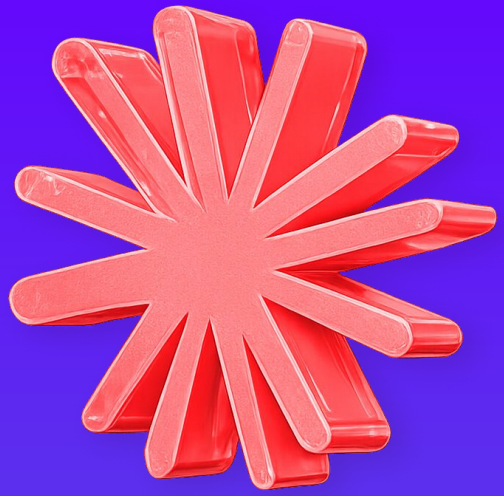
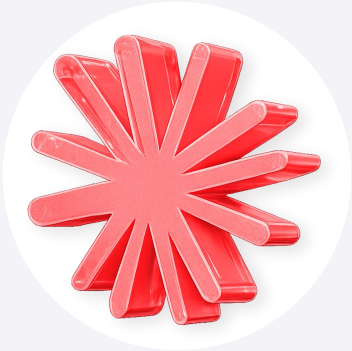


Secure AI Coding with Claude Code Masterclass

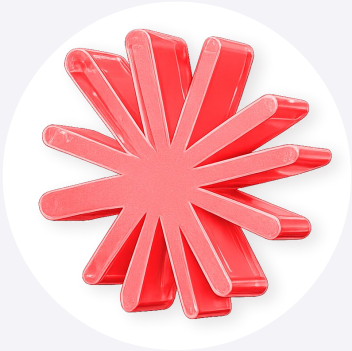
(4 hours)

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.



4-Hour Course Outline

Goal

You get your developers using AI safely without slowing them down or waiting for a full security rollout.

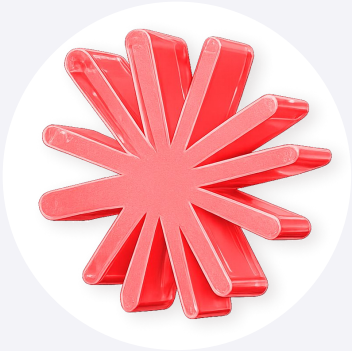
Section 1: Where AI Coding Breaks Your Security Model (30 mins)

You're already using AI to write code. The problem is you've lost control over how that code is generated and executed.

What your team does:

- Identify how tools like Claude interact with local environments, files, and commands
- Map real risks: uncontrolled execution, prompt injection, secret exposure
- Understand how AI changes your threat model inside development workflows

Outcome:
You see where your current workflow introduces risk before it turns into incidents.



Section 2: Controlling Agent Behavior (60 mins)

AI agents don’t just suggest code. They execute actions.

What your team does:

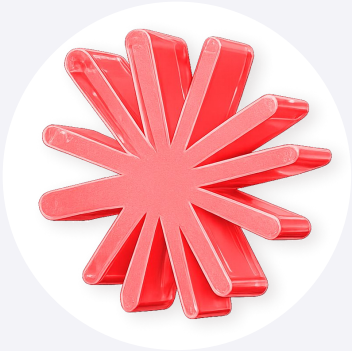
- Restrict permissions and execution scope
- Define what the agent can and cannot access
- Prevent excessive autonomy in AI-driven workflows

Hands-on lab:

- Lock down an AI coding agent and test how it behaves under constrained permissions

Outcome:

- You stop AI tools from acting beyond what your environment should allow.



Section 3: Securing Execution and Local Environments (60 mins)

The biggest risk isn't the code. It's what the agent does while generating it.

What your team does:

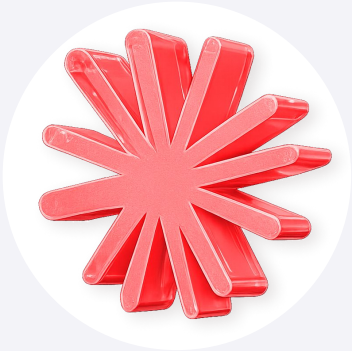
- Isolate execution using sandboxing techniques
- Prevent unsafe system access and command execution
- Control how AI interacts with your local machine and dev environment

Hands-on lab:

- Run AI-assisted workflows inside controlled environments and observe the difference

Outcome:

- You protect developer machines and systems from unsafe AI actions.



Section 4: Guardrails That Actually Work (60 mins)

Security only works if it runs inside the developer workflow.

What your team does:

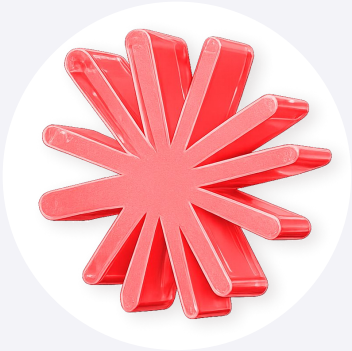
- Add hooks and validation before execution and commits
- Enforce security checks automatically
- Catch unsafe patterns early — without manual reviews

Hands-on lab:

- Implement guardrails that intercept risky actions in real-time

Outcome:

- You reduce reliance on manual reviews and catch issues earlier.



Section 5: Securing Tool Integrations (30 mins)

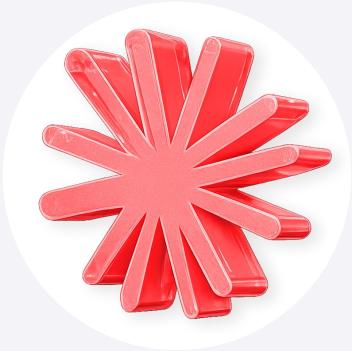
AI agents rely on external tools. That's where supply chain risk shows up.

What your team does:

- Validate and control tool integrations (MCP and beyond)
- Prevent misuse of external services
- Limit what the agent can trigger outside your environment

Outcome:

- You control how AI interacts with external systems.



Wrap-Up: Putting It Into Your Workflow (15–20 mins)

What your team walks away with:

- A clear model for secure AI-assisted development
- Immediate changes they can apply to current workflows
- A baseline for scaling secure AI coding across teams