oktsec

Are Your AI Agents Actually Secure?

The Security Checklist Every Team Needs in 2026

If you use Cursor, Copilot, Claude Code, or ChatGPT, you are running AI agents. Only 14.4% go live with full security approval. This checklist gives you actionable steps for developers, startups, and enterprise teams.

85.6%
of AI agents lack full security review
Gravitee, AI Agent Security Survey 2026

40%
more secrets leaked with AI assistants
GitGuardian, State of Secrets 2025

23.8M
secrets leaked on GitHub (2024)
GitGuardian, State of Secrets 2025

Gustavo Aragón / Oktsec | March 2026

Full guide (47 pages): oktsec.com/ai-agent-security

The State of AI Agent Security

36.8%
of AI agent tools contain security flaws
Snyk, ClawHub Security Analysis 2026

76
confirmed malicious tools in marketplaces
Snyk, ClawHub Security Analysis 2026

< 2 yrs
from proof-of-concept to production attacks
Incident timeline, documented CVEs

Wait, am I even running AI agents?

If you use **Cursor, GitHub Copilot, Claude Code, Windsurf, Cline, or ChatGPT with plugins**, yes, you are. These tools connect to external services (called MCP servers) that can read files, run commands, and access APIs on your behalf. That makes them AI agents. Most people don't realize their coding assistant has the same access as a developer with admin permissions.

5 things you need to know

- **Prompt injection is just the entry point.** Attackers don't stop at tricking your AI. They use it to steal credentials, spread to other systems, and take actions. 21 of 36 documented attacks go through 4+ stages. *(Schneier et al., "The Promptware Kill Chain," Jan 2026)*
- **The tools your AI connects to are compromised.** 36.82% of plugins and tools in AI agent marketplaces contain security flaws. 76 were confirmed malicious. *(Snyk, ClawHub Security Analysis, 2026)*
- **AI coding assistants leak your secrets.** Repos using AI assistants show 40% higher rates of accidentally committing passwords and API keys. *(GitGuardian, State of Secrets Sprawl 2025)*
- **Real worms targeting AI tools exist today.** In Feb 2026, 19 npm packages were caught installing hidden backdoors into Cursor, Claude Code, and Windsurf, stealing credentials with a 48-to-96-hour delay to avoid detection. *(Socket Threat Research Team, Feb 2026)*
- **Every major security standards body is sounding the alarm.** OWASP, NIST, CSA, MITRE, and OpenSSF all published AI agent security frameworks in 2025-2026. This is not hype. It is coordinated urgency.

Scanning data: Aguara Watch observatory, 58,000+ skills across 7 registries, scanned 4x daily with 188+ detection rules. | Full methodology and 50+ sources in the complete guide.

For Everyone Using AI Tools

You use Cursor, Copilot, Claude Code, or ChatGPT? This is for you.

- [ ] **Read what a tool does before you approve it** — CRITICAL
  Your AI assistant asks permission to use tools. Those tool descriptions can contain hidden instructions that hijack what the AI does next. Always read before clicking 'Allow'.
- [ ] **Lock tool versions. Don't auto-install the latest** — CRITICAL
  Commands like 'npx -y package' grab whatever version is newest. If someone hacks the package, your AI runs their code. Pin versions like package@1.2.3.
- [ ] **Never paste API keys or passwords into AI-assisted code** — HIGH
  AI assistants reproduce patterns from their training data, including real secrets. Use environment variables or a vault instead.
- [ ] **Turn on secret scanning in your code repos** — HIGH
  Tools like GitGuardian or pre-commit hooks catch accidentally committed passwords before they reach GitHub. Essential when AI generates code for you.
- [ ] **Run AI tools in a sandbox, not on your main machine** — HIGH
  If your AI coding assistant gets compromised, a sandbox limits the damage. Use containers or virtual environments to isolate AI tools from your real files.
- [ ] **Don't auto-approve tool actions. Review first** — HIGH
  Some tools auto-execute commands without asking. Turn that off. Review what your AI agent wants to run (file reads, API calls, shell commands) before it does it.
- [ ] **Review AI-generated code before committing** — HIGH
  AI can reproduce real API keys and passwords from training data without knowing it. Always review what you are about to push.
- [ ] **Check tools before installing them** — MEDIUM
  Use watch.aguarascan.com to see if a tool has known security issues. Think of it like checking app reviews before downloading.
- [ ] **Keep your AI tools updated** — MEDIUM
  AI coding tools get security patches just like your browser or OS. Outdated tools may have known vulnerabilities that attackers can exploit.

For Startups Shipping AI Agents

Building AI products? Bake security in now. It costs less than fixing a breach later.

- [ ] **Give each agent only the permissions it needs** — CRITICAL
  An agent that helps with customer support doesn't need access to your database or deployment pipeline. Restrict each agent to its specific tools.
- [ ] **Lock down your tool versions like any other dependency** — CRITICAL
  Treat AI agent tools the same way you treat npm packages or Docker images: pin versions, verify checksums, update intentionally.
- [ ] **Scan agent tools in your CI/CD pipeline before deploying** — HIGH
  Static scanners (like Aguara Scanner) catch dangerous patterns in tool definitions before they reach production. Fail builds on high-severity findings.
- [ ] **Isolate each tool in its own container** — HIGH
  If one tool gets compromised, it shouldn't be able to reach your entire system. Run tools in separate containers with minimal permissions.
- [ ] **Use short-lived tokens, not long-lived API keys** — HIGH
  AI agents with permanent API keys are a goldmine if compromised. Rotate credentials frequently and use time-limited tokens.
- [ ] **Log everything your agents do** — HIGH
  Record every tool call: which agent, which tool, what arguments, when, and the outcome. Without logs, you can't investigate incidents.
- [ ] **Know your agent inventory** — HIGH
  Document which agents exist, what permissions they have, who deployed them, and when they were last reviewed. You can't secure what you don't know about.
- [ ] **Set rate limits on agent actions** — MEDIUM
  A compromised agent can make thousands of API calls per second. Rate limits prevent runaway damage and make anomalies visible.
- [ ] **Validate what the agent sends to each tool** — MEDIUM
  AI agents construct their own parameters. Those parameters can include file path traversals, SQL injection, or commands. Validate server-side.

For Enterprise Security Teams

Managing AI at scale? Defense-in-depth across all 7 attack stages.

- [ ] **Deploy a security gateway between agents and tools** — CRITICAL
  A gateway (like an MCP Gateway) gives every agent a verified identity, enforces per-agent permissions, and scans all traffic. Think of it as a firewall for AI.
- [ ] **Create an AI agent governance policy** — CRITICAL
  63% of breached organizations lack or are still developing AI governance policies (IBM, 2025). Define who can deploy agents, what approvals are needed, and what data agents can access.
- [ ] **Adopt a Zero-Trust approach for agents** — CRITICAL
  Don't trust agents by default. Verify identity on every action, limit each agent to explicit capabilities, and separate monitoring from execution.
- [ ] **Map your agents against OWASP's Agentic Top 10 risks** — HIGH
  OWASP published 10 risk categories (ASI01-ASI10) specific to AI agents in 2026. Use them as a checklist to audit your agent deployments.
- [ ] **Feed agent logs into your security monitoring (SIEM)** — HIGH
  Agent attacks follow a 7-stage pattern. Create detection rules for each stage: unusual tool access, credential reads, data exfiltration patterns.
- [ ] **Run red team exercises that target your AI agents** — HIGH
  Traditional pentests don't test prompt injection, tool poisoning, or agent-to-agent lateral movement. Add AI-specific attack scenarios to your security testing.
- [ ] **Require security review for every new agent deployment** — HIGH
  Only 14.4% of agents have full security approval (Gravitee, 2026). Make review mandatory before any agent goes to production.
- [ ] **Isolate agents at the network level** — MEDIUM
  Each agent should have its own network namespace with strict egress rules. Monitor DNS queries. Data exfiltration often uses DNS tunneling.
- [ ] **Monitor for tool tampering between scans** — MEDIUM
  A 'rug pull' is when a tool silently changes its behavior after being approved. Compare tool definitions across scans using hash-based detection.
- [ ] **Write an AI agent incident response playbook** — MEDIUM
  When an agent is compromised, your team needs to know: how to contain it, what logs to pull, and how to trace what the agent accessed. Plan this before an incident.

oktsec

Get the Complete Guide

47 pages of data-backed strategies for securing AI agents in production

oktsec.com/ai-agent-security

✓ The full Promptware Kill Chain analysis (7 stages, 36 studies)
✓ OWASP Top 10 for Agentic Applications deep dive with detection rules
✓ Complete incident timeline: every documented attack through March 2026
✓ Implementation checklists with copy-paste configurations
✓ Defense-in-depth framework with 3 deployment layers

Gustavo Aragón / Oktsec
oktsec.com | aguarascan.com | watch.aguarascan.com

Share this checklist. Tag your security team.