What is AI agent identity?

AI agent identity is the practice of giving each AI agent its own distinct, verifiable identity and credentials, separate from the humans it works for and from other agents, then granting it only the permissions its task needs. Good agent identity combines short-lived, scoped tokens, delegation that records which user the agent acted for, just-in-time access for sensitive actions and a complete audit trail. OWASP lists the failure mode as Identity and Privilege Abuse (ASI03): agents reusing broad shared credentials or cached user tokens.

HOW IT WORKS

01 What goes wrong without it

OWASP's Top 10 for Agentic Applications names this Identity and Privilege Abuse (ASI03). The common patterns:

- Over-privileged shared credentials. An agent that only needs to read tickets holds an admin token, so a single indirect prompt injection can close, delete or exfiltrate everything.
- Borrowed user sessions. The agent acts with the user's full rights everywhere, not only for the task the user asked about.
- Credential leakage. Long-lived keys end up in prompts, tool output, logs or memory, where another user or an attacker can retrieve them.
- Inter-agent trust. One agent accepts instructions or tokens from another without verifying it, so compromising the weakest agent reaches the strongest.
- No attribution. After an incident, nobody can reconstruct which agent made a change, which user it was serving or what prompt led to it.

This is the identity side of excessive agency.

02 How to build agent identity well

- One identity per agent (and per deployment environment), registered with an owner, a purpose and an expiry.

SOURCES

[1] OWASP Top 10 for Agentic Applications (2026)

[2] MCP specification: Authorization

[3] RFC 8707: Resource Indicators for OAuth 2.0

[4] NIST SP 800-207: Zero Trust Architecture

- Delegated, not borrowed, access. When an agent acts for a user, use OAuth flows that produce a token carrying both identities and only the scopes the task needs. For the Model Context Protocol (MCP) servers, the specification requires tokens bound to the specific server (RFC 8707) and forbids passing a user's token through to other services.
- Short-lived, narrowly scoped credentials. Minutes or hours, not months. Start with read-only scopes and step up only when a privileged action is attempted.
- Just-in-time and human approval for high-impact actions such as payments, deletions, permission changes and external sharing.
- Keep secrets out of the model. Credentials should be attached by the runtime or an an MCP gateway, never placed in prompts, context or memory.
- Log with full attribution: agent ID, user ID, tool, arguments, result and the trace that led to the call.
- Apply zero trust between agents. Verify every request, including those from other agents; trust is never implied by network position.

03 How to test it

Hijack the agent in a test environment, through prompt injection or a poisoned tool, and see what its credentials let you reach. Check whether tokens are scoped to the task or reused across services, whether they appear in tool output, logs or memory, whether one tenant's agent can act on another tenant's data, and whether audit logs let you reconstruct the attack afterwards. The answer to "how far can a fully compromised agent go?" is the real measure of your identity design.

Test your AI application before an attacker does.
securelayer7.net/learn/ai-security/ai-agent-identity

Open online