SCENARIO

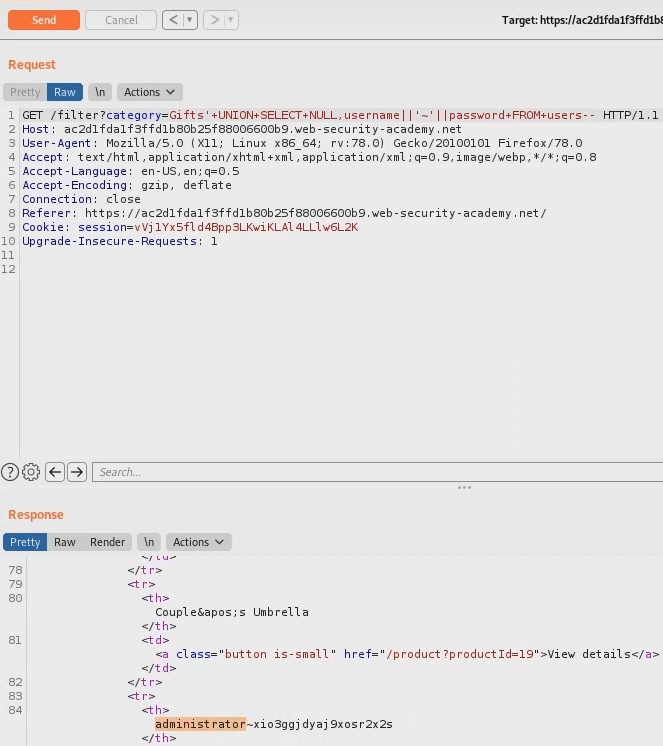
The application's search functionality might be flawed. Initial tests show that searching for products might lead to a SQL injection vulnerability. This vulnerability could be exploited to leak confidential product data that hasn't been released to the public yet.

**PROCEDURE**

1. Run Burp Suite and intercept the request while trying a product search.
2. Inject a basic SQL payload, such as ' OR 'x'='x, in the search input to test for vulnerabilities.
3. If the injection is successful and all products are displayed, try more advanced payloads to fetch more sensitive data or access other database tables.

**PAYLOAD**

' OR 'x'='x

**PROOF OF CONCEPT**

**REMEDIATION**

1. Implement input validation and sanitization to check for and neutralize potentially malicious strings.
2. Utilize ORM frameworks which abstract and sanitize SQL queries, minimizing direct SQL interactions.
3. Regularly update and patch the database system to protect against known vulnerabilities.
4. Limit the visibility of data by using role-based access controls.
5. Periodically review the database logs for unusual queries that could be indicative of an attack.