**PortSwigger Access Control Vulnerability**

**Intern id:** 195

**Lab :** User ID Controlled by Request Parameter

**Environment :** Linux

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective**

Show that the application allows attackers to access other users’ accounts by manipulating the id parameter in the request, without enforcing server-side access control.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Target**

Application: PortSwigger Access Control Lab – User ID Controlled by Request Parameter

Vector: id parameter in account page requests

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Vulnerability Description**

The application determines which user account to display based solely on the id parameter supplied in the request. There is no check that the authenticated user owns the account corresponding to that ID. This enables attackers to retrieve or modify other users’ data.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Steps to Reproduce**

1. Log in as a normal user (wiener:peter).

2. Browse to your account page:

https://<lab-url>/my-account?id=wiener

→ Displays your personal account details.

3. Change the id parameter in the URL:

https://<lab-url>/my-account?id=carlos

→ The page now loads Carlos’s account instead of Wiener’s.

4. Verify sensitive data access:

If the lab objective is to get carlos’s API key → it is displayed.

This confirms unauthorized access.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Technical Explanation**

The server likely executes:

SELECT \* FROM users WHERE id = '<parameter>';

Since the app trusts the client-supplied id, it fetches records for any user ID given in the request.

This is an Insecure Direct Object Reference (IDOR).

The app should enforce session-based ownership checks.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Impact**

Horizontal privilege escalation (any user → any other user).

Disclosure of personal data, API keys, or account info.

Potential account takeover if combined with password reset or modification endpoints.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mitigation**

Never rely on client-side parameters to enforce authorization.

Always validate that the requested id matches the authenticated session user.

Use server-side session data instead of exposing identifiers in URLs.

Apply access control checks consistently across all endpoints.

**\_\_\_\_\_\_\_\_\_\_\_ THANK YOU \_\_\_\_\_\_\_\_\_\_\_**