

## Частина 1:

Генератор створює різні типи паролів з певною імовірністю:

- 75% - береться пароль з топ 100000 найпопулярніших паролів
- 10% - беруться з топ 110 найпопулярніших паролів
- 10% - схожі на справжні, генеруються за допомогою найчастіше використовуваних слів, цифр та спецсимволів
- 5% - генеруються випадковим чином

Випадкові паролі генеруються довжиною від 6 до 16 символів, кожен з яких рандомно обирається серед латинських букв, цифр та символів пунктуації

Схожі на справжні паролі генеруються наступним чином:

- 1) Обирається випадкова довжина від 6 до 16 символів
- 2) Поки довжина недостатня додаємо
  - a) З імовірністю 65% слово з найживаніших англійських слів (з імовірністю 35% перша буква слова змінюється на букву в верхньому регістрі)
  - b) З імовірністю 25% випадкова цифра
  - c) З імовірністю 10% додається один із спецсимволів (\*\_!+-)
- 3) Якщо довжина перевищує обрану, обрізаємо зайві символи

Для кожного з типів хешування (MD5, SHA1, bcrypt) ми створюємо 100000 паролів за допомогою генератора, хешуємо їх відповідним чином та зберігаємо в файлі csv. Для SHA1 поряд в файлі зберігаємо сіль.

```
2020-11-03 20:39:52.342 | INFO | __main__:<module>:21 - Creating password generator
2020-11-03 20:39:52.343 | INFO | __main__:<module>:24 - 100000 password hashes will be generated for each type of hashfunc
2020-11-03 20:39:52.343 | INFO | __main__:<module>:26 - Generating MD5 password hashes and saving into ./hashes/md5.csv
100% (100000 of 100000) | #####| Elapsed Time: 0:00:06 Time: 0:00:06
2020-11-03 20:39:59.028 | INFO | __main__:<module>:36 - Generating SHA1 password hashes and saving into ./hashes/sha1.csv
100% (100000 of 100000) | #####| Elapsed Time: 0:00:09 Time: 0:00:09
2020-11-03 20:40:08.632 | INFO | __main__:<module>:47 - Generating bcrypt password hashes and saving into ./hashes/bcrypt.csv
100% (100000 of 100000) | #####| Elapsed Time: 0:21:54 Time: 0:21:54
2020-11-03 21:02:03.406 | SUCCESS | __main__:<module>:58 - Done!
```

## Частина 2

Згенеровані паролі були взяті з <https://github.com/vladlytvynenko/crypto-labs/tree/master/lab4>

### MD5

Спробуємо розшифрувати паролі захешовані за допомогою MD5, для будемо використовувати список з топ 100 тисяч найпоширеніших паролів. Для цього будемо використовувати "straight" mode (Dictionary attack). Він просто перебирає всі паролі що вказані у заданому файлі і намагається знайти такий же хеш у заданому файлі з хешами невідомих паролів.

***hashcat --optimized-kernel-enable -w 4 --force -a 0 -m 0 generated-md5.csv common100Kpass.txt -o output-md5.csv***

```
Windows PowerShell

[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit => Status.....: Exhausted
Hash.Name.....: MD5
Hash.Target.....: generated-md5.csv
Time.Started.....: Sat Dec 12 07:01:00 2020, (1 min, 13 secs)
Time.Estimated....: Sat Dec 12 07:02:13 2020, (0 secs)
Guess.Base.....: File (common100Kpass.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 18641.7 kH/s (0.58ms) @ Accel:64 Loops:1 Thr:1024 Vec:1
Speed.#3.....: 47 H/s (0.30ms) @ Accel:1024 Loops:1 Thr:8 Vec:4
Speed.#*.....: 18641.8 kH/s
Recovered.....: 12530/177130 (7.07%) Digests
Remaining.....: 164600 (92.93%) Digests
Recovered/Time....: CUR:8538,N/A,N/A AVG:6960,417640,10023375 (Min,Hour,Day)
Progress.....: 99995/99995 (100.00%)
Rejected.....: 1/99995 (0.00%)
Restore.Point.....: 3469/99995 (3.47%)
Restore.Sub.#1....: Salt:0 Amplifier:0-1 Iteration:0-1
Restore.Sub.#3....: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: claudia1 -> crossroad
Candidates.#3....: 123456 -> mykids
Hardware.Mon.#1...: Temp: 54c Util: 0% Core:1354MHz Mem:3504MHz Bus:16
Hardware.Mon.#3...: N/A

Started: Sat Dec 12 07:00:38 2020
Stopped: Sat Dec 12 07:02:14 2020
PS E:\hashcat-6.1.1>
```

```
Windows PowerShell

PS E:\hashcat-6.1.1> .\hashcat.exe --optimized-kernel-enable -w 4 --force -a 0 -m 0 generated-md5.csv common100Kpass.txt -o output-md5.csv
hashcat (v6.1.1) starting...

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.
nvidiaDeviceGetFanSpeed(): Not Supported

CUDA API (CUDA 11.1)
=====
* Device #1: GeForce GTX 1050, 3383/4096 MB, 5MCU

OpenCL API (OpenCL 1.2 CUDA 11.1.114) - Platform #1 [NVIDIA Corporation]
=====
* Device #2: GeForce GTX 1050, skipped

OpenCL API (OpenCL 2.1 ) - Platform #2 [Intel(R) Corporation]
=====
* Device #3: Intel(R) HD Graphics 630, 3158/3222 MB (1611 MB allocatable), 23MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 31

Hashes: 200000 digests; 177130 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

Було відновлено 12530 з 177130 паролів, тобто біля 7 відсотків паролів.

Тепер спробуємо ще раз тільки цього разу методом брутфорс із стандартними налаштуваннями.

***hashcat --optimized-kernel-enable -w 3 -a 3 -m 0 generated-md5.csv -o output-md5-brute.csv***

```
Windows PowerShell
Candidates.#1....: 3a4y89b -> Cn7bze9
Hardware.Mon.#1..: Temp: 68c Util: 88% Core:1645MHz Mem:3504MHz Bus:16

Approaching final keyspace - workload adjusted.

Session.....: hashcat
Status.....: Exhausted
Hash.Name.....: MD5
Hash.Target.....: generated-md5.csv
Time.Started.....: Sat Dec 12 07:17:22 2020 (1 min, 42 secs)
Time.Estimated....: Sat Dec 12 07:19:04 2020 (0 secs)
Guess.Mask.....: ?1?2?2?2?2?2?2 [7]
Guess.Charset....: -1 ?1?d?u, -2 ?1?d, -3 ?1?d*!$@_, -4 Undefined
Guess.Queue.....: 7/15 (46.67%)
Speed.#1.....: 1323.2 MH/s (6.60ms) @ Accel:64 Loops:512 Thr:1024 Vec:8
Recovered.....: 76918/177130 (43.42%) Digests
Remaining.....: 100212 (56.58%) Digests
Recovered/Time...: CUR:7272,N/A,N/A AVG:9683,580995,13943887 (Min,Hour,Day)
Progress.....: 134960504832/134960504832 (100.00%)
Rejected.....: 0/134960504832 (0.00%)
Restore.Point....: 1679616/1679616 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:79872-80352 Iteration:0-512
Candidates.#1....: g8qtlyq -> Xqxxqg
Hardware.Mon.#1..: Temp: 69c Util: 81% Core:1645MHz Mem:3504MHz Bus:16

[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit =>
```

Цього разу вже було відновлено 43 відсотки (76918/177130) паролів всього за 2 хвилини.

## SHA1 + salt

Перейдемо до паролів, що захешовані за допомогою SHA1 + salt

Для цього використаємо скрипт adapter.py, який просто міняє у файлі місцями сіль та хеш.

Тепер пробуємо відновити паролі за допомогою найпопулярніших паролів.

***hashcat -w 3 -d 1 -a 0 -m 110 generated-sha1-reverse.csv common100Kpass.txt -o output-sha1.csv***

```
Выбрать Windows PowerShell
Restore.Point....: 0/99995 (0.00%)
Restore.Sub.#1...: Salt:199999 Amplifier:0-1 Iteration:0-1
Restore.Sub.#3...: Salt:154808 Amplifier:0-1 Iteration:0-1
Candidates.#1....: chucha -> crossroad
Candidates.#3....: 123456 -> cheeks1
Hardware.Mon.#1..: Core:1354MHz Mem:3504MHz Bus:16
Hardware.Mon.#3..: N/A

[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit =>

Session.....: hashcat
Status.....: Running
Hash.Name.....: sha1($pass.$salt)
Hash.Target.....: generated-sha1-reverse.csv
Time.Started.....: Sat Dec 12 08:13:39 2020, (38 mins, 17 secs)
Time.Estimated....: Sat Dec 12 08:52:30 2020, (34 secs)
Guess.Base.....: File (common100Kpass.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 42444.1 kH/s (1.06ms) @ Accel:1024 Loops:1 Thr:64 Vec:1
Speed.#3.....: 2831.6 kH/s (11.04ms) @ Accel:1024 Loops:1 Thr:8 Vec:1
Speed.#*.....: 45275.7 kH/s
Recovered.....: 20180/200000 (10.09%) Digests, 20180/200000 (10.09%) Salts
Remaining.....: 179820 (89.91%) Digests, 179820 (89.91%) Salts
Recovered/Time...: CUR:331,N/A,N/A AVG:627,37663,903917 (Min,Hour,Day)
Progress.....: 18351548611/19999000000 (91.76%)
Rejected.....: 0/18351548611 (0.00%)
Restore.Point....: 0/99995 (0.00%)
Restore.Sub.#1...: Salt:199999 Amplifier:0-1 Iteration:0-1
Restore.Sub.#3...: Salt:154873 Amplifier:0-1 Iteration:0-1
Candidates.#1....: chucha -> crossroad
Candidates.#3....: 123456 -> cheeks1
Hardware.Mon.#1..: Core:1354MHz Mem:3504MHz Bus:16
Hardware.Mon.#3..: N/A
```

Після майже 40 хв. роботи було розшифровано 10 відсотків паролів. Такий достатньо великий час для цього випадку через те що по середині обчислень дискретна відеокарта перестала відповідати і решту обчислень проводилися лише на інтегрований.

Запускаємо ще раз і маємо результат: після 6 хвилин роботи було відновлено 11% паролів.  
Методом dictionary.

Пробуємо брутфорс

Використовуючи брутфорс скористаємося знанням про те з яких символів складається пароль та в якому проміжку буде його довжина.

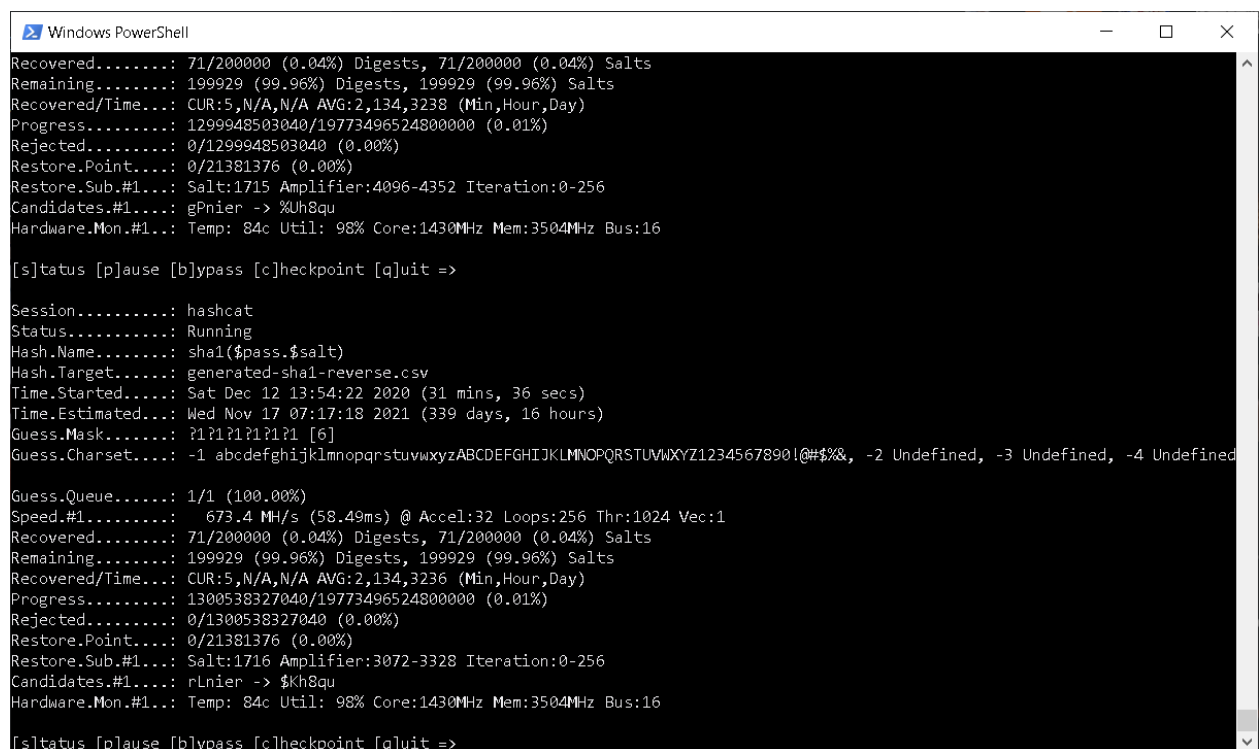
Набір символів:

**abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#\$%&**

Спробуємо знайти паролі довжиною в 5 символів.

**hashcat.exe -w 3 -d 1 -a 3 -m 110 -o output-sha1-brute.csv -1**

**"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#\$%&" generated-sha1-reverse.csv ?1?1?1?1?1**



Можемо спостерігати що після 30 хвилин було розшифровано всього 71 пароль (і це при швидкості 670 МН/с). Для повного перебору знадобиться біля 350 днів.

## bcrypt

Пробуємо відновити паролі методом dictionary. З кількістю раундів для генерації солі, що дорівнює 4.

**hashcat.exe -w 3 -d 1 -a 0 -m 3200 generated-bcrypt.csv common100Kpass.txt -o output-bcrypt.csv**

```
Windows PowerShell
Recovered/Time....: CUR:26,N/A,N/A AVG:28,1731,41562 (Min,Hour,Day)
Progress.....: 13562880/19999000000 (0.07%)
Rejected.....: 0/13562880 (0.00%)
Restore.Point....: 0/99995 (0.00%)
Restore.Sub.#1...: Salt:14129 Amplifier:0-1 Iteration:12-16
Candidates.#1....: 123456 -> sarah
Hardware.Mon.#1...: Temp: 63c Util: 99% Core:1645MHz Mem:3504MHz Bus:16

[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit =>

Paused

[s]tatus [r]esume [b]ypass [c]heckpoint [q]uit =>

Session.....: hashcat
Status.....: Paused
Hash.Name.....: bcrypt $2*$, Blowfish (Unix)
Hash.Target.....: generated-bcrypt.csv
Time.Started....: Sat Dec 12 15:25:03 2020 (35 mins, 25 secs)
Time.Estimated...: Wed Jan 06 09:27:42 2021 (24 days, 17 hours)
Guess.Base.....: File (common100kpass.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 9309 H/s (24.09ms) @ Accel:16 Loops:4 Thr:12 Vec:1
Recovered.....: 934/200000 (0.47%) Digests, 934/200000 (0.47%) Salts
Remaining.....: 199066 (99.53%) Digests, 199066 (99.53%) Salts
Recovered/Time....: CUR:0,N/A,N/A AVG:28,1726,41428 (Min,Hour,Day)
Progress.....: 18089280/19999000000 (0.09%)
Rejected.....: 0/18089280 (0.00%)
Restore.Point....: 0/99995 (0.00%)
Restore.Sub.#1...: Salt:18842 Amplifier:0-1 Iteration:12-16
Candidates.#1....: 123456 -> sarah
Hardware.Mon.#1...: Temp: 59c Util: 0% Core:1354MHz Mem:3504MHz Bus:16

[s]tatus [r]esume [b]ypass [c]heckpoint [q]uit =>
```

За 35 хвилин відновилося всього 934 паролі (0.47%)

## Підсумок

Як бачимо MD5 проявив себе найгірше. За дуже короткий час dictionary відновлює достатньо велику кількість паролів(7%), а перебір відновлює взагалі 43% всього за 2 хвилини.

Sha1 + salt дає значно кращі результати, проте теж дуже вразливий до методу dictionary. Але майже не піддається brute-force атакам.

Bcrypt показує найкращий результат, він достатньо стійкий як до dictionary так і до brute-force.

Найлегше відновлювати паролі що вже є в словниках з найбільш використовуваними, важко зламати ті, що мають велику довжину.

Тому на паролі краще вводити наступні обмеження:

- Велика кількість символів(мінімум 8)
- Якнайбільше різних типів символів(літери як верхнього так і нижнього регістру, цифри та різні спецсимволи)
- Не допускати паролі які входять до найбільш популярних

Для хешування не можна використовувати методи типу MD5. Краще використовувати стійки алгоритми типу bcrypt, scrypt та argon2.