# Access Control Assignment – Password Cracking Questions

CPSC 348 – Computer Security

Fall 2020

1. How many lines are in small\_wordlist.lst? large\_wordlist.lst? (You can open them in Notepad++.)
2. Read the first 100 passwords in small\_wordlist.lst and tell me your favorite one.
3. [Read about word-mangling](https://www.tunnelsup.com/getting-started-cracking-password-hashes/) (see the section “Word mangling rules”) and answer the following question.

In “Password Cracking Results.xlsx”, look at the table *Small vs. Large Wordlist*. You should see that, in the first comparison between test cases 1 and 3, there is something like a 25 – 50% increase in the time to crack, whereas in the second comparison between test cases 2 and 4, there as much as a 1,000 – 1,500% increase. (Similarly with the third comparison (test cases )) This is because test cases 1 and 3 do not use word-mangling whereas test cases 2 and 4 do.  
  
Why does word-mangling affect large wordlists so much more than small wordlists?

1. In “Password Cracking Results.xlsx”, look at the table *Salted vs. Unsalted*. You should see that, in the first comparison between test cases 5 and 9, there is something like a 0 – 10% increase in the time to crack, whereas in the third comparison between test cases 7 and 11, there as much as a 4,000 – 6,000% increase. This is because test cases 5 and 9 use the small wordlist whereas test cases 7 and 11 use the large wordlist.  
     
   Why does salting affect large wordlists so much more than small wordlists?
2. In “Password Cracking Results.xlsx”, look at the tables *Small vs. Large Wordlist* and *Word-Mangling vs. None*. For each row in each table, compute the increase in number of passwords cracked. What is the average increase by using a large wordlist? What is the average increase by using word-mangling? Based on this, do you think it is more important for an attacker to use word-mangling or have a large wordlist?
3. Compare the time to crack test case 11 to that of test case 12. Test case 12 takes around twice as long as test case 11, despite that it’s cracking a much smaller password file. How may entries are in “sha1-salted.txt”? How many entries are in “sha1-salted-tiny.txt”? Despite this, why is test case 12 so much slower than test case 11? (Notice that it is the only test case that uses a salt alongside both the large wordlist and word-mangling.)
4. Compare the time to crack test case 9 to that of test case 13. What is the only difference between the test cases (besides number of passwords cracked and time to crack)? Why does that difference cause such a dramatic difference in the time to crack?
5. [Read some good old Wikipedia](https://en.wikipedia.org/wiki/Cryptographic_hash_function" \l "Cryptographic_hash_algorithms) about the three hash algorithms you cracked in this assignment. Which of them are still considered secure today?