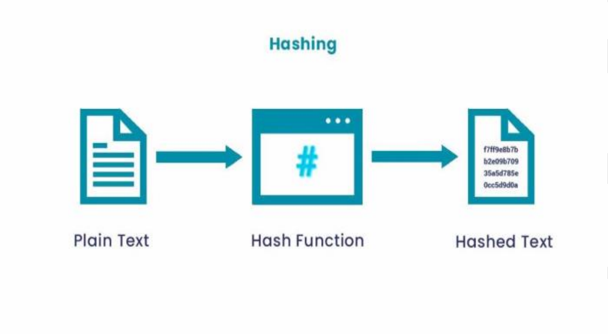
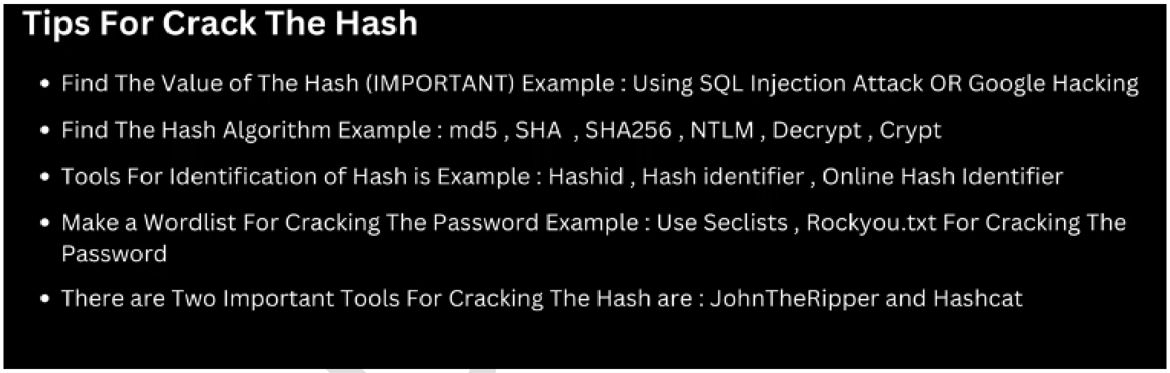
1. a) Password Cracking

Objective: To understand password vulnerabilities and the importance of strong password
Tools: Hashcat, John the Ripper

What is Password Hashing?

Hashing is the process of converting an alphanumeric string into a fixed-size string by using a hash function. A hash function is a mathematical function that takes in the input string and generates another alphanumeric string.





Tips For Crack The Hash

- Find The Value of The Hash (IMPORTANT) Example : Using SQL Injection Attack OR Google Hacking
- Find The Hash Algorithm Example : md5 , SHA , SHA256 , NTLM , Decrypt , Crypt
- Tools For Identification of Hash is Example : Hashid , Hash identifier , Online Hash Identifier
- Make a Wordlist For Cracking The Password Example : Use Seclists , Rockyou.txt For Cracking The Password
- There are Two Important Tools For Cracking The Hash are : JohnTheRipper and Hashcat

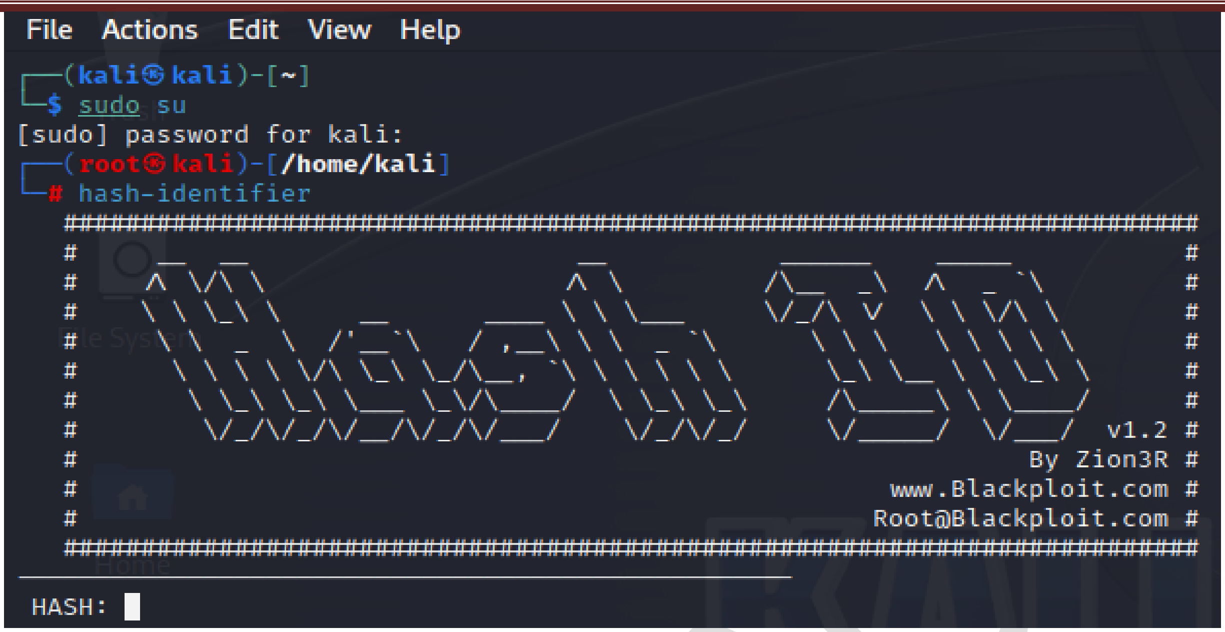
Hashcat : Hashcat is a fast password recovery tool that helps break complex password hashes.
Prerequisites: Generate the hash value using

https://www.browserling.com/tools/all-hashes

Example: 42f749ade7f9e195bf475f37a44cafcb (Password123)
48bb6e862e54f2a795ffc4e541caed4d (easy)

Hash Analyzer: Tunnelsup.com to verify the hash value

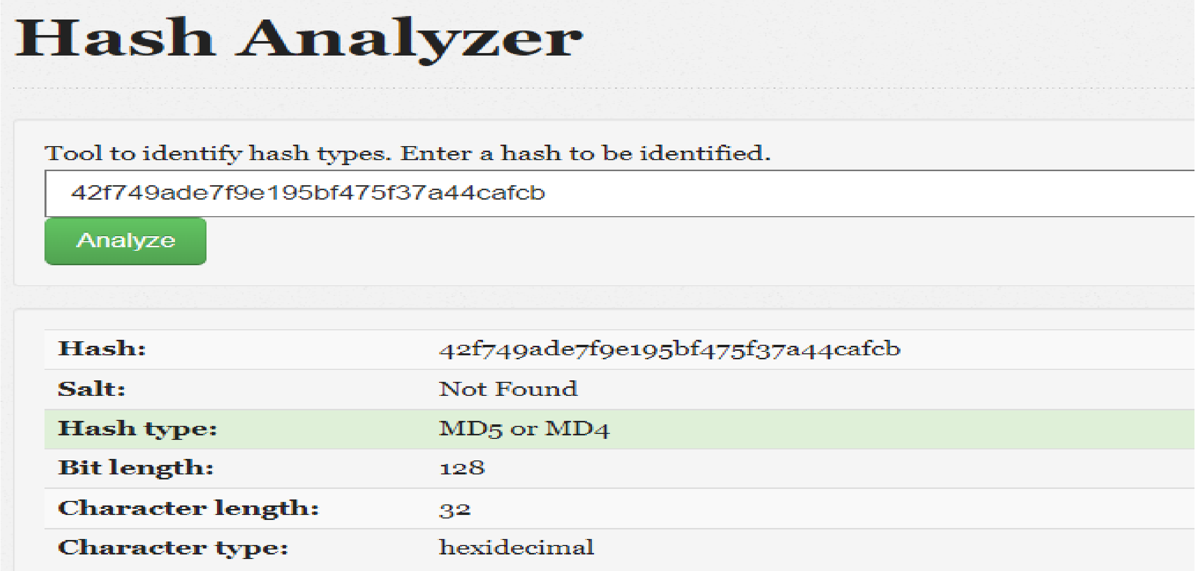
Step1: Become a root user by providing **sudo su**
Step2: Open hash identifier



```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH:
```

Step3: Paste the hash value press enter it will show possible hashs

Step4: Verify the hash using hash analyzer (Tunnelsup.com)



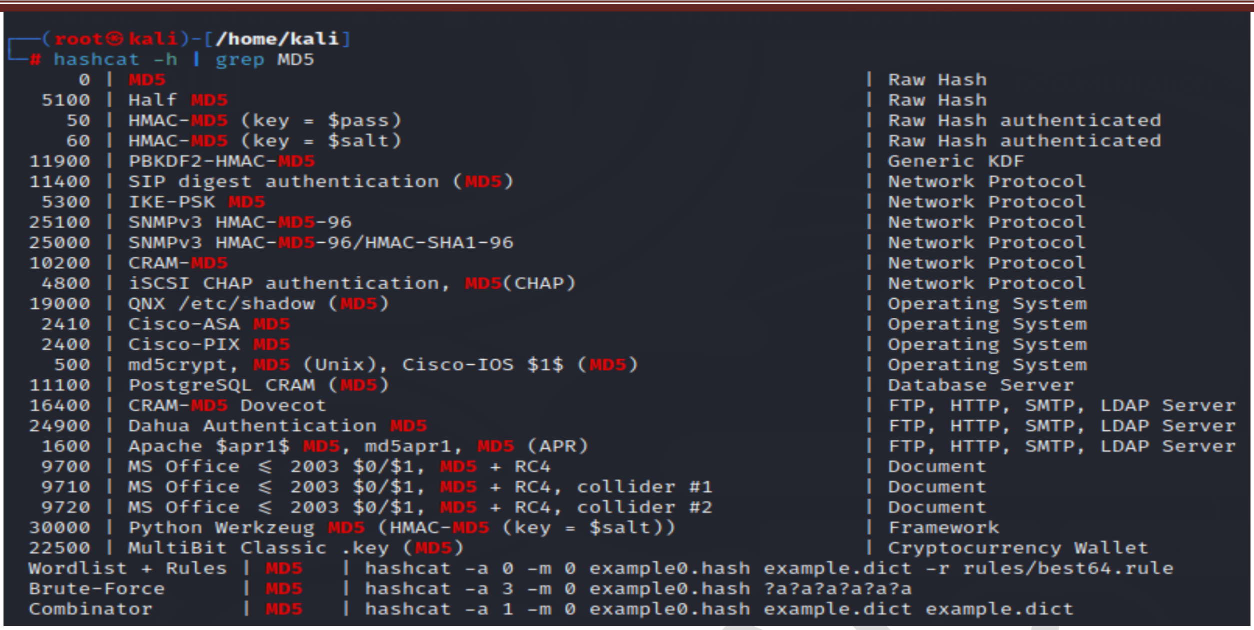
Hash Analyzer

Tool to identify hash types. Enter a hash to be identified.

42f749ade7f9e195bf475f37a44cafcb

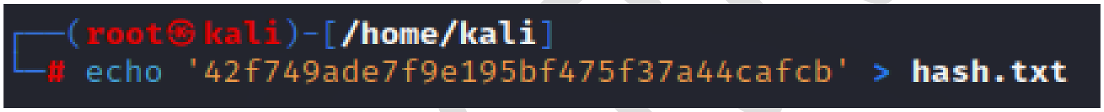
Hash:	42f749ade7f9e195bf475f37a44cafcb
Salt:	Not Found
Hash type:	MD5 or MD4
Bit length:	128
Character length:	32
Character type:	hexidecimal

Step5: Identify the module of MD5 (Since we are dealing with MD5, but in other case module value would be different)



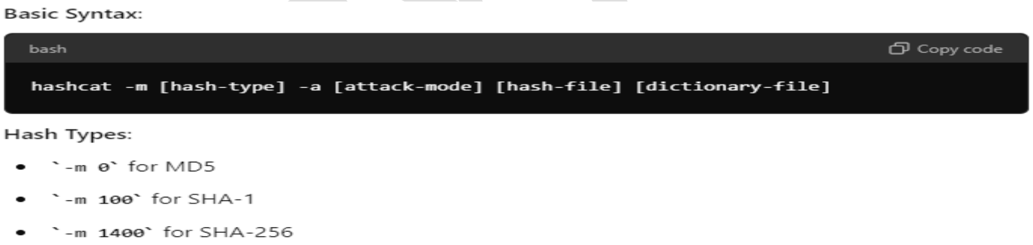
```
┌──(root㉿kali)-[/home/kali]
└─# hashcat -h | grep MD5
      0 | MD5                                                        | Raw Hash
   5100 | Half MD5                                                   | Raw Hash
     50 | HMAC-MD5 (key = $pass)                                     | Raw Hash authenticated
     60 | HMAC-MD5 (key = $salt)                                     | Raw Hash authenticated
  11900 | PBKDF2-HMAC-MD5                                            | Generic KDF
  11400 | SIP digest authentication (MD5)                            | Network Protocol
   5300 | IKE-PSK MD5                                                | Network Protocol
  25100 | SNMPv3 HMAC-MD5-96                                         | Network Protocol
  25000 | SNMPv3 HMAC-MD5-96/HMAC-SHA1-96                            | Network Protocol
  10200 | CRAM-MD5                                                   | Network Protocol
   4800 | iSCSI CHAP authentication, MD5(CHAP)                       | Network Protocol
  19000 | QNX /etc/shadow (MD5)                                      | Operating System
   2410 | Cisco-ASA MD5                                              | Operating System
   2400 | Cisco-PIX MD5                                              | Operating System
    500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)                  | Operating System
  11100 | PostgreSQL CRAM (MD5)                                      | Database Server
  16400 | CRAM-MD5 Dovecot                                           | FTP, HTTP, SMTP, LDAP Server
  24900 | Dahua Authentication MD5                                   | FTP, HTTP, SMTP, LDAP Server
   1600 | Apache $apr1$ MD5, md5apr1, MD5 (APR)                      | FTP, HTTP, SMTP, LDAP Server
   9700 | MS Office ≤ 2003 $0/$1, MD5 + RC4                          | Document
   9710 | MS Office ≤ 2003 $0/$1, MD5 + RC4, collider #1             | Document
   9720 | MS Office ≤ 2003 $0/$1, MD5 + RC4, collider #2             | Document
  30000 | Python Werkzeug MD5 (HMAC-MD5 (key = $salt))               | Framework
  22500 | MultiBit Classic .key (MD5)                                | Cryptocurrency Wallet
  Wordlist + Rules | MD5    | hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
  Brute-Force      | MD5    | hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?a
  Combinator       | MD5    | hashcat -a 1 -m 0 example0.hash example.dict example.dict
```

Step6: Store hashes into any file say hash.txt



```
┌──(root㉿kali)-[/home/kali]
└─# echo '42f749ade7f9e195bf475f37a44cafcb' > hash.txt
```

Step7: Letus crack the password (hash.txt)



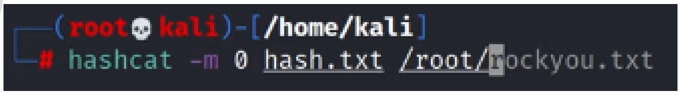
Basic Syntax:

```bash
hashcat -m [hash-type] -a [attack-mode] [hash-file] [dictionary-file]
```

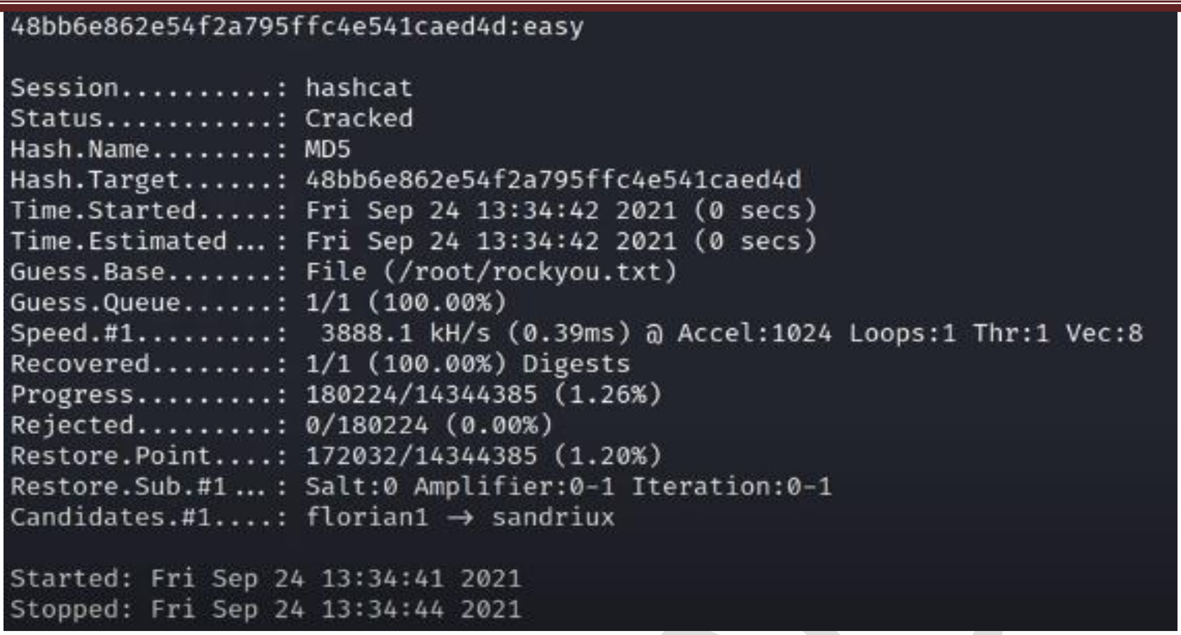
Hash Types:

- `-m 0` for MD5
- `-m 100` for SHA-1
- `-m 1400` for SHA-256

To Search rockyou file in your system provide the command in terminal →**locate rockyou.txt**



```
┌──(root💀kali)-[/home/kali]
└─# hashcat -m 0 hash.txt /root/rockyou.txt
```



```
48bb6e862e54f2a795ffc4e541caed4d:easy

Session..........: hashcat
Status...........: Cracked
Hash.Name........: MD5
Hash.Target......: 48bb6e862e54f2a795ffc4e541caed4d
Time.Started.....: Fri Sep 24 13:34:42 2021 (0 secs)
Time.Estimated...: Fri Sep 24 13:34:42 2021 (0 secs)
Guess.Base.......: File (/root/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  3888.1 kH/s (0.39ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 180224/14344385 (1.26%)
Rejected.........: 0/180224 (0.00%)
Restore.Point....: 172032/14344385 (1.20%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: florian1 → sandriux

Started: Fri Sep 24 13:34:41 2021
Stopped: Fri Sep 24 13:34:44 2021
```

Password123: 42f749ade7f9e195bf475f37a44cafcb

Ref: https://www.youtube.com/watch?v=fVgzY5OJeIE

----

1. **b) John the ripper:** John the Ripper (JtR) is a powerful password cracking tool widely used by security professionals and pen testers.

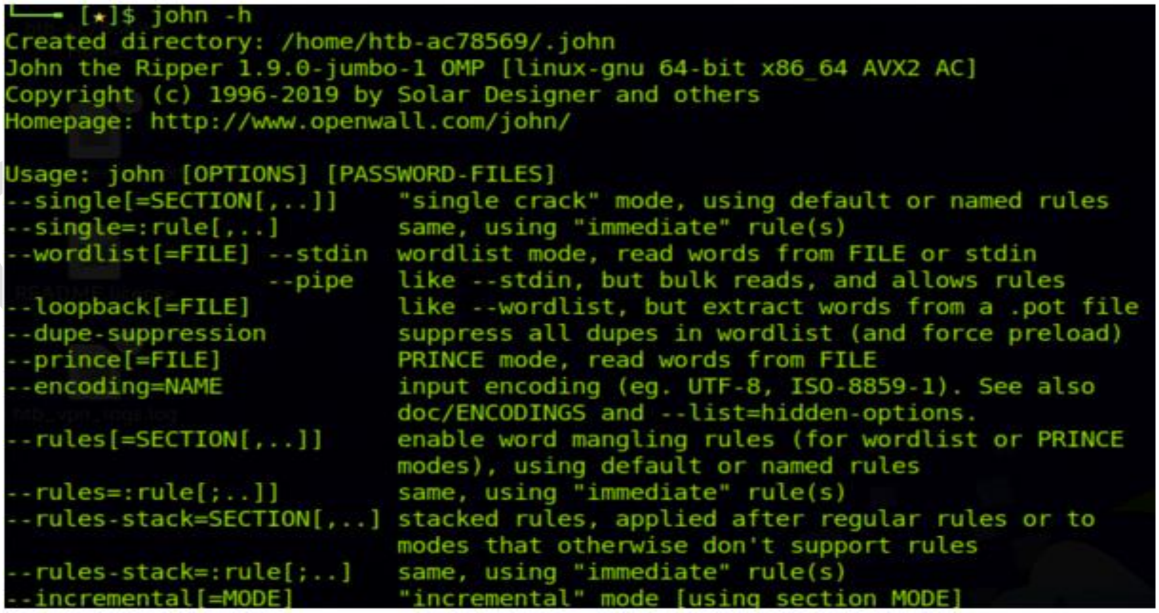
Let's get cracking.

If you are using Kali Linux, John is pre-installed. You can use John by typing the following command:

$ john

The help command can also be used as a reference when working with John.

$ john –h



```
└── [★]$ john -h
Created directory: /home/htb-ac78569/.john
John the Ripper 1.9.0-jumbo-1 OMP [linux-gnu 64-bit x86_64 AVX2 AC]
Copyright (c) 1996-2019 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION[,..]]    "single crack" mode, using default or named rules
--single=:rule[,..]        same, using "immediate" rule(s)
--wordlist[=FILE] --stdin  wordlist mode, read words from FILE or stdin
                  --pipe   like --stdin, but bulk reads, and allows rules
--loopback[=FILE]          like --wordlist, but extract words from a .pot file
--dupe-suppression         suppress all dupes in wordlist (and force preload)
--prince[=FILE]            PRINCE mode, read words from FILE
--encoding=NAME            input encoding (eg. UTF-8, ISO-8859-1). See also
                           doc/ENCODINGS and --list=hidden-options.
--rules[=SECTION[,..]]     enable word mangling rules (for wordlist or PRINCE
                           modes), using default or named rules
--rules=:rule[;..]]        same, using "immediate" rule(s)
--rules-stack=SECTION[,..] stacked rules, applied after regular rules or to
                           modes that otherwise don't support rules
--rules-stack=:rule[;..]   same, using "immediate" rule(s)
--incremental[=MODE]       "incremental" mode [using section MODE]
```

How to Use John the Ripper

The following three modes are used in most of the use cases.

1. Single crack mode
2. Wordlist mode
3. Incremental mode

Let’s look at each one of them in detail.

What is Single Crack Mode?

In single-crack mode, John takes a string and generates variations of that string in order to generate a set of passwords.

variations (STEALTH, Stealth, STealth, and so on).

We use the “format” flag to specify the hash type and the “single” flag to let John know we want to use the single crack mode. We will also create a crack.txt file which will contain the username and the hash value of the password.

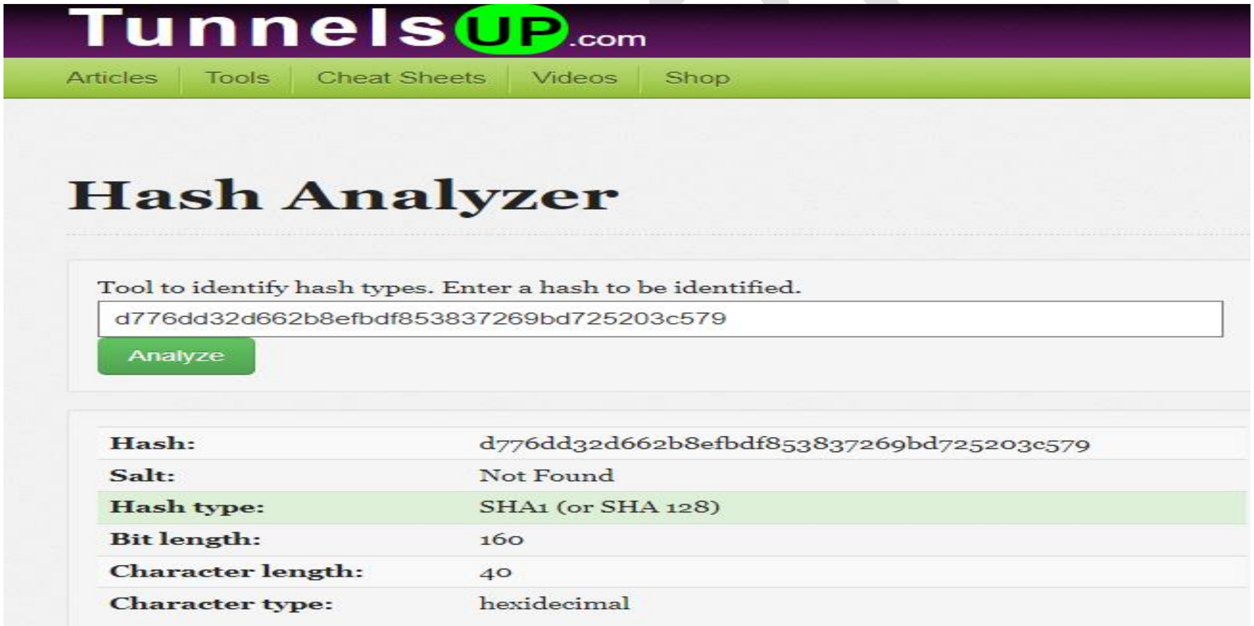
stealth:d776dd32d662b8efbdf853837269bd725203c579

Now we can use the following command to use John’s single crack mode:

```
$ john --single --format=raw-sha1 crack.txt
```

And here is the result. You can see that John has successfully found the correct password “StEaLtH”.

Step1: Identify the hash algorithm used, by using open source platform



Hash Analyzer

Tool to identify hash types. Enter a hash to be identified.

d776dd32d662b8efbdf853837269bd725203c579

Hash:	d776dd32d662b8efbdf853837269bd725203c579
Salt:	Not Found
Hash type:	SHA1 (or SHA 128)
Bit length:	160
Character length:	40
Character type:	hexidecimal

OR

Provide command in terminal as **hashid hash#**

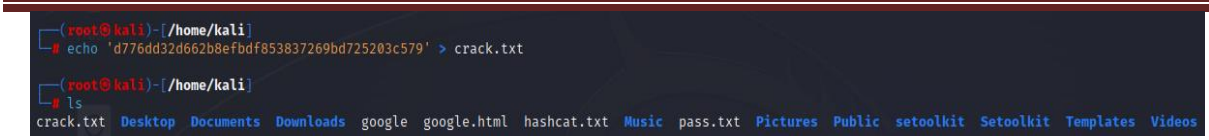
OR

Use online platform: https://hashes.com/en/tools/hash_identifier

OR

Crackstation.com

Step2: save the hash value by using **echo** command

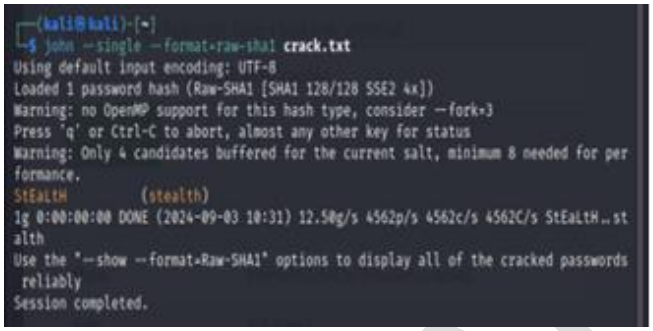


```
┌──(root㉿kali)-[/home/kali]
└─# echo 'd776dd32d662b8efbdf853837269bd725203c579' > crack.txt

┌──(root㉿kali)-[/home/kali]
└─# ls
crack.txt  Desktop  Documents  Downloads  google  google.html  hashcat.txt  Music  pass.txt  Pictures  Public  setoolkit  Setoolkit  Templates  Videos
```

Step3: Crack the password using command

```
$ john --single --format=raw-sha1 crack.txt
```



```
┌──(kali㉿kali)-[~]
└─$ john --single --format=raw-sha1 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for per
formance.
StEaLtH          (stealth)
1g 0:00:00:00 DONE (2024-09-03 10:31) 12.50g/s 4562p/s 4562c/s 4562C/s StEaLtH..st
alth
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords
 reliably
Session completed.
```

Ref: https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/

https://www.youtube.com/watch?v=kuse9Nbs-bI&ab_channel=ManishM.Shivanandhan