Derrick Lee
Lab Exercise 2
CSCI 180
October 12, 2019

1

Modes used:

- wordlist (given dictionary + online password list)
- wordlist + rules (default, Single, KoreLogic)
- Markov
- Incremental (default)
- Incremental (charset with cracked passwords)

The given wordlist given was used along with a wordlist of top passwords found on GitHub at danielmiessler/SecLists.

Commands run that cracked passwords (with corresponding user passwords):

```
$ john --wordlist=dictionary.txt --format=raw-MD5 target.txt
# user87, user88, user92, user93, user94, user00, user98

$ john --wordlist=dictionary.txt --format=raw-MD5 --rules target.txt
# user91, user89, user15, user97, user32, user14, user13, user12, user99,
    user02, user01

$ john --incremental=digits --format=raw-MD5 target.txt
# user95, user56

$ john --markov --max-length=20 --format=raw-MD5 target.txt
# user79, user84, user86, user90, user20, user96

$ john --wordlist=password.lst --rules --format=raw-MD5 target.txt
# user49, user39, user19, user31

$ john --max-length=12 --format=raw-MD5 target.txt
# user11, user10, user09, user36, user72

$ john --wordlist=dictionary.txt --rules=all --format=raw-MD5 target.txt
# user05, user03, user28

$ john --incremental=charset --format=raw-MD5 target.txt
# user18, user74, user47

$ john --wordlist=10-million-password-list-top-100000.txt --format=raw-MD5
    target.txt
# user29
```

```
$ john --wordlist=10-million-password-list-top-100000.txt --rules --format
  =raw-MD5 target.txt
# user07, user06, user40

$ john --wordlist=10-million-password-list-top-100000.txt --rules=
  KoreLogic --format=raw-MD5 target.txt
# user67, user66, user85, user57, user75

$ john --wordlist=dictionary.txt --rules=KoreLogic --format=raw-MD5 target
  .txt
# user65

$ john --wordlist=10-million-password-list-top-100000.txt --rules=Single
  --format=raw-MD5 target.txt
# user69

$ john --wordlist=10-million-password-list-top-1000000.txt --format=raw-
  MD5 target.txt
# user46

$ john --wordlist=10-million-password-list-top-1000000.txt --rules --
  format=raw-MD5 target.txt
# user48
```

Commands with no cracked users or canceled due to excessive runtime:

```
$ john --wordlist=password.lst --format=raw-MD5 target.txt
$ john --wordlist=all.lst --rules --format=raw-MD5 target.txt
$ john --wordlist=passwords.lst --rules --format=raw-MD5 target.txt
$ john --mask=?1?1?1?1?1?1?1?1 --1=[A-Z] --format=raw-MD5 target.txt
$ john --mask=?1?1?1?1?1?1?1?1 --1=[A-Z] --min-length=8 --format=raw-MD5
  target.txt
$ john --markov=200 --max-length=7 target.txt --mkv-stats=markovstats
$ john --markov=200 --max-length=7 --format=raw-MD5 target.txt --mkv-stats
  =markovstats
$ john --markov=10 --max-length=7 --format=raw-MD5 target.txt --mkv-stats=
  markovstats
$ john --markov=100 --max-length=7 --format=raw-MD5 target.txt --mkv-stats
  =markovstats
$ john --incremental=charset --max-length=12 --format=raw-MD5 target.txt
$ john --wordlist=dictionary.txt --rules=Wordlist --format=raw-MD5 target.
  txt
$ john --wordlist=dictionary.txt --rules=Extra --format=raw-MD5 target.txt
$ john --wordlist=dictionary.txt --rules=Wordlist --format=raw-MD5 target.
  txt
$ john --wordlist=dictionary.txt --rules=Extra --format=raw-MD5 target.txt
$ john --wordlist=10-million-password-list-top-100000.txt --rules --format
  =raw-MD5 target.txt
$ john --wordlist=10-million-password-list-top-1000000.txt --rules=Single
  --format=raw-MD5 target.txt
$ john --wordlist=10-million-password-list-top-1000000.txt --rules=Single
  --format=raw-MD5 target.txt
$ john --wordlist=10-million-password-list-top-1000000.txt --rules=
```

```
    KoreLogic --format=raw-MD5 target.txt
$ john --incremental=charset --format=raw-MD5 target.txt
$ john --incremental=charset --max-length=10 --format=raw-MD5 target.txt
$ john --loopback --rules --format=raw-MD5 target.txt
$ john --loopback --rules=all --format=raw-MD5 target.txt
$ john --loopback --rules=all --format=raw-MD5 target.txt
$ john --mask=?1?1?1?1?1?1?1?1 --1=[A-Z] --min-length=8 --format=raw-MD5
    target.txt
$ john --markov --max-run-time=10 --format=raw-MD5 target.txt
$ john --markov --max-run-time=50 --format=raw-MD5 target.txt
$ john --incremental=Digits --format=raw-MD5 target.txt
$ john --incremental=Digits --max-length=4 --format=raw-MD5 target.txt
$ john --incremental=Digits --min-length=4 --max-length=8 --format=raw-MD5
    target.txt
$ john --incremental=Digits --min-length=8 --max-length=16 --format=raw-
    MD5 target.txt
$ john --incremental=Digits --min-length=8 --max-length=12 --format=raw-
    MD5 target.txt
$ john --incremental=Alnum --max-length=8 --format=raw-MD5 target.txt
$ john --incremental=Alnum --max-length=4 --format=raw-MD5 target.txt
$ john --incremental=Alnum --max-length=6 --format=raw-MD5 target.txt
$ john --incremental=ASCII --max-length=6 --format=raw-MD5 target.txt
```

Passwords cracked (54 password hashes cracked, 46 left):

```
user00:hashemi
user01:8ferret
user02:ruben6
user03:criminal16
user05:f00tba11
user06:dingding1
user07:goodday1
user09:babigirl1
user10:candy1992
user11:sunset15
user12:homedepot5
user13:riverside!
user14:butthead2
user15:motorhead1
user18:1mateo4
user19:1medical
user20:1memme
user28:1nothing1
user29:1nternet
user31:1onelove
user32:1orange
user36:1susan2
user39:1teddybear
user40:1texasboy
user46:1Vipers
user47:1webstar
user48:1westsider
user49:1winnie
user56:20013694
user57:2006acura
user65:20hopedale
user66:20inches
user67:20september
user69:210592w
user72:2123546a
user74:212head
user75:212sammyd
user79:maxx13
user84:a1234666
user85:21norway
user86:portinga
user87:casper
user88:badone
user89:lebanon1
user90:fildaman
user91:tacoma1
user92:brookstone
user93:knockers
user94:braindamage
user95:8661234
user96:iamadam
user97:smoesmoe
user98:qwertyui
user99:bubbles4
```

2

Using a variety of different modes listed above, my password was not cracked. Since I use a password manager (previously KeePass and now Bitwarden), many of my passwords are randomly generated, and thus difficult to crack. Depending on the website, they range from short (around 16 characters) to longer ones (up to 64 characters).

An example password would be as follows (**NOT** a password in use, this is newly generated just for this assignment):

M&wncHtDTTWfp^merr^KPEd8%m*N9ef3

The more effective methods of password cracking from Q1 were dictionary based attacks which would not be able to crack a randomly generated password as shown above (assuming it's not a reused password that could potentially end up on a password list online).

Using randomly generated passwords provides very good protection against various attacks. For example the one above has a character set of uppercase and lowercase characters, numbers, and special characters (!@#$%^&*) – a total of 70 possible characters.

With a minimum of 16 characters, the number of password guesses required in in order to correctly find a given random password is as follows:

$$70^{16} = 33232930569601000000000000000$$

For even longer passwords up to 64 characters long:

$$70^{64} \approx 1.219 \cdot 10^{118}$$

Without having parts of the password as words or phrases, the only option to crack these passwords would be an incremental brute force attack which would require an immense amount of time and computational power. This is, again, assuming that these passwords were not leaked elsewhere in a plaintext database. If a password is reused on a website or service that stored passwords in plaintext and that password were to be part of a data breach, it would be part of a dictionary attack to easily retrieve the same password that is properly hashed.

While a lot of of my passwords are randomly generated, there are still a number of them that are memorized, such as the password to unlock my password manager. These passwords are quite lengthy with a minimum of 20 characters. While not as secure, they're still also difficult to crack. Some of my older passwords are not very secure, found in several data breaches and along with some associated older recently accounts hijacked (though they're not in use and the passwords were easily changed to a random one).