

SEC546: Securing Agentic AI™

5 Day Program | 30 CPEs | Laptop Required

You Will Be Able To

- Model agentic AI threats, trust boundaries, and core attack surfaces
- Enforce secure input, output, identity, and permission boundaries
- Defend against prompt injection, context poisoning, and tool tampering
- Harden agent memory, runtime operations, and rogue agent containment
- Secure MCP flows, tool execution, desktop agents, and dependencies
- Protect multi-agent, browser, and computer-use agent ecosystems
- Apply cyber-physical safeguards through hands-on live-fire defense exercises

Business Takeaways:

- Reduce enterprise risk from autonomous agents, tools, and connected systems
- Establish enforceable guardrails for agent actions, data, and permissions
- Improve resilience against prompt injection and agent-driven compromise paths
- Strengthen governance for agent memory, runtime behavior, and oversight
- Secure desktop, browser, and multi-agent workflows at scale
- Prepare teams to contain rogue agents and limit operational blast radius
- Build confidence for safer adoption of agentic AI in production



Defending Agentic AI Across the Full Lifecycle

As organizations adopt AI agents quickly, SEC546 gives security professionals a practical, defense-focused approach to protecting systems that can reason, act, use tools, maintain memory, and coordinate across business processes. Students learn how agentic systems are attacked, where trust breaks down, and how to reduce risk through layered defensive controls.

The course shows how to establish trusted boundaries around agent inputs and outputs, defend against prompt-based manipulation, apply secure development practices, and enforce identity, permission, and least-privilege controls so agents operate only within clearly defined limits.

From there, SEC546 moves into the operational realities of defending agentic environments at scale. Students protect memory and context from tampering, detect and isolate unsafe or rogue behavior, and apply runtime oversight that keeps agents observable, governed, and aligned with policy as conditions change.

The course then hardens what agents depend on at runtime: securing MCP data flows, sandboxing tool execution, defending desktop agents, and validating the provenance of dependencies, skills, and prompts. Students defend the runtime environment where agents communicate with peers, drive browsers and operating systems, and delegate authority, then establish fail-safe controls for agents that act on the physical world.

The course also prepares defenders for emerging threats in increasingly connected and autonomous environments. By the end of the week, students can harden agentic AI across the full lifecycle and apply disciplined defensive practices to one of the fastest-moving areas in cyber defense, proven through a comprehensive live-fire defense capstone.

Author Statement

“I created SEC546 because organizations are moving quickly to adopt AI agents, but most security programs were built for traditional software and are not ready for systems that can reason, make decisions, use tools, access data, maintain memory, coordinate with other agents, and act across browsers, desktops, and even physical devices with limited human oversight. This creates new risks, including prompt injection, unsafe tool use, weak identity controls, excessive permissions, memory and context tampering, MCP and supply chain compromise, cross-agent data leakage, rogue behavior, and poor visibility into runtime decisions.

“SEC546 was designed to help close that gap by giving defenders practical ways to secure agentic systems through strong input and output boundaries, secure design patterns, least-privilege access, memory integrity protections, runtime governance, defensive MCP gateways, tool execution sandboxing, multi-agent trust controls, continuous monitoring, and containment strategies. The course also helps organizations understand that adopting these mitigations does not require waiting for perfect standards or complete platform maturity. They can begin now by embedding these controls into architecture reviews, secure development practices, deployment pipelines, operational monitoring, and governance processes so AI agents are introduced in a controlled, measurable, and defensible way.”

—Viswanath (Vis) Chirravuri

Section Descriptions

SECTION 1: Foundations of Agentic AI Security

Section 1 builds the foundation for defending agentic AI by helping students understand the risk landscape, establish trusted boundaries, resist prompt injection, apply secure design patterns, and enforce strong identity and least-privilege controls. It equips defenders to secure agents before they act.

TOPICS:

- Introduction to Agentic AI, Risks, Threat Model
- Enforcing Input and Output Boundaries
- Defending Against Prompt Injection
- Secure Agent Development Patterns
- Agent Identity, Permissions, and Least Agency

SECTION 3: Secure MCP, Desktop Agents and Runtime Defenses

Section 3 hardens the building blocks agents depend on at runtime, including MCP data flows and desktop agent environments. Participants will detect context poisoning and tool response tampering, sandbox tool execution paths, secure desktop agents such as OpenCode, and validate the provenance of dependencies, skills, and prompts.

TOPICS:

- MCP Data Integrity and Context Security
- Agent Tool Execution Sandboxing
- Securing Desktop Agents
- Agent Supply Chain and AIBOM Defense

SECTION 5: Cyber-Physical Agent Security and Emerging Frontiers

Section 5 establishes strict fail-safe mechanisms for agents that act on the physical world or hardware devices, then present the current frontier of agentic security defenses with confidential computing. The day closes with a comprehensive live-fire defense exercise that requires participants to apply every defensive technique from the week against a multi-stage compromise of a production-grade autonomous system.

TOPICS:

- Physical-World Agent Safety
- Emerging Agentic Security Topics
- Agent Defense Capstone

SECTION 2: Agent Operations, Hardening, and MCP Defense

Section 2 moves from foundational controls to hardening production agentic AI systems at enterprise scale. Participants will secure persistent agent memory against poisoning, detect and safely terminate rogue agents, enforce runtime governance policies on live agent actions, and deploy defensive gateways that govern how agents reach external tools, data, and services.

TOPICS:

- Securing Agent Memory and Context Stores
- Detecting, Containing, and Isolating Rogue Agents
- Observability, Governance, and Continuous Defense
- MCP Gateway Defense and Policy

SECTION 4: Multi-Agent, Browser and Computer-Use Agent Security

Section 4 defends the runtime ecosystem where agents communicate with peers, drive browsers and operating systems, and delegate authority. Participants build verifiable trust chains between agents, sandbox browser and computer-use actions with Cua, scope delegated authorization across agent-to-agent handoffs, and prevent task data from crossing tenant boundaries.

TOPICS:

- Multi-Agent A2A Protocol Defense
- Securing Browser and Computer-Use Agents
- Delegated Agent Authorization Defense
- Cross-Agent Data Leakage Defense

Who Should Attend

SEC546 is designed for security professionals responsible for designing, reviewing, or defending systems that include AI agents, and for engineers and architects applying traditional security disciplines to a class of systems that did not exist in production a few years ago. The course assumes working comfort with Python, command line, and containers, along with a foundational understanding of LLMs and GenAI concepts.

Prerequisites

- Python familiarity: Ability to read, modify, and run Python scripts, including using libraries, APIs, and basic program flow
- GenAI fundamentals: Working understanding of large language models, prompting, retrieval-augmented generation, and tool-calling concepts
- Application security basics: General knowledge of vulnerabilities such as injection, authentication, authorization, and trust boundary enforcement
- Command-line comfort: Ability to navigate directories, install packages, edit files, and run lab exercises from a terminal
- Cloud and API awareness: Basic familiarity with API keys, environment variables, JSON, and connecting applications to external services