Wagner\_Mark

Lab 7 – php sql

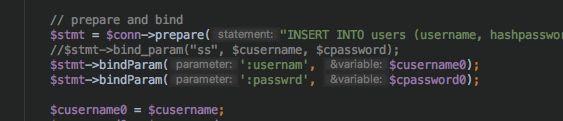
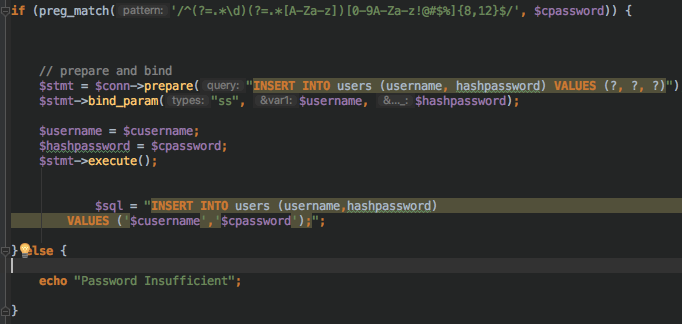
SDEV300-7983

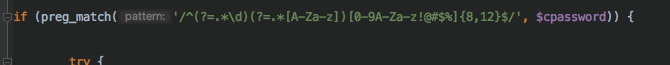
5/3/2019

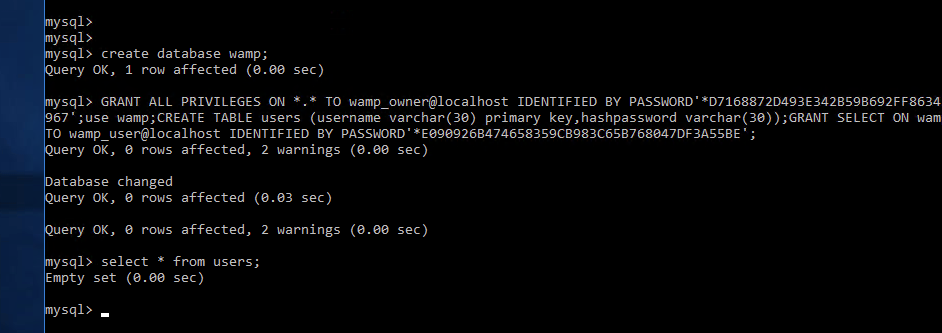
So using good ole’ php, I build this application. Rather than controlling access to the database with hashed passwords, I stored a table INSIDE the database of passwords.

Additionally, rather than controlling select access from the password column in the user table, I just initiated a session variable upon successful login (to a successfully registered account) and used that to control all further modifications to the database.

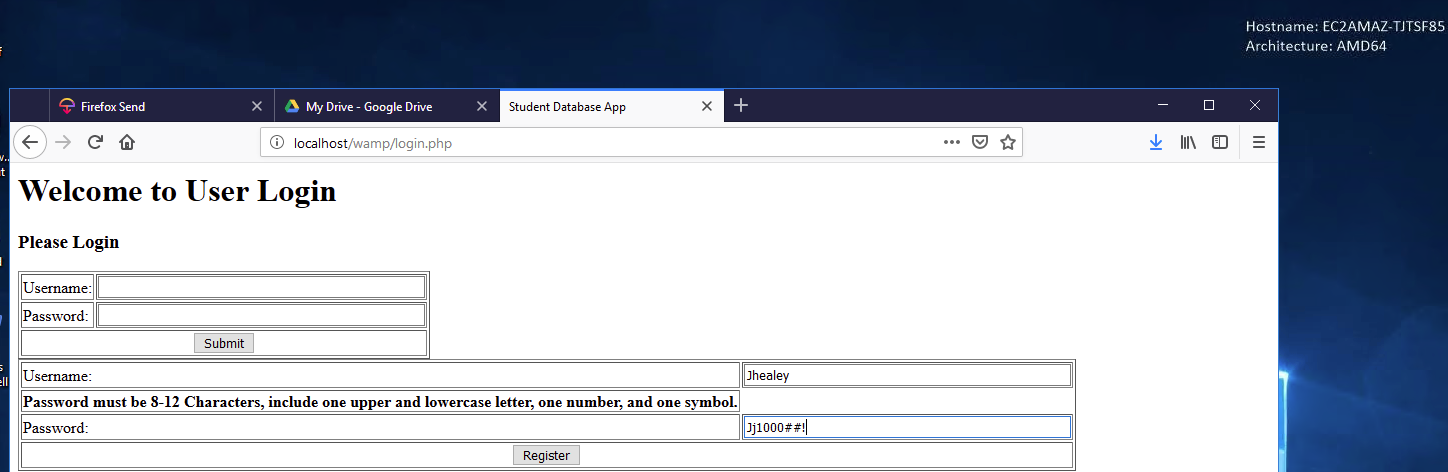
It can be a little glitch and so at the beginning of the login.php, I added a session clearing method so that going back to incognito view for each try isn’t necessary.

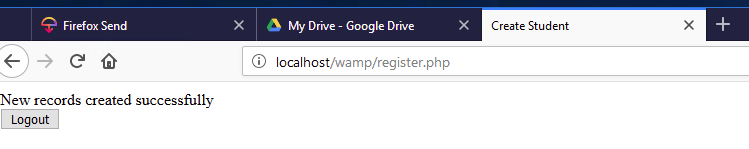
The one secure aspect of this database is that I used the binding queries to prevent sql injection: I also used a pattern function I found online to ensure that passwords registered were STIG (using upper, lower, number, symbols) compliant.



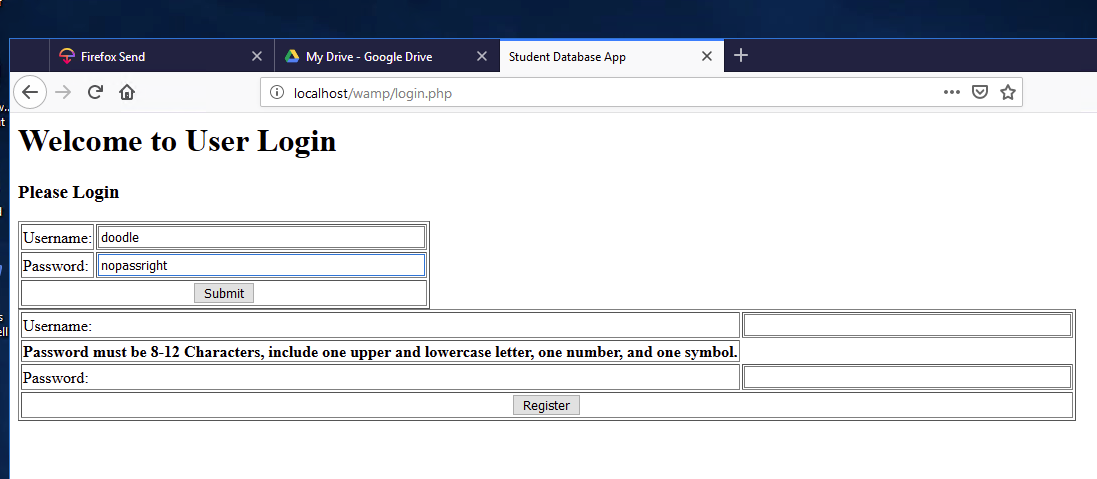
I then made redirects to the home/login pages as appropriate to ensure a smooth flow. Database setup:

Here a first user is added:



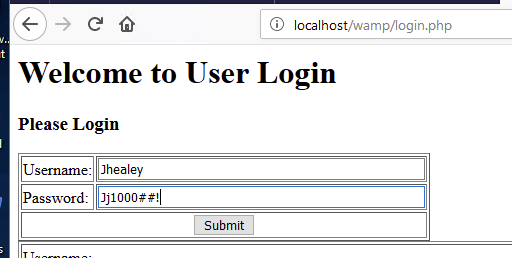


here a user with an non-compliant password is created:

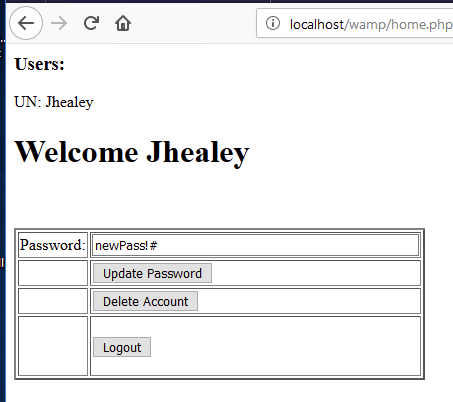


SystemSub:Users:Terrapin:Desktop:Screen Shot 2019-05-03 at 6.26.27 PM.png

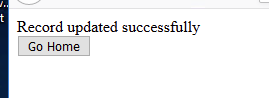
Here I log in with correct credentials:

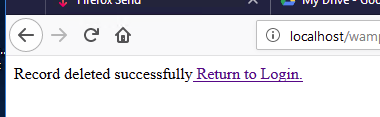


I am greeted and the session variable is set: I can do anything with this account:

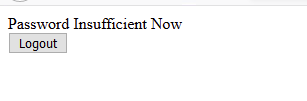


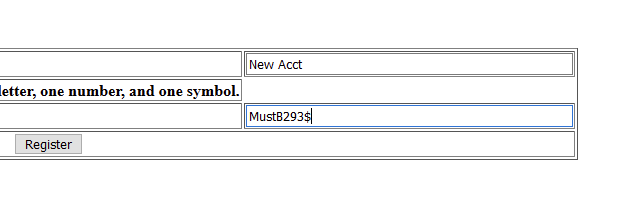
So I change my password:

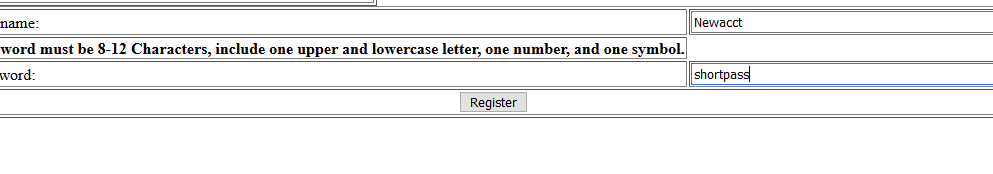


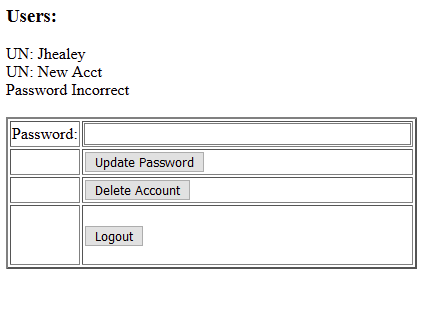


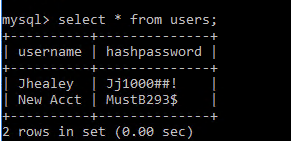
And Delete my account.



Now I log out and create a second account.

The one above was from earlier (it didn’t work because of the password)

Now I am failed to log in but you can see that the other account was created. I’ll be honest, the delete function is now glitching but it WAS working on mariaDB on my mac before I switched to the AMI. I just cannot determine why records are not deleting at this point but I suspect a database mistake. In any case, it did work. 



And here it is, straight from the db itself.

While this is super insecure, it is fast and does protect against some basic things. Many of the possible errors would probably be patched just by building it in react router, angular or as a single page vue app. Ahh,, progress!

Thank you for viewing my project! - Mark