Password Cracking

Code can be found here: https://github.com/c2003-tamu/413

Environment Setup

- Clone git repository
- Navigate to passwordcracking directory
- Ensure python3 and john the ripper are installed
    - sudo apt install john
    - sudo apt install python3

User Configuration

To ease the grading effort, I made a bash script to add users to a machine.

NOTE: do not run this on your local machine, ensure that a VM is being used.

To run script:

- Navigate to 413/passwordcracking directory
- Run command: ./addusers.sh

This script should add the following users and passwords to your machine:

User	Password
user1	4056
user2	34867
user3	598212
user4	7728694
user5	13063382

NOTE: These passwords were randomly generated using the script pwordgen.py and can be found in pwords.txt. Digits 0-9 were used for simplicity. MD5 hash was also used for simplicity.

John the Ripper

To compile the necessary files to run john the ripper, run command: sudo unshadow /etc/passwd /etc/shadow > unshadow.txt
NOTE: The unshadow.txt used for the demonstration is not included in the zip file.

NOTE: this was found in this youtube video: https://www.youtube.com/watch?v=tJRz9j2REb4

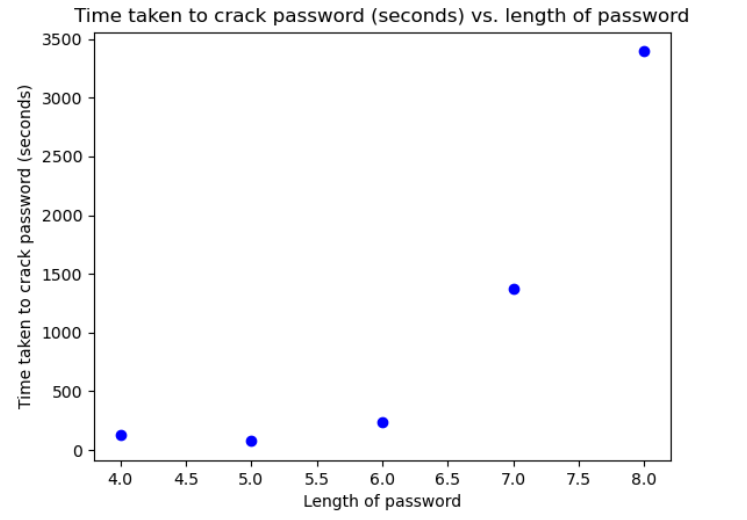
To run john the ripper on your hashes, run command: john –format=md5crypt –incremental=digits unshadow.txt

This will brute force the passwords on your machine, as seen below:

```
vboxuser@pwcrack:~/Desktop/413/passwordcracking$ john --format=md5crypt --increm
ental=digits unshadow.txt | while read line; do echo "$(date) $line"; done
Warning: OpenMP is disabled; a non-OpenMP build may be faster
Wed Mar 26 10:36:23 PM UTC 2025 Loaded 5 password hashes with 5 different salts
(md5crypt [MD5 32/64 X2])
Warning: MaxLen = 20 is too large for the current hash type, reduced to 15
Press 'q' or Ctrl-C to abort, almost any other key for status
Wed Mar 26 10:37:39 PM UTC 2025 34867            (user2)
Wed Mar 26 10:38:30 PM UTC 2025 4056             (user1)
Wed Mar 26 10:40:18 PM UTC 2025 598212           (user3)
3g 0:00:12:56 0.003861g/s 6136p/s 14424c/s 14424C/s 1801418..1800191
Wed Mar 26 10:59:18 PM UTC 2025 7728694          (user4)
Wed Mar 26 11:32:56 PM UTC 2025 13063382         (user5)
5g 0:00:56:32 0.001473g/s 12331p/s 15611c/s 15611C/s 13069681..13021192
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Exponential Time

When we plot the numbers found above, we get the following scatterplot:



This scatterplot shows an obvious exponential relationship between length of password and time taken to crack password. This plot was generated with scatterplot.py