Comprehensive Guide to Crunch Tool

posted in HACKING TOOLS , PENETRATION TESTING on MARCH 13, 2018 by RAJ CHANDEL

SHARE

Hello friends!! Today we will demonstrate how a pentester can generate his own wordlist for username either password using the most powerful tool CRUNCH. In kali Linux you can easily get crunch by exploring **Application > Password Attacks > Crunch**

Crunch can generate a wordlist subject to the conditions you specify and its output file can be used in any other another program or file.

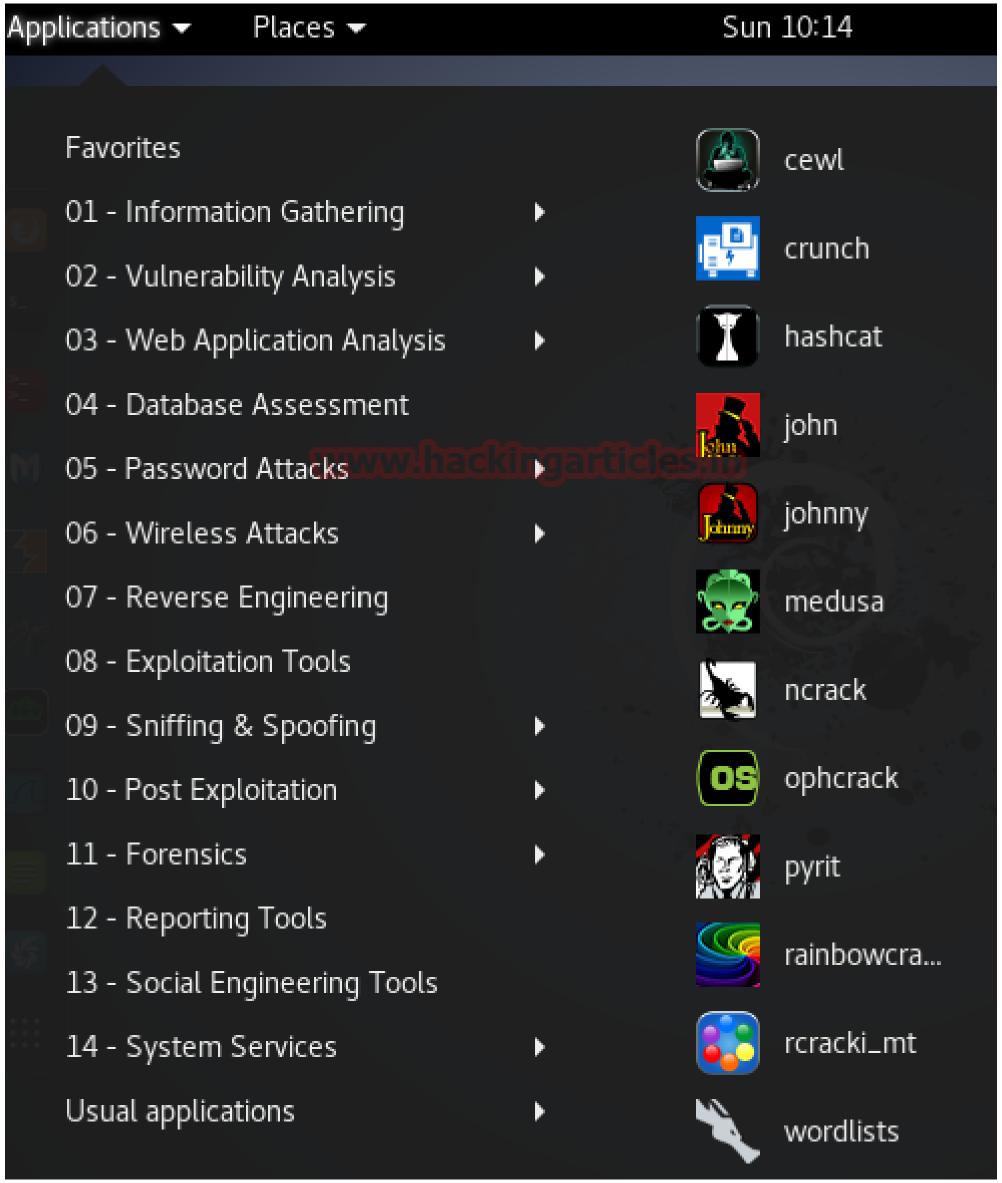
Search

ENTER KEYWORD

Subscribe to Blog via Email

Email Address

SUBSCRIBE



We are using crunch version 3.6 for this tutorial and followed given below parameters for generating wordlist.

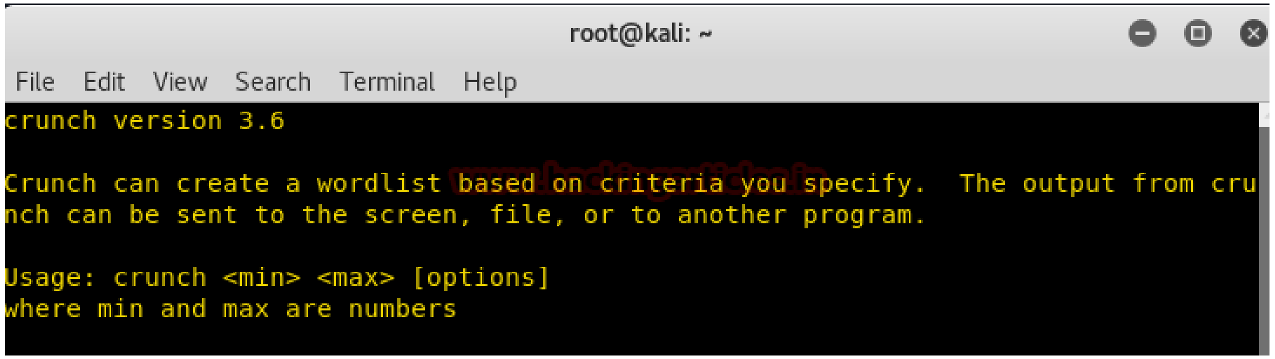
Syntax: <min> <max> [character-string] [options]

Min-len: This parameter specify minimum length string required for crunch to start generating wordlist.

Max-len: This parameter specify maximum length string required for crunch to end.

Charset string: This parameter specify character sets for crunch to use for generating wordlist from that string, if you have not specified any string then crunch will default characters string.

Options: crunch serves you a list of options which increase its functionality for generating wordlist as per your requirement.



```
crunch version 3.6

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

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

Generating wordlist without using character string

Execute given below command which will generate a dictionary that contains minimum 2 character letters and maximum 3 by using default character sets. It will start from aa and end with zzz.

crunch 2 3 -o /root/Desktop/0.txt

Here we had used following parameters for generating a dictionary:

Min_len: 2 for two character letters

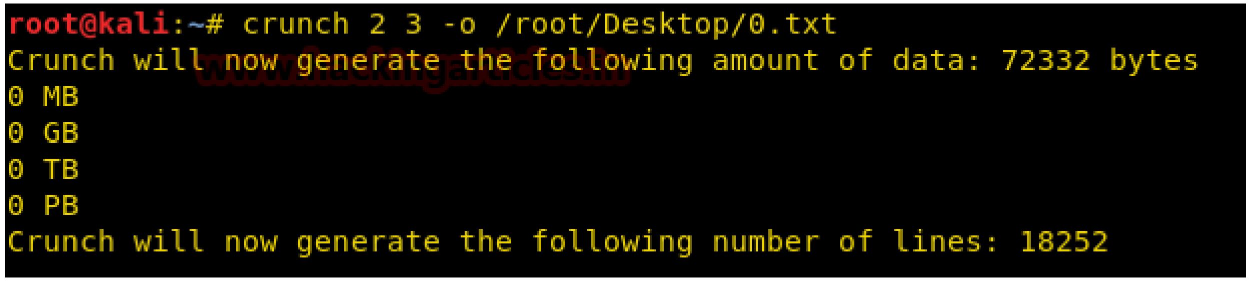
Categories

- BackTrack 5 Tutorials
- Best of Hacking
- Browser Hacking
- Cryptography & Stegnography
- CTF Challenges
- Cyber Forensics
- Database Hacking
- Domain Hacking
- Email Hacking
- Footprinting
- Hacking Tools
- Kali Linux
- Nmap
- Others
- Penetration Testing
- Social Engineering Toolkit
- Trojans & Backdoors
- Website Hacking
- Window Password Hacking
- Windows Hacking Tricks
- Wireless Hacking
- Youtube Hacking

Max_len: 3 for three character letters

-o: This option denotes the path to save the output in a text file.

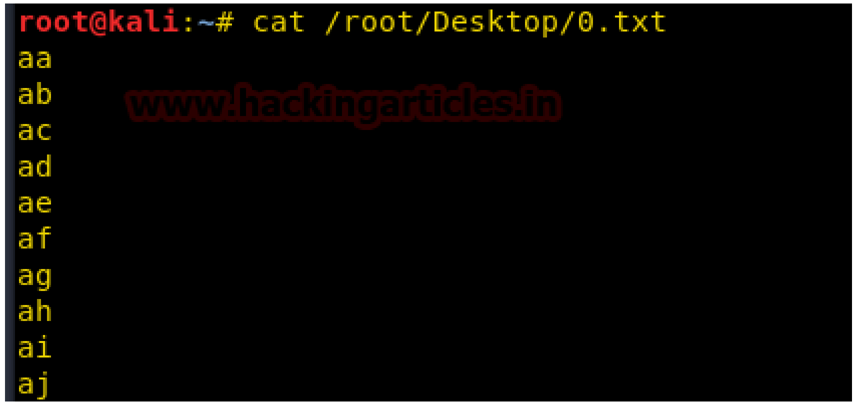
From given below image you can observe that it has generated **18252 number of lines** and **saved** in **0.txt** file.



```
root@kali:~# crunch 2 3 -o /root/Desktop/0.txt
Crunch will now generate the following amount of data: 72332 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 18252
```

Now here we had used **cat command** to read the content from inside **0.txt** file where we can perceive that it has start from aa and end with zzz as shown in given below image.

cat /root/Desktop/0.txt



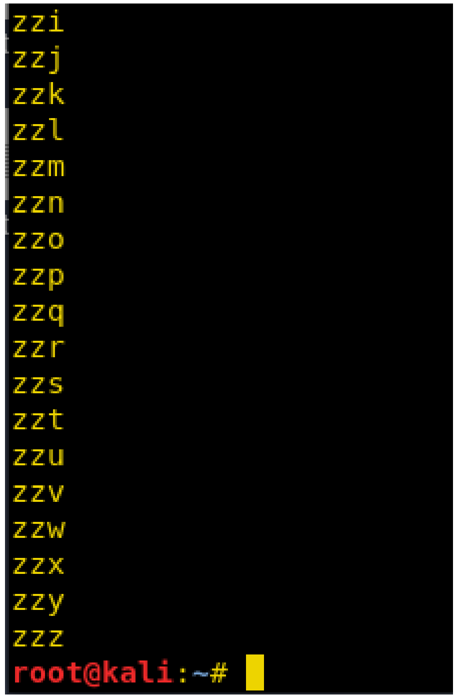
```
root@kali:~# cat /root/Desktop/0.txt
aa
ab
ac
ad
ae
af
ag
ah
ai
aj
```

Articles

Select Month

Facebook Page

Be the first of your friends to like this



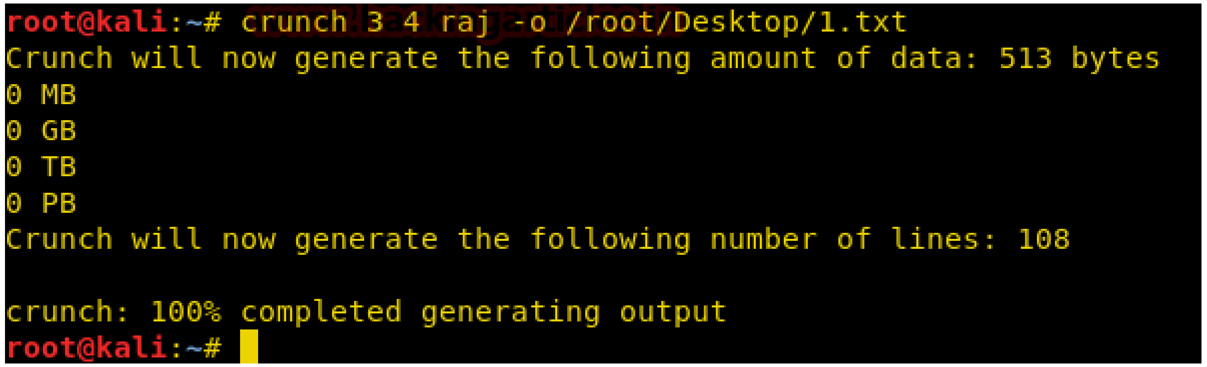
```
zzi
zzj
zzk
zzl
zzm
zzn
zzo
zzp
zzq
zzr
zzs
zzt
zzu
zzv
zzw
zzx
zzy
zzz
root@kali:~#
```

Generating wordlist using character string

Now execute given below command which will generate a dictionary that contains minimum 3 character letters and maximum 4 by using "raj" as specified string. Similarly it will start from rrr and end with jjjj.

crunch 3 4 raj -o /root/Desktop/1.txt

From given below image you can observe that it has generated 108 number of lines and **saved** in **1.txt** file.



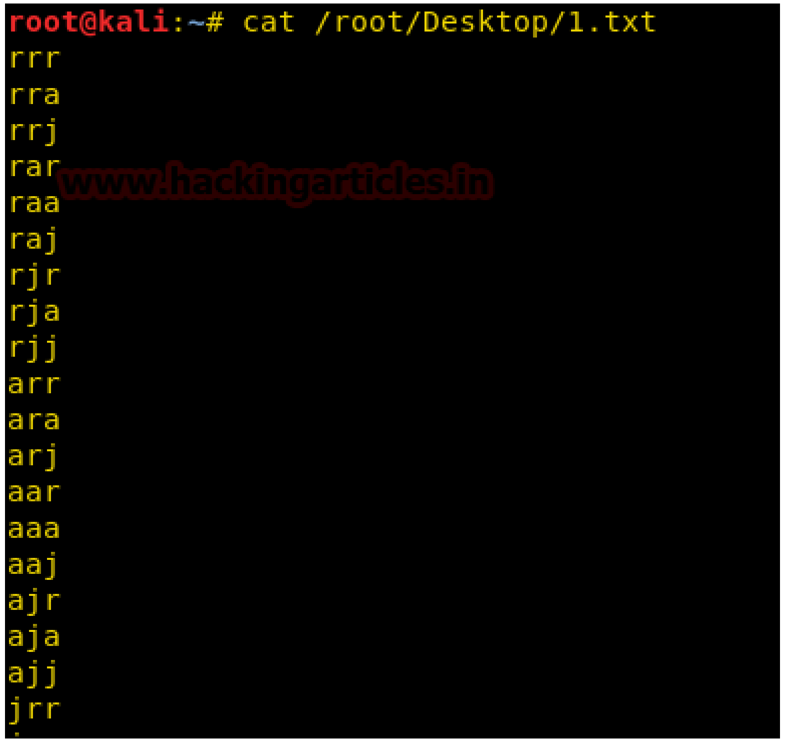
```
root@kali:~# crunch 3 4 raj -o /root/Desktop/1.txt
Crunch will now generate the following amount of data: 513 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 108

crunch: 100% completed generating output
root@kali:~# 
```

Now we had used **cat command** to read the content from inside **1.txt** file where we can perceive that it has start from rrr and end with jjjj.

cat /root/Desktop/1.txt

Similarly we can use string of any number for making a dictionary which contains numeric characters.

For example: some users set their date of birth as password and we would like to generate a dictionary that contains combination of four number such that it represent month and date for instant 25th May as 2505 then you can use "2505" as character string for generating a **numeric wordlist**.



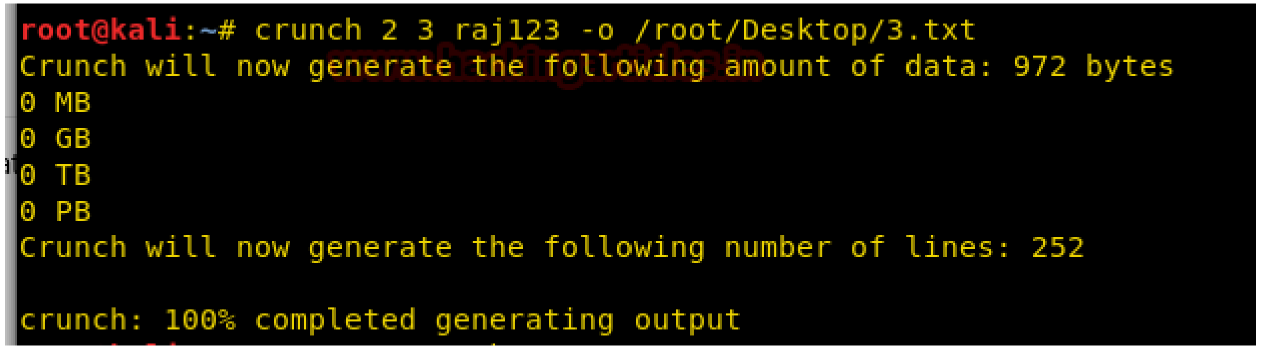
```
root@kali:~# cat /root/Desktop/1.txt
rrr
rra
rrj
rar
raa
raj
rjr
rja
rjj
arr
ara
arj
aar
aaa
aaj
ajr
aja
ajj
jrr
```

Generating alpha-numeric wordlist

You can generate you own alpha-numeric wordlist, execute given below command which will generate a dictionary that contains minimum 2 character letters and maximum 3 by using “raj123” as specified string.

You can set minimum and maximum length for your wordlist as per your requirement.

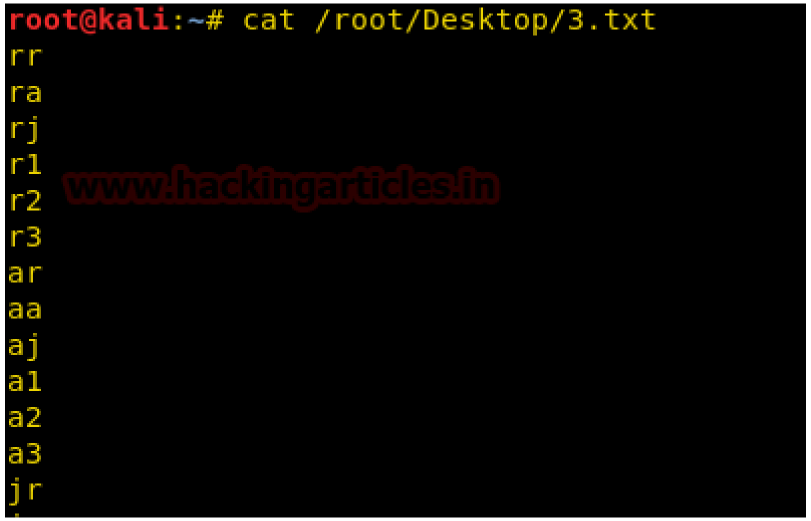
crunch 2 3 raj123 -o /root/Desktop/3.txt



```
root@kali:~# crunch 2 3 raj123 -o /root/Desktop/3.txt
Crunch will now generate the following amount of data: 972 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 252

crunch: 100% completed generating output
```

Again we had used **cat command** to read the content from inside **3.txt** file where we can perceive that it has combination of alpha-numeric character.

cat /root/Desktop/3.txt

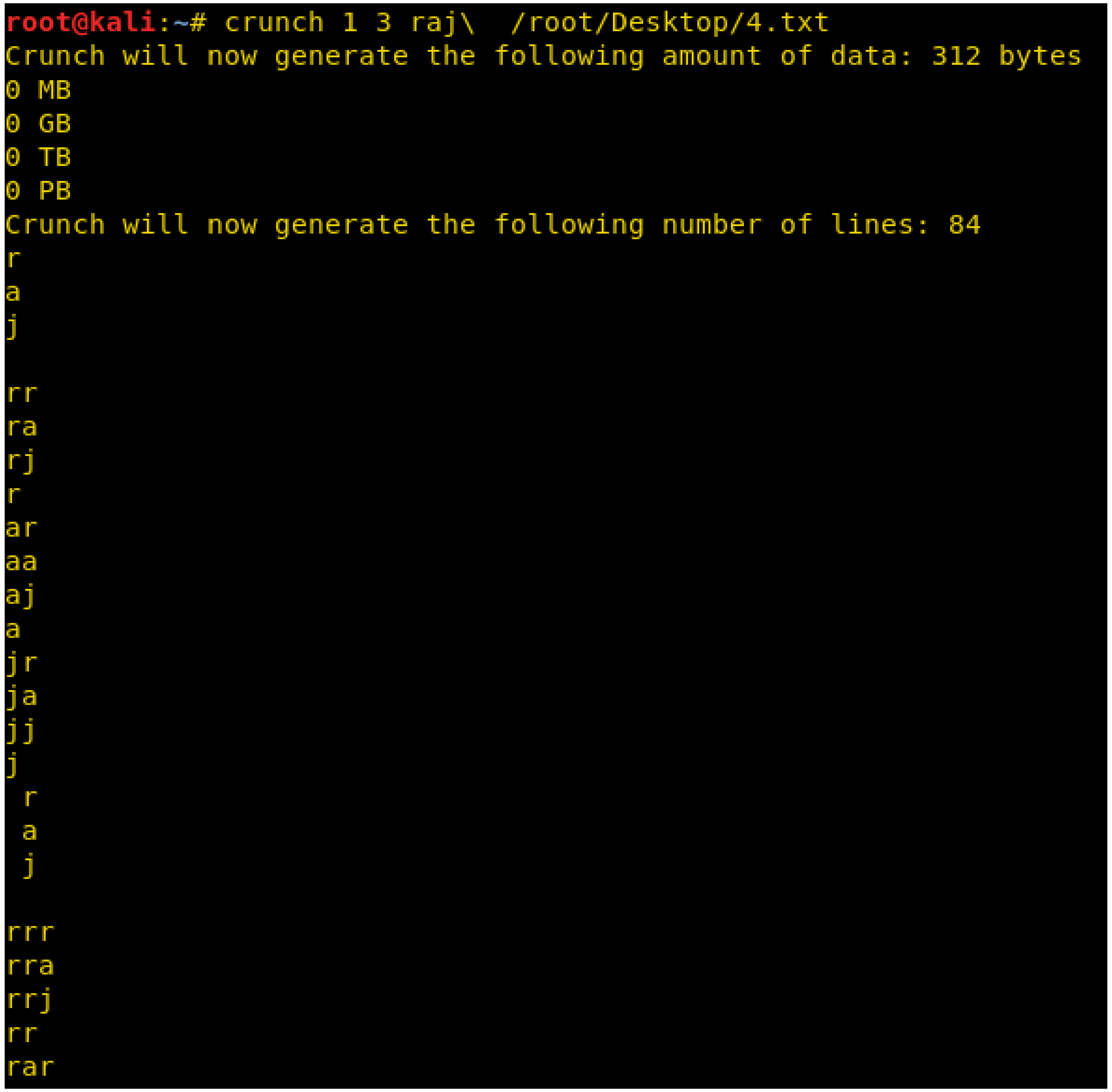


```
root@kali:~# cat /root/Desktop/3.txt
rr
ra
rj
r1
r2
r3
ar
aa
aj
a1
a2
a3
jr
```

Generating wordlist along with space character

Following command will generate wordlist using space character (\) with string “raj”. Instead of using (\) you can also use double quotes around string as “raj ” along with space within double quotes.

crunch 1 3 raj\ -o /root/Desktop/4.txt



```
root@kali:~# crunch 1 3 raj\  /root/Desktop/4.txt
Crunch will now generate the following amount of data: 312 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 84
r
a
j

rr
ra
rj
r
ar
aa
aj
a
jr
ja
jj
j
 r
 a
 j

rrr
rra
rrj
rr
rar
```

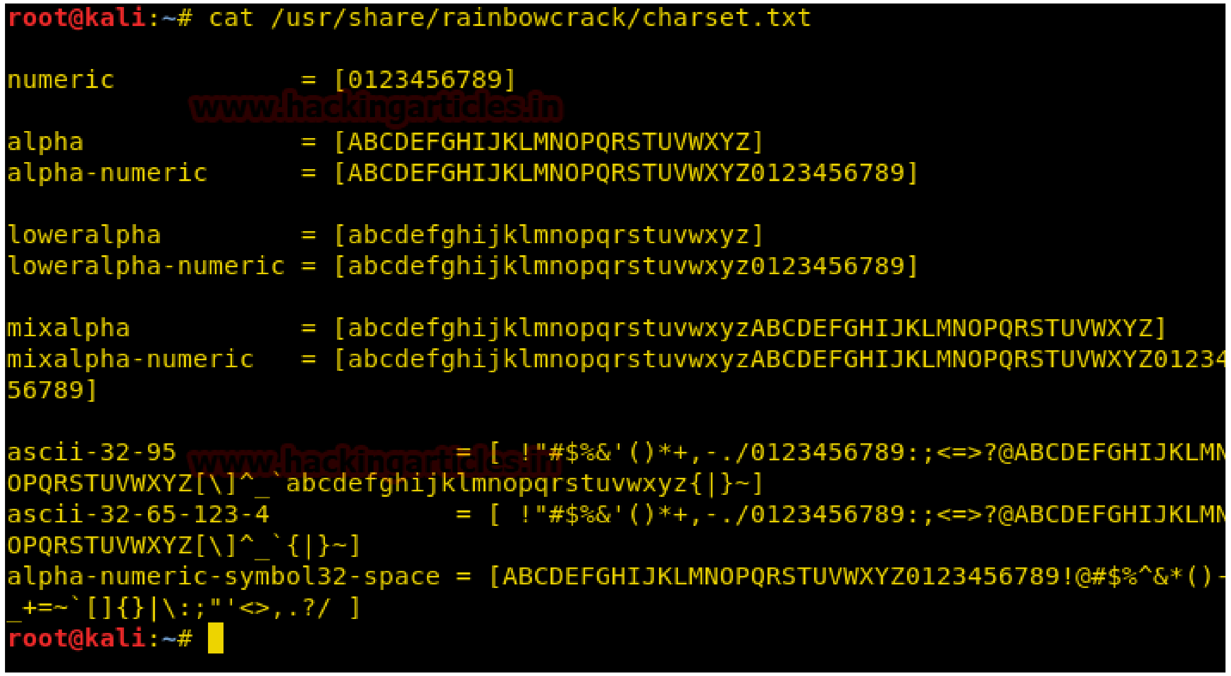
Create wordlist using character set file of RainbowCrack

As we known rainbow crack has a character set file which is used for cracking hashes by using rainbow table, but we'll use this character set file for generating a complex wordlist as per situation demands.

cat /usr/share/rainbowcrack/charset.txt

We had used cat command to express the list of character set that has been stored in **charset.txt** of rainbowcrack. From given below image you can observed that it is showing following list of character set.

- Numeric
- Alpha
- Alpha-numeric
- Loweralpha
- Loweralpha numeric
- Mixalpha
- Mixalpha-numeric
- Ascii -32-95
- Ascii -32-65-123-4
- Alpha-numeric-symbol32-space

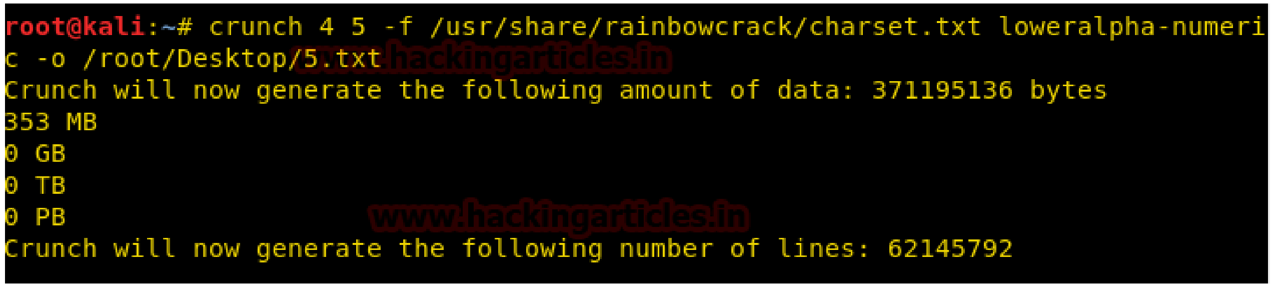


```
root@kali:~# cat /usr/share/rainbowcrack/charset.txt

numeric              = [0123456789]

alpha                = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
alpha-numeric        = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]

loweralpha           = [abcdefghijklmnopqrstuvwxyz]
loweralpha-numeric = [abcdefghijklmnopqrstuvwxyz0123456789]

mixalpha             = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ]
mixalpha-numeric     = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234
56789]

ascii-32-95                    = [ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN
OPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~]
ascii-32-65-123-4              = [ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMN
OPQRSTUVWXYZ[\]^_`{|}~]
alpha-numeric-symbol32-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-
_+=~`[]{}|\:;"'<>,.?/ ]
root@kali:~#
```

Now you can choose any character set for generating wordlist. Let suppose I want to generate a wordlist which contains lower alphabets letter along with numeric number for 5 letter words so for that I will execute following command.

crunch 4 5 -f /usr/share/rainbowcrack/charset.txt loweralpha-numeric -o /root/Desktop/5.txt

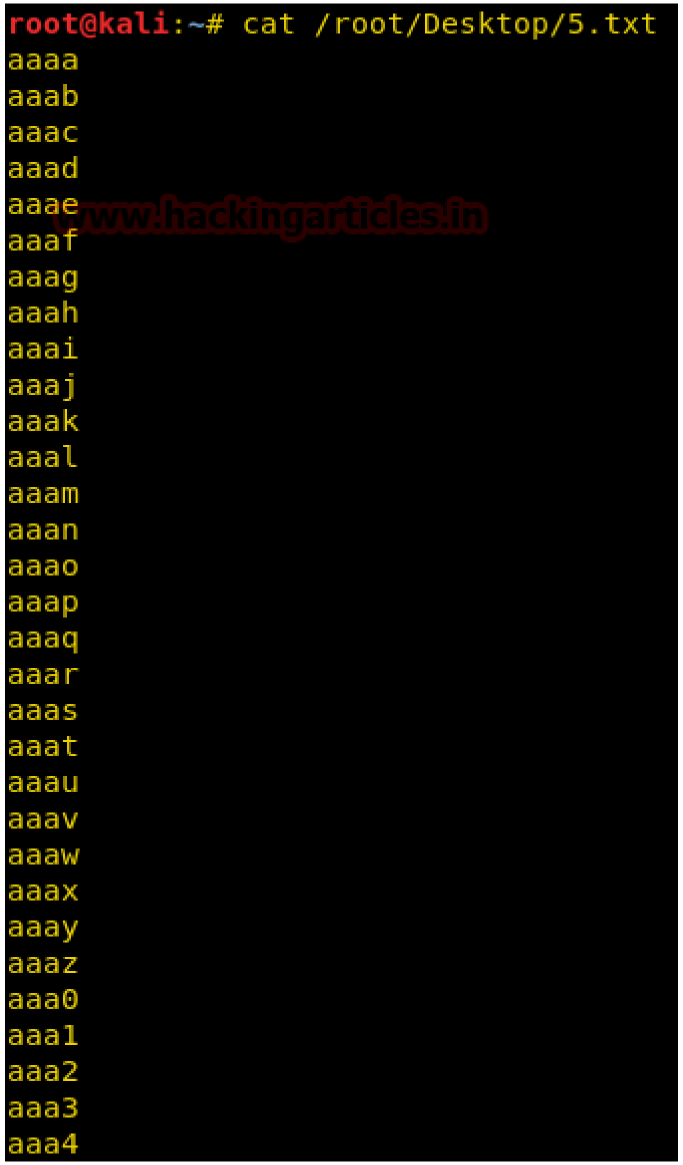
Here **–f** denotes Specifies a character set from the charset.lst



```
root@kali:~# crunch 4 5 -f /usr/share/rainbowcrack/charset.txt loweralpha-numeri
c -o /root/Desktop/5.txt
Crunch will now generate the following amount of data: 371195136 bytes
353 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 62145792
```

Again we had used **cat command** to read the content from inside **5.txt** file where we can perceive that it has combination of alpha-numeric character.

cat /root/Desktop/5.txt



```
root@kali:~# cat /root/Desktop/5.txt
aaaa
aaab
aaac
aaad
aaae
aaaf
aaag
aaah
aaai
aaaj
aaak
aaal
aaam
aaan
aaao
aaap
aaaq
aaar
aaas
aaat
aaau
aaav
aaaw
aaax
aaay
aaaz
aaa0
aaa1
aaa2
aaa3
aaa4
```

Generate wordlist with specific Pattern

Crunch provides **–t option** to generate a wordlist using a specific pattern as per your requirement.

Using option –t you can generate 4 type patters as specified below:

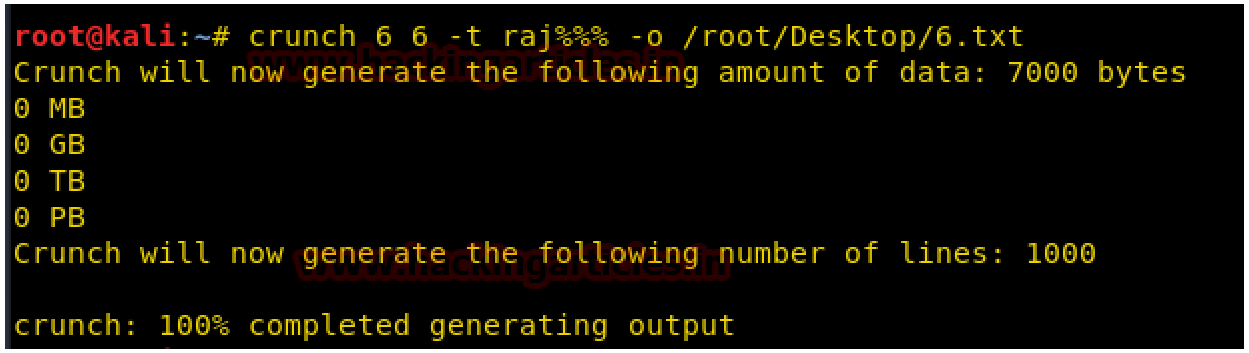
- Use **@** for lowercase alphabets
- Use **,** for uppercase alphabets
- Use **%** for numeric character
- Use **^** for special character symbol

For generating a wordlist that contains 3 numeric characters on the right side of string “raj” for instant **raj123**, we need to execute following command.

Since we have 3 letters from string raj and we are assuming 3 more numeric number after the given string, therefore the minimum length should be sum of string and pattern character.

crunch 6 6 -t raj%%% -o/root/Desktop/6.txt

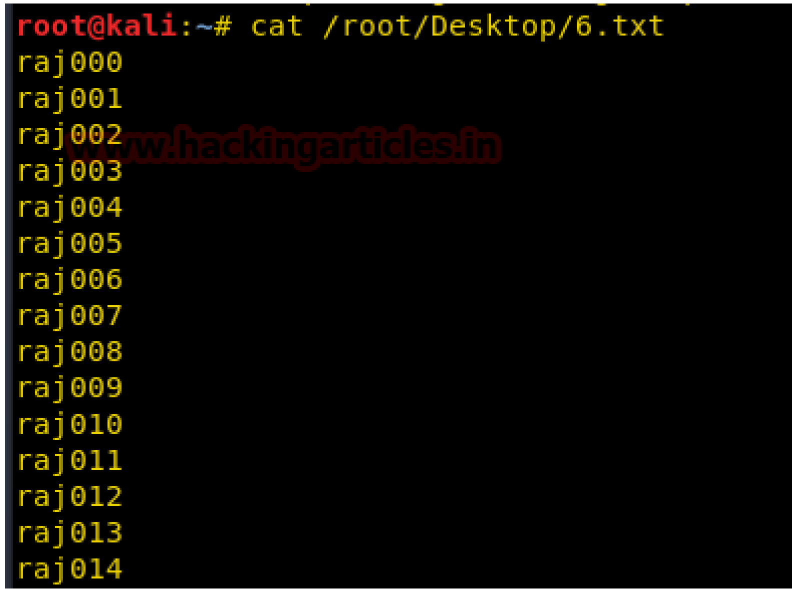
Here –t denotes % pattern is used for editing 3 numeric character.



```
root@kali:~# crunch 6 6 -t raj%%% -o /root/Desktop/6.txt
Crunch will now generate the following amount of data: 7000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 1000

crunch: 100% completed generating output
```

Again we had used **cat command** to read the content from inside **6.txt** file where we can perceive that it has combination of alpha-numeric character.

cat /root/Desktop/6.txt



```
root@kali:~# cat /root/Desktop/6.txt
raj000
raj001
raj002
raj003
raj004
raj005
raj006
raj007
raj008
raj009
raj010
raj011
raj012
raj013
raj014
```

Generate wordlist with Duplicate character limit

Crunch let you bound the repetition of character by **using –d** parameters along with the given pattern.

As we saw, above the pattern for raj%%% starts with raj000 which means every single number will consecutive either twice or thrice such as it will contain word as raj000, raj001, raj111, raj110 and so on in the wordlist.

If you don't wish to create a wordlist with repeated number then you can **use –d** option to set filter for repetition.

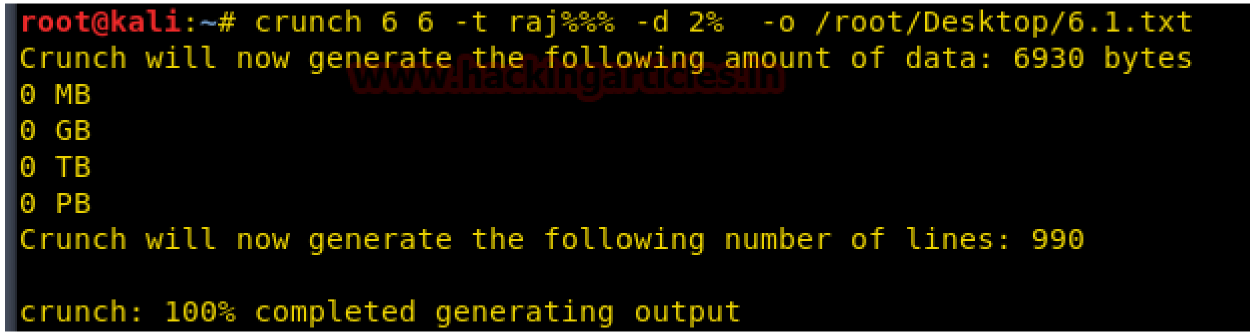
For example: I want to generate a wordlist by using above pattern i.e. raj%%% and consecutive repetition of each number almost twice. For implementing such type of dictionary we need to execute below command.

crunch 6 6 -t raj%%% -d 2% -o/root/Desktop/6.1.txt

here we had use following parameter

–t denotes % pattern is used for editing 3 numeric character

-d denote % pattern is used for editing 3 numeric character with repetition of each number almost twice.

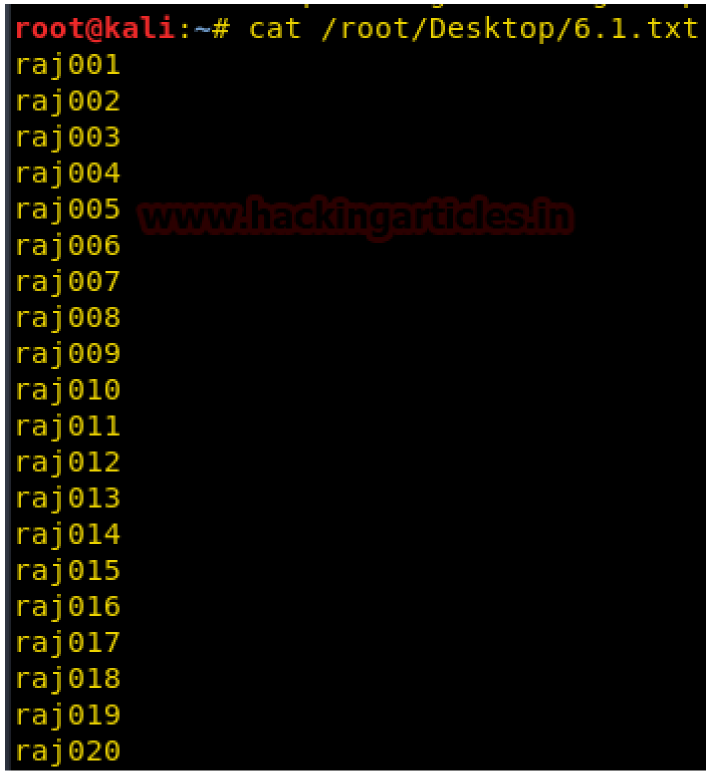


```
root@kali:~# crunch 6 6 -t raj%%% -d 2%  -o /root/Desktop/6.1.txt
Crunch will now generate the following amount of data: 6930 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 990

crunch: 100% completed generating output
```

Again we had used **cat command** to read the content from inside **6.1.txt** file where we can perceive that it has combination of alpha-numeric character with repetition of each number two times.

cat /root/Desktop/6.1.txt

Now if you will compare output file 6.txt and 6.1.txt then you can notice difference of number repetition.



```
root@kali:~# cat /root/Desktop/6.1.txt
raj001
raj002
raj003
raj004
raj005
raj006
raj007
raj008
raj009
raj010
raj011
raj012
raj013
raj014
raj015
raj016
raj017
raj018
raj019
raj020
```

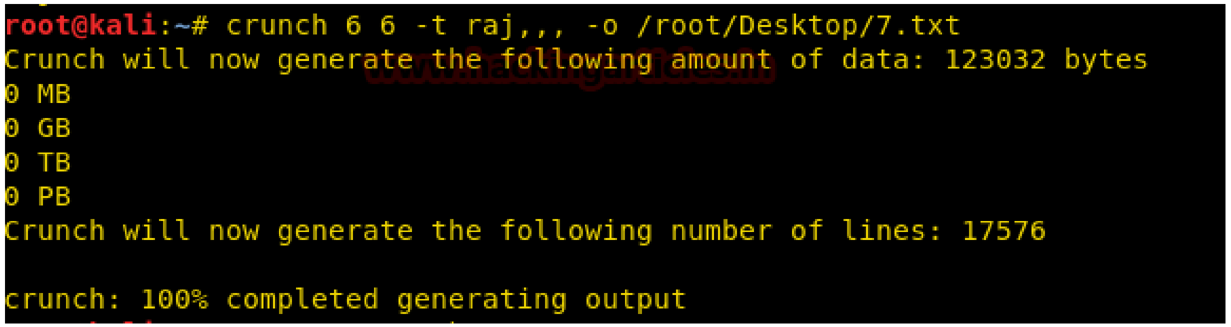
Generate wordlist with Pattern for uppercase letter

For generating a wordlist that contains 3 uppercase characters on the right side of string "raj" for instant **rajABC**, we need to execute following command.

Since we have 3 letters from string raj and we are assuming 3 more uppercase letter after the given string, therefore the minimum length should be sum of string and pattern character.

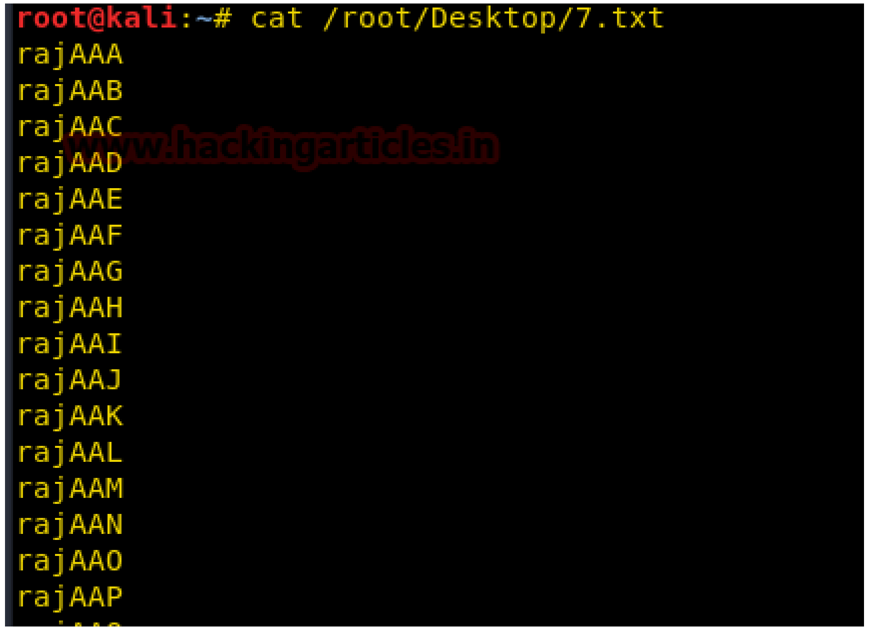
crunch 6 6 -t raj,,, -o/root/Desktop/7.txt

Here –t denotes (,) pattern is used for editing 3 uppercase letter character.


```
root@kali:~# crunch 6 6 -t raj,,, -o /root/Desktop/7.txt
Crunch will now generate the following amount of data: 123032 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 17576

crunch: 100% completed generating output
```

Again we had used **cat command** to read the content from inside **7.txt** file where we can perceive that it has combination of mix-alpha character.

cat /root/Desktop/7.txt

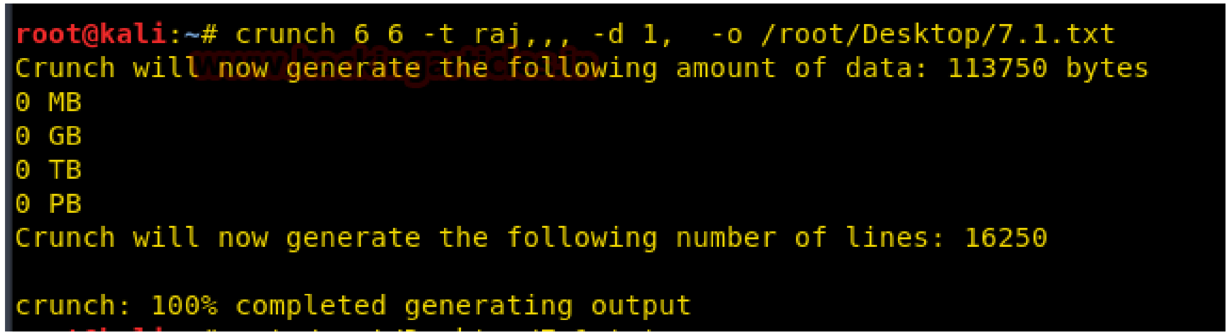

```
root@kali:~# cat /root/Desktop/7.txt
rajAAA
rajAAB
rajAAC
rajAAD
rajAAE
rajAAF
rajAAG
rajAAH
rajAAI
rajAAJ
rajAAK
rajAAL
rajAAM
rajAAN
rajAAO
rajAAP
```

Similarly we can set limit for uppercase letter repletion as done above. So if I want that alphabets should not be consecutive then we can execute given below command for generating such type of dictionary.

crunch 6 6 -t raj,,, -d 1, -o/root/Desktop/7.1.txt

–t denotes (,) pattern is used for editing 3 uppercase character

-d denote (,) pattern is used for editing 3 uppercase character with repetition of each number almost one.

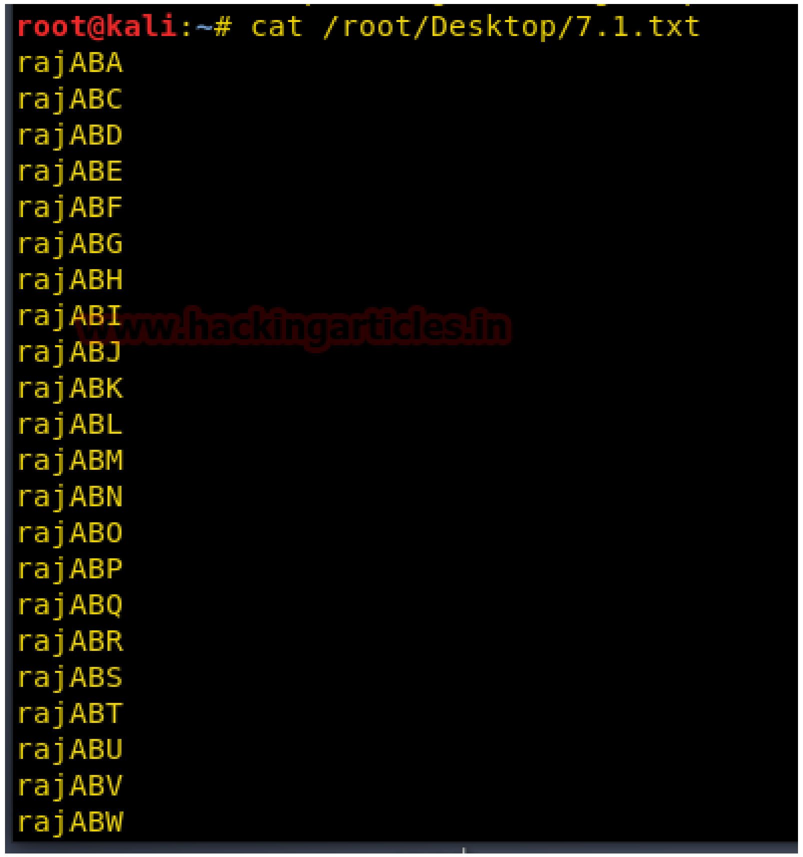


```
root@kali:~# crunch 6 6 -t raj,,, -d 1,  -o /root/Desktop/7.1.txt
Crunch will now generate the following amount of data: 113750 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 16250

crunch: 100% completed generating output
```

Again we had used **cat command** to read the content from inside **7.1.txt** file where we can perceive that it has combination of mix-alpha character with repetition of each number two times.

cat /root/Desktop/7.1.txt

Now if you will compare output file 7.txt and 7.1.txt then you can notice difference of alphabet repetition.



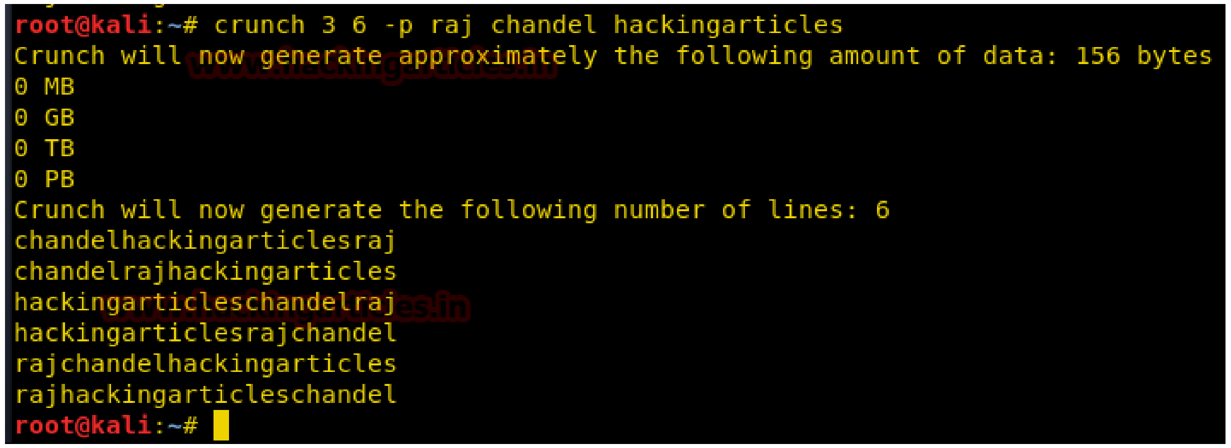
```
root@kali:~# cat /root/Desktop/7.1.txt
rajABA
rajABC
rajABD
rajABE
rajABF
rajABG
rajABH
rajABI
rajABJ
rajABK
rajABL
rajABM
rajABN
rajABO
rajABP
rajABQ
rajABR
rajABS
rajABT
rajABU
rajABV
rajABW
```

Use Permutation for generating wordlist

-p option is used for generating wordlist with help of permutation, here can ignore min and max length of character string. Moreover it can be used with one word string or multiple words string as given below.

crunch 3 6 –p raj chandel hackingarticles

From given below image you can analysis the output result and get maximum number of permutation generated.



```
root@kali:~# crunch 3 6 -p raj chandel hackingarticles
Crunch will now generate approximately the following amount of data: 156 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 6
chandelhackingarticlesraj
chandelrajhackingarticles
hackingarticleschandelraj
hackingarticlesrajchandel
rajchandelhackingarticles
rajhackingarticleschandel
root@kali:~#
```

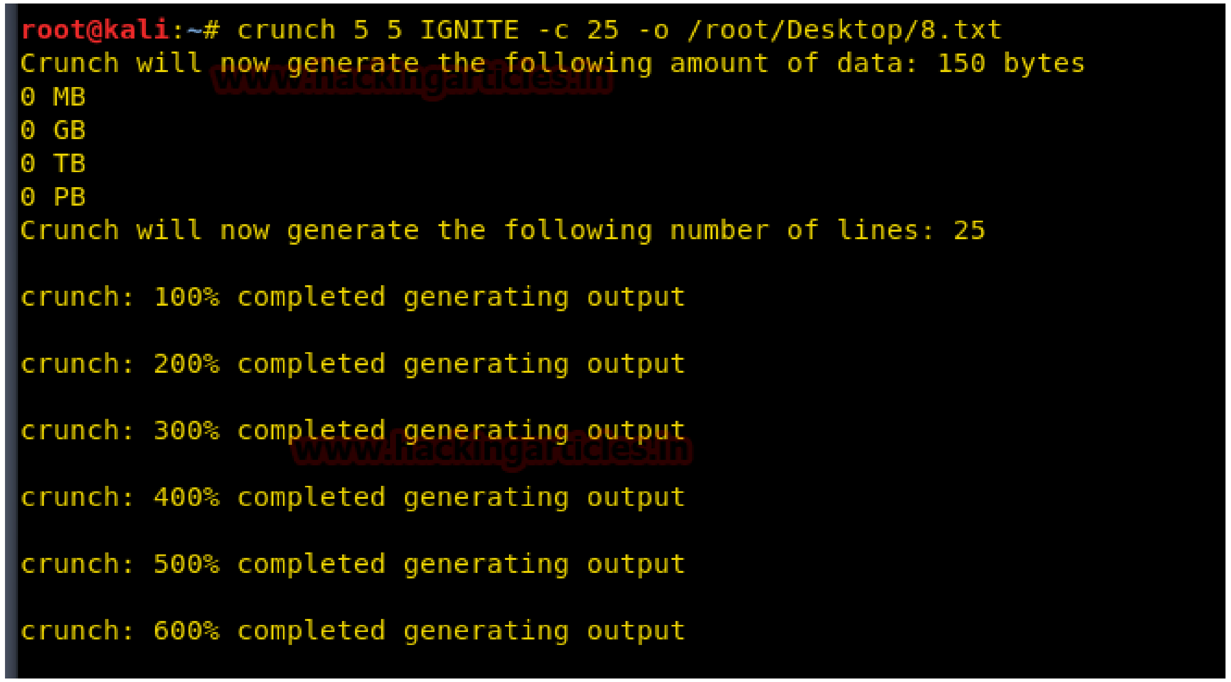
Generate Dictionary with limited words

If you will observe above all output result then you will find crunch has generated dictionary and displays the number of line for each dictionary. For instance text file 0.txt has 18252 number of line and each line contains one word only.

So if you wish to set filter for certain number of line should be generated then execute given below line.

crunch 5 5 IGNITE -c 25 -o /root/Desktop/8.txt

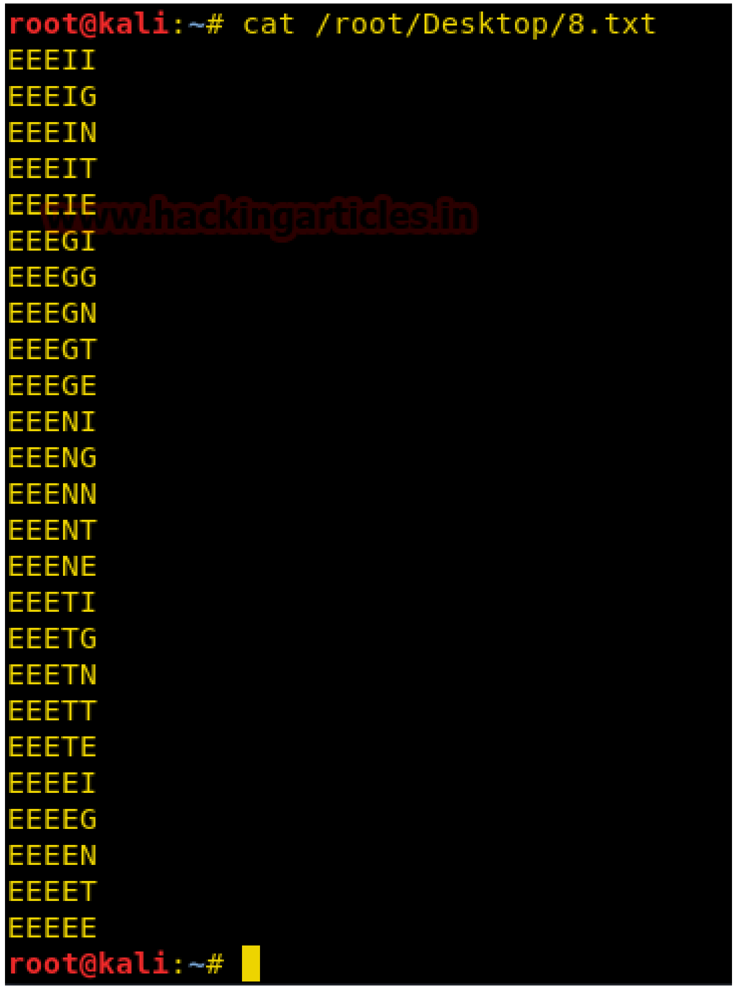
It will generate a dictionary of 25 words only and save output in 8.txt.



```
root@kali:~# crunch 5 5 IGNITE -c 25 -o /root/Desktop/8.txt
Crunch will now generate the following amount of data: 150 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 25

crunch: 100% completed generating output

crunch: 200% completed generating output

crunch: 300% completed generating output

crunch: 400% completed generating output

crunch: 500% completed generating output

crunch: 600% completed generating output
```

Again we had used **cat command** to read the content from inside**8.txt** file where we can perceive that it has only 25 alpha character.

cat /root/Desktop/8.txt



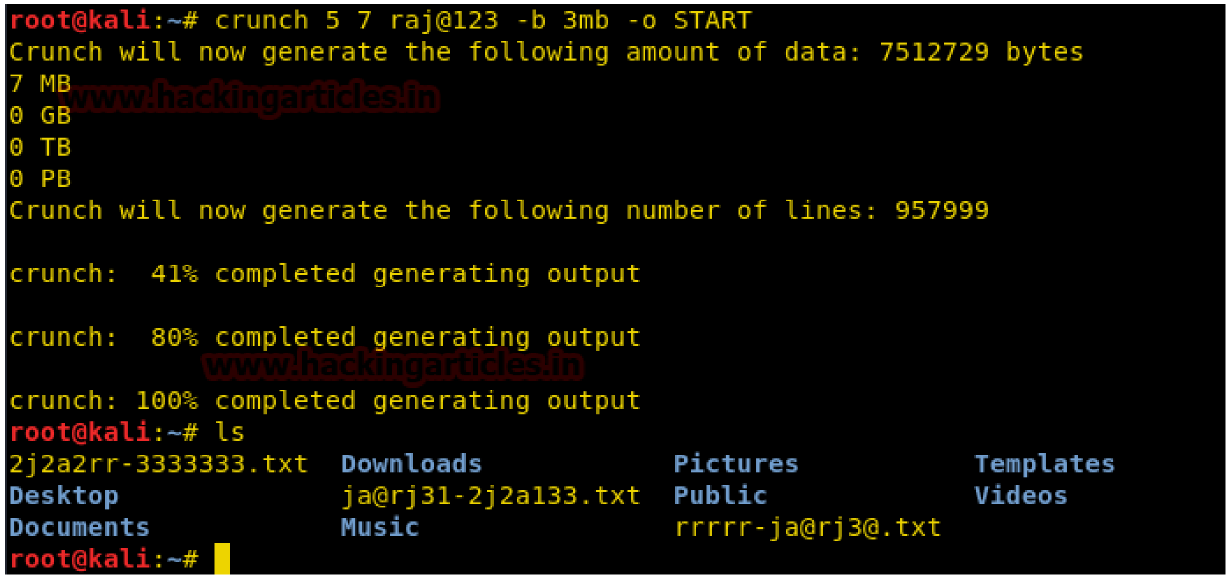
```
root@kali:~# cat /root/Desktop/8.txt
EEEII
EEEIG
EEEIN
EEEIT
EEEIE
EEEGI
EEEGG
EEEGN
EEEGT
EEEGE
EEENI
EEENG
EEENN
EEENT
EEENE
EEETI
EEETG
EEETN
EEETT
EEETE
EEEEI
EEEEG
EEEEN
EEEET
EEEEE
root@kali:~#
```

Wordlist Fragmentation

Use **–b option** for wordlist fragmentation that split a single wordlist into multi wordlist. It is quite useful option for dividing wordlist which is in GB can break into MB.

crunch 5 7 raj@123 -b 3mb –o START

From given below image you can observe that it has divided a 7MB file into three text file.



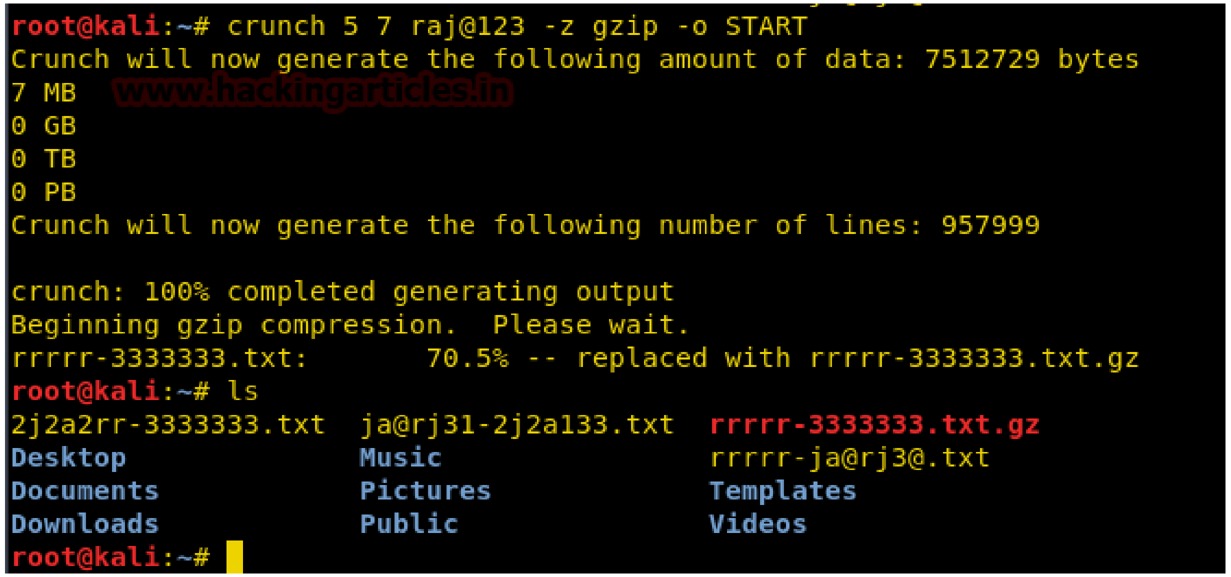
```
root@kali:~# crunch 5 7 raj@123 -b 3mb -o START
Crunch will now generate the following amount of data: 7512729 bytes
7 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 957999

crunch:  41% completed generating output

crunch:  80% completed generating output

crunch: 100% completed generating output
root@kali:~# ls
2j2a2rr-3333333.txt  Downloads            Pictures           Templates
Desktop              ja@rj31-2j2a133.txt  Public             Videos
Documents            Music                rrrrr-ja@rj3@.txt
root@kali:~#
```

Generate compressed Dictionary

Crunch let you generate compress wordlist with **option –z** and other parameters are gzip, bzip2, lzma, and 7z, execute given below command for compression.

crunch 5 7 raj@123 –z gzip –o START

From given below image you can observe that it has generated compress text file.



```
root@kali:~# crunch 5 7 raj@123 -z gzip -o START
Crunch will now generate the following amount of data: 7512729 bytes
7 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 957999

crunch: 100% completed generating output
Beginning gzip compression.  Please wait.
rrrrr-3333333.txt:       70.5% -- replaced with rrrrr-3333333.txt.gz
root@kali:~# ls
2j2a2rr-3333333.txt  ja@rj31-2j2a133.txt  rrrrr-3333333.txt.gz
Desktop              Music                rrrrr-ja@rj3@.txt
Documents            Pictures             Templates
Downloads            Public               Videos
root@kali:~# 
```

Author: AArti Singh is a Researcher and Technical Writer at Hacking Articles an Information Security Consultant Social Media Lover and Gadgets. Contact **here**

Share this:

Like this:

Loading...

ABOUT THE AUTHOR

RAJ CHANDEL

Raj Chandel is a Skilled and Passionate IT Professional especially in IT-Hacking Industry. At present other than his name he can also be called as An Ethical Hacker, A Cyber Security Expert, A Penetration Tester. With years of quality Experience in IT and software industry

PREVIOUS POST

← 6 WAYS TO HACK VNC LOGIN PASSWORD

NEXT POST

NMAP SCAN WITH TIMING PARAMETERS →

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

☐

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.