Mobile Device Forensics involves the collection of forensically sound evidence (or admissible evidence that can be presented to the court of law). Using the program you developed in lab 2, find the hash code of the following two files MDF-SF1.docx and MDF-SF2.docx. Since these two files are password protected, write a Python program to break the password. After breaking the password find the hash code again. Compare the values of hash code before and after breaking passwords. List and use two more password management tools for breaking password. Compare the hash codes with those generated by your program. Make sure that you are program is documented. It is important that your program is uploaded in GitHub as well or any other web based repository

MD5 Hash of both documents before any cracking :

MDF-SF2.docx : dd06b1c03f4ad67c99d4f371f4001fa4

MDF-SF1.docx : 69a6db156d051c48e655d5a95cff84b4

How to crack a password.

I wanted to use an open source cracking tool and there are many. I picked john the ripper and hashcat

Next option was which way to crack.

There are three main way to crack

Dictionary Attack

Dictionary attacks are just what they sound like: you [use the dictionary](http://www.quickanddirtytips.com/education/grammar/the-oxford-english-dictionary-needs-your-help)to find a password. Hackers basically have very large text files that include millions of generic passwords, such as password, iloveyou, 12345, admin, or 123546789.

Mask/Character Set Attack

If a hacker can’t guess your password from a dictionary of known passwords, their next option will be to use some general rules to try a lot of combinations of specified characters. This means that instead of trying a list of passwords, a hacker would specify a list of characters to try.

For example, if I knew your password was [just numbers](http://www.quickanddirtytips.com/education/math), I would tell my program to only try number combinations as passwords. From here, the program would try every combination of numbers until it cracked the password. Hackers can specify a ton of other settings, like minimum and maximum length, how many times to repeat a specific character in a row, and many more. This decreases the amount of work the program would need to do.

So, let's say I had an 8 character password made up of just numbers. Using my graphics card, it would take about 200 seconds--just over 3 minutes--to crack this password. However, if the password included lowercase letters and numbers, the same 8 character password would take about 2 days to decode.

Bruteforce

If an attacker has had no luck with these two methods, they may also "bruteforce" your password. A bruteforce tries every character combination until it gets the password. Generally, this type of attack is impractical, though--as anything over 10 characters would take millions of years to figure out!

The first method is to use a wordlist. Tere are many avavnble online . One of the best seems to be the rockyou list. (BOWES, 2016)

This list is the result of a hack on rockyou.com It had a data breached that resulted in the exposure of over 32 Million user accounts. To compound the severity of the security breach, it was found that RockYou are storing all user account data in plain text in their database, exposing all that information to attackers

(Cubrilovic, 2016)

Stages of cracking

Pass your file into office2john a python program . it extract the hashes. One example is: *365-2013-password.docx:$office$\*2013\*100000\*256\*16\*d4fc9302eedabf9872b24ca700a5258b\*7c9554d582520747ec3e872f109a7026\*1af5b5024f00e35eaf5fd8148b410b57e7451a32898acaf14275a8c119c3a4fd*

Next stage is inputting this hash and other options into john.

./john --wordlist=rockyou.lst doc.hash

.

I did not have the skills to write a program to do this . I decedid to aoutomate the process. I wrote a scipt with ask for the office file name the the word list you want to use . It will then pass the details to office2john and output the results to a file , then inut this with outer varables into the john application.





