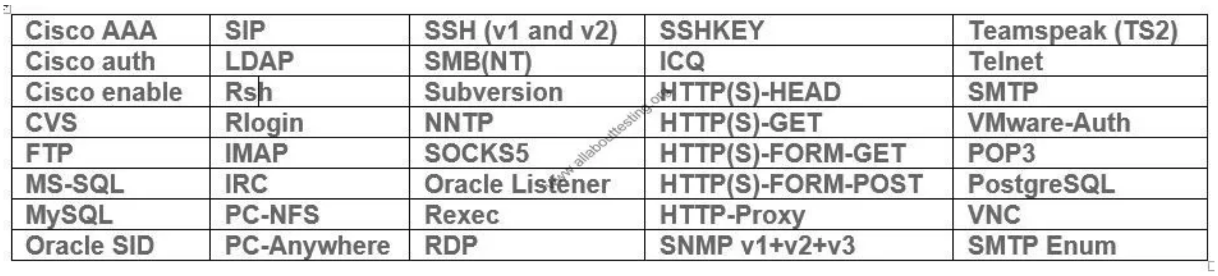
WHAT İS HYDRA?

Weak passwords are still big problem in security. it is becoming easier to cracking algorithms with brute-forcing which is a major kind of attack. Hydra is a parallelized login cracker which supports many of protocols to attack remotely. With Hydra, it is getting faster and flexible and there is a chance to add new modules easily. Hydra uses wordlists to attack systems. Wordlists, contains most used passwords and we can customize our own wordlists to attack specific persons. In this case, this situation shows the importance of using a strong password.

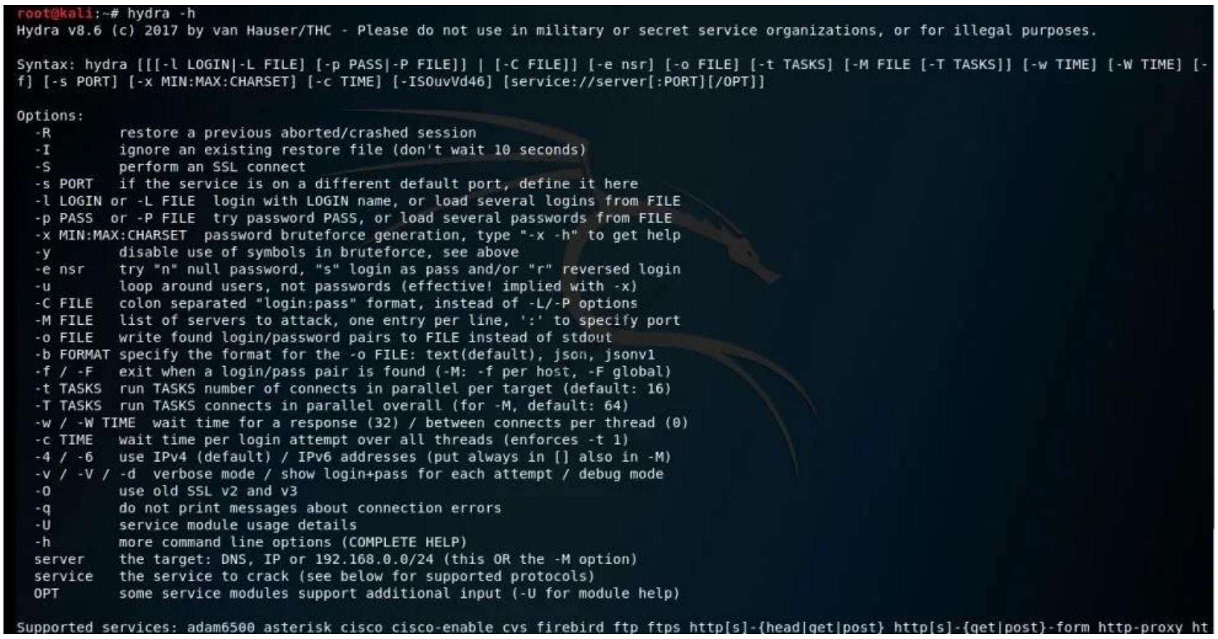


Cisco AAA	SIP	SSH (v1 and v2)	SSHKEY	Teamspeak (TS2)
Cisco auth	LDAP	SMB(NT)	ICQ	Telnet
Cisco enable	Rsh	Subversion	HTTP(S)-HEAD	SMTP
CVS	Rlogin	NNTP	HTTP(S)-GET	VMware-Auth
FTP	IMAP	SOCKS5	HTTP(S)-FORM-GET	POP3
MS-SQL	IRC	Oracle Listener	HTTP(S)-FORM-POST	PostgreSQL
MySQL	PC-NFS	Rexec	HTTP-Proxy	VNC
Oracle SID	PC-Anywhere	RDP	SNMP v1+v2+v3	SMTP Enum

the list of all protocols supported by hydra.

Hydra is a pre-installed tool in Kali Linux. You can check the usage of Hydra with using this command. This page gives us the knowledge about parameters which we can use with Hydra tool:

#hydra -h



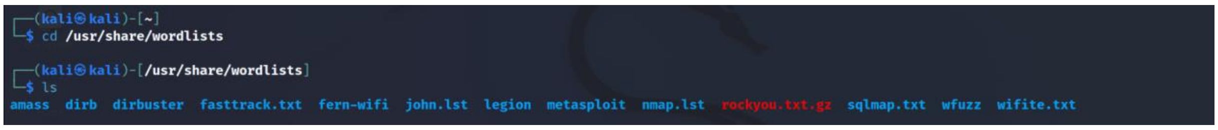
```
root@kali:~# hydra -h
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-
f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [service://server[:PORT][/OPT]]

Options:
  -R        restore a previous aborted/crashed session
  -I        ignore an existing restore file (don't wait 10 seconds)
  -S        perform an SSL connect
  -s PORT   if the service is on a different default port, define it here
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -x MIN:MAX:CHARSET  password bruteforce generation, type "-x -h" to get help
  -y        disable use of symbols in bruteforce, see above
  -e nsr    try "n" null password, "s" login as pass and/or "r" reversed login
  -u        loop around users, not passwords (effective! implied with -x)
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -o FILE   write found login/password pairs to FILE instead of stdout
  -b FORMAT specify the format for the -o FILE: text(default), json, jsonv1
  -f / -F   exit when a login/pass pair is found (-M: -f per host, -F global)
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -T TASKS  run TASKS connects in parallel overall (for -M, default: 64)
  -w / -W TIME  wait time for a response (32) / between connects per thread (0)
  -c TIME   wait time per login attempt over all threads (enforces -t 1)
  -4 / -6   use IPv4 (default) / IPv6 addresses (put always in [] also in -M)
  -v / -V / -d  verbose mode / show login+pass for each attempt / debug mode
  -O        use old SSL v2 and v3
  -q        do not print messages about connection errors
  -U        service module usage details
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-{head|get|post} http[s]-{get|post}-form http-proxy ht
```

To brute-force ssh username and password

#hydra <Target_IP> ssh -l <username> -P <password_file> -s 22 -vV

We have username and password list to attack the target by using Hydra. You can access the wordlist files with this directory (for instance rockyou.txt file iz zipped. You have to unzip to use.)

cd /usr/share/wordlists



```
┌──(kali㉿kali)-[~]
└─$ cd /usr/share/wordlists

┌──(kali㉿kali)-[/usr/share/wordlists]
└─$ ls
amass  dirb  dirbuster  fasttrack.txt  fern-wifi  john.lst  legion  metasploit  nmap.lst  rockyou.txt.gz  sqlmap.txt  wfuzz  wifite.txt
```

To brute-force FTP username and password:

#hydra -L <username_file> -P <password_file> ftp://<Target_IP>

-l : input login(not the list of usernames)

-L: username list

-p: if you will try single password

-P: list of passwords to do brute-forcing

-v : gives us details

-s : we can add the spesific ports that we want to attack with that parameter

-S: If it is possible, Hydra will try to connect with SSL with that parameter

-M: We will specified the location of the list of ports of the servers with this parameter

There is no difference between ssh and ftp connections usage . Parameters are same.

To brute-force telnet username and password:

#hydra -l <username> -p <password> telnet://<Target_IP>

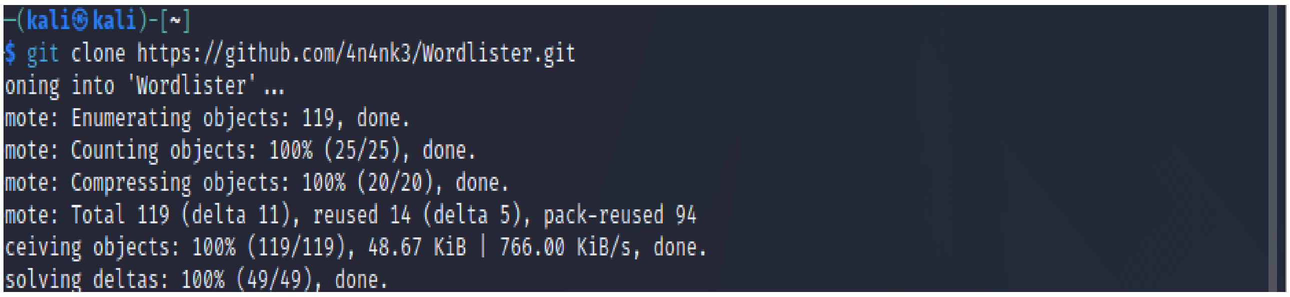
xHydra

Hydra also has interface page name that xHydra. You can open the xHydra page in terminal with “xhydra” command. In this interface you can pick the protocols and ports and other options from there. All the options are same with Hydra.

WORDLISTER TOOL

Firstly, we should install the Wordlister.

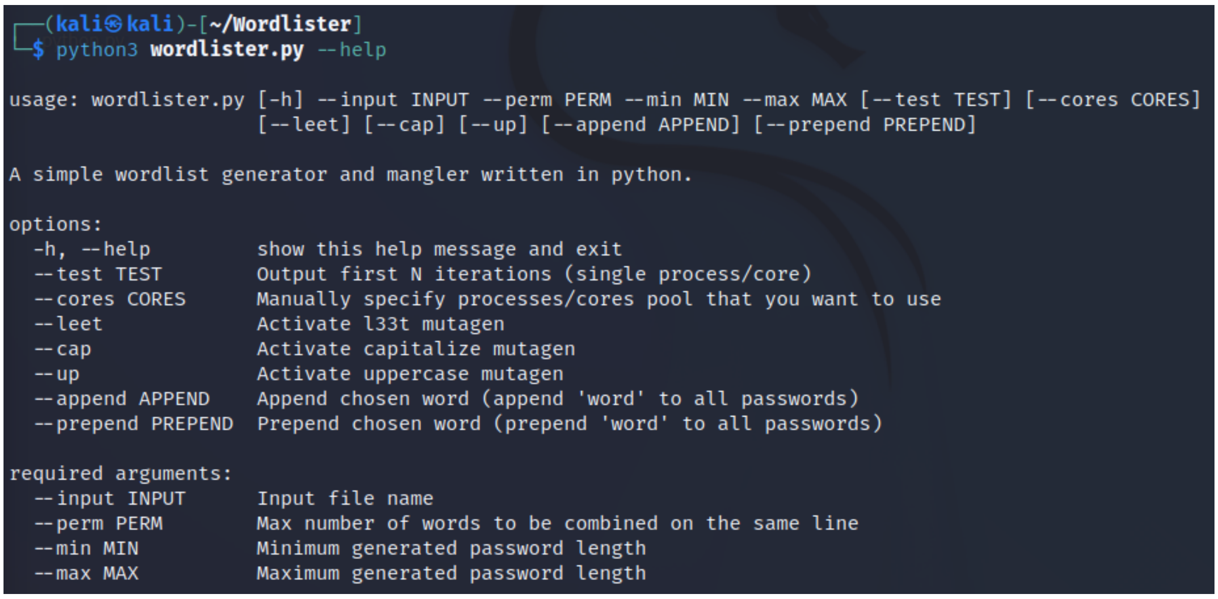
git clone https://github.com/4n4nk3/Wordlister.git command will help us.



```
-(kali㉿kali)-[~]
$ git clone https://github.com/4n4nk3/Wordlister.git
oning into 'Wordlister'...
mote: Enumerating objects: 119, done.
mote: Counting objects: 100% (25/25), done.
mote: Compressing objects: 100% (20/20), done.
mote: Total 119 (delta 11), reused 14 (delta 5), pack-reused 94
ceiving objects: 100% (119/119), 48.67 KiB | 766.00 KiB/s, done.
solving deltas: 100% (49/49), done.
```

python3 wordlister.py –help

This command will give us options about how we can use the tool.



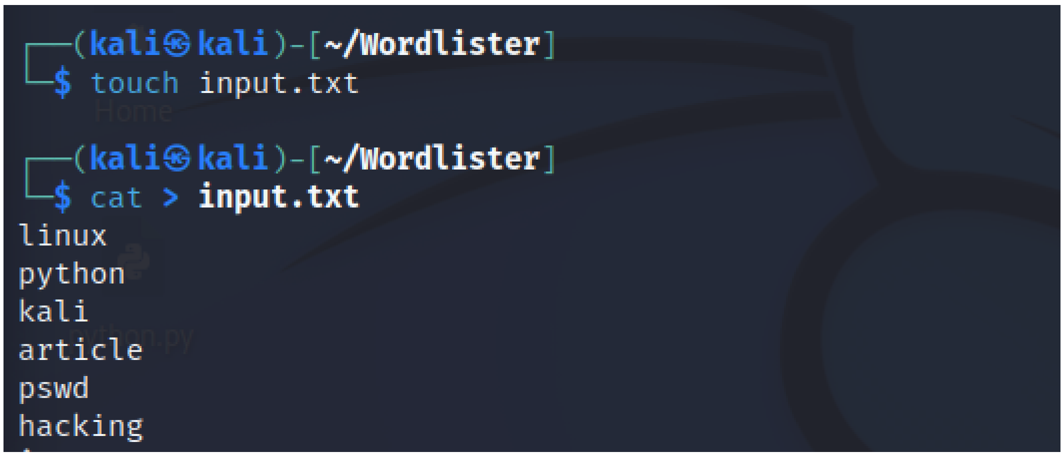
```
┌──(kali㉿kali)-[~/Wordlister]
└─$ python3 wordlister.py --help

usage: wordlister.py [-h] --input INPUT --perm PERM --min MIN --max MAX [--test TEST] [--cores CORES]
                     [--leet] [--cap] [--up] [--append APPEND] [--prepend PREPEND]

A simple wordlist generator and mangler written in python.

options:
  -h, --help         show this help message and exit
  --test TEST        Output first N iterations (single process/core)
  --cores CORES      Manually specify processes/cores pool that you want to use
  --leet             Activate l33t mutagen
  --cap              Activate capitalize mutagen
  --up               Activate uppercase mutagen
  --append APPEND    Append chosen word (append 'word' to all passwords)
  --prepend PREPEND  Prepend chosen word (prepend 'word' to all passwords)

required arguments:
  --input INPUT      Input file name
  --perm PERM        Max number of words to be combined on the same line
  --min MIN          Minimum generated password length
  --max MAX          Maximum generated password length
```

We must have input word list to give Python to make wordlister with all combinations.

Then, we will give the details. For instance, we can choose the character size. Like you have to choose minimum 6 character for passwords.

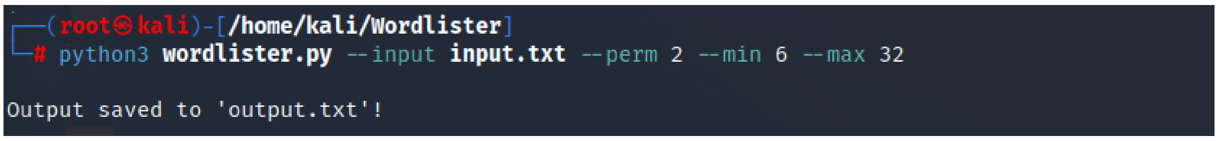
python3 wordlister.py –input list.txt –perm 2 –min 6 –max 32

this command perm command tells us max 2 words can be conbined on the same possible password.



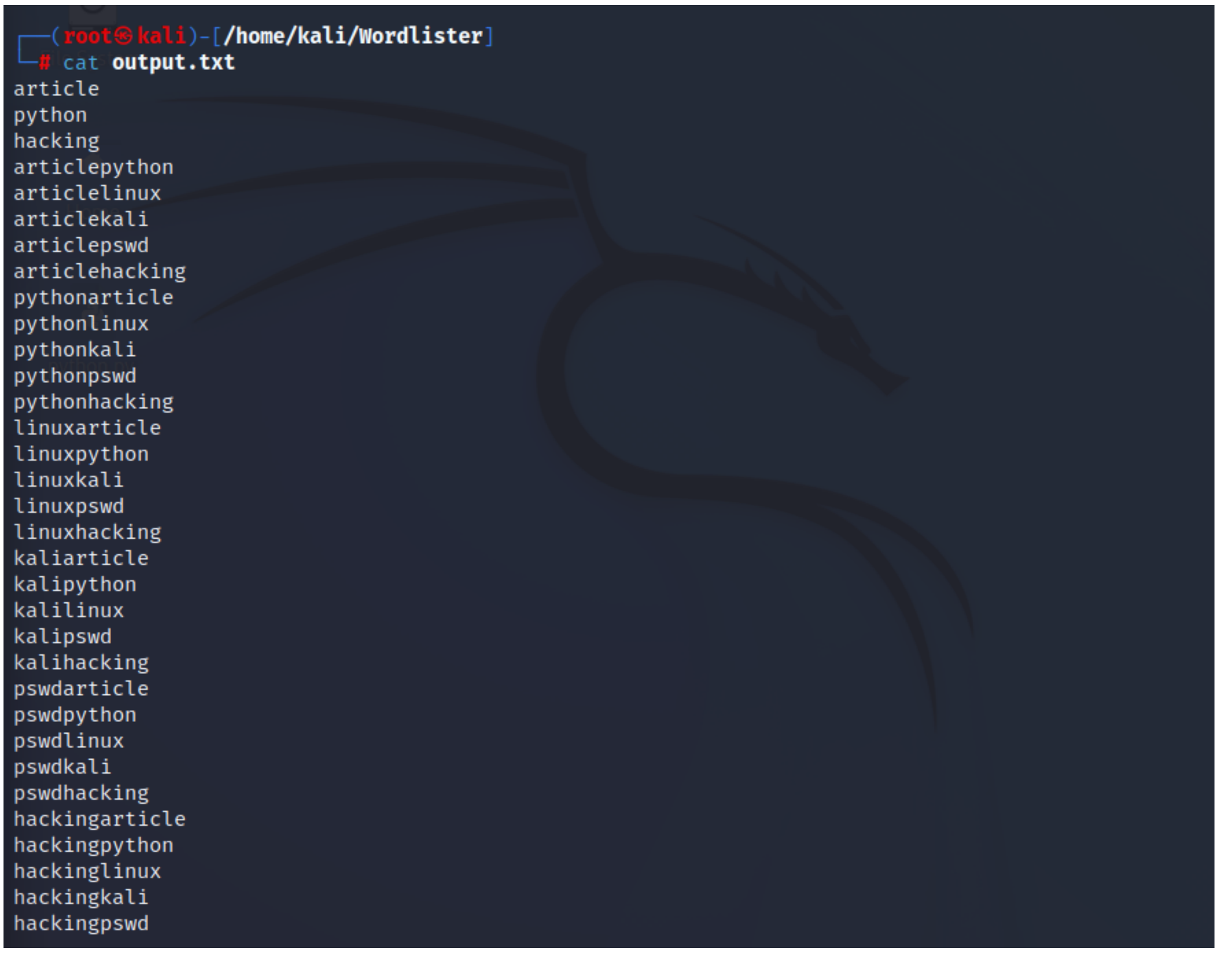
```
┌──(kali㉿kali)-[~/Wordlister]
└─$ touch input.txt

┌──(kali㉿kali)-[~/Wordlister]
└─$ cat > input.txt
linux
python
kali
article
pswd
hacking
```

We put the words in a txt file. Then run the command.



```
┌──(root㉿kali)-[/home/kali/Wordlister]
└─# python3 wordlister.py --input input.txt --perm 2 --min 6 --max 32

Output saved to 'output.txt'!
```

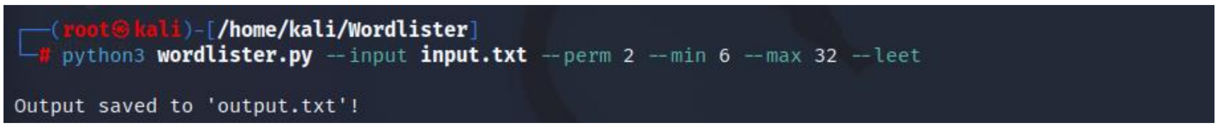
We have the txt file that contains combinations.



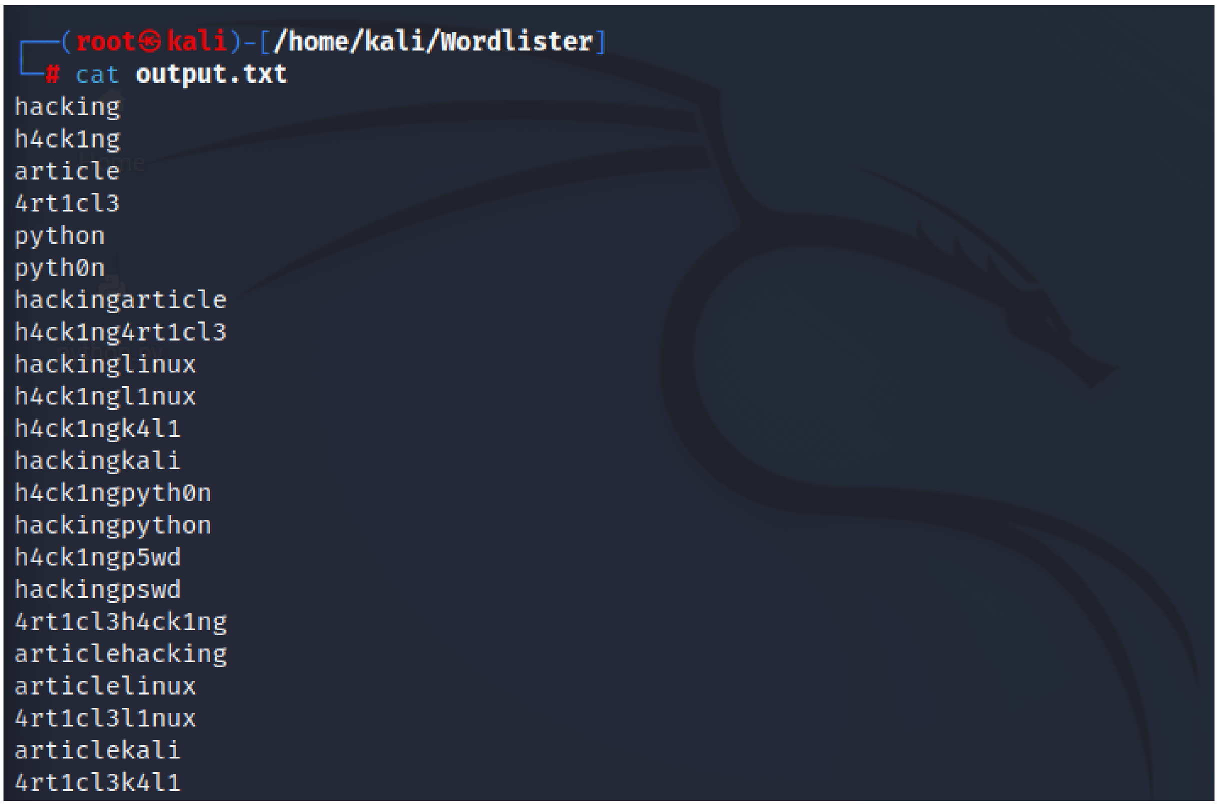
```
┌──(root㉿kali)-[/home/kali/Wordlister]
└─# cat output.txt
article
python
hacking
articlepython
articlelinux
articlekali
articlepswd
articlehacking
pythonarticle
pythonlinux
pythonkali
pythonpswd
pythonhacking
linuxarticle
linuxpython
linuxkali
linuxpswd
linuxhacking
kaliarticle
kalipython
kalilinux
kalipswd
kalihacking
pswdarticle
pswdpython
pswdlinux
pswdkali
pswdhacking
hackingarticle
hackingpython
hackinglinux
hackingkali
hackingpswd
```

Using Leet mod

python3 wordlister.py --input input.txt --perm 2 --min 6 --max 32 --leet



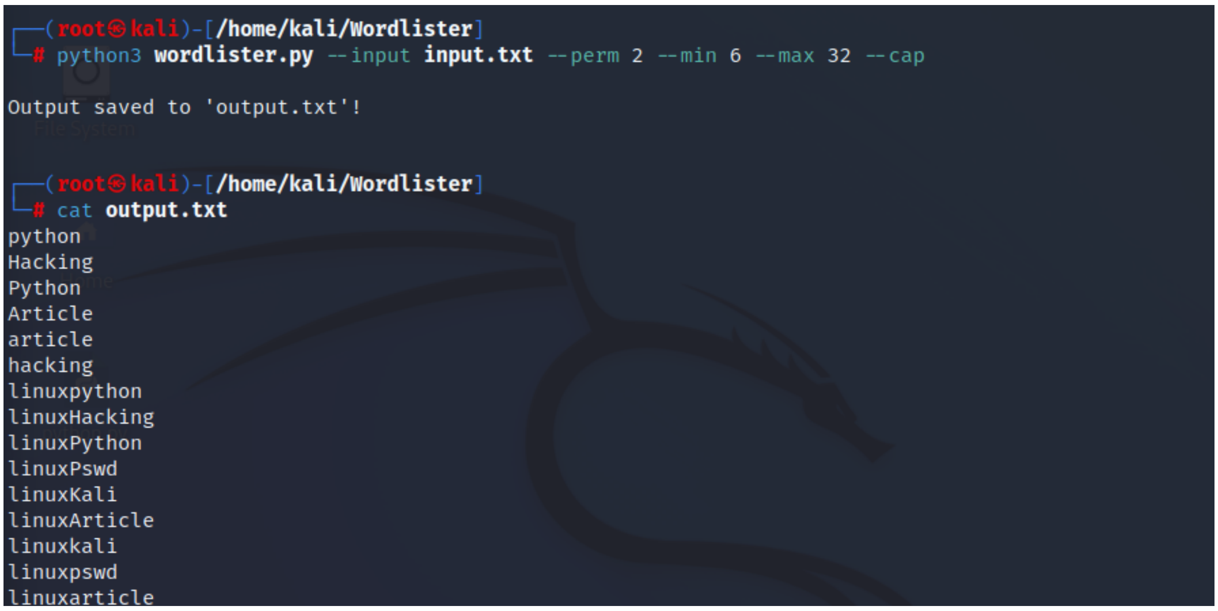
```
┌──(root㉿kali)-[/home/kali/Wordlister]
└─# python3 wordlister.py --input input.txt --perm 2 --min 6 --max 32 --leet

Output saved to 'output.txt'!
```



```
┌──(root㉿kali)-[/home/kali/Wordlister]
└─# cat output.txt
hacking
h4ck1ng
article
4rt1cl3
python
pyth0n
hackingarticle
h4ck1ng4rt1cl3
hackinglinux
h4ck1ngl1nux
h4ck1ngk4l1
hackingkali
h4ck1ngpyth0n
hackingpython
h4ck1ngp5wd
hackingpswd
4rt1cl3h4ck1ng
articlehacking
articlelinux
4rt1cl3l1nux
articlekali
4rt1cl3k4l1
```

Using cap option to capitalize the first letter:

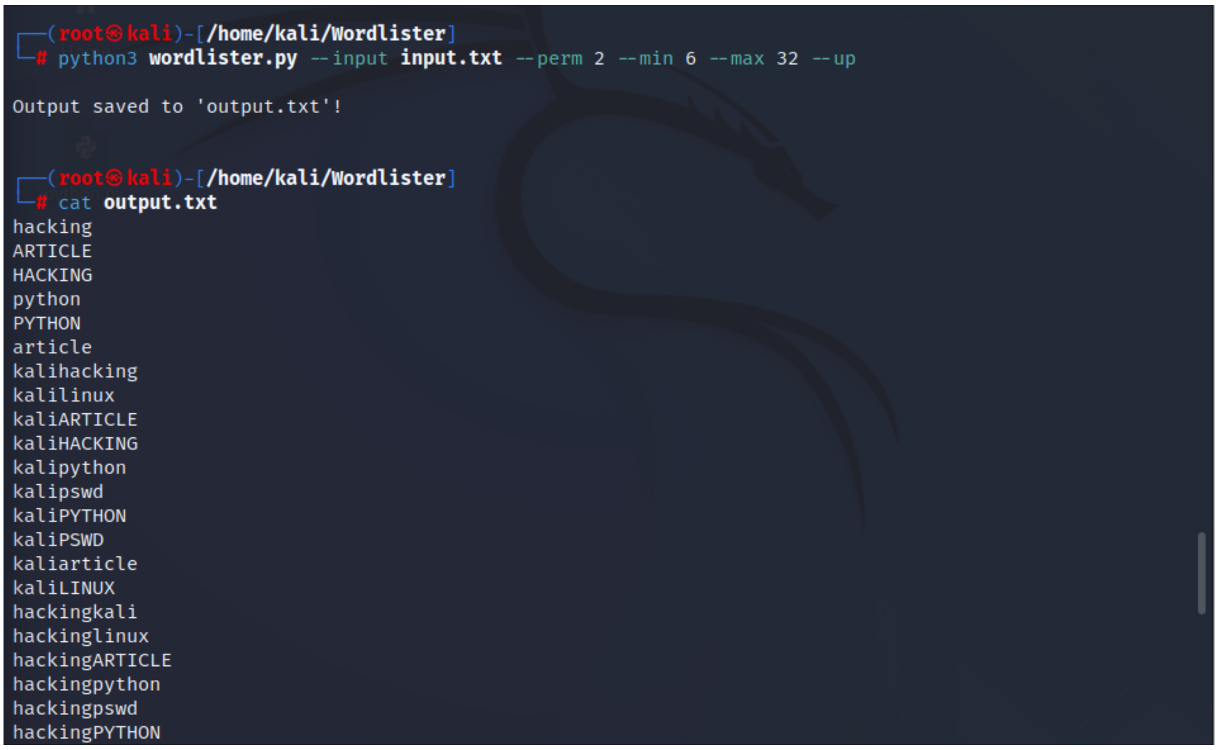
python3 wordlister.py –input list.txt –perm 2 –min 6 –max 32 –cap



```
┌──(root㉿kali)-[/home/kali/Wordlister]
└─# python3 wordlister.py --input input.txt --perm 2 --min 6 --max 32 --cap

Output saved to 'output.txt'!

┌──(root㉿kali)-[/home/kali/Wordlister]
└─# cat output.txt
python
Hacking
Python
Article
article
hacking
linuxpython
linuxHacking
linuxPython
linuxPswd
linuxKali
linuxArticle
linuxkali
linuxpswd
linuxarticle
```

Transform every letter in a word into uppercase:

python3 wordlister.py –input list.txt –perm 2 –min 6 –max 32 –up



```
┌──(root㉿kali)-[/home/kali/Wordlister]
└─# python3 wordlister.py --input input.txt --perm 2 --min 6 --max 32 --up

Output saved to 'output.txt'!


┌──(root㉿kali)-[/home/kali/Wordlister]
└─# cat output.txt
hacking
ARTICLE
HACKING
python
PYTHON
article
kalihacking
kalilinux
kaliARTICLE
kaliHACKING
kalipython
kalipswd
kaliPYTHON
kaliPSWD
kaliarticle
kaliLINUX
hackingkali
hackinglinux
hackingARTICLE
hackingpython
hackingpswd
hackingPYTHON
```

https://www.geeksforgeeks.org/

https://allabouttesting.org/

https://www.kali.org/tools/hydra/

https://anilcelik.medium.com/