Patrick O’Brien, Marco Rojas

5/1/2019

CSC 160 Combo

Final Project Proposal

For our final project, we would like to design a program that serves as a tool that can perform various hashing functions. One tool would allow the adding of users to a password database lists, either with or without salt on the password. Accordingly, a user will be able to login and it is checked against the database. Another tool would be a dictionary password attack tool that would use the hashes from a list of supplied words to attempt to determine the plaintext of a list of unknown hashes.

To accomplish our project, there are several objects, files, and arrays. We will have classes that create 3 classes: one object that creates MD5 hashes from strings like passwords; one that creates MD5 hashes with salt added to the password; and the driver class. We will each make one of the object classes and both work on menu options in the driver class for program functions. There will be files that contain passwords and hashes for the dictionary attack, and there will also be files that store the database of user login information. In the driver class, we will have menus for each of the options.

In order to design a program that can accomplish these tasks, we will need to read in two text files, one is the list of 10,000 most common passwords, the other is a list of hashes that we are attempting to determine the plaintext of. We will have a class that will take information from the plaintext file and create a two-field object out of each line: one field containing the plaintext password, the other containing the hash of the password. We will create one class for MD5 and look at using a second as well. After we have the hashes of the Passwords.txt passwords, we can compare the unknown hashes in Hashes.txt to of Passwords.txt hashes to determine what the plaintext is of our unknown hashes. To allow for a user to check their password against our list of known passwords, we will hide the user text so any onlookers cannot see it. As a final use of hashes, we would like to allow a user to input a file that was downloaded, compare its hash to its checksum to confirm file integrity.

To complete this project, we will likely each have a class object that we do the primary writing for. One of the objects will be the password hash object, the other will be and object that reads a file and compares it to a checksum value. The driving class we will work on together and function as a menu to choose the option of what function the user would like to perform.