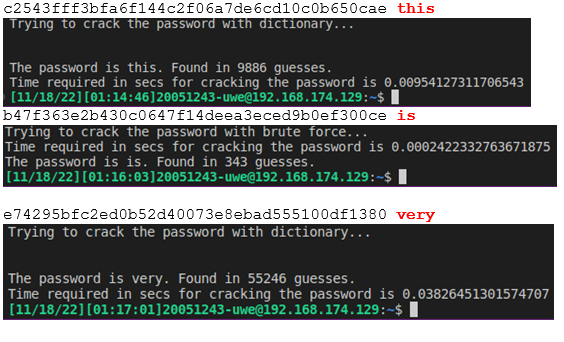
The hash from Set A is a set of passwords which contains a maximum of six lower-case letters and digits, the Set B is a set of BCH password codes.

**Set A password:**

Let’s begin with the Set A, the way that I cracked Set A password used three different attack method for it, which are dictionary attack, hybrid dictionary attack and finally brute force attack, and I will elaborate each method below:

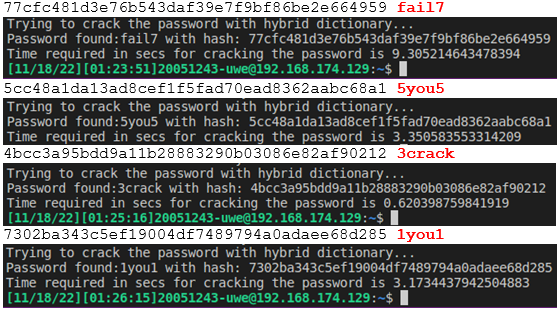
**Dictionary Attack:**



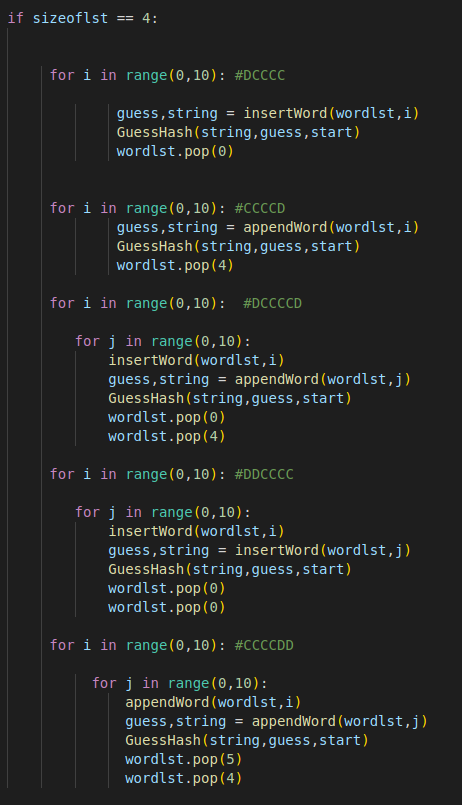
***Figure 1- Result on cracking vocabulary words***

I used a dictionary list which contains 10 million passwords, although 10 million passwords sound gigantic it literally spent no time to crack the words within the dictionary list, in other words it is efficient on cracking passwords based on vocabulary. And the Figure 1 have shown that the time I have spent on cracking vocabulary word passwords, most of them could be found within a second.

**Hybrid Dictionary Attack:**



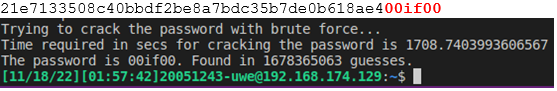
**Figure 2 – Password in vocabulary word mixed with digits**



**Figure 3 - Hybrid attack try all combination of a word in the list under 6 lengths**

This attack will be used if the dictionary attack could not find any password within it, my program will start crack the password with Hybrid dictionary attack, Hybrid Dictionary Attack simply just iterate the password cracking process as the regular dictionary attack did but the only difference is the function will check every single word in my dictionary list which their length is less than 6, if that word is smaller than 6 words than my function will insert digital numbers from 0-9 to the word, for example like the password “fail7” shown in **Figure 2**, it is a combination of vocabulary word “fail” + “7”. Assume my word list contains the word “fail”, then the function will try all combinations of digital numbers and the vocab word under 6 lengths like “fail0, fail1…fail9” or “0fail,1fail,9fail…” **(See Figure 3).**

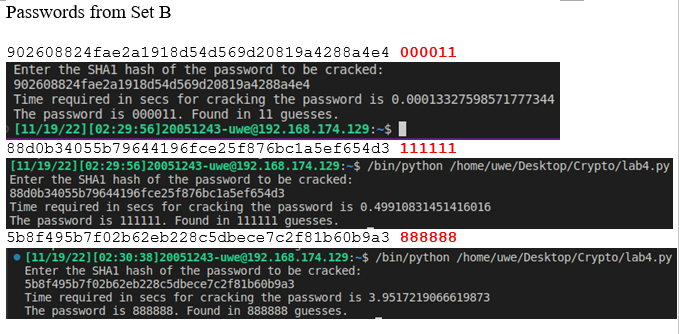
**Brute Force Attack**



**Figure 4 – Brute force attack the password**

Finally, if both methods could not recover the password which means the password most likely it is not a vocabulary just a combination of random letters and numbers or the word that doesn’t contains in the dictionary list. The brute force attack will try every single combination within lower case character and digital numbers. In ***Figure 4*** the password “00if00” that the if was not contained in my dictionary list that brute force did spend near 30 minutes to crack the password.

**Set B**



***Figure 5 – The time for cracking BCH code***

Set B password is a set of BCH code which the program I have done in Task 2 provided a big help, all I need to do is simply transfer all BCH code to a hash in order to find the matching password. Overall the cracking time spent under 5 seconds.