Lab 8 Report

CSL 6010 - Cyber Security

Rahul Barodia

B20CS047

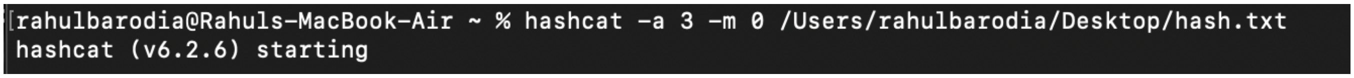
1) Hashcat

I installed Hashcat on my Mac OS. And the password cracking using Hashcat will be executed in the terminal.

I created a .txt file in Desktop named hash.txt.

hash.txt contains the md5 hash of my password. In my example I have chosen 123 as my password. The md5 hash for 123 is202cb962ac59075b964b07152d234b70 .

For executing hashcat we need to execute the following command on terminal



```
[rahulbarodia@Rahuls-MacBook-Air ~ % hashcat -a 3 -m 0 /Users/rahulbarodia/Desktop/hash.txt
hashcat (v6.2.6) starting
```

```
hashcat -a 3 -m 0 /path of hash.txt file/
```

In Hashcat, the -a option is used to specify the attack mode, and the -m option is used to specify the hash type.

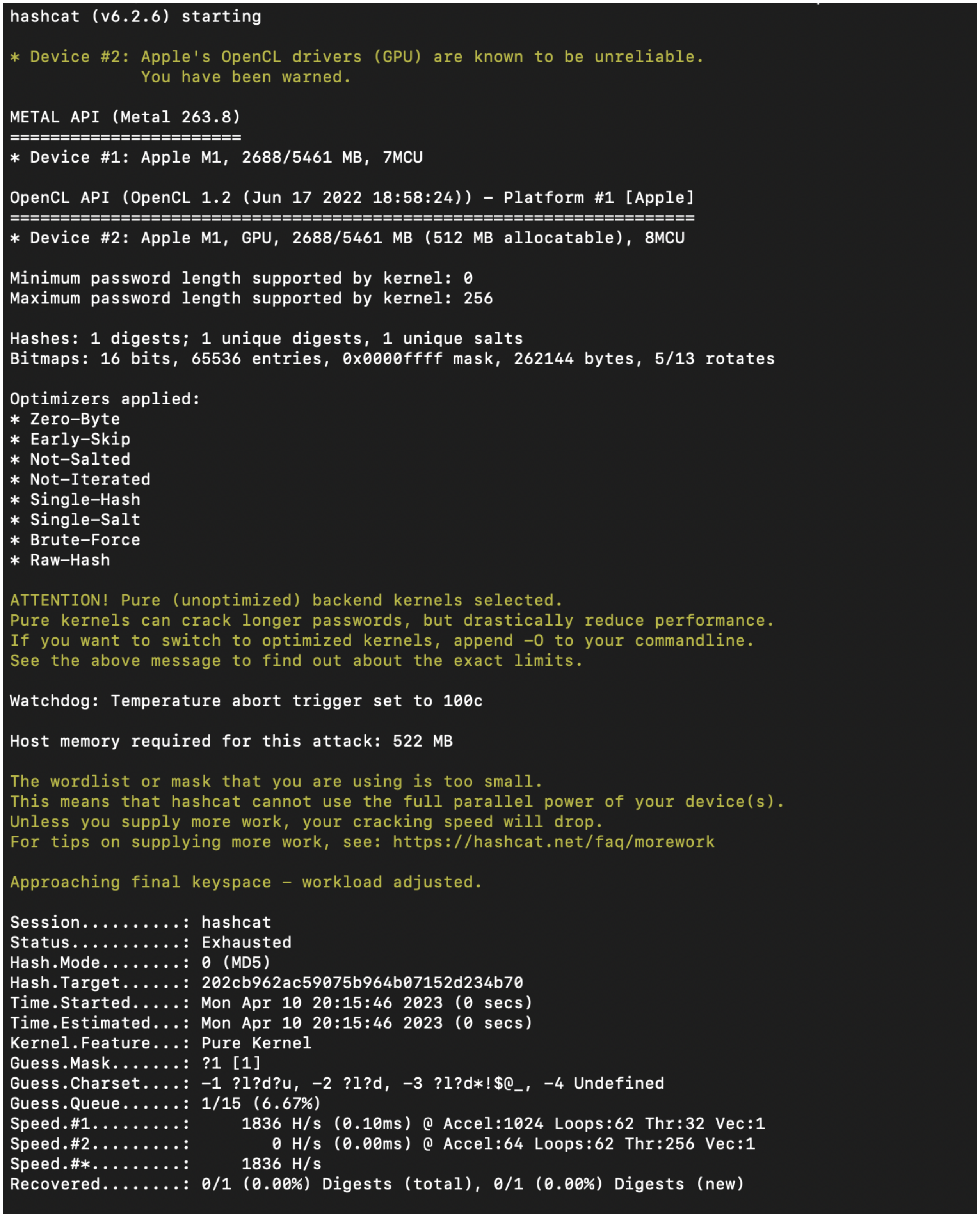
-a 3 specifies a brute force attack mode

-m 0 indicates that the hash stored in hash.txt file is md5 type hash.

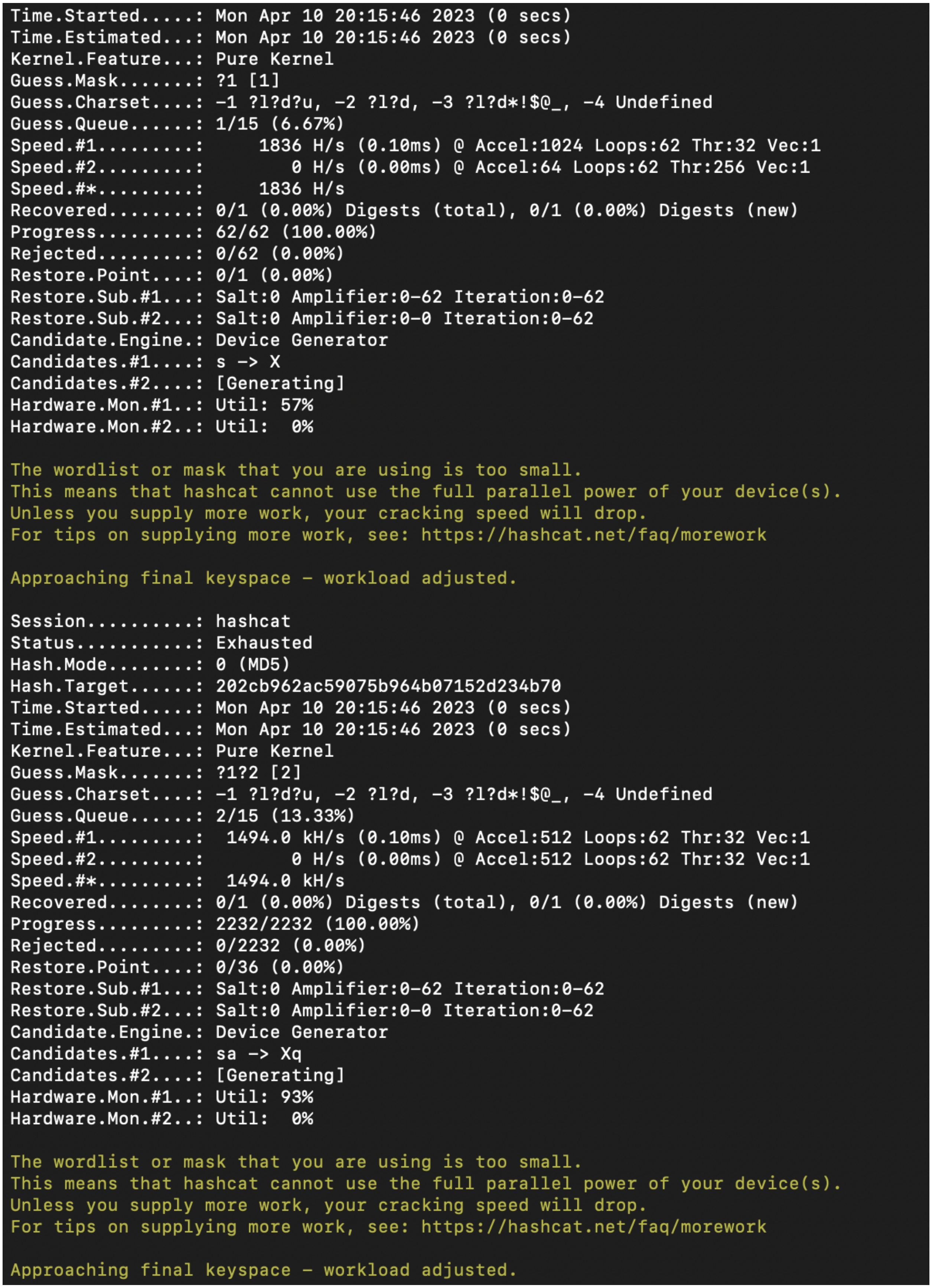
Other attack modes are: -a 0 : Dictionary attack mode, -a 1 : Hybrid attack mode ,etc.

Other hash types are : -m 500 : md5crypt , -m 1000 : SHA 256. etc.

Upon clicking enter we see the following :

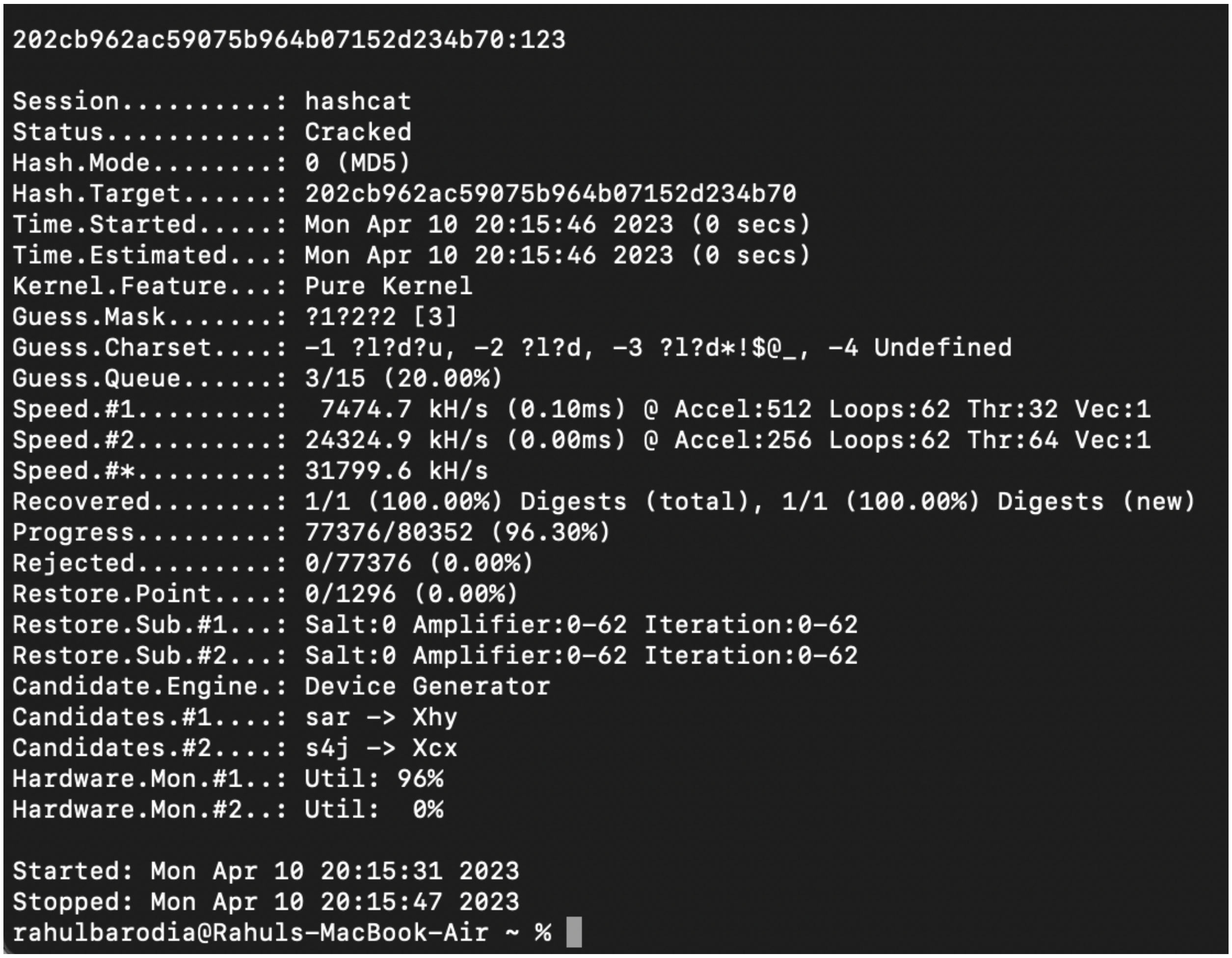


```
hashcat (v6.2.6) starting

* Device #2: Apple's OpenCL drivers (GPU) are known to be unreliable.
             You have been warned.

METAL API (Metal 263.8)
=======================
* Device #1: Apple M1, 2688/5461 MB, 7MCU

OpenCL API (OpenCL 1.2 (Jun 17 2022 18:58:24)) - Platform #1 [Apple]
====================================================================
* Device #2: Apple M1, GPU, 2688/5461 MB (512 MB allocatable), 8MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

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

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Brute-Force
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 100c

Host memory required for this attack: 522 MB

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 0 (MD5)
Hash.Target......: 202cb962ac59075b964b07152d234b70
Time.Started.....: Mon Apr 10 20:15:46 2023 (0 secs)
Time.Estimated...: Mon Apr 10 20:15:46 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?1 [1]
Guess.Charset....: -1 ?l?d?u, -2 ?l?d, -3 ?l?d*!$@_, -4 Undefined
Guess.Queue......: 1/15 (6.67%)
Speed.#1.........:     1836 H/s (0.10ms) @ Accel:1024 Loops:62 Thr:32 Vec:1
Speed.#2.........:        0 H/s (0.00ms) @ Accel:64 Loops:62 Thr:256 Vec:1
Speed.#*.........:     1836 H/s
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
```



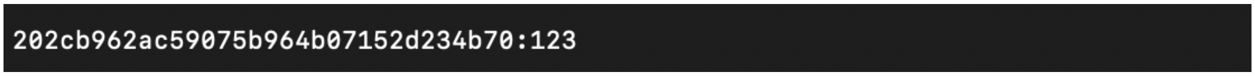
```
Time.Started.....: Mon Apr 10 20:15:46 2023 (0 secs)
Time.Estimated...: Mon Apr 10 20:15:46 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?1 [1]
Guess.Charset....: -1 ?l?d?u, -2 ?l?d, -3 ?l?d*!$@_, -4 Undefined
Guess.Queue......: 1/15 (6.67%)
Speed.#1.........:     1836 H/s (0.10ms) @ Accel:1024 Loops:62 Thr:32 Vec:1
Speed.#2.........:        0 H/s (0.00ms) @ Accel:64 Loops:62 Thr:256 Vec:1
Speed.#*.........:     1836 H/s
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 62/62 (100.00%)
Rejected.........: 0/62 (0.00%)
Restore.Point....: 0/1 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-62 Iteration:0-62
Restore.Sub.#2...: Salt:0 Amplifier:0-0 Iteration:0-62
Candidate.Engine.: Device Generator
Candidates.#1....: s -> X
Candidates.#2....: [Generating]
Hardware.Mon.#1..: Util: 57%
Hardware.Mon.#2..: Util:  0%

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 0 (MD5)
Hash.Target......: 202cb962ac59075b964b07152d234b70
Time.Started.....: Mon Apr 10 20:15:46 2023 (0 secs)
Time.Estimated...: Mon Apr 10 20:15:46 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?1?2 [2]
Guess.Charset....: -1 ?l?d?u, -2 ?l?d, -3 ?l?d*!$@_, -4 Undefined
Guess.Queue......: 2/15 (13.33%)
Speed.#1.........:  1494.0 kH/s (0.10ms) @ Accel:512 Loops:62 Thr:32 Vec:1
Speed.#2.........:        0 H/s (0.00ms) @ Accel:512 Loops:62 Thr:32 Vec:1
Speed.#*.........:  1494.0 kH/s
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 2232/2232 (100.00%)
Rejected.........: 0/2232 (0.00%)
Restore.Point....: 0/36 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-62 Iteration:0-62
Restore.Sub.#2...: Salt:0 Amplifier:0-0 Iteration:0-62
Candidate.Engine.: Device Generator
Candidates.#1....: sa -> Xq
Candidates.#2....: [Generating]
Hardware.Mon.#1..: Util: 93%
Hardware.Mon.#2..: Util:  0%

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.
```

Password cracking:



```
202cb962ac59075b964b07152d234b70:123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 202cb962ac59075b964b07152d234b70
Time.Started.....: Mon Apr 10 20:15:46 2023 (0 secs)
Time.Estimated...: Mon Apr 10 20:15:46 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?1?2?2 [3]
Guess.Charset....: -1 ?l?d?u, -2 ?l?d, -3 ?l?d*!$@_, -4 Undefined
Guess.Queue......: 3/15 (20.00%)
Speed.#1.........:  7474.7 kH/s (0.10ms) @ Accel:512 Loops:62 Thr:32 Vec:1
Speed.#2.........: 24324.9 kH/s (0.00ms) @ Accel:256 Loops:62 Thr:64 Vec:1
Speed.#*.........: 31799.6 kH/s
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 77376/80352 (96.30%)
Rejected.........: 0/77376 (0.00%)
Restore.Point....: 0/1296 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-62 Iteration:0-62
Restore.Sub.#2...: Salt:0 Amplifier:0-62 Iteration:0-62
Candidate.Engine.: Device Generator
Candidates.#1....: sar -> Xhy
Candidates.#2....: s4j -> Xcx
Hardware.Mon.#1..: Util: 96%
Hardware.Mon.#2..: Util:  0%

Started: Mon Apr 10 20:15:31 2023
Stopped: Mon Apr 10 20:15:47 2023
rahulbarodia@Rahuls-MacBook-Air ~ %
```

From the first line of the above screenshot we see that Hashcat successfully cracked the password.



```
202cb962ac59075b964b07152d234b70:123
```

2) Rainbow Crack

Rainbow Crack also uses brute force attacks and dictionary attacks against various hashed password formats.

It used pre computed rainbow tables.

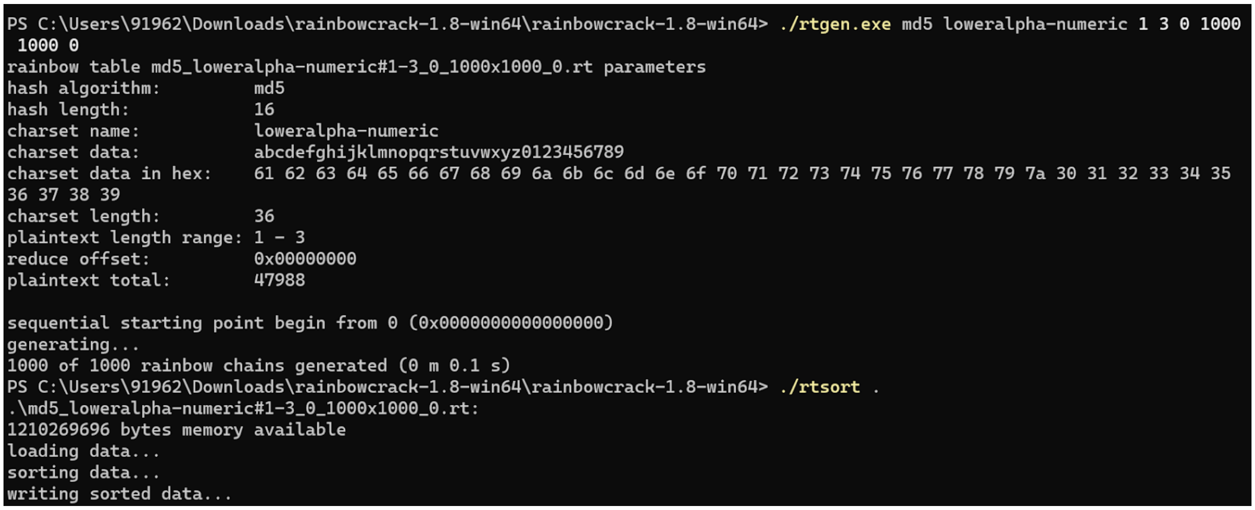
I stored my password “bye” in md5 hashed format.

Md5 for bye is bfa99df33b137bc8fb5f5407d7e58da8.

The following command can be used :

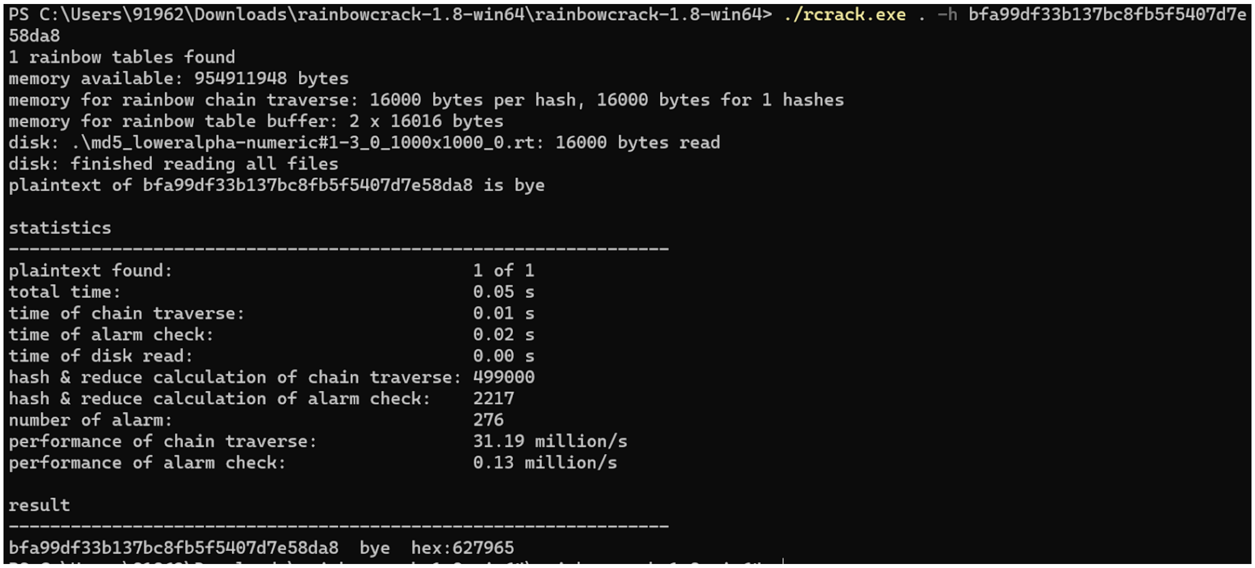
```
rtgen md5 loweralpha-plain 6 6 0 1000000 rainbowtable.rt
```

Execution:



```
PS C:\Users\91962\Downloads\rainbowcrack-1.8-win64\rainbowcrack-1.8-win64> ./rtgen.exe md5 loweralpha-numeric 1 3 0 1000
 1000 0
rainbow table md5_loweralpha-numeric#1-3_0_1000x1000_0.rt parameters
hash algorithm:         md5
hash length:            16
charset name:           loweralpha-numeric
charset data:           abcdefghijklmnopqrstuvwxyz0123456789
charset data in hex:    61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 30 31 32 33 34 35
36 37 38 39
charset length:         36
plaintext length range: 1 - 3
reduce offset:          0x00000000
plaintext total:        47988

sequential starting point begin from 0 (0x0000000000000000)
generating...
1000 of 1000 rainbow chains generated (0 m 0.1 s)
PS C:\Users\91962\Downloads\rainbowcrack-1.8-win64\rainbowcrack-1.8-win64> ./rtsort .
.\md5_loweralpha-numeric#1-3_0_1000x1000_0.rt:
1210269696 bytes memory available
loading data...
sorting data...
writing sorted data...
```

Password Cracking:



```
PS C:\Users\91962\Downloads\rainbowcrack-1.8-win64\rainbowcrack-1.8-win64> ./rcrack.exe . -h bfa99df33b137bc8fb5f5407d7e
58da8
1 rainbow tables found
memory available: 954911948 bytes
memory for rainbow chain traverse: 16000 bytes per hash, 16000 bytes for 1 hashes
memory for rainbow table buffer: 2 x 16016 bytes
disk: .\md5_loweralpha-numeric#1-3_0_1000x1000_0.rt: 16000 bytes read
disk: finished reading all files
plaintext of bfa99df33b137bc8fb5f5407d7e58da8 is bye

statistics
----------------------------------------------------------------
plaintext found:                             1 of 1
total time:                                  0.05 s
time of chain traverse:                      0.01 s
time of alarm check:                         0.02 s
time of disk read:                           0.00 s
hash & reduce calculation of chain traverse: 499000
hash & reduce calculation of alarm check:    2217
number of alarm:                             276
performance of chain traverse:               31.19 million/s
performance of alarm check:                  0.13 million/s

result
----------------------------------------------------------------
bfa99df33b137bc8fb5f5407d7e58da8  bye  hex:627965
```

As we can see from the above screenshot the password bye is successfully cracked.

3) Ncrack

Ncarck also uses brute force attacks and dictionary attacks against various network protocols.

I will demonstrate a dictionary attack with the use of Ncrack.

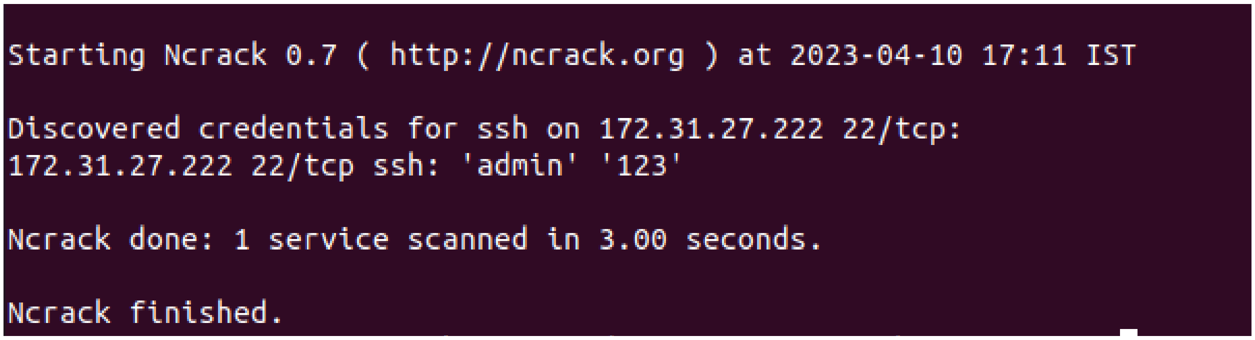
I have created two lists named usernames.txt and passwords.txt.

username.txt contains the list of most common user names and passwords.txt contains most common passwords. Ncrack will find if any password present is weak and tries to crack it.

For executing Ncrack following command is used:

```
ncrack -U /file path of usernames/ -P. /file path of passwords/ ssh://ip address
```

After executing the command:



```
Starting Ncrack 0.7 ( http://ncrack.org ) at 2023-04-10 17:11 IST

Discovered credentials for ssh on 172.31.27.222 22/tcp:
172.31.27.222 22/tcp ssh: 'admin' '123'

Ncrack done: 1 service scanned in 3.00 seconds.

Ncrack finished.
```

As we can see from the above screenshot that the ncrack cracked the credentials of the user admin.

The password is 123.

Time taken = 3seconds

Comparison Factor	Hashcat	Rainbow Crack	Ncrack
Type of attack	Brute force, Dictionary attacks, combinator attacks, hybrid attacks, etc.	Brute force, Dictionary attacks and hybrid attacks.	Brute force and Dictionary attacks.
Used for	Cracking many type of hashed passwords. For ex md5, sha1 ,etc.	Cracking many type of hashed passwords. For ex md5, sha1 ,etc. Uses rainbow tables.	Cracking various protocols such as ssh, http,etc.
Type of password able to crack	Md5, sha-1,sha-256,etc.	Md5, sha-1,sha-256,etc.	Ssh, http, etc.
Platform	Supports windows, MacOS and Linux.	Supports windows, MacOS and linux.	Supports windows, MacOS and Linux.
Time Taken	In minutes or seconds (Faster than Ncrack). Depends on hardware and optimization.	In minutes or seconds (Faster than Ncrack). Depends on optimized settings.	In minutes or seconds. Depends on protocol being attacked.
Performance	Equivalent to Rainbow crack and superior than Ncrack	Equivalent to Hashcat and superior than Ncrack	Slowest among three.