Password Attacks

THE PROJECT SECURITY SYSTEMS

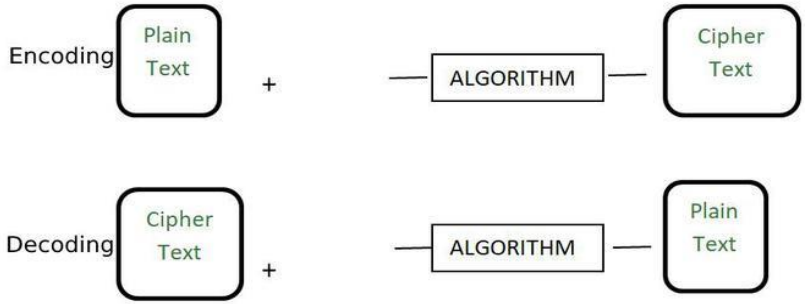
How to storage password

When storage password in database must hashing process before store for keep security it

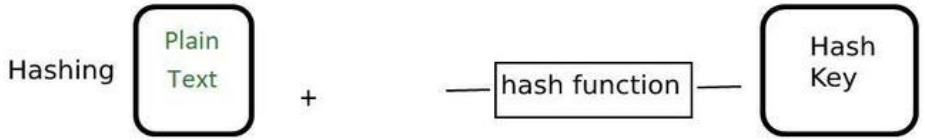
Hashing is the process of transforming any given key or a string of characters into another value. This is usually represented by a shorter, fixed-length value or key that represents and makes it easier to find or employ the original string. The most popular use of hashing is for setting up hash tables.

Note : the process hashing is not same encryption ???

encryption



hashing



Difference Between Hashing and Encryption

ENCRYPTION	HASHING
• It is a process to convert information to a shorter fixed value known as the key that is used to represent the original information. • The purpose of hashing is indexing and retrieving items from the database. The process is very fast. • The hash code or key can not be reversed to the original information by any means. It can only be mapped and the hash code is checked if the hash code is the same the information is the same otherwise not. The original information can not be retrieved. • The output of a hashing algorithm is a fixed-size hash value	• It is the process to encode data securely such that only the authorized user who knows the key or password is able to retrieve the original data for everyone else it is just garbage. • The purpose of encryption is to transform data to keep it secret from others. • The original information can be easily retrieved if we know the encryption key and algorithm used for encryption. • the output of an encryption algorithm is ciphertext of the same size or larger than the original data

Type the password

The divide password for several type and which she :

1. Default password
   She is password which put to software from company the creation and she is knowing for policy
2. Weak password
   Password which is be simple , short and component from sequent number
3. Leaked password
   she is password may be storage ,private and cover in time from time but someone in case can than access to this password and display for public people

Note ➔ (“ for successful attack must be we have good wordlist ?! ”)

Step one for cracking password

For work cracking password must building word list than via some tools existing in kali liunxe

And success this attack depend word list this means that Strong word list Increases the possibility successful attack

Such as : crunch tools , wordlist ready…etc.

There are type for word list:

1. Combine Word List
2. Customized Word List
3. Username Word List

Crunch Tool

Crunch Tools

- Consider about tool working on create word list for using in crack password

- (In case not know any think on tool you can use following comment which will your forget description of tool)
- →$ man crunch

```
       crunch - generate wordlists from a character set

SYNOPSIS
       crunch <min-len> <max-len> [<charset string>] [options]

DESCRIPTION
       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.  The required parameters ar
e:

       min-len
              The minimum length string you want crunch to start at.  This option is re
quired even for parameters that won't use the value.

       max-len
              The maximum length string you want crunch to end at.  This option is requ
ired even for parameters that won't use the value.

       charset string
              You  may  specify character sets for crunch to use on the command line or
 if you leave it blank crunch will use the default character sets.  The order MUST BE l
ower
              case characters, upper case characters, numbers, and then symbols.  If yo
u don't follow this order you will not get the results you want.  You MUST  specify  ei
ther
              values  for  the  character type or a plus sign.  NOTE: If you want to in
clude the space character in your character set you must escape it using the \ characte
r or
              enclose your character set in quotes i.e. "abc ".  See the examples 3, 11
, 12, and 13 for examples.

OPTIONS
       -b number[type]
              Specifies the size of the output file, only works if -o START is used, i.
e.: 60MB  The output files will be in the format of starting letter-ending letter for e
xam-
              ple:  ./crunch  4 5 -b 20mib -o START will generate 4 files: aaaa-gvfed.t
log file: S
```

Crunch Tool

→ $ crunch 1 3 -o list.txt

In this commend the tool will create list contain on word component 3 digits and which start from (**a**) to (**zzz**)

And use lower letters only after put to file named **list.txt**

This is simple word list but word list which use in password attack must be build on privacy policy own with company for successful attacker in break password

```
┌──(wizzo㉿wizzo)-[~]
└─$ crunch 1 3 -o rrr.txt
Crunch will now generate the following amount of data: 72384 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 18278

crunch: 100% completed generating output
```

Crunch Tool

You can get on more and more examples about way building the format then through next command in section example

→$ man crunch

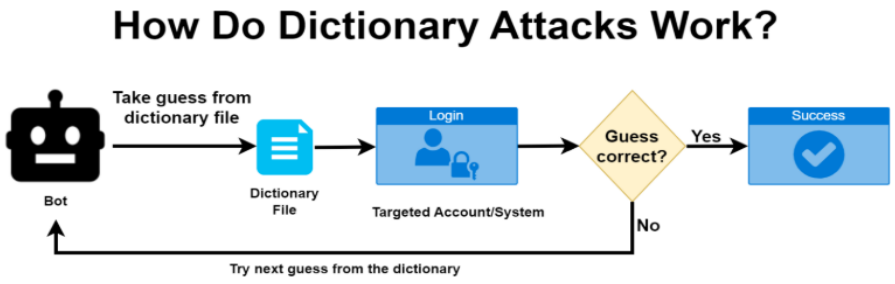
→Section Examples

```
EXAMPLES
       Example 1
       crunch 1 8
       crunch will display a wordlist that starts at a and ends at zzzzzzzz

       Example 2
       crunch 1 6 abcdefg
       crunch will display a wordlist using the character set abcdefg that starts at
       a and ends at gggggg

       Example 3
       crunch 1 6 abcdefg\
       there is a space at the end of the character string.  In order for crunch  to
       use  the space you will need to escape it using the \ character.  In this ex-
       ample you could also put quotes around the letters and not need the  \,  i.e.
       "abcdefg  ".   Crunch will display a wordlist using the character set abcdefg
       that starts at a and ends at (6 spaces)

       Example 4
       crunch 1 8 -f charset.lst mixalpha-numeric-all-space -o wordlist.txt
       crunch will use the mixalpha-numeric-all-space character set from charset.lst
       and  will  write  the  wordlist  to a file named wordlist.txt.  The file will
       start with a and end with "        "

       Example 5
       crunch 8 8  -f  charset.lst  mixalpha-numeric-all-space  -o  wordlist.txt  -t
       @@dog@@@ -s cbdogaaa
       crunch  should generate a 8 character wordlist using the mixalpha-number-all-
       space character set from charset.lst and will write the wordlist  to  a  file
       named wordlist.txt.  The file will start at cbdogaaa and end at "  dog   "

       Example 6
       crunch 2 3 -f charset.lst ualpha -s BB
       crunch with start generating a wordlist at BB and end with ZZZ.  This is use-
       ful if you have to stop generating a wordlist in the middle.  Just do a  tail
 Manual page crunch(1) line 124/382 46% (press h for help or q to quit)
```

After word List

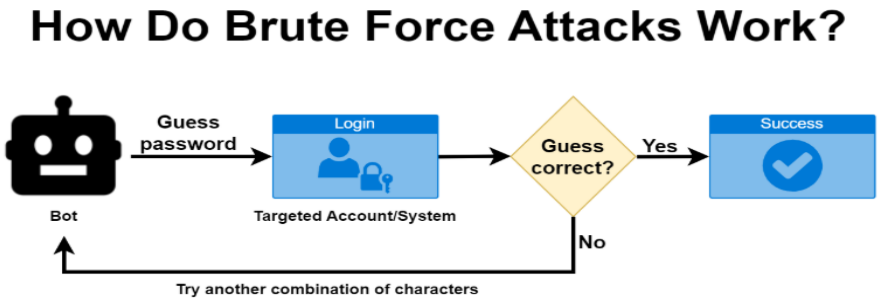
There are more attacks for break password but we will focus on two type then attacks :

1. Dictionary attack
   - Attempts to guess a password by systematically trying out every possible word in a dictionary
   - Fast but limited by the words in the dictionary
   - Usually limited to passwords of a reasonable length



2. Brute force attack
   - Attempts to guess a password by systematically trying out every

   Possible combination of characters
   - Slow and computationally intensive
   - Can guess passwords of any length



hashcat tool

This is tool work on break hash which hide password with use algorithms own to encryption

And available execute Dictionary , Brute force attacks and other password attacks

Advantages

1. Supports a Wide Range of Hash Algorithms and More mode attacks
2. Customizability
3. Open-Source and Active Development

disadvantages

1. Requires Large Datasets for Effective Cracking
2. Slow Performance on CPUs

hashcat tool

To explain for hashcat tool

We are can use this is website (freetoolonline.com) for get hash Based on MD5 for clarification

Work the tool :

Refer for type Dictionary or Straight attack

This is hash which you can get thought objection network or request https for website

Refer for MD5 hash

```
┌──(wizzo㉿wizzo)-[~/Desktop]
└─$ hashcat -a 0 -m 0 4e6ba569d68abd8b8d329956da8d9e07 out.txt
```

Keyword in liunex

This is word list

The choose for type hash

This choose for determine type attacks

hashcat tool

The hashcat support more than type attacks mode and might sum between two types from password attacks

```
Attack mode
        0 = Straight
        1 = Combination
        3 = Brute-force
        6 = Hybrid Wordlist + Mask
        7 = Hybrid Mask + Wordlist
```

And hashcat support version mode hash some it's apparition new and the other old and popular are MD5 , SHA256, HMAC .. etc

```
Hash types
        0 = MD5
        10 = md5($pass.$salt)
        20 = md5($salt.$pass)
        30 = md5(unicode($pass).$salt)
        40 = md5($salt.unicode($pass))
        50 = HMAC-MD5 (key = $pass)
        60 = HMAC-MD5 (key = $salt)
        100 = SHA1
        110 = sha1($pass.$salt)
        120 = sha1($salt.$pass)
        130 = sha1(unicode($pass).$salt)
        140 = sha1($salt.unicode($pass))
        150 = HMAC-SHA1 (key = $pass)
        160 = HMAC-SHA1 (key = $salt)
        200 = MySQL323
        300 = MySQL4.1/MySQL5
        400 = phpass, MD5(Wordpress), MD5(phpBB3), MD5(Joomla)
        500 = md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
        900 = MD4
        1000 = NTLM
        1100 = Domain Cached Credentials (DCC), MS Cache
        1400 = SHA256
        1410 = sha256($pass.$salt)
        1420 = sha256($salt.$pass)
        1430 = sha256(unicode($pass).$salt)
        1431 = base64(sha256(unicode($pass)))
        1440 = sha256($salt.unicode($pass))
        1450 = HMAC-SHA256 (key = $pass)
        1460 = HMAC-SHA256 (key = $salt)
        1600 = md5apr1, MD5(APR), Apache MD5
        1700 = SHA512
        1710 = sha512($pass.$salt)
        1720 = sha512($salt.$pass)
        1730 = sha512(unicode($pass).$salt)
        1740 = sha512($salt.unicode($pass))
 Manual page hashcat(1) line 359/489 83% (press h for help or q to quit)
```

hashcat tool

Example: work on break this is hash **485c4fbb3e8a794482289a00193eef35** by use hashcat tool

Type hash MD5 :

-a = 0 (because we are use Dictionary attack)

-m = 0 (because is was complete determine type hash in Question)

485c4fbb3e8a794482289a00193eef35 = jm8s
This hash will work hashcat on storage to file

```
Dictionary cache hit:
* Filename..: out.txt
* Passwords.: 175760
* Bytes.....: 878800
* Keyspace..: 175760

485c4fbb3e8a794482289a00193eef35:jm8s

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 485c4fbb3e8a794482289a00193eef35
Time.Started.....: Tue Apr 29 04:21:05 2025 (0 secs)
Time.Estimated...: Tue Apr 29 04:21:05 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (out.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  3248.1 kH/s (0.20ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 64512/175760 (36.70%)
Rejected.........: 0/64512 (0.00%)
Restore.Point....: 62976/175760 (35.83%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: ji2e → jo1f
Hardware.Mon.#1..: Util: 33%

Started: Tue Apr 29 04:21:05 2025
Stopped: Tue Apr 29 04:21:07 2025
```

John on Ripper

John the Ripper is an offline password cracking tool that was developed in 1996 by Openwall Project. It is notable for supporting a diversity of password formats.

(Initially, I called it Cracker John, but a friend suggested the name John the Ripper.)

Advantages

1. Built on Unix what make available on multiple platforms
2. Determent type hash automatically
3. Available different type password attacks

Disadvantages

1. Limited algorithm support on GPU
2. Requires manual config for multi-GPU support

John on Ripper

The John supports multi-password attacks mode but might complex for Beginner in use

For work Brute force attack with john we will writer (John - - format= MD5 hash.txt) in this is case we are not determent word list for execute attack

This is example for attack Brute force with use john

Key word for determine type hash

Keyword in liunx

```
┌──(wizzo㉿wizzo)-[~/Desktop]
└─$ john --format=zip  lue.txt
```

This is file which contain hash

In case we want work Dictionary Attack with simple

will use keyword (--wordlist= (set word list))

```
┌──(wizzo㉿wizzo)-[~/Desktop]
└─$ john --format=zip --wordlist=output.txt lue.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Cost 1 (HMAC size) is 42 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
raf6W*          (test2.zip/test)
1g 0:00:00:00 DONE (2025-04-30 07:22) 1.492g/s 73361p/s 73361c/s 73361C/s raf0D)..raf7H
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

John on Ripper

The important keyword which may be use when handle with john the ripper :

-- format : this key word work on determent type for hash which want with break

-- wordlist : the choose is for take ability word list password

--max-length : for determent max length for password

--min-length: for determent min length for password

There are more and more key word others but we are not discuss them they are can see description for the tool

John on Ripper

Example: work on break this is hash **485c4fbb3e8a794482289a00193eef35** by use john the ripper tool and using Butter Force mode

Type hash MD5 :

Take The john the ripper 2 secerned for break

this hash

485c4fbb3e8a794482289a00193eef35 = jm8s

```
┌──(wizzo㉿wizzo)-[~/Desktop]
└─$ john --max-length=6 --min-length=4 --format=Raw-MD5 value.hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=3
Proceeding with single, rules:Single, lengths:4-6
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, lengths: 4-6
Proceeding with incremental:ASCII, lengths: 4-6
jm8s             (?)
1g 0:00:00:02 DONE 3/3 (2025-05-01 03:41) 0.4347g/s 42573Kp/s 42573Kc/s 42573KC/s ju86.
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords relia
Session completed.

┌──(wizzo㉿wizzo)-[~/Desktop]
└─$
```

Hydra tool

Hydra is a popular and powerful tool that is commonly used in penetration testing. It is designed to automate the process of cracking passwords or performing brute-force attacks on various protocols and service

Advantages

1. Parallel Attacks
2. The Hydra tool supports various protocols such as HTTP, FTP, SSH, etc
3. very fast and flexible

Disadvantages

1. Not a smart tool
2. Do not bypass multi-factor authentication (2FA)
3. Many services limit the number of attempts to it, which makes it slow.
4. Attacks using Hydra are easily detected by intrusion detection systems (I (WAF/IPS).

Hydra tool

For set username

Key word

Username for test

Set for word list password

This word list

This key for exist in correct login

This key for visualization

Where will enter login and
 password and what is result
in correct login

```
┌──(wizzo㉿wizzo)-[~]
└─$ hydra -l admin -P passwordlist.txt -f -V -w 1000 http-post-form "/DVWA/vulnerabilities/brute/index.php usernam
oed:^PASS^ :F=welcome to the password protected area admin " 127.0.0.1:80
```

the attack will execute localhost