Securing LLM Applications and AI Agents

SEC502

Secure LLM applications and AI agents as an engineer: threat-model what they read and can do, prove each weakness by its effect, build controls that hold whatever the model is persuaded to do, gate releases, and respond to incidents.

Modules	21
Lessons	119
Phases	4
Free preview	Orientation and LLM Application Architecture (6 lessons)

The method

Controls that hold when the model does not. Model behavior can fail, so every module builds a control that stays enforceable whatever the model is persuaded to do, outside the model, and measures the defenses that only change the odds, such as markers, prompts and guardrails, for what they are.

Two applications, three places to practice. A knowledge assistant over a shared index and a service desk agent with tools, memory and approvals carry the whole course. The AI Range in your browser proves each weakness, the Python lab pack is working code you fix until its check passes, attacks and legitimate work together, and a month of recorded application telemetry is investigated in the query console.

Named, versioned, and checked. Each weakness is named against the OWASP Top 10 for LLM Applications 2026, the OWASP Top 10 for Agentic Applications 2026 and MITRE ATLAS, with the version on every page that relies on one. The course ends on a 40-question exam that includes a case study of each application.

Full syllabus

Phase 1: Foundations

Orientation and LLM Application Architecture FREE PREVIEW

The engineer's job and the principle behind the course, how an LLM application is built, three kinds of failure, testing safely, the lab and its evidence, and the review, prove, fix and verify loop run once.

1. 0.1 What This Course Is For
2. 0.2 How an LLM Application Is Built
3. 0.3 Application Security, Model Safety and Output Reliability
4. 0.4 Testing Safely
5. 0.5 Your Lab and How Evidence Is Reported
6. 0.6 A First Weakness, Proved and Fixed

Threat Modeling and Trust Boundaries

Threat-modeling the service-desk agent before it goes live: its design read from the code, trust boundaries, abuse paths from writer to harm, coverage against OWASP, ATLAS and ATT&CK, the threat-model board, and a ranked, written threat model.

1. 1.1 Reading the Agent's Design
2. 1.2 Trust Boundaries
3. 1.3 From Text to Harm: Abuse Paths
4. 1.4 Naming What You Found
5. 1.5 Threat-Modeling the Agent on the Board
6. 1.6 Ranking Abuse Cases and Writing the Threat Model

Phase 2: Inputs, Data and Outputs

Prompt Injection and Instruction Manipulation

How a customer's ticket steers the service-desk agent, the forms injection takes, attacks that blend in, the layers that lower the rate and how to measure them, and the call check that holds, built and verified in the lab.

1. 2.1 How Injection Reaches the Agent
2. 2.2 The Forms an Instruction Takes
3. 2.3 Riding a Real Task, and Multi-Turn
4. 2.4 Defenses That Lower the Rate
5. 2.5 Measuring Injection
6. 2.6 The Control That Holds

Sensitive Data and Tenant Isolation

How a shared knowledge assistant hands one tenant's documents to another, the confused deputy behind it, where the permission check belongs, the check and the cache key built in the lab, and the result verified as evidence.

1. 3.1 What the Index Can Read, and What Each User May Read
2. 3.2 Proving It With an Ordinary Question
3. 3.3 Where the Check Belongs
4. 3.4 Implementing the Permission Check
5. 3.5 The Other Routes Tenant Data Takes
6. 3.6 Verifying and Recording Isolation

Securing RAG and Knowledge Ingestion

Who can put text into the knowledge assistant's index: the write paths, poisoning them, the editor, tenant and second-approval gates, provenance and removing every copy, and the boundary verified as evidence.

1. 4.1 How Content Gets Into the Index
2. 4.2 Poisoning the Index
3. 4.3 Ingestion Approval
4. 4.4 Provenance
5. 4.5 Instructions Inside Approved Content
6. 4.6 Verifying the Ingestion Boundary

Handling Model Outputs Safely

What happens to the model's output after it is written: rendering replies safely, generated URLs, queries and paths, schema and semantic validation, and output handling verified as evidence.

1. 5.1 Model Output Is Untrusted Input
2. 5.2 Rendering Safely
3. 5.3 Generated URLs
4. 5.4 Generated Queries
5. 5.5 Generated Paths and Validation
6. 5.6 Verifying Output Handling

Phase 3: Agents and Their Actions

Agent Identity and Authorization

Whose authority the agent uses when it calls an API: application versus delegated access, the confused deputy, one scope and one record per call, tokens as credentials, and the identity boundary verified as evidence.

1. 6.1 Whose Authority the Agent Uses
2. 6.2 Application Access and Its Blast Radius
3. 6.3 Delegated Access
4. 6.4 One Approved Dataset
5. 6.5 Credentials, Lifetime and the Agent With No User
6. 6.6 Verifying the Identity Boundary

Tool and MCP Security

Securing an agent's tools over MCP: a token per server, no passthrough, audience checks at the server you run, tool definitions approved by hash, and the integration verified as evidence.

1. 7.1 What MCP Adds to the Agent
2. 7.2 Tokens Across Hops
3. 7.3 Tool Definitions Are Input
4. 7.4 Which Tools the Model Sees
5. 7.5 Servers You Run and Servers You Don't
6. 7.6 Verifying the MCP Integration

Containing Agent Actions

Securing what an agent does: approvals bound to the exact action and shown from its arguments, destinations matched against a registry, a budget per request, an idempotency key, and the approval matrix.

1. 8.1 What the Agent Can Now Do
2. 8.2 An Approval Bound to the Action
3. 8.3 What the Approver Is Shown
4. 8.4 Where the Agent May Send
5. 8.5 Stopping a Runaway Workflow
6. 8.6 Verifying Action Containment

Memory and Multi-Agent Security

Securing what an agent remembers and who it listens to: provenance on every memory write, recall scoped by customer and expiry by kind, handoffs accepted only from known agents for their own field, and agent-to-agent trust by role.

1. 9.1 What the Agent Remembers, and Who It Listens To
2. 9.2 Provenance on Every Memory Write
3. 9.3 Scope and Expiry
4. 9.4 Another Agent's Output
5. 9.5 Agents That Talk to Agents
6. 9.6 Verifying Memory and Handoffs

Phase 4: Releases and Operations

Supply Chain and Deployment

Securing what an agent loads and the updates proposed for it: files verified against their publisher's record, formats that run nothing on load, a model swap gated on every earlier case, backdoors and what holds them, and connector and configuration changes reviewed by reach.

1. 10.1 What an Update Brings In
2. 10.2 Verifying What You Load
3. 10.3 A Model Swap Is a Release
4. 10.4 Poisoning and the Backdoor Tests Miss
5. 10.5 Connector and Deployment Changes
6. 10.6 Verifying the Release Gate

Security Evaluation and Release Gates

Turning security checks into an evaluation that can decide a release: six measures with their conditions, enough runs for the claim, judging by effects, guardrails measured beside controls, a suite grown with variants, and a gate for every change.

1. 11.1 What an Evaluation Must Report
2. 11.2 Trials, and What Zero Failures Means
3. 11.3 Judging by Effects
4. 11.4 Guardrails Measured Beside Controls
5. 11.5 Growing the Suite
6. 11.6 Gating Every Change

Monitoring and Incident Response

Investigating and responding to incidents in LLM applications and agents: what to record, scoping across services by trace, containing without destroying evidence, deciding what is an incident, and verifying containment by its cause.

1. 12.1 What an Agent Must Record
2. 12.2 Scoping a Memory Incident
3. 12.3 Following Data Out
4. 12.4 Containing It
5. 12.5 Is This an Incident
6. 12.6 Verifying Containment and Closing

Phase 0: Course Resources

LLM Application Security Cheatsheets

One sheet per area of LLM application and agent security: the question, the control or check that settles it, how to use it and what it does not prove.

1. Architecture and Trust Boundaries
2. Prompt Injection
3. Sensitive Data and Tenant Isolation
4. RAG and Ingestion
5. Output Handling
6. Agent Identity and Authorization
7. Tools and MCP
8. Containing Agent Actions
9. Memory and Multi-Agent
10. Supply Chain and Deployment
11. Evaluation and Release Gates
12. Investigation and Response
13. Lookup Values

LLM Application Security Cookbooks

Eight runbooks for the controls an LLM application or agent needs, each built in the lab and tested from both sides.

1. Retrieval That Respects Authorization
2. An Approval Bound to the Action
3. Tool-Call Checks Independent of the Model
4. Memory Writes With Provenance and Expiry
5. Safe Output Handling
6. Assessing a Model, Connector or Dependency Update
7. An Evaluation Suite as a Release Gate
8. Incident Response From Findings

Lab Setup

Installing the SEC502 lab, the ai-app workspace the console queries, and checking both work before the first exercise.

1. Installing the Lab
2. The ai-app Workspace
3. An Optional Local Model
4. Verify the Lab

Walkthroughs

Three cases worked end to end across the modules, including the wrong turns, each ending in a scorecard, a report card and the point that transfers.

1. Reviewing the Service Desk Agent for Release
2. A Poisoned Index to a Verified Fix
3. A Cheaper Model and a Guardrail

Playbooks

Eight playbooks, one per thing that lands on an LLM application team: what to check first, how to contain it, and what the record must say.

1. Prompt Injection Reported
2. Data Disclosed in an Answer
3. Memory Poisoning Found
4. A Runaway Agent or Cost Spike
5. An MCP Server or Tool Compromised
6. The Model or Vendor Changes
7. A Guardrail Bypass Reported
8. Hidden Context Exposed

Playground

The range to practice on freely, and every tool scenario, predict block and console query in the course, by module.

1. Free Play and Everything to Operate

References

The standards, lists, protocols and tools the course relies on, each with the version it was read at.

1. Standards and Frameworks
2. Telemetry, Protocols and OAuth
3. Tools and Versions

Course Completion

Course Exam

Securing LLM Applications and AI Agents end-of-course exam: 40 questions in four parts, from threat modeling to incident response, ending on a case study.

1. Course Completion. Securing LLM Applications and AI Agents

Generated 2026-11-06 from the published course. The current syllabus is always at ridgelinecyber.com/training/courses/securing-llm-applications-agents/