

Arcade

Security checklist for production AI agents

Introduction

This checklist is a practical set of security standards for deploying multi-user agents into production. Security standards commonly lag the adoption of new technologies. Yet, the speed at which AI is being deployed and the depth at which agents are connecting into sensitive business systems to act on critical data sets AI apart.

While [NIST](#), [OpenID](#), the [Cloud Security Alliance](#), [OWASP](#), and others are building agentic security guidance, the need to securely govern agents acting on production systems is now.

The checklist has five pillars with 24 core standards all multi-user agents should meet prior to acting on production systems and five additional advanced standards for higher security environments.

Identity & authentication

- ❑ Standards-based authentication (CORE). Leverage existing standards for agent identity such as OAuth 2.1 with PKCE and OpenID Connect, with SPIFFE/SPIRE for workload identity, until standards emerge treating agents in a first-class manner.
- ❑ No static API keys (CORE). Static keys treated as deprecated for agents, which make thousands of calls per hour where a key can be exfiltrated by prompt injection and replayed indefinitely.
- ❑ Credential storage and key control (CORE). Agent credentials live in a dedicated store separate from the agent, encrypted at rest and in transit, never present in prompts, model context, code, or logs. The agent receives a credential only at the moment of execution. Where assurance demands it, the customer controls the keys that encrypt the store.

- ❑ Token lifecycle management (CORE). Token lifetimes measured in minutes rather than hours or days, with documented maximums. Credentials rotate on schedule and can be revoked on demand, and terminating an agent revokes its sessions and tokens everywhere at once.
- ❑ Managed OAuth flows (CORE). An OAuth broker handles client registration, redirect URIs, refresh, and multi-tenant apps; developers never manage plaintext credentials.
- ❑ Bind each token to its holder (ADVANCED). Tokens bound to the holder via DPoP or mutual TLS, so an exfiltrated token cannot be replayed.

Authorization & delegation

- ❑ Delegation, not impersonation (CORE). Delegated agent actions on behalf of users are auditable back to both the human principal and the agent actor (e.g. the OAuth act claim).
- ❑ Preserve the requesting user's identity on every downstream call (CORE). The original user's identity travels with each service hop (via OAuth Token Exchange) instead of flattening into a shared service identity.
- ❑ Per-user, per-action least privilege (CORE). Each action authorized at the intersection of the agent's and the requesting user's permissions, never the union. Least privilege is enforced deterministically by scoping the agent's tool list to each user, so agents can only access tools the user has authorization for. Tokens scope-limited, audience-restricted, effective-dated; no standing broad privileges.

- ❑ Multi-principal and shared-context resolution (CORE). When one agent operates in a shared context with users of different permissions, it resolves whose authority applies before acting and never leaks one user's data to another.
- ❑ Just-in-time authorization (CORE). Privileges provisioned at the moment of need, not granted broadly at deployment.
- ❑ Runtime authorization in the execution loop (CORE). Each tool call and state-changing action re-evaluated against current policy and context, not authorized once at session start.
- ❑ Human-in-the-loop at the policy layer (CORE). High-risk actions trigger out-of-band approval (CIBA with Rich Authorization Requests); routine actions don't prompt, avoiding consent fatigue.
- ❑ No self-escalation (CORE). Agents cannot expand their own scopes beyond an explicit pre-authorized maximum, even under a prompt-injected task.
- ❑ Maintain permissions at each sub-agent hop (ADVANCED). When agents spawn sub-agents, permissions never expand at any hop and stay traceable to the originating user.

Governance

- ❑ Dedicated policy decision point (CORE). Authorization decisions live in a dedicated authorization server or PDP, separated from agent code, so policy changes don't require redeploying agents.
- ❑ Deterministic controls for hard limits (CORE). Hard rules (spending caps, no external email after reading untrusted data) are enforced deterministically with context: who is asking, which tool, how sensitive the operation.
- ❑ Enforce policy before and after every tool call (CORE). Before a call runs, verify that this agent is allowed to take this action on behalf of the user who asked for it. After it returns, inspect the result before it reaches the model, redacting sensitive data and stripping injection payloads. Disallowed calls are denied outright rather than merely logged, so the downstream system never sees a blocked request.
- ❑ Uniform enforcement across every client path (CORE). Controls apply identically across all AI clients (IDEs, desktop and web apps, SDKs, CI); any client outside the control plane is a bypass route.

- ❑ Extend existing policy, do not recreate it (CORE). Agent authorization decisions draw on the identity, entitlement, and data-classification systems the enterprise already runs, and policy is expressed and versioned there rather than in a separate agent-only silo.
- ❑ Centralized configuration distribution (CORE). Agent and AI-client configurations are pushed and version-controlled from a central point so a fix propagates immediately.
- ❑ Multi-agent and A2A governance (ADVANCED). Agent-to-agent flows carry user identity context end to end, scope attenuates at each hop, and human oversight applies at decisive points.

Audit

- ❑ Attribution-complete audit coverage (CORE). The audit trail covers both what agents did and every change to the system that governed them: tool calls, policy and configuration changes, server and tool registrations, and credential grants and revocations.
- ❑ Every record carries the agent's identity, who authorized the action, the specific delegation grant, and the context.
- ❑ Full tool-call capture (CORE). Inputs, outputs, parameters, and timestamps for every tool invocation, with intermediate reasoning where feasible.
- ❑ Logs the agent cannot alter (CORE). The record of what an agent did lives outside the agent's reach, so agents cannot edit or delete the evidence of their own actions.
- ❑ SIEM integration (CORE). Agent activity streams to existing security tooling in real time for critical events.
- ❑ Delegation-chain tracing (ADVANCED). Correlation IDs tie agent, user, tools, inputs, and outputs together across multi-step and multi-agent runs.

Inventory & registry

- ❑ Unified agent and tool registry (CORE). One registry of record for every agent, MCP server, connector, tool, and skill or instruction file in use, each carrying a named owner, a version, and its provenance, so every team building agents knows what is approved before they ship.
- ❑ Tool and skill scanning (ADVANCED). Tool descriptions and skill code scanned for poisoning, hidden characters, and embedded instructions, and re-scanned after modification.

How Arcade helps

Arcade is the actions runtime that governs enterprise AI agents. We enforce your policies on every action, bring every agent under one control plane, and clear the path to production. Some of the largest enterprises in the world already trust Arcade to help them meet these agent security standards.

Enforce. Most of these controls come down to one question: Can this agent, on behalf of this user, take this action right now? Arcade answers it on every call, outside the model. Agents act as delegates, never principals, authorized at the intersection of the agent's scope and the user's permissions, against the IdP you already run. Agents get just-in-time access with exactly the privileges they need and hold none afterward. Credentials stay vaulted, so the agent never holds one, and the token lifecycle is managed by Arcade. Your DLP, SIEM, and IdP policies fire on each action, and you can add custom contextual policies for risks unique to agents.

Govern. Every action is authorized and written to one audit trail, attributed to a named user, agent, and system. The record does not depend on how well the agent or the MCP server logs on its own. Everything routed through Arcade emits the same rich log, the request, every authorization and governance check with its verdict, and exactly what the agent saw after redaction. Those logs stream to your SIEM, so answering for one action is a single query instead of a manual pull across a dozen systems. The same control plane is where you grant, revoke, and version tools and manage the whole estate through a single registry.

Enable. You review the runtime once, not every agent. Every team inherits your authorization model, policies, and audit trail by default. Those controls hold no matter which framework, client, model, or agent they choose, so the golden path is also the governed path.

Get in touch to secure your multi-user agents

See how Arcade covers each item on this checklist for your organization.

Book a demo with a member of our technical staff



Arcade