800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

AI Application Security: Securing LLMs, RAG, and Agents

Applied Technology Academy

Secure the AI applications your own organization is building and deploying, over three instructor-led days of attacking and then hardening a real application. Developers and security professionals work with an enterprise knowledge assistant to investigate prompt injection, unauthorized data access and unsafe tool use, then fix what they found and prove the fix held. Teams leave with a repeatable method for assessing an AI feature before release and monitoring it afterwards.

LEVEL	DURATION	DELIVERY
Intermediate	**3 days**	**Instructor-led**

Course Overview

- Three instructor-led days spent securing a working AI application, not discussing AI risk in the abstract.
- Students attack an enterprise knowledge assistant, then harden it and prove the fix held.
- Roughly half lecture and instructor demonstration, half student labs.
- Every exercise ends in evidence: explain the failure, apply a control, show what changed.
- Guided exercises use synthetic enterprise data and controlled services.
- A short model-risk comparison separates application flaws from model evasion, poisoning, extraction and inference attacks.
- Available in person or Online Live.

Course Outline

- **Day 1: Threats and trust boundaries**
- **Module 1. AI application architecture and threat modeling**
  - Tokens, context windows, embeddings, retrieval and tool calls, and how training and fine-tuning differ from inference. Identifies the roles of application code, identities and infrastructure, discovers input paths and reachable data and APIs, then connects attacker access to business impact using current OWASP and MITRE guidance.
  - Hands-on lab: Map the knowledge assistant's data flows and trust boundaries, identify attack paths and propose testable security requirements.
- **Module 2. Prompt injection and application controls**
  - Direct, indirect and multi-turn injection through user requests, documents and tool responses. Separates content-policy jailbreaks from unauthorized application actions, and examines why prompts and keyword filters protect so little - and where application controls have to enforce policy instead.

800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

  - Hands-on lab: Discover and reproduce an injection path, add a control, then compare a supplied multi-turn attempt, malicious retrieved content and legitimate requests.
- **Module 3. Securing retrieval-augmented generation**
  - Protects document ingestion and retrieval with trusted identity, access controls, tenant separation, provenance and data-handling rules. Examines poisoned content and the limits of answer citations.
  - Hands-on lab: Diagnose a retrieval authorization flaw, enforce access before content reaches the model, then test permitted and prohibited access.
- **Module 4. Agent and tool security**
  - Constrains tool permissions, validates arguments, enforces authorization and binds approval to the action actually being executed. Covers tool-response tampering and external integrations, including MCP server trust and scoped credentials, and introduces the persistent-memory risks picked up on day three.
  - Hands-on lab: Prevent an injected request from causing an unauthorized tool action while preserving approved work.
- **Day 2: Controls and security evidence**
- **Module 5. Safe outputs, secrets and resource limits**
  - Treats generated content as untrusted input to everything downstream. Traces how an LLM integration exposes ordinary web and API vulnerabilities through unsafe HTML, queries, commands and URLs, then applies safe handling, restricts external destinations, protects secrets and bounds requests and agent loops.
  - Hands-on lab: Reproduce and correct one unsafe output path, then verify the preconfigured resource and external-access restrictions.
- **Module 6. AI security testing and regression evaluation**
  - Builds a scoped assessment from the threat model, combining manual testing, a preconfigured open-source scanner and application-specific regression cases. Evaluates attack outcomes, legitimate-task completion, false positives and operational cost, separating tuning examples from held-out tests and interpreting model variability.
  - Hands-on lab: Run a prepared scan, manually verify one finding, extend a regression suite, then compare vulnerable and hardened behavior across repeated model trials.
- **Module 7. AI supply chain and secure deployment**
  - Compares security responsibilities across managed model APIs, self-hosted models and cloud or hybrid deployments. Assesses workload identity, secrets, model-loading formats, dependency provenance, artifact integrity, ingestion sources and release permissions, and separates an integrity check from any assurance that an artifact is safe.
  - Hands-on lab: Inspect a release bundle, find an unauthorized artifact or configuration change, then configure one supplied CI/CD release check.
- **Module 8. Data protection and privacy**

  - Traces data through prompts, retrieval, provider services, logs and retention systems. Applies minimization and access controls, and establishes what differential privacy, synthetic data and federated learning can and cannot demonstrate.
  - Hands-on lab: Find sensitive-data exposure in application traces, apply redaction and retention controls, then document the remaining privacy risks.
- **Day 3: Runtime defense, incidents and release decisions**
- **Module 9. Runtime monitoring and agent containment**
  - Correlates model, retrieval, policy and tool traces. Protects persistent memory and context from poisoning, detects unexpected action sequences, enforces runtime policy, and plans capability revocation, cleanup and safe termination.
  - Hands-on lab: Investigate a poisoned-context or tampered-tool-response fixture, apply a control, then verify that stopping an agent prevents a pending unauthorized action.
- **Module 10. AI incident investigation and response**
  - Uses application traces, access logs, artifact versions and retrieval history to tell malicious behavior apart from drift or an ordinary defect. Plans evidence preservation, scoped containment, communication and recovery.
  - Hands-on lab: Investigate an evidence bundle, apply a containment change, replay the attack and produce an incident brief.
- **Module 11. Model risks, release assurance and risk ownership**
  - Compares evasion, training and fine-tuning poisoning, retrieval poisoning, extraction and privacy inference under explicit attacker assumptions. Connects technical findings to organizational risk decisions using NIST guidance, and defines control owners, release evidence, escalation requirements and reassessment triggers.
  - Workshop: Review model-risk scenarios and a release evidence package, then recommend release, remediation or deferral with reasons.
- **Module 12. Integrated security capstone**
  - Combines threat modeling, attack testing, hardening, evaluation and response in a new variation of the course application.
  - Hands-on lab and debrief: Discover an unfamiliar attack path, demonstrate a failure, implement a control, run regression checks and defend a release recommendation.

Prerequisites

Participants should have:

- basic Python: functions, dictionaries, exceptions and running supplied code
- familiarity with APIs, authentication, authorization and common application-security concepts
- a basic understanding of LLM applications; a preparation guide introduces RAG and tool calling
- comfort reading JSON request and response data and running supplied commands in a terminal
- ability to run Jupyter notebooks and Python applications

Not required:

- deep-learning model training
- student GPUs or personal paid-model subscriptions
- Model access is provided through the instructor-managed lab environment.
- Environment preparation and a readiness check are completed before class.

Who Should Attend

- Application developers, application security engineers and security architects.
- AI/ML engineers and security analysts who assess or support AI-enabled applications.
- Mixed technical teams, provided everyone can read and modify small Python examples.

What You'll Learn

By the end of this course, participants will be able to:

- map an AI application's components, sensitive data, identities and trust boundaries
- discover an assistant's inputs and reachable data and APIs, then test direct, indirect and multi-turn attack paths
- distinguish application compromise from a content-policy jailbreak and verify controls at the affected boundary
- enforce retrieval permissions and prevent unauthorized cross-user or cross-tenant data access
- constrain agent tools with application authorization, validated arguments, limited permissions and approval gates
- handle model outputs safely and limit resource use, external access and secret exposure
- measure attack outcomes and legitimate-task performance with prepared tools and regression cases
- assess dependencies, model artifacts, data provenance and privacy controls before release
- monitor agent actions, protect persistent context, and verify safe termination and containment
- separate model-level attacks from application defects and choose controls that match the attacker's access
- investigate AI application incidents, verify containment and define recovery criteria
- present a security assessment with evidence, remaining risks and prioritized actions

Hands-On Work

Every module ends in a lab on the course application:

- Map data flows and trust boundaries, then propose testable security requirements
- Reproduce an injection path, add a control and compare it against a multi-turn attempt
- Diagnose a retrieval authorization flaw and enforce access before content reaches the model

- Prevent an injected request from causing an unauthorized tool action
- Correct an unsafe output path and verify resource and external-access limits
- Run a prepared scan, verify a finding by hand and extend a regression suite
- Inspect a release bundle for an unauthorized artifact, then configure a CI/CD check
- Find sensitive-data exposure in traces and apply redaction and retention controls
- Investigate a poisoned-context fixture and verify that stopping an agent blocks a pending action
- Investigate an evidence bundle, apply containment, replay the attack and write an incident brief
- Discover an unfamiliar attack path in the capstone and defend a release recommendation

Students leave with:

- completed threat models, security test cases and defensive code changes
- evaluation results, a containment procedure and an incident brief
- a reusable release-assessment checklist
- These artifacts support work on their own applications, bounded by the threats and tests actually performed.

Environment and Tools

- Python and Jupyter, with an instrumented RAG and tool-calling application and a model endpoint.
- A prepared garak scan and a small regression harness for measuring attack outcomes.
- PyRIT is introduced as an optional tool for further automated assessment.
- Students inspect real request, retrieval, policy and tool-execution evidence.
- One prepared environment with clean checkpoints, so class time goes on security work.
- Cloud platform administration and model training sit outside the required exercises.
- Tested versions and preparation instructions ship with the course materials.

Customization Options

A two-day essentials version is available:

- LLM application security foundations, threat modeling, prompt injection, retrieval access and tool permissions
- Output handling, prepared testing tools, a combined supply-chain and privacy review, and an incident exercise
- Runtime controls are introduced inside the tool and incident exercises rather than in a lab of their own
- Six lecture and demonstration hours and six student-lab hours, keeping the same half-and-half format
- Smaller guided tasks and preconfigured controls, with less depth in deployment, privacy and release assurance

- Choose the three-day course for the fuller implementation and assessment experience.

Next Steps

- Advanced AI red teaming and evaluation
- Secure AI deployment and operations
- Adversarial machine learning
- AI risk management and privacy engineering

Ready to enroll? Call 800.674.3550 or email Info@AppliedTechAc.com — private team cohorts, GSA MAS purchasing, military credentialing funding, VA GI Bill and VR&E and student financing available.