Security Notes – OWASP Juice Shop

Scope :-

The objective of this activity was to perform basic security testing on

OWASP Juice Shop to understand common web application vulnerabilities

and practice documenting security observations professionally.

Target :-

OWASP Juice Shop running locally on:

http://localhost:3000

Observations :-

- SQL Injection was observed in the product search API where user input

  was not properly sanitized.

- Basic testing showed how backend queries can behave unexpectedly

  when special characters are passed in input fields.

- Chrome DevTools Network tab was helpful to inspect API requests

  and responses.

Questions :-

- What additional validation mechanisms can prevent SQL Injection?

- How can automated tools like SQLmap help in deeper testing?

- What is the correct way to prioritize vulnerabilities in real projects?

Learning :-

- Understood how input validation failures lead to security issues.

- Learned how to observe HTTP requests and responses using Chrome DevTools.

- Gained experience in writing clean and structured security notes using Markdown.

- Learned how GitHub can be used as a documentation portfolio.

Vulnerability 2: SQL Injection in Product Search API

Title :- SQL Injection in Product Search API –

Description

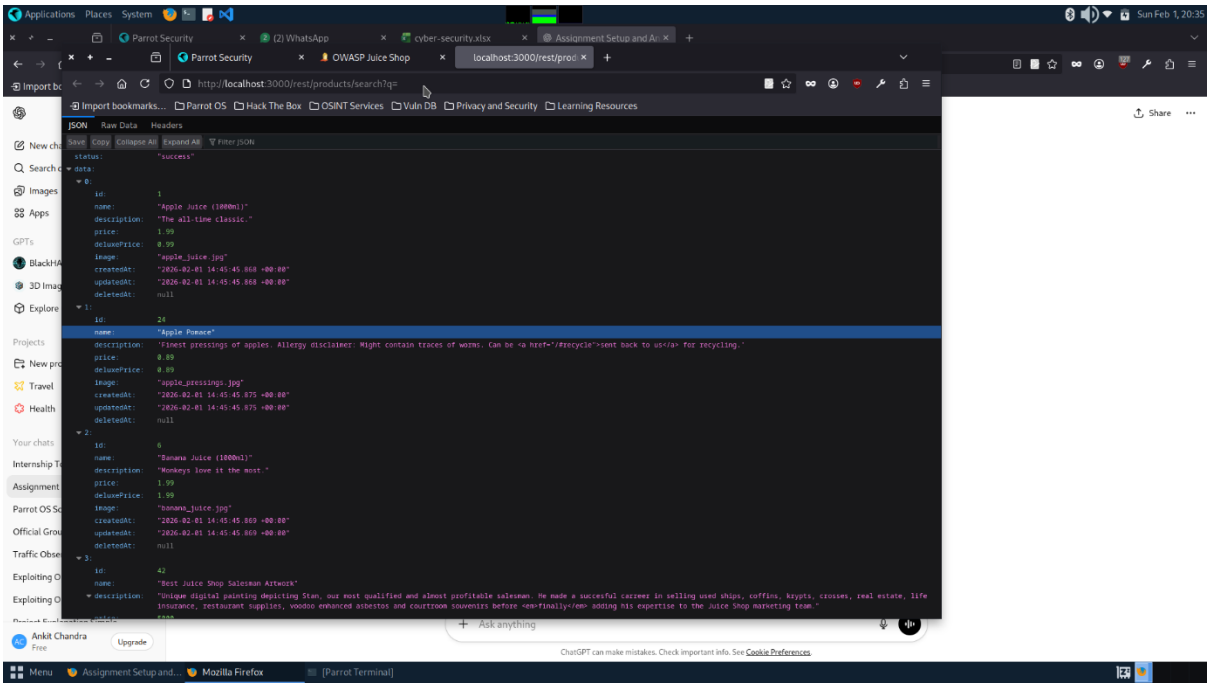
The product search API endpoint does not properly sanitize user input passed through the `q` query parameter. By injecting a crafted SQL payload, the backend database query is manipulated, causing the API to return all product records instead of filtered results.

This confirms the presence of an SQL Injection vulnerability in the search API endpoint.

Impact

- An attacker can retrieve unauthorized or excessive data from the database
- Sensitive information may be exposed
- The vulnerability can be leveraged for further attacks such as data extraction or database enumeration

Evidence

- SQL Injection payload used:
- 

' OR 1=1--

 Vulnerable API endpoint:

http://localhost:3000/rest/products/search?q=' OR 1=1--

 The API returned multiple product records without applying search filters

 Screenshot attached showing JSON response containing all products

 Confirms backend SQL query manipulation

URL

http://localhost:3000/rest/products/search

Severity :-

- **Login Bypass Attempt:** Medium

- **Product Search SQL Injection:** High

Tools Used :-

- Web Browser (Firefox / Chrome)

- Browser Developer Tools

- OWASP Juice Shop (Docker)

Conclusion

The OWASP Juice Shop application is vulnerable to SQL Injection due to improper input validation and lack of secure query handling. These vulnerabilities can lead to authentication logic bypass, unauthorized data access, and potential database compromise. Proper input sanitization and the use of parameterized queries are recommended to mitigate these issues.