

Crack The Hash

Kumpel7

November 2023

I used ubuntu machine and openVPN. All of the solutions (with questions) are provided at the end of the document.

1 Level 1

All of the hashes names I guessed because of hints, experience + knowledge of how many characters one hash is built of. For example Sha-256 has 256 bits, so 64 letters. MD5 is 128 bits, so 32 letters.

Answer the questions below

48bb6e862e54f2a795ffc4e541caed4d

Answer format: ****

CBFDAC6008F9CAB4083784CBD1874F76618D2A97

Answer format: *****

1C8BFE8F801D79745C4631D09FFF36C82AA37FC4CCE4FC946683D7B336B63032

Answer format: *****

\$2y\$12\$Dwt1BZj6pcyc3Dy1FWZ5ieeUznr71EeNkJkUlypTsgbX1H68wsRom

Answer format: ****

279412f945939ba78ce0758d3fd83daa

Answer format: *****

Rysunek 1: The hashes we need to crack

The first one is an MD5 hash. I insert it into john and immediately get the answer. I am using the rockyou.txt file as a dictionary. So my command is (as a root):

```
1 john hash1.txt --format=Raw-MD5 --wordlist=../wordlists/rockyou/rockyou.txt
```

```
(root@kumpel)-[/home/dan/Prezentacja]
# john hash1.txt --format=Raw-MD5 --wordlist=../wordlists/rockyou/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
easy (?)
1g 0:00:00:00 DONE (2023-11-18 18:22) 16.67g/s 2873Kp/s 2873Kc/s 2873Kc/s florian1..dynamic
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

The second one I didn't even have to guess, as john found out that this is Sha-1.

```

└─# john hash2.txt --wordlist=./wordlists/rockyou/rockyou.txt
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "Raw-SHA1-AxCrypt"
Use the "--format=Raw-SHA1-AxCrypt" option to force loading these as that type instead
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "Raw-SHA1-LinkedIn"
Use the "--format=Raw-SHA1-LinkedIn" option to force loading these as that type instead
Warning: detected hash type "Raw-SHA1", but the string is also recognized as "ripemd-160"
Use the "--format=ripemd-160" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 256/256 AVX2 8x])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2023-11-18 18:29) 25.00g/s 34600p/s 34600c/s 34600C/s jesse..password123
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.

```

The 3rd one is Sha-256.

```

└─# john hash3.txt --format=raw-sha256 --wordlist=./wordlists/rockyou/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
letmein          (?)
1g 0:00:00:00 DONE (2023-11-18 18:34) 20.00g/s 2621Kp/s 2621Kc/s 2621KC/s 123456..koryna
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.

```

hashes that start with \$2y\$ are called bcrypt. Here we knew that password will have 4 letters, so i used the additional lenght option.

```

└─# john hash4.txt --format=bcrypt --wordlist=./wordlists/rockyou/rockyou.txt --length=4
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 4096 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
bleh             (?)
1g 0:00:00:18 DONE (2023-11-18 18:46) 0g/s 34.69p/s 34.69c/s 34.69C/s spam..5050
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

```

The last one from this level is md4 hash. This comes with a little suprise, as it is not from rockyou.txt. For this one i used the john preinstalled wordlist.

```

└─# john hash5.txt --format=raw-md4
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD4 [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=8
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/home/dan/src/john/run/password.lst
Enabling duplicate candidate password suppressor
Eternity22       (?)
1g 0:00:00:01 DONE 2/3 (2023-11-18 18:52) 0g/s 2069Kp/s 2069Kc/s 2069KC/s Iamanurse..Kaylin01
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

```

2 Level 2

The first one is easy, because it is sha-256.

```

└─# john hash6.txt --format=raw-sha256 --wordlist=./wordlists/rockyou/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
paule            (?)
1g 0:00:00:00 DONE (2023-11-18 18:57) 16.67g/s 2184Kp/s 2184Kc/s 2184KC/s 123456..koryna
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.

```

Answer the questions below

Hash: F09EDCB1FCEFC6DFB23DC3505A882655FF77375ED8AA2D1C13F640FCCC2D0C85

Answer format: *****

Submit

Hash: 1DFECA0C002AE40B8619ECF94819CC1B

Answer format: *****

Submit

Hint

Hash: \$6\$aReallyHardSalt\$6WKUTzqz.UQQmrm0p/T7MPpMbGNzXPMAXi4bJmI9be.cfi3/qxIf.hsGpS41BqMhSrHVXgMpdjS6xeKZAs02.

Salt: aReallyHardSalt

Answer format: *****

Submit

Hash: e5d8870e5bdd26602cab8dbe07a942c8669e56d6

Salt: tryhackme

Answer format: *****

Submit

Hint

Rysunek 2: New Exercises

The second one is hard to guess, as I encounter it for the first time in the rooms, it is NTLM from microsoft.

```
(root@kumpel) - [ /home/dan/Prezentacja ]
# john hash7.txt --format=nt --wordlist=../wordlists/rockyou/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
n63umy8lkf4i (?)
1g 0:00:00:00 DONE (2023-11-18 19:01) 2.273g/s 11907Kp/s 11907Kc/s 11907Kc/s n65452..n601325
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

The next 2 hashes are with salt. The 3rd one should be sha-512 as it has \$6\$ before the hash. This password took a while to crack, so i used length=6, because i see I will have 6-letter password :).

```
(root@kumpel) - [ /home/dan/Prezentacja ]
# john --wordlist=../wordlists/rockyou/rockyou.txt --format=sha512crypt --length=6 hash8.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
0g 0:00:00:14 0.45% (ETA: 20:31:09) 0g/s 2116p/s 2116c/s 2116C/s 021792..030703
0g 0:00:00:19 0.66% (ETA: 20:27:26) 0g/s 2194p/s 2194c/s 2194C/s 042281..cendol
0g 0:00:00:25 0.87% (ETA: 20:27:02) 0g/s 2162p/s 2162c/s 2162C/s xiexie..jack94
waka99 (?)
1g 0:00:05:25 DONE (2023-11-18 19:44) 0g/s 1949p/s 1949c/s 1949C/s waniel..wajwaj
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

And now the last hash, for which john has the problem... <https://github.com/openwall/john/issues/4259> ...

So we need to crack it using something different. I found a program called hashcat, so let's use that, and because I have some driver issues with hashcat, I will use it on the attackbox provided on tryhackme.

```
1 hashcat -m 160 test.txt Tools/wordlists/rockyou.txt --force
```

-m 160 means what hash am i cracking, -force ignores some stupid attackbox errors about drivers. This gives a lot of output, but the interesting part is at the end.

```

root@kali:~/rockyou# ./rockyou1.txt
root@kali:~/rockyou# ./rockyou1.txt: @ hashcat -m 100 test.txt tools/wordlists/rockyou.txt --force
hashcat (v1.1.66-g8af15086) starting...

You have enabled --force to bypass dangerous warnings and errors!
It is not from a trusted platform and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

OptimCL ZPP (OptimCL 1.2 LHM2) - Platform #1 [Intel(R) Corporation]
=====
* Device #1: AMD EPYC 7772, 3842/3806 MB (576 MB allocated), 80C

OptimCL ZPP (OptimCL 1.2 pocl 1.1 Nemuclworks, LLVM 6.0.0, SPIR, SLEEF, DISTNO, PGOCL_DEBUG) - Platform #2 [The pocl project]
=====
* Device #2: gallium AMD EPYC 7772, 8496MB

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests; 1 unique salts
Bitmaps: 16 bits; 61336 entries; 8a0000ffff mask; 262144 bytes; 5/13 rotates
Rules: 1

Liable optimizers applied:
* No-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels means creating longer passwords but for the price of drastically reduced performance.
If you want to switch to optimized backend kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache building Tools/wordlists/rockyou.txt: 33533435 byDictionary cache building Tools/wordlists/rockyou.txt: 67186873 byDictionary cache building Tools/wordlists/rockyou.txt: 108668389 byDictionary cache building Tools/wordlists/rockyou.txt: 136213745 byDictionary cache built:
* Filename: Tools/wordlists/rockyou.txt
* Passwords: 14364091
* Bytes: 130921497
* Tempfile: 14364084
* Runtime: 12 secs

#58887b5bd256d2cab8d8a07d942f86d9a5d6:tryhackme481616481616

Session.....: hashcat
Status.....: Finished
Hash Name.....: 809C5D81 (key = Salt)
Hash Target.....: #58887b5bd256d2cab8d8a07d942f86d9a5d6:tryhackme
Time Started.....: Sat Nov 18 21:12:16 2023 (11 secs)
Time Ended.....: Sat Nov 18 21:12:27 2023 (11 secs)
Guess Rate.....: File [Tools/wordlists/rockyou.txt]
Guesses/Sec.....: 125.120 80%
Speed #1.....: 1112.7 MB/s (1.48ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recover.....: 12 1280 0MB Digits
Progress.....: 12514624/14344384 (85.85%)
Rejected.....: 812216424 (0.00%)
Rejection Point.....: 12512176/14344384 (85.84%)
Rejection Sub #1.....: Salt:8 Impl:Iter:0-1 Iteration:0-1
Candidates #1.....: 48162440 -> 481616481616

Started: Sat Nov 18 21:11:40 2023
Stopped: Sat Nov 18 21:12:28 2023

```

If you don't see it well, the password is after the second ":" sign : 481616481616. That's the end.