Assignment 4: Crack the hash.
Grade: 7%

Submission:
You must Edit the lab document. Do not create a new word document.

For every cracked password, include (paste) a labeled screenshot directly below the corresponding question. Each screenshot must display your system username. This assignment consists of multiple challenges. Read the document carefully. You may encounter roadblocks, so start early to allow enough time for troubleshooting.

Submit a single PDF.

Hashcat is a popular and effective password cracker widely used by both penetration testers and sysadmins as well as criminals and spies.

Passwords are no longer stored in plaintext (or shouldn't be, anyway). Instead, passwords are encrypted using a one-way function called a hash. Calculating a password like "Password1" into a hash is lightning quick. What if all you've got is the hash? A brute-force attack to reverse the hash function and recover the password could be computationally infeasible. Like, until the heat death of the universe infeasible.

Luckily, or unluckily depending on your point of view, none of us is likely to live that long, but there are many ways to reverse a hash to recover the original password without resorting to a probably fruitless brute-force attack. It turns out humans are so predictable in their password choices that hashcat can often recover a password.

Uses:
Cracking passwords has many legitimate uses besides the obvious criminal and espionage ones. A sysadmin may wish to pre-emptively check the security of user passwords. If hashcat can crack them, so can an attacker.

Penetration testers on engagement will frequently find themselves cracking stolen password hashes to move laterally inside a network, or to escalate privileges to an admin user. Since penetration testers work to find security holes on purpose, under contract, so that their customer can improve their security, this is also a perfectly legitimate use case.

The real takeaway is that both illegal attackers and legit defenders use hashcat. The best way to prevent an attacker from using hashcat against you is to test your own defenses first to make sure any such attack can't succeed.

How does hashcat work?
At its most basic level, hashcat guesses a password, hashes it, and then compares the resulting hash to the one it's trying to crack. If the hashes match, we know the password. If not, keep guessing. There are numerous attacks sort of a full brute-force attempt, including dictionary attacks, combinator attacks, mask attacks, and rule-based attacks. Hashcat can also harness the power of your GPU to brute force if you have the computing rig for it — and time to spare.

Hashcat examples
Hashcat dictionary attack: Since humans tend to use really bad passwords, a dictionary attack is the first and obvious place to start. The rockyou.txt word list is a popular option. Containing more than 14 million passwords sorted by frequency of use, it begins with common passwords such as "123456", "12345", "123456789", "password", "iloveyou", "princess", "1234567", and "rockyou", all the way to less common passwords such as "xCvBnM", "ie168", "abygurl69", "a6_123", and "*7¡Vamos!".

Many other free wordlists exist on the internet, especially targeted at specific languages. Hashcat lets you specify the wordlist of your choice.

Read Hashcat documentation for more details.

Hashcat 101

Let's say you are hacking a Linux box and all you have is a shadow.log (download this file from Blackboard for sample hashes)

ubuntu:6MOCeF0du$eCgZ9.I.hS5CDST1aQHozhLbBH6rAUj97vvW/22eaCynqLv/whZKM1freAN3n2XQiCWjDr0UFreVv0IAvI.fl0:15549:0:99999:7:::

No problem with a handy tool like hashcat and a trusty password list like rockyou.txt. You can possibly crack the password.

Required: Linux machine with Hashcat installed.

Tutorial: How To Install Hashcat
Commands:

sudo apt update
sudo apt upgrade
sudo apt install hashcat

You can for example run a benchmark to make sure everything is working properly:

hashcat -b

1. Copy the hash (let us crack, say the manager account password).

Copy the hash that you want to find the password into a new file, for example, user.hash
For the manager, I copied the following into my user.hash file.

6MOCeF0du$eCgZ9.I.hS5CDST1aQHozhLbBH6rAUj97vvW/22eaCynqLv/whZKM1freAN3n2XQiCWjDr0UFreVv0IAvI.fl0

2. Download rockyou.txt file and save it into the same folder as user.hash. Alternatively, you can use newrockyou.txt for faster execution.

3. Then execute.
   hashcat –a 0 user.hash rockyou.txt –m 1800

Ensure all the files are in the same folder and the commands are executed from the same folder. I recommend downloading rockyou.txt from the web.

Where:

-a 0	> use dictionary attack
user.hash	> file that contains hash to be cracked
rockyou.txt	> file containing password list
-m 1800	> Unix type 6 password hashes

You can also execute the command as (if above command gives you error)
hashcat –m 1800 –a 0 user.hash rockyou.txt

```
robinchataut@ubunturobin:~/Desktop/hash$ hashcat -m 1800 -a 0 user.hash rockyou.txt
hashcat (v6.2.5) starting
```

3. It takes time. Do some stretching or drink a coffee 🙄
Once it is done you can actually see the cracked password besides the hash like in below screenshot

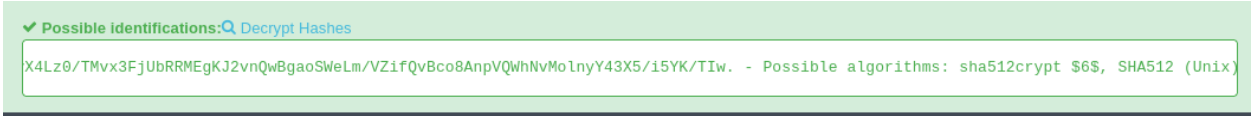
```
Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

$6$SaltVal2$ybuPu7Nmo9LKn0p0ozhFhFw2SS2cqkLsx8c5OEAWFkIJjtXBEJqxUQzLh900QMgFTGiw6YuFDueNAapfLKt0f1:forgot

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target......: $6$SaltVal2$ybuPu7Nmo9LKn0p0ozhFhFw2SS2cqkLsx8c5OEA...LKt0f1
Time.Started.....: Mon Sep  4 03:59:46 2023 (1 sec)
Time.Estimated...: Mon Sep  4 03:59:47 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     2289 H/s (5.29ms) @ Accel:256 Loops:256 Thr:1 Vec:2
Recovered........: 1/1 (100.00%) Digests
Progress.........: 3072/14344384 (0.02%)
Rejected.........: 0/3072 (0.00%)
Restore.Point....: 2816/14344384 (0.02%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4864-5000
Candidate.Engine.: Device Generator
Candidates.#1....: pirate -> dangerous
```

Congrats on cracking your first password Mr. Hacker!

Challenge 1 (20 Points):

1. Try to find passwords for other users in the given files. You must find at least one other passwords. Attach screenshot of cracked password. Screenshot must show your username.

Going to do Sean in the shadow file.




```
┌──(kali㉿kali)-[~/Desktop/Assignment4 Cybersec]
└─$ hashcat -m 1800 -a 0 sean.hash 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-penryn-AMD Ryzen 7 7800X3D 8-Core Processor, 3825/7714 MB (1024 MB allocatable), 3MCU

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:
* Zero-Byte
* Single-Hash
* Single-Salt
* Uses-64-Bit

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 90c

Initializing backend runtime for device #1. Please be patient…
Host memory required for this attack: 0 MB



Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 271
* Bytes.....: 2121
* Keyspace..: 271

$6$SaltVal4$rIpTjZrVyyX4Lz0/TMvx3FjUbRRMEgKJ2vnQwBgaoSWeLm/VZifQvBco8AnpVQWhNvMolnyY43X5/i5YK/TIw.:spectre

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target......: $6$SaltVal4$rIpTjZrVyyX4Lz0/TMvx3FjUbRRMEgKJ2vnQwBg ... K/TIw.
Time.Started.....: Mon Feb 24 20:48:21 2025 (0 secs)
Time.Estimated...: Mon Feb 24 20:48:21 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     2001 H/s (5.88ms) @ Accel:64 Loops:1024 Thr:1 Vec:2
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 192/271 (70.85%)
Rejected.........: 0/192 (0.00%)
Restore.Point....: 128/271 (47.23%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4096-5000
Candidate.Engine.: Device Generator
Candidates.#1....: monica → 00000
Hardware.Mon.#1..: Util: 10%

Started: Mon Feb 24 20:48:20 2025
Stopped: Mon Feb 24 20:48:23 2025

┌──(kali㉿kali)-[~/Desktop/Assignment4 Cybersec]
```

agmorales117

Cracked password: spectre

2. What other credentials (username:password) could have been used to gain access also have SUDO privileges? Refer to shadow.log and sudoers.log.

- From the sudoers.log file, the following users have sudo privileges:
- guest
- manager
- sean
- roger
- pierce

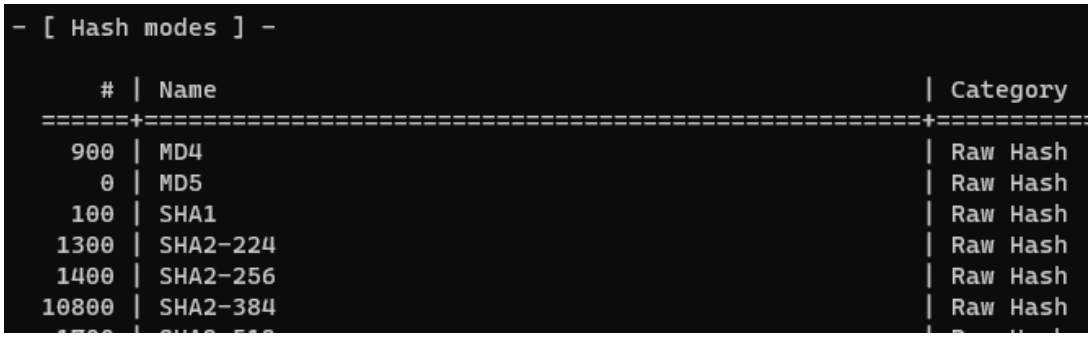
guest:6SaltVal1$...
manager:6SaltVal2$...
sean:6SaltVal4$...
roger:6SaltVal6$...
pierce:6SaltVal8$...

all of the above ones from the shadow file can be cracked and use sudo privileges

Challenge 2 (50 Points):

Find the hash mode and password for following hashes. Note that the hash type for the hashes might be different.

Hashcat supports hundreds of hashes, and the chosen hash mode needs to be stated for hashcat to know what to attack. The modes can be found using **hashcat --help** (note: hashcat cannot attack multiple hash types in a single session but there are other tools that can). Recent versions of hashcat also support hash auto-detection if **-m <$mode>** isn't passed, however as several hash types are constructed similarly, false positives can occur and therefore it isn't encouraged.
For example, MD5 would be **-m 0**, SHA1 would be **-m 100** and so on.



Find a way/tool to identify hash type before you proceed. Each cracked password is 5 points. Attach a screenshot of each password you cracked. Screenshot must show your username. See the sample below.

I used this to identify the hash
https://hashes.com/en/tools/hash_identifier

a. eb61eead90e3b899c6bcbe27ac581660

MD5 (mode 0)

Password: HELLO

b. 48bb6e862e54f2a795ffc4e541caed4d

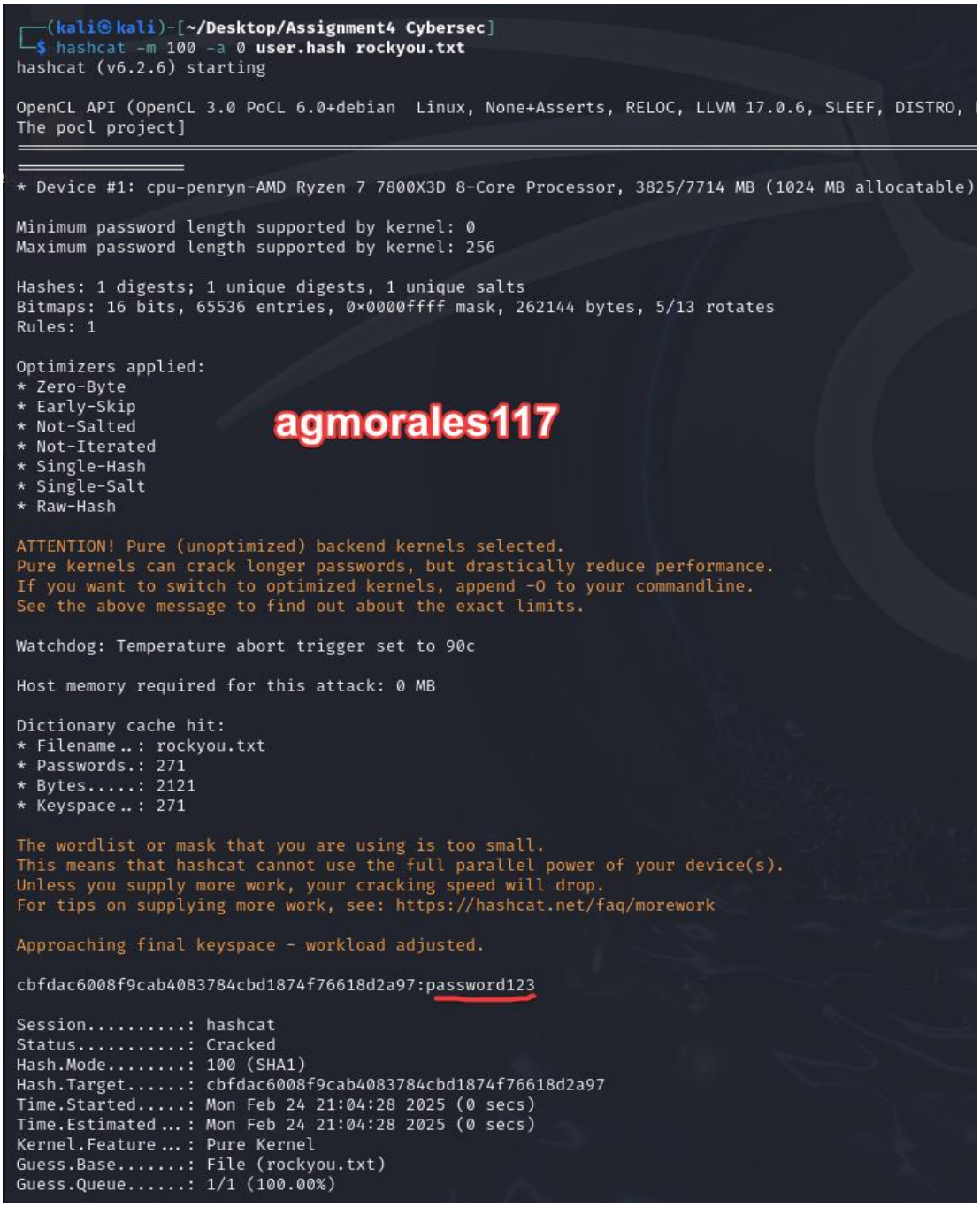
MD5 (mode 0)

```
┌──(kali㉿kali)-[~/Desktop/Assignment4 Cybersec]
└─$ hashcat -m 0 -a 0 user.hash 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) - Pl
The pocl project]
═══════════════════════════════════════════════════════════════════════════════
* Device #1: cpu-penryn-AMD Ryzen 7 7800X3D 8-Core Processor, 3825/7714 MB (1024 MB allocatable), 3MCU

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:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* 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 90c

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 271
* Bytes.....: 2121
* Keyspace..: 271

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.

48bb6e862e54f2a795ffc4e541caed4d:easy

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 48bb6e862e54f2a795ffc4e541caed4d
Time.Started.....: Mon Feb 24 21:02:01 2025 (0 secs)
Time.Estimated...: Mon Feb 24 21:02:01 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   362.3 kH/s (0.12ms) @ Accel:512 Loops:1 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 271/271 (100.00%)
Rejected.........: 0/271 (0.00%)
Restore.Point....: 0/271 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456 → celtic
Hardware.Mon.#1..: Util:  6%
```


Password: easy

c. CBFDAC6008F9CAB4083784CBD1874F76618D2A97

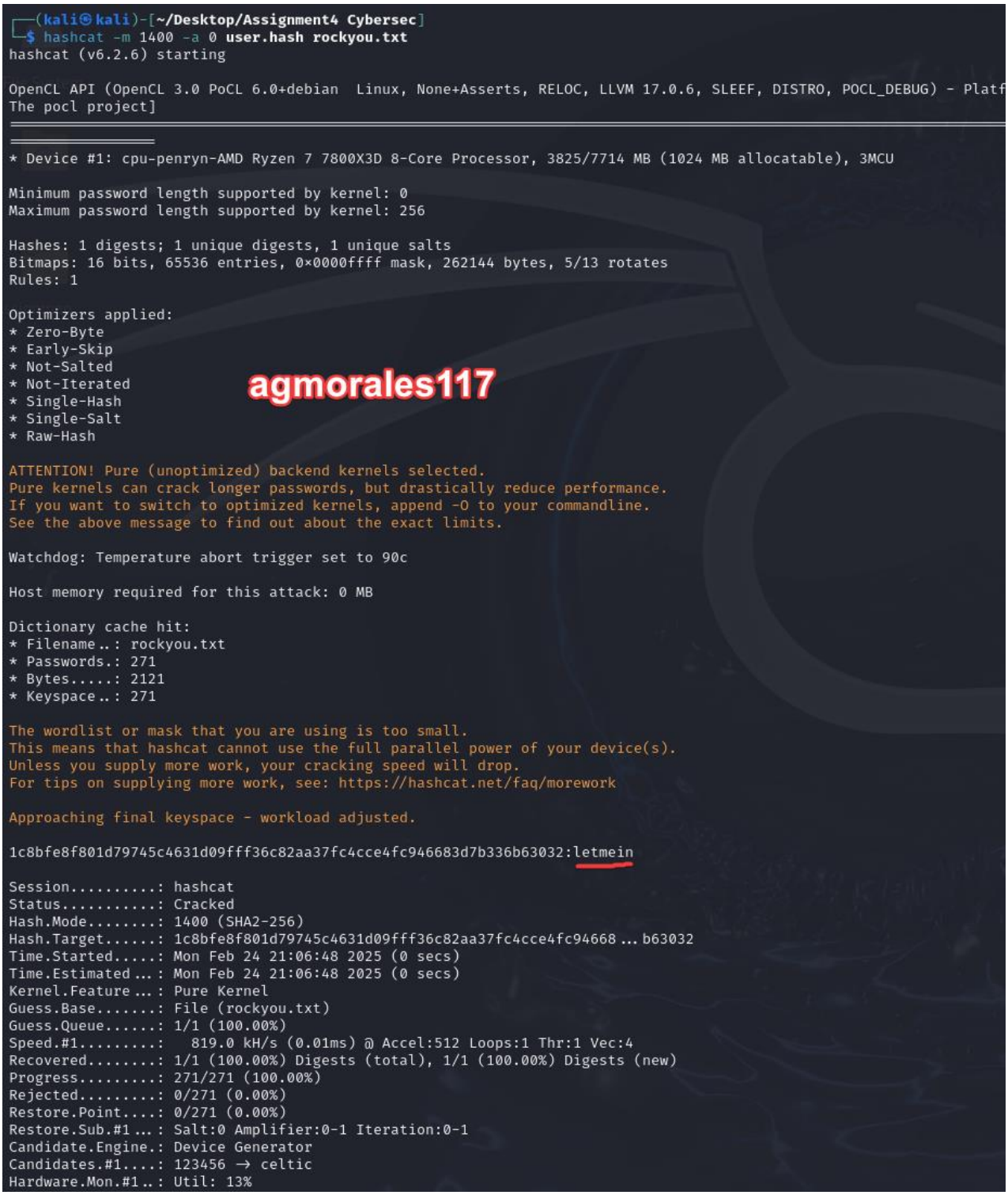
SHA1 (mode100)



Password: password123

d. 1C8BFE8F801D79745C4631D09FFF36C82AA37FC4CCE4FC946683D7B336B63032

SHA256(mode 1400)



password: letmein

e. $2y$12$Dwt1BZj6pcyc3Dy1FWZ5ieeUznr71EeNkJkUlypTsgbX1H68wsRom

Possible algorithms: bcrypt $2*$, Blowfish (Unix) which is mode 3200 for hashcat

Password: bleh

f. 279412f945939ba78ce0758d3fd83daa

MD4 (mode 900)

Password: Eternity22

g. F09EDCB1FCEFC6DFB23DC3505A882655FF77375ED8AA2D1C13F640FCCC2D0C85

SHA256 (mode 1400 for hashcat)

Password: paule

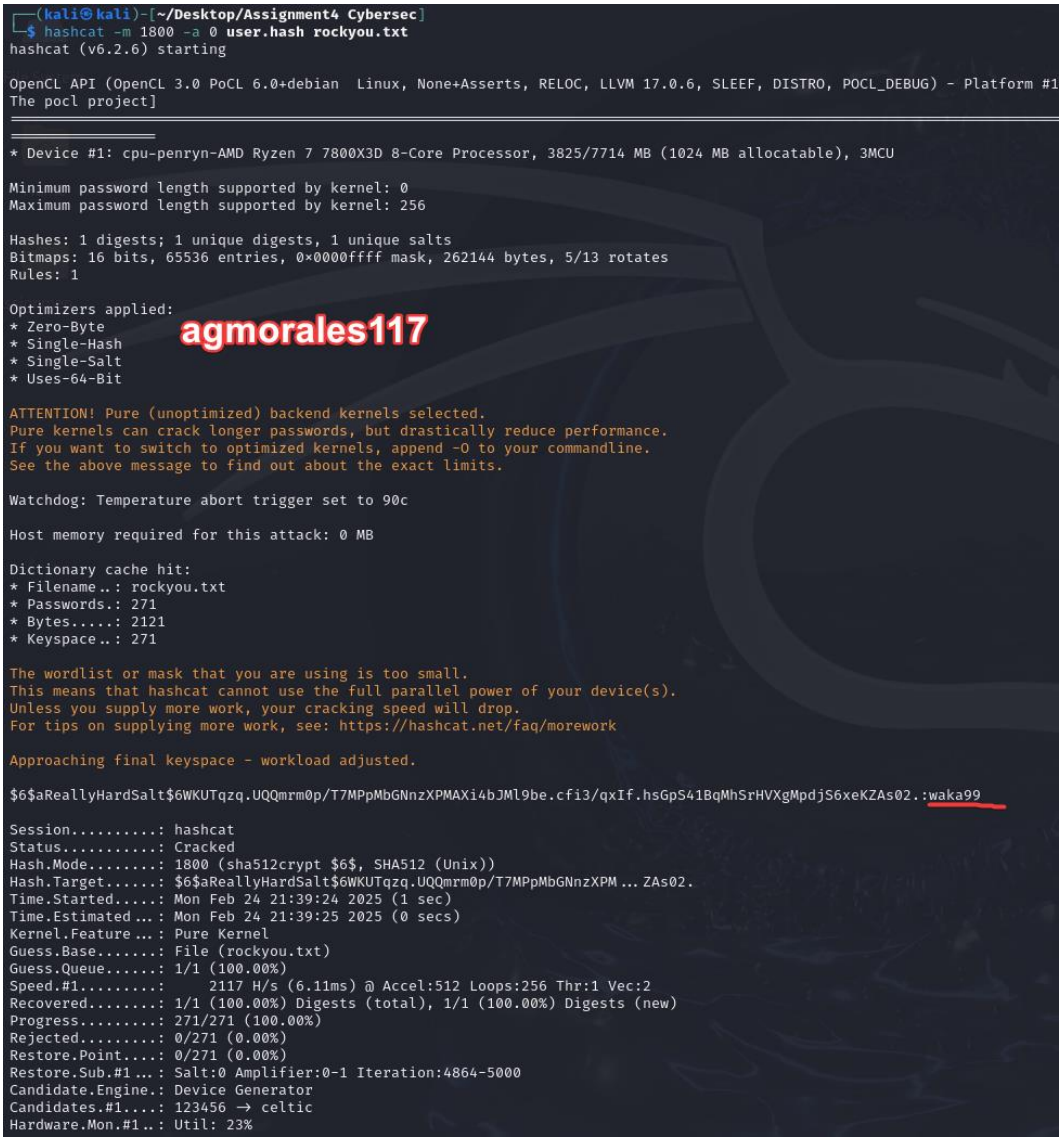
h. 1DFECA0C002AE40B8619ECF94819CC1B

NTLM (mode 1000). Needed to use the entire rockyou txt file from google. Wasn't in the newrockyou.txt word list you gave us.

Password: n63umy8lkf4i

i. 6aReallyHardSalt$6WKUTqzq.UQQmrm0p/T7MPpMbGNnzXPMAXi4bJMl9be.cfi3/qxIf.hsGpS41BqMhSrHVXgMpdjS6xeKZAs02.

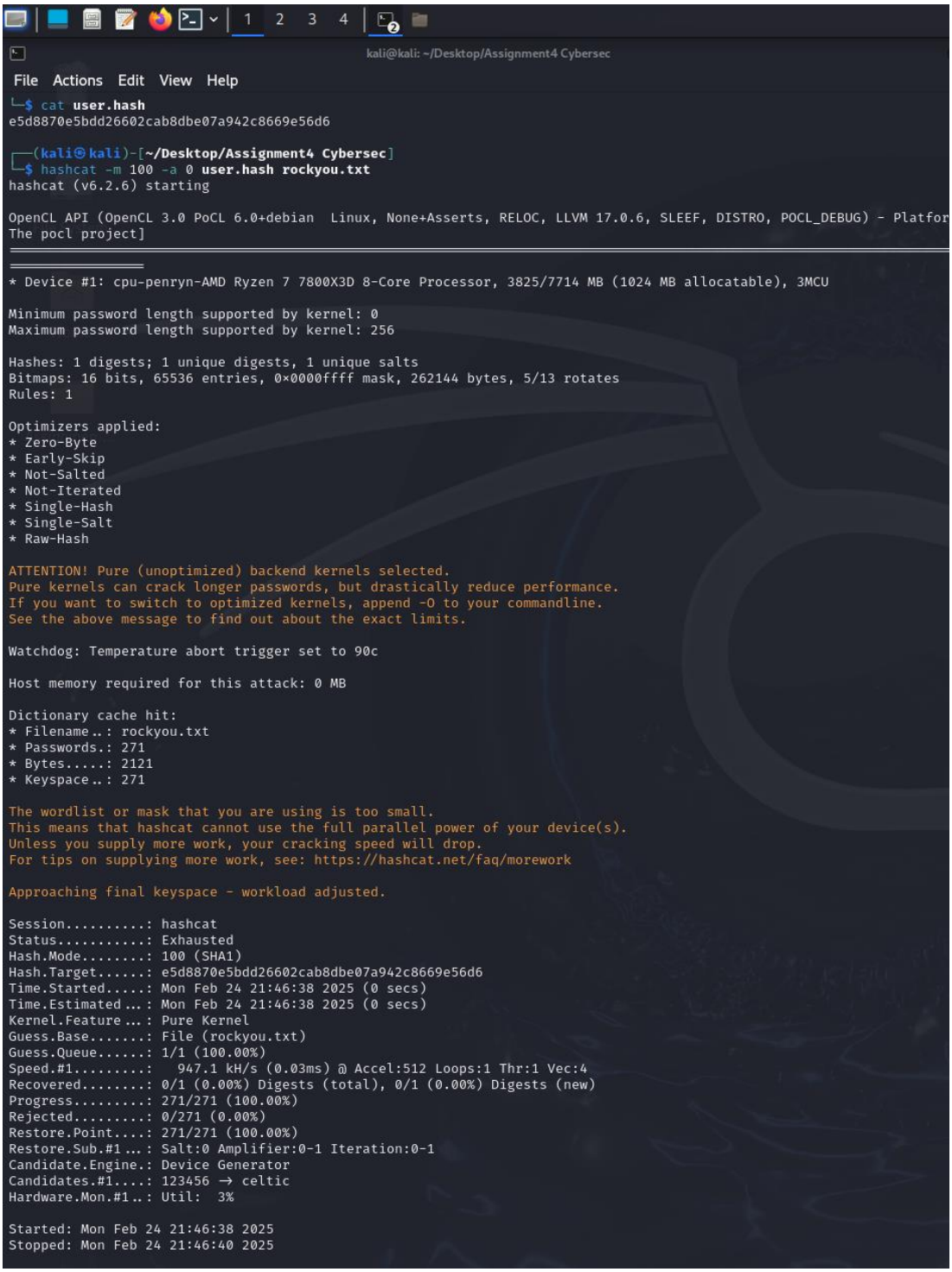
sha512crypt 6, SHA512 (Unix). which is 1800 for hashcat mode



password: waka99

j. e5d8870e5bdd26602cab8dbe07a942c8669e56d6

SHA1 encryption which is mode 100 in hashcat, couldn't crack this password.



Challenge 3 (20 Points):

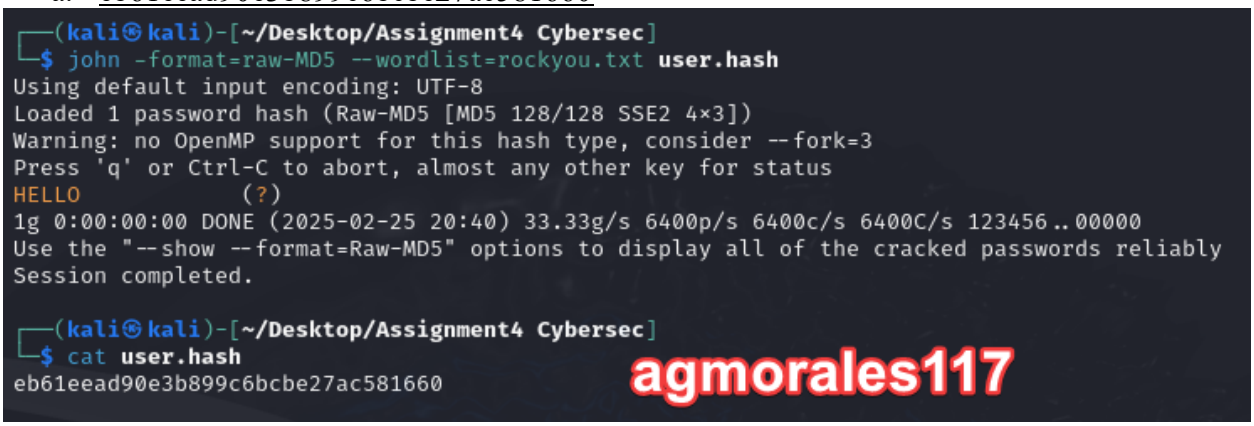
Now crack the first two hashes in Challenge 2 using John the Ripper. John the Ripper is a password-cracking tool used to identify weak passwords by trying different combinations. It's used

for security testing, including dictionary and brute-force attacks, and can crack various password hash formats. Its primary purpose is to assess and strengthen password security.

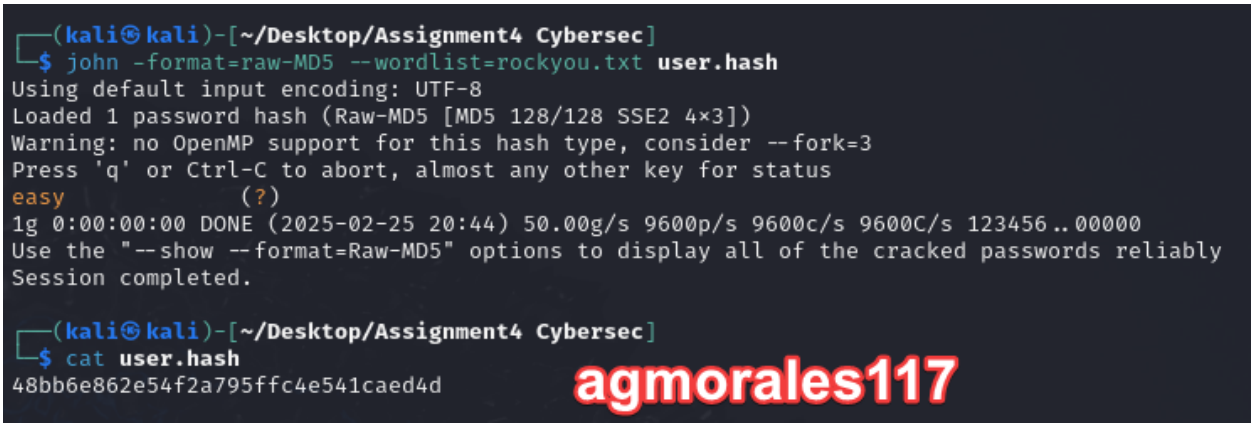
Attach a screenshot of at first **two** passwords (from challenge 2) you cracked using John the Ripper.

First two hashes:

a. eb61eead90e3b899c6bcbe27ac581660



b. 48bb6e862e54f2a795ffc4e541caed4d



Installation:

sudo apt install snapd

sudo snap install john-the-ripper

See Sample run for (b):

Note: You can always use newrockyou.txt for faster execution.

hash: CBFDAC6008F9CAB4083784CBD1874F76618D2A97

hash type: SHA1

hashcat -m 100 user.hash rockyou.txt

john -format=raw-sha1 --wordlist=rockyou.txt user.hash

```
robinchataut@ubunturobin:~/Desktop/hash$ hashcat -m 100 user.hash rockyou.txt
hashcat (v6.2.5) starting
```

```
Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

cbfdac6008f9cab4083784cbd1874f76618d2a97:password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 100 (SHA1)
Hash.Target......: cbfdac6008f9cab4083784cbd1874f76618d2a97
Time.Started.....: Fri Sep 15 01:48:07 2023 (0 secs)
Time.Estimated...: Fri Sep 15 01:48:07 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  3292.8 kH/s (0.09ms) @ Accel:256 Loops:1 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2048/14344384 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 1024/14344384 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: kucing -> lovers1

Started: Fri Sep 15 01:47:59 2023
Stopped: Fri Sep 15 01:48:09 2023
robinchataut@ubunturobin:~/Desktop/hash$
```

```
robinchataut@ubunturobin:~/Desktop/hash$ john -format=raw-sha1 --wordlist=rockyou.txt user.hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 ASIMD 4x])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2023-09-15 01:54) 100.0g/s 138400p/s 138400c/s 138400C/s liberty..password123
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.
robinchataut@ubunturobin:~/Desktop/hash$ locate john.pot
/home/robinchataut/snap/john-the-ripper/584/.john/john.pot
robinchataut@ubunturobin:~/Desktop/hash$ cat /home/robinchataut/snap/john-the-ripper/584/.john/john.pot

$dynamic_26$cbfdac6008f9cab4083784cbd1874f76618d2a97:password123
```

Questions (10 Points):

1. After using both tools, what key differences did you observe in their functionality, speed, and ease of use? Which tool do you prefer for password cracking tasks, and why? Provide a brief explanation to justify your choice.

From cracking the various different hashes provided, the commands and format methods of hashcat was a lot easier to understand what to format and the table in the –help proved to be very useful. I did find john the ripper to be a quicker tool at cracking the hash than hashcat. Hashcat provided a lot of information after cracking while john kept it a lot shorter and had a lot less details than hashcat. Hashcat is useful for easy to type commands cracking solution that provides a lot of information while john keeps it quick and just gives you the password cracked.

2. Password cracking tools are often associated with malicious activities, but they also serve crucial legitimate purposes in cybersecurity. Identify two legitimate use cases where password cracking is ethically and legally used. Briefly explain how and why password cracking is valuable in these scenarios.

The first legitimate use case for password cracking would be for the main thing that would come to mind for this course. Penetration testing a work environment to find the different ways to access employee logins would be a great way to test employee passwords for strength and see how long it would take for a malicious attacker to get in. This would help find the weaker passwords and require users to change them to be more complex. The second situation which password cracking can be applied is for account recovery. I think a good example is needing to get into an account of a deceased family member in order to get legal documents would be necessary. If someone forgets their password in a work place or law enforcement requires a file to be unencrypted, password cracking would be very useful for that.

References:

- https://www.csoonline.com/profile/j-m-porup/, Created by JM Porup
- https://cybersecurityfreeresource.wordpress.com/2022/03/31/hashcat-101-cracking-password-hashes/
- https://hashcat.net/wiki/doku.php?id=combinator_attack