Matthew Olson

Lab 9 CSSE 333

Q1. SHA-1 encrypts the user's password, so even if an attacker was able to retrieve the information they would only have the hash result and not the password itself. Storing the username with the password increases the security of the password, because it increases the size of the hash that needs to be decrypted. This is only true as long as the attacker does not know that the username is being concatenated, which could result of the key being broken by a partial message attack.

Q2. Mysqli_connect() Connects to a MySql DBMS

Mysqli_select_db() Selects the database to run the commands in

Mysqli_error() returns the error that was caused by the DBMS

Q3. It is valuable to use a stored procedure here so the DBMS can compare the passwords instead of the PHP client. This greatly increases security as the information is processed entirely on the server, and not in the browser. Stored procedures should be used when sensitive data should not be seen by the end user and when it would be substantially faster to use a stored procedure. It makes sense to run queries directly when the queries are simple.

Q4. The DBMS ran the query in the name category and created a new user that did not come from the provided inputs. This is dangerous because the attacker can run any sort of query they desire from this form, including querying data, updating data, or ever deleting the data without correct privileges.

Q5. The post body was interpreted by PHP as an html command and executed on the index page. This could be very dangerous for end users because it can redirect them to malicious webpages or seemingly identical pages that could steal their user data. Both SQL injections and XSS are dangerous, SQL injections seem to be more dangerous to the database and XSS seem to be able to do more damage to the end users. I think that SQL injections are more dangerous because of the possible scale of damage that they can cause.