Dreamhack csrf-2

먼저 소스 코드를 열어보았다.

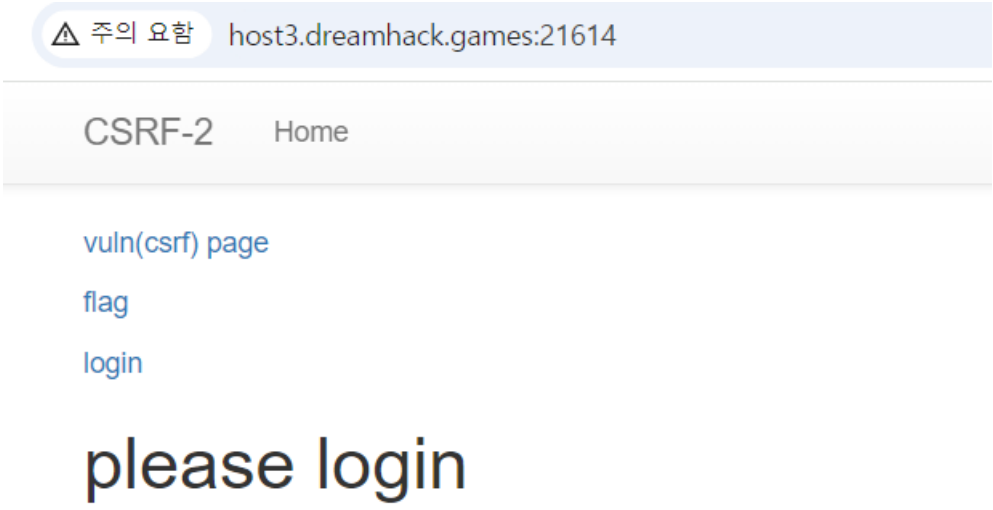
```python
@app.route("/change_password")
def change_password():
    pw = request.args.get("pw", "")
    session_id = request.cookies.get('sessionid', None)
    try:
        username = session_storage[session_id]
    except KeyError:
        return render_template('index.html', text='please login')

    users[username] = pw
    return 'Done'

app.run(host="0.0.0.0", port=8000)
```

전체적으로 csrf-1의 소스 코드와 유사하나 이 부분이 추가되어 있는 것이 눈에 띈다. 이 부분을 분석해보았다.

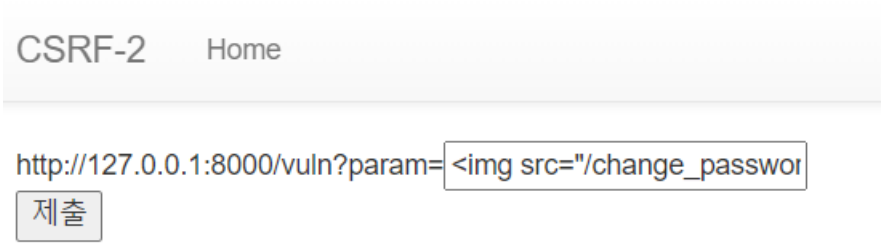
- request.args.get("pw", ""): URL query 파라미터로 전달된 pw 값을 가져온다. 예를 들어, /change_password?pw=newpassword라면 newpassword가 pw의 값이 된다.
- request.cookies.get('sessionid', None): 브라우저의 쿠키에서 sessionid라는 쿠키 값을 가져온다. 해당 쿠키가 없다면 None이 된다.
- session_storage[session_id]: 세션 스토리지에서 세션 ID에 해당하는 사용자 이름을 가져온다.
- except KeyError: 세션 ID가 session_storage에 없을 경우(즉, 유효한 세션이 아닐 경우) KeyError 예외가 발생한다. 이 경우 로그인하지 않은 사용자로 간주하여 index.html 템플릿을 렌더링하고, "please login"이라는 텍스트를 함께 출력한다.
- 비밀번호 변경: 세션 ID에 해당하는 사용자 이름을 찾은 후, 그 사용자의 비밀번호를 새로운 값으로 업데이트한다.
- 완료 메시지: 비밀번호가 성공적으로 변경되면 'Done'이라는 응답을 반환한다.

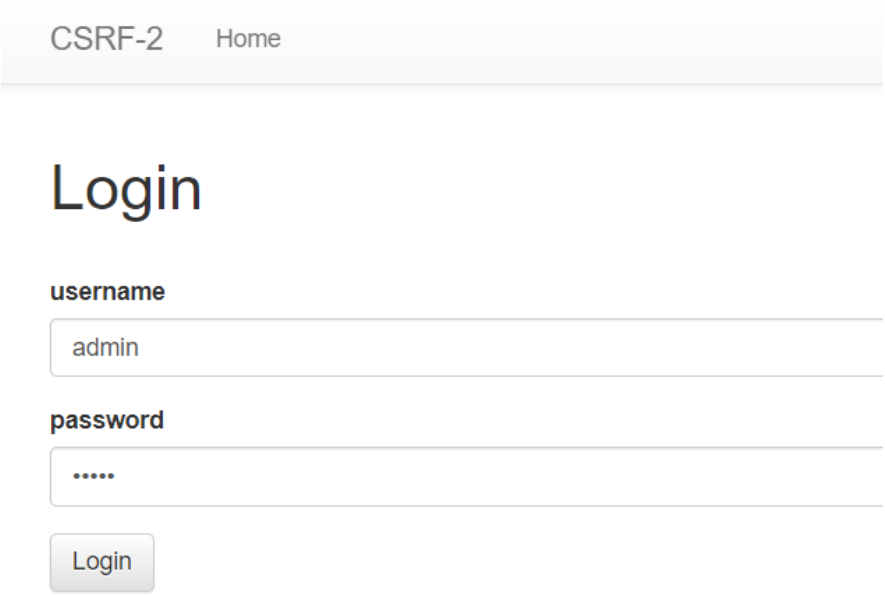
문제의 목표는 admin으로 로그인하는 것이다. Pw에 원하는 비밀번호를 넣어 바꿔주면 로그인이 될 것 같다.

서버를 열어보았다.

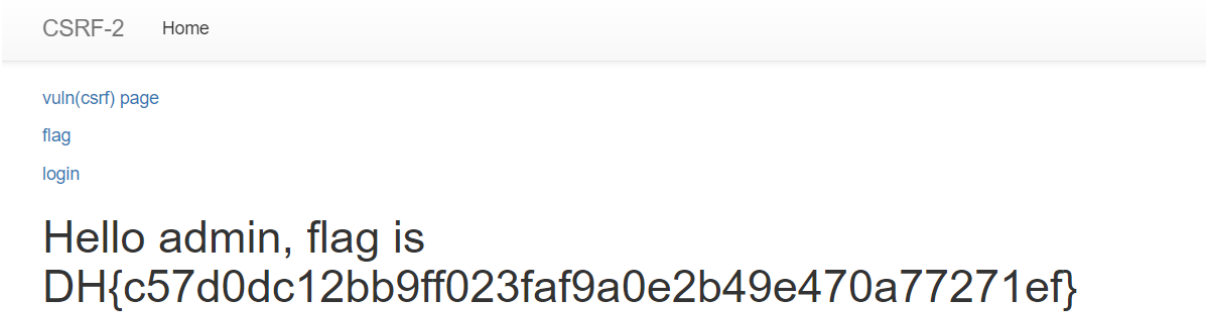


Csrf-1 에서 사용한 img src 태그를 여기에서도 사용할 수 있을 것 같다. **<img src="/change_password?pw=admin">** 을 flag 페이지에 넣어주었다.





이후 아이디 admin/패스워드 admin으로 로그인을 해준다.



플래그는 **DH{c57d0dc12bb9ff023faf9a0e2b49e470a77271ef}** 임을 알 수 있다.