

# CSAW - LLM

Suriya Prakash Jambunathan - sj3828  
Krishnan Navadia - kn2501

## web/cookie\_injection

### Observation:

The challenge provides us with the source code for a flask application. It is a cookie injection problem.

### Approach:

1. Observe the source code.
2. Launch the application using the Dockerfile and the shell-script provided.
3. Look into vulnerabilities in the code, especially login.py
4. Identify the attack string that will exploit the vulnerability in the app.
5. Extract the printed flag.

### Solution:

1. Prompted ChatGPT with the source code of login.py to identify vulnerabilities. (Step 3).
  - a. GPT identified the following vulnerabilities

```
SQL injection
Session Hijacking
Cookie Injection
Cross Site Scripting
```

2. Prompted ChatGPT to give some possible attack strings. (Step 4).
  - a. GPT mentioned the following payloads

- i. **Cookie Value Manipulation:**

Attackers may attempt to manipulate the values of existing cookies to gain unauthorized access. For example:

```
`username=admin`
`user_role=admin`
`authenticated=true`
```

- ii. **Encoded Payloads:**

Attackers might encode malicious payloads to obfuscate their intentions.

For example:

```
%3Cscript%3Ealert('XSS')%3C/script%3E (URL-encoded script tag)
```

iii. **Cross-Site Scripting (XSS) Payloads:**

If an attacker can inject scripts into cookies, they can execute arbitrary code in the context of other users. For example:

```
<script>malicious_code_here</script>  
"><script>alert('XSS')</script>
```

iv. **SQL Injection Attempts:**

If cookies are used in database queries without proper validation, attackers may attempt SQL injection via cookies. For example:

```
' ; DROP TABLE users; --
```

v. **Command Injection Attempts:**

Similar to SQL injection, attackers might attempt command injection via cookies. For example:

```
; ls -la  
| cat /etc/passwd
```

**Flag:** *Flag not given with the files. Although GPT helped in giving the above payloads, we were unable to extract the flag.*

**Chat:** [CSAW LLM - Cookie Injection \(openai.com\)](#)