



AI Agent Security Checklist

The AI agent security checklist: what to check before an agent ships, and again after | 34 items | version 2026.10, October 17, 2026

An AI agent is an assistant that can act: it reads documents, calls tools, writes to systems and sends messages. That is what makes it useful and what makes a single injected sentence expensive. This checklist is the set of questions we ask of every agent before it ships and again after every model or prompt change.

It is written for the people who own the deployment, not the model vendor. Most items are configuration, permissions and process. A few are tests. None of them require a RedLens account, and the PDF is free to print, share and staple into a change record.

1. Scope and inventory

You cannot secure an agent you have not described.

- ☐ The agent has a named owner, a written purpose, and a list of the systems it can reach.
- ☐ Every input channel is listed and marked trusted or untrusted (user chat, retrieved documents, email, web pages, tool results, other agents).
- ☐ The model, version, system prompt (or its hash), tools and retrieval corpus are recorded, with a date, so a test can be repeated against the same build.
- ☐ The data classes the agent can see are listed (PHI, CJI, PII, secrets, financial), and the risk analysis names the agent as a system that touches them.

For a HIPAA covered entity this is the risk analysis at 45 CFR 164.308(a)(1)(ii)(A).

2. Inputs and prompt injection

Instructions and data arrive through the same channel. Assume untrusted text will try to steer the agent.

- ☐ The system prompt cannot be edited by the user, and retrieved content is handed to the model labelled as data, not as instructions.
- ☐ Tool results are treated as untrusted input, the same as a web page.
- ☐ The agent has been tested with instructions planted in the document types it actually reads (indirect prompt injection), not only with hostile chat messages (direct).
- ☐ Known override and extraction patterns are tested on every release, and the results are kept.
- ☐ The agent cannot be talked into revealing its system prompt, its tool list or another user's data.

3. Tools and permissions

Excessive agency turns a leak into an action.

- ☐ Each tool the agent can call is listed with what it can read, write, send or spend.
- ☐ The agent has the least privilege its task needs: a summarizer has no send-message tool, a reader has no write access.
- ☐ High-impact actions (payments, record changes, outbound messages, deletions) require a human approval step that cannot be satisfied by the model's own output.
- ☐ Tool credentials are scoped to the agent and rotated; the agent never holds a human's personal credentials or a long-lived admin token.
- ☐ Rate and spend limits exist per tool and per session, and they fail closed.

4. Data and secrets

The context window is the blast radius.

- ☐ Retrieval is scoped to the record, tenant or case in context, never the whole corpus.
- ☐ Secrets, API keys and connection strings never appear in prompts, tool descriptions or retrieved documents.
- ☐ Conversation history and memory are stored with the same access controls as the source data, and are purged on a schedule.
- ☐ Testing uses synthetic records, never production PHI, CJI or customer data.

5. Outputs and egress

What the agent can render or reach is how data leaves.

- ☐ Links and images in model output are not fetched automatically; markdown and HTML from the model are sanitised before rendering.
- ☐ Outbound network destinations for the agent and its tools are on an allowlist; everything else is denied and logged.
- ☐ Model output is validated before it reaches a downstream system (no raw SQL, shell, or code execution from model text).
- ☐ Sandboxes used for code execution have no route to production data or credentials.

6. Logging and monitoring

Without provenance you can see that the agent leaked; you cannot see which document told it to.

- ☐ Every prompt, retrieved document, tool call, tool result and final output is logged with a timestamp and the channel that supplied each piece of text.
- ☐ Logs are retained for as long as the data they describe requires (six years under the HIPAA Security Rule's documentation standard), and are reviewed, not just stored.
- ☐ Known attack patterns from testing are watched for at runtime, so an attempt is found and flagged when it happens, with a trace back to the finding that defined it.
- ☐ Someone is named to receive and act on those flags, with a target response time.

7. Testing and evidence

A test from spring says little about the model answering today.

- ☐ The assembled system (prompt, retrieval, tools, guardrails) is red-teamed before go-live, not only the base model.
- ☐ Each finding records the exact input, the exact output, the judgment rule, the reproduction rate, the severity with its reasoning, and the control it maps to.
- ☐ Findings cite the frameworks your reviewers already use: OWASP GenAI Top 10, MITRE ATLAS, NIST AI RMF, and HIPAA or CJIS where they apply.
- ☐ Testing repeats on a schedule and after every model, prompt, tool or corpus change, and the evidence is kept where an auditor can find it.

8. Operations and incident response

Agents fail at machine speed. The plan has to exist before the first incident.

- ☐ There is a kill switch that disables the agent's tools (or the agent) without a deploy.
- ☐ An incident playbook names what to preserve (logs, the carrier document, the conversation), who to notify, and how to decide whether data left the boundary.
- ☐ Vendor model and prompt changes are tracked as changes, with a retest condition.
- ☐ The people who rely on the agent's output know it can be steered by the documents it reads, and know how to report a suspicious result.

Nothing on this list makes an agent safe on its own, and no vendor, RedLens included, can certify that it is. What the list does is turn a vague worry into a list of answerable questions, with the evidence that each answer needs. Print it, work through it with the team that owns the agent, and keep the filled-in copy with the change record.

Free assessment: generate real adversarial prompts against the agent you describe at <https://redlens.ai/poc> | This checklist: <https://redlens.ai/guides/ai-agent-security-checklist>