Cracked Passwords Report

I was able to crack only 13 passwords from the given 19 hashes. Firstly I have analyzed which hashing algorithm is used to hash these codes and then I have used Hashcat to decode the maximum number of passwords possible. I have used both dictionaries like rockyou and john the ripper to decode and bruteforce attacking method. But both are resulted in decoding of 13 only. (Brute force gave an estimated time of 140 days to complete).

BruteForce:   
C:\Users\kalya\OneDrive\Desktop\hashcat\hashcat-6.1.0>hashcat -m 0 -a 3 --username passwords.txt ?a?a?a?a?a?a?a?a

hashcat (v6.1.0) starting...

\* Device #1: Unstable OpenCL driver detected!

This OpenCL driver has been marked as likely to fail kernel compilation or to produce false negatives.

You can use --force to override this, but do not report related errors.

OpenCL API (OpenCL 3.0 ) - Platform #1 [Intel(R) Corporation]

=============================================================

\* Device #1: Intel(R) Iris(R) Xe Graphics, skipped

OpenCL API (OpenCL 3.0 D3D12 Implementation) - Platform #2 [Microsoft]

======================================================================

\* Device #2: Intel(R) Iris(R) Xe Graphics, 3843/3907 MB (976 MB allocatable), 1MCU

\* Device #3: Microsoft Basic Render Driver, skipped

Minimum password length supported by kernel: 0

Maximum password length supported by kernel: 256

Hashes: 19 digests; 19 unique digests, 1 unique salts

Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Applicable optimizers applied:

\* Zero-Byte

\* Early-Skip

\* Not-Salted

\* Not-Iterated

\* Single-Salt

\* Brute-Force

\* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.

Using pure kernels enables cracking 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.

INFO: Removed 13 hashes found in potfile.

Host memory required for this attack: 81 MB

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

Session..........: hashcat

Status...........: Running

Hash.Name........: MD5

Hash.Target......: passwords.txt

Time.Started.....: Sat Jun 22 05:56:20 2024 (29 secs)

Time.Estimated...: Wed Nov 06 06:37:15 2024 (137 days, 1 hour)

Guess.Mask.......: ?a?a?a?a?a?a?a?a [8]

Guess.Queue......: 1/1 (100.00%)

Speed.#2.........: 559.8 MH/s (0.09ms) @ Accel:64 Loops:1024 Thr:1024 Vec:1

Recovered........: 13/19 (68.42%) Digests

Progress.........: 15904800768/6634204312890625 (0.00%)

Rejected.........: 0/15904800768 (0.00%)

Restore.Point....: 0/7737809375 (0.00%)

Restore.Sub.#2...: Salt:0 Amplifier:242688-243712 Iteration:0-1024

Candidates.#2....: Y'uerane -> AN0)eman

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

Session..........: hashcat

Status...........: Running

Hash.Name........: MD5

Hash.Target......: passwords.txt

Time.Started.....: Sat Jun 22 05:56:20 2024 (46 secs)

Time.Estimated...: Sun Nov 03 06:57:42 2024 (134 days, 2 hours)

Guess.Mask.......: ?a?a?a?a?a?a?a?a [8]

Guess.Queue......: 1/1 (100.00%)

Speed.#2.........: 572.7 MH/s (0.09ms) @ Accel:64 Loops:1024 Thr:1024 Vec:1

Recovered........: 13/19 (68.42%) Digests

Progress.........: 26239565824/6634204312890625 (0.00%)

Rejected.........: 0/26239565824 (0.00%)

Restore.Point....: 0/7737809375 (0.00%)

Restore.Sub.#2...: Salt:0 Amplifier:399360-400384 Iteration:0-1024

Candidates.#2....: +8Cerane -> EPm)eman

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

Session..........: hashcat

Status...........: Running

Hash.Name........: MD5

Hash.Target......: passwords.txt

Time.Started.....: Sat Jun 22 05:56:20 2024 (1 min, 33 secs)

Time.Estimated...: Wed Oct 23 05:31:39 2024 (122 days, 23 hours)

Guess.Mask.......: ?a?a?a?a?a?a?a?a [8]

Guess.Queue......: 1/1 (100.00%)

Speed.#2.........: 624.4 MH/s (0.07ms) @ Accel:64 Loops:1024 Thr:1024 Vec:1

Recovered........: 13/19 (68.42%) Digests

Progress.........: 57866649600/6634204312890625 (0.00%)

Rejected.........: 0/57866649600 (0.00%)

Restore.Point....: 65536/7737809375 (0.00%)

Restore.Sub.#2...: Salt:0 Amplifier:25600-26624 Iteration:0-1024

Candidates.#2....: x$h(^han -> 9)s&%999

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

Session..........: hashcat

Status...........: Running

Hash.Name........: MD5

Hash.Target......: passwords.txt

Time.Started.....: Sat Jun 22 05:56:20 2024 (2 mins, 15 secs)

Time.Estimated...: Sun Oct 20 19:55:30 2024 (120 days, 13 hours)

Guess.Mask.......: ?a?a?a?a?a?a?a?a [8]

Guess.Queue......: 1/1 (100.00%)

Speed.#2.........: 636.8 MH/s (0.09ms) @ Accel:64 Loops:1024 Thr:1024 Vec:1

Recovered........: 13/19 (68.42%) Digests

Progress.........: 85985263616/6634204312890625 (0.00%)

Rejected.........: 0/85985263616 (0.00%)

Restore.Point....: 65536/7737809375 (0.00%)

Restore.Sub.#2...: Salt:0 Amplifier:454656-455680 Iteration:0-1024

Candidates.#2....: ?oA(^han -> #^W&%999

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

Session..........: hashcat

Status...........: Running

Hash.Name........: MD5

Hash.Target......: passwords.txt

Time.Started.....: Sat Jun 22 05:56:20 2024 (5 mins, 5 secs)

Time.Estimated...: Sun Oct 27 04:26:01 2024 (126 days, 22 hours)

Guess.Mask.......: ?a?a?a?a?a?a?a?a [8]

Guess.Queue......: 1/1 (100.00%)

Speed.#2.........: 604.9 MH/s (0.10ms) @ Accel:64 Loops:1024 Thr:1024 Vec:1

Recovered........: 13/19 (68.42%) Digests

Progress.........: 184337367040/6634204312890625 (0.00%)

Rejected.........: 0/184337367040 (0.00%)

Restore.Point....: 196608/7737809375 (0.00%)

Restore.Sub.#2...: Salt:0 Amplifier:240640-241664 Iteration:0-1024

Candidates.#2....: aFoV[=12 -> =VJNDPON

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

Session..........: hashcat

Status...........: Running

Hash.Name........: MD5

Hash.Target......: passwords.txt

Time.Started.....: Sat Jun 22 05:56:20 2024 (8 mins, 23 secs)

Time.Estimated...: Mon Oct 21 18:05:56 2024 (121 days, 12 hours)

Guess.Mask.......: ?a?a?a?a?a?a?a?a [8]

Guess.Queue......: 1/1 (100.00%)

Speed.#2.........: 631.9 MH/s (0.10ms) @ Accel:64 Loops:1024 Thr:1024 Vec:1

Recovered........: 13/19 (68.42%) Digests

Progress.........: 311479173120/6634204312890625 (0.00%)

Rejected.........: 0/311479173120 (0.00%)

Restore.Point....: 327680/7737809375 (0.00%)

Restore.Sub.#2...: Salt:0 Amplifier:465920-466944 Iteration:0-1024

Candidates.#2....: qZhpw... -> nQl1Esan

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

Session..........: hashcat

Status...........: Running

Hash.Name........: MD5

Hash.Target......: passwords.txt

Time.Started.....: Sat Jun 22 05:56:20 2024 (12 mins, 24 secs)

Time.Estimated...: Thu Nov 14 00:57:40 2024 (144 days, 19 hours)

Guess.Mask.......: ?a?a?a?a?a?a?a?a [8]

Guess.Queue......: 1/1 (100.00%)

Speed.#2.........: 530.2 MH/s (0.07ms) @ Accel:64 Loops:1024 Thr:1024 Vec:1

Recovered........: 13/19 (68.42%) Digests

Progress.........: 433453596672/6634204312890625 (0.01%)

Rejected.........: 0/433453596672 (0.00%)

Restore.Point....: 458752/7737809375 (0.01%)

Restore.Sub.#2...: Salt:0 Amplifier:612352-613376 Iteration:0-1024

Candidates.#2....: ;Qz|c$$$ -> \* c'dDON

C:\Users\kalya\OneDrive\Desktop\hashcat\hashcat-6.1.0>hashcat -m 0 --username passwords.txt --show

ortspoon:d8578edf8458ce06fbc5bb76a58c5ca4:qwerty

bookma:25d55ad283aa400af464c76d713c07ad:12345678

heroanhart:7c6a180b36896a0a8c02787eeafb0e4c:password1

simmson56:96e79218965eb72c92a549dd5a330112:111111

interestec:25f9e794323b453885f5181f1b624d0b:123456789

eatingcake1994:fcea920f7412b5da7be0cf42b8c93759:1234567

edi\_tesla89:6c569aabbf7775ef8fc570e228c16b98:password!

reallychel:5f4dcc3b5aa765d61d8327deb882cf99:password

experthead:e10adc3949ba59abbe56e057f20f883e:123456

popularkiya7:e99a18c428cb38d5f260853678922e03:abc123

liveltekah:3f230640b78d7e71ac5514e57935eb69:qazxsw

blikimore:917eb5e9d6d6bca820922a0c6f7cc28b:Pa$$word1

johnwick007:f6a0cb102c62879d397b12b62c092c06:bluered  
  
  
Dictionaries method:  
C:\Users\kalya\OneDrive\Desktop\hashcat\hashcat-6.1.0>hashcat -m 0 -a 0 --username passwords.txt dicts/rockyou.txt --force

hashcat (v6.1.0) 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.

OpenCL API (OpenCL 3.0 ) - Platform #1 [Intel(R) Corporation]

=============================================================

\* Device #1: Intel(R) Iris(R) Xe Graphics, 3062/3126 MB (1563 MB allocatable), 80MCU

OpenCL API (OpenCL 3.0 D3D12 Implementation) - Platform #2 [Microsoft]

======================================================================

\* Device #2: Intel(R) Iris(R) Xe Graphics, 3843/3907 MB (976 MB allocatable), 1MCU

\* Device #3: Microsoft Basic Render Driver, skipped

Minimum password length supported by kernel: 0

Maximum password length supported by kernel: 256

Hashes: 19 digests; 19 unique digests, 1 unique salts

Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Rules: 1

Applicable optimizers applied:

\* Zero-Byte

\* Early-Skip

\* Not-Salted

\* Not-Iterated

\* Single-Salt

\* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.

Using pure kernels enables cracking 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.

INFO: Removed 13 hashes found in potfile.

Host memory required for this attack: 321 MB

Dictionary cache hit:

\* Filename..: dicts/rockyou.txt

\* Passwords.: 14344384

\* Bytes.....: 139921497

\* Keyspace..: 14344384

Approaching final keyspace - workload adjusted.

Session..........: hashcat

Status...........: Exhausted

Hash.Name........: MD5

Hash.Target......: passwords.txt

Time.Started.....: Sat Jun 22 05:53:40 2024, (3 secs)

Time.Estimated...: Sat Jun 22 05:53:43 2024, (0 secs)

Guess.Base.......: File (dicts/rockyou.txt)

Guess.Queue......: 1/1 (100.00%)

Speed.#1.........: 2909.0 kH/s (7.76ms) @ Accel:256 Loops:1 Thr:8 Vec:1

Speed.#2.........: 1956.2 kH/s (0.00ms) @ Accel:1024 Loops:1 Thr:64 Vec:1

Speed.#\*.........: 4865.2 kH/s

Recovered........: 13/19 (68.42%) Digests

Progress.........: 14344384/14344384 (100.00%)

Rejected.........: 0/14344384 (0.00%)

Restore.Point....: 14327100/14344384 (99.88%)

Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1

Restore.Sub.#2...: Salt:0 Amplifier:0-1 Iteration:0-1

Candidates.#1....: 0201393 -> $am$ungu

Candidates.#2....: $HEX[24616d24756e67] -> $HEX[042a0337c2a156616d6f732103]

Started: Sat Jun 22 05:53:07 2024

Stopped: Sat Jun 22 05:53:44 2024

C:\Users\kalya\OneDrive\Desktop\hashcat\hashcat-6.1.0>hashcat -m 0 --username passwords.txt --show

ortspoon:d8578edf8458ce06fbc5bb76a58c5ca4:qwerty

bookma:25d55ad283aa400af464c76d713c07ad:12345678

heroanhart:7c6a180b36896a0a8c02787eeafb0e4c:password1

simmson56:96e79218965eb72c92a549dd5a330112:111111

interestec:25f9e794323b453885f5181f1b624d0b:123456789

eatingcake1994:fcea920f7412b5da7be0cf42b8c93759:1234567

edi\_tesla89:6c569aabbf7775ef8fc570e228c16b98:password!

reallychel:5f4dcc3b5aa765d61d8327deb882cf99:password

experthead:e10adc3949ba59abbe56e057f20f883e:123456

popularkiya7:e99a18c428cb38d5f260853678922e03:abc123

liveltekah:3f230640b78d7e71ac5514e57935eb69:qazxsw

blikimore:917eb5e9d6d6bca820922a0c6f7cc28b:Pa$$word1

johnwick007:f6a0cb102c62879d397b12b62c092c06:bluered

Observations:

1)What type of hashing algorithm was used to protect passwords?

Ans: Md5 is used for all 19 codes.

2)What level of protection does the mechanism offer for passwords?

Ans: Here, the MD5 algorithm is used, although it is one of the famous cryptographic hash functions, it offers limited protection for passwords. MD5 produces a 128-bit hash value, often displayed as a 32-character hexadecimal number. However, it is prone to collision attacks, where different inputs generate the same hash. Additionally, MD5's fast computation speed makes it vulnerable to brute-force attacks. Precomputed rainbow tables further weaken its security, as they allow attackers to reverse-engineer hash values. MD5 also lacks salting, meaning identical passwords yield identical hashes

3)What controls could be implemented to make cracking much harder for the hacker in the event of a password database leaking again?

1. Use Strong Hashing Algorithms: Replace MD5 with stronger algorithms like bcrypt, Argon2, or PBKDF2.
2. Salting: Add a unique salt to each password before hashing to ensure different hash values for identical passwords.
3. Peppering: Add a secret value (pepper) to passwords before hashing, kept secret within the application.
4. Rate Limiting: Limit the number of login attempts per user or IP address within a specific timeframe.
5. **Multi-Factor Authentication (MFA)**: Require additional verification factors like one-time passwords (OTPs).

4)What can you tell about the organization’s password policy (e.g. password length, key space, etc.)?

1. There is no minimum password length requirement.
2. There is no requirement for a mix of different types of cases (uppercase and lowercase).
3. There is no strength meter to guide users in creating strong passwords.
4. There is no mandate for the use of at least one special character.

5)What would you change in the password policy to make breaking the passwords harder?

i) The password must be a minimum of 12 characters.

ii) The password must include a mix of uppercase letters, lowercase letters, numbers, and special characters.

iii) An external API-based tool should be used to check and ensure that the password strength is rated as strong.

iv) Passwords should not include common passwords from a banned list to avoid easily guessable options.

v) Password history should be enforced to prevent reuse of recent passwords, and passwords should be changed periodically (e.g., every 90 days).