SCENARIO

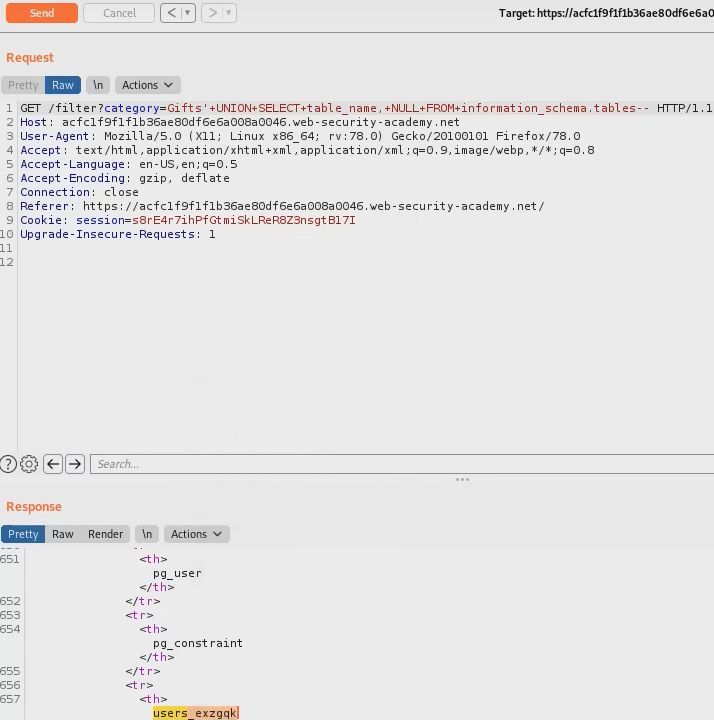
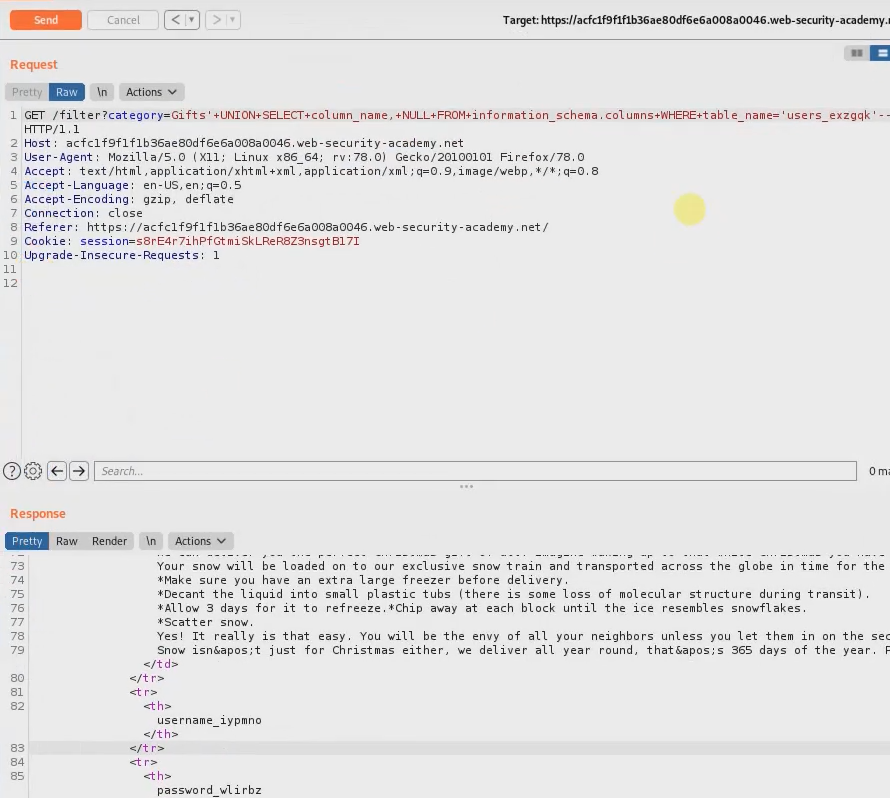
The application's product category filter has a SQL injection vulnerability. Given the application's response, a UNION attack may expose data from other tables, especially user credentials.

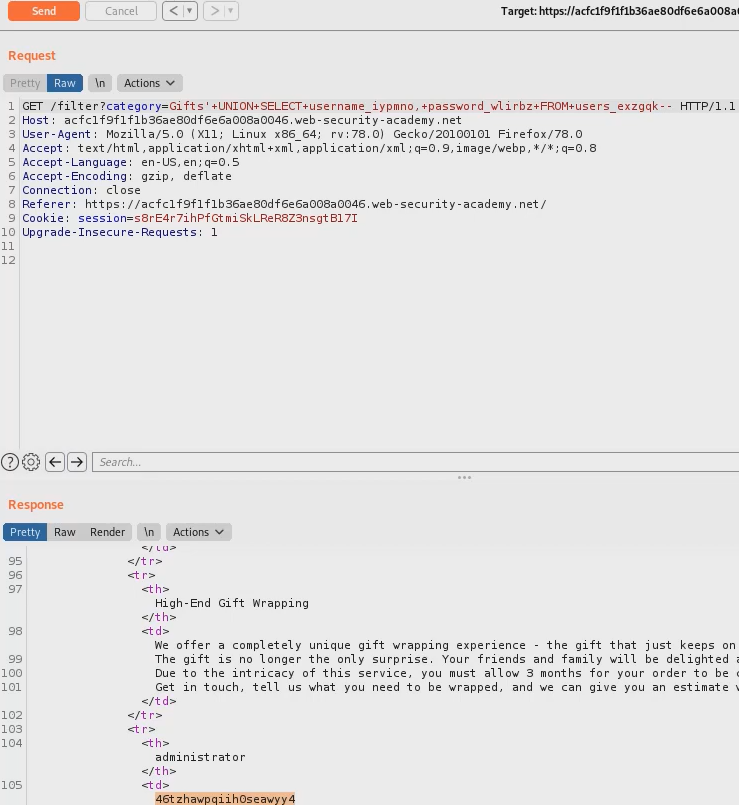
**PROCEDURE**

1. Start Burp Suite and capture the request modifying the product category filter.
2. Confirm the number of columns returned by the initial query by adjusting the category parameter to: '+UNION+SELECT+'abc','def'--.
3. To fetch user credentials from the users table, modify the payload to: '+UNION+SELECT+username,+password+FROM+users-- and inspect the response for any usernames and passwords.

**PAYLOAD**

'+UNION+SELECT+username,+password+FROM+users--

**PROOF OF CONCEPT**

****

**REMEDIATION**

1. Adopt a whitelist approach for input validation to prevent malicious inputs.
2. Use parameterized queries or ORM (Object-Relational Mapping) tools to avoid direct SQL queries.
3. Educate developers on secure coding practices, emphasizing avoiding the use of dynamic SQL queries.
4. Monitor and analyze database logs for suspicious activities.
5. Regularly back up the database and ensure that backups are encrypted and stored securely.