010 - So what exactly is BAC?

Introduction

In the beginning, the applications that We created were very static but as time progressed these static applications moved to a much more dynamic nature. With this dynamic nature. We also saw a need arise for a much more segregated system of controls. After all, certain users should be able to perform certain actions and some users should only be able to perform the most basic of actions. This is usually done with an access control module but that is not always an easy task. We have to ensure that every single user is checked properly and that their authorization is taken into account. Now, this is the first time that you will see me use the word authorization, but it is certainly not the last.

What is access control?

Now that we've talked about access control we also need to talk about authorization. With the Need for access control, we have a need to also apply that to a specific user we do this through the authorization control mechanisms. When a user logs into an application they authenticate, the difference between authentication is that you are being authenticated AKA you proving who you are. If we look at authorization you are authorized to do something meaning you are already authenticated and then authorization occurs.

So now that we are both authenticated and authorised, now do we have a way to enforce that access control that we are setting out to enforce with our application. All of this comes together to allow the user to perform certain actions and that can be either on a specific object or access control in general like a functionality.

When is it broken and when not?
How to translate this?

You might be wondering now that we know how it's supposed to work when exactly is it broken. To know when it's broken we need to first identify what a user is or is not allowed to do it's important to do this for every single Endpoint or every single functionality within the application. You can easily do this and combined it all to make an Excel sheet, in this Excel sheet you will note down every single piece of functionality or every Endpoint that you have in the application. Now you want to start with writing down all of the specific user rights for user groups. Having done this you need to make sure that you know down whatever user or user group is or is not allowed to do. Mark all of the things that are not allowed in red.

Table 1

	Normal user	Sales	Management	Admin
Can edit invoice				
Can view invoice				
Can print invoice				
Can manage users				
...				

Now every single area in red that you see should be tested fully. Testing in areas like this fully often means retesting your application over and over again until every single user-type has been exhausted. If you happen to find something in the red where you're allowed to do it well you should not be able to do that then you'll have a broken access control issue.

BAC vs IDOR

IDOR or insecure direct object references are also a type of broken access control. This variation of broken access control occurs when we have an object involved. Objects in modern programming applications are often marked with an identifier. When we have that unique

identifier on a specific object and we are allowed to see that object or interact with it otherwise well we should not normally be able to do this we can speak of an IDOR.