//code to create hash value in PHP

<?php

echo hash('ripemd160', 'The quick brown fox jumped over the lazy dog.');

?>

// It will give us an output like this:

//output of the hash value created by PHP

ec457d0a974c48d5685a7efa03d137dc8bbde7e3

--------------

// code and output of wordlists

lrwxrwxrwx 1 root root 25 Jun 9 22:18 dirb -> /usr/share/dirb/wordlists

lrwxrwxrwx 1 root root 30 Jun 9 22:18 dirbuster -> /usr/share/dirbuster/wordlists

lrwxrwxrwx 1 root root 35 Jun 9 22:18 dnsmap.txt -> /usr/share/dnsmap/wordlist\_TLAs.txt

lrwxrwxrwx 1 root root 41 Jun 9 22:18 fasttrack.txt -> /usr/share/set/src/fasttrack/wordlist.txt

lrwxrwxrwx 1 root root 45 Jun 9 22:18 fern-wifi -> /usr/share/fern-wifi-cracker/extras/wordlists

lrwxrwxrwx 1 root root 46 Jun 9 22:18 metasploit -> /usr/share/metasploit-framework/data/wordlists

lrwxrwxrwx 1 root root 41 Jun 9 22:18 nmap.lst -> /usr/share/nmap/nselib/data/passwords.lst

-rw-r--r-- 1 root root 139921507 Mar 3 2013 rockyou.txt

lrwxrwxrwx 1 root root 34 Jun 9 22:18 sqlmap.txt -> /usr/share/sqlmap/txt/wordlist.txt

lrwxrwxrwx 1 root root 25 Jun 9 22:18 wfuzz -> /usr/share/wfuzz/wordlist

root@kali:/usr/share/wordlists# gunzip rockyou.txt.gz

---------------

//code and output from a wordlist password file

root@kali:/usr/share/wordlists# cat rockyou.txt

123456

12345

123456789

password

iloveyou

princess

1234567

rockyou

12345678

abc123

nicole

daniel

babygirl

monkey

lovely

jessica

654321

michael

…

// It’s a very long list so I have to cut it short in the output.

------------------------

//output of wordlists directiry

root@kali:/usr/share/wordlists# ls

dirb dirbuster dnsmap.txt fasttrack.txt fern-wifi metasploit nmap.lst rockyou.txt sqlmap.txt wfuzz

---------------------

//output of metaspoilt directory inside wordlists

root@kali:/usr/share/wordlists# cd metasploit

root@kali:/usr/share/wordlists/metasploit# ls

adobe\_top100\_pass.txt multi\_vendor\_cctv\_dvr\_users.txt

av\_hips\_executables.txt namelist.txt

av-update-urls.txt oracle\_default\_hashes.txt

burnett\_top\_1024.txt oracle\_default\_passwords.csv

burnett\_top\_500.txt oracle\_default\_userpass.txt

cms400net\_default\_userpass.txt password.lst

common\_roots.txt piata\_ssh\_userpass.txt

dangerzone\_a.txt postgres\_default\_pass.txt

dangerzone\_b.txt postgres\_default\_userpass.txt

-------------------

//code of creating a dictionary using crunch

root@kali:/usr/share/wordlists/metasploit# crunch 6 6 -t test%%

Crunch will now generate the following amount of data: 700 bytes

0 MB

0 GB

0 TB

0 PB

----------------

//code for crunch manual

man crunch

----------------

// code and output of crunch generating password dictionary

root@kali:/usr/share/wordlists# crunch 6 6 -t test@@ -o test.txt

Crunch will now generate the following amount of data: 4732 bytes

0 MB

0 GB

0 TB

0 PB

Crunch will now generate the following number of lines: 676

crunch: 100% completed generating output

----------------

//checking the existence of test.txt file first

root@kali:/usr/share/wordlists# ls

dirb dnsmap.txt fern-wifi nmap.lst sqlmap.txt wfuzz

dirbuster fasttrack.txt metasploit rockyou.txt test.txt

----------------

root@kali:/usr/share/wordlists# cat test.txt

testaa

testab

testac

testad

testae

testaf

testag

testah

testai

testaj

testak

testal

...

-------------

// code and output of creating thing.txt in nano

nano thing.txt

---------------

//code and output of rsmangler help page

root@kali:/usr/share/wordlists# rsmangler -h

rsmangler v 1.4 Robin Wood (robin@digininja.org) <www.randomstorm.com>

To pass the initial words in on standard in do:

cat wordlist.txt | rsmangler --file - > new\_wordlist.rb

All options are ON by default, these parameters turn them OFF

Usage: rsmangler [OPTION]

--help, -h: show help

--file, -f: the input file, use - for STDIN

--max, -x: maximum word length

--min, -m: minimum word length

--perms, -p: permutate all the words

--double, -d: double each word

--reverse, -r: reverser the word

--leet, -t: l33t speak the word

--full-leet, -T: all posibilities l33t

--capital, -c: capitalise the word

--upper, -u: uppercase the word

--lower, -l: lowercase the word

--swap, -s: swap the case of the word

--ed, -e: add ed to the end of the word

--ing, -i: add ing to the end of the word

--punctuation: add common punctuation to the end of the word

--years, -y: add all years from 1990 to current year to start and end

--acronym, -a: create an acronym based on all the words entered in order and add to word list

--common, -C: add the following words to start and end: admin, sys, pw, pwd

--pna: add 01 - 09 to the end of the word

--pnb: add 01 - 09 to the beginning of the word

--na: add 1 - 123 to the end of the word

--nb: add 1 - 123 to the beginning of the word

--force - don't check ooutput size

--space - add spaces between words

------------------

//code of using a few options of rsmangler and redirect it to a new file

root@kali:/usr/share/wordlists# rsmangler -a -c -d -e -i -l -p -r -s -u -y --pnb -f thing.txt > toughthing.txt

--------------------

// checking the existence of file

root@kali:/usr/share/wordlists# ls

dirb dnsmap.txt fern-wifi nmap.lst rockyou.txt test.txt toughthing.txt

dirbuster fasttrack.txt metasploit passthing.txt sqlmap.txt thing.txt wfuzz

------------------

//output of the file "toughthing.txt"

root@kali:/usr/share/wordlists# cat toughthing.txt

thing

pwthing

thingpw

pwdthing

thingpwd

adminthing

thingadmin

systhing

..

----------------

//copying the "phile.txt" in home directory

root@kali:~# cp /etc/shadow phile.txt

root@kali:~# ls

Desktop Documents Downloads Music phile.txt Pictures Public Templates Videos

-----------------

//code of running John

root@kali:~# john phile.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 5 password hashes with 5 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])

Remaining 3 password hashes with 3 different salts

Press 'q' or Ctrl-C to abort, almost any other key for status

pg@ (pg)

1g 0:00:00:24 1.40% 2/3 (ETA: 23:43:09) 0.04127g/s 362.9p/s 473.9c/s 473.9C/s nina..2001

Use the "--show" option to display all of the cracked passwords reliably

Session aborted

-------------------

//code to show the cracked password

root@kali:~# john --show phile.txt

sanjib:ss:17695:0:99999:7:::

ss:ss:17697:0:99999:7:::

pg:pg@:17697:0:99999:7:::

3 password hashes cracked, 2 left

--------------

//output of the inside of rainbowcrack and going inside

root@kali:~# ls /usr/share/rainbowcrack/

alglib0.so ntlm\_loweralpha#6-6\_0\_3800x335540\_0.rt readme.txt rtc2rt rtmerge

charset.txt rcrack rt2rtc rtgen rtsort

root@kali:~# cd /usr/share/rainbowcrack/

root@kali:/usr/share/rainbowcrack#

---------------

//code to create rainbow table for ntlm hash

root@kali:/usr/share/rainbowcrack# rtgen ntlm loweralpha 6 6 0 3800 335540 0

rainbow table ntlm\_loweralpha#6-6\_0\_3800x335540\_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: 6 - 6

reduce offset: 0x00000000

plaintext total: 308915776

sequential starting point begin from 0 (0x0000000000000000)

generating...

65536 of 335540 rainbow chains generated (0 m 20.9 s)

131072 of 335540 rainbow chains generated (0 m 21.1 s)

196608 of 335540 rainbow chains generated (0 m 20.8 s)

262144 of 335540 rainbow chains generated (0 m 20.7 s)

327680 of 335540 rainbow chains generated (0 m 20.7 s)

335540 of 335540 rainbow chains generated (0 m 2.6 s)

------------------------

//code to test ntlm hash value

root@kali:/usr/share/rainbowcrack# rcrack \*.rt -h 866437cb7a794bce2b727acc0362ee27

-----------------------

//code for creating rainbow table for md5 hash

root@kali:/usr/share/rainbowcrack# rtgen md5 loweralpha 6 6 0 3800 335540 0

rainbow table md5\_loweralpha#6-6\_0\_3800x335540\_0.rt parameters

hash algorithm: md5

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: 6 - 6

reduce offset: 0x00000000

plaintext total: 308915776

sequential starting point begin from 0 (0x0000000000000000)

generating...

65536 of 335540 rainbow chains generated (0 m 30.2 s)

131072 of 335540 rainbow chains generated (0 m 32.6 s)

196608 of 335540 rainbow chains generated (0 m 30.6 s)

262144 of 335540 rainbow chains generated (0 m 30.2 s)

327680 of 335540 rainbow chains generated (0 m 30.9 s)

335540 of 335540 rainbow chains generated (0 m 3.7 s)

------------------

rtgen md5 loweralpha 6 6 0 3800 335540 0

---------------