W4D2 -- Intro to Authentication

Let’s start by looking at the basics of authentication. With authentication you have 2 factors that need to be implemented. You have the Username, this is who the user is, and you have the password this is the “authorizing” part of authentication. This is what keeps someone out of someone else’s account. We talked about SQL injections before, how they can create problems if you don’t protect against them, today we will talk about that.

Problems with authentication

1. The first thing you should know is that YOU WILL GET HACKED, the question is more of a how badly you get hacked.
2. The first step in securing your users, is to let someone else deal with the password protection, if you try to write the authentication code yourself, you will likely get it wrong. not to get you down or anything, but authentication is a complex issue, and can be very vulnerable to attacks if done incorrectly.
3. If you must do work with local or site based authentication, use known and tested libraries. It’s a good thing that ASP.Net provides authentication for us.
4. The worst thing you can do is store passwords in plain text, make sure your passwords are encrypted and hashed.
5. The biggest issue with authentication is with the passwords. Many people have easy passwords to hack. If your password is Password or abc12abc, you might want to change that password immediately… and then write a 10 page paper apologizing to your parents and the school. Of course I’m joking, but still change the password
6. ComputerPhile has a video on how to store Passwords that you need to watch, and answer the questions.  
   How NOT to Store passwords: <https://www.youtube.com/watch?v=8ZtInClXe1Q>
   1. What is Tom’s main suggestion?
   2. What is the worst way to store passwords?
   3. What is the current way to store passwords, according to the video?
   4. What is wrong with just encrypting passwords?
   5. What is wrong with just hashing passwords?
   6. What is the salt Tom talks about?

What makes a good password?

1. XKCD has a good comic on what makes a good password. <https://www.xkcd.com/936/>
   1. What is the main problem that XKCD pointed out about passwords?
   2. How does he suggest that you deal with this problem?
   3. Would you switch to this scheme if you could?
2. Computerphile creates another video about how to chose a good password, watch it and answer the questions.  
   <https://www.youtube.com/watch?v=3NjQ9b3pgIg>
   1. So does mike mention the XKCD comic?
   2. How does he describe it?
   3. What are mike’s suggestions on how to structure your passwords?
   4. What is his final suggestion on how to password?

So now that we’ve talked about the problems with authentication, let’s talk about the benefits

Earlier we set up a web page that let you make and store students. This page was accessible from outside and completely vulnerable to outside manipulation. Here we are going to talk about methods to prevent an unauthorized user from gaining access to these pages.

ASP.Net has an AuthorizeAttribute. This attribute makes it so that certain views or functions can only be viewed if a user is logged in or authorized to use or view them.   
  
 [Authorize]

Function()

ASP.Net also allows the creation of roles. This allows us to modify what each user can do. So you can set up a “user” that is only authorized to create a profile and modify that profile. We can also set up an admin that can modify and delete entries in the database.

[Authorize(Roles = “Administrator”)]

Function()

With these attributes you can set an entire controller to require authentication, or you can set individual sections to require authentication.

Next time we will go over how to set authentication on a project.