Unprotected admin functionality with unpredictable URL

**1. Summary**

**Bug Title:**  Unprotected admin functionality discloses the URL of the admin panel in some java scrpit code which unuthenticated userhave full access to it

**Severity**: Critical.

**Description**: Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits..

**Date Discovered:** 1/8/2024.

**Status:**  Solved.

**2. Bug Details**

**Vulnerability Type:** Broken Access Control.

**Affected URL/Endpoint:**  /admin-7bzj0z and JS code in source page functionality discloses the URL of the admin panel .

**Description:** A critical vulnerability has been identified in the web application where an attacker can access unprotected admin panel with full control . The attacker can find the URL to this unprotected admin panel via examine the source code of the app because there is some JS code that discloses the URL. This vulnerability highlights a severe case of broken access control, allowing unauthorized/unauthenticated users to gain elevated privileges and potentially compromise the entire system.

**Steps to Reproduce:**

1. Go to lab URL: <https://portswigger.net/web-security/access-control/lab-unprotected-admin-functionality-with-unpredictable-url> and access the lab

2. view page sorce code (CTRL+u) you will find a JS code contains a URL for Admin panel



4. access this admin functionality via this URL: <https://0a3500df0393d31e80b230bd00560031.web-security-academy.net/admin-7bzj0z>

5. delete user carlos to solve the lab

**Proof of Concept (PoC):** lab solved



**Impact:** The application improperly expose unprotected admin functionality in JS code in the source code . By accessing it , an attacker can gain unauthorized access to administrative functionalities, including the ability to delete user accounts..

**3. Recommendations**

* Role-Based Access Control (RBAC): Ensure that access to administrative functions is based on clearly defined roles and permissions. Only authenticated and authorized users should have administrative privileges
* Server-Side Authorization Checks: Always perform authorization checks on the server side to verify user permissions before granting access to sensitive operations.
* protect admin functinality via stricted authentication scheme

**4. Conclusion**

**Summary:** A critical vulnerability has been identified in the web application where an attacker can escalate privileges to an administrative level by access un protected admin functionality. This allows unauthorized users to access admin functionalities and delete user accounts, highlighting a severe case of broken access control.

**5. Appendices**

**Tools Used:** Browser.

**References:**

* <https://owasp.org/Top10/A01_2021-Broken_Access_Control/>
* <https://portswigger.net/web-security/access-control>