Varatep Buranintu
Daniel Jameson
Yun Kim
Cale Watson
Professor Vahab
May 3, 2015

COMP 424 Course Project

Phase 1 Report

The first thing we will discuss is our approaches to design. We emulated Linux Ubuntu 14.04 LTS. We made our own website that actually synchronizes with the MySQL database itself. Instead of a single-page application, we separated the web application into four main components/pages: index, register, sign in, and user credential recovery. The index page redirects to the register page. We also mainly developed outside of the Ubuntu box. Since we knew that our initial script was in bash. So once we retrieved that initial script we were able to use the credential which were from the PHP MySQL connection. For this decision we thought it would be the best course of action because it would be easier for it to test it right then and there to see if it could connect to the server. We choose this way over the alternates because we believe that our way would make it easier to test, use, and be more efficient.

For implementation we thought about using the essentials. First and foremost for convenience sake we wanted a user created that could easily authenticate with mysql so that anyoneone running the script could immediately access the database. The script uses one command to install a unix lamp stack before delving into installation of php my admin. We then give all permissions to a user named “binker” to ensure authentication and access is easy and convenient for them. We then create a database named Comp424. Afterwards we create a table called student with a number of attributes including: id, first name, last name, birthdate, email, password, and today. Today is the current date and is to be used for checking if there are too many attempted logins in one day. By querying our sql database with a count function for this value for each unique id, we can than further check if there have been any malicious attempts to force in to a users account.

We have implemented in our project a web page that we got to work operational. We created an index page that allows us to redirect to the register page, login page, and a recover credential page. As we created the web applications and scripts we tested them right away. We used an Ubuntu image using Oracle VM VirtualBox. We have created server provisioning bash scripts. Which also consisted of the installation.sh file. So that we could set up the LAMP stack. We also felt it would be best if we tried to make it so that the installation script would move the contents to the correct folder. So that way the user doesn’t have to drop it into the folder. We tried to get the security implemented. But we could not get it to work the way we wanted because we kept running into errors.

Our team also got a USER for the MySQL database, and finally the tables required for the web application. The webpage itself is up and working. We created an index page that allows us to redirect to the register page, login page, and a recover credential page. As we created the web applications and scripts we tested them right away. We used an Ubuntu image using Oracle VM VirtualBox.

Through our experience we had a bunch of problems and issues. Some we were able to fix. Others we couldn't fix entirely. We also encountered was that half of our team used PC's and the other half used Apple Laptops. So in order to bypass this problem we came to the decision to use MAMP. So that we could create the web application outside of Ubuntu. There were also a handful we also kept running into. The registration page would not work properly because the database name would be case sensitive. There would even be times where everything would work the way it's supposed to on some of our teammates' laptops, but the PHP wouldn't work for the others. The connection to the database would even work properly. Sometimes our internet on our Oracle VM VirtualBox wouldn't even work. So it made it really hard to fix things.

Before we started this project we had to ask ourselves a couple questions first. As we do this project should we focus more on the security aspect than usability? Or would it be more beneficial to us and the clients to lean more towards usability. At first we weren't entirely sure, but eventually we all came to the conclusion that we should focus more on usability. However this does not mean we neglected security. We decided to do this because we felt it was more of a top priority for us to get everything working so we could use them and learn more from what we were trying to do. After we have that taken care of then we focused on the security. We tried to focus on certain attacks like brute force, SQL Injection, buffer overflow, XSS, and Cross Site Request forgery.

Lastly if we were to start this whole project all over again. We would most likely go about doing a couple things differently. We would setup the web page to be visually more appealing. So that way it would be more obvious what page the client is in. We also try to implement better security. We would try to add in an additional feature where in order to login in more securely the system will email you a code for you to input along with your correct username and password. We also focus more heavily on password requirements. Where we would want the password to at least have a minimum of 16 characters, at least one number, one symbol, and at least one empty space to ensure the strongest password to hold against attacks. We could also ask better security questions. So that the user can better prove who he or she really is and can enter their account. We would also focus more on errors as well. So basically in case we do run into future problems. The system can automatically tell us what kind of errors were encountering.