TABLE OF CONTENTS:

SECTION 1:

Brute Force Attack

Overview:

Attack/tools→ Crunch, a password generating tool is used to generate a list of potential passwords. Depending on the parameters passed to the crunch command, (uppercase, lowercase, numbers and symbols) can be arranged in custom ways to accommodate a range of password characters. Using another tool called john the ripper, all of these password values generated from the crunch tool are then hashed and compared to the password file containing the hash that we intend to crack. If there is a match, john the ripper ends and the password is output.

Mitigation → Encrypt your data | Limit logins | Use PBKDF2, BECRYPT, ARGON, Scrypt2 hashing

---

Steps to run brute force attack:

Step 1: Run crunch brute force command
Step 2: Check that previous command outputs file on local machine
Step 3: Create a user
Step 4: Combine passwd & shadow files in order to output to text file [ex: bill_info.txt]
Step 5: Cat bill_info.txt → verify that user and corresponding password hash exists in file
Step 6: Run john the ripper together on both files [crunch password file & user passwd/shadow file]

Step 1:
Crunch brute force command
[http://project-rainbowcrack.com/charset.txt] for more info on rainbow table char-sets
[**crunch 8 8 -t @@@@0629 -f /usr/share/rainbowcrack/charset.txt loweralpha-numeric -o crunch_bruteforce.txt**]

```
root@kali:~# crunch 8 8 -t @@@@0629 -f /usr/share/rainbowcrack/charset.txt loweralpha-numeric -o crunch_bruteforce.txt
Crunch will now generate the following amount of data: 15116544 bytes
14 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1679616

crunch: 100% completed generating output
```

Step 2:

Check that previous command outputs file on local machine

```
Applications  Places  Text Editor                Fri 16:28
Open  crunch_bruteforce.txt  Save
bill_info.txt   crunch_bruteforce.txt
aaaa0629
aaab0629
aaac0629
aaad0629
aaae0629
aaaf0629
aaag0629
aaah0629
aaai0629
```

Step 3:

Create a user

```
Applications  Places  Terminal                   Fri 13:05
root@kali: ~
File  Edit  View  Search  Terminal  Help
root@kali:~# adduser bill
Adding user `bill' ...
Adding new group `bill' (1008) ...
Adding new user `bill' (1008) with group `bill' ...
Creating home directory `/home/bill' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
Sorry, passwords do not match
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bill
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
root@kali:~#
```

Step 4:

Combine passwd & shadow files in order to output [bill_info.txt]

Step 5:

Cat bill_info.txt → verify that user and corresponding password hash exists in file

```
dradis:*:133:141::/var/lib/dradis:/usr/sbin/nologin
beef-xss:*:134:142::/var/lib/beef-xss:/usr/sbin/nologin
bill:$6$CTLXlYc2$agqcml50dhkQEkXKpHHchnpjC/yTo7c/5R6aEKDpwQwPR9CYb0wwK0I1YBMmw9xTvck//6epBQiuh4Pn199G30:1008:1008:,,,:/home/bill:/bin/bash
```

Step 6:

run john the ripper on crunch_bruteforce list against [bill_info] passwd and shadow file

---

Dictionary Based Attack

Overview:(notes from class)

Attack/tools→ rockyou.txt, a pre-generated 14 million word list is parsed with john the ripper. The possible password values in the rockyou file are hashed and compared to the password file containing the hash that we intend to crack. If a match exists, the password is cracked and output.

Mitigation → Encrypt your data | Limit logins | Use PBKDF2, BECRYPT, ARGON2 hashing

---

Steps to run dictionary based attack:

Step 1 - Create a user
Step 2 - output passwd and shadow files to new text file
Step 3 - run john the ripper with rock_you dictionary against new text file

Step 1: Create user **[ex: Christine]**

```
root@kali:~#
root@kali:~# adduser christine
Adding user `christine' ...
Adding new group `christine' (1000) ...
Adding new user `christine' (1000) with group `christine' ...
Creating home directory `/home/christine' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for christine
Enter the new value, or press ENTER for the default
        Full Name []: christine
        Room Number []: 07
        Work Phone []: 512 7648
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
root@kali:~#
```

Step 2: combine passwd and shadow files and output to text file **[ex: christine_info.txt]**

```
root@kali:~# unshadow /etc/passwd /etc/shadow > christine_info.txt
root@kali:~#
```

Step 3: run john the ripper with rock_you dictionary against christine_info

```
root@kali:~#
root@kali:~# john --wordlist=rockyou.txt christine_info.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Remaining 1 password hash
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:01:56 0.50% (ETA: 23:38:44) 0g/s 733.0p/s 733.0c/s 733.0C/s ilovereece..hollis1
```

Rainbow Table Attack

Overview(notes from class)**:** Hash each plaintext one by one, but store each generated hash in a sorted table so that you can easily look the hash up later without generating the hashes again. A hash function maps plaintexts to hashes, the reduction function maps hashes to plaintexts**.**
Key point: The chains which make up rainbow tables are chains of one way hash and reduction functions starting at a certain plaintext, and ending at a certain hash. A chain in a rainbow table starts with an arbitrary plaintext, hashes it, reduces the hash to another plaintext, hashes the new plaintext, and so on. The table only stores the starting plaintext, and the final hash you choose to end with, and so a chain with millions of hashes can be represented with only a single starting plaintext, and a single finishing hash. [http://kestas.kuliukas.com/RainbowTables/]
Time memory tradeoff - what you lose in memory, you gain in time.

Steps to run Rainbow table attack:

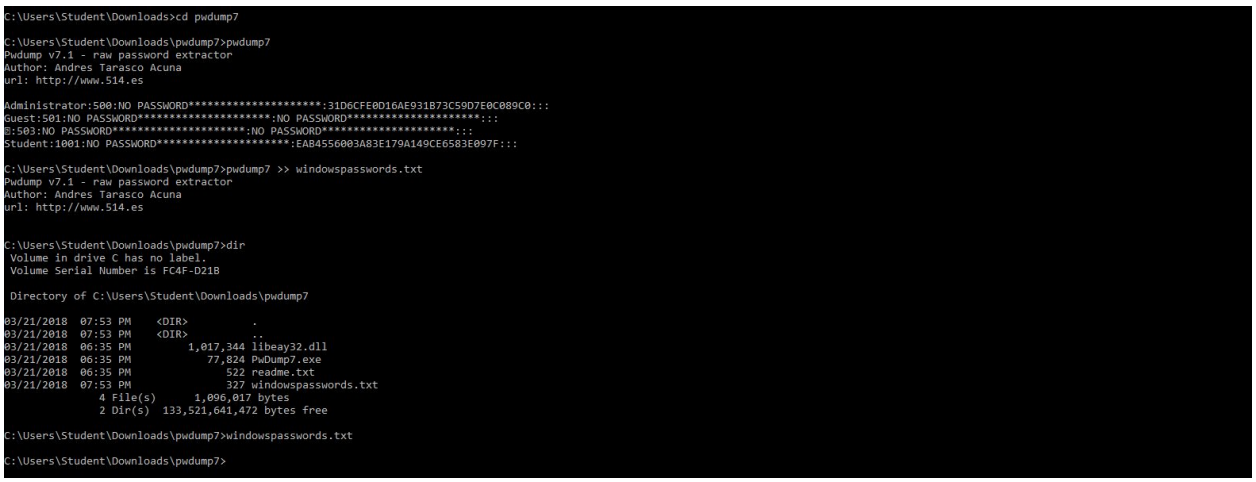
Step 1 - [download ophcrack & ophcrack tables - vista free and vista proba free]
Step 2 - create password dumpfile from pwdump7 console output
Step 3: Verify that pwdump file includes the usernames and password hashes for the local windows machine
Step 4: open pwdump file in ophcrack
Step 5: crack windowspasswords.txt file in 4 seconds

Step 1:
[http://ophcrack.sourceforge.net/] for Ophcrack download (all platforms)
[http://ophcrack.sourceforge.net/tables.php] for both vista free and vista proba free downloads
[www.tarasco.org/security/pwdump_7/] for downloading pwdump
Extract & upload both tables to Ophcrack.
[tables → install → (navigate to/select table file → open) → repeat for both table files]
[see image below for what it should look like when complete] NOTE: Green dots next to file name

Step 2:

Run as admin to create password dumpfile from pwdump7 console output

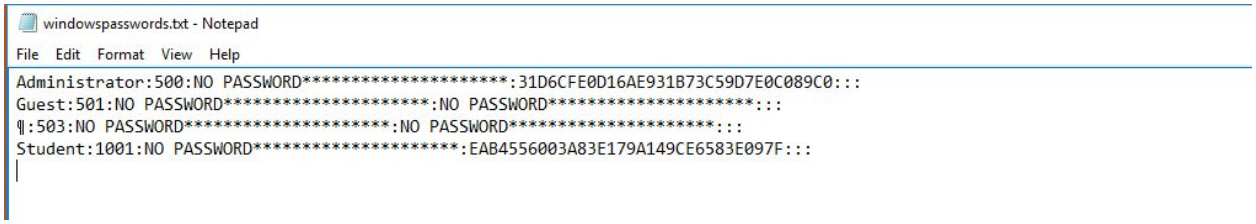
[ex: windowspasswords.txt]



```
C:\Users\Student\Downloads>cd pwdump7

C:\Users\Student\Downloads\pwdump7>pwdump7
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:NO PASSWORD*********************:31D6CFE0D16AE931B73C59D7E0C089C0:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
¶:503:NO PASSWORD*********************:NO PASSWORD*********************:::
Student:1001:NO PASSWORD*********************:EAB4556003A83E179A149CE6583E097F:::

C:\Users\Student\Downloads\pwdump7>pwdump7 >> windowspasswords.txt
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es


C:\Users\Student\Downloads\pwdump7>dir
 Volume in drive C has no label.
 Volume Serial Number is FC4F-D21B

 Directory of C:\Users\Student\Downloads\pwdump7

03/21/2018  07:53 PM    <DIR>          .
03/21/2018  07:53 PM    <DIR>          ..
03/21/2018  06:35 PM         1,017,344 libeay32.dll
03/21/2018  06:35 PM            77,824 PwDump7.exe
03/21/2018  06:35 PM               522 readme.txt
03/21/2018  07:53 PM               327 windowspasswords.txt
               4 File(s)      1,096,017 bytes
               2 Dir(s)  133,521,641,472 bytes free

C:\Users\Student\Downloads\pwdump7>windowspasswords.txt

C:\Users\Student\Downloads\pwdump7>
```

Step 3:

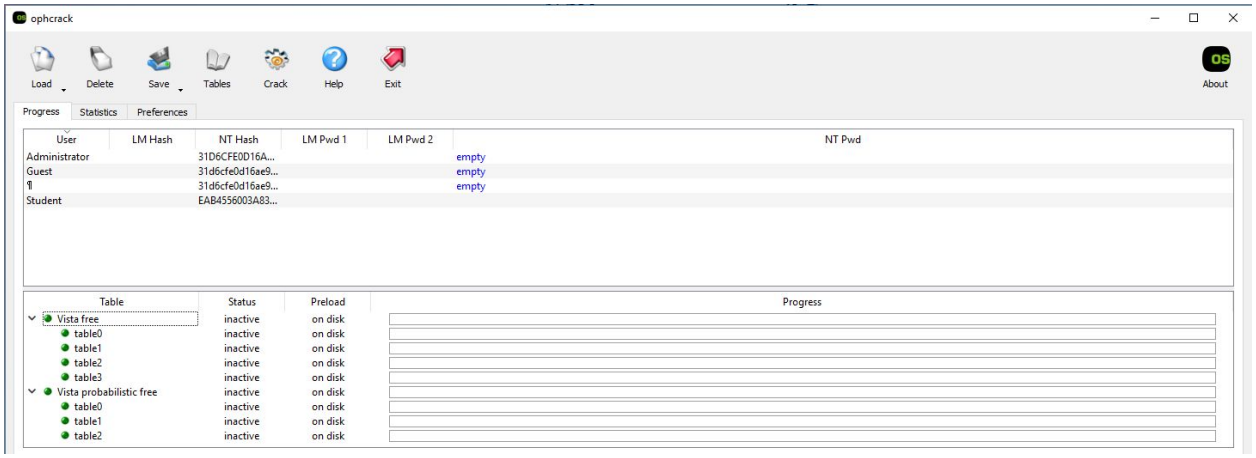
Verify that windows password.txt includes the usernames and password hashes for the local windows machine



```
windowspasswords.txt - Notepad
File  Edit  Format  View  Help
Administrator:500:NO PASSWORD*********************:31D6CFE0D16AE931B73C59D7E0C089C0:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
¶:503:NO PASSWORD*********************:NO PASSWORD*********************:::
Student:1001:NO PASSWORD*********************:EAB4556003A83E179A149CE6583E097F:::
```

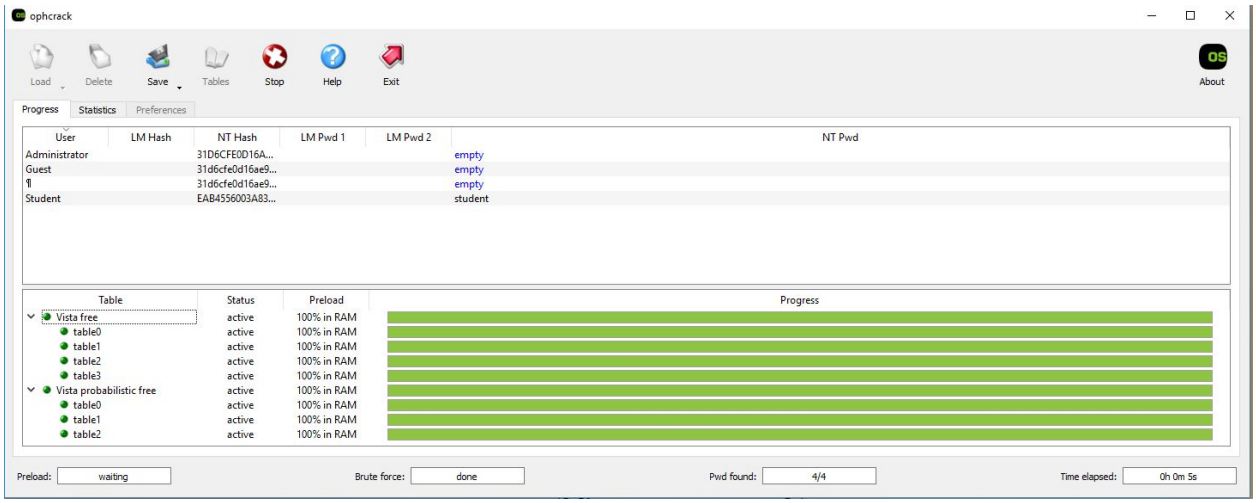
Step 4:

open & select pwdump [windowspasswords.txt] file in ophcrack



Step 5:

crack windowspasswords.txt file



SECTION 2:

EASY

Kali Users info for *Bruteforce* & *Dictionary* attacks on **[Easy]** pw level

User: lexie
Hash function SHA512 6J4qOkjJv$/Pvkx4VNJFOP0Nfm1YB1PyMV88xt7NWBLbaOfLXp9luM.PT24/R37KKrtMQugXxd.nOrXlc12loae6VHrrBea
password: apple

Windows User info for Rainbow table attacks on **[Easy]** pw level

User: student:1001:NO PASSWORD
Hash function MD5 EAB4556003A83E179A149CE6583E097F
Password: student

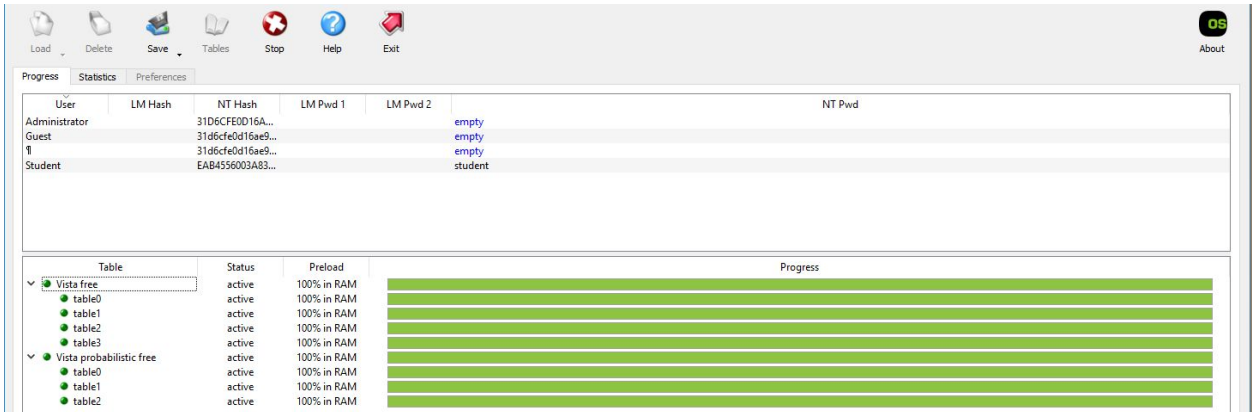
BF → Approx time to to complete: 6.06 mins

```
crunch: 100% completed generating output
root@kali:~# john --wordlist=crunch_bruteforce.txt lexie_info.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Remaining 1 password hash
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:11 0.07% (ETA: 00:51:11) 0g/s 758.4p/s 758.4c/s 758.4C/s aamzg..aanbr
0g 0:00:00:29 0.19% (ETA: 01:03:09) 0g/s 749.8p/s 749.8c/s 749.8C/s abgts..abgwd
0g 0:00:01:32 0.59% (ETA: 01:03:49) 0g/s 758.9p/s 758.9c/s 758.9C/s adzwi..adzyt
0g 0:00:05:03 1.91% (ETA: 01:09:33) 0g/s 745.6p/s 745.6c/s 745.6C/s amwwm..amwyx
apple            (lexie)
1g 0:00:06:06 DONE (2018-03-28 20:50) 0.002728g/s 747.9p/s 747.9c/s 747.9C/s appki..appmt
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~#
```

DT → Approx time to to complete: 1 second

```
root@kali:~#
root@kali:~# unshadow /etc/passwd /etc/shadow > lexie_info.txt
root@kali:~# john --wordlist=rockyou.txt lexie_info.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Remaining 1 password hash
Press 'q' or Ctrl-C to abort, almost any other key for status
apple            (lexie)
1g 0:00:00:01 DONE (2018-03-28 21:01) 0.9345g/s 717.7p/s 717.7c/s 717.7C/s bambam..james1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~#
```

RT → Approx time to complete: 0.5 seconds



Medium

Kali User info for *Bruteforce* & *Dictionary* attacks on **[Medium]** pw level

User: dave
Hash function SHA512 6CTLXlYc2$agqcml50dhkQEkXKpHHchnpjC/yTo7c/5R6aEKDpwQwPR9CYb0wwK0I1YBMmw9xTvck//6epBQiuh4Pn199G30
password: Dave0513

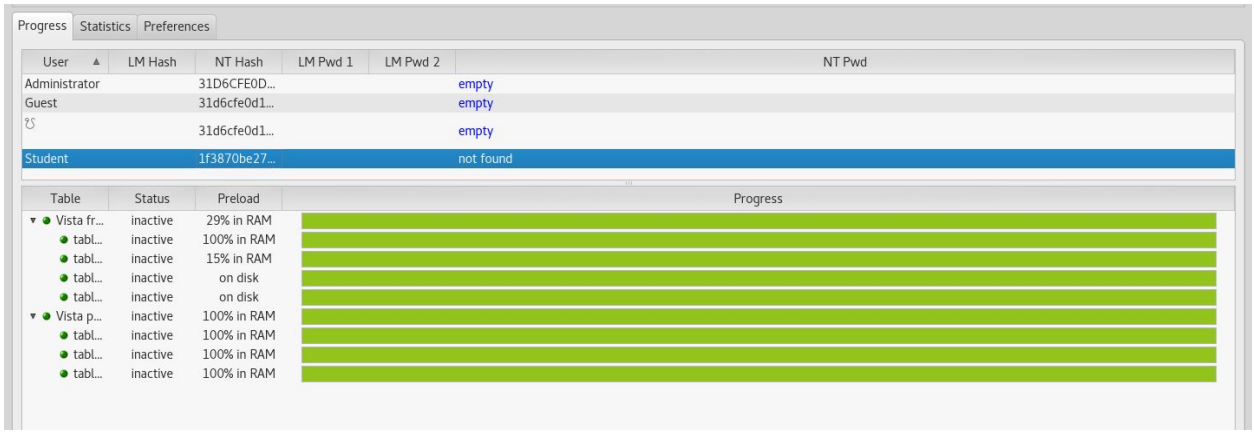
BF → Approx time to to run: 2 hrs 55 mins [**Did not crack**]

```
crunch: 100% completed generating output
root@kali:~# john --wordlist=crunch_bruteforce.txt dave_info.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Remaining 1 password hash
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:02:12 1.26% (ETA: 22:56:28) 0g/s 697.2p/s 697.2c/s 697.2C/s aIia0513..aIjl0513
0g 0:00:04:17 2.48% (ETA: 22:55:08) 0g/s 703.8p/s 703.8c/s 703.8C/s bpbC0513..bpcN0513
0g 0:00:10:59 6.28% (ETA: 22:57:10) 0g/s 696.6p/s 696.6c/s 696.6C/s dnVK0513..dnWV0513
0g 0:00:13:58 8.13% (ETA: 22:54:16) 0g/s 708.5p/s 708.5c/s 708.5C/s elLm0513..elMx0513
0g 0:00:32:07 18.92% (ETA: 22:52:11) 0g/s 717.5p/s 717.5c/s 717.5C/s jRyW0513..jRAh0513
0g 0:00:41:49 24.32% (ETA: 22:54:23) 0g/s 708.4p/s 708.4c/s 708.4C/s mHzC0513..mHAN0513
0g 0:01:11:15 40.52% (ETA: 22:58:15) 0g/s 692.8p/s 692.8c/s 692.8C/s vdFe0513..vdGp0513
0g 0:01:27:25 49.02% (ETA: 23:00:43) 0g/s 683.3p/s 683.3c/s 683.3C/s zzEy0513..zzFJ0513
0g 0:01:31:32 51.23% (ETA: 01:08:30) 0g/s 681.9p/s 681.9c/s 681.9C/s AHnm0513..AHox0513
0g 0:02:06:34 69.86% (ETA: 01:11:00) 0g/s 672.5p/s 672.5c/s 672.5C/s KraW0513..Krch0513
0g 0:02:55:11 DONE (2018-03-30 01:05) 0g/s 695.5p/s 695.5c/s 695.5C/s ZZYO0513..ZZZZ0513
Session completed
root@kali:~#
```

DT → Approx time to to run: 5 hrs 17 mins [**Did not crack**]

```
File  Edit  View  Search  Terminal  Help
root@kali:~# john --wordlist=rockyou.txt christine_info.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Remaining 1 password hash
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:12:17 3.39% (ETA: 20:05:58) 0g/s 764.8p/s 764.8c/s 764.8C/s yuni123..yuli2007
0g 0:00:15:58 4.44% (ETA: 20:03:15) 0g/s 765.4p/s 765.4c/s 765.4C/s 1831990..182419
0g 0:00:42:49 12.42% (ETA: 19:48:19) 0g/s 763.7p/s 763.7c/s 763.7C/s bridge10..brick2386
0g 0:02:37:08 48.83% (ETA: 19:25:31) 0g/s 753.4p/s 753.4c/s 753.4C/s jdlover277..jdlcldvq0514
0g 0:02:55:17 54.82% (ETA: 19:23:26) 0g/s 753.7p/s 753.7c/s 753.7C/s geepops..geenjongens!
0g 0:03:12:45 60.50% (ETA: 19:22:17) 0g/s 754.2p/s 754.2c/s 754.2C/s deann413..deanmassie
0g 0:03:48:52 72.31% (ETA: 19:20:14) 0g/s 754.1p/s 754.1c/s 754.1C/s adolphus1976..adolfotam
0g 0:05:17:04 DONE (2018-03-24 19:20) 0g/s 753.9p/s 753.9c/s 753.9C/s    0109381602..[illegible]¡Vamos![illegible]
Session completed
root@kali:~#
```

RT → Approx time to run: 30.5 mins [**did not crack**]



NOTE: Attempted: ophcrack(as seen above) | WinRtgen | rainbowcrack[linux](Rtgen, rtcrack)
Consistent error: can not find rainbow table, after many trials and errors, this message remained

Rainbow crack → RTGen (successfully created rainbow table)

```
root@kali:~# pwd
/root
root@kali:~# cd /usr /share/rainbowcrack
bash: cd: too many arguments
root@kali:~# /usr /share/rainbowcrack
bash: /usr: Is a directory
root@kali:~# usr /share/rainbowcrack
bash: usr: command not found
root@kali:~# cd /usr/share/rainbowcrack/
root@kali:/usr/share/rainbowcrack# ls
alglib0.so   md5_loweralpha#5-7_0_5000x6553600_0.rt   rcrack      rt2rtc  rtgen    rtsort
charset.txt  ntlm_loweralpha#5-7_0_5000x6553600_0.rt  readme.txt  rtc2rt  rtmerge
root@kali:/usr/share/rainbowcrack# rtgen ntlm loweralpha 5 7 0 5000 6553600 0
rainbow table ntlm_loweralpha#5-7_0_5000x6553600_0.rt parameters
hash algorithm:         ntlm
hash length:            16
charset name:           loweralpha
charset data:           abcdefghijklmnopqrstuvwxyz
charset data in hex:    61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a
charset length:         26
plaintext length range: 5 - 7
reduce offset:          0x00000000
plaintext total:        8352607328

precomputation of this rainbow table is finished
root@kali:/usr/share/rainbowcrack#
```

Rainbowcrack → rcrack (consistent rainbow table not found error)

```
root@kali:/usr/share/rainbowcrack# ls
alglib0.so  charset.txt  md5_loweralpha#1-5_0_2500x3556432_0.rt  ntlm_loweralpha#5-7_0_5000x6553600_0.rt  rcrack  readme.txt  rt2rtc  rtgen  rtmerge  rtsort
root@kali:/usr/share/rainbowcrack# rcrack md5_loweralpha-numeric#1-5_0_3800x33554432_0.rt -h cd73502828457d15655bbd7a63fb0bc8

no rainbow table found

result
---------------------------------------------------------------
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack#
root@kali:/usr/share/rainbowcrack# rcrack md5_loweralpha-numeric#1-5_0_3800x33554432_0.rt -h cd73502828457d15655bbd7a63fb0bc8

no rainbow table found

result
---------------------------------------------------------------
root@kali:/usr/share/rainbowcrack#
```