Chase Richards

CMPSCI 3780

Project 2 Analysis

	Hash and Hash + Salt Password Cracking Range	Hash and Hash + Salt Number of Password Guesses Range
One Letter Password	0.034 - 0.061 seconds	1 - 26
Two Letter Password	0.070 – 0.373 seconds	27 - 702
Three Letter Password	0.223 – 10.430 seconds	703 - 18,278
Four Letter Password	7.653 – 266.006 seconds	18,279 – 475,254
Five Letter Password	193.464 – 2251.164 seconds	475,255 – 12,356,630
Six Letter Password	2955.71 – 75000 seconds (50 min – 21 hrs)	12,356,631 – 321,272,406
Seven Letter Password	75500 – 1,997,500 seconds (21 hrs – 554 hrs)	321,272,406 – 8,353,082,582
Eight Letter Password	2,000,000 – 51,937,080 seconds (555 hrs – 601 days)	8,353,082,583 – 217,180,147,158

***Red numbers are estimates based on multiplying by 26

In my program, I included the time and the number of guesses it took to crack each password in the output. In testing I gave the password files passwords ranging from a-z, so for two letter passwords, I tested passwords aa, jk, and zz. For six letter passwords, I gave the files aaaaaa, apples, and zzzzzz. I tested using a similar variety of passwords for all the other password lengths. I did this with both hash and hash + salt passwords up to six character passwords, but not with seven or eight because they were taking more time then I had available to leave my computer sitting.

First, looking at the number of guesses range I noticed that each lower bound times 26 is equal to the upper bound. As seen in the table above these numbers grow significantly the closer you get to an eight letter password. Once we get to a nine character password we are in the trillions of guesses. You can also see that the average of cracking hash and hash + salt grows by about 26 times for each character that is added. The time it took my machine to crack passwords in the later letters of 6 letter passwords (ex. zzzzzz), and greater number of letters was too long for me to test, so I made the estimates seen in red.

Hash + Salt passwords are more difficult to crack. Though the results of my project don't fully show how much more difficult it would be in real practice. The reason is when I was cracking the hash + salt password I knew what the salt was going into it because I chose the first password in the file, so I did not have to worry about trying to crack the salt as well.

For my machine brute forcing a password whether it's just hash or hash + salt that begins with the beginning of the alphabet towards the middle will take a few days but as the first letter gets later in the alphabet it takes up to 20 days. Once it gets to eight characters it would take anywhere from 20 days to over a year which in my opinion would not be a worth while use of time to break one password.

Now Imagine brute forcing a password eight or more characters that combines upper and lower case letters, symbols, and numbers. These times would increase significantly because rather than multiplying by 26 for each letter added, you could be multiplying these times and guesses by over 70 depending on

how many special characters are allowed. Overall, if even if the password only allows lowercase letters I would say an 8 character password would be safe against a brute force attack because the user will most likely update their password before the attacker has completed the brute force attack.