**PortSwigger Access Control Vulnerability**

**Intern id:** 195

**Lab :** Unprotected Admin Functionality

**Environment :** Linux

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

**Objective**

Demonstrate that an administrative function is accessible without authentication or proper access control, allowing an attacker to perform privileged actions.

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

**Target**

Application: PortSwigger Access Control Lab – Admin Functionality

Vector: Direct access to /admin panel

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

**Vulnerability Description**

The application exposes an admin panel endpoint but does not enforce authentication or authorization checks. Any user (even unauthenticated) can directly browse to the endpoint and use administrative features.

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

**Steps to Reproduce**

1. Browse the website as a normal user.

No admin options are visible in the UI.

2. Review the site structure:

Check robots.txt:

https://<lab-url>/robots.txt

→ May disclose /admin.

Alternatively, try common paths directly:

https://<lab-url>/admin

3. Access the admin panel:

The page loads without requiring authentication.

Example: It shows an option to delete users.

4. Trigger a privileged action:

Navigate to:

https://<lab-url>/admin/delete?username=carlos

Observe: The user carlos is deleted, confirming privilege escalation.

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

**Technical Explanation**

The backend lacks proper access control, such as:

GET /admin/delete?username=carlos HTTP/1.1

Host: <lab-url>

Cookie: session=attacker

The server processes the request without validating whether the requester is an admin user.

This violates the principle of vertical access control.

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

**Impact**

Any unauthenticated or low-privileged user can:

Access the admin panel.

Perform sensitive operations (e.g., deleting accounts, changing data).

Leads to account takeover and loss of integrity.

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

**Mitigation**

Restrict access to admin endpoints by enforcing role-based access control (RBAC).

Hide sensitive paths (though obscurity alone is not sufficient).

Implement strong session-based authorization checks on every request.

Use separation of duties (admin accounts only).

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