LLM Security

Duration: 5 Days

Day 1: LLM Threat Landscape & Prompt Injection

Module 1: LLM Threat Landscape

- OWASP LLM Top 10
- Attack surfaces
- Trust boundaries
- Risk classification

Lab Exercises:

1. Map OWASP Top 10 risks to a sample LLM app architecture
2. Identify attack surfaces in a pre-built RAG chatbot demo

Module 2: Direct Prompt Injection

- Injection mechanics
- Jailbreak patterns
- Role override attacks
- Detection basics

Lab Exercises:

3. Execute 5 direct injection attacks on a sandboxed LLM (Ollama)
4. Build a basic input filter using LangChain to block jailbreak patterns

Module 3: Indirect Prompt Injection

- Indirect vs direct
- Web/doc injection
- Tool call hijacking
- Real-world cases

Lab Exercises:

5. Inject malicious instructions into a retrieved document in a RAG pipeline
6. Analyse indirect injection in a LangChain tool-use workflow

Day 2: Data Exposure, Poisoning & Output Handling

Module 4: Sensitive Data Exposure

- PII leakage vectors
- Training data recall
- Membership inference
- Data minimisation

Lab Exercises:

7. Probe an Ollama model for memorised PII using structured queries
8. Implement output scanning with a regex + LLM filter to redact sensitive data

Module 5: Training Data Poisoning

- Poisoning taxonomy
- Backdoor triggers
- Supply chain risk
- Dataset auditing

Lab Exercises:

9. Simulate a label-flipping poisoning attack on a small classifier (scikit-learn)
10. Audit an open dataset for anomalous samples using statistical outlier detection

Module 6: Insecure Output Handling & DoS

- Output injection
- SSRF via LLM
- Token flooding
- Rate limiting controls

Lab Exercises:

11. Demonstrate XSS via unsanitised LLM output rendered in a simple HTML page
12. Simulate a token-exhaustion DoS attack and implement rate limiting with FastAPI

Day 3: Agentic AI Security & Multi-Agent Trust

Module 7: Agentic AI Attack Surface

- Agent architecture risks
- Tool call abuse
- Privilege escalation
- Action boundaries

Lab Exercises:

13. Build a minimal LangChain agent and demonstrate unauthorised tool invocation
14. Implement a tool allowlist and permission check layer in the agent workflow

Module 8: Excessive Agency

- Least-privilege design
- Scope creep
- Irreversible actions
- Human-in-the-loop

Lab Exercises:

15. Identify excessive-agency violations in a provided multi-step agent scenario
16. Redesign the agent with constrained permissions and human approval checkpoints

Module 9: Multi-Agent Trust Boundaries

- Orchestrator/sub-agent trust
- Agent impersonation
- Message signing basics
- Audit logging

Lab Exercises:

17. Simulate an agent impersonation attack in a two-agent LangGraph pipeline
18. Add cryptographic message signing and a trust-verification step between agents

Day 4: Defensive Controls & LLM Red Teaming

Module 10: Guardrails & Input/Output Validation

- Input sanitisation
- Output classifiers
- Constitutional AI basics
- Guardrails AI framework

Lab Exercises:

19. Configure Guardrails AI validators for a sample LLM API endpoint

20. Benchmark guardrail effectiveness against 10 adversarial prompt variants

Module 11: LLM Red Teaming Methodology

- Red team workflow
- Garak scanner
- Attack taxonomy
- Report writing

Lab Exercises:

21. Run Garak against a local Ollama model and analyse the vulnerability report
22. Execute a structured red team exercise using the NIST AI RMF adversarial testing playbook

Module 12: Monitoring & Incident Response

- LLM-specific logging
- Anomaly detection
- Incident playbook
- Observability stack

Lab Exercises:

23. Build a prompt/response logging pipeline with SQLite and flag anomalous patterns
24. Run a tabletop incident response exercise for a simulated prompt injection breach

Day 5: Governance, Compliance & Secure Architecture

Module 13: AI Governance & Risk Frameworks

- NIST AI RMF
- EU AI Act basics
- GDPR & LLMs
- Policy templates

Lab Exercises:

25. Map a sample LLM application to NIST AI RMF core functions
26. Produce a compliance gap report against EU AI Act requirements for a high-risk use case

Module 14: Secure LLM Architecture Design

- Separation of concerns
- Secure RAG design
- Least-privilege integration
- Audit trails

Lab Exercises:

27. Design a secure RAG architecture diagram with security controls annotated at each layer
28. Review a provided insecure architecture, identify 5+ flaws, and produce a remediation plan

Module 15: Applied Capstone Project

- Full threat model
- Architecture review
- Controls mapping
- Peer presentation

Lab Exercises:

29. Build a complete threat model for a realistic LLM use case using STRIDE methodology
30. Present secure architecture design covering top 3 attack mitigations and governance alignment