# SEC543: AI-Assisted Source Code Analysis and Exploitation for Penetration Testers™

2 Day Course | 12 CPEs | Laptop Required

### You Will Be Able To

- Map and analyze large codebases using AI to identify entry points and trust boundaries
- Generate custom security tools, fuzzers, and exploits tailored to each target
- Discover logic flaws and authorization bypasses that automated scanners miss
- Extract high-fidelity context from source code to get accurate AI analysis
- Isolate vulnerable components for safe, controlled testing and exploitation
- Produce validated findings with reproducible evidence for pen test reports

### Business Takeaways

- Find critical vulnerabilities faster with AI-augmented source code analysis
- Generate purpose-built exploitation tools in minutes instead of days
- No software development experience required for this course; the AI reads code while your team provides security expertise
- Move beyond scanner output to context-aware findings with real business impact
- Immediately applicable methodology that can be adapted to many codebases and languages

### Who Should Attend

- Penetration testers looking to add AI-augmented source code analysis to their methodology
- Red team operators who want to accelerate vulnerability discovery and custom tool development
- Security researchers exploring AI-assisted vulnerability research and bug hunting
- Application security professionals who perform code review or security assessments
- Security consultants and engineers responsible for assessing application security posture

### AI-Powered Penetration Testing: Find What Scanners Miss

Modern applications hide their most critical vulnerabilities in logic flaws, subtle authorization failures, and complex multi-step workflows that automated scanners were never designed to find. Attackers already have access to the same AI platforms you do, and they are using them to analyze codebases, identify subtle vulnerabilities, and rapidly develop custom exploits. SEC543 equips you with the same capabilities for authorized pen tests and security assessments.

Over two intensive days, you will learn the Just-in-Time Toolsmith methodology: a repeatable workflow for using AI coding agents to map unfamiliar codebases, extract security-relevant context, discover vulnerabilities, and generate purpose-built exploitation tools. The methodology applies across a wide range of programming languages and codebase sizes, from small web applications to large enterprise systems.

Programming experience is not required for this course. The AI handles code comprehension and tool generation. You provide the security expertise, direct the analysis, and verify the results: the same operator-and-instrument relationship that makes modern pen testing effective.

Students work hands-on against OverflowStock, a realistic e-commerce application with intentionally embedded vulnerabilities spanning authentication flaws, authorization bypasses, injection points, and business logic errors. Ten labs take you from configuring your AI agent environment through validated exploitation with professional evidence capture.

By the end of the course, you will have a complete, battle-tested methodology for AI-augmented source code.

## Section Descriptions

### SECTION 1: Foundations, Environment Provisioning, and Repository Mapping

Build the foundations for AI-augmented penetration testing. Establish the operator mindset for working with AI models, configure a secure agent environment, learn to map unfamiliar codebases without reading every line of code, and master the context engineering techniques that determine whether AI produces actionable results or hallucinated noise.

**TOPICS:** AI Failure Modes, Hallucination Patterns, and Verification Discipline; Agentic Stack Configuration: Models, Tools, and Secure Workspaces; Strategic Repository Mapping with AI-Assisted Codebase Navigation; Context Engineering and High-Fidelity Artifact Creation; End-to-End Vulnerability Discovery Pipeline from Code to Finding

### SECTION 2: Tool Generation, Execution, and Validation

Turn source code knowledge into validated findings. Generate custom security tools on demand, hunt for logic flaws and authorization bypasses, isolate vulnerable components for safe testing, and produce confirmed exploits with reproducible evidence structured for professional pen test reports.

**TOPICS:** Just-in-Time Toolsmithing: custom parsers, scanners, and analysis tools; Logic flaw identification and vulnerability hypothesis development; Component isolation for safe, focused exploitation testing; Automated exploit generation, fuzzing, and validation; Evidence capture and professional pen test report writing

**"I wanted to stop guessing about how to apply AI tooling to security work and learn a defensible methodology, how to analyze large codebases efficiently, when to trust the output, and where it breaks. Tools change fast—the judgment about how to use them is what lasts."**

—SEC543 Student