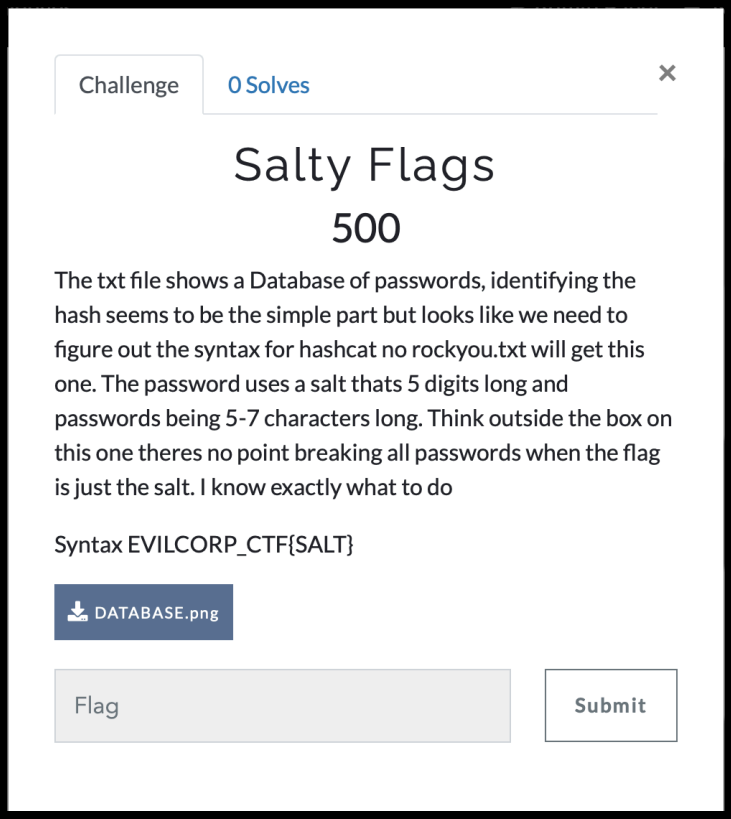
MR. ROBOT

1. This challenge is using hashcat but in a different way this time. No wordlist is needed and the flag is only the SALT part of the password. Lets break down this challenge into parts to get a more understanding.
    1. Salt is 5 digits long so every password has 5 digits at the start and that is what's needed for the flag.
    2. All passwords are 5-7 characters long so the password length is between including the salt 10-12 characters in length.



2. Next let's download the DATABASE.png and look at what's going on here. Okay lets break down what's going on with these hash passwords and look at the information that's provided for us to get the flag.
    1. Passwords changed after 01/01/2020 are alphanumeric 5-7 characters long
    2. Passwords that have been changed before this date only contain digits .
    3. All passwords have a salt that is digits only and 5 characters.

DATABASE.png

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* WE NEED YOU'RE HELP TO HELP RETRIEVE PASSWORDS, BELOW IS A DUMP OF THE DATABASE. WE CAN'T CRACK THEM WE'VE TRIED RAINBOW TABLES   *
* TO TRY CRACK THEM AND NO LUCK. THE PASSWORDS ARE SALTED PLEASE HELP US                                                          *
*                                                                                                                                  *
* PASSWORDS THAT HAVE BEEN CHANGED AFTER 2020 JANUARY 1ST HAVE BEEN CHANGED TO HAVING ALPHANUMERIC AND BEING 5-7 CHARACTERS IN      *
* LENGTH.                                                                                                                          *
* PASSWORDS THAT HAVE BEEN CHANGED BEFORE THIS DATE ONLY CONTAIN OF DIGITS AND ARE 5-7 CHARACTERS IN LENGTH.                        *
*                                                                                                                                  *
* ALL PASSWORDS CONTAIN THE SAME SALT WHICH THE SALT IS 5 DIGITS IN LENGTH.                                                        *
*                                                                                                                                  *
* CAPTURE CODE HAS REVEALED PASSWORDS GET STORED LIKE hash($salt,$pass)                                                            *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Join Date	Name	Hash	Password Changed
1988-06-17	Terry Colby	391EF481FC5A31BB4CB5BD4F860E4333C54302EB	2020-08-12
1993-02-02	Phillip Price	0892B289CE18E0CF6EED997C5E126C5E4AE7D9C7	2021-01-24
2010-05-07	Tyrell wellick	ED3DF713E7284670F4135FEBE6E97160C96D7DBA	2019-01-07
1988-06-17	Scott Knowles	34DFB0B054C6A1AAF3929CA8266ABB7A646FAC4C	2020-08-01

3. So after breaking down what's going on in the database, first I realised that I don't have to crack every single password. The easiest password to crack in this is the digits only password which is Tyrell Wellicks password and I know its 10-12 digits long due to his password not being modified after 01/01/2020. Below is an example of a hashcat command brute forcing a password without using a wordlist.

```
- [ Basic Examples ] -

Attack-Mode        | Hash-Type | Example command
===================+=========+=================================================================
Wordlist           | $P$       | hashcat -a 0 -m 400 example400.hash example.dict
Wordlist + Rules   | MD5       | hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
Brute-Force        | MD5       | hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?a
Combinator         | MD5       | hashcat -a 1 -m 0 example0.hash example.dict example.dict
```

4. So we've broke down the command above before but notice that instead of the dictionary wordlist there is ?a?a?a?a which will be looked at in a future step. For now lets get the attack mode which -a 3 will be used for brute force.

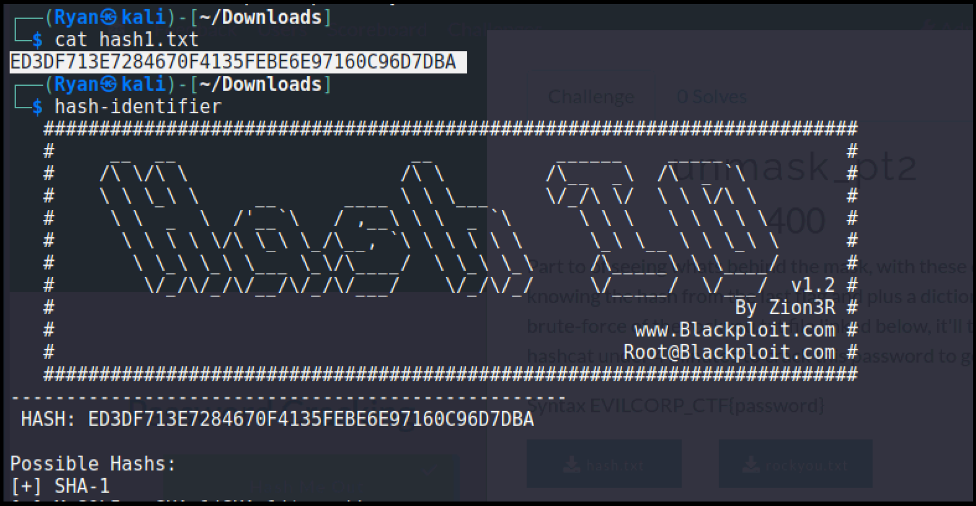
```
- [ Attack Modes ] -

  # | Mode
 ===+======
  0 | Straight
  1 | Combination
  3 | Brute-force
  6 | Hybrid Wordlist + Mask
  7 | Hybrid Mask + Wordlist
```

5. Looking through the hashcat -h, the -I option can be used to set increments of the length of the password and from the information the passwords have a min length of 10 and a max length of 12 which will be used in the command.

```
-i, --increment                 |      | Enable mask increment mode                  |
    --increment-min             | Num  | Start mask incrementing at X                | --increment-min=4
    --increment-max             | Num  | Stop mask incrementing at X                 | --increment-max=8
```

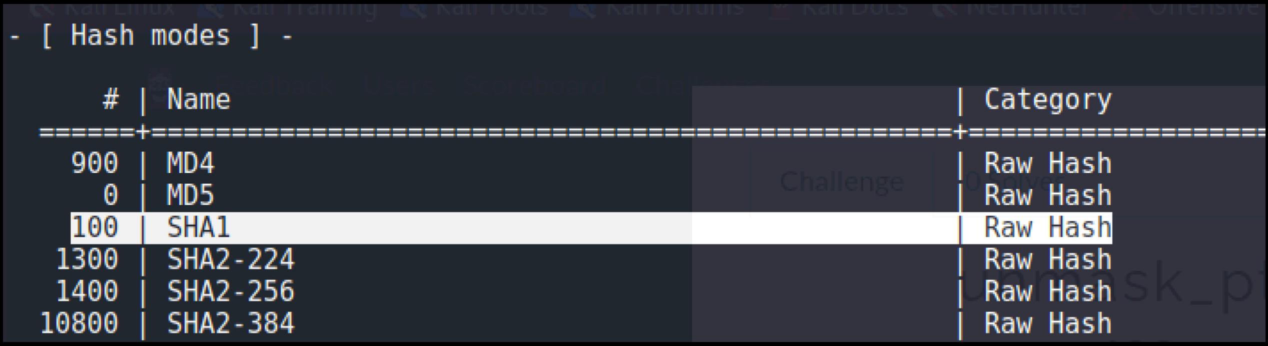
6. Next is to determine the hash mode but first identifying the hash is important, just like what was done in unmask.pt1 challenge, put Tyrell Wellicks hash password in a txt file and use google tool or hash-identifier to identify the hash. The hash identified is a SHA-1 so now use hashcat -h to find the hash mode.



```
┌──(Ryan㉿kali)-[~/Downloads]
└─$ cat hash1.txt
ED3DF713E7284670F4135FEBE6E97160C96D7DBA
┌──(Ryan㉿kali)-[~/Downloads]
└─$ hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: ED3DF713E7284670F4135FEBE6E97160C96D7DBA

Possible Hashs:
[+] SHA-1
```

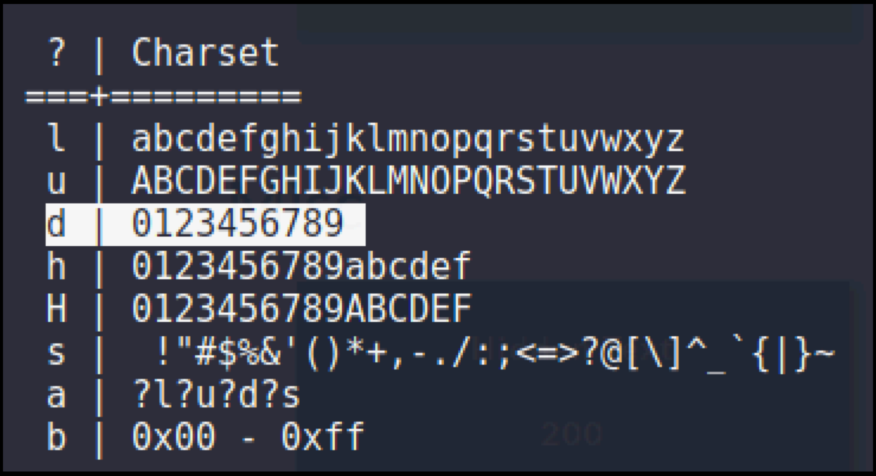
7. Hash mode for SHA-1 is 100 so -m 100 will be used



```
- [ Hash modes ] -

      # | Name                                               | Category
  ======+====================================================+==================
    900 | MD4                                                | Raw Hash
      0 | MD5                                                | Raw Hash
    100 | SHA1                                               | Raw Hash
   1300 | SHA2-224                                           | Raw Hash
   1400 | SHA2-256                                           | Raw Hash
  10800 | SHA2-384                                           | Raw Hash
```

8. Hashcat has a great feature but can take longer is when you can use charset modes to guess a character in the password. Look at the screenshot below notice how ?d means only digits and because Tyrell Wellicks password is only digits this is what is needed. The password is 12 digits in length max so 12 ?d will be used.



```
  ? | Charset
 ===+=========
  l | abcdefghijklmnopqrstuvwxyz
  u | ABCDEFGHIJKLMNOPQRSTUVWXYZ
  d | 0123456789
  h | 0123456789abcdef
  H | 0123456789ABCDEF
  s |  !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
  a | ?l?u?d?s
  b | 0x00 - 0xff
```

8. Lets put all these options together and run the command.

```
┌──(Ryan㉿kali)-[~/Downloads]
└─$ hashcat -m 100 -a 3 -i --increment-min=10 --increment-max=12 hash1.txt ?d?d?d?d?d?d?d?d?d?d?d?d --force
```

9. Password takes a few minutes to crack so be patient like before, if cracked you should see what is shown below. As you can see Tyrell Wellicks password is 0032548672, and the flag only requires the salt and the salt of the password is the first 5 digits of the password so the flag is 00325

```
ed3df713e7284670f4135febe6e97160c96d7dba:0032548672

Session..........: hashcat
Status...........: Cracked
Hash.Name........: SHA1
Hash.Target......: ed3df713e7284670f4135febe6e97160c96d7dba
Time.Started.....: Fri Apr  9 12:33:24 2021, (2 mins, 19 secs)
Time.Estimated...: Fri Apr  9 12:35:43 2021, (0 secs)
Guess.Mask.......: ?d?d?d?d?d?d?d?d?d?d [10]
Guess.Queue......: 1/3 (33.33%)
Speed.#1.........: 40414.6 kH/s (12.32ms) @ Accel:512 Loops:1000 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 5748224000/10000000000 (57.48%)
Rejected.........: 0/5748224000 (0.00%)
Restore.Point....: 5747712/10000000 (57.48%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1000 Iteration:0-1000
Candidates.#1....: 1232038672 -> 6880259262
```