SCENARIO

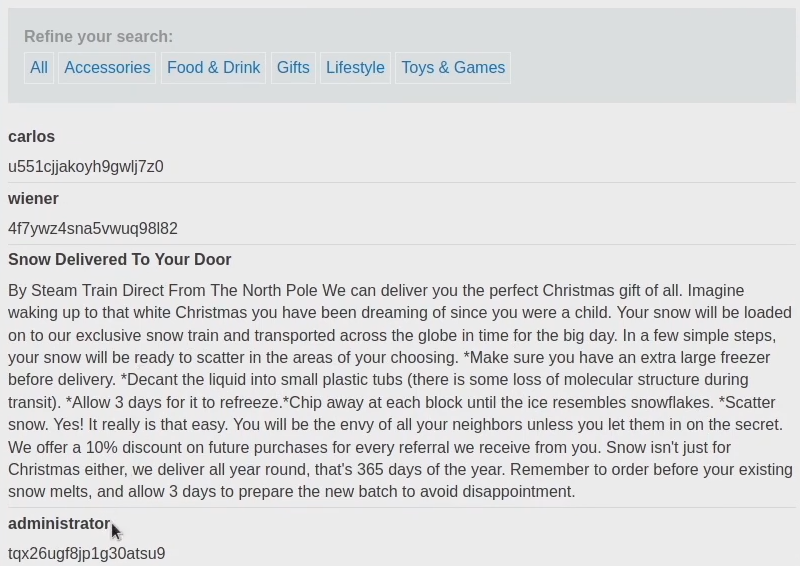
The application's feedback submission function appears to be vulnerable. There's a strong indication that SQL injection might be possible when submitting feedback. The potential exists for attackers to use this vector to gain access to sensitive data.

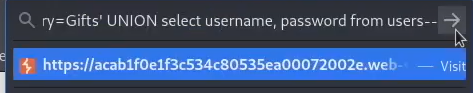
**PROCEDURE**

1. Activate Burp Suite and capture the feedback submission request.
2. Experiment by inserting common SQL injection payloads in the feedback text area.
3. Through the feedback field, try the classic SQL injection test: '+OR+'1'='1.
4. If successful, escalate the attack by probing the database for further information.

**PAYLOAD**

'+OR+'1'='1

**PROOF OF CONCEPT**



**REMEDIATION**

1. Use prepared statements with parameterized queries. This ensures that parameters (or inputs) passed into SQL statements are treated safely, and not executable.
2. Adopt a Content Security Policy (CSP) to reduce the risk of injection attacks.
3. Limit the privileges of database accounts used by web applications to only what's needed.
4. Encrypt sensitive data in databases and ensure encrypted data cannot be decrypted via SQL injection.
5. Avoid disclosing detailed error messages that can give attackers clues about the database structure.