Abel Lawal:

New User Page and Extra Credit Implementation:

My part of the project was to work on the new user page and to try and implement the extra credit 2-step verification procedures that we discussed. When initially designing the web page, my plan was to emulate things that I have encountered in my own experience in my own online endeavors. The main objective that I wanted to achieve was instantaneous feedback. I have always thought that it was and bad design not to mention really annoying when a user is trying to fill out a form, hits submit, waits for the page to load, only for it to tell you that the user name is already taken.

I decided to create an HTML form in order to obtain the information from the user. I made the form straightforward to try to prevent any confusion. Very early in the project we decided to emphasize on security so for the most part the pages look very bare. I think that the first step in any secure system is making sure to obtain clear, concise and accurate information from the user. For every field I set prompts that alert the user what data needs to be provided, this was all implemented to avoid confusion. Whether it’s the fact that they can’t leave the field empty or that the password needs to be of a certain length. This information is all communicated via a pop up alert that is activated when the user tries to leave the field. Deciding how to let the user know of a data entry/typographical error was a tricky as it came down to a choice between ease of coding vs usability of the website. In the end I decided to use the JS method setcustomvalidity() to personalize the error messages and provide helpful feedback to the user. I decided to use this method as even though it was tricky to implement at the start, it uses the default method of the browser viewing the website. This choice adds to the credibility of the site; though recently it has changed it used to be the case that I could tell the phishing sites simply by the way they were designed. Even though I was trying to keep the site design simple I didn’t want to make it seem so basic that it was untrustworthy.

In some ways the strength of the password is the final decider of how secure a system is. This is due to the fact that even if every other part of the system is exceptionally secure, if the password is known to be a single lower case letter, then it wouldn’t take too long for that system to be comprised. Therefore I decided that at the minimum the password for our system must be: At least 6 characters formatted in the following way: At least 1 Upper Case Letter, 1 Special Character, 1 number, 2 lower case letters and be at least 6 characters long. I decided against going a longer password with fewer requirements as I feel that it is harder for most people to come up with an 8 letter word. Therefore a common password which is used across different sites is created, a very risky behavior trait. The validation of the user’s password was done using JavaScript and regular expressions; this is a simple and efficient way to ensure that the password was composed of a certain length and that it had certain characteristics (i.e. uppercase, lowercase and a special character etc.). The password field also gives real time feedback to the user about the strength of the password.

I will write a little more about sql injection to finish up my part of the write up