LinkedIn: Kelvin Kimotho

Crack the hash

Cracking hashes challenges

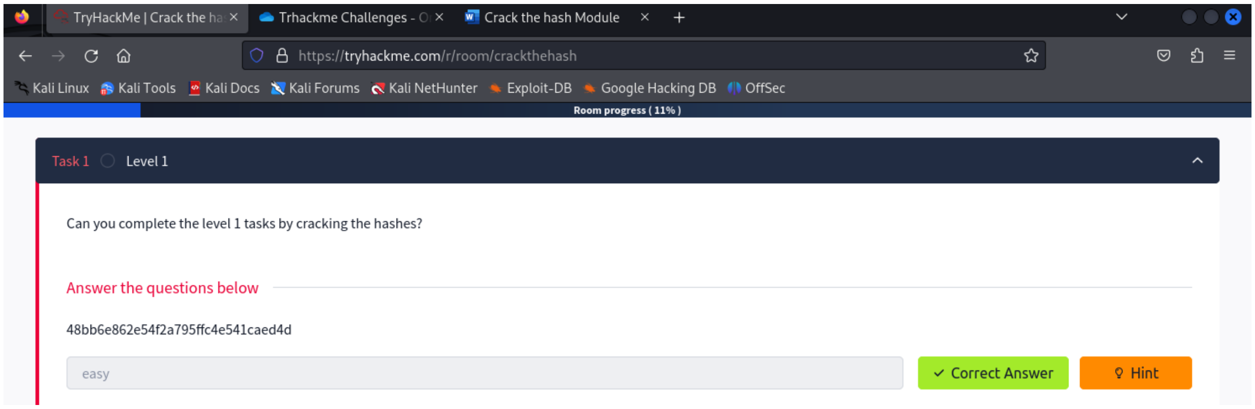
Level 1

Can you complete the level 1 tasks by cracking the hashes?

Answer the questions below

1. 48bb6e862e54f2a795ffc4e541caed4d

Answer: easy



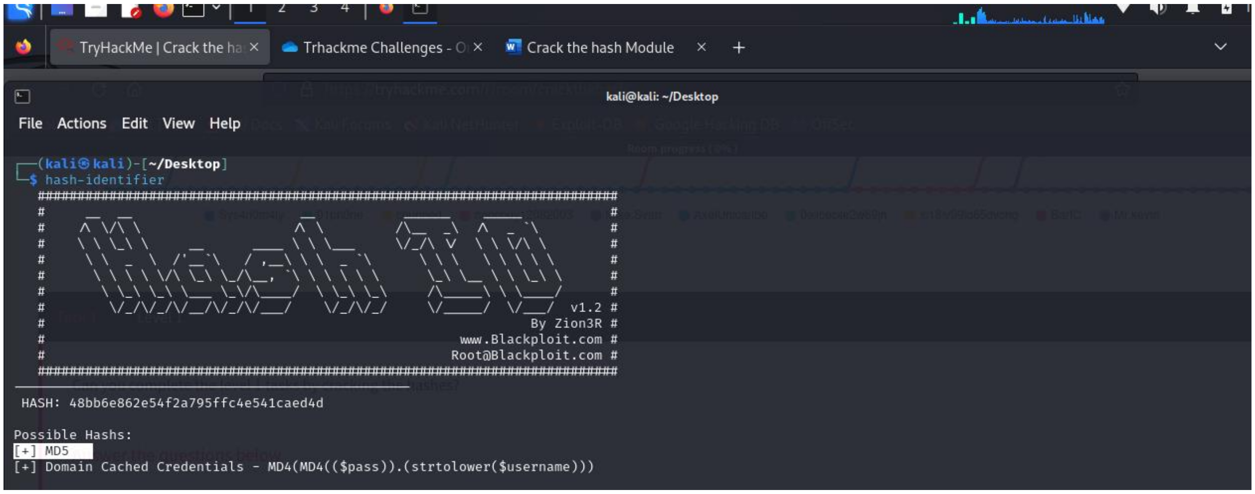
Can you complete the level 1 tasks by cracking the hashes?

Answer the questions below

48bb6e862e54f2a795ffc4e541caed4d

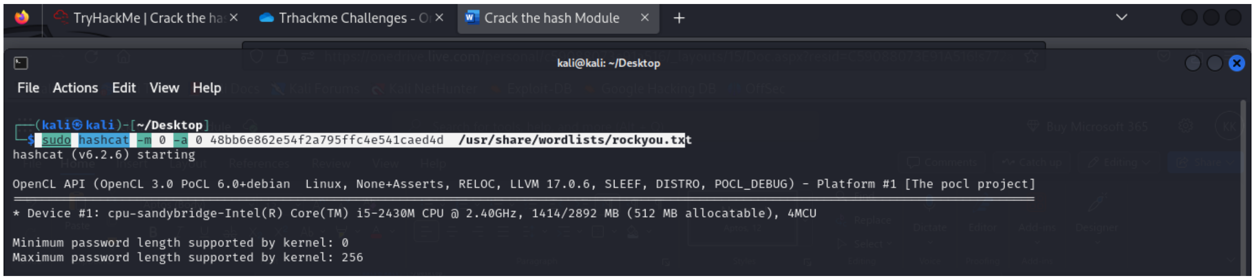
easy

The first this was to determine what type the hash is using a **hash-identifie**r tool which comes pre-installed on kali**.**

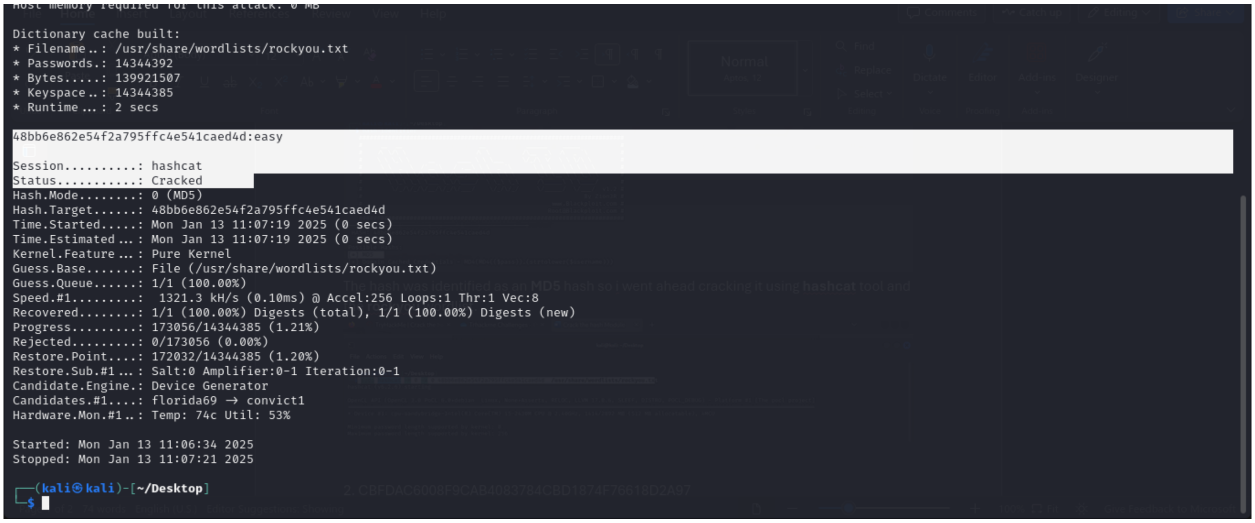


```
┌──(kali㉿kali)-[~/Desktop]
└─$ hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: 48bb6e862e54f2a795ffc4e541caed4d

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

The hash was identified as an **MD5** hash so i went ahead cracking it using **hashcat** tool and the **rockyou** wordlist.



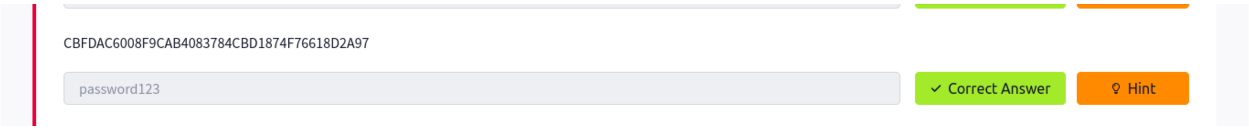
```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo hashcat -m 0 -a 0 48bb6e862e54f2a795ffc4e541caed4d  /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 1414/2892 MB (512 MB allocatable), 4MCU

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



```
Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 2 secs

48bb6e862e54f2a795ffc4e541caed4d:easy

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 48bb6e862e54f2a795ffc4e541caed4d
Time.Started.....: Mon Jan 13 11:07:19 2025 (0 secs)
Time.Estimated...: Mon Jan 13 11:07:19 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1321.3 kH/s (0.10ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 173056/14344385 (1.21%)
Rejected.........: 0/173056 (0.00%)
Restore.Point....: 172032/14344385 (1.20%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: florida69 → convict1
Hardware.Mon.#1..: Temp: 74c Util: 53%

Started: Mon Jan 13 11:06:34 2025
Stopped: Mon Jan 13 11:07:21 2025

┌──(kali㉿kali)-[~/Desktop]
└─$ █
```

2. CBFDAC6008F9CAB4083784CBD1874F76618D2A97

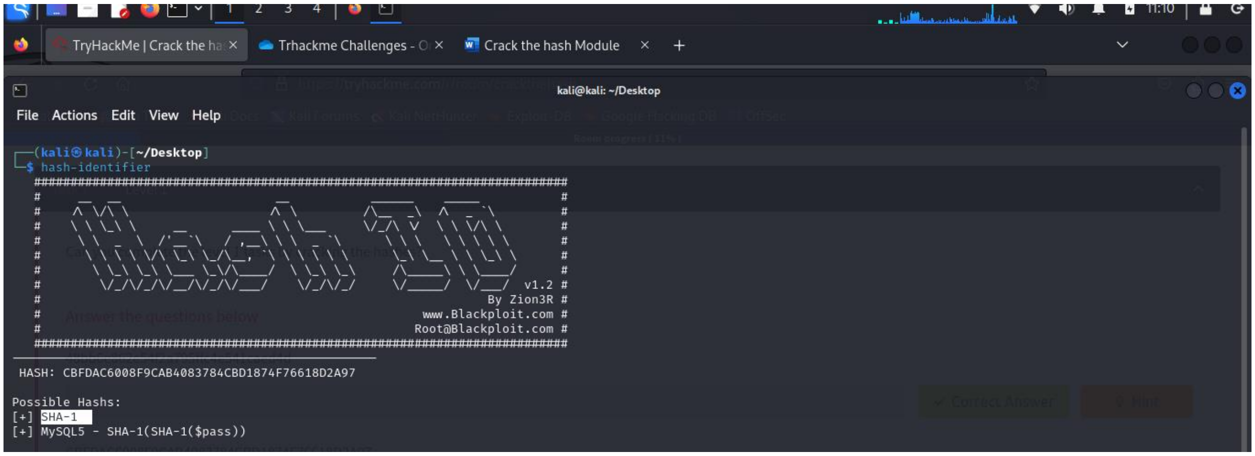
Answer: password123



CBFDAC6008F9CAB4083784CBD1874F76618D2A97

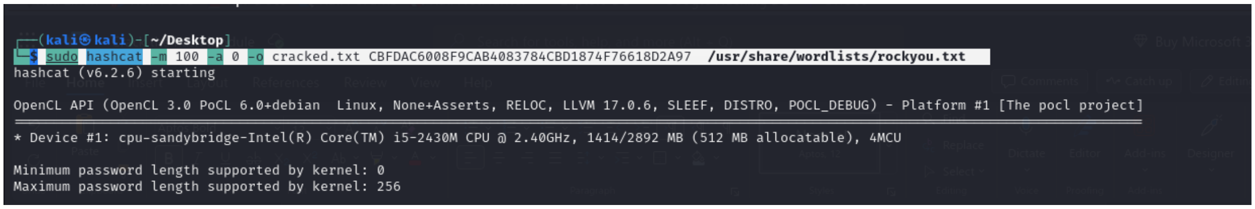
password123

The first this was to determine what type the hash is using a **hash-identifier** tool. It was a **sha-1** hash.

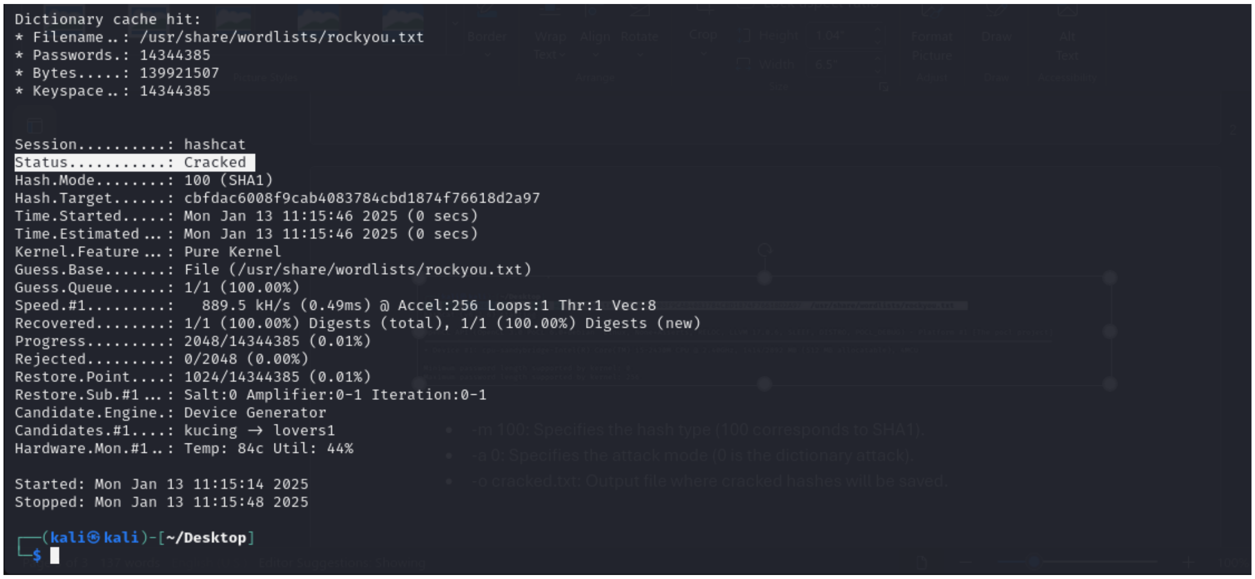


```
┌──(kali㉿kali)-[~/Desktop]
└─$ hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: CBFDAC6008F9CAB4083784CBD1874F76618D2A97

Possible Hashs:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))
```

Went ahead cracking it using **hashcat** tool and the **rockyou** wordlist.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo hashcat -m 100 -a 0 -o cracked.txt CBFDAC6008F9CAB4083784CBD1874F76618D2A97  /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==================================================================================================================================================
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 1414/2892 MB (512 MB allocatable), 4MCU

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

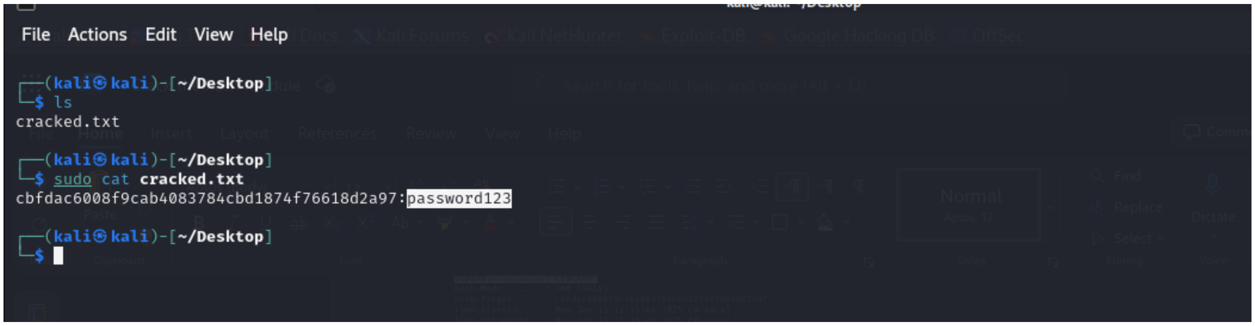


```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385


Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 100 (SHA1)
Hash.Target......: cbfdac6008f9cab4083784cbd1874f76618d2a97
Time.Started.....: Mon Jan 13 11:15:46 2025 (0 secs)
Time.Estimated...: Mon Jan 13 11:15:46 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   889.5 kH/s (0.49ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 2048/14344385 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 1024/14344385 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: kucing → lovers1
Hardware.Mon.#1..: Temp: 84c Util: 44%

Started: Mon Jan 13 11:15:14 2025
Stopped: Mon Jan 13 11:15:48 2025

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

- -m 100: Specifies the hash type (100 corresponds to SHA1).
- -a 0: Specifies the attack mode (0 is the dictionary attack).

- -o cracked.txt: Output file where cracked hashes will be saved.

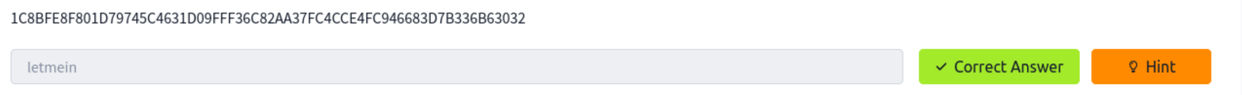
The answer was:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ls
cracked.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo cat cracked.txt
cbfdac6008f9cab4083784cbd1874f76618d2a97:password123

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

3. 1C8BFE8F801D79745C4631D09FFF36C82AA37FC4CCE4FC946683D7B336B63032

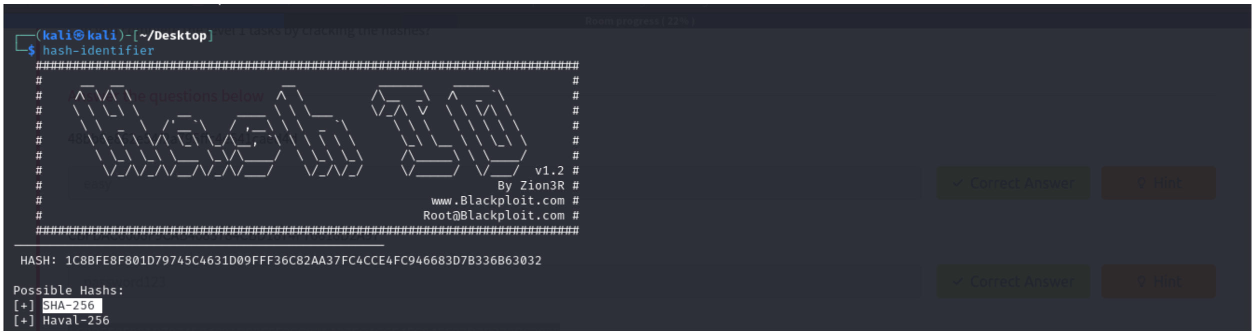
Answer: letmein



1C8BFE8F801D79745C4631D09FFF36C82AA37FC4CCE4FC946683D7B336B63032

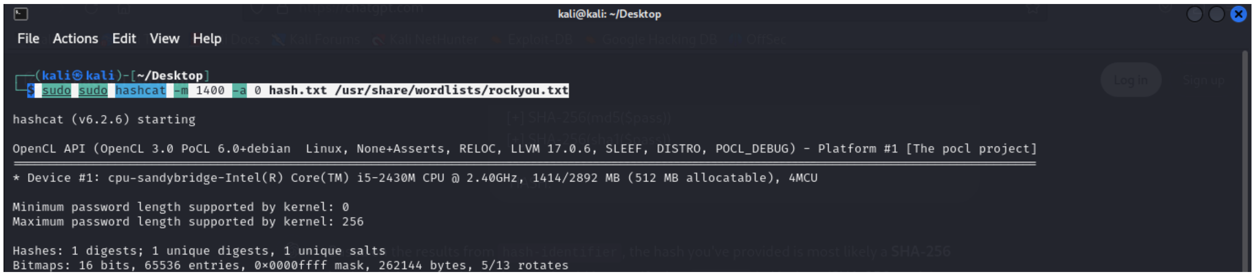
letmein

The first this was to determine what type the hash is using a **hash-identifie**r tool. The hash was identified as a **SHA-256** hash.

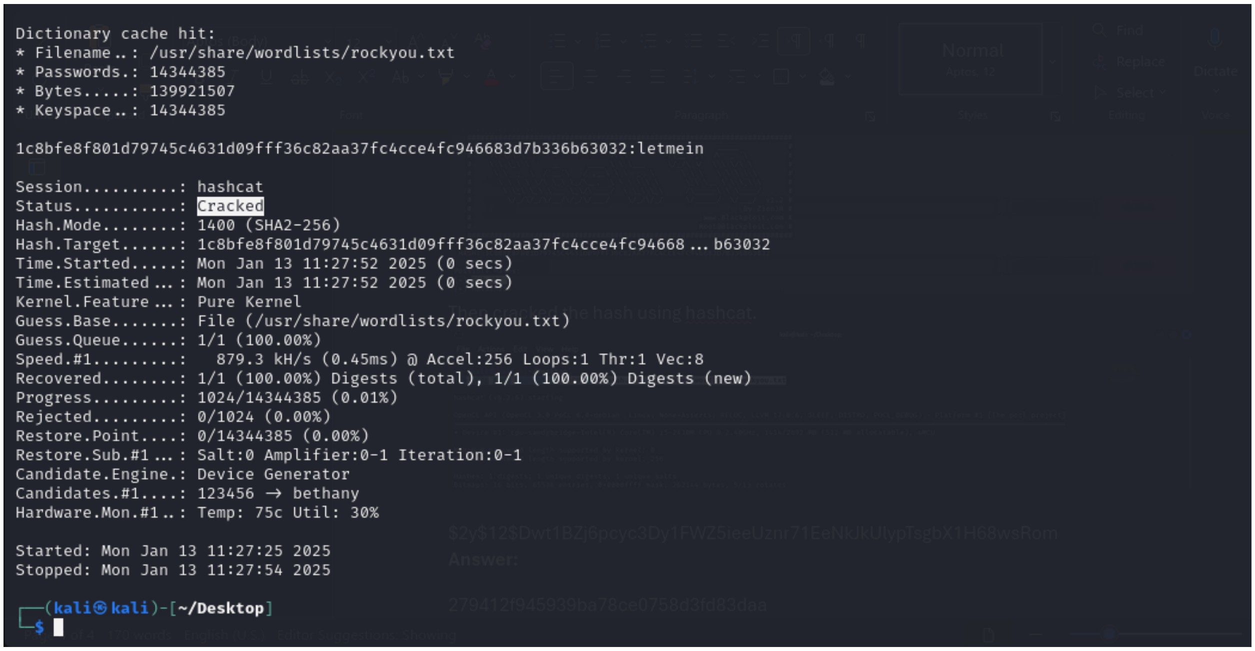


```
┌──(kali㉿kali)-[~/Desktop]
└─$ hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: 1C8BFE8F801D79745C4631D09FFF36C82AA37FC4CCE4FC946683D7B336B63032

Possible Hashs:
[+] SHA-256
[+] Haval-256
```

Then cracked the hash using hashcat.



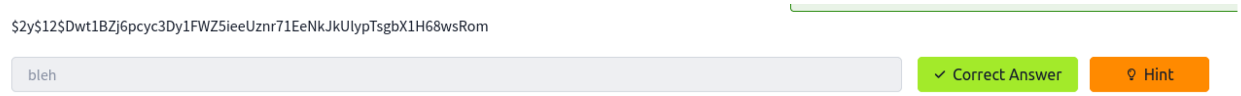
```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo sudo hashcat -m 1400 -a 0 hash.txt /usr/share/wordlists/rockyou.txt

hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 1414/2892 MB (512 MB allocatable), 4MCU

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
```



```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

1c8bfe8f801d79745c4631d09fff36c82aa37fc4cce4fc946683d7b336b63032:letmein

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1400 (SHA2-256)
Hash.Target......: 1c8bfe8f801d79745c4631d09fff36c82aa37fc4cce4fc94668...b63032
Time.Started.....: Mon Jan 13 11:27:52 2025 (0 secs)
Time.Estimated...: Mon Jan 13 11:27:52 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   879.3 kH/s (0.45ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 1024/14344385 (0.01%)
Rejected.........: 0/1024 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456 -> bethany
Hardware.Mon.#1..: Temp: 75c Util: 30%

Started: Mon Jan 13 11:27:25 2025
Stopped: Mon Jan 13 11:27:54 2025

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

4. $2y$12$Dwt1BZj6pcyc3Dy1FWZ5ieeUznr71EeNkJkUlypTsgbX1H68wsRom

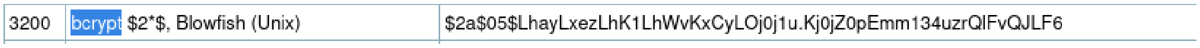
Answer: bleh



$2y$12$Dwt1BZj6pcyc3Dy1FWZ5ieeUznr71EeNkJkUlypTsgbX1H68wsRom

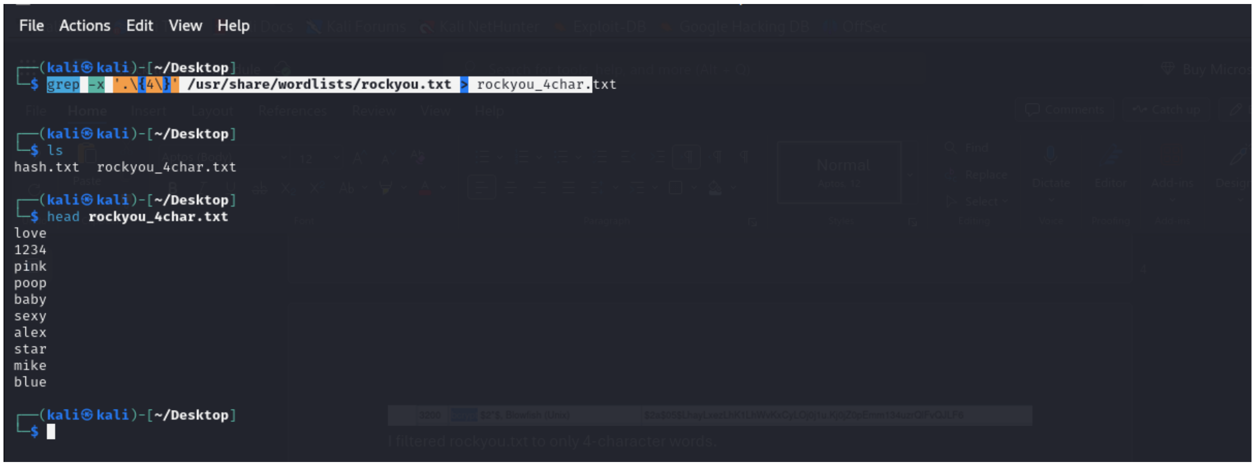
bleh

I went searching from the hashcat examples page (https://hashcat.net/wiki/doku.php?id=example_hashes) for $2y$ and found out that the hash was a **bcrypt** hash.



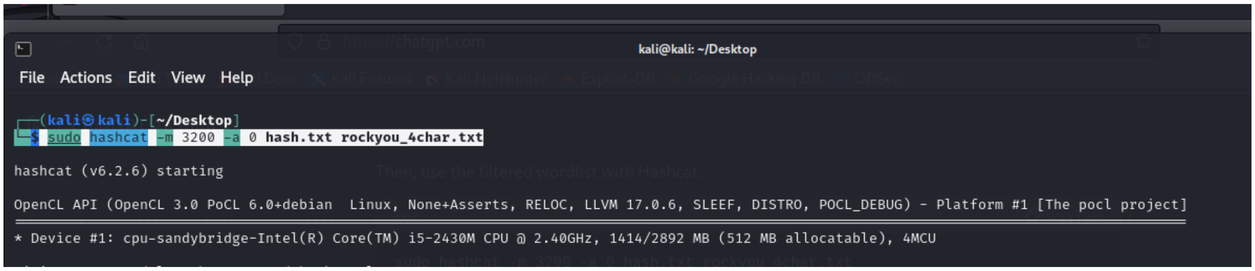
3200	bcrypt $2*$, Blowfish (Unix)	$2a$05$LhayLxezLhK1LhWvKxCyLOj0j1u.Kj0jZ0pEmm134uzrQlFvQJLF6

I filtered rockyou.txt to only 4-character words.

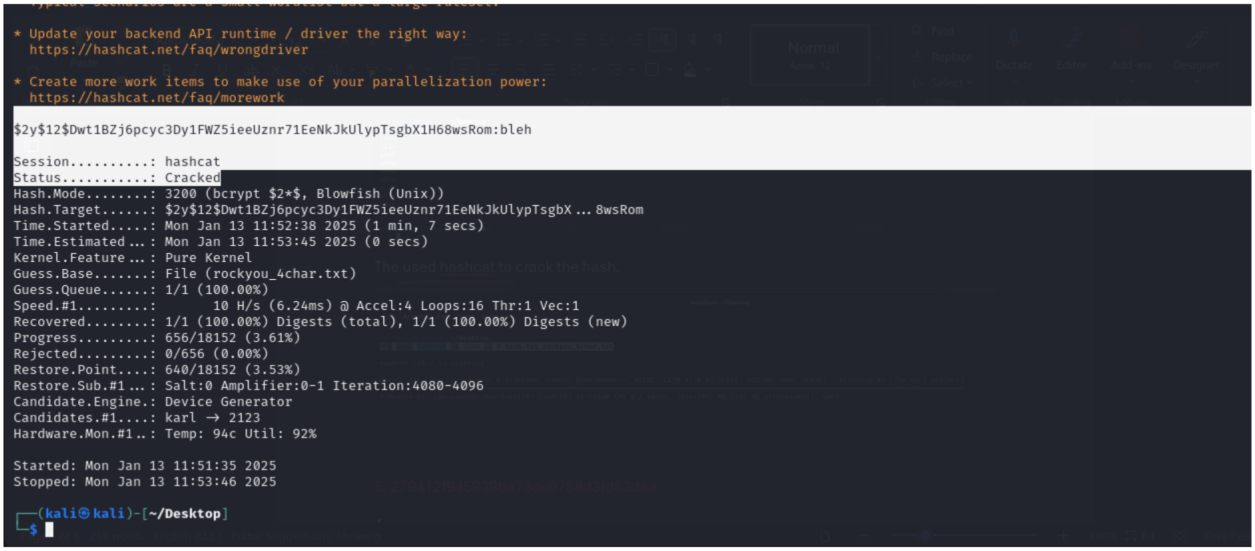


```
┌──(kali㉿kali)-[~/Desktop]
└─$ grep -x '.\{4\}' /usr/share/wordlists/rockyou.txt > rockyou_4char.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ ls
hash.txt  rockyou_4char.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ head rockyou_4char.txt
love
1234
pink
poop
baby
sexy
alex
star
mike
blue

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

The used hashcat to crack the hash.



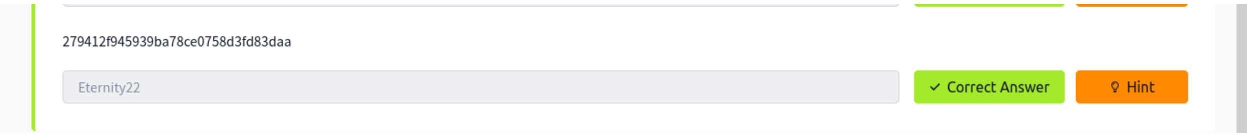
```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo hashcat -m 3200 -a 0 hash.txt rockyou_4char.txt

hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 1414/2892 MB (512 MB allocatable), 4MCU
```



```
Typical scenarios are a small wordlist but a large ruleset.

* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

$2y$12$Dwt1BZj6pcyc3Dy1FWZ5ieeUznr71EeNkJkUlypTsgbX1H68wsRom:bleh

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 3200 (bcrypt $2*$, Blowfish (Unix))
Hash.Target......: $2y$12$Dwt1BZj6pcyc3Dy1FWZ5ieeUznr71EeNkJkUlypTsgbX...8wsRom
Time.Started.....: Mon Jan 13 11:52:38 2025 (1 min, 7 secs)
Time.Estimated...: Mon Jan 13 11:53:45 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou_4char.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       10 H/s (6.24ms) @ Accel:4 Loops:16 Thr:1 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 656/18152 (3.61%)
Rejected.........: 0/656 (0.00%)
Restore.Point....: 640/18152 (3.53%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4080-4096
Candidate.Engine.: Device Generator
Candidates.#1....: karl → 2123
Hardware.Mon.#1..: Temp: 94c Util: 92%

Started: Mon Jan 13 11:51:35 2025
Stopped: Mon Jan 13 11:53:46 2025

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

5. 279412f945939ba78ce0758d3fd83daa

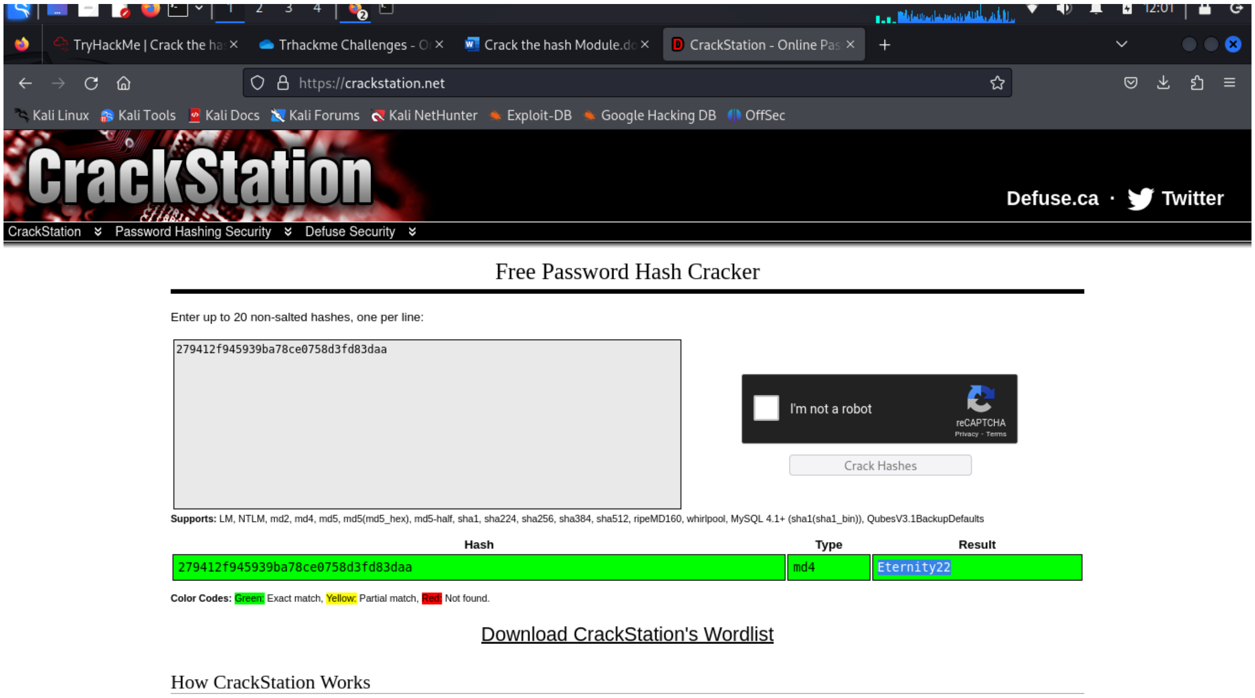
Answer: Eternity22



279412f945939ba78ce0758d3fd83daa

Eternity22

I used crackstation service online to crack this hash.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
279412f945939ba78ce0758d3fd83daa
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
279412f945939ba78ce0758d3fd83daa	md4	Eternity22

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist

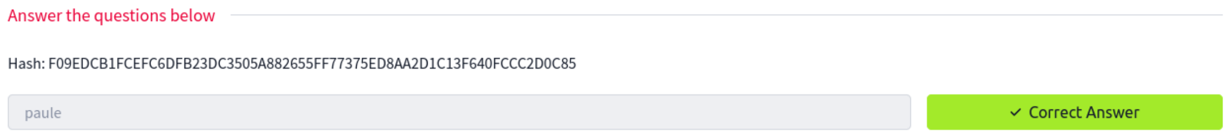
How CrackStation Works

Level 2

Answer the questions below

1. Hash: F09EDCB1FCEFC6DFB23DC3505A882655FF77375ED8AA2D1C13F640FCCC2D0C85

Answer: paule

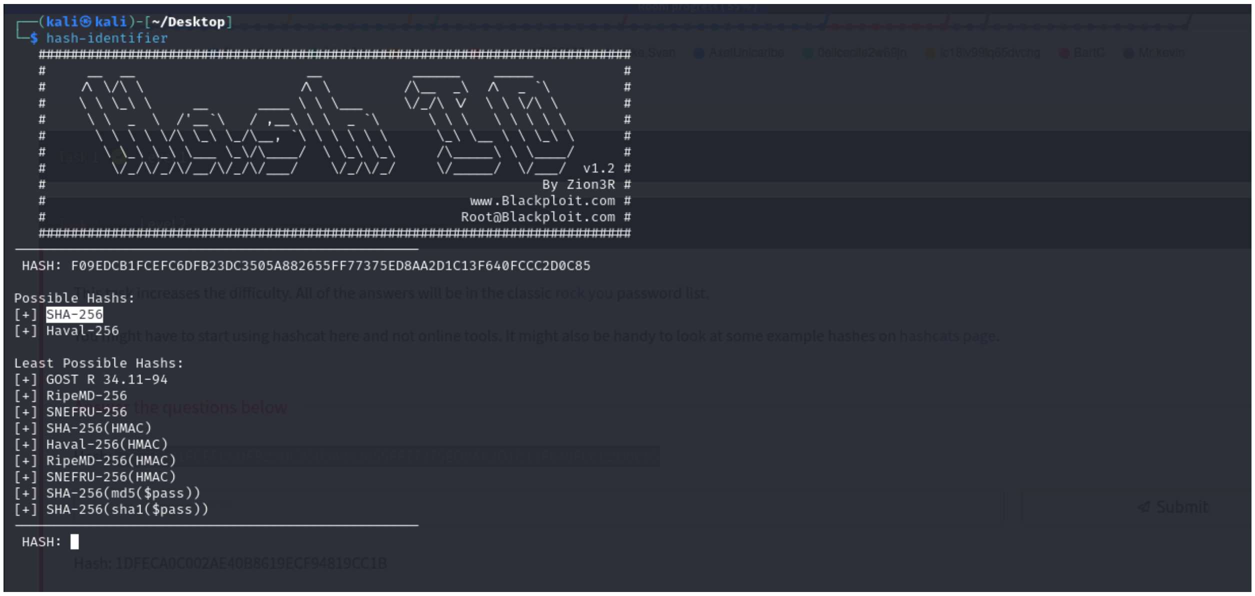


Answer the questions below

Hash: F09EDCB1FCEFC6DFB23DC3505A882655FF77375ED8AA2D1C13F640FCCC2D0C85

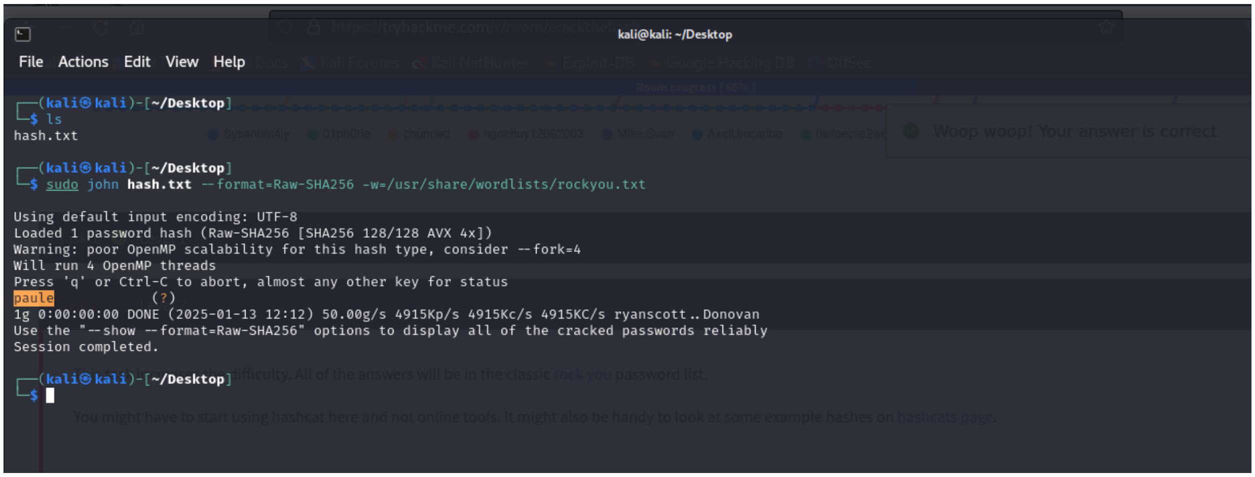
paule

I used **hash-identifie**r tool to determine the type of the hash which turned to be as **sha-256**.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: F09EDCB1FCEFC6DFB23DC3505A882655FF77375ED8AA2D1C13F640FCCC2D0C85

Possible Hashs:
[+] SHA-256
[+] Haval-256

Least Possible Hashs:
[+] GOST R 34.11-94
[+] RipeMD-256
[+] SNEFRU-256
[+] SHA-256(HMAC)
[+] Haval-256(HMAC)
[+] RipeMD-256(HMAC)
[+] SNEFRU-256(HMAC)
[+] SHA-256(md5($pass))
[+] SHA-256(sha1($pass))
--------------------------------------------------
 HASH: █
```

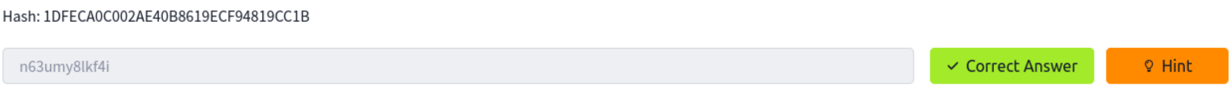
I then went ahead and used **john the ripper** tool to crack the hash.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ls
hash.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo john hash.txt --format=Raw-SHA256 -w=/usr/share/wordlists/rockyou.txt

Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 AVX 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
paule            (?)
1g 0:00:00:00 DONE (2025-01-13 12:12) 50.00g/s 4915Kp/s 4915Kc/s 4915KC/s ryanscott..Donovan
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~/Desktop]
└─$ █
```

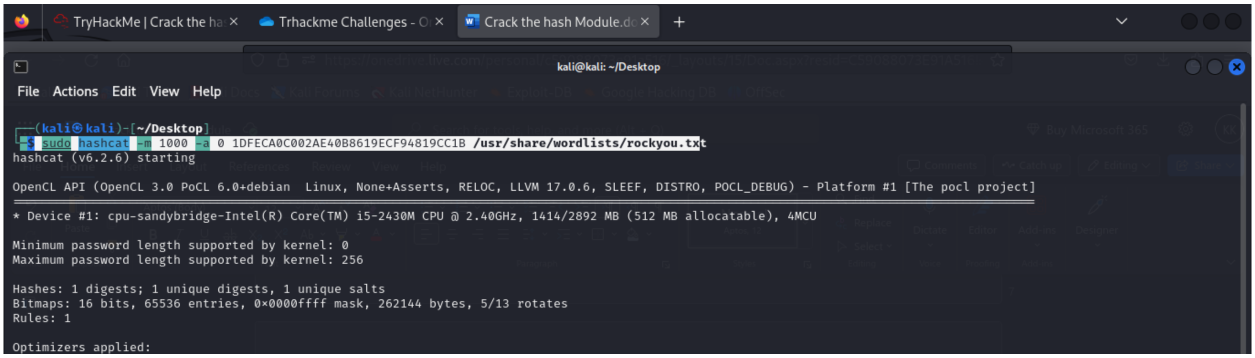
2. Hash: 1DFECA0C002AE40B8619ECF94819CC1B

Answer: n63umy8lkf4i

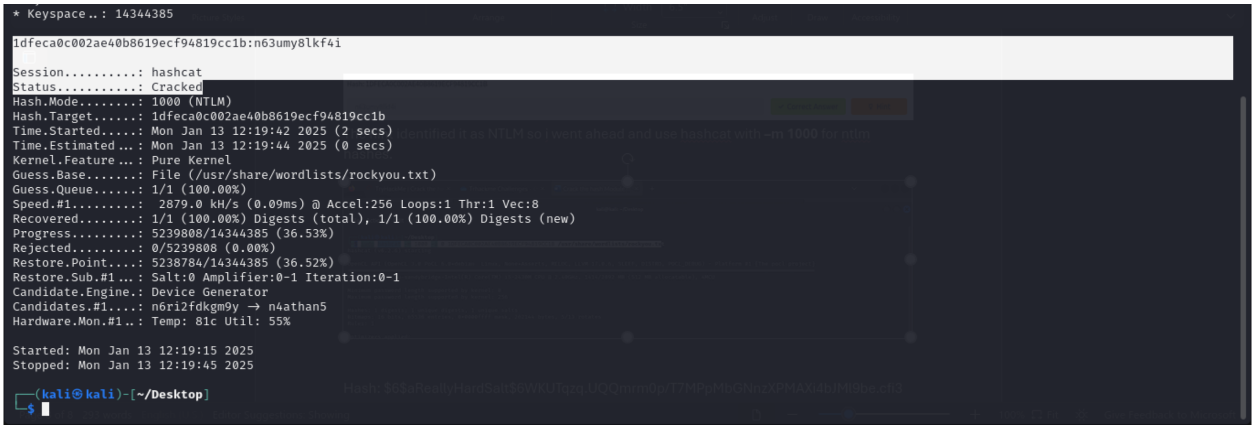

Hash: 1DFECA0C002AE40B8619ECF94819CC1B

n63umy8lkf4i

The hint identified it as NTLM so i went ahead and use hashcat with **–m 1000** for ntlm hashes.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo hashcat -m 1000 -a 0 1DFECA0C002AE40B8619ECF94819CC1B /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 1414/2892 MB (512 MB allocatable), 4MCU

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, 0×0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
```

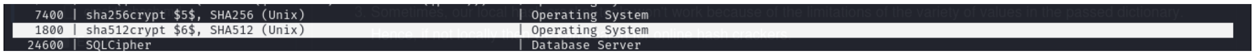


```
* Keyspace..: 14344385

1dfeca0c002ae40b8619ecf94819cc1b:n63umy8lkf4i

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1000 (NTLM)
Hash.Target......: 1dfeca0c002ae40b8619ecf94819cc1b
Time.Started.....: Mon Jan 13 12:19:42 2025 (2 secs)
Time.Estimated...: Mon Jan 13 12:19:44 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  2879.0 kH/s (0.09ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 5239808/14344385 (36.53%)
Rejected.........: 0/5239808 (0.00%)
Restore.Point....: 5238784/14344385 (36.52%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: n6ri2fdkgm9y → n4athan5
Hardware.Mon.#1..: Temp: 81c Util: 55%

Started: Mon Jan 13 12:19:15 2025
Stopped: Mon Jan 13 12:19:45 2025

┌──(kali㉿kali)-[~/Desktop]
└─$ █
```

3.Hash: $6$aReallyHardSalt$6WKUTqzq.UQQmrm0p/T7MPpMbGNnzXPMAXi4bJMl9be.cfi3/qxIf.hsGpS41BqMhSrHVXgMpdjS6xeKZAs02.

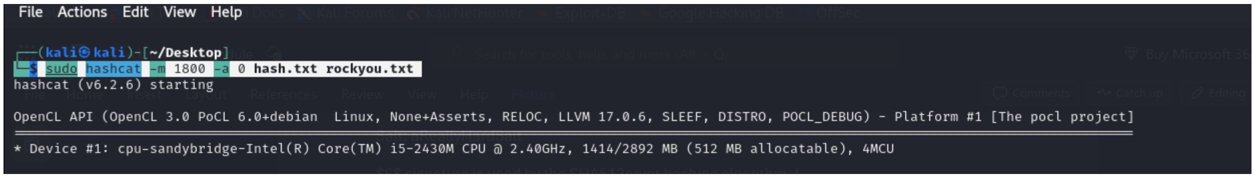
Salt: aReallyHardSalt

$6$ signature is used by the SHA512crypt hashing algorithm. I

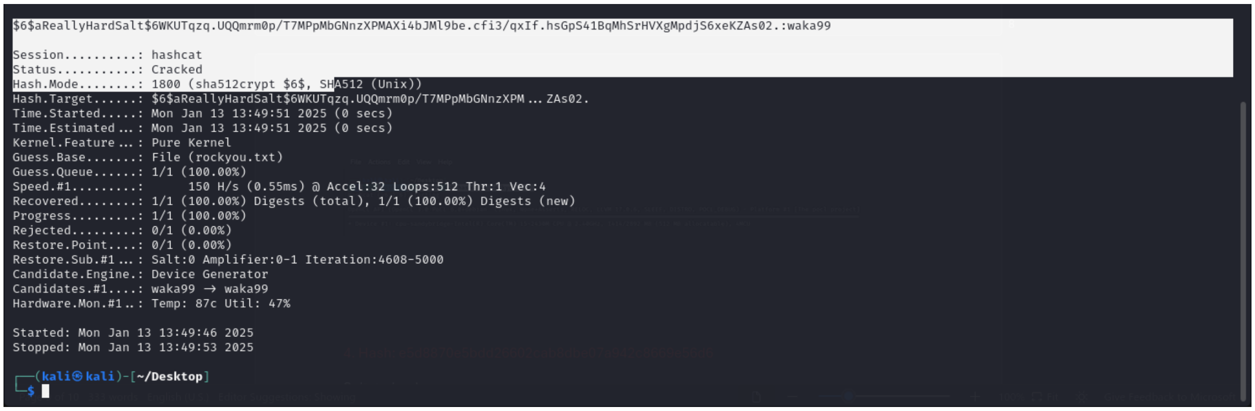


```
   7400 | sha256crypt $5$, SHA256 (Unix)                             | Operating System
   1800 | sha512crypt $6$, SHA512 (Unix)                             | Operating System
  24600 | SQLCipher                                                  | Database Server
```

I cracked this hash using mode **-m 1800** in hashcat.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo hashcat -m 1800 -a 0 hash.txt rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 1414/2892 MB (512 MB allocatable), 4MCU
```

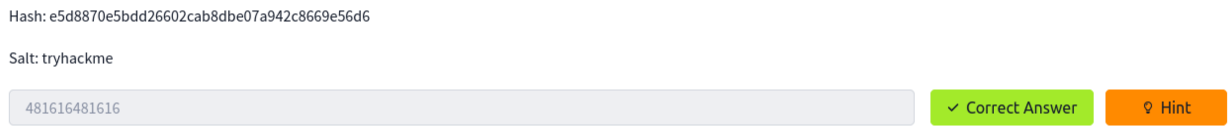


```
$6$aReallyHardSalt$6WKUTqzq.UQQmrm0p/T7MPpMbGNnzXPMAXi4bJMl9be.cfi3/qxIf.hsGpS41BqMhSrHVXgMpdjS6xeKZAs02.:waka99

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target......: $6$aReallyHardSalt$6WKUTqzq.UQQmrm0p/T7MPpMbGNnzXPM...ZAs02.
Time.Started.....: Mon Jan 13 13:49:51 2025 (0 secs)
Time.Estimated...: Mon Jan 13 13:49:51 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      150 H/s (0.55ms) @ Accel:32 Loops:512 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 1/1 (100.00%)
Rejected.........: 0/1 (0.00%)
Restore.Point....: 0/1 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4608-5000
Candidate.Engine.: Device Generator
Candidates.#1....: waka99 → waka99
Hardware.Mon.#1..: Temp: 87c Util: 47%

Started: Mon Jan 13 13:49:46 2025
Stopped: Mon Jan 13 13:49:53 2025

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

4. Hash: e5d8870e5bdd26602cab8dbe07a942c8669e56d6

Salt: tryhackme

Answer: 481616481616

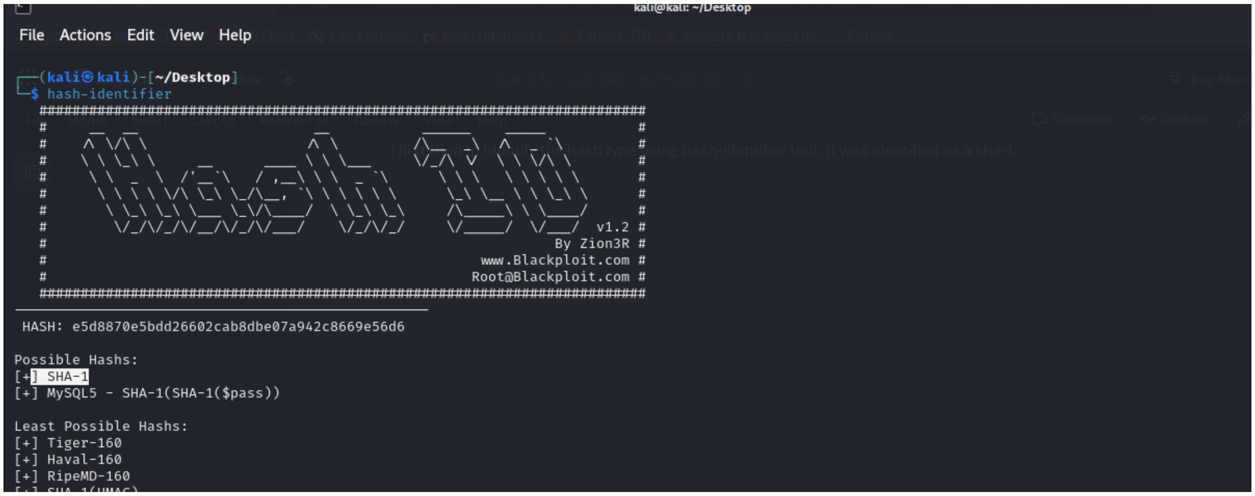


Hash: e5d8870e5bdd26602cab8dbe07a942c8669e56d6

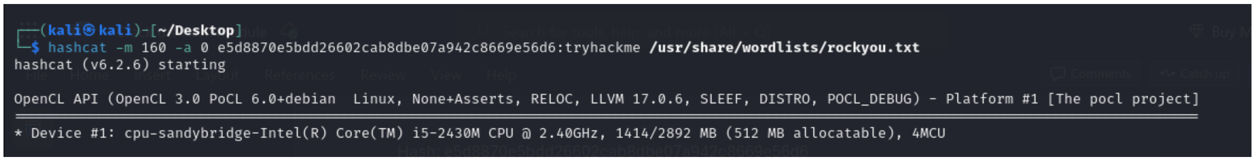
Salt: tryhackme

481616481616

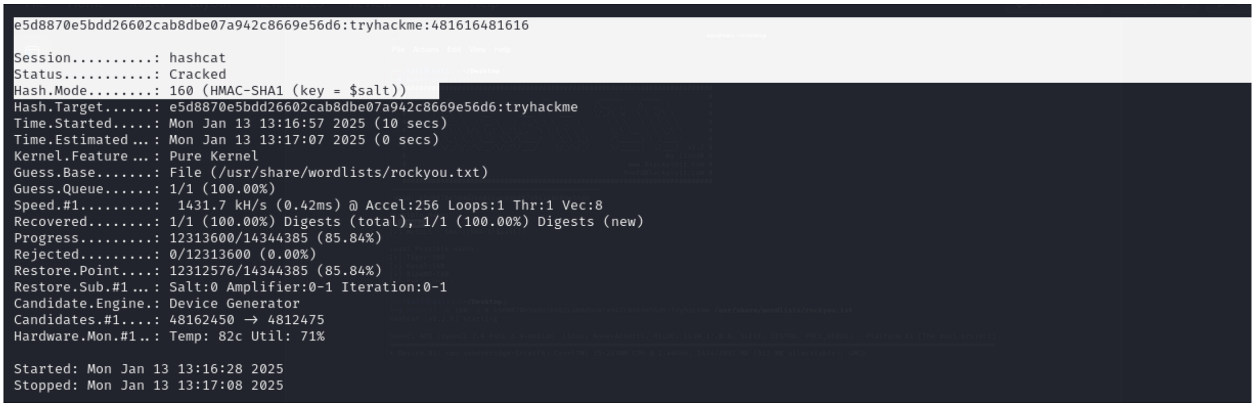
I first tried to identify the hash type using hash-identifier tool. It was identified as a sha-1.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: e5d8870e5bdd26602cab8dbe07a942c8669e56d6

Possible Hashs:
[+] SHA-1
[+] MySQL5 - SHA-1(SHA-1($pass))

Least Possible Hashs:
[+] Tiger-160
[+] Haval-160
[+] RipeMD-160
[+] SHA-1(HMAC)
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ hashcat -m 160 -a 0 e5d8870e5bdd26602cab8dbe07a942c8669e56d6:tryhackme /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 1414/2892 MB (512 MB allocatable), 4MCU
```



```
e5d8870e5bdd26602cab8dbe07a942c8669e56d6:tryhackme:481616481616

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 160 (HMAC-SHA1 (key = $salt))
Hash.Target......: e5d8870e5bdd26602cab8dbe07a942c8669e56d6:tryhackme
Time.Started.....: Mon Jan 13 13:16:57 2025 (10 secs)
Time.Estimated...: Mon Jan 13 13:17:07 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1431.7 kH/s (0.42ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 12313600/14344385 (85.84%)
Rejected.........: 0/12313600 (0.00%)
Restore.Point....: 12312576/14344385 (85.84%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 48162450 -> 4812475
Hardware.Mon.#1..: Temp: 82c Util: 71%

Started: Mon Jan 13 13:16:28 2025
Stopped: Mon Jan 13 13:17:08 2025
```

Congratulations on completing Crack the hash!!! 🎉

Conclusion

Through the process of completing Level 1 and Level 2 tasks, I was able to gain valuable hands-on experience in the field of hash cracking and password security. Each challenge required me to identify the type of hash and apply the appropriate cracking method using various tools such as **Hashcat**, **John the Ripper**, and online services like **CrackStation**.

In Level 1, I started with basic hashes like **MD5**, **SHA-1**, and **SHA-256**, and learned how to identify these hash types using tools like **hash-identifier**. By applying dictionary attacks with **rockyou.txt**, I was able to successfully crack the passwords. Additionally, for bcrypt hashes, I filtered the wordlist to 4-character words, further enhancing my skills in customizing wordlists for specific use cases.

In Level 2, I faced more complex hash types such as **SHA-512crypt** and **NTLM**, which deepened my understanding of advanced hashing algorithms. I used the **-m 1800** mode in Hashcat for **SHA-512 crypt** hashes and **-m 1000** for **NTLM**, which allowed me to efficiently crack the hashes. I also learned how important it is to choose the right tools and methods based on the hash type and available hints.