Fourth Homework Cybersecurity

Andrea Morelli 1845525

1 First Ideas

The basic idea that i thought to solve this homework was to get get a dictionary of most used passwords and for each word get the ten different salts and concatenate the password to those.If there is a match in the resulting hash with the hashes to crack then return the password.

2 Trials and Errors

I starting implementing this simple idea using hashlib to generate the sha256 hash and utilizing the list of 10000000 most used passwords. I first start loading in the memory the lists of passwords and hashes to crack and then loading the dictionary. For each word in the dictionary i concatenate it to every salts and get the hashes.If there is a match then i procede to print the password found. The idea seemed reasonable but i couldn't get any results not even a password cracked. At this point i started analyzing the way in which the salts and the passwords were generated and i thought that a end of line character could be present in every password and since i was doing a strip on the passwords of the dictionary i could be missing something.

3 Final results

After a bit of time spent trying to find the passwords i removed the strip in the passwords of the dictionary loaded and saw the results.
123456
password
12345678
qwerty
123456789
12345
111111
1234567890
qwerty123
1q2w3e
Every password has an end of line character at the end and those passwords corresponding to the ith row of the given salts and passwords to crack.

4 Final script source code

The first script i wrote was :

```
import hashlib
import fileinput
file=[]
hashes={}
def loadfile():
    with open("12bits_salts_and_salted_passwords_file.txt","r") as f:
        for i in f:
```

```
            i=i.strip()
            file.append((i[:12],i[12:len(i)]))
def solver():
    out=open("out.txt","w")
    loadfile()
    hashes={i[1] for i in file}
    with open("10-million-password-list-top-1000000 (1).txt","r",errors="ignore") as f:
        for i in f:
            for j in file:
                a=j[0]
                b=i
                val=hashlib.sha256((a+b).encode("utf-8")).hexdigest()
                if val in hashes:
                    print(i)
                    with open("out.txt","a")as f:
                        out.write(i)
                    break
if __name__=="__main__":
    solver()
```