### A01 Broken Access Control

Access control policies are designed to prevent users from performing actions outside their permissions (OWASP, 2021a). They ensure that only authorized users or systems get access to specific resources while unauthorized entities access is denied (Vivian, 2024). However, when access control is improperly enforced the system becomes vulnerable to broken access control (BAC) (OWASP, 2021a). This vulnerability allows attackers to bypass security checks and gain unauthorized access to sensitive data, perform restricted actions, or escalate privileges (OWASP, 2021a). One common example is insecure direct object reference (IDOR), where an web application uses IDs in the URLs or API endpoints to access resources (Vivian, 2024). Attackers can exploit this by manipulating the ID in the request, enabling them to view or modify another user data (OWASP, 2021a). This vulnerability is critical as it can be easily exploited in a web application and pose severe consequences for the integrity and confidentiality of data.

|  |  |  |  |
| --- | --- | --- | --- |
| Summary of Vulnerability Tests | Tools and Techniques | Justification for using Tools/Technique | Findings |
| The URLs were manipulated to test for IDOR, assessing whether an unprivileged user could gain unauthorized access to admin account. | * Manual testing | Manual testing enables direct URL manipulation to detect IDOR vulnerabilities that automated tools might miss. | IDOR was not found in the system. |

* **Broken access control**

First, I logged into the admin account using the password "password". A point to note is that the field value for the admin account files is 3.A screenshot of a computer

Description automatically generated

After clicking on the admin profile icon, a drop-down list appeared including Settings, Users, and Logout option. I clicked on the **Users** option to view the list of users in the system.

A screenshot of a computer

Description automatically generated

The Users page showed that there are two accounts: admin and Luke.A screenshot of a computer

Description automatically generated

Next, I logged into Luke’s account using the password "12345678". It’s important to note that Luke’s field value is 21.A screenshot of a computer

Description automatically generated

While logged in as Luke, I attempted to access the admin’s files by changing the field value in the URL to 3. However, the attempt was unsuccessful — Luke is not authorized to access admin’s files through URL manipulation.A blue background with white clouds

Description automatically generated

* **Cryptographic Failure**

1. Testing with Nmap

I ran an Nmap scan to determine whether the system uses encrypted communication or relies on unencrypted HTTP. The results indicate that the HTTP port is open signaling at server reliance on unencrypted protocol for communication.



1. **Testing with Gobuster and John the ripper**

During the Gobuster scan, an HTML page named *register.html* was discovered.A screenshot of a computer

Description automatically generated

After navigating to the page it appears to contain usernames and hashed passwords

A screenshot of a computer

Description automatically generated

The usernames and passwords were saved to a file for use in password-cracking attemptsA screenshot of a computer

Description automatically generated

I used John the ripper to attempt to crack the hashes. After sometime, the tool managed to crack 4 out of the 8 passwords. Meaning the passwords were not salted and secured properly**.**

A screenshot of a computer program

Description automatically generated