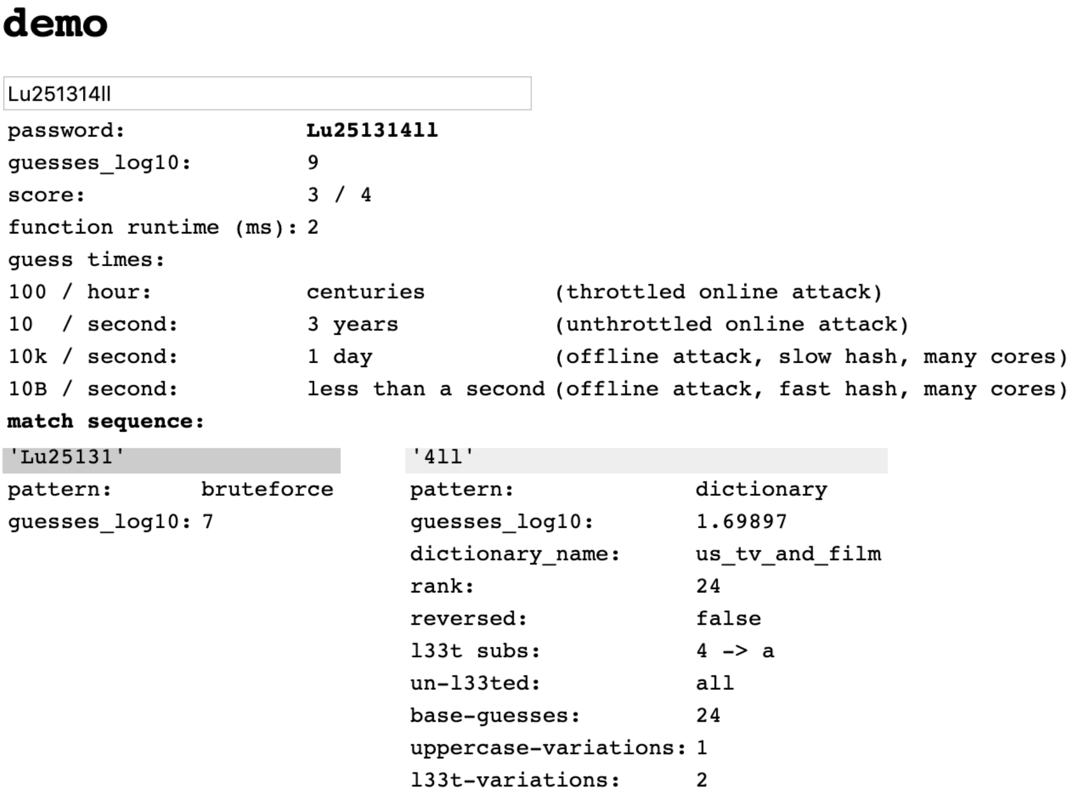
Lab1

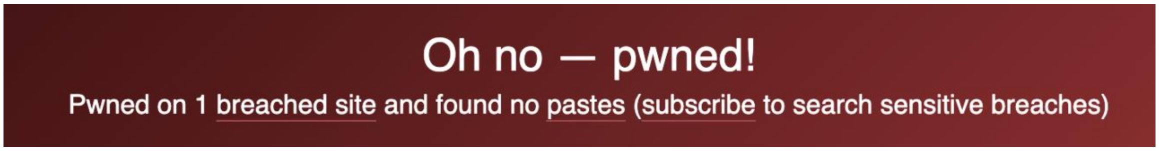
Task 1: Test Password Security 1. Visit the following URL: https://lowe.github.io/tryzxcvbn/



demo

```
Lu251314ll
password:                Lu251314ll
guesses_log10:           9
score:                   3 / 4
function runtime (ms):   2
guess times:
100 / hour:              centuries          (throttled online attack)
10  / second:            3 years            (unthrottled online attack)
10k / second:            1 day              (offline attack, slow hash, many cores)
10B / second:            less than a second (offline attack, fast hash, many cores)
match sequence:
'Lu25131'                      '4ll'
pattern:       bruteforce      pattern:                dictionary
guesses_log10: 7               guesses_log10:          1.69897
                               dictionary_name:        us_tv_and_film
                               rank:                   24
                               reversed:               false
                               l33t subs:              4 -> a
                               un-l33ted:              all
                               base-guesses:           24
                               uppercase-variations:   1
                               l33t-variations:        2
```

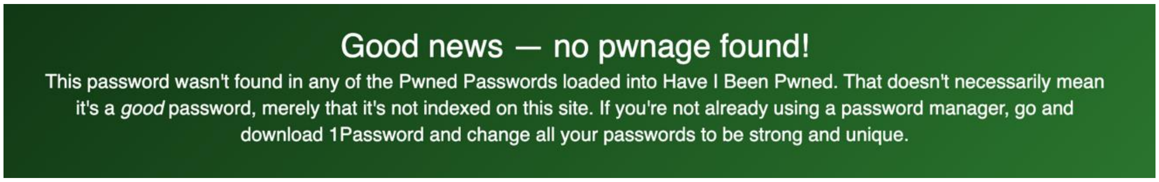
Task 2: Check an Account for a Prior Data Breach 1. Check to see if one of your online accounts has already been breached. Visit: https://haveibeenpwned.com. Type in one of your email accounts or usernames to see if it has already been compromised in a data breach.



Oh no — pwned!

Pwned on 1 breached site and found no pastes (subscribe to search sensitive breaches)

2. Next visit: https://haveibeenpwned.com/Passwords Try out some passwords to see if they have already been compromised in a data breach.



Good news — no pwnage found!

This password wasn't found in any of the Pwned Passwords loaded into Have I Been Pwned. That doesn't necessarily mean it's a *good* password, merely that it's not indexed on this site. If you're not already using a password manager, go and download 1Password and change all your passwords to be strong and unique.

3. Finally, visit: https://haveibeenpwned.com/NotifyMe Sign up to be notified when one of your accounts is breached in the future.

Notify me

You've subscribed to notifications pending email verification

Task 3

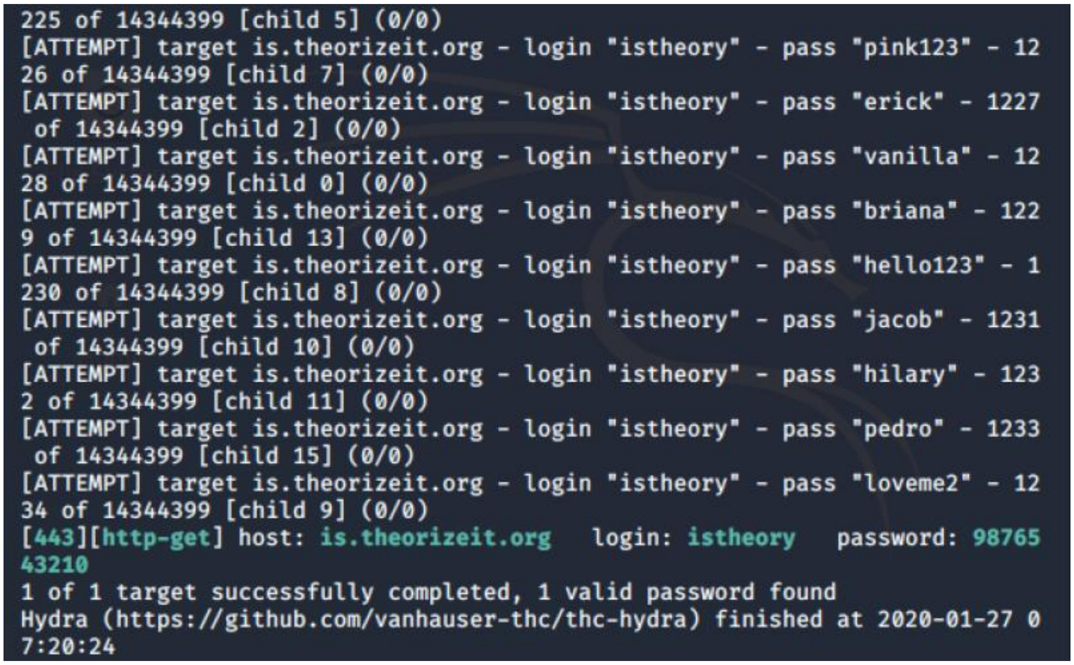
How long did it take for the password to be cracked? Record those times here: User1:______

How long did it take for the password to be cracked? Record those times here: User2:______

How long did it take for the password to be cracked? Record those times here: User3:______

Question: Did you notice a correlation between the times it took to crack a password versus the complexity of the password? What did you learn in this exercise?

Task4



```
225 of 14344399 [child 5] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "pink123" - 12
26 of 14344399 [child 7] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "erick" - 1227
 of 14344399 [child 2] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "vanilla" - 12
28 of 14344399 [child 0] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "briana" - 122
9 of 14344399 [child 13] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "hello123" - 1
230 of 14344399 [child 8] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "jacob" - 1231
 of 14344399 [child 10] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "hilary" - 123
2 of 14344399 [child 11] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "pedro" - 1233
 of 14344399 [child 15] (0/0)
[ATTEMPT] target is.theorizeit.org - login "istheory" - pass "loveme2" - 12
34 of 14344399 [child 9] (0/0)
[443][http-get] host: is.theorizeit.org   login: istheory   password: 98765
43210
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-01-27 0
7:20:24
```

Question: What was the password (Scan the results to find the line beginning with [443][httpget])?

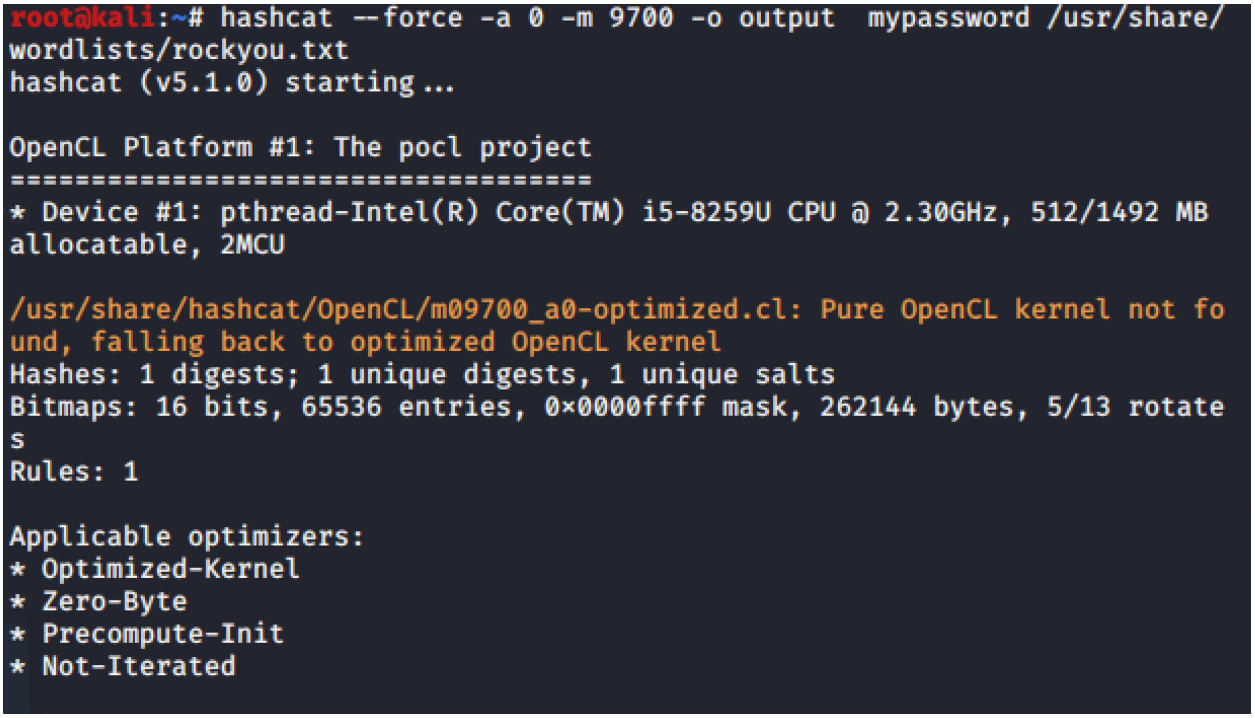
9876543210

Question: Approximately how many passwords a second were you able to try? Hint: You may need to calculate this from the start and end time

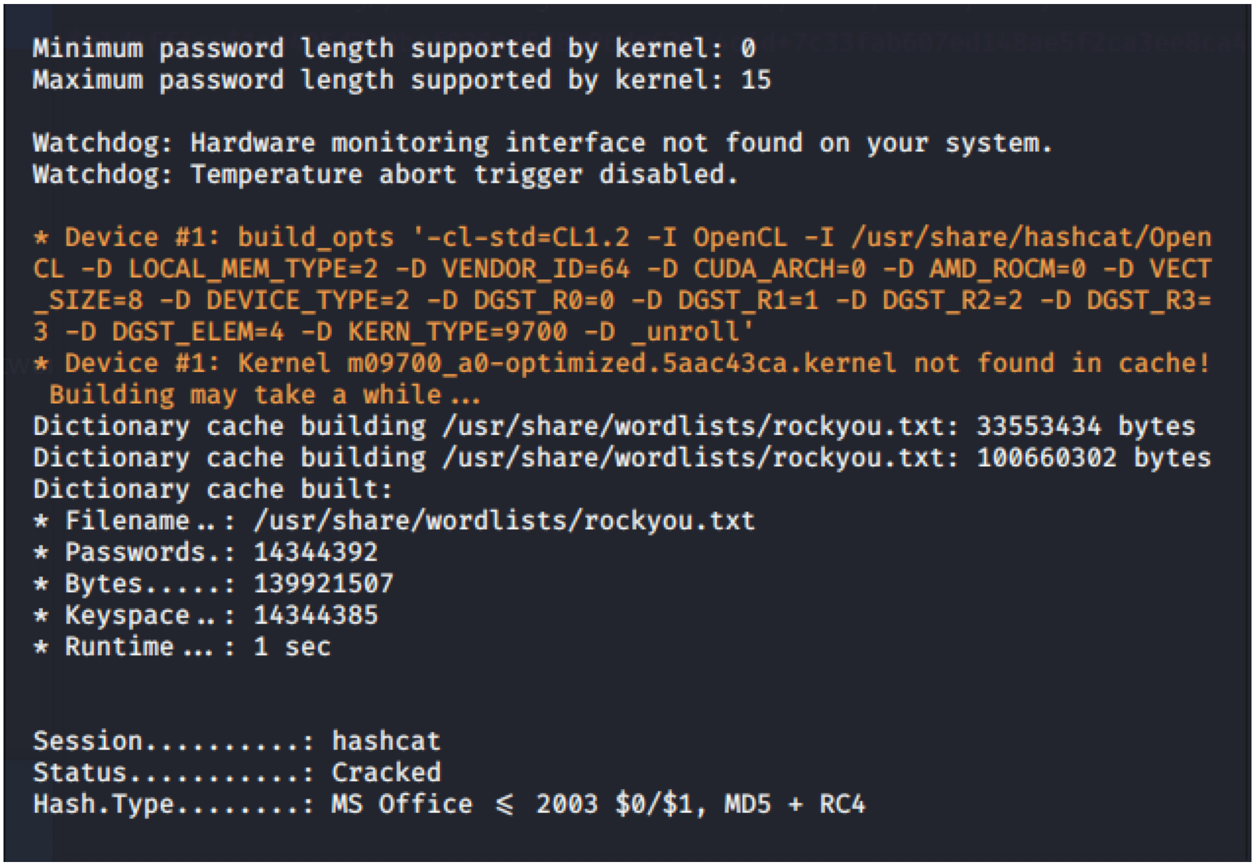
along with number of guesses made. (You can look at sample output from a `hydra` run, and determine how many passwords were tried per second in the sample output.)
It starts at 7:20:04 and ends at 7:20:24, it runs 1235 time in 20s, by calculating, I tried 61 times per second.

Task 5:

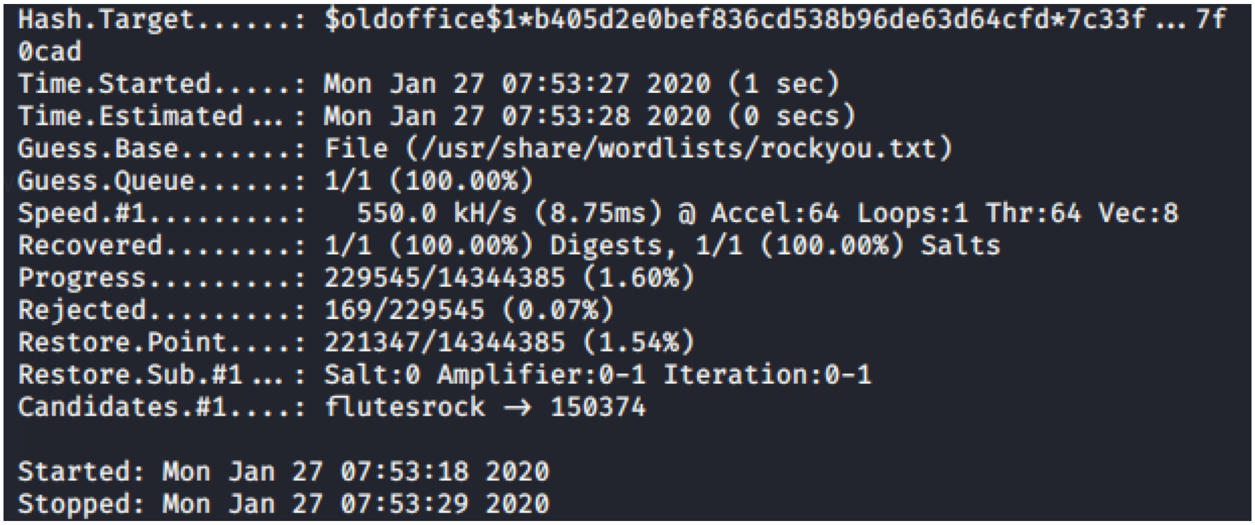
Running for hashcat.doc



```
root@kali:~# hashcat --force -a 0 -m 9700 -o output  mypassword /usr/share/
wordlists/rockyou.txt
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Core(TM) i5-8259U CPU @ 2.30GHz, 512/1492 MB
allocatable, 2MCU

/usr/share/hashcat/OpenCL/m09700_a0-optimized.cl: Pure OpenCL kernel not fo
und, falling back to optimized OpenCL kernel
Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotate
s
Rules: 1

Applicable optimizers:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Not-Iterated
```

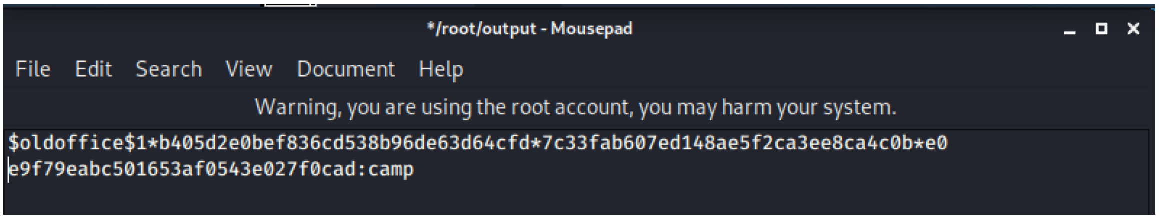


```
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 15

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

* Device #1: build_opts '-cl-std=CL1.2 -I OpenCL -I /usr/share/hashcat/Open
CL -D LOCAL_MEM_TYPE=2 -D VENDOR_ID=64 -D CUDA_ARCH=0 -D AMD_ROCM=0 -D VECT
_SIZE=8 -D DEVICE_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST_R2=2 -D DGST_R3=
3 -D DGST_ELEM=4 -D KERN_TYPE=9700 -D _unroll'
* Device #1: Kernel m09700_a0-optimized.5aac43ca.kernel not found in cache!
 Building may take a while...
Dictionary cache building /usr/share/wordlists/rockyou.txt: 33553434 bytes
Dictionary cache building /usr/share/wordlists/rockyou.txt: 100660302 bytes
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec


Session..........: hashcat
Status...........: Cracked
Hash.Type........: MS Office ⩽ 2003 $0/$1, MD5 + RC4
```



```
Hash.Target......: $oldoffice$1*b405d2e0bef836cd538b96de63d64cfd*7c33f...7f
0cad
Time.Started.....: Mon Jan 27 07:53:27 2020 (1 sec)
Time.Estimated...: Mon Jan 27 07:53:28 2020 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   550.0 kH/s (8.75ms) @ Accel:64 Loops:1 Thr:64 Vec:8
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 229545/14344385 (1.60%)
Rejected.........: 169/229545 (0.07%)
Restore.Point....: 221347/14344385 (1.54%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: flutesrock → 150374

Started: Mon Jan 27 07:53:18 2020
Stopped: Mon Jan 27 07:53:29 2020
```

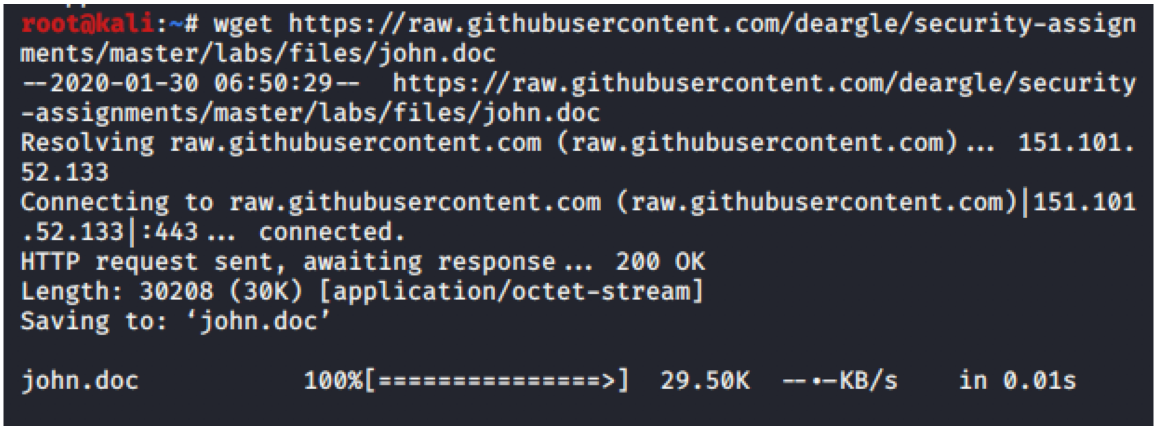
Output for hashcat.doc



Warning, you are using the root account, you may harm your system.

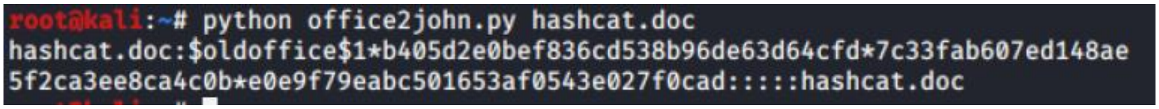
```
$oldoffice$1*b405d2e0bef836cd538b96de63d64cfd*7c33fab607ed148ae5f2ca3ee8ca4c0b*e0
e9f79eabc501653af0543e027f0cad:camp
```

Running for John.doc



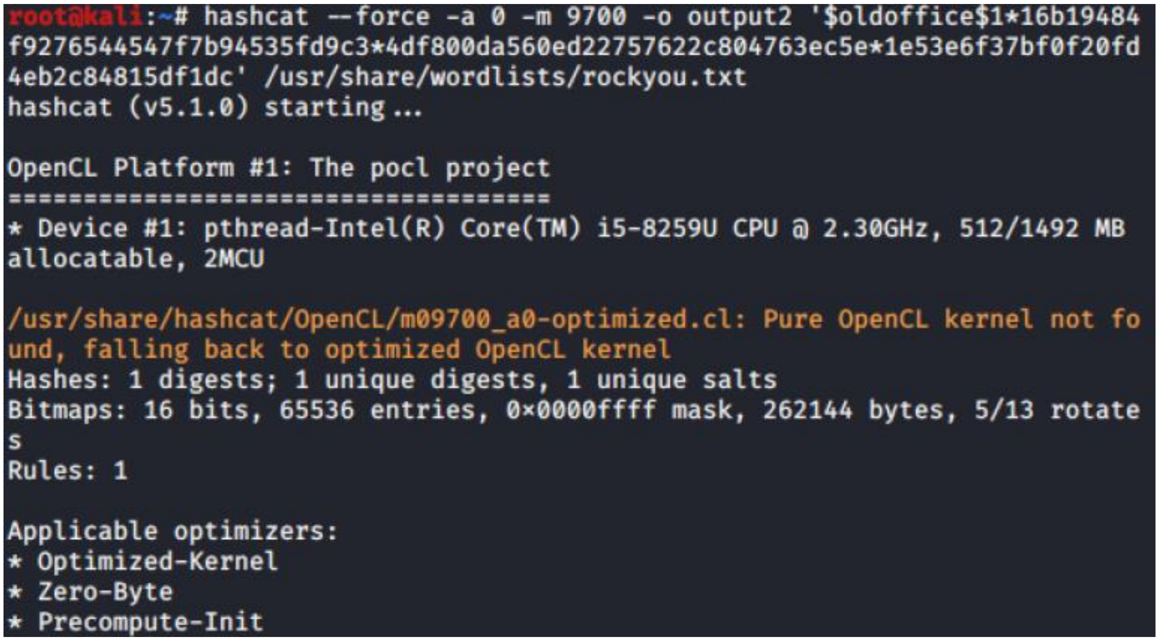
```
root@kali:~# wget https://raw.githubusercontent.com/deargle/security-assign
ments/master/labs/files/john.doc
--2020-01-30 06:50:29--  https://raw.githubusercontent.com/deargle/security
-assignments/master/labs/files/john.doc
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.
52.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101
.52.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30208 (30K) [application/octet-stream]
Saving to: ‘john.doc’

john.doc           100%[===============>]  29.50K  --.-KB/s    in 0.01s
```

Getting hash

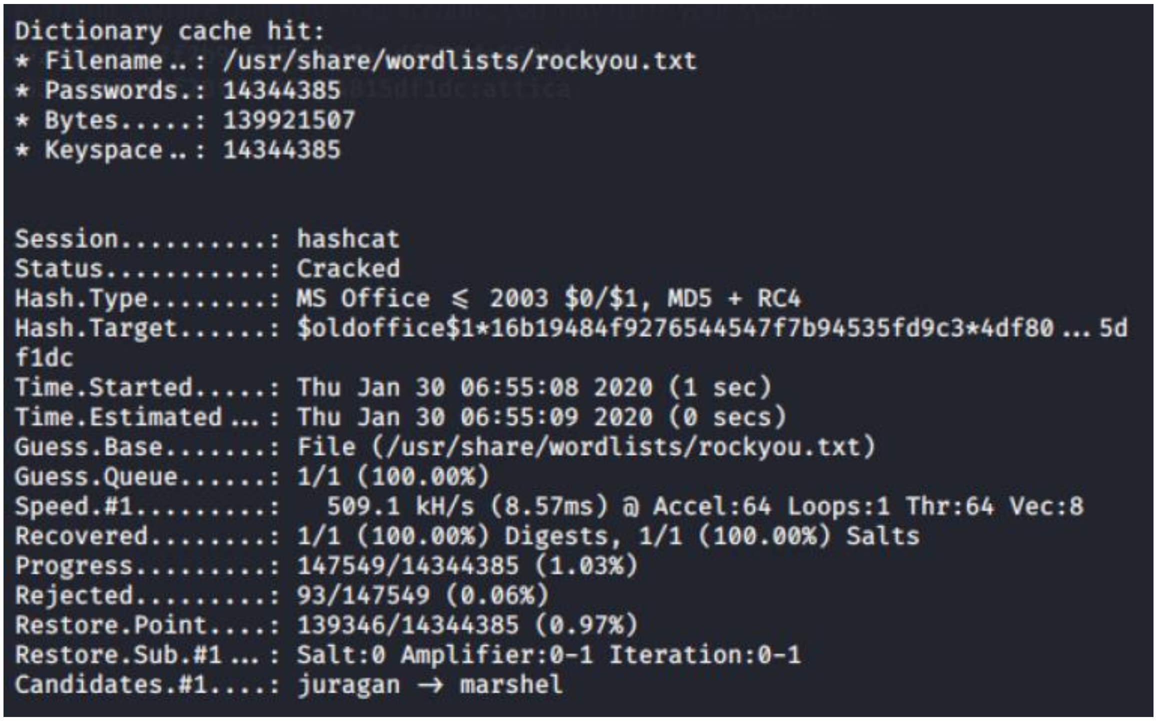


```
root@kali:~# python office2john.py hashcat.doc
hashcat.doc:$oldoffice$1*b405d2e0bef836cd538b96de63d64cfd*7c33fab607ed148ae
5f2ca3ee8ca4c0b*e0e9f79eabc501653af0543e027f0cad::::hashcat.doc
```

Cracking Hash

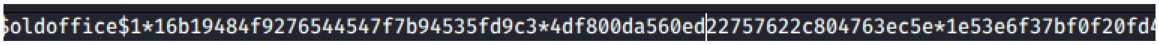


```
root@kali:~# hashcat --force -a 0 -m 9700 -o output2 '$oldoffice$1*16b19484
f9276544547f7b94535fd9c3*4df800da560ed22757622c804763ec5e*1e53e6f37bf0f20fd
4eb2c84815df1dc' /usr/share/wordlists/rockyou.txt
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Core(TM) i5-8259U CPU @ 2.30GHz, 512/1492 MB
allocatable, 2MCU

/usr/share/hashcat/OpenCL/m09700_a0-optimized.cl: Pure OpenCL kernel not fo
und, falling back to optimized OpenCL kernel
Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotate
s
Rules: 1

Applicable optimizers:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
```



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


Session..........: hashcat
Status...........: Cracked
Hash.Type........: MS Office ⩽ 2003 $0/$1, MD5 + RC4
Hash.Target......: $oldoffice$1*16b19484f9276544547f7b94535fd9c3*4df80...5d
f1dc
Time.Started.....: Thu Jan 30 06:55:08 2020 (1 sec)
Time.Estimated...: Thu Jan 30 06:55:09 2020 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   509.1 kH/s (8.57ms) @ Accel:64 Loops:1 Thr:64 Vec:8
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 147549/14344385 (1.03%)
Rejected.........: 93/147549 (0.06%)
Restore.Point....: 139346/14344385 (0.97%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: juragan → marshel
```

Output



```
$oldoffice$1*16b19484f9276544547f7b94535fd9c3*4df800da560ed22757622c804763ec5e*1e53e6f37bf0f20fd
```

Question: What is the password for hashcat.doc? Do the same for the file john.doc (use wget as above to obtain it from url https://raw.githubusercontent.com/deargle/securityassignments/master/labs/files/john.doc). Question: What is the password for john.doc?

Hashcat:14344392

John: