Secure programming -course project: Password recovery tool

Description:

Program works with MD5, SHA-256 and SHA-512 hashes, also text file is needed which is full of passwords, all in separate rows. Program compares user given hash to already existing password that are in text file which are changed into hashes for comparing.

Program can be used to test passwords that you use every day or to penetration testing purposes.

Structure:

Program works with Python 3. Program uses hashlib library to hash passwords. Program works with text file and python file which works with given text file. First program is given hash that user wishes to checkout if it is in some password list found in internet for example. After giving hash user gives password list where user wants program to check hash. Program identifies next if hash is MD5, SHA-256 or SHA-512 by its length. If hash is identified as some of these hash types program continues to that hash's function to work. First flag value is set to zero(0) in function text file is opened for reading and after that gone through with For-loop. In For-loop every password in text file will be changed to hash and compared to user given hash. If there is match word is given to pass_print function that prints recovered password in text to user, also flag is given value one(1) which prevents IF-statement 'if flag == 0' from working. If there is no match 'Password not in the password list' will be printed.

Secure programming solutions:

Because of offensiveness of the program, only little secure programming is needed. Program can be used for basic brute-force attack or more sophisticated attacks testing attacks. This all depends on text file that is given, program just hashes anything that is given to it and compares it to user given hash.

Not implemented features:

I didn’t have time to implement parallel programming features. This would have possibly cut time spent going through very large text files.

Improvements:

To make program use parallel programming, connect this program to other one were, it could get hash straight away.

Hash salting isn’t included in the program so salt detection or make program manage with salt so it would work still.

Faster processing and stress resistance.

Generating new passwords from already existing ones.