CSEC 744 Network Security

Name : Shriram Karpoora Sundara Pandian

Course Title : Authentication and Remote Access

Lab : 8

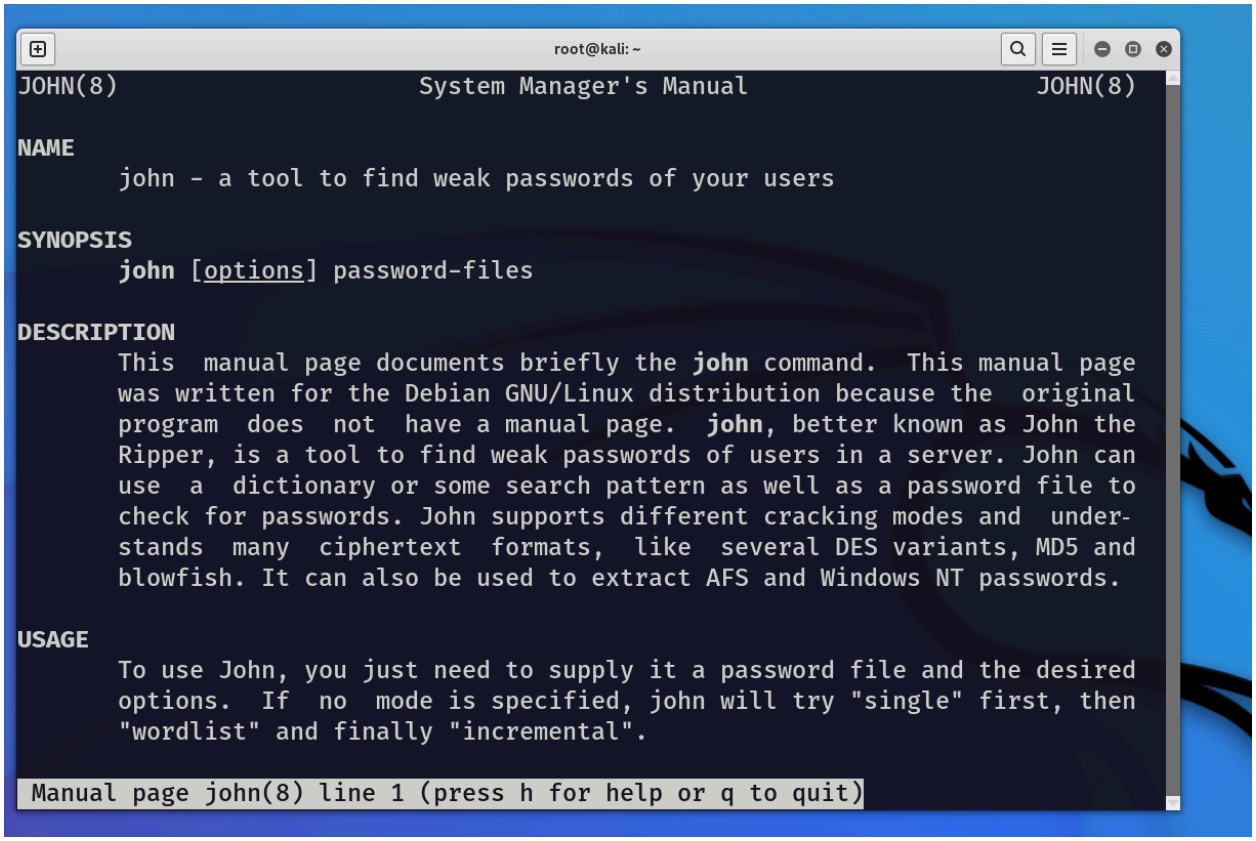
Chapter : 11 (Security Plus)

Step 1 :

A.

```
root@kali:~# sudo john
Created directory: /root/.john
John the Ripper 1.9.0-jumbo-1+bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 OMP
[linux-gnu 64-bit x86_64 AVX2 AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]

Use --help to list all available options.
```

B.


root@kali: ~
JOHN(8) System Manager's Manual JOHN(8)
NAME
john - a tool to find weak passwords of your users
SYNOPSIS
john [options] password-files
DESCRIPTION
This manual page documents briefly the john command. This manual page was written for the Debian GNU/Linux distribution because the original program does not have a manual page. john, better known as John the Ripper, is a tool to find weak passwords of users in a server. John can use a dictionary or some search pattern as well as a password file to check for passwords. John supports different cracking modes and understands many ciphertext formats, like several DES variants, MD5 and blowfish. It can also be used to extract AFS and Windows NT passwords.
USAGE
To use John, you just need to supply it a password file and the desired options. If no mode is specified, john will try "single" first, then "wordlist" and finally "incremental".
Manual page john(8) line 1 (press h for help or q to quit)

C.

```
root@kali:~# john --test
Will run 8 OpenMP threads
Benchmarking: descrypt, traditional crypt(3) [DES 256/256 AVX2]... (8xOMP) DONE
Many salts:     14327K c/s real, 1972K c/s virtual
Only one salt:  9805K c/s real, 1342K c/s virtual

Benchmarking: bsdicrypt, BSDI crypt(3) ("_J9..", 725 iterations) [DES 256/256 AVX2]... (8xOMP) DONE
Speed for cost 1 (iteration count) of 725
Many salts:     406558 c/s real, 56716 c/s virtual
Only one salt:  360692 c/s real, 49999 c/s virtual

Benchmarking: md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3]... (8xOMP) DONE
Many salts:     104310 c/s real, 14549 c/s virtual
Only one salt:  106985 c/s real, 14569 c/s virtual

Benchmarking: md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64]... (8xOMP) DONE
Raw:    14496 c/s real, 1995 c/s virtual

Benchmarking: bcrypt ("$2a$05", 32 iterations) [Blowfish 32/64 X3]... (8xOMP) DONE
Speed for cost 1 (iteration count) of 32
Raw:    4047 c/s real, 607 c/s virtual

Benchmarking: scrypt (16384, 8, 1) [Salsa20/8 128/128 AVX]... (8xOMP) DONE
Speed for cost 1 (N) of 16384, cost 2 (r) of 8, cost 3 (p) of 1
Raw:    107 c/s real, 17.4 c/s virtual

Benchmarking: LM [DES 256/256 AVX2]... (8xOMP) DONE
Raw:    41508K c/s real, 5785K c/s virtual

Benchmarking: AFS, Kerberos AFS [DES 48/64 4K]... DONE
Short:  215552 c/s real, 215552 c/s virtual
Long:   188160 c/s real, 189105 c/s virtual

Benchmarking: tripcode [DES 256/256 AVX2]... (8xOMP) DONE
Raw:    686567 c/s real, 101821 c/s virtual

Benchmarking: AndroidBackup [PBKDF2-SHA1 256/256 AVX2 8x AES]... (8xOMP) DONE
Speed for cost 1 (iteration count) of 10000
Raw:    1923 c/s real, 319 c/s virtual
```

D.

```
root@kali:~# sudo adduser weissman
Adding user `weissman' ...
Adding new group `weissman' (1001) ...
Adding new user `weissman' (1001) with group `weissman' ...
Creating home directory `/home/weissman' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for weissman
Enter the new value, or press ENTER for the default
        Full Name []: jonathan weissman
        Room Number []: 123
        Work Phone []: 58595758585
        Home Phone []: 58595748483
        Other []: nothing
Is the information correct? [Y/n] y
```

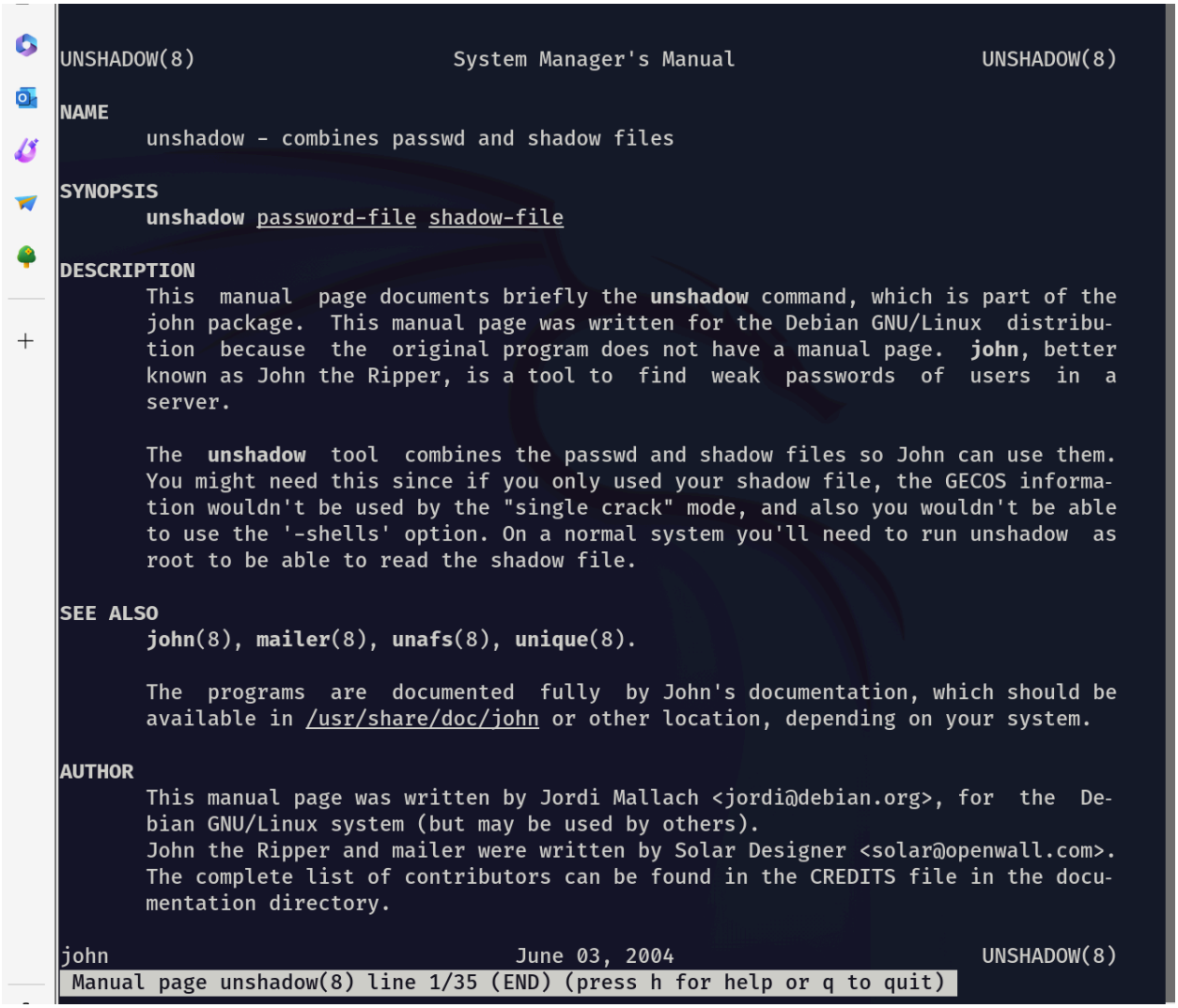
E.

```
root@kali:~# sudo adduser mixed
Adding user `mixed' ...
Adding new group `mixed' (1004) ...
Adding new user `mixed' (1004) with group `mixed' ...
Creating home directory `/home/mixed' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mixed
Enter the new value, or press ENTER for the default
        Full Name []: mixed
        Room Number []: 234
        Work Phone []: 234234
        Home Phone []: 141442
        Other []: hello
Is the information correct? [Y/n] y
root@kali:~# sudo adduser story
Adding user `story' ...
Adding new group `story' (1005) ...
Adding new user `story' (1005) with group `story' ...
Creating home directory `/home/story' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for story
Enter the new value, or press ENTER for the default
        Full Name []: story
        Room Number []: 23
        Work Phone []: 23423
        Home Phone []: 234234
        Other []: hello well
Is the information correct? [Y/n] y
```

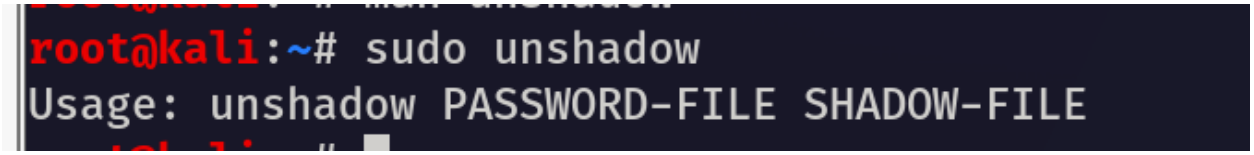
F.

```
:/bin/bash
upper:x:1002:1002:upper,2,2342,234234,2342:/home/upper:/bin/bash
lower:x:1003:1003:lower,42,24243,234234,jalfk:/home/lower:/bin/bash
mixed:x:1004:1004:mixed,234,234234,141442,hello:/home/mixed:/bin/bash
story:x:1005:1005:story,23,23423,234234,hello well:/home/story:/bin/bash
```

G.


UNSHADOW(8) System Manager's Manual UNSHADOW(8)
NAME
unshadow - combines passwd and shadow files
SYNOPSIS
unshadow password-file shadow-file
DESCRIPTION
This manual page documents briefly the unshadow command, which is part of the john package. This manual page was written for the Debian GNU/Linux distribution because the original program does not have a manual page. john, better known as John the Ripper, is a tool to find weak passwords of users in a server.
The unshadow tool combines the passwd and shadow files so John can use them. You might need this since if you only used your shadow file, the GECOS information wouldn't be used by the "single crack" mode, and also you wouldn't be able to use the '-shells' option. On a normal system you'll need to run unshadow as root to be able to read the shadow file.
SEE ALSO
john(8), mailer(8), unafs(8), unique(8).
The programs are documented fully by John's documentation, which should be available in /usr/share/doc/john or other location, depending on your system.
AUTHOR
This manual page was written by Jordi Mallach <jordi@debian.org>, for the Debian GNU/Linux system (but may be used by others).
John the Ripper and mailer were written by Solar Designer <solar@openwall.com>. The complete list of contributors can be found in the CREDITS file in the documentation directory.
john June 03, 2004 UNSHADOW(8)
Manual page unshadow(8) line 1/35 (END) (press h for help or q to quit)

H.


root@kali:~# sudo unshadow
Usage: unshadow PASSWORD-FILE SHADOW-FILE

I and J.

```
root@kali:~# sudo unshadow /etc/passwd /etc/shadow > rochester.txt
root@kali:~# cat rochester.txt
root:$y$j9T$45yBK5Gh/nZE1xsVy5.wP.$25x5TAzinAYVAevEsE396P60x/PU3yFw/FytUdEhjQ1:0:0:root:/
root:/usr/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy:/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:*:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:*:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:!:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:!:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:!:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:!:103:110:MySQL Server,,,:/nonexistent:/bin/false
systemd-timesync:!:104:111:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
redsocks:!:105:112::/var/run/redsocks:/usr/sbin/nologin
rwhod:!:106:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:!:107:65534::/run/iodine:/usr/sbin/nologin
messagebus:!:108:113::/nonexistent:/usr/sbin/nologin
miredo:!:109:65534::/var/run/miredo:/usr/sbin/nologin
tcpdump:!:110:119::/nonexistent:/usr/sbin/nologin
sshd:!:111:65534::/run/sshd:/usr/sbin/nologin
_rpc:!:112:65534::/run/rpcbind:/usr/sbin/nologin
```

```
beef-xss:!:128:139::/var/lib/beef-xss:/usr/sbin/nologin
weissman:$y$j9T$W5gP3KCHLcnn8MM89zO2n/$Ut5gBEC.mTqQ50KtpPhO6oueAzhlV3V/CEj6ZApxVy3:1001:1
001:jonathan weissman,123,58595758585,58595748483,nothing:/home/weissman:/bin/bash
upper:$y$j9T$vFG/HY1nSeT6AYwqw9nXI1$QSncBCAcW7nC2OKqBuv92nBVAzRCEWY5yWYzJiddQZ/:1002:1002
:upper,2,2342,234234,2342:/home/upper:/bin/bash
lower:$y$j9T$XPnQK9XdiQE1xjjCfuUdJ1$Oz1WXUtW/gNpnUlNHmvybjhAf6.OZXekeFRMBLvi9d5:1003:1003
:lower,42,24243,234234,jalfk:/home/lower:/bin/bash
mixed:$y$j9T$xlJDoU/xguyVTP1HtCFuc1$jRmCLQBfBCQxXHcWa4efZJB8Nx/oSJJMOwimUR7zyc2:1004:1004
:mixed,234,234234,141442,hello:/home/mixed:/bin/bash
story:$y$j9T$ksvi7CGdZzDn94pYLeZ1V0$tecCIEgPnldx18OwCpZyCqZbbqXRn.N14fD3/79wlG2:1005:1005
:story,23,23423,234234,hello well:/home/story:/bin/bash
```

K.

```
root@kali:~# sudo john --wordlist=/usr/share/john/password.lst --format=crypt rochester.t
xt
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt])
is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
jonathan         (weissman)
password         (lower)
Password         (mixed)
PASSWORD         (upper)
3bears           (story)
kali             (kali)
6g 0:00:01:13 DONE (2024-04-01 11:32) 0.08198g/s 48.45p/s 179.6c/s 179.6C/s !@#$%..sss
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

L.

```
root@kali:~# sudo john --show --format=crypt rochester.txt
kali:kali:1000:1000:kali,,,:/home/kali:/usr/bin/zsh
weissman:jonathan:1001:1001:jonathan weissman,123,58595758585,58595748483,nothing:/home/w
eissman:/bin/bash
upper:PASSWORD:1002:1002:upper,2,2342,234234,2342:/home/upper:/bin/bash
lower:password:1003:1003:lower,42,24243,234234,jalfk:/home/lower:/bin/bash
mixed:Password:1004:1004:mixed,234,234234,141442,hello:/home/mixed:/bin/bash
story:3bears:1005:1005:story,23,23423,234234,hello well:/home/story:/bin/bash

6 password hashes cracked, 1 left
```

M.

```
6 password hashes cracked, 1 left
root@kali:~# sudo rm /root/.john/john.pot
root@kali:~#
```

N.

```
root@kali:~# sudo adduser scott
Adding user `scott' ...
Adding new group `scott' (1006) ...
Adding new user `scott' (1006) with group `scott' ...
Creating home directory `/home/scott' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for scott
Enter the new value, or press ENTER for the default
        Full Name []: scott
        Room Number []: 10314
        Work Phone []: ^Cadduser: `/bin/chfn scott' exited from signal 2. Exiting.
```

O and P.

```
root@kali:~# sudo unshadow /etc/passwd /etc/shadow > rochester2.txt
root@kali:~# sudo john --format=crypt rochester2.txt
Using default input encoding: UTF-8
Loaded 8 password hashes with 8 different salts (crypt, generic crypt(3) [?/64])
Remaining 1 password hash
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt])
is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
scott10314       (scott)
1g 0:00:00:00 DONE 1/3 (2024-04-01 11:58) 1.333g/s 128.0p/s 128.0c/s 128.0C/s scott..tt10
3
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Q.

```
root@kali:~# sudo john --show --format=crypt rochester2.txt
root:toor:0:0:root:/root:/usr/bin/bash
kali:kali:1000:1000:kali,,,:/home/kali:/usr/bin/zsh
weissman:jonathan:1001:1001:jonathan weissman,123,58595758585,58595748483,nothing:/home/w
eissman:/bin/bash
upper:PASSWORD:1002:1002:upper,2,2342,234234,2342:/home/upper:/bin/bash
lower:password:1003:1003:lower,42,24243,234234,jalfk:/home/lower:/bin/bash
mixed:Password:1004:1004:mixed,234,234234,141442,hello:/home/mixed:/bin/bash
story:3bears:1005:1005:story,23,23423,234234,hello well:/home/story:/bin/bash
scott:scott10314:1006:1006:scott,10314,,:/home/scott:/bin/bash

8 password hashes cracked, 0 left
```

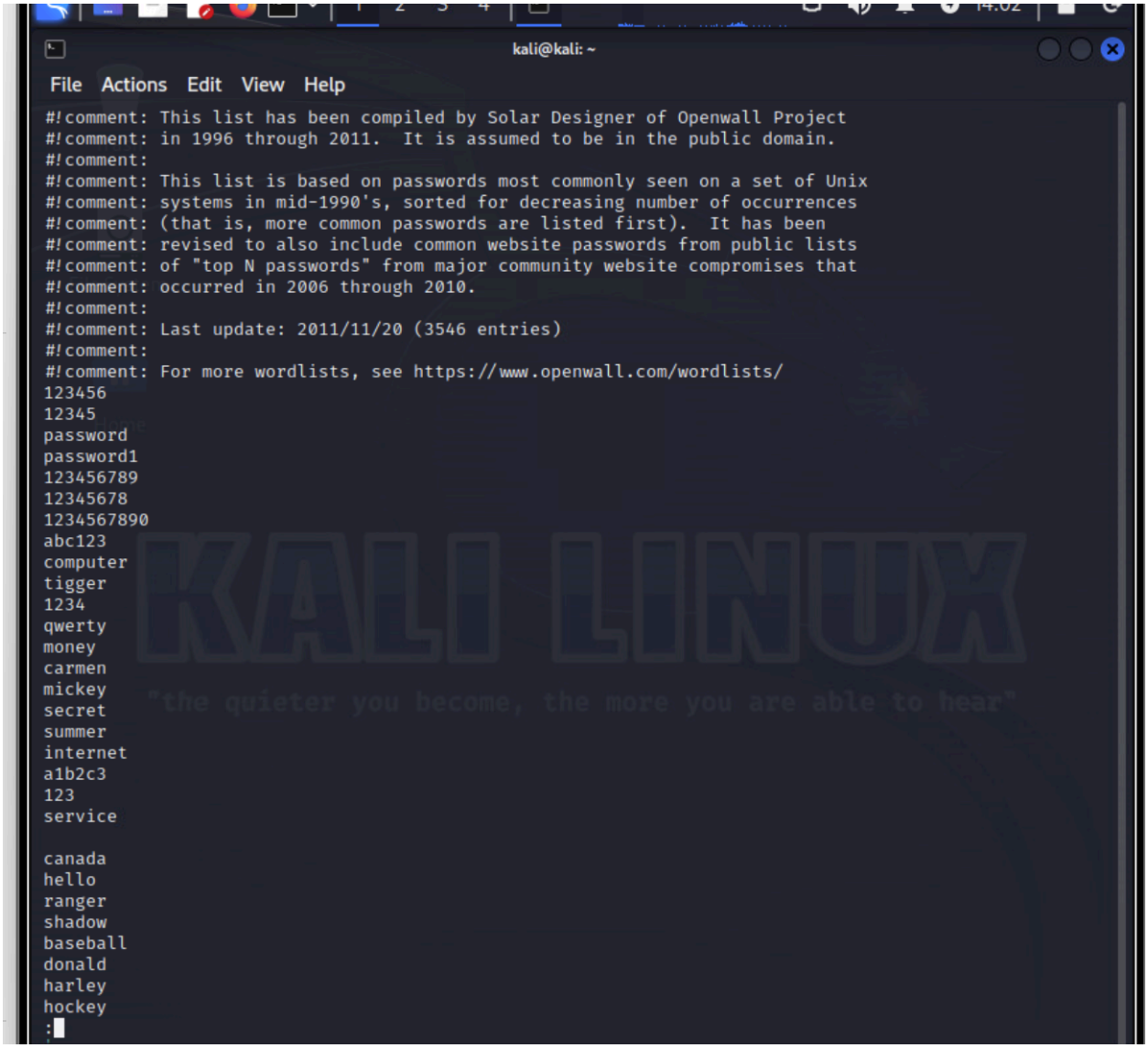
Step 2: (From here i am using RLES Kali 3)
A,B,C.

```
┌──(kali㉿kali)-[~]
└─$ cp /usr/share/wordlists/rockyou.txt.gz .

┌──(kali㉿kali)-[~]
└─$ gzip -d rockyou.txt.gz

┌──(kali㉿kali)-[~]
└─$ ls -l /usr/share/john/password.lst
-rw-r--r-- 1 root root 26326 Nov  2  2021 /usr/share/john/password.lst

┌──(kali㉿kali)-[~]
└─$ ls -l rockyou.txt
-rw-r--r-- 1 kali kali 139921507 Apr  1 13:45 rockyou.txt
```

D.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install leafpad
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Suggested packages:
  evince-gtk
The following NEW packages will be installed:
  leafpad
0 upgraded, 1 newly installed, 0 to remove and 1916 not upgraded.
Need to get 90.9 kB of archives.
After this operation, 465 kB of additional disk space will be used.
Get:1 http://mirrors.jevincanders.net/kali kali-rolling/main amd64 leafpad amd64 0.8.18.1-5 [90.9 kB
]
Fetched 90.9 kB in 0s (1,177 kB/s)
Selecting previously unselected package leafpad.
(Reading database ... 392692 files and directories currently installed.)
Preparing to unpack .../leafpad_0.8.18.1-5_amd64.deb ...
Unpacking leafpad (0.8.18.1-5) ...
Setting up leafpad (0.8.18.1-5) ...
update-alternatives: using /usr/bin/leafpad to provide /usr/bin/gnome-text-editor (gnome-text-editor
) in auto mode
Processing triggers for kali-menu (2023.1.7) ...
Processing triggers for desktop-file-utils (0.26-1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
```

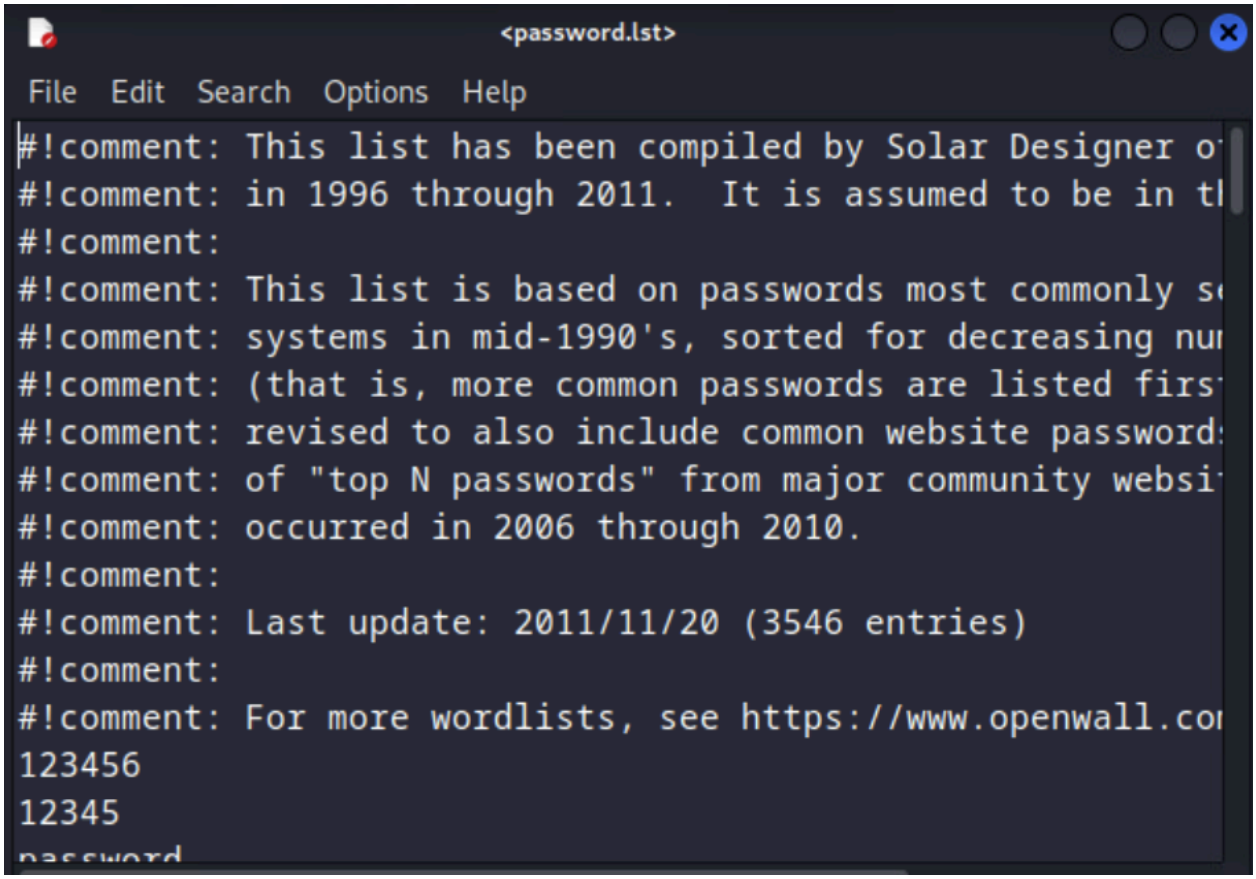
E.

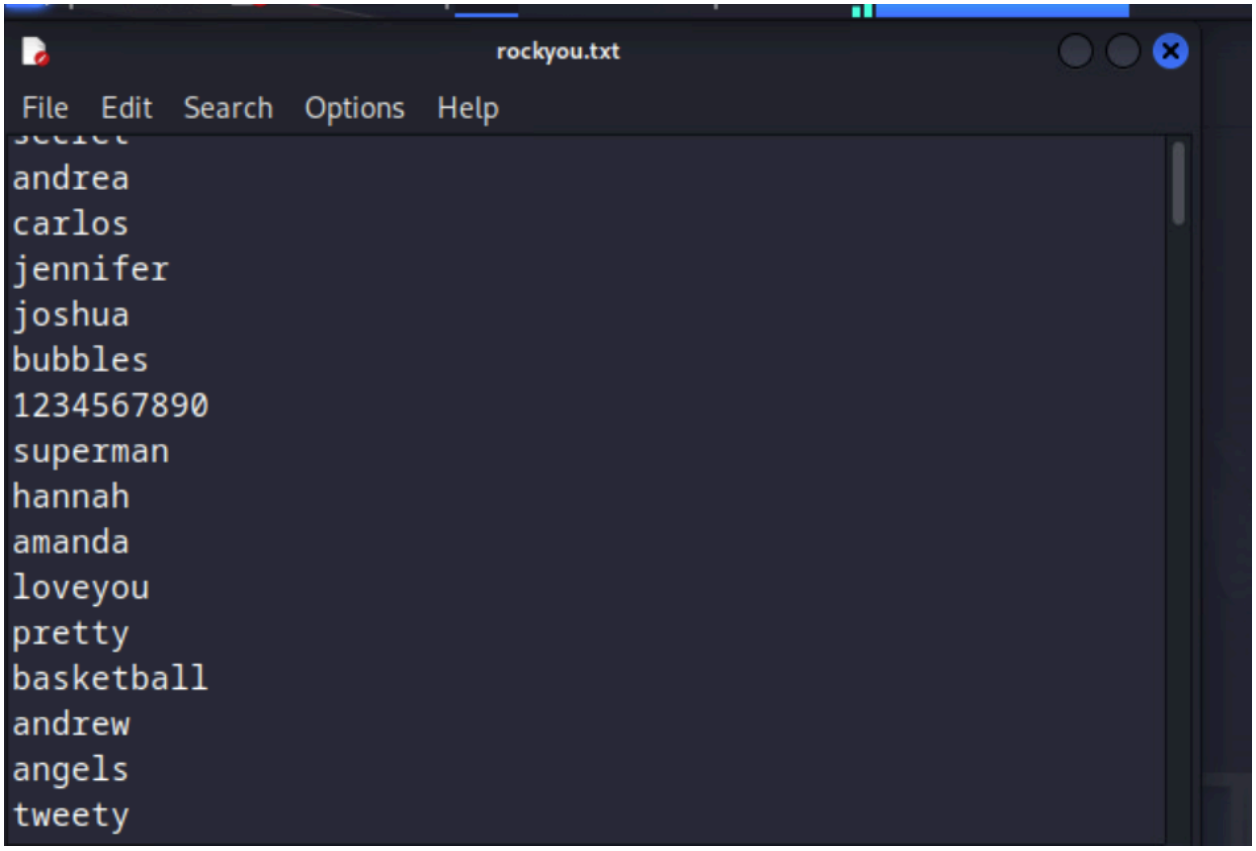


Rockyou.txt

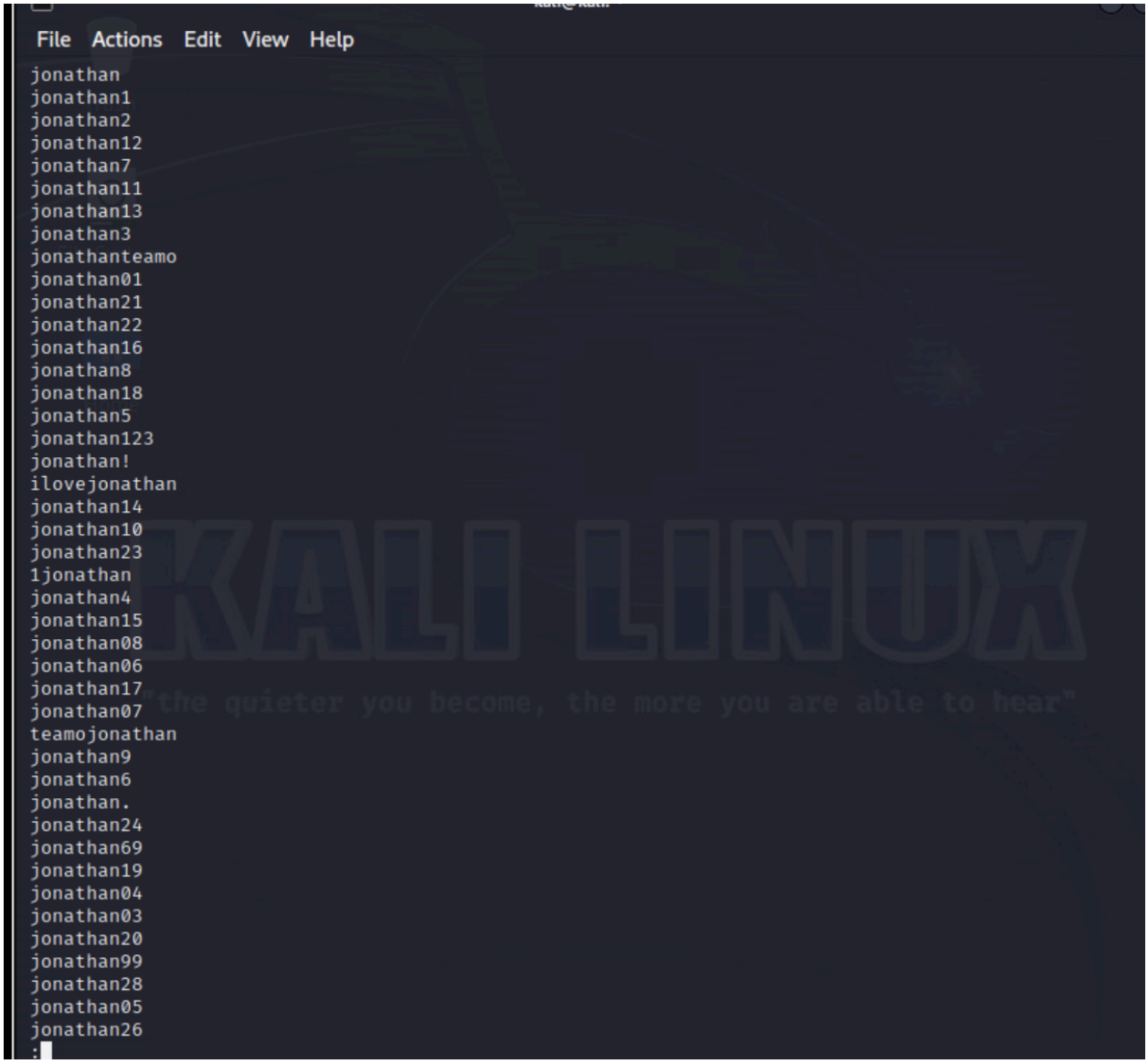

```
┌──(kali㉿kali)-[~]
└─$ cat /usr/share/john/password.lst | less

┌──(kali㉿kali)-[~]
└─$ cat rockyou.txt | less
```

F.

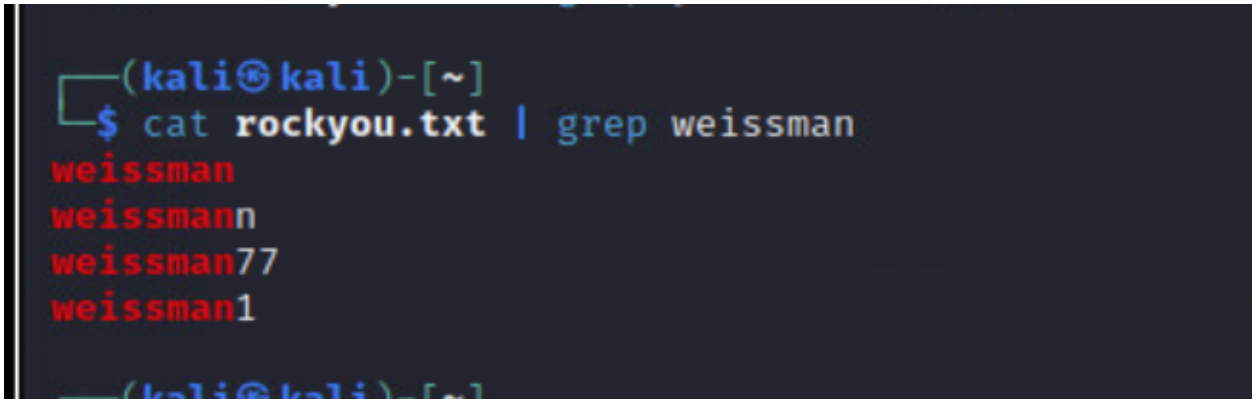

<password.lst>
File Edit Search Options Help
#!comment: This list has been compiled by Solar Designer o
#!comment: in 1996 through 2011. It is assumed to be in t
#!comment:
#!comment: This list is based on passwords most commonly s
#!comment: systems in mid-1990's, sorted for decreasing nu
#!comment: (that is, more common passwords are listed firs
#!comment: revised to also include common website password
#!comment: of "top N passwords" from major community websi
#!comment: occurred in 2006 through 2010.
#!comment:
#!comment: Last update: 2011/11/20 (3546 entries)
#!comment:
#!comment: For more wordlists, see https://www.openwall.co
123456
12345
password


rockyou.txt
File Edit Search Options Help
andrea
carlos
jennifer
joshua
bubbles
1234567890
superman
hannah
amanda
loveyou
pretty
basketball
andrew
angels
tweety

G.


File Actions Edit View Help
jonathan
jonathan1
jonathan2
jonathan12
jonathan7
jonathan11
jonathan13
jonathan3
jonathanteamo
jonathan01
jonathan21
jonathan22
jonathan16
jonathan8
jonathan18
jonathan5
jonathan123
jonathan!
ilovejonathan
jonathan14
jonathan10
jonathan23
1jonathan
jonathan4
jonathan15
jonathan08
jonathan06
jonathan17
jonathan07
teamojonathan
jonathan9
jonathan6
jonathan.
jonathan24
jonathan69
jonathan19
jonathan04
jonathan03
jonathan20
jonathan99
jonathan28
jonathan05
jonathan26
KALI LINUX
"the quieter you become, the more you are able to hear"

H.


(kali㉿kali)-[~]
$ cat rockyou.txt | grep weissman
weissman
weissmann
weissman77
weissman1

I.

```
┌──(kali㉿kali)-[~]
└─$ cat rockyou.txt | grep shriram
jaishriram
shriram
shriramа
shriramkumar
shriramcs
shriram1490
jayshriramji
jayshriram
jaishriramji
jaishriram8
bolojaishriram4hakathe58
```

```
┌──(kali㉿kali)-[~]
└─$ cat rockyou.txt | grep anime
anime
anime1
animelover
animes
iloveanime
animefreak
anime123
animegirl
animefan
anime101
anime12
anime2
anime7
anime5
animerox
animerules
animemanga
animelove
anime4ever
```

```
┌──(kali㉿kali)-[~]
└─$ cat rockyou.txt | grep looping
looping
looping76
looping104
looping103
looping1
looping04
blooping
```

J.

```
┌──(kali㉿kali)-[~]
└─$ sudo adduser shriram
[sudo] password for kali:
Adding user `shriram' ...
Adding new group `shriram' (1001) ...
Adding new user `shriram' (1001) with group `shriram (1001)' ...
Creating home directory `/home/shriram' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for shriram
Enter the new value, or press ENTER for the default
        Full Name []: shriram
        Room Number []: 18
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
Adding new user `shriram' to supplemental / extra groups `users' ...
Adding user `shriram' to group `users' ...

┌──(kali㉿kali)-[~]
└─$ sudo adduser swathy
Adding user `swathy' ...
Adding new group `swathy' (1002) ...
Adding new user `swathy' (1002) with group `swathy (1002)' ...
Creating home directory `/home/swathy' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for swathy
Enter the new value, or press ENTER for the default
        Full Name []: swathy
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
Adding new user `swathy' to supplemental / extra groups `users' ...
Adding user `swathy' to group `users' ...
```

```
┌──(kali㉿kali)-[~]
└─$ sudo unshadow /etc/passwd /etc/shadow > rochester3.txt
Created directory: /root/.john
```

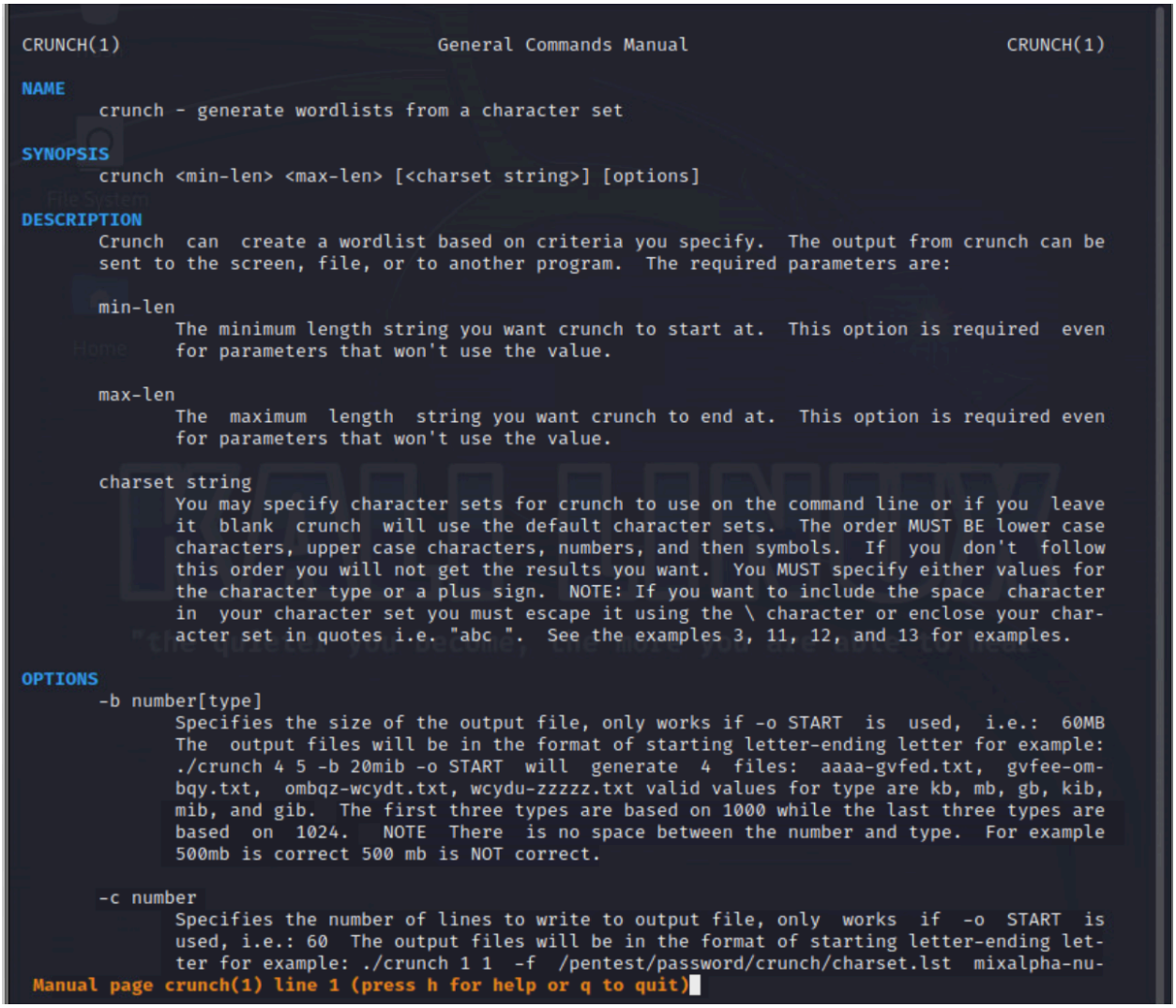
K. I got for one password for the username, but otherone took so song

```
┌──(kali㉿kali)-[~]
└─$ sudo john --wordlist=rockyou.txt --format=crypt rochester3.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for al
l loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
anime123         (swathy)
```

Exercise 11. 02

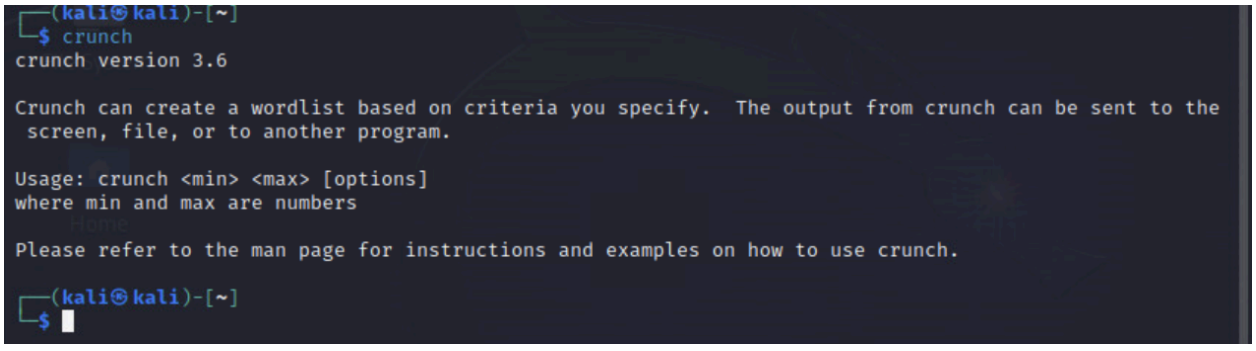
Step 1:

A.



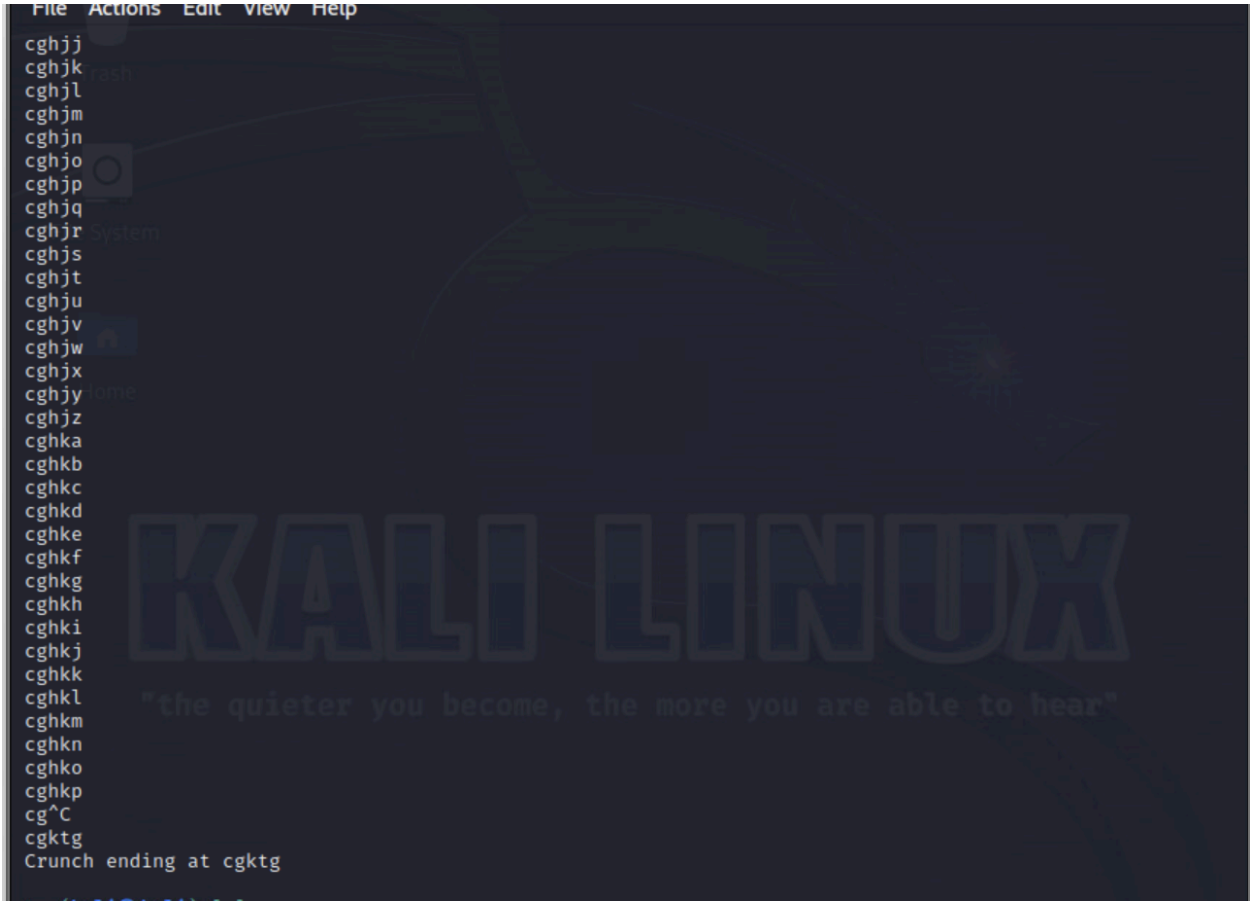
```
CRUNCH(1)                        General Commands Manual                        CRUNCH(1)

NAME
       crunch - generate wordlists from a character set

SYNOPSIS
       crunch <min-len> <max-len> [<charset string>] [options]

DESCRIPTION
       Crunch  can  create a wordlist based on criteria you specify.  The output from crunch can be
       sent to the screen, file, or to another program.  The required parameters are:

       min-len
              The minimum length string you want crunch to start at.  This option is required  even
              for parameters that won't use the value.

       max-len
              The  maximum  length  string you want crunch to end at.  This option is required even
              for parameters that won't use the value.

       charset string
              You may specify character sets for crunch to use on the command line or if you  leave
              it  blank  crunch  will use the default character sets.  The order MUST BE lower case
              characters, upper case characters, numbers, and then symbols.  If  you  don't  follow
              this order you will not get the results you want.  You MUST specify either values for
              the character type or a plus sign.  NOTE: If you want to include the space  character
              in  your character set you must escape it using the \ character or enclose your char-
              acter set in quotes i.e. "abc ".  See the examples 3, 11, 12, and 13 for examples.

OPTIONS
       -b number[type]
              Specifies the size of the output file, only works if -o START  is  used,  i.e.:  60MB
              The  output files will be in the format of starting letter-ending letter for example:
              ./crunch 4 5 -b 20mib -o START  will  generate  4  files:  aaaa-gvfed.txt,  gvfee-om-
              bqy.txt,  ombqz-wcydt.txt, wcydu-zzzzz.txt valid values for type are kb, mb, gb, kib,
              mib, and gib.  The first three types are based on 1000 while the last three types are
              based  on  1024.   NOTE  There  is no space between the number and type.  For example
              500mb is correct 500 mb is NOT correct.

       -c number
              Specifies the number of lines to write to output file, only  works  if  -o  START  is
              used, i.e.: 60  The output files will be in the format of starting letter-ending let-
              ter for example: ./crunch 1 1  -f  /pentest/password/crunch/charset.lst  mixalpha-nu-
 Manual page crunch(1) line 1 (press h for help or q to quit)
```

B

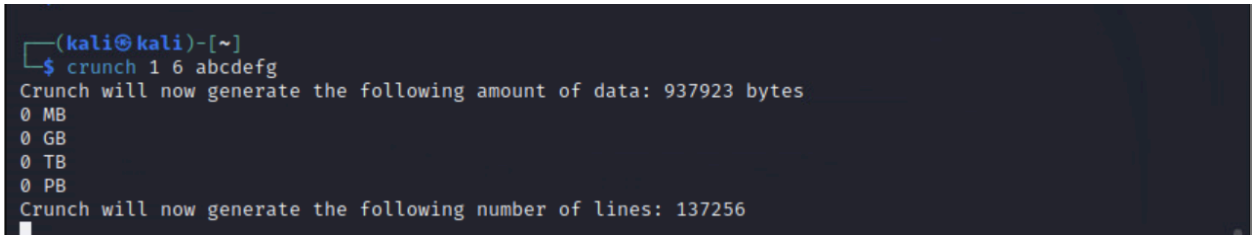


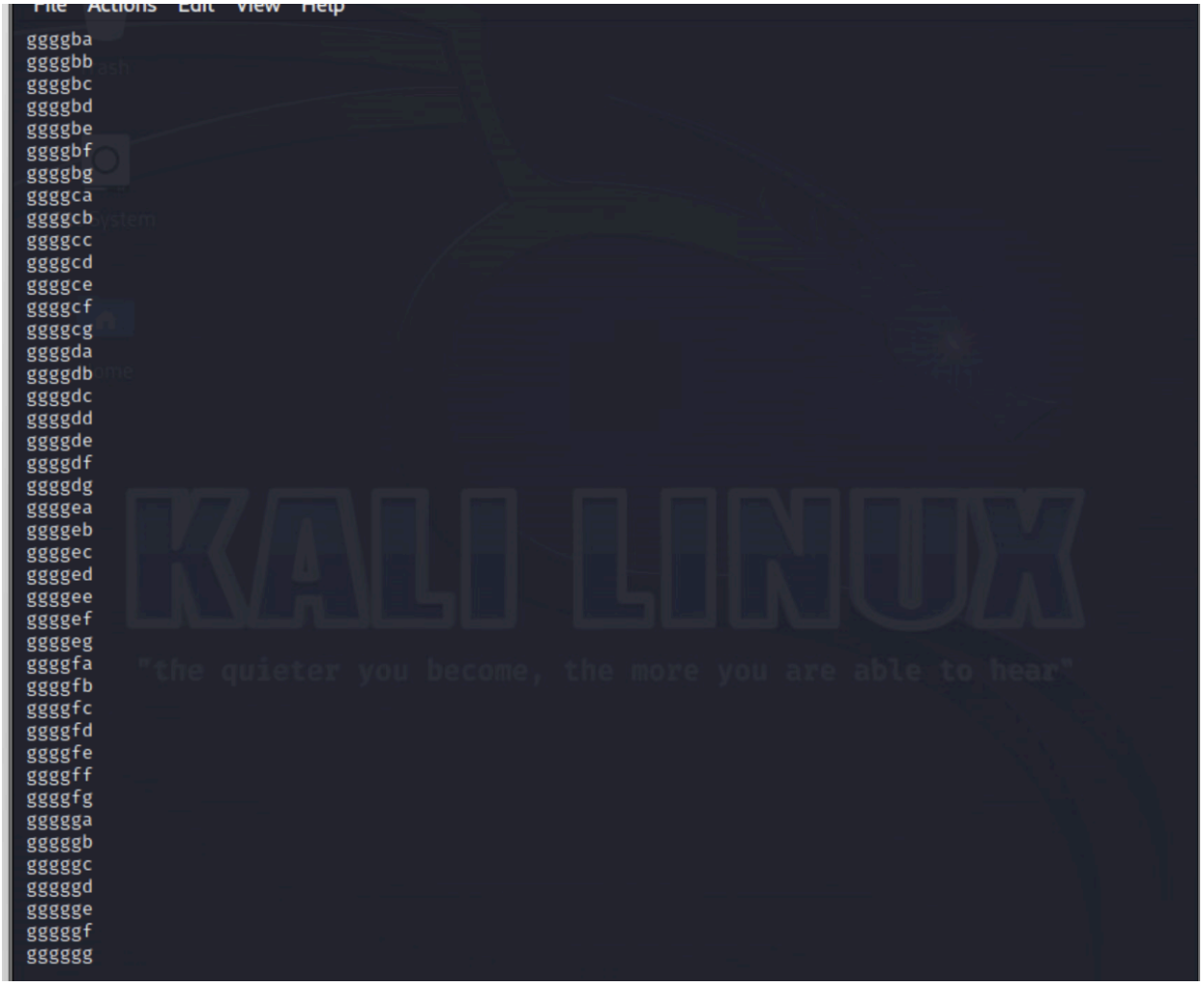
```
┌──(kali㉿kali)-[~]
└─$ crunch
crunch version 3.6

Crunch can create a wordlist based on criteria you specify.  The output from crunch can be sent to the
 screen, file, or to another program.

Usage: crunch <min> <max> [options]
where min and max are numbers

Please refer to the man page for instructions and examples on how to use crunch.

┌──(kali㉿kali)-[~]
└─$
```

C.

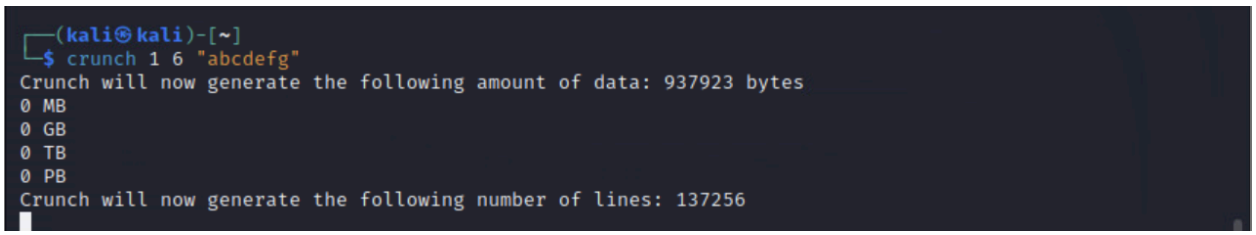

File Actions Edit View Help
cghjj
cghjk
cghjl
cghjm
cghjn
cghjo
cghjp
cghjq
cghjr
cghjs
cghjt
cghju
cghjv
cghjw
cghjx
cghjy
cghjz
cghka
cghkb
cghkc
cghkd
cghke
cghkf
cghkg
cghkh
cghki
cghkj
cghkk
cghkl
cghkm
cghkn
cghko
cghkp
cg^C
cgktg
Crunch ending at cgktg
KALI LINUX
"the quieter you become, the more you are able to hear"

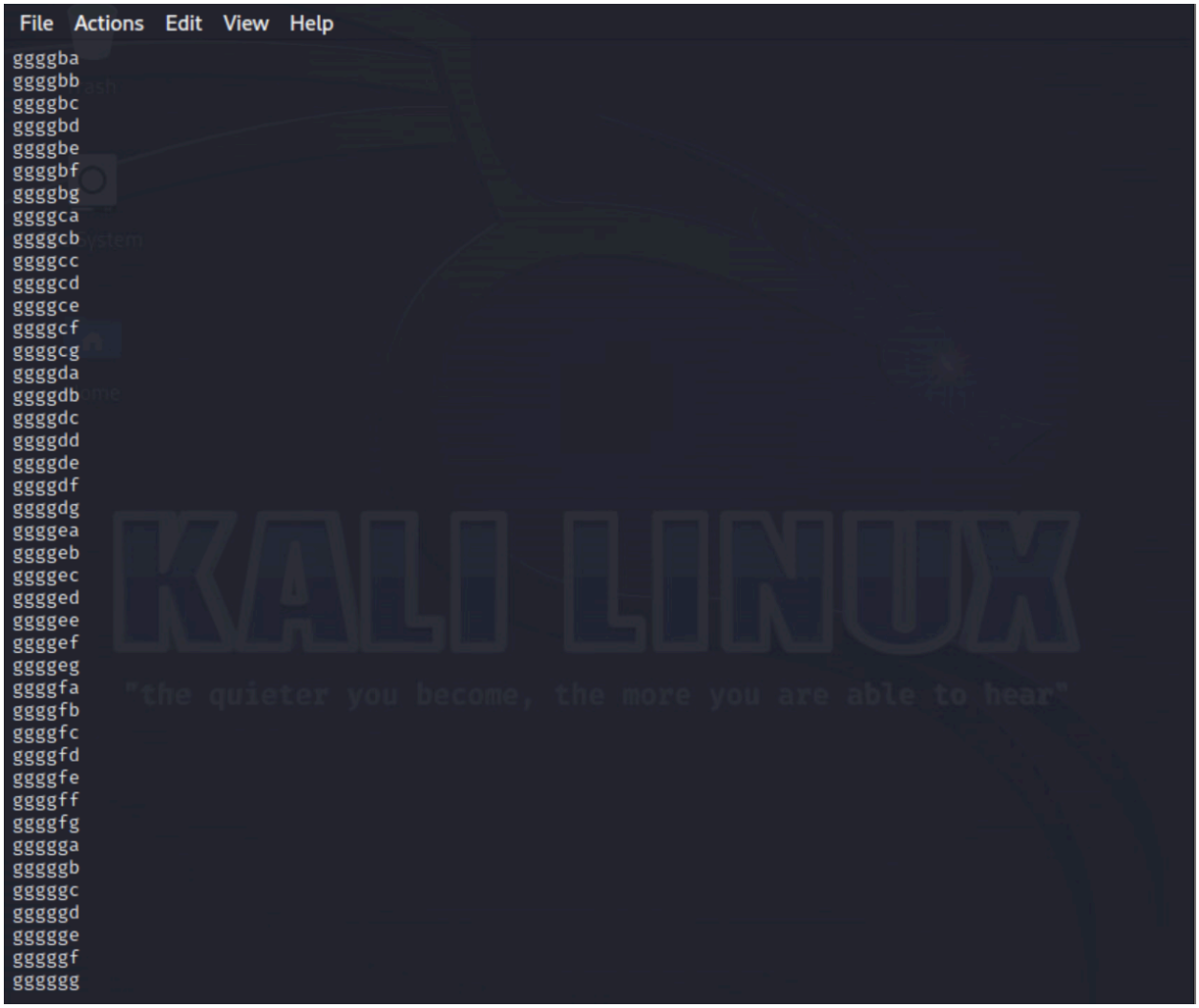
D.


┌──(kali㉿kali)-[~]
└─$ crunch 1 6 abcdefg
Crunch will now generate the following amount of data: 937923 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 137256

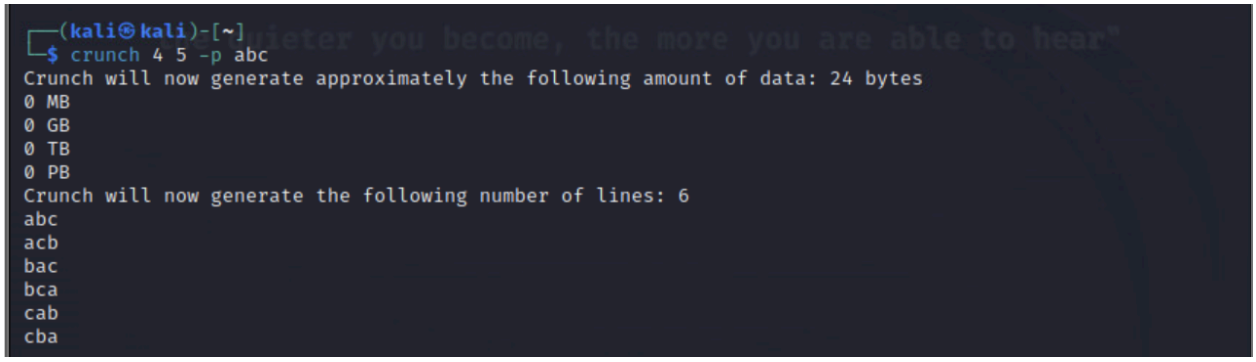

File Actions Edit View Help
ggggba
ggggbb
ggggbc
ggggbd
ggggbe
ggggbf
ggggbg
ggggca
ggggcb
ggggcc
ggggcd
ggggce
ggggcf
ggggcg
ggggda
ggggdb
ggggdc
ggggdd
ggggde
ggggdf
ggggdg
ggggea
ggggeb
ggggec
gggged
ggggee
ggggef
ggggeg
ggggfa
ggggfb
ggggfc
ggggfd
ggggfe
ggggff
ggggfg
ggggga
gggggb
ggggbc
ggggbd
gggggge
ggggbf
gggggg
KALI LINUX
"the quieter you become, the more you are able to hear"

E.


┌──(kali㉿kali)-[~]
└─$ crunch 1 6 "abcdefg"
Crunch will now generate the following amount of data: 937923 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 137256


File Actions Edit View Help
ggggba
ggggbb
ggggbc
ggggbd
ggggbe
ggggbf
ggggbg
ggggca
ggggcb
ggggcc
ggggcd
ggggce
ggggcf
ggggcg
ggggda
ggggdb
ggggdc
ggggdd
ggggde
ggggdf
ggggdg
ggggea
ggggeb
ggggec
gggged
ggggee
ggggef
ggggeg
ggggfa
ggggfb
ggggfc
ggggfd
ggggfe
ggggff
ggggfg
gggggga
ggggggb
gggggc
ggggggd
gggggge
ggggggf
gggggg

F.


(kali㉿kali)-[~]
$ crunch 4 5 -p abc
Crunch will now generate approximately the following amount of data: 24 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 6
abc
acb
bac
bca
cab
cba

G.

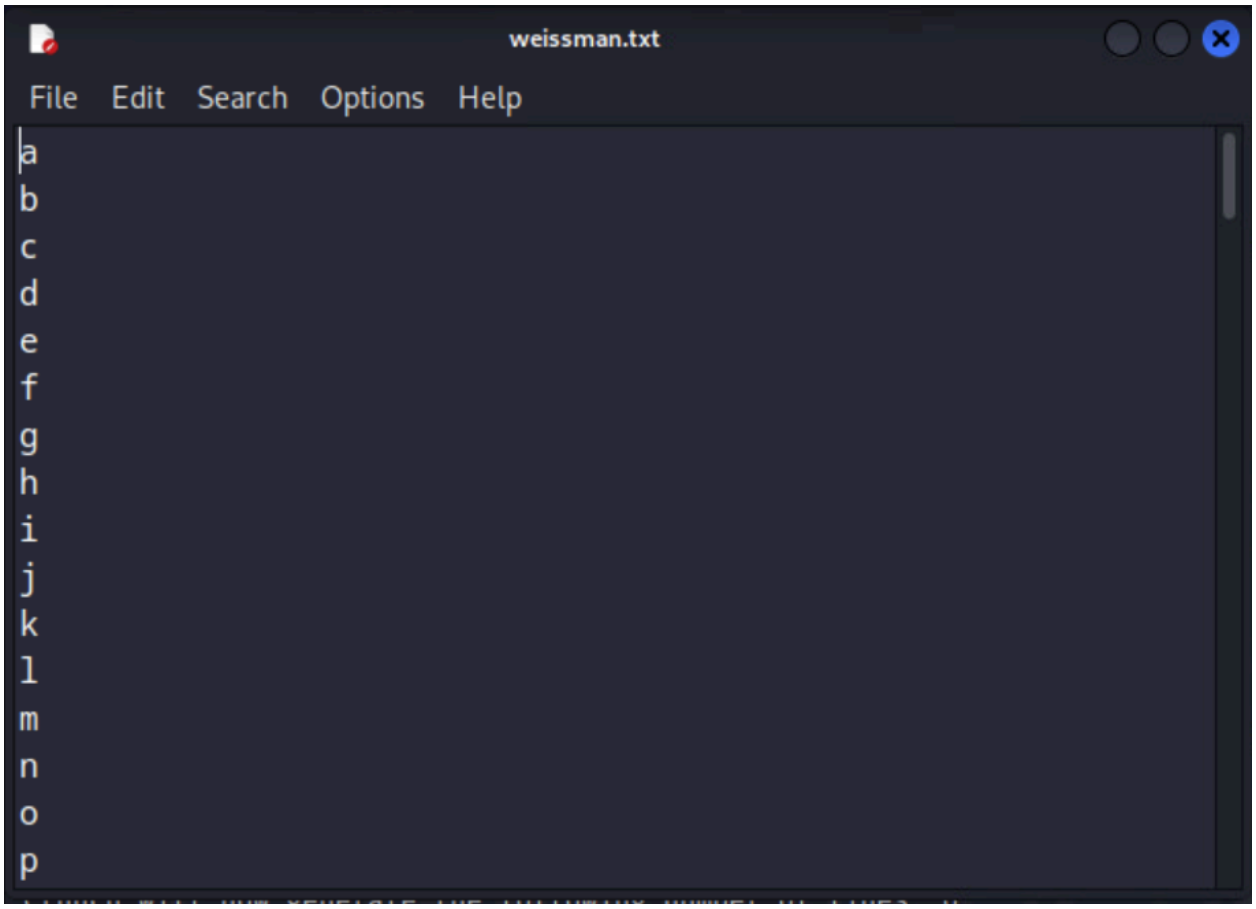
```
┌──(kali㉿kali)-[~]
└─$ crunch 4 5 -p dog cat bird
Crunch will now generate approximately the following amount of data: 66 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 6
birdcatdog
birddogcat
catbirddog
catdogbird
dogbirdcat
dogcatbird
```

Step 2:

A.

```
┌──(kali㉿kali)-[~]
└─$ crunch 1 3 -o weissman.txt
Crunch will now generate the following amount of data: 72384 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 18278

crunch: 100% completed generating output
```

B.



C.

```
┌──(kali㉿kali)-[~]
└─$ cat /usr/share/crunch/charset.lst
# charset configuration file for winrtgen v1.2 by Massimiliano Montoro (mao@oxid.it)
# compatible with rainbowcrack 1.1 and later by Zhu Shuanglei <shuanglei@hotmail.com>

hex-lower                     = [0123456789abcdef]
hex-upper                     = [0123456789ABCDEF]

numeric                       = [0123456789]
numeric-space                 = [0123456789 ]

symbols14                     = [!@#$%^&*()-_+=]
symbols14-space               = [!@#$%^&*()-_+= ]

symbols-all                   = [!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/]
symbols-all-space             = [!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/ ]

ualpha                        = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
ualpha-space                  = [ABCDEFGHIJKLMNOPQRSTUVWXYZ ]
ualpha-numeric                = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
ualpha-numeric-space          = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ]
ualpha-numeric-symbol14       = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=]
ualpha-numeric-symbol14-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+= ]
ualpha-numeric-all            = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/]
ualpha-numeric-all-space      = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/
]

lalpha                        = [abcdefghijklmnopqrstuvwxyz]
lalpha-space                  = [abcdefghijklmnopqrstuvwxyz ]
lalpha-numeric                = [abcdefghijklmnopqrstuvwxyz0123456789]
lalpha-numeric-space          = [abcdefghijklmnopqrstuvwxyz0123456789 ]
lalpha-numeric-symbol14       = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=]
lalpha-numeric-symbol14-space = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+= ]
lalpha-numeric-all            = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/]
lalpha-numeric-all-space      = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/
]

mixalpha                    = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ]
mixalpha-space              = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ ]
mixalpha-numeric            = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
mixalpha-numeric-space      = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ]
```

```
#########################
# Lowercase             #
#########################
lalpha-sv                       = [abcdefghijklmnopqrstuvwxyzåäö]
lalpha-space-sv                 = [abcdefghijklmnopqrstuvwxyzåäö ]
lalpha-numeric-sv               = [abcdefghijklmnopqrstuvwxyzåäö0123456789]
lalpha-numeric-space-sv         = [abcdefghijklmnopqrstuvwxyzåäö0123456789 ]
lalpha-numeric-symbol14-sv      = [abcdefghijklmnopqrstuvwxyzåäö0123456789!@#$%^&*()-_+=]
lalpha-numeric-symbol14-space-sv = [abcdefghijklmnopqrstuvwxyzåäö0123456789!@#$%^&*()-_+= ]
lalpha-numeric-all-sv           = [abcdefghijklmnopqrstuvwxyzåäö0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<
>,.?/]
lalpha-numeric-all-space-sv     = [abcdefghijklmnopqrstuvwxyzåäö0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<
>,.?/ ]

#########################
# Mixcase               #
#########################
mixalpha-sv                     = [abcdefghijklmnopqrstuvwxyzåäöABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ]
mixalpha-space-sv               = [abcdefghijklmnopqrstuvwxyzåäöABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ ]
mixalpha-numeric-sv             = [abcdefghijklmnopqrstuvwxyzåäöABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ0123456789]
mixalpha-numeric-space-sv       = [abcdefghijklmnopqrstuvwxyzåäöABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ0123456789
]
mixalpha-numeric-symbol14-sv    = [abcdefghijklmnopqrstuvwxyzåäöABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ0123456789!
@#$%^&*()-_+=]
mixalpha-numeric-symbol14-space-sv = [abcdefghijklmnopqrstuvwxyzåäöABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ012345
6789!@#$%^&*()-_+= ]
mixalpha-numeric-all-sv         = [abcdefghijklmnopqrstuvwxyzåäöABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ0123456789!
@#$%^&*()-_+=~`[]{}|\:;"'◇,.?/]
mixalpha-numeric-all-space-sv = [abcdefghijklmnopqrstuvwxyzåäöABCDEFGHIJKLMNOPQRSTUVWXYZÅÄÖ0123456789!
@#$%^&*()-_+=~`[]{}|\:;"'◇,.?/ ]
```

D.

```
┌──(kali㉿kali)-[~]
└─$ crunch 8 8 -f /usr/share/crunch/charset.lst mixalpha-numeric
Crunch will now generate the following amount of data: 1965060950264064 bytes
1874028158 MB
1830105 GB
1787 TB
1 PB
Crunch will now generate the following number of lines: 218340105584896
```

```
aaaaenjR
aaaaenjS
aaaaenjT
aaaaenjU
aaaaenjV
aaaaenjW
aaaaenjX
aaaaenjY
aaaaenjZ
aaaaenj0
aaaaenj1
aaaaenj2
aaaaenj3
aaaaenj4
aaaaenj5
aaaaenj6
aaaaenj7
aaaaenj8
aaaaenj9
aaaaenka
aaaaenkb
aaaaenkc
aaaaenkd
aaaaenke
aaaaenkf
aaaaenkg
aaaaenkh
aaaaenki
aaaaenkj
aaaaenkk
aaaaenkl
aaaaenkm
aaaaenkn
aaaaenko
aaaaenkp
aaaaenkq
aaaae^C
aaaaenCy
Crunch ending at aaaaenCy
```

We can see the alphabet and numbers together getting mixed by crunch.

E.

```
┌──(kali㉿kali)-[~]
└─$ crunch 8 8 -f /usr/share/crunch/charset.lst mixalpha-numeric-all-space
Crunch will now generate the following amount of data: 59707838816015625 bytes
56941832366 MB
55607258 GB
54303 TB
53 PB
Crunch will now generate the following number of lines: 6634204312890625
```

```
aaaabw;,
aaaabw;.
aaaabw;?
aaaabw;/
aaaabw;
aaaabw"a
aaaabw"b
aaaabw"c
aaaabw"d
aaaabw"e
aaaabw"f
aaaabw"g
aaaabw"h
aaaabw"i
aaaabw"j
aaaabw"k
aaaabw"l
aaaabw"m
aaaabw"n
aaaabw"o
aaaabw"p
aaaabw"q
aaaabw"r
aaaabw"s
aaaabw"t
aaaabw"u
aaaabw"v
aaaabw"w
aaaabw"x
aaaabw"y
aaaabw"z
aaaabw"A
aaaabw"B
aaaabw"C
aaaabw"D
```

Mixture of symbols, Numbers and all types of characters in this case.

F.

```
┌──(kali㉿kali)-[~]
└─$ crunch 8 8 -t @@@@0415 -f /usr/share/crunch/charset.lst mixalpha-numeric-all-space
Crunch will now generate the following amount of data: 733055625 bytes
699 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 81450625
```

```
bu.S0415
bu.T0415
bu.U0415
bu.V0415
bu.W0415
bu.X0415
bu.Y0415
bu.Z0415
bu.00415
bu.10415
bu.20415
bu.30415
bu.40415
bu.50415
bu.60415
bu.70415
bu.80415
bu.90415
bu.!0415
bu.@0415
bu.#0415
bu.$0415
bu.%0415
bu.^0415
bu.&0415
bu.*0415
bu.(0415
bu.)0415
bu.-0415
bu._0415
bu.+0415
bu.=0415
bu.~0415
bu.`0415
bu.[0415
bu.]0415
bu.{0^C
bvi[0415
Crunch ending at bvi[0415
```

G.

```
┌──(kali㉿kali)-[~]
└─$ crunch 8 8 -t alice@@@ -f /usr/share/crunch/charset.lst mixalpha-numeric-all-space
Crunch will now generate the following amount of data: 7716375 bytes
7 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 857375
```


alice`lR
alice`lS
alice`lT
alice`lU
alice`lV
alice`lW
alice`lX
alice`lY
alice`lZ
alice`l0
alice`l1
alice`l2
alice`l3
alice`l4
alice`l5
alice`l6
alice`l7
alice`l8
alice`l9
alice`l!
alice`l@
alice`l#
alice`l$
alice`l%
alice`l^
alice`l&
alice`l*
alice`l(
alice`l)
alice`l-
alice`l_
alice`l+
alice`l=
alice`l~
alice`l`
alice`l[
alice`l]
alice`l{
alice`l}
alice`l|
alice`l\
alice`l:
alice`l;
alice`l"
alice`l'
"the quieter you become, the more you ar

Step 3

A.

```
└─$ sudo adduser arun
[sudo] password for kali:
Adding user `arun' ...
Adding new group `arun' (1003) ...
Adding new user `arun' (1003) with group `arun (1003)' ...
Creating home directory `/home/arun' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for arun
Enter the new value, or press ENTER for the default
        Full Name []: arun
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
Adding new user `arun' to supplemental / extra groups `users' ...
Adding user `arun' to group `users' ...

┌──(kali㉿kali)-[~]
└─$ sudo adduser adam
Adding user `adam' ...
Adding new group `adam' (1004) ...
Adding new user `adam' (1004) with group `adam (1004)' ...
Creating home directory `/home/adam' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for adam
Enter the new value, or press ENTER for the default
        Full Name []: adam
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
Adding new user `adam' to supplemental / extra groups `users' ...
Adding user `adam' to group `users' ...
```

B.

```
┌──(kali㉿kali)-[~]
└─$ sudo unshadow /etc/passwd /etc/shadow > rochester4.txt
```

C.

```
┌──(kali㉿kali)-[~]
└─$ crunch 1 3 -o crunch_list3.txt
Crunch will now generate the following amount of data: 72384 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 18278

crunch: 100% completed generating output
```

D.

```
┌──(kali㉿kali)-[~]
└─$ sudo john --wordlist=crunch_list3.txt --format=crypt rochester4.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Remaining 2 password hashes with 2 different salts
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for al
l loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
cat              (adam)
dog              (arun)
2g 0:00:00:21 DONE (2024-04-01 18:44) 0.09350g/s 148.1p/s 246.8c/s 246.8C/s dne..dqv
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Step 4:

A.

```
┌──(kali㉿kali)-[~]
└─$ sudo passwd scott
New password:
Retype new password:
passwd: password updated successfully
```

B, C.

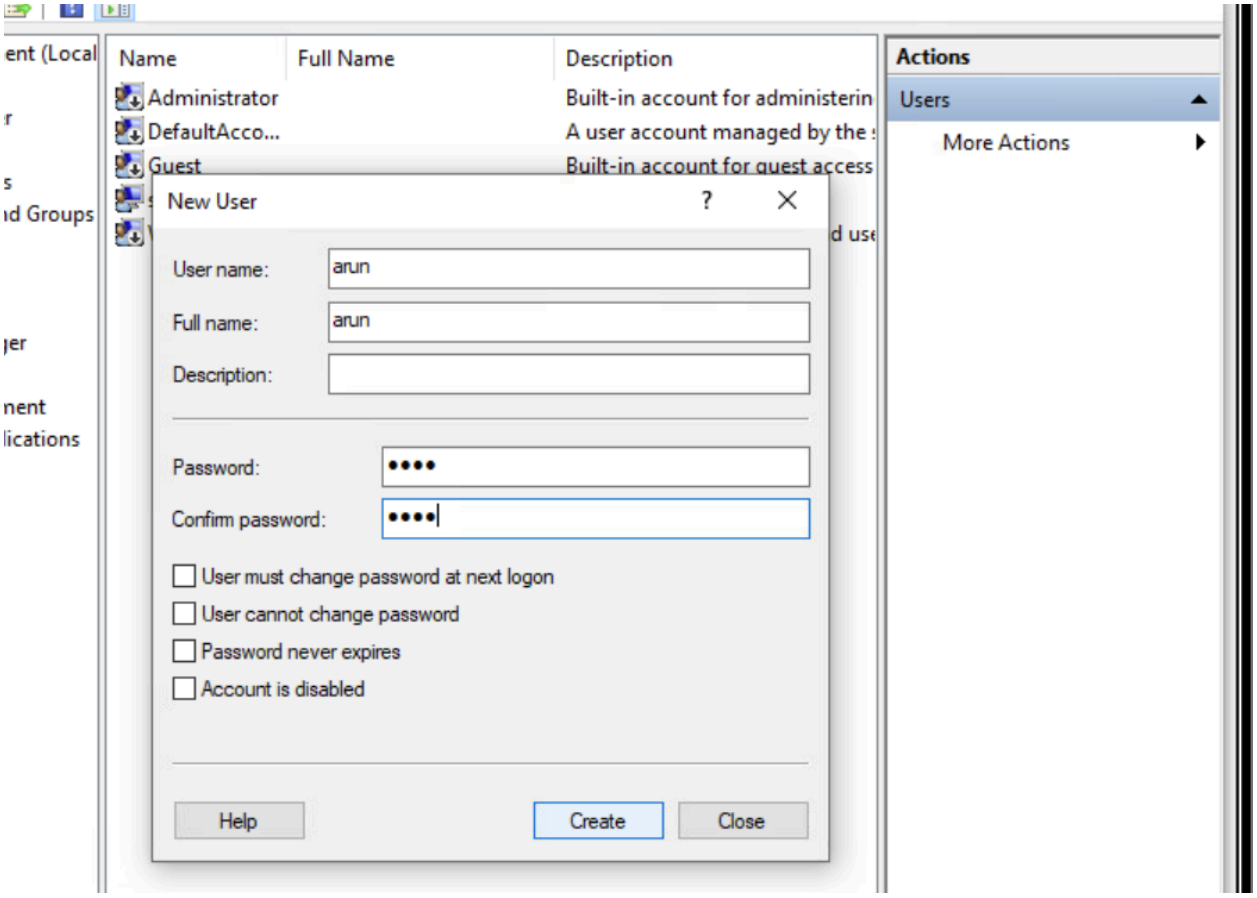
```
┌──(kali㉿kali)-[~]
└─$ sudo unshadow /etc/passwd /etc/shadow > rochester5.txt

┌──(kali㉿kali)-[~]
└─$ sudo crunch 4 4 | sudo john --format=crypt rochester5.txt --stdin
Crunch will now generate the following amount of data: 2284880 bytes
2 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 456976
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (crypt, generic crypt(3) [?/64])
Remaining 1 password hash
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for al
l loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
aabb             (scott)
1g 0:00:00:03  0.3086g/s 29.62p/s 29.62c/s 29.62C/s aaaa..aadr
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

D.

```
┌──(kali㉿kali)-[~]
└─$ sudo crunch 3 3 | sudo john --format=crypt rochester6.txt --stdin
Crunch will now generate the following amount of data: 70304 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 17576
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (crypt, generic crypt(3) [?/64])
Remaining 1 password hash
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for al
l loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
zzz              (scott)
1g 0:00:01:13  0.01353g/s 237.9p/s 237.9c/s 237.9C/s zzs..zzz
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

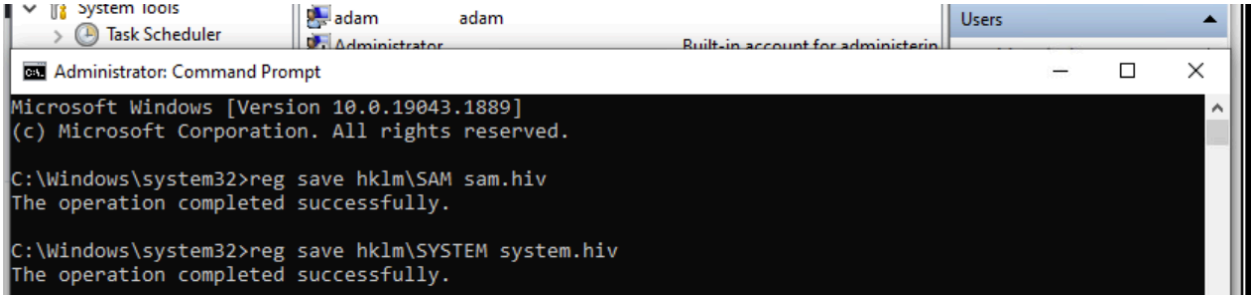
Exercise 11. 03:

Step 2:
F.

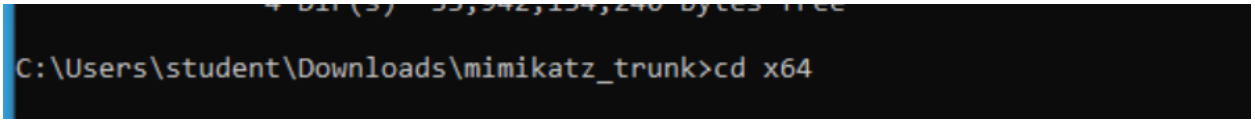


Step 3:

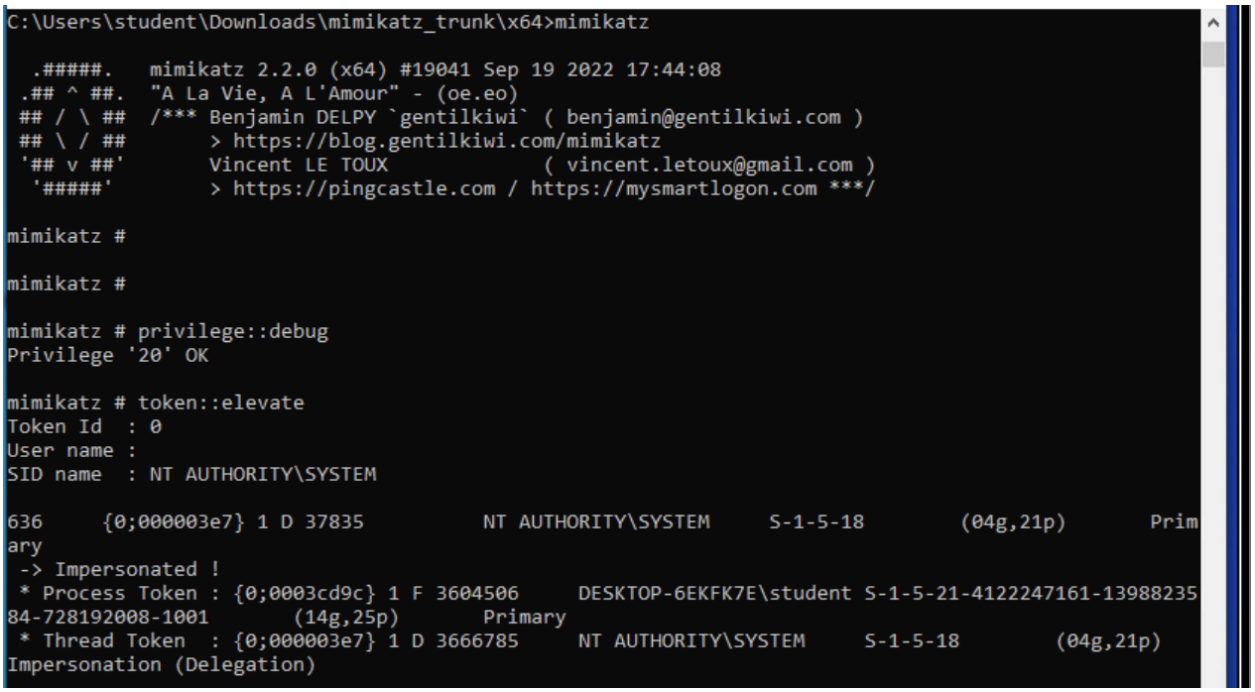
A and B.



C.



D and E.



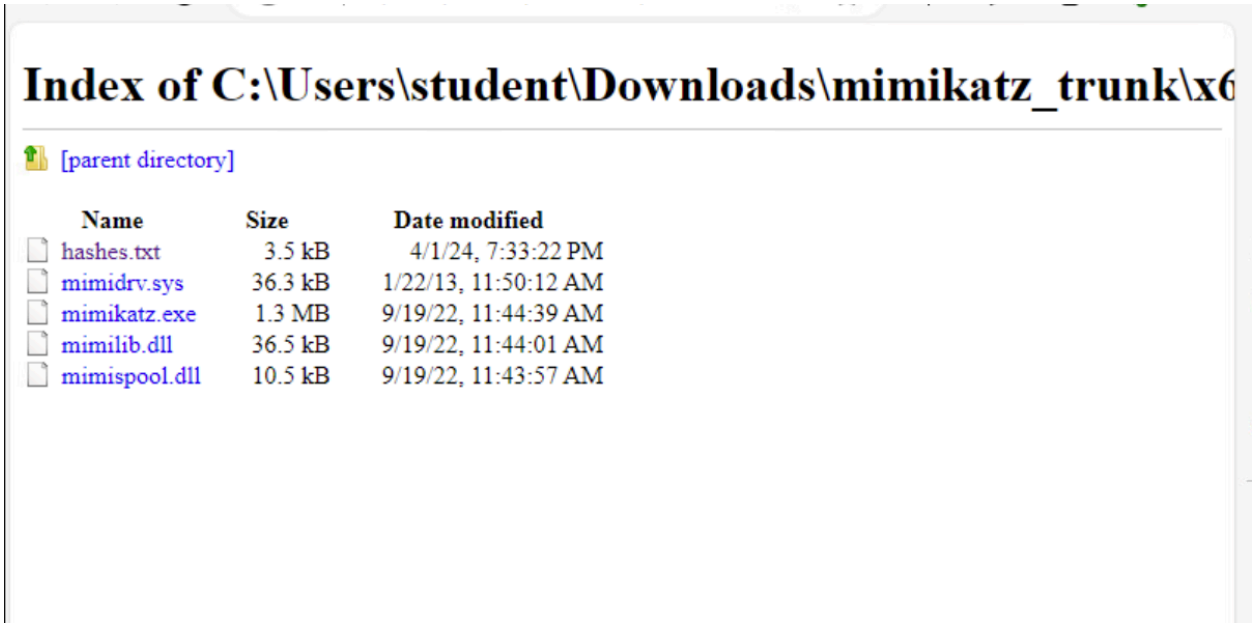
F and G.

```
mimikatz # log hashes.txt
Using 'hashes.txt' for logfile : OK

mimikatz # lsadump::sam sam.hiv system.hiv
Domain : DESKTOP-6EKFK7E
SysKey : 914d2fda6e61f6f6343ec4abd52b091e
Local SID : S-1-5-21-4122247161-1398823584-728192008

SAMKey : 0916a6dcbdb12d446ea296532a66b4bc

RID  : 000001f4 (500)
User : Administrator

RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaultAccount

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 6437fc3942fc8a7822dc27dd814862fd

Supplemental Credentials:
```

```
RID  : 000003ea (1002)
User : arun
  Hash NTLM: 873c6ff2711086ed73d62891ef3d0734

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 9c47dd6b5ce40520ecb050b305f8ed89

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-6EKFK7Earun
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 6c6504702324a1b75540ba296fc2904b626194eb26a80a20f5005b4fb45b7424
      aes128_hmac       (4096) : 237b7c3d96ed16fcaf5e4bdc440a5b52
      des_cbc_md5       (4096) : dffe2a8cd35eb080

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : DESKTOP-6EKFK7Earun
    Credentials
      des_cbc_md5       : dffe2a8cd35eb080
```

```
RID  : 000003eb (1003)
User : adam
  Hash NTLM: accc357c875bd2bf58878b02fb123185

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 0c7c6b7e670dfc4709f8dcf0c057885b

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-6EKFK7Eadam
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : a703318fae37af32cc79b03cc2d7adf6cc85b905c39448ee33d2ce26ce76ca9d
      aes128_hmac       (4096) : e84fd60fabd17b249de1ffdfc6eb86fe
      des_cbc_md5       (4096) : 26da9413b5d9dfe9

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : DESKTOP-6EKFK7Eadam
    Credentials
      des_cbc_md5       : 26da9413b5d9dfe9
```

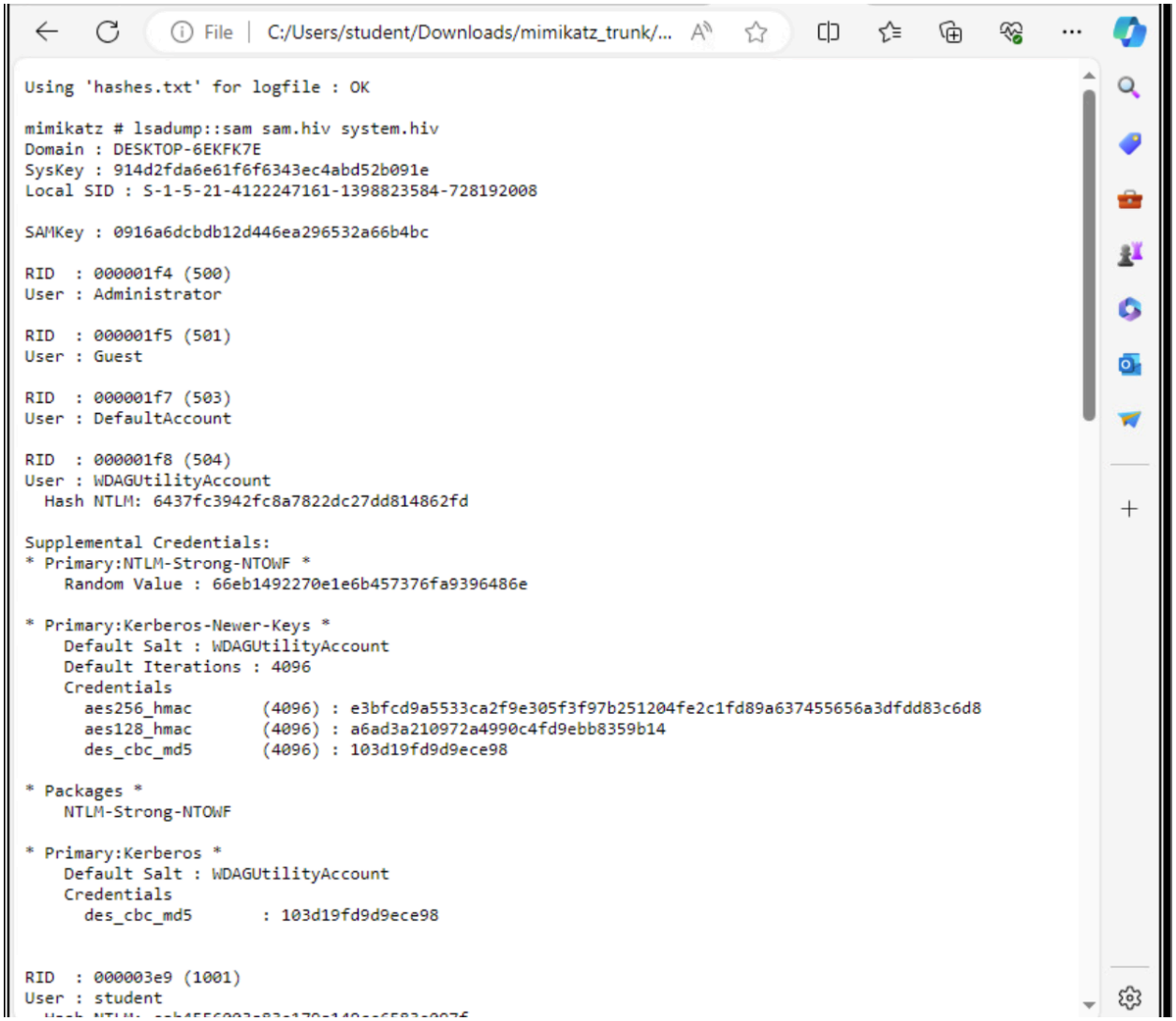
H.



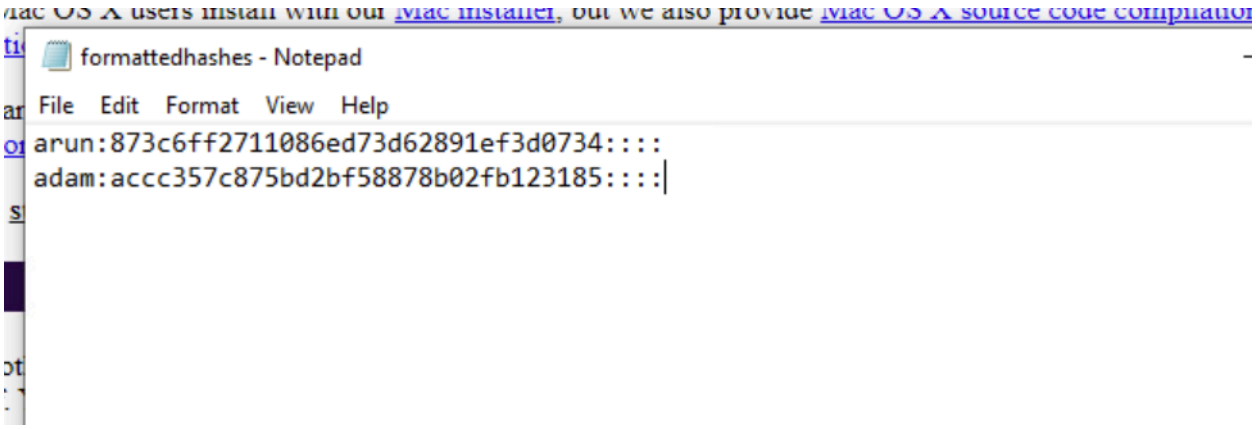
Index of C:\Users\student\Downloads\mimikatz_trunk\x6

[parent directory]

Name	Size	Date modified
hashes.txt	3.5 kB	4/1/24, 7:33:22 PM
mimidrv.sys	36.3 kB	1/22/13, 11:50:12 AM
mimikatz.exe	1.3 MB	9/19/22, 11:44:39 AM
mimilib.dll	36.5 kB	9/19/22, 11:44:01 AM
mimispool.dll	10.5 kB	9/19/22, 11:43:57 AM


File | C:/Users/student/Downloads/mimikatz_trunk/...
Using 'hashes.txt' for logfile : OK
mimikatz # lsadump::sam sam.hiv system.hiv
Domain : DESKTOP-6EKFK7E
SysKey : 914d2fda6e61f6f6343ec4abd52b091e
Local SID : S-1-5-21-4122247161-1398823584-728192008
SAMKey : 0916a6dcbdb12d446ea296532a66b4bc
RID : 000001f4 (500)
User : Administrator
RID : 000001f5 (501)
User : Guest
RID : 000001f7 (503)
User : DefaultAccount
RID : 000001f8 (504)
User : WDAGUtilityAccount
Hash NTLM: 6437fc3942fc8a7822dc27dd814862fd
Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
Random Value : 66eb1492270e1e6b457376fa9396486e
* Primary:Kerberos-Newer-Keys *
Default Salt : WDAGUtilityAccount
Default Iterations : 4096
Credentials
aes256_hmac (4096) : e3bfcd9a5533ca2f9e305f3f97b251204fe2c1fd89a637455656a3dfdd83c6d8
aes128_hmac (4096) : a6ad3a210972a4990c4fd9ebb8359b14
des_cbc_md5 (4096) : 103d19fd9d9ece98
* Packages *
NTLM-Strong-NTOWF
* Primary:Kerberos *
Default Salt : WDAGUtilityAccount
Credentials
des_cbc_md5 : 103d19fd9d9ece98
RID : 000003e9 (1001)
User : student

I.


formattedhashes - Notepad
File Edit Format View Help
arun:873c6ff2711086ed73d62891ef3d0734::::
adam:accc357c875bd2bf58878b02fb123185::::

J and K.

```
┌──(kali㉿kali)-[~]
└─$ leafpad windowshashes.txt

┌──(kali㉿kali)-[~]
└─$ cat windowshashes.txt
arun:873c6ff2711086ed73d62891ef3d0734::::
adam:accc357c875bd2bf58878b02fb123185::::
```

L.

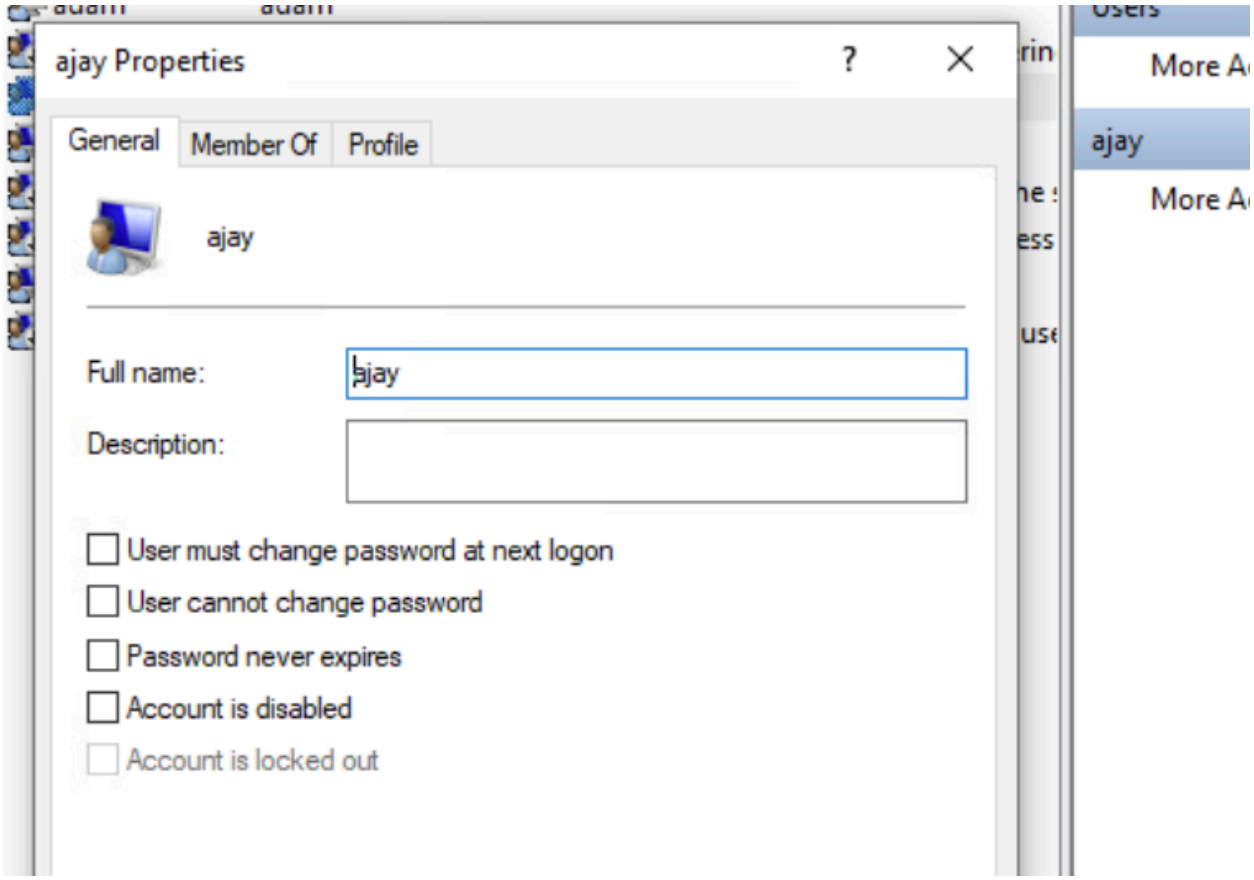
```
┌──(kali㉿kali)-[~]
└─$ sudo crunch 4 4 | sudo john --format=NT windowshashes.txt --stdin
[sudo] password for kali:
Crunch will now generate the following amount of data: 2284880 bytes
2 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 456976
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type
Press Ctrl-C to abort, or send SIGUSR1 to john process for status
aabc             (adam)
aabb             (arun)
2g 0:00:00:02  0.7067g/s 33.92p/s 33.92c/s 67.84C/s aaaa..aadr
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

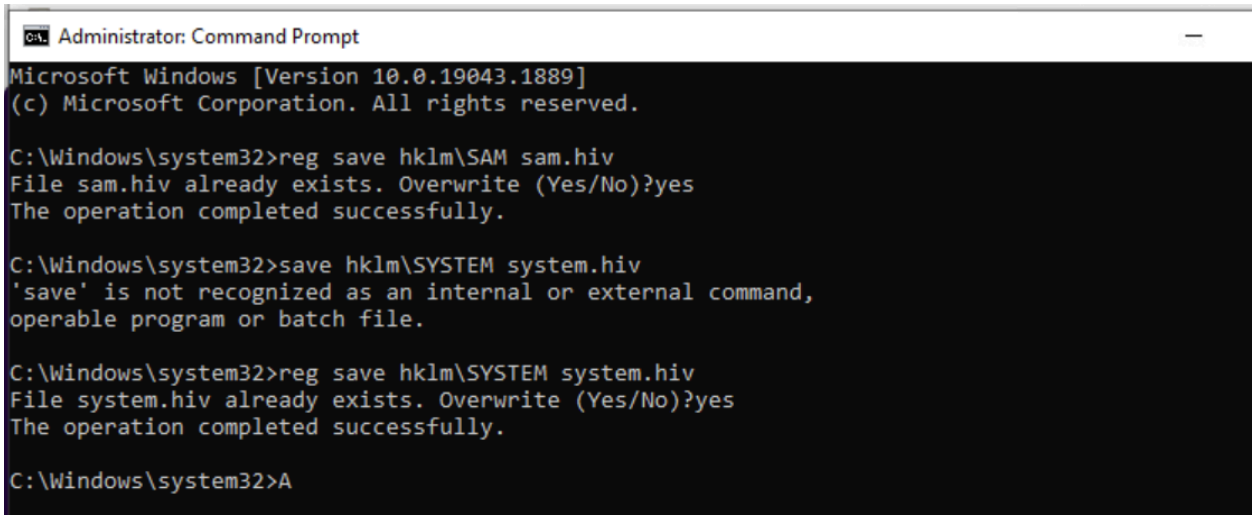
M, N, O.

I did crack both the passwords at the same time, as I listed both the hashes in windowshashes.txt file:

```
┌──(kali㉿kali)-[~]
└─$ sudo john --show --format=NT windowshashes.txt
arun:aabb::::
adam:aabc::::

2 password hashes cracked, 0 left

┌──(kali㉿kali)-[~]
└─$ sudo john --format=NT windowshashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 128/128 AVX 4×3])
No password hashes left to crack (see FAQ)

┌──(kali㉿kali)-[~]
└─$ sudo john --show --format=NT windowshashes.txt
arun:aabb::::
adam:aabc::::

2 password hashes cracked, 0 left
```

Step 4:


ajay Properties
General
Member Of
Profile
ajay
Full name:
ajay
Description:
User must change password at next logon
User cannot change password
Password never expires
Account is disabled
Account is locked out
More A
ajay
More A


Administrator: Command Prompt
Microsoft Windows [Version 10.0.19043.1889]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\system32>reg save hklm\SAM sam.hiv
File sam.hiv already exists. Overwrite (Yes/No)?yes
The operation completed successfully.
C:\Windows\system32>save hklm\SYSTEM system.hiv
'save' is not recognized as an internal or external command,
operable program or batch file.
C:\Windows\system32>reg save hklm\SYSTEM system.hiv
File system.hiv already exists. Overwrite (Yes/No)?yes
The operation completed successfully.
C:\Windows\system32>A

```
RID  : 000003ec (1004)
User : ajay
  Hash NTLM: 4039730e1bf6e10dd01eaac983db4d7c

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 5be825fcd0b1f3efc0e20fc3ae2fca8b

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-6EKFK7Eajay
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 2cde6c9d74865d74672863433b5afb00645f1a1e07aaafa24c5b7af275f86e52
      aes128_hmac       (4096) : 88fb2ddb06f52829738063064088f462
      des_cbc_md5       (4096) : 0efd08dc6d38dc9d

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : DESKTOP-6EKFK7Eajay
    Credentials
      des_cbc_md5       : 0efd08dc6d38dc9d
```

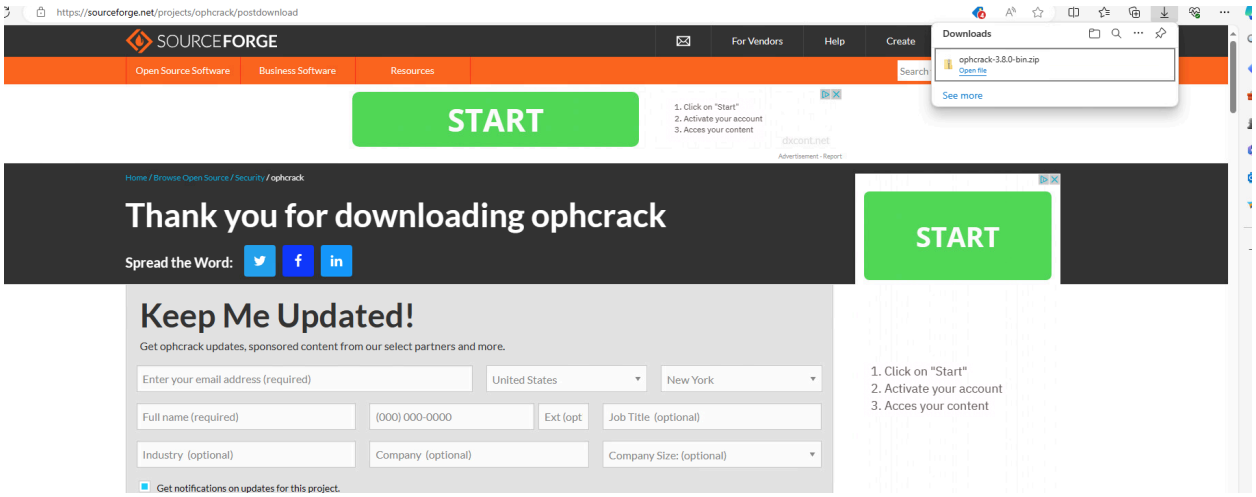
```
┌──(kali㉿kali)-[~]
└─$ leafpad windowshashes.txt

┌──(kali㉿kali)-[~]
└─$ cat windowshashes.txt
arun:873c6ff2711086ed73d62891ef3d0734::::
adam:accc357c875bd2bf58878b02fb123185::::
ajay:4039730e1bf6e10dd01eaac983db4d7c::::
```

We cracked the password for new user ajay successfully which is “alice”.

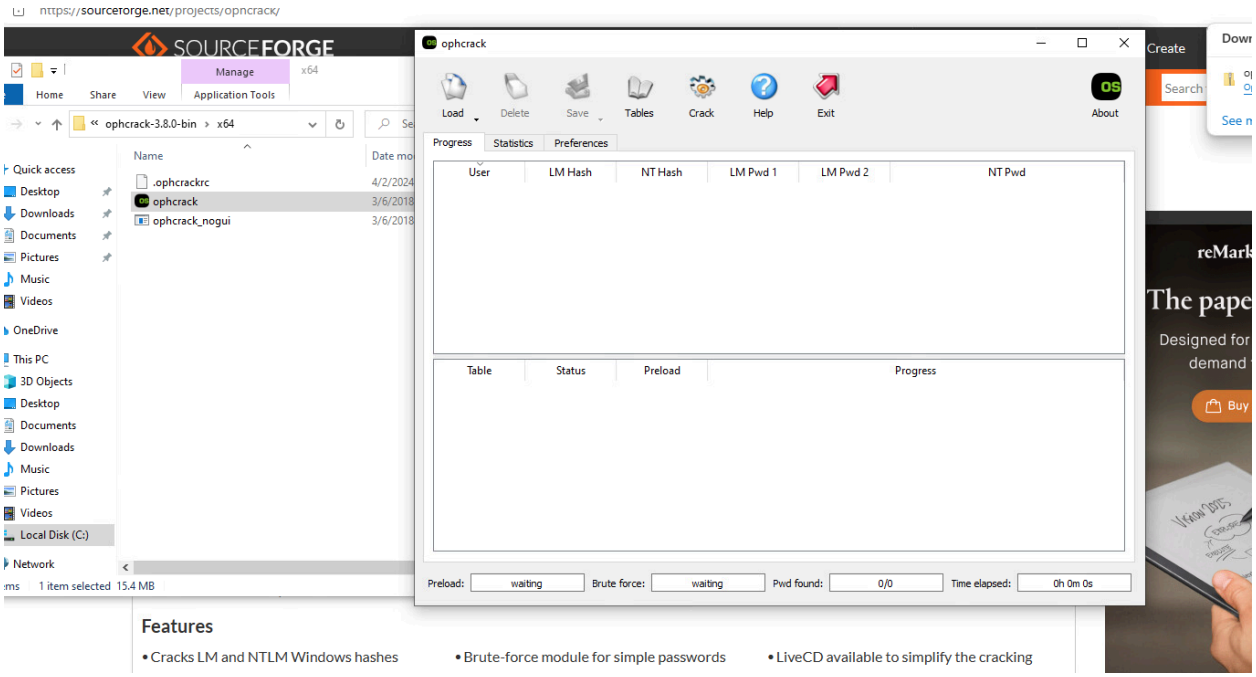
```
┌──(kali㉿kali)-[~]
└─$ sudo john --format=NT windowshashes.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (NT [MD4 128/128 AVX 4×3])
Remaining 1 password hash
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 12 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
alice            (ajay)
1g 0:00:00:00 DONE 2/3 (2024-04-01 20:05) 100.0g/s 122800p/s 122800c/s 122800C/s purple..larry
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Exercise 11. 04

Step 1:



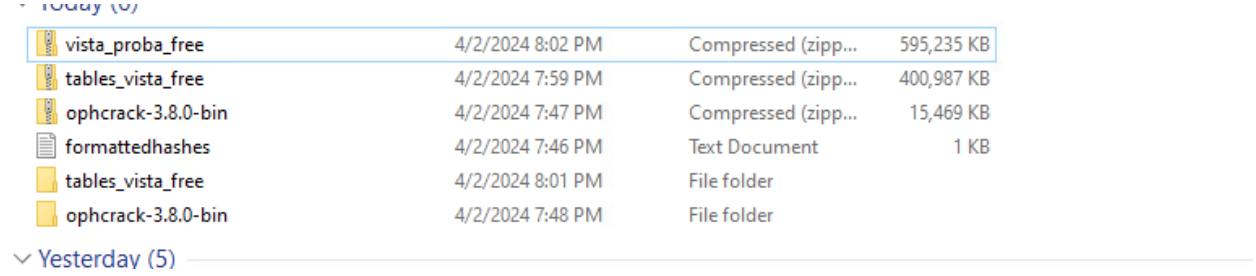
B,C, D.



E,F,G

Downloading those files one by one.

H.



I,J,K.

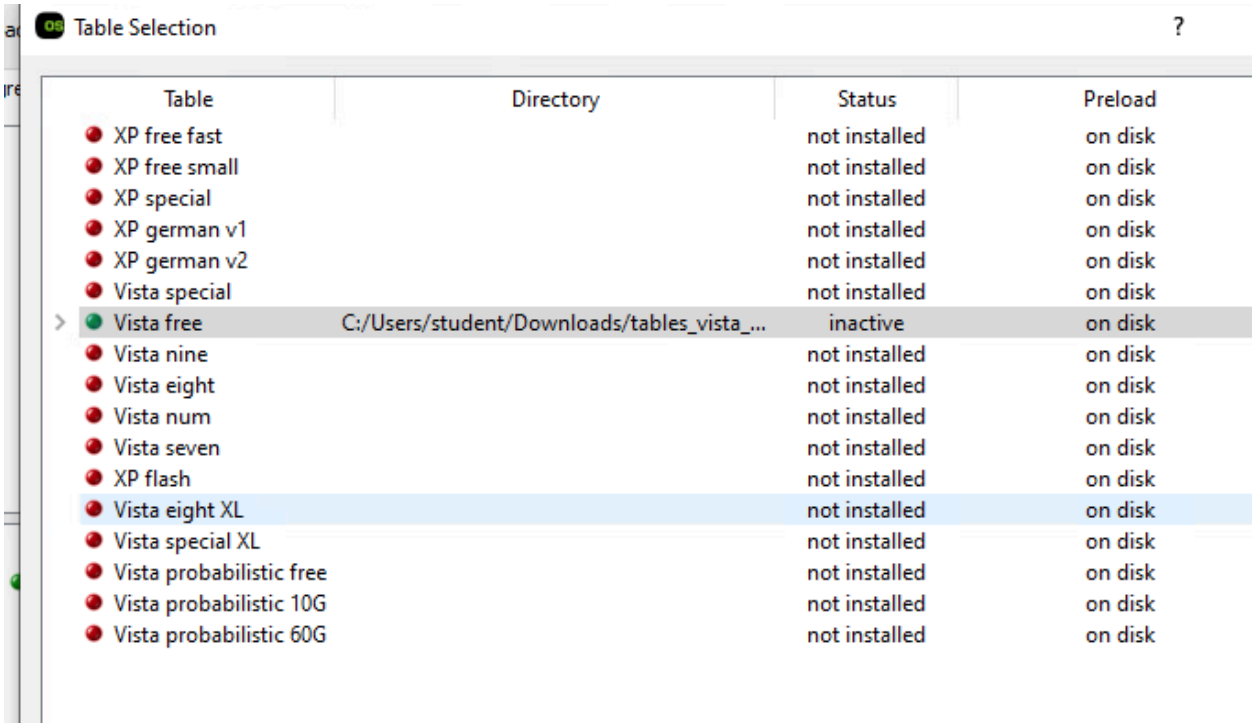


Table Selection
?
Table
Directory
Status
Preload
XP free fast
not installed
on disk
XP free small
not installed
on disk
XP special
not installed
on disk
XP german v1
not installed
on disk
XP german v2
not installed
on disk
Vista special
not installed
on disk
Vista free
C:/Users/student/Downloads/tables_vista_...
inactive
on disk
Vista nine
not installed
on disk
Vista eight
not installed
on disk
Vista num
not installed
on disk
Vista seven
not installed
on disk
XP flash
not installed
on disk
Vista eight XL
not installed
on disk
Vista special XL
not installed
on disk
Vista probabilistic free
not installed
on disk
Vista probabilistic 10G
not installed
on disk
Vista probabilistic 60G
not installed
on disk

L,M

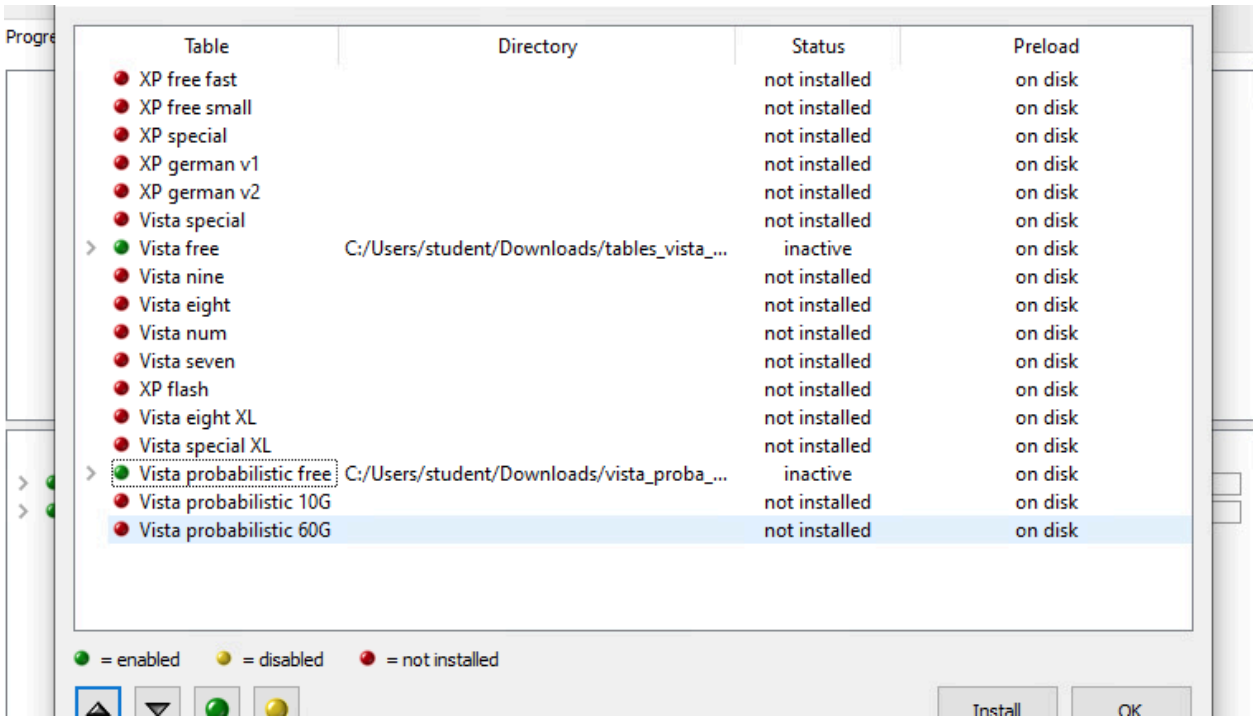
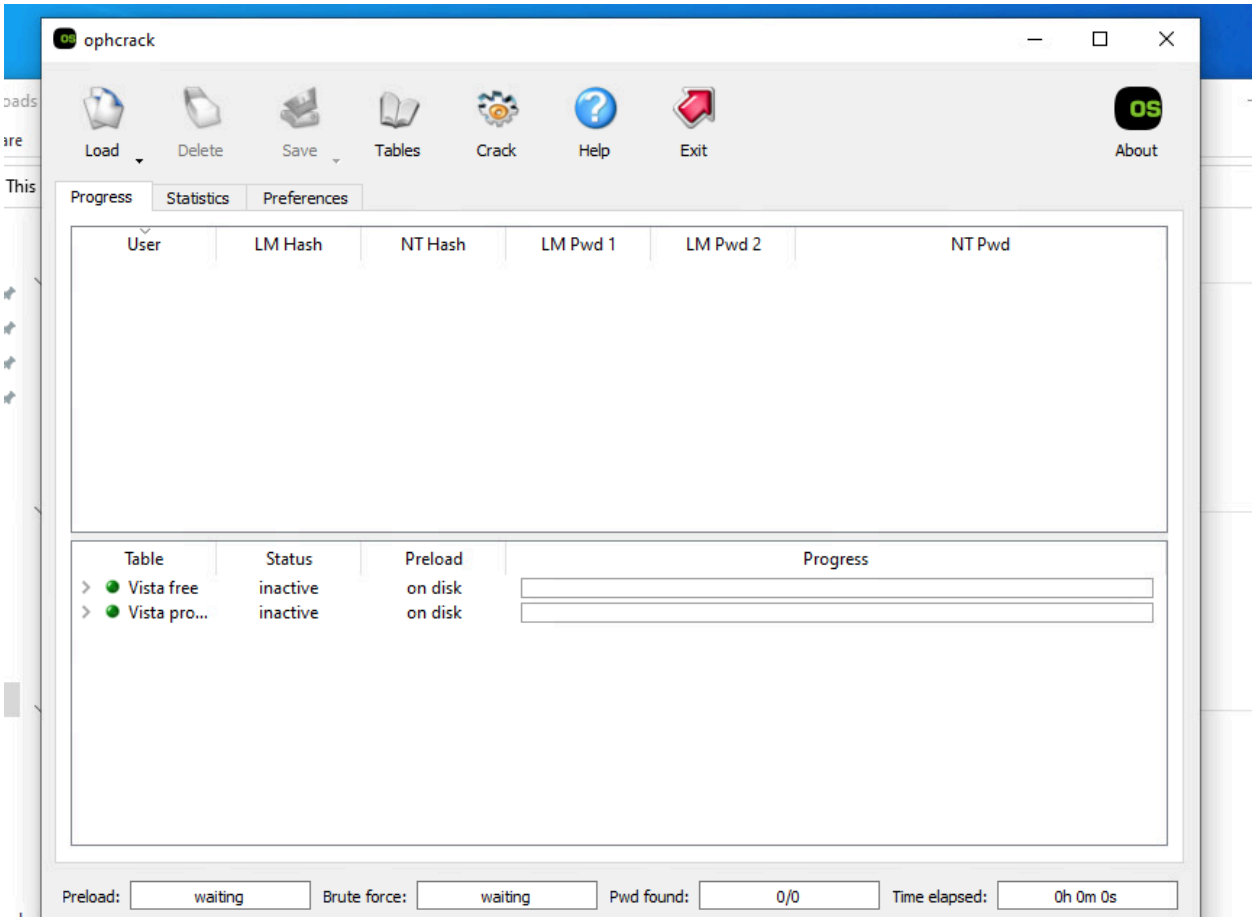


Table
Directory
Status
Preload
XP free fast
not installed
on disk
XP free small
not installed
on disk
XP special
not installed
on disk
XP german v1
not installed
on disk
XP german v2
not installed
on disk
Vista special
not installed
on disk
Vista free
C:/Users/student/Downloads/tables_vista_...
inactive
on disk
Vista nine
not installed
on disk
Vista eight
not installed
on disk
Vista num
not installed
on disk
Vista seven
not installed
on disk
XP flash
not installed
on disk
Vista eight XL
not installed
on disk
Vista special XL
not installed
on disk
Vista probabilistic free
C:/Users/student/Downloads/vista_proba_...
inactive
on disk
Vista probabilistic 10G
not installed
on disk
Vista probabilistic 60G
not installed
on disk
= enabled
= disabled
= not installed
Install
OK

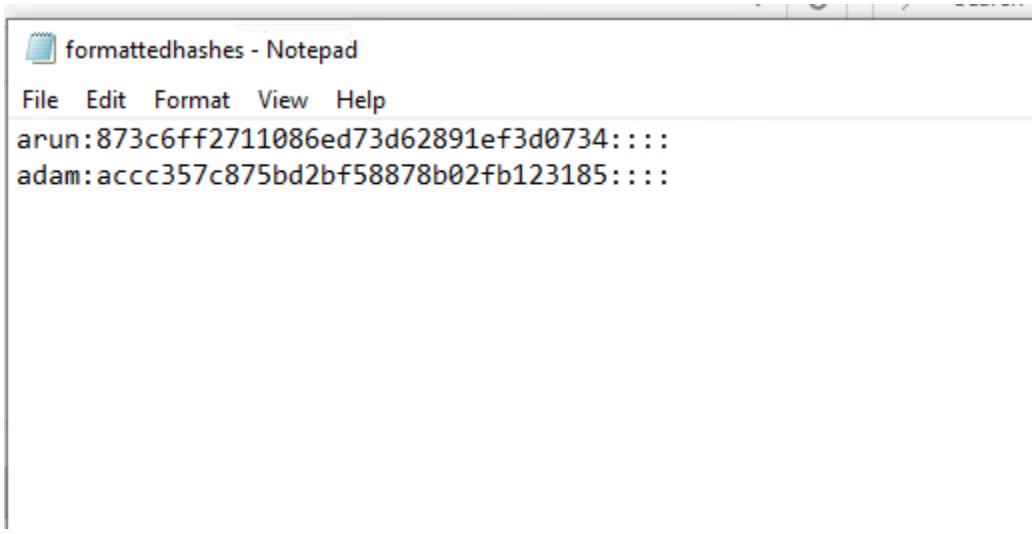
N.



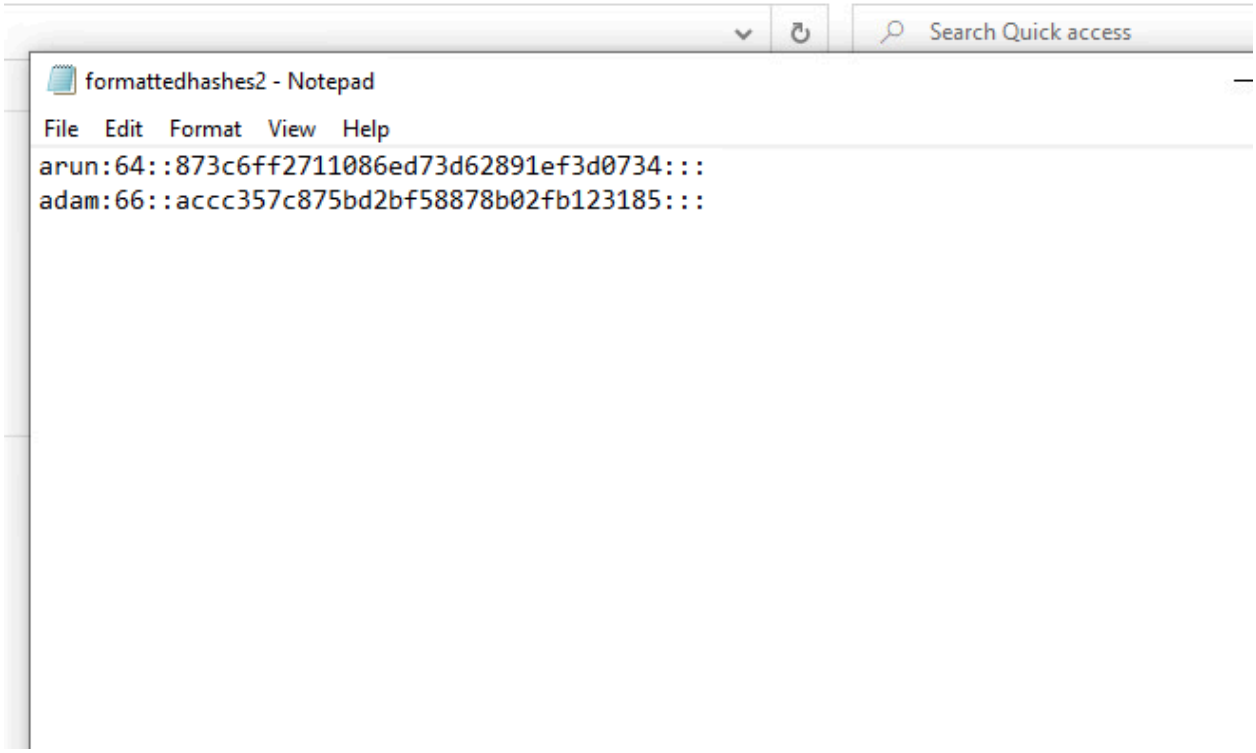
Step 2:

A.

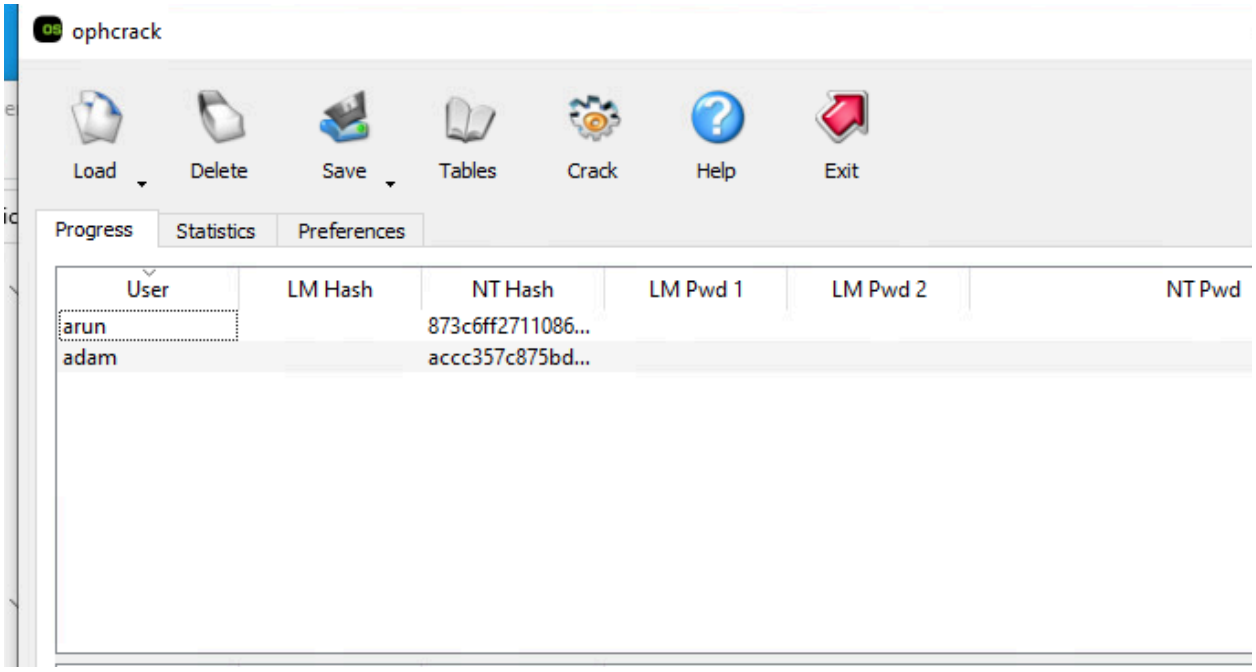
Hashes we got from mimikatz.



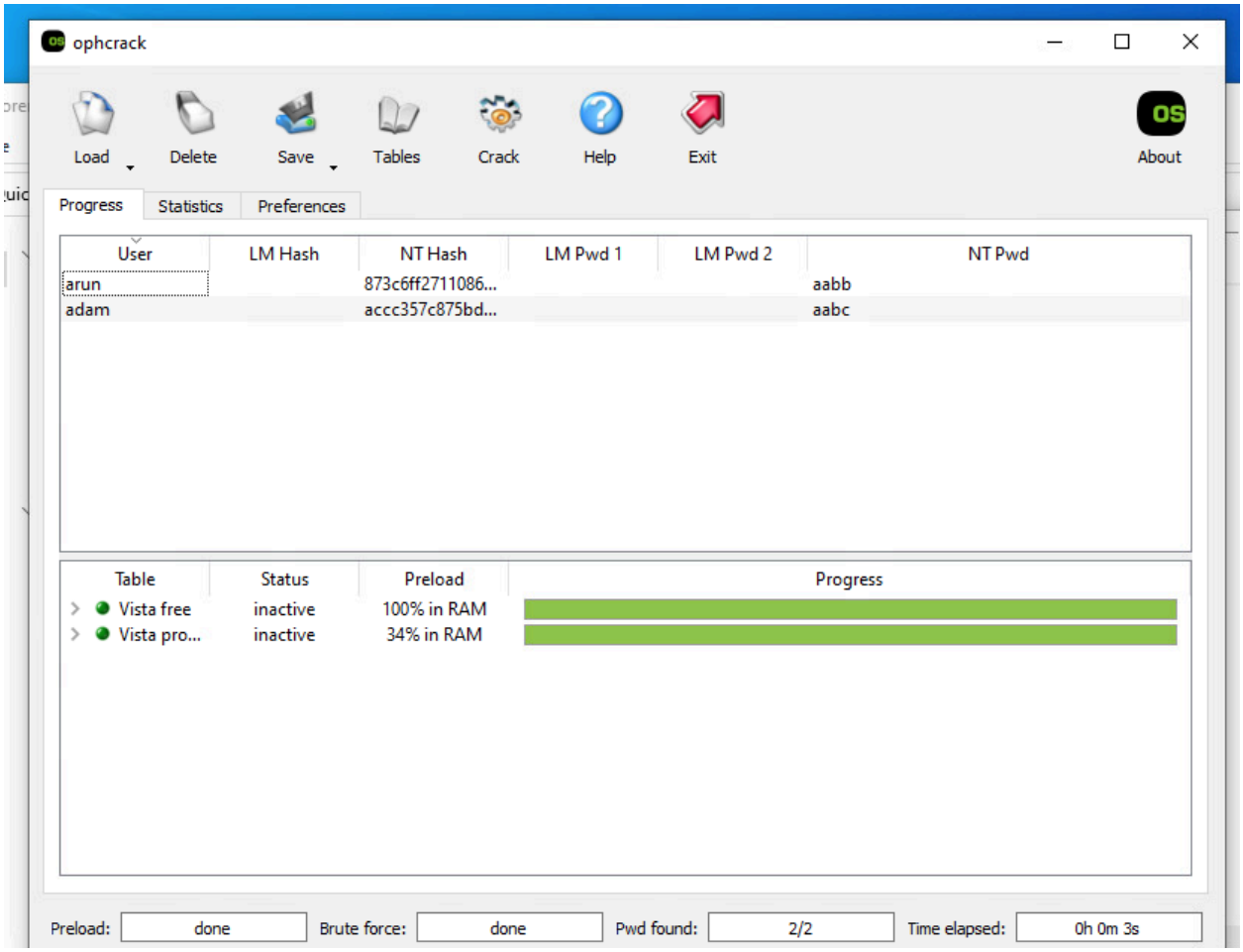
B.


Search Quick access
formattedhashes2 - Notepad
File Edit Format View Help
arun:64::873c6ff2711086ed73d62891ef3d0734:::
adam:66::accc357c875bd2bf58878b02fb123185:::

C.


ophcrack
Load
Delete
Save
Tables
Crack
Help
Exit
Progress
Statistics
Preferences
User
LM Hash
NT Hash
LM Pwd 1
LM Pwd 2
NT Pwd
arun
873c6ff2711086...
adam
accc357c875bd...

D, E.



It was so quick and we finally cracked the password.

Lab Analysis:

1. **What is a dictionary attack?**
   As name says, this attack uses the dictionary which contains predefined words which are commonly used, the attack is automated by the tool, in this case john the ripper, will map out all the different possibilities of the password to the username and brute forcing. This type of attack is not always the best until the password is common word.
2. **What is brute force attack?**
   Brute force attack is better than dictionary attack in a way of guessing password, for example crunch which combines all the possibilities of the word to get the password, however this attack may be effective, but this take lots and lots of time. Because of brute forcing it consumes lot of storage and power and not efficient way of finding the password. If the length of the password is high and it will take forever to brute force.
3. **What is rainbow table attack?**
   This attack is interesting and combines both dictionary and brute force I would say, here we use hash functions and reduction functions where words can be created from other words and it will start growing the database slowly, we need to have all the words to do the rainbow table attack. It will harvest the words on their own through reduction and hash functions. Passwords with hash collisions are vulnerable to this attack and can be exploited. However as brute force attack, this attack also needs significant power and storage to handle and maintain.
4. **Why should passwords always be stored in hashed format?**
   Passwords should be stored in hashed format for security purpose, so that while transferring data in layer 4, even though the packets are captured, the hash will make no sense.
   Hash function is compared when the user inputs the passwords and login the user for better security of the passwords.
   Hash is irreversible, which is main feature of hash function to be used in the first place. Also hash function with no collision should be used to have better result and protection.

Key Term Quiz:

1. **Salt**
2. **SAM**
3. **Shadow**
4. **Rockyou**
5. **Mimikatz**
6. **John the Ripper**
7. **ophcrack**