William Croslen
Diego Aguirre CS 2302

Lab 1 Report

Introduction -

The problem that we were trying to solve is to recursively find the passwords for 100 different accounts by brute force, given a txt file with 100 records containing information about 100 system accounts. Each record has a username, a salt value, and a *hashed* password.

Solution -

My solution to this problem is to recursively generate every possible numerical password between 3 and 8 digits. To do so I had two parameters, one is of the string to test the password and the other is the number of digits long the string has to be. I incremented the string by checking every combination of digits 0 - 9 and added 1 to the string each recursive call.
If the string length was equal to the number of digits that were being checked then it would call the check method. The check method combined the string with the salt value and checked if that new string hashed matched with the hashed password. It did this with every user and if it matched, then it would print out the user as well as the corresponding password.

Experimental Results -

In the experiments that I performed, I tested my code with different User names, passwords and salt values. I did so by using the hash256 with a concatenated value of the numbers and salt values that I tested which were:

Test1,haller,8eb2595d765f8d026e9fafbcbbdeeb15eed34e5e96ee451516601b64289fbe56
Test2,abcdef,2db04fcd9a844acc292bee12504cb3fb11d9868c407cee96f8774b682e45f1d3
Test3,abcdef,cbbf60f8a2fcdb8751810b38d05112fa851b05e85609f0e4ea7c608ebddebe30
SkeetSkeet,poooop,297f19821c1381ddc00da51433abeb8ba13353424ecf41623b8a666d47469556

All of the passwords that were hashed were found during the testing of my code.

Conclusion -

During this project I became more familiar with the syntax of python and I learned how passwords in the real world actually have hashed values and salt values to better secure the password. I also learned how ineffective brute force can actually be as a hacking technique because the length of the password can easily prevent hacking from this method as it would continue to take longer to generate the proper password.

William Croslen
Diego Aguirre CS 2302

Appendix – Source code

```
import hashlib

def hash_with_sha256(str):

   hash_object = hashlib.sha256(str.encode('utf-8'))

   hex_dig = hash_object.hexdigest()

   return hex_dig

 def main():

   hex_dig = hash_with_sha256('This is how you hash a string with sha256')

   print(hex_dig)
```

Academic Honesty

I, ***William Croslen***, certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.