

DAY 8 TASK 8 – SQL Injection Practical Exploitation

Objective:

To understand and exploit SQL injection vulnerabilities in a web app (I used OWASP Juice Shop)

Tools Used :

- OWASP Juice Shop (locally by docker)
- Browser Dev Tools
- Manual sql injection payloads.

Sql injection was identified in the search , By intercepting the /rest/products found initial payloads confirmed unsanitized input in the search parameter.

' OR1=1-- = bypassed search logic and **broken the query**

It returns a sqlite error

' = returned a result by **search?q=**

This confirms unsanitized user input being directly embedded into SQL queries, leading to error-based SQL Injection and improper error handling.

It confirms that the user input is directly concatenated into sql queries

Backend queries are leaking.

Prevention Techniques:

- Parameterized Queries(prepared statements)
- Input validation before sending to database(backend) and sanitization
- Use ORM(Object Relational Mapping)
- Least Privilege Database access
- Error Handling
- WAF(web application firewall)