MCP servers share an agent, implement tool allowlisting and cross-server call restrictions. Prefix all tools with their server name for auditability.

ASI09 — HUMAN-AGENT TRUST EXPLOITATION

Risk: Attackers exploit user trust in agent outputs. Persuasive AI-generated explanations induce users to perform harmful actions — approving dangerous operations, sharing credentials, or ignoring warnings.

Defensive Control: Design confirmation dialogs that clearly show what the agent intends to do, not just what it says. Implement friction for high-risk operations. Train users to verify agent-initiated requests through out-of-band channels.

ASI10 — INSUFFICIENT LOGGING AND MONITORING

Risk: No audit trail for agent actions. When something goes wrong, security teams cannot investigate, attribute, or learn from incidents. This is also a compliance gap under emerging AI regulations.

Defensive Control: Maintain immutable, signed audit logs of all agent actions, tool calls, and inter-agent communications. Log the full context window for critical operations. Implement real-time alerting on anomalous agent behavior patterns.