AI Governance & Cybersecurity Labs — Getting Started Guide

This guide will help you set up your environment to run the hands-on labs for AI Governance & Cybersecurity. You can run the labs in three ways: 1. Directly with Python 2. Using Docker containers (recommended for safety & consistency) 3. Using GitHub Actions (to simulate CI/CD security scans)

1. Running Labs Directly with Python

- Install Python 3.10+ (3.11 preferred). Verify with: python3 --version
- Install pip (Python package manager).
- Install required packages: pip install flask requests beautifulsoup4
- Run labs directly: python lab1_prompt_injection/exploit.py

2. Running Labs with Docker (Recommended)

- Install Docker (v20+) and Docker Compose (v2). Verify with: docker --version docker compose version
- Navigate to a lab folder (e.g., Lab 2): cd lab2_ai_generated_code_vulns
- Build and run vulnerable + defended apps: docker compose up --build
- Open another terminal and run exploits (e.g., python exploit_calc.py).

3. GitHub Actions (CI/CD)

- Push the lab repository to your own GitHub account.
- GitHub Actions will automatically run: - Semgrep (static analysis for dangerous patterns) - Bandit (Python security linter)
- Check the 'Actions' tab in your repo to view scan results.
- Locally, you can also run: pip install semgrep bandit semgrep scan --config lab2_ai_generated_code_vulns/semgrep_rules.yml bandit -r .

4. Optional Developer Tools

- Postman or curl for testing HTTP endpoints.
- VS Code or PyCharm for editing and debugging.
- Pre-commit hooks (black, ruff, bandit) for local lint/security checks before commits.

■ You are ready to start! Each lab folder contains a README.md with specific instructions for exploits and defenses. For safety, remember all secrets are fake and labs are for educational use only. Always

run in a controlled environment.