Certified Secure AI Agent Developer

Curriculum - 2026

This training the team builds a working agent, an MCP connection, a RAG pipeline, and a skill on day one — then spends the rest of the program attacking and hardening what they built. By the end, engineers can secure agent identity, memory, retrieval, tooling, and secrets by default — not bolt controls on after something breaks — and prove it under a proctored capstone with a live defense.

D1 — LLM & Agent Fundamentals + Prompt Injection

Learn why the data/instruction boundary doesn't exist inside a model, then execute direct and indirect prompt injection (OWASP LLM01) against an agent built in the opening build sprint, and watch it fail.

Topics covered

- Why the data/instruction boundary doesn't exist inside a transformer-based model
- Direct prompt injection: crafting and executing a working exploit
- Indirect prompt injection (OWASP LLM01): injecting through tool outputs, documents, and retrieved content
- Why traditional input validation doesn't stop injection, and what does

LAB: Break Your Own Agent Execute both direct and indirect prompt injection against the agent built in the opening build sprint, and document exactly where it failed.

D2 — OWASP Agentic Top 10 — Attack & Defend

Work the OWASP Agentic Top 10 as a lab track: excessive agency, ASI03 identity and privilege abuse, ASI06 memory and context poisoning, guardrails, and semantic validation.

Topics covered

- Excessive agency: scoping what an agent is allowed to decide vs. execute
- ASI03 — identity and privilege abuse in agentic systems
- ASI06 — memory and context poisoning
- Guardrails and semantic validation as a defense layer, not a checkbox

LAB: Agentic Top 10 Attack Range Work through a lab track exploiting excessive agency, identity/privilege abuse, and memory poisoning against a shared vulnerable agent, then apply the matching defenses.

D3 — Agent Identity, Access & Non-Human IAM

Give every agent a short-lived, least-privilege identity — non-human IAM and role-based access control on retrieval — instead of one shared API key.

Topics covered

- Why one shared API key per agent is a systemic risk, not a shortcut
- Designing short-lived, least-privilege identity for non-human actors
- Role-based access control for retrieval and tool calls
- Auditing and revoking agent identity at scale

LAB: Give Every Agent Its Own Identity Replace a shared static credential with short-lived, least-privilege non-human IAM and RBAC-scoped retrieval access on the agent built earlier, then verify least privilege holds under test.

D4 — Memory, Context & Retrieval Security

Secure the RAG pipeline and context window an attacker can poison as easily as a database.

Topics covered

- How a RAG pipeline and context window can be poisoned like a database
- Attacking your own retrieval pipeline: injected documents, embedding manipulation
- Context window hygiene: what should never enter it in the first place
- Defenses: retrieval filtering, provenance tracking, and context sanitization

LAB: Poison and Patch the Pipeline Poison the RAG pipeline built earlier with a malicious document, trace how it reaches the agent's context, then implement filtering and provenance controls to stop it.

D5 — MCP & the Agent Supply Chain

Defend against tool poisoning and shadow MCP servers with scanning, allowlisting, and gateways — the supply-chain problem most teams haven't noticed yet.

Topics covered

- Tool poisoning: how a malicious or compromised MCP server takes over an agent
- Shadow MCP servers: the supply-chain risk most teams haven't inventoried
- Scanning, allowlisting, and gateway patterns for MCP tool access
- OWASP Top 10 for MCP as a working checklist

LAB: Defend the MCP Supply Chain Stand up a gateway/allowlist in front of the MCP connection built earlier, detect and block a tool-poisoning attempt, and document the control.

D6 — Agent Secrets & Runtime Containment

Scan the skill supply chain, run secretless agents, and contain what an agent can do at runtime instead of trusting it not to misbehave.

Topics covered

- Scanning the skill supply chain before an agent ever loads a skill
- Running secretless agents: eliminating static credentials from the runtime entirely
- Runtime containment: bounding what an agent can do, not just what it's told to do
- Building the threat model and self-attack write-up used in the capstone

LAB: Build the Secretless, Contained Agent Reconfigure the agent, MCP connection, and skill built across the program to run secretless with runtime containment, then write the threat model and self-attack summary that feeds into the capstone.

CERTIFICATION EXAM DETAILS

Part 1 - Knowledge & Judgment Exam

Scenario-based questions tied to a specific environment or proctored artifact — not lookup trivia.

Part 2 · Challenge Exam

Four to six graded challenges in fresh, randomized, browser-based lab environments, auto-graded against a verifiable end state.

Part 3 · Proctored Capstone + Live Defense

One continuous proctored sitting. Candidates build a hardened, secure-by-default agent — identity, MCP hygiene, RAG access control, secretless credentials, and containment — plus a threat model and a self-attack written in-session, then defend it live.

To pass, candidates must clear a minimum threshold in every instrument — no averaging a strong knowledge score past a weak capstone — plus a pass on the live defense.

Certification is valid for 24 months. Candidates leave with a verifiable digital badge for LinkedIn.