ADVANCED AI SECURITY: ATTACKS, DEFENSES, AND APPLICATIONS

Expert-Led Cybersecurity Training · Advanced · 14 Hands-On Modules

This course is the direct follow-up and next step to our original course, Practical AI Security: Attacks, Defenses, and Applications. Complete that foundational course first, then take this advanced course to go deeper.

COURSE OVERVIEW

Advanced AI Security is the deep, build-heavy follow-up to Practical AI Security. Where the foundational course teaches how LLMs, RAG pipelines, AI agents and MCP work and how to attack them at a basic level, this course drops you into the frontier of AI security — both as an attacker and a defender — and keeps you at the keyboard.

On the offensive side you inspect model internals and detect malicious models, automate web and mobile security testing with AI agents, build LLM-powered code scanners, master advanced prompt injection and jailbreaking, exploit MCP servers, and plant and detect model backdoors. On the defensive side you red team AI systems at scale with Garak and PyRIT, evaluate and deploy commercial AI gateways, and work through advanced fine-tuning attacks and defenses using local MLX workflows on Apple Silicon.

Every module is anchored on hands-on labs against real, deliberately-vulnerable AI systems. You leave with working tools and code, not slides. The course culminates in a full AI security assessment and the CAAISP (Certified Advanced AI Security Professional) examination.

8KSEC

WHY SHOULD YOU TAKE THIS COURSE?

This is a completely hands-on, advanced course for practitioners who already understand the basics of LLM and AI security and want to operate at the frontier. Instead of theory, you build real offensive and defensive tooling guided by expert instructors. Both on-site and virtual attendees get GPU-backed cloud lab environments to perform exercises with LLMs, agents, vector databases and AI frameworks without complex setup. A dedicated Slack channel ensures pre-course readiness and post-course collaboration for continuous learning.

WHAT WILL THE STUDENT GET?

- Certificate of completion
- Complete course materials (slides, lab guides)
- Source code for all vulnerable AI applications used in class
- Source code for exploit PoCs and red-team harnesses
- All Python scripts, agents and tools built during the course
- GPU-backed cloud lab instances for the duration of the course
- Access to a pre-configured advanced lab environment
- Slack access for collaboration and AI security discussions
- MCP server templates and a custom security-tools repository

WHO SHOULD ATTEND?

Security engineers, red teamers, penetration testers, AI/ML engineers, and AppSec professionals who have completed Practical AI Security (or have equivalent LLM security experience) and want to operate at an advanced, hands-on level attacking and defending agentic AI, LLM pipelines, and production ML systems.

PREREQUISITES

- Completion of 8kSec Practical AI Security (or equivalent LLM security experience)
- Comfortable reading and writing Python
- Familiarity with the command line, Git and Docker
- Working understanding of core web, mobile or cloud security

8KSEC

KEY LEARNING OBJECTIVES

- Inspect model internals and formats (pickle, safetensors, GGUF, ONNX), craft and detect malicious models, and verify integrity with scanning and cosign signing.
- Master AI coding tools (Claude Code, Gemini CLI, OpenAI Codex) and MCP integrations, and understand the full security attack surface of agentic developer tooling.
- Automate web application security testing with AI agents — building custom skills for recon, OWASP Top 10, XSS, SQLi, SSRF, auth bypass and API testing, orchestrated with CrewAI and LangGraph.
- Build AI-powered mobile assessment agents: reasoning over decompiled Android/iOS code, AI-generated Frida hooks, SSL-pinning bypass, and autonomous UI traversal with Maestro MCP.
- Build LLM-powered code scanners — hybrid SAST with Semgrep + LLM triage, prompt-injection-resistant scanners, evaluation harnesses, and scaling to real repositories.
- Use AI for vulnerability research: source auditing, coverage-guided fuzzing, variant analysis, binary RE with Ghidra + AI, exploit development, and patch diffing.
- Execute advanced prompt injection and jailbreaking — PAIR, TAP, GCG, AutoDAN, Crescendo, many-shot, indirect injection via poisoned RAG, and tool-use data exfiltration.
- Run multimodal and cross-modal attacks against vision, audio and document-processing models, including adversarial images, steganographic payloads and OCR pipeline exploitation.
- Exploit MCP servers — tool poisoning, chained-tool RCE, OAuth scope escalation, trojaned packages and rug-pulls (incl. CVE-2025-6514) — and build secure, authenticated servers.
- Implant and detect model backdoors and sleeper agents (BadNets, LoRA backdoors, RAG poisoning) using activation analysis and Neural Cleanse.
- Red team AI systems at scale with Garak, PyRIT and custom LLM-as-judge harnesses, integrated into CI/CD with scoring and automated reporting.
- Evaluate and deploy commercial AI gateways (Cloudflare, Portkey, LiteLLM) with BYOK, guardrails, DLP and defense-in-depth stacks.
- Perform advanced fine-tuning attacks and defenses with local MLX workflows on Apple Silicon — safety removal, backdoor injection, and audit pipelines.

8KSEC

DETAILED SYLLABUS — 14 MODULES

Module 1 Course Overview & Advanced Lab Environment

- Deploying the vulnerable AI applications stack and building a custom vulnerable AI app
- Setting up the advanced lab locally and in Google Colab; configuring multi-provider LLM backends
- Running local LLMs — Ollama, LM Studio and model selection
- Generating API keys for Anthropic, Google Gemini and OpenAI
- Threat modeling an AI application with STRIDE-AI
- Benchmarking AI security posture with automated scanning
- Ethical use and responsible disclosure guidelines

Module 2 Understanding Model Architecture & Formats

- Anatomy of a model — weights, config, tokenizer and adapters (worked safetensors dissection)
- Model formats: PyTorch, TensorFlow, safetensors, GGUF and ONNX (inspect a GGUF model)
- Pickle and deserialization: how a model file runs code — craft and detect a malicious pickle
- Multi-format model scanning with ModelScan, picklescan and fickling
- Adapters and fine-tuning artifacts: the small files that carry big risk (inspect a LoRA adapter)
- Integrity and signing: proving a model is what it claims — sign and verify with cosign
- Where models come from: sources and the AI supply chain

Module 3 AI Tools, Agents & Plugin Ecosystem

- Agents, MCPs, workflows and agent teams — what's the difference
- Claude Code, Gemini CLI and OpenAI Codex — terminal AI for security work
- Claude Code deep dive: CLAUDE.md, memory, hooks and the permission system; building security skills
- OpenAI Codex CLI — setup, sandbox model and security implications
- Finding IDOR vulnerabilities with the Burp Suite MCP
- MCP servers — protocol, connecting, and security
- Installing RTK, measuring token savings, and analysing the hook and trust model

Module 4 AI-Powered Web Application Security Testing

- Building custom Claude Code skills for web reconnaissance and OWASP Top 10 testing
- XSS, SQL injection and SSRF detection skills
- Authentication and authorization bypass testing skills
- API security testing skills: REST, GraphQL and gRPC
- Advanced web testing: fuzzing, frontend analysis and LLM-powered payloads
- Building web pentest pipelines and bug-bounty automation
- Full web-scan orchestration and web security agents with CrewAI and LangGraph

Module 5 AI-Powered Mobile Security Testing & Autonomous Assessment Agents

- AI-powered mobile security architecture and the agent loop
- Android APK static analysis with Claude skills on a real vulnerable target (InsecureBankv2)
- Dynamic analysis with Frida — AI-generated hooks with validation
- SSL pinning bypass, advanced binary reversing, and intent/deeplink vulnerability discovery
- LLM-assisted mobile binary analysis, reverse engineering, and iOS security testing with AI
- Autonomous UI traversal with Maestro MCP
- Orchestrating mobile assessments with pipelines and a LangGraph state graph

8KSEC

Module 6 LLM-Powered Code Scanning — Open-Source & Commercial Pipelines

- Pluggable, model-agnostic code scanner with hot-swappable backends
- Hybrid SAST: deterministic rules (Semgrep) plus LLM intelligence and triage
- Custom vulnerability rule engines with LLM triage
- Prompt injection in scanned code and scanner hardening (injection-resistant scanner)
- Evaluation harness — measuring scanner quality on the OWASP benchmark
- Scaling LLM scanning to real repositories and CI/CD integration
- RAG-augmented vulnerability triage with the CWE database

Module 7 AI-Powered Vulnerability Research

- AI-assisted source-code auditing beyond simple pattern matching
- AI-driven fuzzing: smart corpus generation and coverage-guided mutation (LLM-guided REST API fuzzing)
- Variant analysis with AI — finding families of similar vulnerabilities (Semgrep + LLM)
- AI for binary analysis and RE acceleration (Ghidra + AI; agentic RE with MCP + Ghidra/IDA)
- LLM-assisted exploit development for known CVEs
- Decompilation annotation, vulnerability-pattern matching and patch diffing
- AI-powered smart-contract vulnerability scanning

Module 8 Advanced Prompt Injection & Jailbreaking

- Taxonomy of prompt injection: direct, indirect, multi-turn and compositional
- Automating jailbreaks with PAIR, TAP, GCG, AutoDAN and next-gen techniques
- Crescendo and multi-turn persuasion attacks; many-shot jailbreaking
- Indirect prompt injection via poisoned RAG documents
- Exploiting tool-use / function-calling LLMs to exfiltrate data
- Context-window poisoning and token smuggling
- Building a custom prompt-injection fuzzer

Module 9 Advanced Multimodal & Cross-Modal Attacks

- Adversarial attacks on vision-language models (GPT-4V, LLaVA, Gemini)
- Crafting adversarial images and QR codes that hijack VLM responses
- Typography and steganographic attacks — hidden instructions in PDFs, DOCX and spreadsheets
- Audio adversarial attacks on speech-to-text and voice assistants (inaudible perturbations)
- Cross-modal chaining — combining image, text and audio payloads (attack on an AI meeting summarizer)
- Exploiting OCR-based document pipelines
- Video-frame injection attacks on multimodal agents

Module 10 Advanced MCP Server Exploitation

- MCP protocol deep dive: transport, capabilities and auth flows
- Tool poisoning — hidden malicious instructions in MCP tool descriptions (data exfiltration)
- Achieving RCE via chained MCP tool calls; SSRF and RCE chains through integrations
- Exploiting OAuth misconfigurations and scope escalation in remote MCP servers
- Creating and detecting a trojaned MCP server package; marketplace supply-chain attacks
- MCP rug-pull: dynamic tool-behaviour switching
- CVE-2025-6514 analysis and building secure MCP servers

8KSEC

Module 11 Advanced Data Poisoning & Model Backdoors

- Clean-label poisoning — making backdoors invisible in training data
- BadNets backdoor on a real classifier (GTSRB); trojan attacks on LLMs via instruction-tuning datasets
- LoRA backdoor fine-tuning; sleeper agents (time- and context-triggered backdoors)
- Detecting and mitigating backdoors with activation analysis (Neural Cleanse)
- RAG poisoning — corrupting knowledge bases and vector stores via the service API
- Trojan attacks on code-generation models

Module 12 Red Teaming AI Systems at Scale

- AI red teaming methodology: scoping, enumeration, exploitation and reporting
- Automated campaigns with Microsoft PyRIT (architecture and attack orchestration)
- Garak: LLM vulnerability scanning at scale; automated guardrail stress testing
- Building custom red-team harnesses with LLM-as-judge evaluation and scoring
- Evasion attacks on ML classifiers; Counterfit and the Adversarial Robustness Toolbox
- End-to-end red-team pipeline with scoring and report generation
- Continuous red teaming with CI/CD integration

Module 13 AI Gateways — Cloudflare, Vercel & LiteLLM (Attack & Defense)

- The AI gateway landscape — why you buy one, not build one
- Cloudflare AI Gateway — setup, BYOK, Llama Guard + DLP, and running an attack corpus
- Portkey AI Gateway — virtual keys, guardrails and budgets
- LiteLLM Proxy (self-hosted, MIT-licensed) — virtual keys and budgets
- Attack-versus-defense evaluation across the gateways
- Defense-in-depth gateway stacks, with free-tier hands-on throughout

Module 14 Advanced Fine-Tuning — Attacks, Defenses & Local MLX Workflows

- Fine-tuning fundamentals for security practitioners
- MLX LoRA fine-tuning for security analysis — local, on Apple Silicon
- Safety removal via fine-tuning; adversarial fine-tuning attacks
- Backdoor injection via fine-tuning
- Fine-tuning defenses and an audit pipeline
- Format-hopping persistence — backdoors that survive the LoRA-to-safetensors hop

8KSEC