Lab 4.2 Report

Хеші для виконання другої частини було взято з репозиторію: https://github.com/Yurwar/human-password-generator/tree/master/src/main/resources.

Для взлому хешів використовувалась програма hashcat та типи атак dictionary (використовуючи список найбільш популярних паролів) та brute-force (використовуючи повний перебір, з маскою яка включає від 4 до 16 маленький, великих букв чи цифр).

MD5.

Для атакти dictionary використовуємо команду hashcat -m 0 -a 0 -o cracked-passwords2.txt md5.txt common-passwords-long.txt

```
Session.....: hashcat
Status.....: Exhausted
Hash.Type.....: MD5
Hash.Target....: md5.txt
Time.Started...: Fri Dec 11 19:59:39 2020 (2 secs)
Time.Estimated...: Fri Dec 11 19:59:41 2020 (0 secs)
Guess.Base....: File (common-passwords-long.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.Dev.#1...: 476.6 kH/s (2.55ms)
Recovered.....: 104035/590480 (17.62%) Digests, 0/1 (0.00%) Salts
Recovered/Time..: CUR:N/A,N/A,N/A AVG:2947977,176878632,-2147483648 (Min,Hour,Day)
Progress.....: 999998/99998 (100.00%)
Rejected.....: 0/999998 (0.00%)
Restore.Point...: 999998/999998 (100.00%)
Candidates.#1...: vrs38761995 -> vjht008
HWMon.Dev.#1...: Temp: 51c Util: 28% Core:1354MHz Mem:3504MHz Bus:8

Started: Fri Dec 11 19:59:36 2020
Stopped: Fri Dec 11 19:59:43 2020
```

В результаті атаки ми змогли дізнатись 17,62% паролів. Ймовірно саме така кількість була взята з списку найбільш вживаних паролів при генерації. Як бачимо, ця атака тривала всього 2 секунди, що свідчить про те, що хеші md5 можна дуже швидко перебирати.

Для атаки brute-force використовуємо наступну команду

hashcat -m 0 -a 3 -o cracked-passwords-md-5-brute.txt md5.txt -1?l?u?d ?1?1?1?1?1?1?1?1?1?1?1?1?1?1?1 --increment --increment-min 4 --increment-max 16

Після кількох годин роботи, алгоритм перебрав всі варіанти паролів розміром від 4 до 7 та відгадав правильно 25.54% паролів. Оскільки для того, щоб перебрати паролі розміром 8 потрібно було 2 дні, ми вирішили зупинити атаку. Можна зробити висновок, що маючи більші обчислювальні потужності, а не один ноутбук, можна досягти дуже великого відсотку зламаних паролів за відносно невеликий час.

SHA1 + salt.

Для атакти dictionary використовуємо команду hashcat -m 110 -a 0 -o cracked-passwords4.txt salted-sha1.csv common-passwords-long.txt

```
Session......: hashcat
 Status..... Running
Hash.Type.....: sha1($pass.$salt)
Hash.Target.....: salted-sha1.csv
Hash.Target.....: salted-sha1.csv
Time.Started....: Fri Dec 11 22:27:42 2020 (1 hour, 37 mins)
Time.Estimated...: Sat Dec 12 01:45:09 2020 (1 hour, 39 mins)
Guess.Base.....: File (common-passwords-long.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.Dev.#1....: 56517.4 kH/s (2.82ms)
Recovered.....: 405960/1000000 (40.60%) Digests, 405960/1000000 (40.60%) Salts
Recovered/Time...: CUR:36,103882,N/A AVG:4151,249106,5978553 (Min,Hour,Day)
Progress......: 429310443520/999998000000 (42.93%)
Rejected.....: 0/429310443520 (0.00%)
Rejected.....: 0/429310443520 (0.00%)
Restore.Point...: 327680/999998 (32.77%)
  Candidates.#1....: Booga31 -> massa123
 HWMon.Dev.#1....: Temp: 72c Util: 98% Core:1670MHz Mem:3504MHz Bus:8
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => s
 Session...... hashcat
 Status..... Running
 Hash.Type.....: sha1($pass.$salt)
 Hash.Target.....: salted-sha1.csv
 Time.Started....: Fri Dec 11 22:27:42 2020 (2 hours, 1 min)
Time.Estimated...: Sat Dec 12 01:48:10 2020 (1 hour, 18 mins)
Time.Estimated...: Sat Dec 12 01:48:10 2020 (1 hour, 18 mins)
Guess.Base......: File (common-passwords-long.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.Dev.#1....: 54788.1 kH/s (2.84ms)
Recovered.....: 407002/1000000 (40.70%) Digests, 407002/1000000 (40.70%) Salts
Recovered/Time...: CUR:41,3396,N/A AVG:3339,200392,4809409 (Min,Hour,Day)
Progress.....: 564363755520/999998000000 (56.44%)
Rejected.....: 0/564363755520 (0.00%)
Restore.Point...: 491520/999998 (49.15%)
Candidates.#1...: massa021 -> 922846418
HWMon.Dev.#1....: Temp: 72c Util: 95% Core:1670MHz Mem:3504MHz Bus:8
  HWMon.Dev.#1....: Temp: 72c Util: 95% Core:1670MHz Mem:3504MHz Bus:8
 [s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => q
```

Після двох годин атаки вдалось зламати 40.70% паролів. Протягом довгого часу відсоток зламаних паролів не змінювався і ми припустили, що всі паролі, які були створені з списку найбільш вживаних паролів вже відгадано і вони містились серед перших 50% всіх паролів, тому атаку було зупинено.

Для атаки brute-force використовуємо наступну команду

hashcat -m 110 -a 3 -o cracked-passwords-sha1-brute.txt salted-sha1.csv -1?l?u?d ?1?1?1?1?1?1?1?1?1?1?1?1?1?1?1?1 --increment --increment-min 4 --increment-max 16

```
Session...... hashcat
Status..... Quit
Hash.Type.....: sha1($pass.$salt)
Hash.Target.....: salted-sha1.csv
Time.Started.....: Sat Dec 12 00:34:40 2020 (1 hour, 29 mins)
Time.Estimated...: Sat Dec 12 08:46:03 2020 (6 hours, 42 mins)
Guess.Mask.....: ?1?1?1?1 [4]
Guess.Charset....: -1 ?l?u?d, -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue.....: 1/13 (7.69%)
Speed.Dev.#1....: 497.2 MH/s (9.83ms)
Recovered.....: 11659/1000000 (1.17%) Digests, 11659/1000000 (1.17%) Salts
Recovered/Time...: CUR:138,7848,N/A AVG:130,7836,188080 (Min,Hour,Day)
Progress.....: 2726961971200/14776336000000 (18.45%)
Rejected......: 0/2726961971200 (0.00%)
Restore.Point....: 0/238328 (0.00%)
Candidates.#1....: sari -> 7gN6
HWMon.Dev.#1....: Temp: 89c Util: 96% Core:1645MHz Mem:3504MHz Bus:8
Started: Sat Dec 12 00:34:32 2020
Stopped: Sat Dec 12 02:03:57 2020
mhnatyshyn@mhnatyshyn:~/Завантаження$
```

Після півтори години атаки відсоток зламаних паролів був лише 1.17, проте було перебрано тільки паролі довжиною 4 символи. Ми зупинили атаку, так як вона зайняла б дуже багато часу. Можемо зробити висновок, що цю схему хешування складніше брутфорсити, для цього потрібно більше часу і більш потужні обчислювальні ресурси, проте це все ще можливо зробити за реальний час.

Bcrypt

Для атакти dictionary використовуємо команду

hashcat -m 3200 -a 0 -o cracked-passwords4.txt bcrypt.csv common-passwords-long.txt

```
Session......: hashcat
Status..... Exhausted
Hash.Type.....: bcrypt $2*$, Blowfish (Unix)
Hash.Target.....: bcrypt.csv
Time.Started.....: Sat Dec 12 08:25:34 2020 (30 mins, 14 secs)
Time.Estimated...: Sun Nov 30 10:25:19 2031 (10 years, 353 days)
Guess.Base.....: File (common-passwords-long.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.Dev.#1....: 275 H/s´(3.09ms)
Recovered.....: 4879/100000 (4.88%) Digests, 4879/100000 (4.88%) Salts
Recovered/Time...: CUR:108,N/A,N/A AVG:161,9680,232328 (Min,Hour,Day)
Progress.....: 1220160/99999800000 (0.00%)
Rejected...... 0/1220160 (0.00%)
Restore.Point....: 0/999998 (0.00%)
Candidates.#1....: ->
HWMon.Dev.#1....: Temp: 54c Util: 0% Core:1354MHz Mem:3504MHz Bus:8
Started: Sat Dec 12 08:25:30 2020
Stopped: Sat Dec 12 08:55:50 2020
mhnatyshyn@mhnatyshyn:~/Завантаження$
```

Після півгодини атаки відсоток зламаних паролів був 4.88. Атаку було зупинено, так як по розрахунках програми на неї потрібно було б 10 років :) Можемо зробити висновок, що навіть таку атаку як dictionary дуже складно провести на алгоритм bcrypt, тому його можна вважати доволі стійким до такої атаки.

Атака brute-force не проводилась, так як вона б зайняла кілька десятків років на наявних обчислювальних можливостях. Тому можемо зробити висновок що атакувати даний алгоритм за допомогою повного перебору майже нереально, навіть якщо збільшити обчислювальні можливості.

Висновки

Серед розглянутих алгоритмів хешування найкраще себе показав bcrypt. На нього дуже складно провести атаку dictionary, а атаку з повним перебором майже нереально за розумний час. Два інші алгоритми показали гірший результат, але в порівнянні алгоритм sha1+salt показав себе кращим, ніж md5.

Найпростіше було зламати паролі з списку часто вживаних паролів, а також такі, які мають невелику довжину, та невеликий набір символів.

При створенні паролю варто дотримуватися таких рекомендацій:

- Пароль не повинен бути коротким(довжина має бути більше 8 символів)
- Не використовувати реальні слова
- Використовувати спецсимволи, великі та малі літери, цифри для того, щоб збільшити кількість варіантів при bruteforce
- Не використовувати часто вживані паролі

При хешуванні паролю варто використовувати стійкі до злому алгоритми хешування(bcrypt, argon2)