Assignment Submission

Labs Completed

1. Unprotected admin functionality
2. User role controlled by request parameter
3. Username enumeration via different responses
4. Password reset broken logic
5. SQL injection – retrieve hidden data
6. SQL injection – login bypass

Reflections

1. Unprotected Admin Functionality

This lab highlighted how often developers rely on obscurity instead of real access control. The most surprising part was how easily `/robots.txt` exposed the admin panel. No authentication, no session check—nothing. It reinforced the principle that administrative endpoints must always require proper authorization, regardless of how hidden they appear. The challenge was simply remembering to check the common weak points first.

---

2. User Role Controlled by Request Parameter

This lab clearly demonstrated why role verification should never be handled on the client side. Modifying a simple `admin=false` cookie value gave full admin access, showing how dangerous trust in client-supplied data can be. The challenge here was not technical—Burp made it easy—but understanding how careless developers can be with authorization logic. It reinforced the importance of server-side validation for all privilege checks.

---

3. Username Enumeration via Different Responses

This lab taught me how subtle differences in server responses can leak sensitive information. Response length, status codes, and messages become powerful indicators when testing authentication mechanisms. Burp Intruder made it efficient to automate the attack. The most challenging part was recognizing which response difference mattered and avoiding false positives. It strengthened my understanding of how weak error handling directly enables enumeration attacks.

---

4. Password Reset Broken Logic

The lab exposed how insecure password-reset workflows can compromise entire accounts. By modifying the reset request in Burp, I bypassed token validation entirely and reset another user's password. The biggest takeaway was that password-reset mechanisms must validate both the token and the correct user pairing. The challenge was identifying exactly where the server failed its validation, but once found, the exploit was straightforward.

---

5. SQL Injection – Retrieve Hidden Data

This lab showed how simple SQL injection can be when user input is concatenated directly into queries. Injecting `' OR 1=1--` bypassed the category filter and returned all products, including hidden ones. The key learning point was how a single operator like `OR 1=1` can undermine the entire logic of a query. The challenge was ensuring the payload matched the query structure to avoid syntax errors.

---

6. SQL Injection – Login Bypass

This lab demonstrated how SQL injection can bypass authentication entirely. Breaking the query with `'` and using `' OR 1=1--` allowed login without a password. It reinforced the necessity of parameterized queries and input sanitization. The challenge lay in understanding how the backend query was originally structured, but once the pattern was clear, the injection became predictable. It emphasized how dangerous poorly written login logic can be.