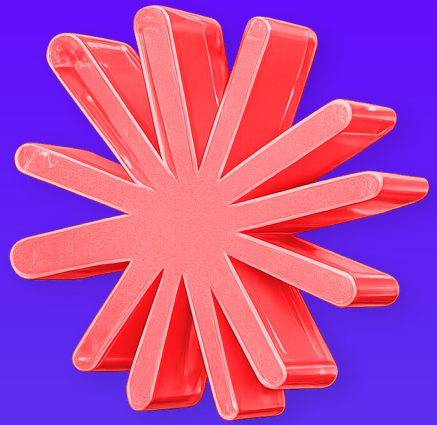
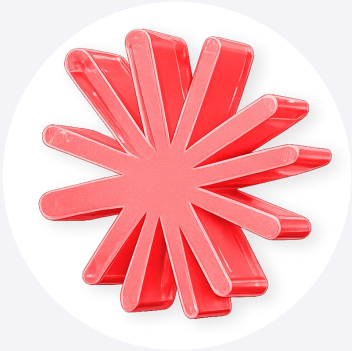


Secure AI Coding with Claude Code Masterclass

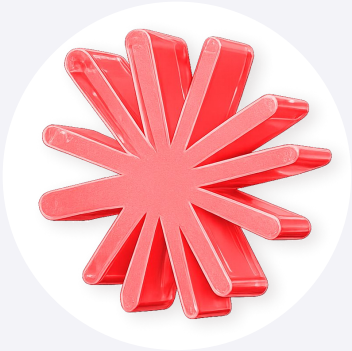
(8 hours)

Last updated: March 2026



AI coding tools now write code, execute commands, and interact directly with local environments. That changes the risk profile of development. Instead of reviewing code after it's written, teams now need to control how code is generated, executed, and integrated into their systems in real time.

This training focuses on securing that workflow end-to-end. It covers how to define permissions, isolate execution, enforce guardrails, and control tool integrations so AI-assisted development runs within clear boundaries without exposing systems, codebases, or sensitive data.



Full-day (8 hours)

Hands-On Labs | Securing AI-Driven Development Workflows

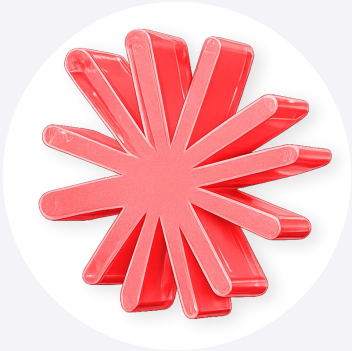
Section 1: The Vibe Coding Workflow and Risk Model

What you'll learn

- How AI coding agents interact with local systems, files, and terminals
- Where risks emerge when agents gain execution capabilities
- Why traditional AppSec controls don't fully apply to agent-driven workflows

Hands-On

- Explore how Claude Code executes commands and modifies codebases
- Identify risk points across the local development workflow
- Map trust boundaries between agent, system, and external tools



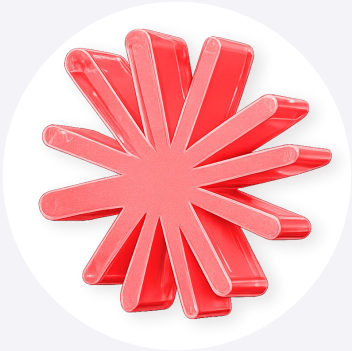
Section 2: Securing Agent Configuration, Permissions, and Secrets

What you'll learn

- How to control agent permissions and execution scope
- How to prevent excessive autonomy and unsafe actions
- How secrets exposure happens in AI-assisted workflows

Hands-On

- Configure secure boundaries for agent execution
- Implement ignore rules and context restrictions for sensitive data
- Validate that credentials and secrets are not exposed to the agent



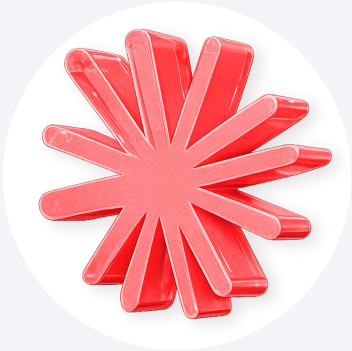
Section 3: Sandboxing and Execution Isolation

What you'll learn

- Why local execution needs isolation beyond built-in controls
- How containerization and OS-level controls limit agent impact
- How to reduce blast radius when agents execute unsafe actions

Hands-On

- Build sandboxed environments using Docker and DevContainers
- Apply restrictions on file access, network access, and privileges
- Test containment by executing unsafe or malicious commands safely



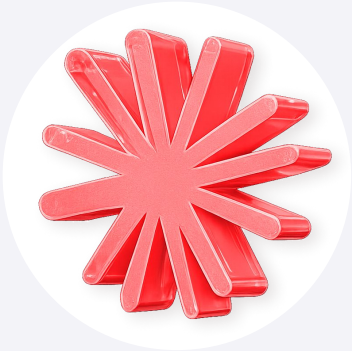
Section 4: Guardrails and Policy Enforcement

What you'll learn

- How to intercept and validate agent actions before execution
- How to enforce security policies within AI-assisted workflows
- How to integrate security checks into the development loop

Hands-On

- Implement pre-execution and pre-commit hooks
- Block unsafe commands and enforce policy checks
- Integrate SAST and SCA scans into agent-driven workflows



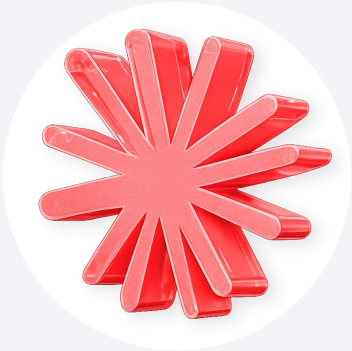
Section 5: Securing Tool Integrations and MCP Ecosystem

What you'll learn

- How agents extend capabilities through external tools and MCP
- Where supply chain and tool misuse risks emerge
- How to validate and restrict tool access

Hands-On

- Analyze and validate external tools connected to the agent
- Detect and prevent tool misuse and shadowing risks
- Apply least-privilege controls to agent-tool interactions



Section 6: Building a Secure AI Development Workflow

What you'll learn

- How to combine controls into a complete secure workflow
- How to enforce consistency across teams and environments
- How to operationalize secure AI-assisted development

Hands-On

- Build a fully controlled AI-assisted development setup
- Apply permissions, sandboxing, guardrails, and tool restrictions together
- Validate the workflow against real misuse scenarios