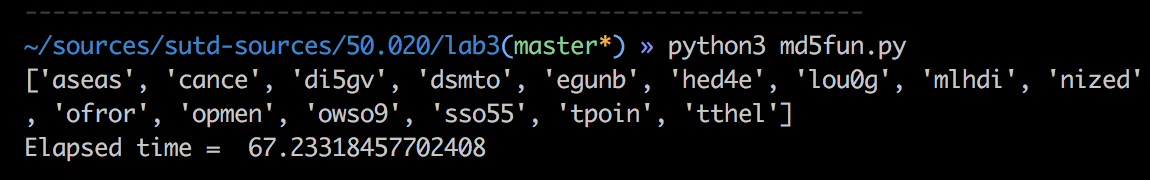
50.020 Security

# Lab 3 by Nguyen Trung Huan (1001704)

Brute-force to crack 5-character loweralpha-numeric passwords

The brute-force method implemented in *md5fun.py* did not use the provided dictionary. Instead, it uses a nested for-loop to compute all different permutations and use real-time hashing to check against the hashes. Once it reaches 15 passwords, the program termintes.

Timing is 67 seconds.



Compared to this, using pre-computed rainbow table yields much better results:

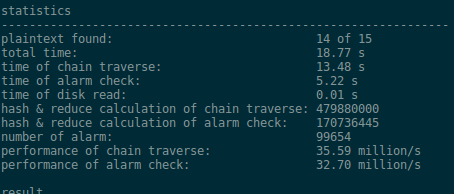
|  |  |
| --- | --- |
| For m = 1000000, d = 5000: | For m = 1200000, d = 1900: |
| https://lh3.googleusercontent.com/BWoakJoiSIGigS4nv-BCB0J1Q2y_Ys3gCLa1Od1DA8eVdLLrC33TFxF2bREWHFfZbSadXYF1eh-T8ZDpZB2b3kcJPbeqmQV7DM-WxnwtO4-0N8oRJk2ucbVxsnRPlcmUTS8A9wD- | https://lh5.googleusercontent.com/IZU_wukhReoYvqMZoGwlYulrZf_EttcyUcSgZmMcDb_62gD15QFY5XDBlyoSPBDi4pLEN43wA_GI1_gQyls1Bck8Fs-LOPOY0upjLlYUF1aLjv8CsJtQQyvwreLyK7JNttoKiyqs |

Due to known password length, we could allow the chain length to be shorted, while slightly increasing the number of chains. This effectively reduced the running time from 7.45s to 1.41s.

Rainbow table to crack salted passwords

Due to increasing the length of password by 1, the input space now becomes 366 instead of 365. This increases the size of the rainbow table by a significant amount, generating and storing computing the rainbow table much more expensive.

However, with m = 1,000,000 and d = 8,000, 14 out of the 15 salted passwords were broken in 18.77s.



With this implementation using rainbow table, the approaches towards breaking normal passwords and salted passwords are quite similar. The main difference lies in choosing the chain length and the chain number. For longer passwords, the input space is larger, hence we need to create larger rainbow tables.

Cracking passwords

Using the code from the first part of *md5fun.py*, we can crack 4 passwords from the weak portion. Inside the submitted file, there are snippets of code that are used to break the hashes file with different difficulty.