Lab 2 - Vulnerability Scanning

> Secure System Development - Spring 2025

In this lab, you will

- Experiment with popular SAST tools and different programming languages.
- Practice exploiting basic web app vulnerabilities.

Create a `.md` step-by-step report of the actions you took with screenshots of key results.

Task 1 - SAST Tools

Create a Python virtual environment, where you will install the tools and experiment with them.

1.1. bandit (Python)

> Guide: https://bandit.readthedocs.io/en/latest/start.html

1. Install and run Bandit scan against a local clone of Vulpy
2. Explain one `High`, one `Medium`, and one `Low` severity finding
3. Mention the relevant CWE and propose a mitigation for each.

1.2. flawfinder (C)

> Guide: https://github.com/david-a-wheeler/flawfinder/blob/master/INSTALL.md

1. Install and run FlawFinder scan against a local clone of DVCP
2. Explain one vulnerability of levels 1, 2, and 3.
3. Mention the relevant CWE and propose a mitigation for each.
4. Explain one false-positive finding.

1.3. njsscan (NodeJS)

> Guide: https://github.com/ajinabraham/njsscan

1. Install and run njsscan against a local clone of DVNA
2. Explain one `ERROR`, one `WARNING`, and one `INFO` severity finding.
3. Mention the relevant CWE and propose a mitigation for each.

Task 2 - Web Security Mini Labs

> Mini Labs: https://hub.docker.com/repository/docker/sh3b0/vuln/general

1. Install BurpSuite (Community Edition)
2. Run vulnerable apps locally (bind to `127.0.0.1` to minimize exposure). Example:

   ```
   docker run -p 127.0.0.1:5000:5000 sh3b0/vuln:xss
   ```

3. Solve the mini labs (i.e., exploit vulnerabilities).
   - Cross Site Scripting

    - Path Traversal
    - SQL Injection
    - File Upload
    - Command Injection
4. Report steps taken with unique screenshots. Screenshots should be unique for your submission (e.g., contain your student ID or telegram alias)
5. Include a brief explanation on why the found exploits are dangerous and how we can protect from them (general best practices).