

## ENTERPRISE PROGRAM BRIEF

# AI Agent Security Engineering

Apply application security principles to retrieval, tool use, model output, and agent execution. Participants investigate controlled failure cases, enforce access and action boundaries, and produce a remediation and retest report for a supplied AI application.

VNode ITes

VAI-205

Coming soon

**DURATION**  
24 hours

**LEVEL**  
Intermediate to Advanced

**FORMAT**  
Private cohort

## AUDIENCE PROFILE

### Built for these roles

Application security engineers, cloud security engineers, experienced AI developers, and technical security reviewers.

## OVERVIEW

### Executive overview

Apply application security principles to retrieval, tool use, model output, and agent execution. Participants investigate controlled failure cases, enforce access and action boundaries, and produce a remediation and retest report for a supplied AI application.

## READINESS

### Prerequisites

- Application authentication and authorisation, API security, threat modelling, basic Python, and familiarity with LLM tool use. This is an AI specialisation for existing practitioners, not an entry-level cybersecurity course.

## PROGRAM OUTCOMES

### Capabilities your teams will gain

- Map agent-specific trust boundaries and threat paths.
- Test direct and indirect injection cases in a controlled setting.
- Enforce source and tool permissions outside model instructions.
- Constrain consequential actions and execution environments.
- Document remediation, residual risk, and regression evidence.

## ENTERPRISE CUSTOMIZATION

### Enterprise customization

Tailor this program to your organization's priorities: Threat-model and harden AI applications in an authorised training environment

- 3 learning days of 8 hours, or 6 sessions of 4 hours. Learning hours exclude breaks and include module exercises and capstone time. Setup is completed before the course.

## CURRICULUM

### Curriculum roadmap

1

MODULE 1 · 2 PROPOSED HOURS

Agent threat models and assets

A model-integrated application adds new paths for untrusted text to influence behaviour. Identify the assets, identities, sources, tools, and execution environments before testing. A threat model should explain possible consequences, not merely list attack names.

- Assets and trust boundaries
- Retrieval and tool threat paths
- Authorised assessment scope

Planned lab: Map the supplied application and identify three plausible misuse paths.

Expected output: Threat model and test scope.

## 2 MODULE 2 · 4 PROPOSED HOURS

### Injection and context risks

Instructions may arrive through user input or retrieved material. Access-approved content can still contain malicious instructions. A system prompt can reduce some risks but should not be treated as the sole enforcement mechanism.

- Direct and indirect prompt injection
- Untrusted retrieved content
- Evidence handling and misleading outputs

Planned lab: Use harmless markers to observe instruction/data confusion in the local corpus, then restore the baseline.

Expected output: Controlled findings with reproduction steps.

## 3 MODULE 3 · 4 PROPOSED HOURS

### Identity and source access controls

Permission decisions belong in application and platform controls. Filter evidence before it reaches the model and validate direct-source access separately. Test attempts to claim a privileged role through request content.

- Authenticated caller and workload identities
- Permission-trimmed retrieval
- Cross-user and direct access tests

Planned lab: Demonstrate denied restricted retrieval, denied direct access, and ineffective role spoofing.

Expected output: Access test suite and boundary report.

## 4 MODULE 4 · 4 PROPOSED HOURS

### Tool authority and action containment

A valid tool call can still be inappropriate for a user or task. Validate business scope and parameters before execution. Bind human approval to a specific action and prevent a replay from repeating the side effect.

- Tool schemas and policy enforcement
- Approval binding and argument validation
- Least privilege and duplicate protection

Planned lab: Test a harmless state-changing tool with denied actions, changed arguments after approval, and repeated requests.

Expected output: Hardened action boundary and negative tests.

## 5 MODULE 5 · 3 PROPOSED HOURS

### Runtime secrets and telemetry

Reduce what an unexpected execution can reach. Separate secret management from prompts, restrict outbound destinations where the environment supports it, and minimise logged content. Local teaching controls must not be described as enterprise isolation.

- Execution and network constraints
- Secret exposure prevention
- Security events and audit limitations

Planned lab: Inspect configuration and logs for unnecessary sensitive data and propose containment controls for the deployment design.

Expected output: Containment checklist and telemetry review.

## 6 MODULE 6 · 3 PROPOSED HOURS Remediation and adversarial regression

A finding needs a reproducible case, impact explanation, owner, and verification method. Retest the original failure after a change and include nearby variants. State residual risks that the local test cannot rule out.

- Severity and remediation planning
- Retest and regression cases
- Residual risk and release gates

Planned lab: Implement one control improvement and compare before/after behaviour using a fixed local test set.

Expected output: Remediation and retest report.

## 7 MODULE 7 · 4 PROPOSED HOURS Security capstone

Present a focused security review for the supplied agent application. Demonstrate one boundary before and after remediation and explain which tests support the conclusion. Avoid broad claims that a small test suite proves safety.

- Evidence consolidation
- Boundary demonstration
- Operational escalation and release recommendation

Planned lab: Deliver a scoped assessment with findings, implemented fixes, regression results, and an explicit release recommendation.

Expected output: Security evidence pack and remediation plan.

### CAPSTONE

## Capstone

A scoped security review and hardening exercise for a synthetic knowledge-and-tool application, including role controls and reviewed actions.

### ASSESSMENT

## Assessment

Threat model 20%; tests and evidence 30%; remediation 30%; residual-risk and handover explanation 20%. Critical unresolved control failures block the capstone.

### DELIVERABLES

## Participant deliverables

- Threat model and scoped test plan
- Reproducible benign test cases
- Control changes and regression evidence
- Residual-risk and release memo

### SCOPE

## Scope boundaries

Offensive operations against third-party systems, malware, advanced kernel/container hardening, and claims of complete red-team or senior security-role qualification. \*\*Delivery options:\*\* 3 learning days of 8 hours, or 6 sessions of 4 hours. Learning hours exclude breaks and include module exercises and capstone time. Setup is completed before the course.

### DELIVERY

## Delivery options

3 learning days of 8 hours, or 6 sessions of 4 hours. Learning hours exclude breaks and include module exercises and capstone time. Setup is completed before the course. Proposed hours exclude breaks and are subject to pilot validation.

**PRIVATE DELIVERY****Plan a private cohort for your team**

Scope delivery format, role mix, and rollout timeline with our solutions team.

[Plan Private Delivery](#)[Book advisory call](#)