AI Governance & Cybersecurity Labs — Instructor Demo Script

This script provides step-by-step instructions for demonstrating each lab live in class. For each lab: (1) run the exploit, (2) show evidence, (3) switch to defense, (4) explain governance implications.

Lab 1 — Prompt Injection & Secret Exfiltration

- Run exploit: cd lab1_prompt_injection; python exploit.py
- Expected: Agent output shows fake secret string.
- Run defense: python agent_defended.py
- Expected: Refusal message or safe response.
- Governance tie-in: Confidentiality, access control, context separation.

Lab 2 — Insecure AI-Generated Code (eval & SQLi)

- Start: cd lab2_ai_generated_code_vulns; docker compose up --build
- Exploit eval: python exploit_calc.py → should show 'PWNED'.
- Exploit SQL injection: python exploit_sqli.py → dumps all users.
- Defended endpoints: 5002 blocks malicious payloads, parameterized queries safe.
- Governance tie-in: Secure SDLC, static analysis (Semgrep/Bandit), SOC-2 Change Mgmt.

Lab 3 — MCP-like RCE

- Start: cd lab3_mcp_rce; docker compose up --build
- Exploit vulnerable: python exploit.py → command executes (metacharacters).
- Try against defended: python exploit.py --defended → restricted to allowlist.
- Governance tie-in: Configuration mgmt, system integrity, least privilege.

Lab 4 — Agentic Browser Prompt Injection & Exfiltration

- Start: cd lab4_agentic_browser; docker compose up --build
- Observe logs: Vulnerable agent posts fake secret to receiver (8088).
- Defended agent: Refuses instructions or only posts harmless telemetry.
- Governance tie-in: Post-market monitoring, risk mgmt, accountability, EU AI Act safety.

10-Minute Mini Demo (if time is short)

- Bring up Lab 2 with docker compose.
- Run python exploit_calc.py (vuln → PWNED).
- Show defended calc blocks malicious input.
- Run python exploit_sqli.py (vuln → full user dump).
- Show defended endpoint blocks injection.
- Tie back to governance: CI/CD scans catch these before prod.

Reminder: All secrets are fake. Labs are for educational use only. Never point exploits at systems you do not own or have permission to test.