Day-5

Broken Access Control

[Task 18] [Day 5] Broken Access Control

Websites have pages that are protected from regular visitors, for example only the site's admin user should be able to access a page to manage other users. If a website visitor is able to access the protected page/pages that they are not authorised to view, the access controls are broken.

A regular visitor being able to access protected pages, can lead to the following:• Being able to view sensitive information
• Accessing unauthorized functionality

OWASP have a listed a few attack scenarios demonstrating access control weaknesses:

Scenario #1: The application uses unverified data in a SQL call that is accessing account information:pstmt.setString(1, request.getParameter("acct"));ResultSet results = pstmt.executeQuery( );

An attacker simply modifies the 'acct' parameter in the browser to send whatever account number they want. If not properly verified, the attacker can access any user's account.http://example.com/app/accountInfo?acct=notmyacct

Scenario #2: An attacker simply force browses to target URLs. Admin rights are required for access to the admin page.http://example.com/app/getappInfohttp://example.com/app/admin_getappInfo

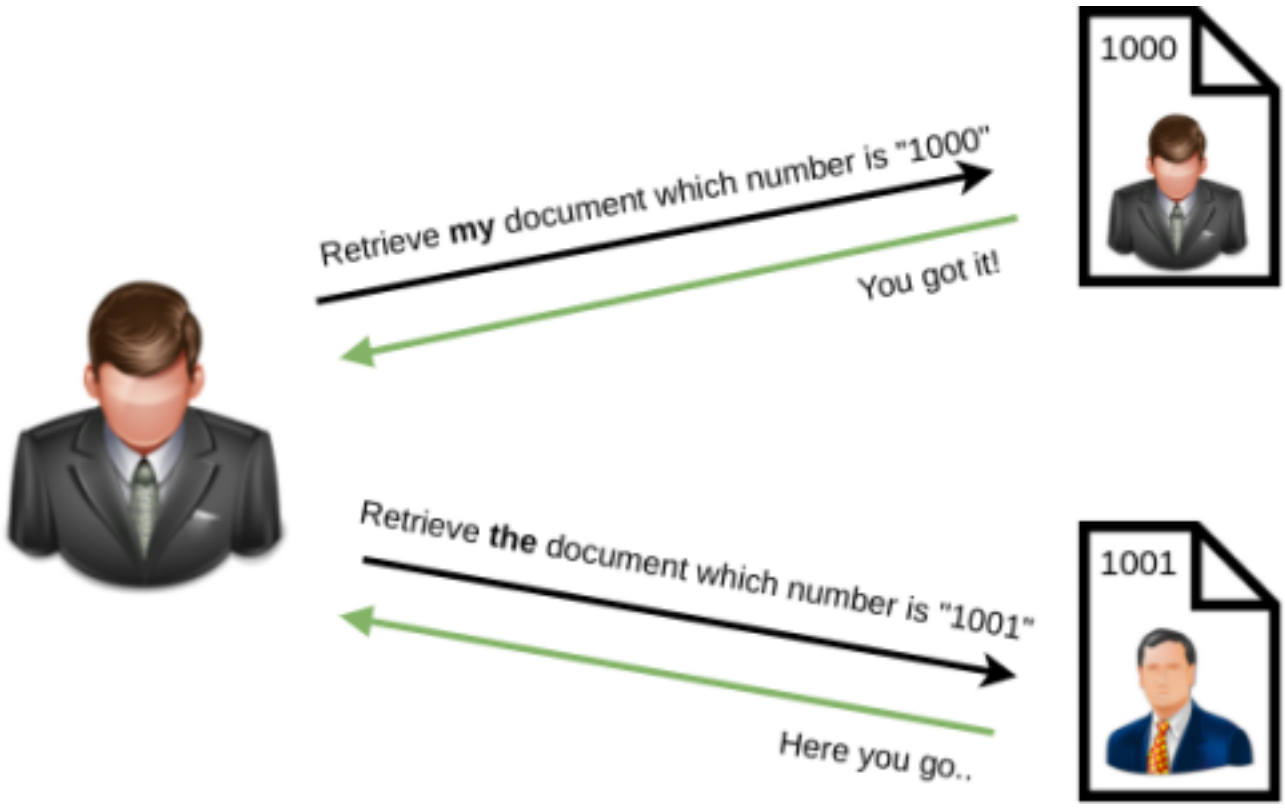
If an unauthenticated user can access either page, it's a flaw. If a non-admin can access the admin page, this is a flaw (reference to scenarios).

To put simply, broken access control allows attackers to bypass authorization which can allow them to view sensitive data or perform tasks as if they were a privileged user.

#1
Read and understand how broken access control works.

No answer needed

[Task 19] [Day 5] Broken Access Control (IDOR Challenge)



IDOR, or Insecure Direct Object Reference, is the act of exploiting a misconfiguration in the way user input is handled, to access resources you wouldn't ordinarily be able to access. IDOR is a type of access control vulnerability.

For example, let's say we're logging into our bank account, and after correctly authenticating ourselves, we get taken to a URL like this https://example.com/bank?account_number=1234. On that page we can see all our important bank details, and a user would do whatever they needed to do and move along their way thinking nothing is wrong.

There is however a potentially huge problem here, a hacker may be able to change the account_number parameter to something else like 1235, and if the site is incorrectly configured, then he would have access to someone else's bank information.

#1
Read and understand how IDOR works.

No answer needed

#2
Deploy the machine and go to http://MACHINE_IP Login with the username being noot and the password test1234.

No answer needed

#3
Look at other users notes. What is the flag?

flag{fivefourthree}