The AI Agent Security Audit You Haven't Done

Enterprise AI agents with tool use, web access, and file permissions are in production. Your security team hasn't reviewed them.

01

Background

What changed when AI stopped generating text and started taking actions.

01 — BACKGROUND

Agents with tool use are not the same security problem as chat.

- 2024–2025: enterprise AI shifted from retrieval-augmented generation (text answers) to agent frameworks (tool use, web browsing, code execution, file write, email sending). Different threat surface.
- Common enterprise agent capabilities in production: web search and retrieval, SQL query execution, file read/write, outbound email via SMTP/Graph API, calendar access, Slack/Teams messaging, Salesforce record updates, code execution.
- Security review cadence has not kept pace. Most agents passed an initial LLM provider review. The agent architecture — what the model can do — has not been reviewed at the same bar as traditional software.
- Prompt injection is the primary exploit vector: adversarial instructions embedded in a web page, document, or email that change what the agent does. No current AI framework has a reliable technical defense.
- OWASP LLM Top 10: prompt injection ranked #1 for two consecutive years. The mitigations are architectural, not model-level.
- Most enterprises have no documented permission model for any production AI agent. Tool lists have expanded incrementally without security review.

02

Decision Required

The question your security team has not been asked yet.

02 — DECISION REQUIRED

For every production AI agent: what can it do, and who reviewed it?

An AI agent's attack surface is its tool list. A model that can send email and browse the web has a fundamentally different risk profile than one that only answers questions.

The governing question: for each production agent, is there a documented permissions model — analogous to a service account access review — specifying which tools are permitted, which data is accessible, and which actions require human confirmation?

If the answer is no, you have production software with undefined permissions in an environment where control flow can be manipulated by adversarial inputs in the agent's context.

03 — OPTIONS

Three audit postures.

Option A

No dedicated AI agent audit — extend existing security review processes

Misses prompt injection, tool permission creep, and adversarial context threats. Standard pen testing and code review do not cover these vectors.

Option B

One-time AI agent security audit

Enumerate agents, document permissions, test for prompt injection, produce remediation backlog. Better than status quo; insufficient without an ongoing review process.

Option C

Recommended

Structured AI agent security program

Formal permission review at deployment, quarterly tool-permission audit, prompt injection testing in security cycle, AI agent incidents classified in SOC, change-control gate on tool list modifications.

04 — RECOMMENDATION

Start with the inventory. Build toward the program.

Step 1: Enumerate. Ask every business unit and dev team to report any AI system with tool use, file access, API calls, or outbound communication. You will find more than IT knows about.

Step 2: For each agent, document: tool list, data access scope, action permissions (what requires human confirmation), and the accountable owner.

Step 3: Prioritize agents with high-risk tool combinations — email sending + web browsing, or file write + document ingestion. These have the largest prompt injection surface.

Step 4: Test the top three highest-risk agents for prompt injection. Put adversarial instructions in content the agent is likely to process. Observe what happens.

Step 5: Establish two process controls: (1) change-control gate for agent tool list modifications; (2) AI agent incident category in your SOC classification framework.

05 — RISKS

Four material risks.

1.

Prompt injection in production agents

An adversarial instruction in a web page, document, or email the agent processes can redirect subsequent actions — data exfiltration, unauthorized communications, CRM record modifications. No current framework has a reliable technical defense.

2.

Tool permission creep

Developers add capabilities incrementally. There is no default alert when a production agent acquires a new tool. The agent reviewed in January with read-only DB access may have write access and web browsing by June.

3.

Incident attribution gap

When an agent takes an unexpected action, most frameworks log tool calls but not the full context window. Without context, you cannot distinguish prompt injection from model error — a gap for insurance claims and regulatory reporting.

4.

Insurance and regulatory coverage holes

Cyber insurance policies predate AI agents. Coverage for autonomous agent actions may fall in a gap. EU AI Act Article 73 requires a 72-hour incident report for high-risk AI serious incidents — most enterprises have no workflow ready.

06 — QUESTIONS YOUR TEAM SHOULD BE ANSWERING

If your team cannot answer these, that is your first deliverable.

1. Can you enumerate every AI agent in production — any LLM-based system with tool use, file access, API calls, or outbound communication? If not, the audit starts with discovery.
2. For each production agent: is there a documented permissions model reviewed by a security function, not only the dev team that built it?
3. Has your organisation tested any production agent for prompt injection? If not, you have not assessed the primary exploit surface.
4. When an agent's tool list changes, is there a security review gate with documented authority to approve new permissions?
5. Does your incident response plan include an AI agent incident category with detection, containment, and reporting procedures?
6. Have you reviewed your cyber insurance policy for AI agent coverage — specifically whether autonomous agent actions are covered under current policy language?

AI INSIGHT LAB

The Deployment Memo

One enterprise AI deployment, dissected every Tuesday.
Written for executives who have to decide, not just read.

Subscribe at aiinsightlab.cloud — free during beta.