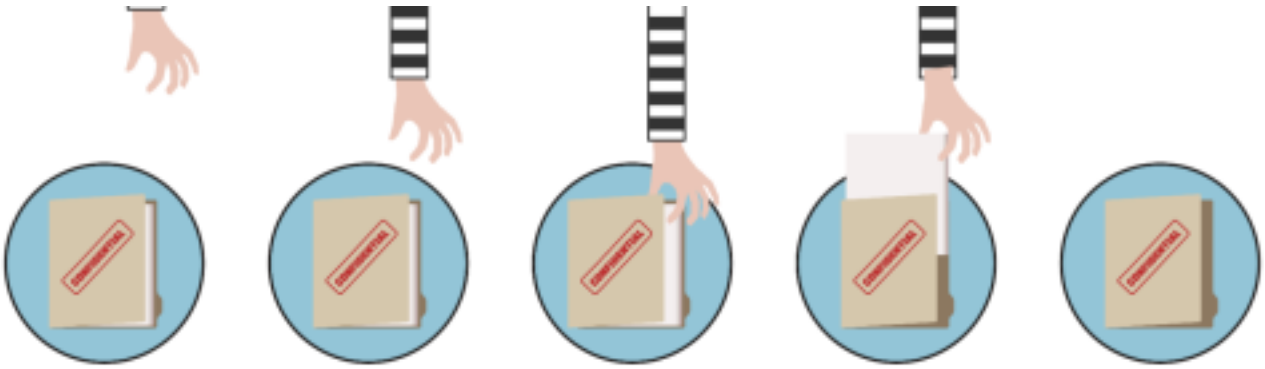
Day-2

Broken Authentication

[Task 7] [Day 2] Broken Authentication



Authentication and session management constitute core components of modern web applications. Authentication allows users to gain access to web applications by verifying their identities. The most common form of authentication is using a username and password mechanism. A user would enter these credentials, the server would verify them. If they are correct, the server would then provide the users' browser with a session cookie. A session cookie is needed because web servers use HTTP(S) to communicate which is stateless. Attaching session cookies means that the server will know who is sending what data. The server can then keep track of users' actions.

If an attacker is able to find flaws in an authentication mechanism, they would then successfully gain access to other users' accounts. This would allow the attacker to access sensitive data (depending on the purpose of the application).

Some common flaws in authentication mechanisms include:

• Brute force attacks: If a web application uses usernames and passwords, an attacker is able to launch brute force attacks that allow them to guess the username and passwords using multiple authentication attempts.
• Use of weak credentials: web applications should set strong password policies. If applications allow users to set passwords such as 'password1' or common passwords, then an attacker is able to easily guess them and access user accounts. They can do this without brute forcing and without multiple attempts.
• Weak Session Cookies: Session cookies are how the server keeps track of users. If session cookies contain predictable values, an attacker can set their own session cookies and access users' accounts.

There can be various mitigation for broken authentication mechanisms depending on the exact flaw:

• To avoid password guessing attacks, ensure the application enforces a strong password policy.
• To avoid brute force attacks, ensure that the application enforces an automatic lockout after a certain number of attempts. This would prevent an attacker from launching more brute force attacks.
• Implement Multi Factor Authentication - If a user has multiple methods of authentication, for example, using username and passwords and receiving a code on their mobile device, then it would be difficult for an attacker to get access to both credentials to get access to their account.

#1
I've understood broken authentication mechanisms.

[Task 8] [Day 2] Broken Authentication Practical

For this example, we'll be looking at a logic flaw within the authentication mechanism.

A lot of times what happens is that developers forgets to sanitize the input(username & password) given by the user in the code of their application, which can make them vulnerable to attacks like SQL injection. However, we are going to focus on a vulnerability that happens because of a developer's mistake but is very easy to exploit i.e re-registration of an existing user.

Let's understand this with the help of an example, say there is an existing user with the name admin and now we want to get access to their account so what we can do is try to re-register that username but with slight modification. We are going to enter " admin"(notice the space in the starting). Now when you enter that in the username field and enter other required information like email id or password and submit that data. It will actually register a new user but that user will have the same right as normal admin. That new user will also be able to see all the content presented under the user admin.

To see this in action go to http://MACHINE_IP:8888 and try to register a user name darren, you'll see that user already exists so then try to register a user " darren" and you'll see that you are now logged in and will be able to see the content present only in Darren's account which in our case is the flag that you need to retrieve.

#1
What is the flag that you found in darren's account?

fe86079416a21a3c99937fea8874b667

#2
Now try to do the same trick and see if you can login as arthur.

No answer needed

#3
What is the flag that you found in arthur's account?

d9ac0f7db4fda460ac3edeb75d75e16e