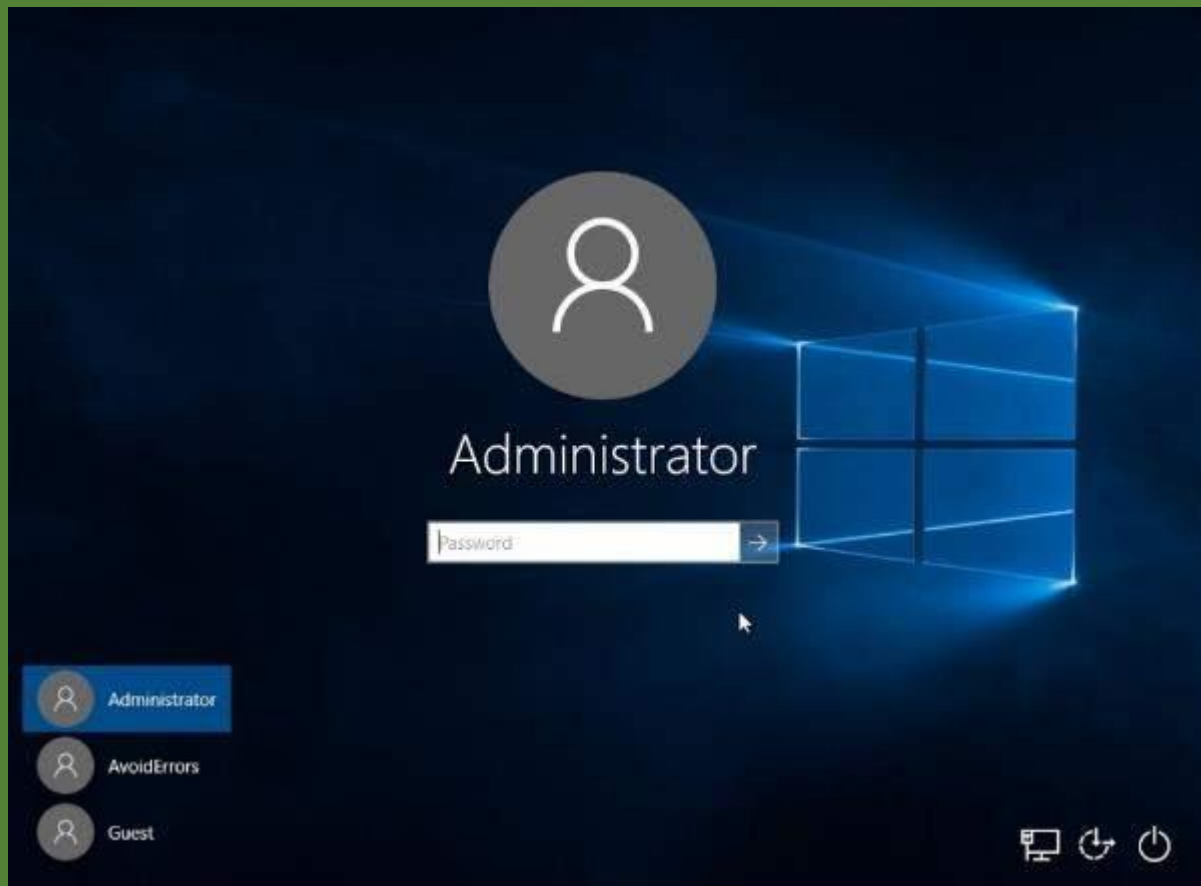


PASSWORD CRACKING OF WINDOWS OPERATING SYSTEM



Author: Sunny thakur

Table of Contents

INTRODUCTION TO PASSWORD CRACKING	2
PASSWORD CRACKING TECHNIQUES	3
TOOL I: MIMIKATZ TOOL	4
GETTING HASH OF PASSWORD WITH MIMIKATZ TOOL	5
TOOL II: HASHCAT TOOL	16
RECOVERING PLAINTEXT FROM NTLM HASH WITH HASHCAT	

TOOL 16
COUNTERMEASURES 27
REFERENCES 29

INTRODUCTION TO PASSWORD CRACKING

- Passwords are used to protect the system from an unauthorized access.
- Computers with Windows operating system stores password in Security Account Manager (SAM) file in the form of New Technology LAN Manager (NTLM) hash.
- Passwords are stored in the form of hash due to its irreversible property. This means that password in plaintext can be converted to hash but a hash can't be converted back to plaintext.
- Password cracking in Windows operating system is a process to recover passwords from a SAM file.
- The purpose of password cracking is to recover forgotten password. The forensic team can perform password cracking on a computer system to recover the data after getting the password.
- This is usually accomplished by recovering the passwords from data stored in the SAM file in the form of NTLM hash value.



Figure 1: Password Cracking

PASSWORD CRACKING TECHNIQUES

The password cracking techniques are discussed as follows:

- **BRUTE FORCE:** A brute force technique is an attempt to crack passwords using permutation and combination approach. This method takes a lot of time and memory consumption depending on the length and complexity of password.
- **DICTIONARY:** A dictionary technique is an attempt to store in-build passwords in a file known as dictionary. Instead of trying all combination of passwords, it creates a word-list of most common passwords and calculates the hash values while cracking the passwords. It will only able to crack the password if it is stored in dictionary file. This technique takes less time as compared to brute-force technique to crack the password.

- **RAINBOW TABLES:** This technique is same as dictionary, but instead of calculating hash vales during password cracking; it stores the in-built hash values of password in the tables. Thus, this technique takes less time as compared to brute-force and dictionary technique to crack the password.

TOOL I: MIMIKATZ TOOL

- The Mimikatz tool [1] was first developed in 2007 by Benjamin Delpy.
- Mimikatz is an open-source application and postexploitation Windows operating system tool that allows users to view authentication credentials.
- This tool provides hashes from SAM file of Windows operating system to users.
- Windows store password data in an NTLM hash. The forensics team can use Mimikatz tool to get the hash string and use hashcat tool to get plain text and pass it to the target computer to login.

GETTING HASH OF PASSWORD WITH MIMIKATZ TOOL

The NTLM hash of password can be accessed with mimikatz tool with following steps:

Step 1: Open Run box by clicking “Window + R” on keyboard and type “regedit” as shown in Figure 2 and Figure 3 respectively. Click “OK” to proceed.

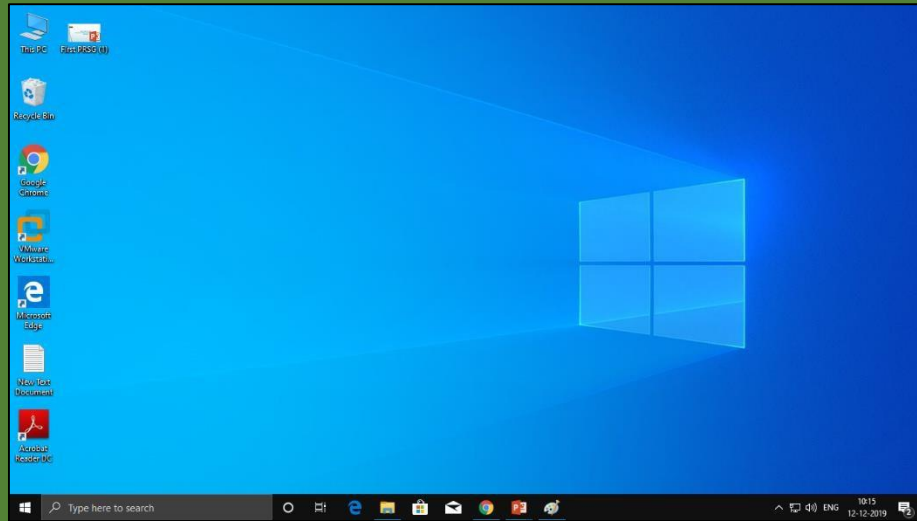


Figure 2: Windows 10 operating system

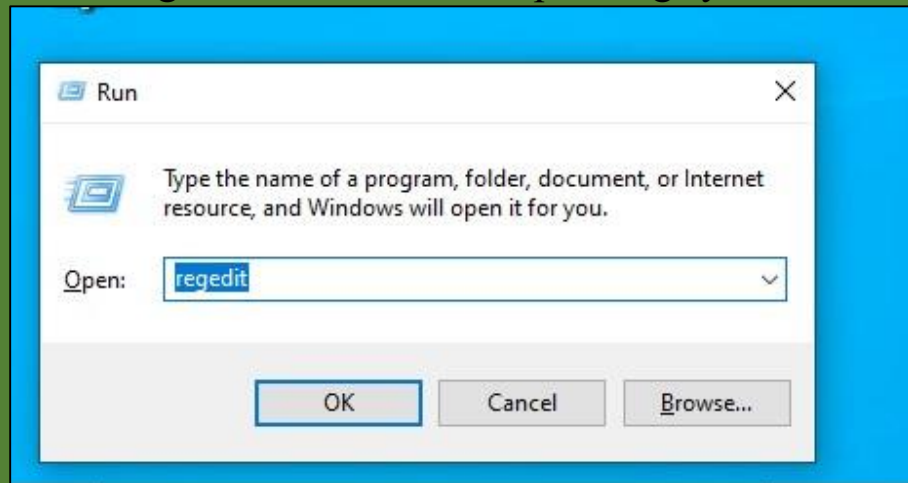


Figure 3: Opening Windows registry file

Step 2: A Registry Editor file with SAM and SYSTEM folder will open as shown in Figure 4.

The SAM and SYSTEM files are located in “C:\Windows\System32\config” path as shown in Figure 5.

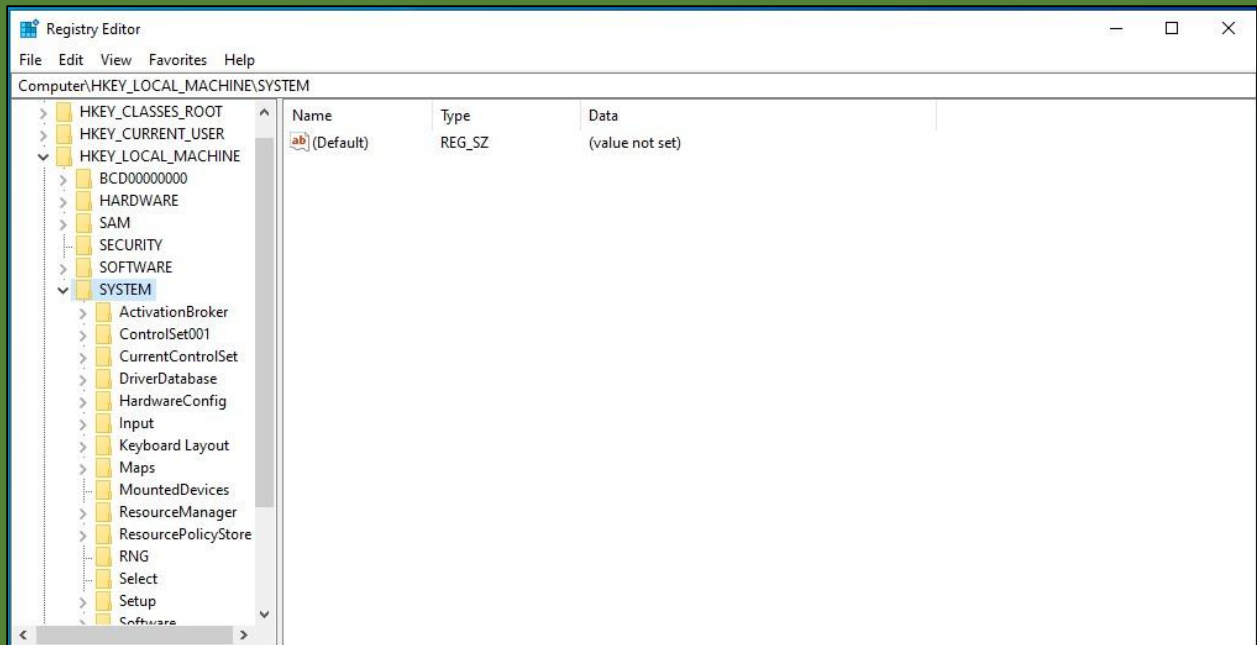


Figure 4: Registry Editor File

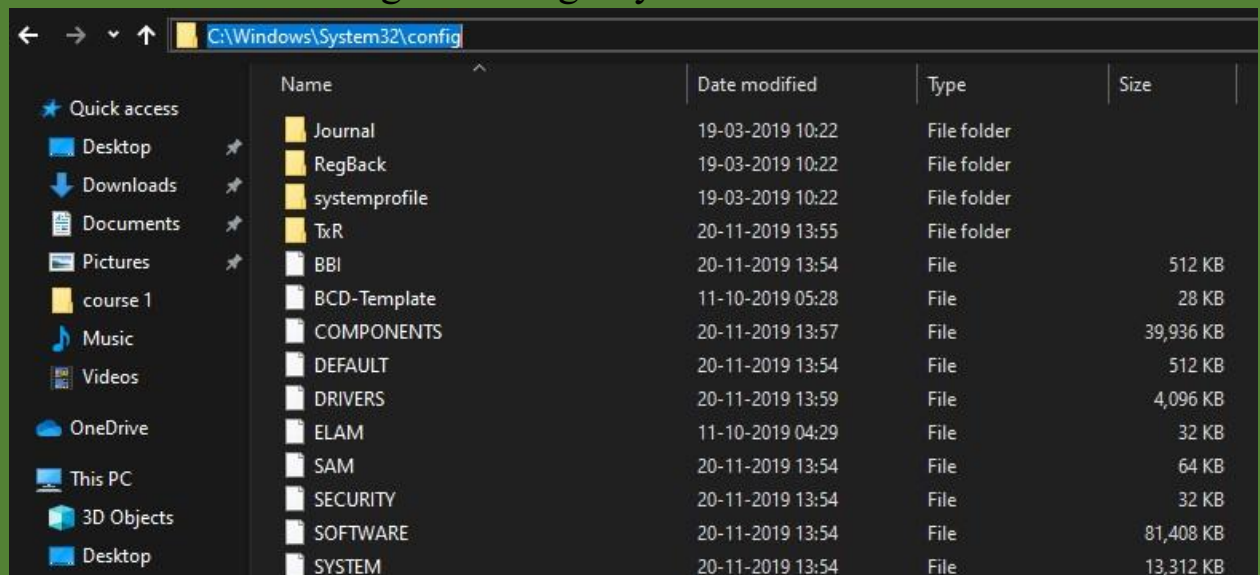


Figure 5: SAM and SYSTEM file

Step 3: These SAM and SYSTEM files can be accessed by registry editor after giving administrative permissions. Right click on the SAM file as shown in Figure 6. Then allow “Full Control” and “Read” by clicking the check box as shown in Figure 7.

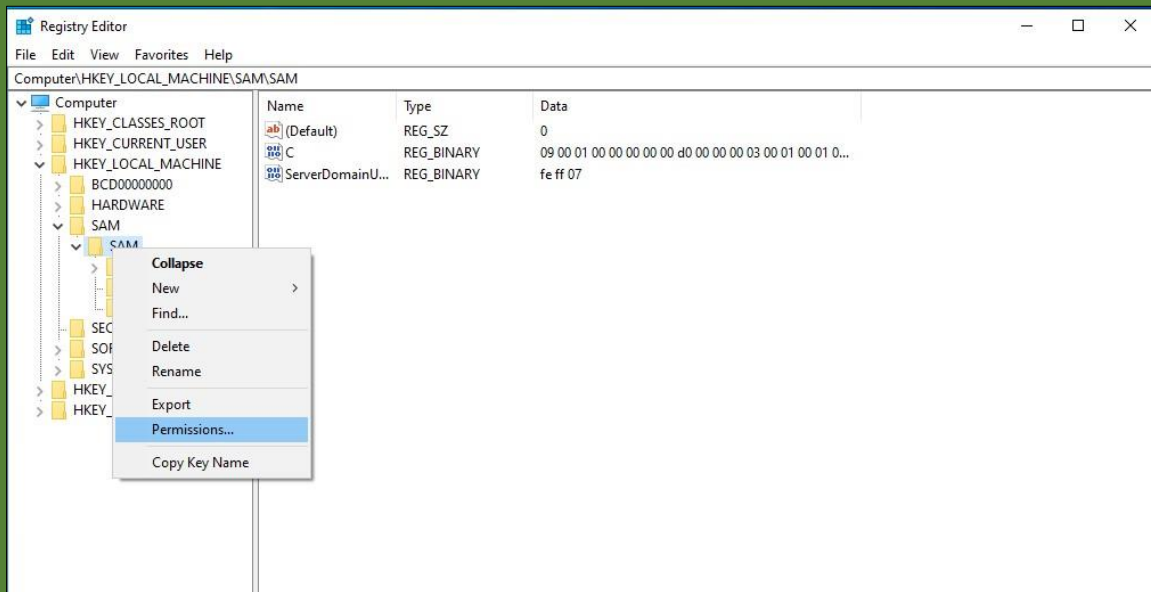


Figure 6: Checking permissions of SAM file

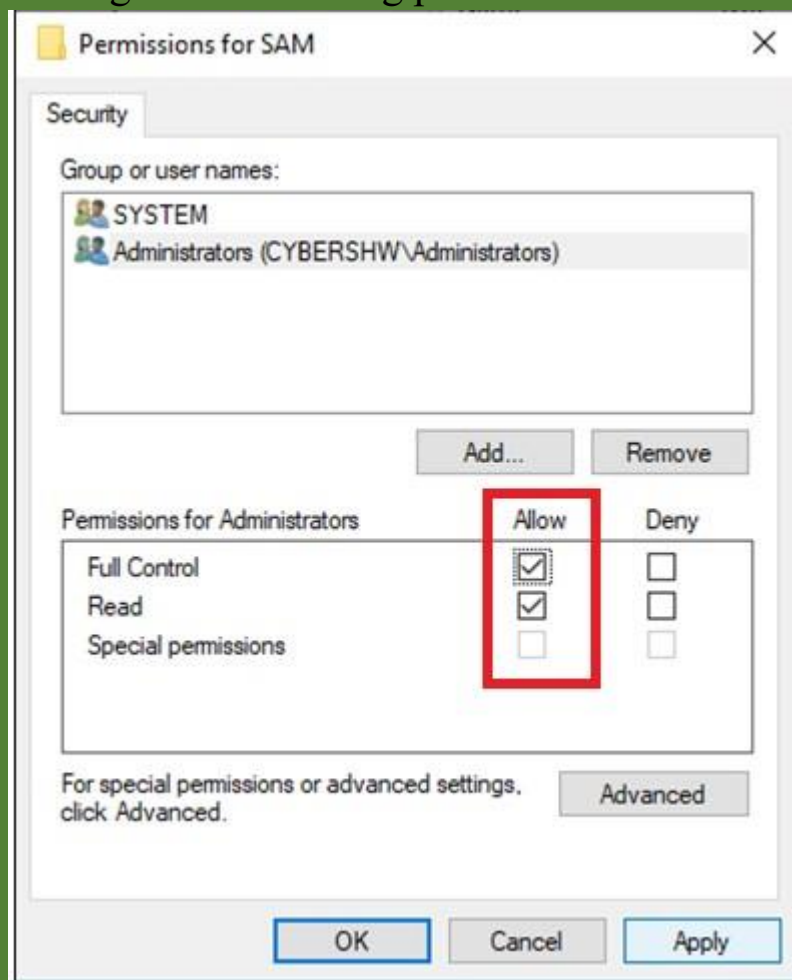


Figure 7: Giving permissions to the SAM file

Step 4: Export the SAM file after giving the administrative permissions. Right click on the SAM file and click “Export” as shown in Figure 8. Save the file by giving file name as “SAM” and type as “Registry Hive Files” as shown in Figure 9.

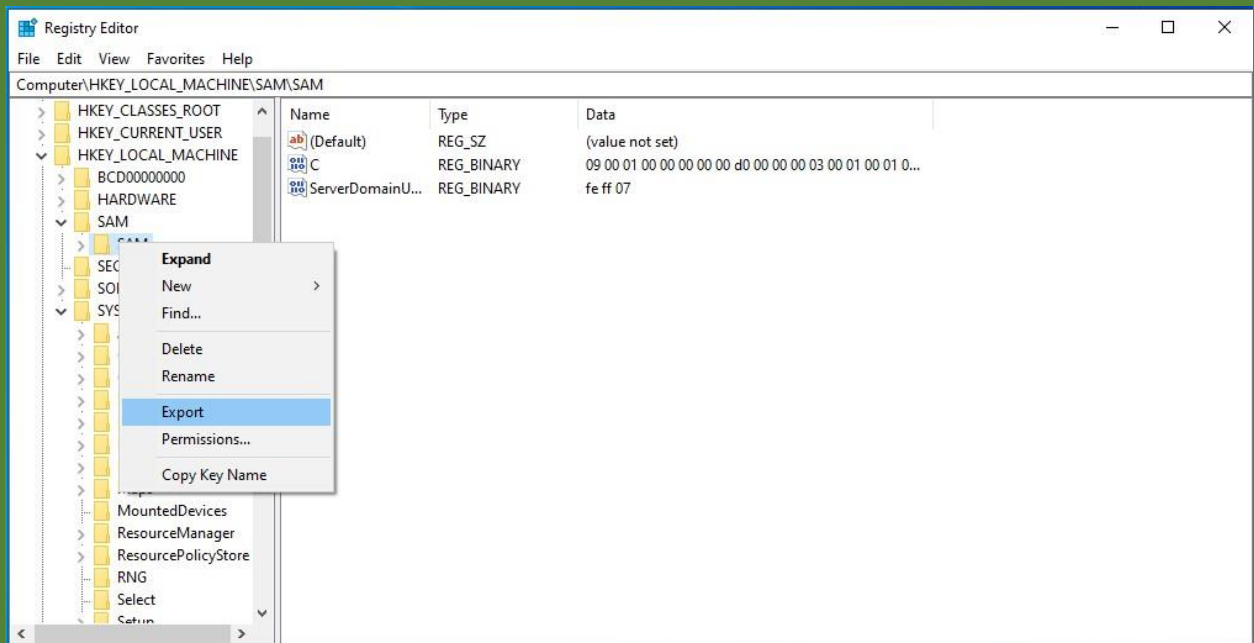


Figure 8: Exporting the SAM file

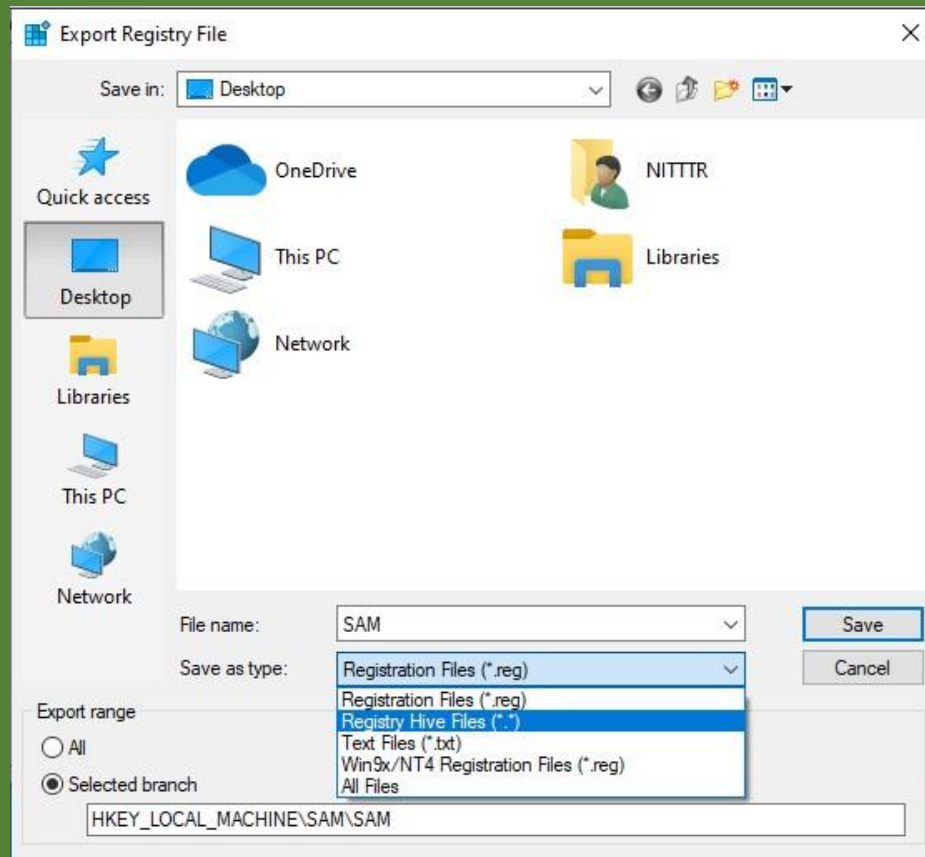


Figure 9: Saving the SAM file

5: In a similar fashion, right click on the SYSTEM file and give administrative permissions by allowing “Full Control” and “Read” after clicking the check box as shown in Figure 10.

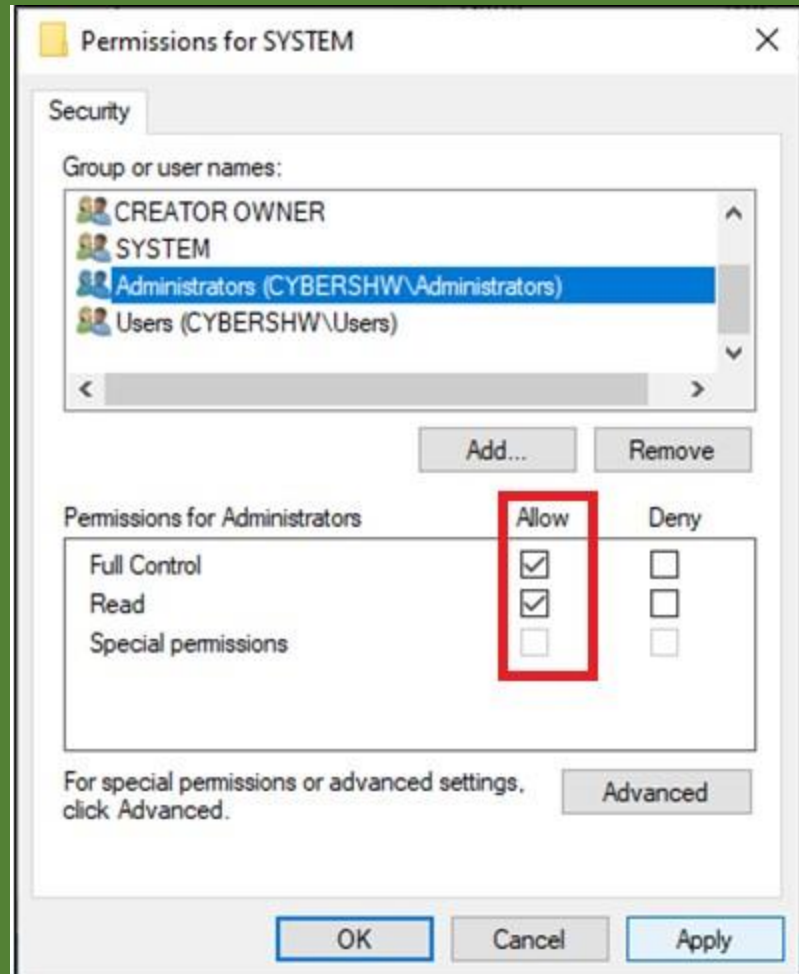


Figure 10: Giving permissions to the SYSTEM file

Step 6: Export the SYSTEM file after giving the administrative permissions. Right click on SYSTEM file and click “Export” as shown in Figure 11. Save the file by giving file name as “SYSTEM” and type as “Registry Hive Files” as shown in Figure 12.

Step

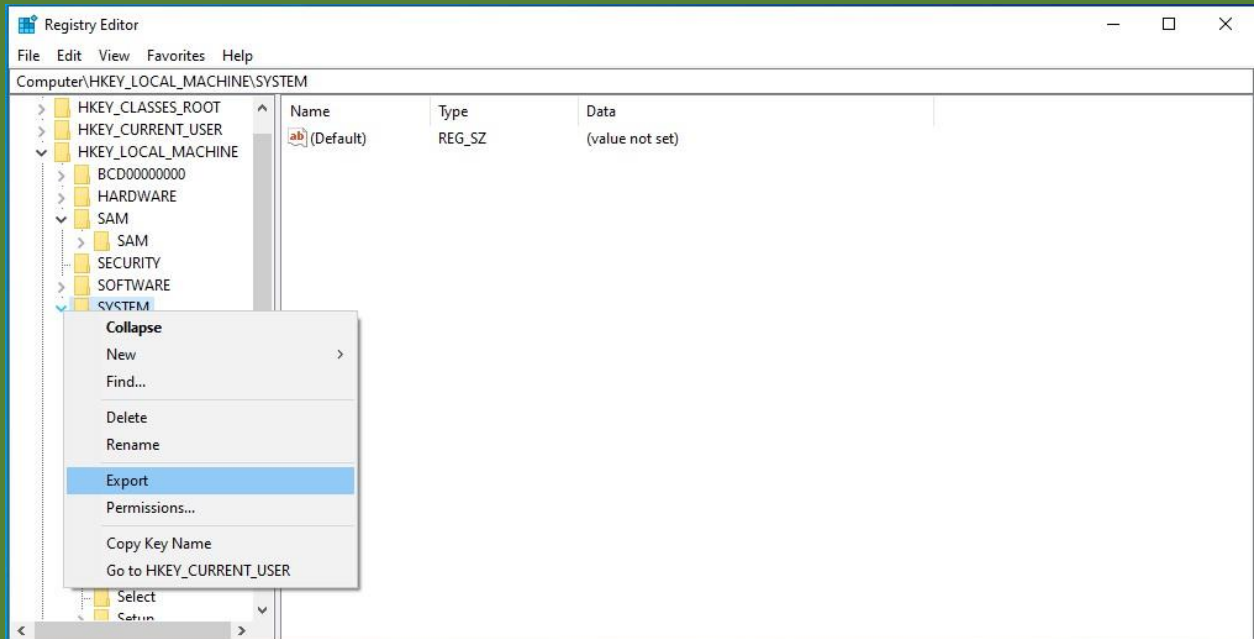


Figure 11: Exporting the SYSTEM file

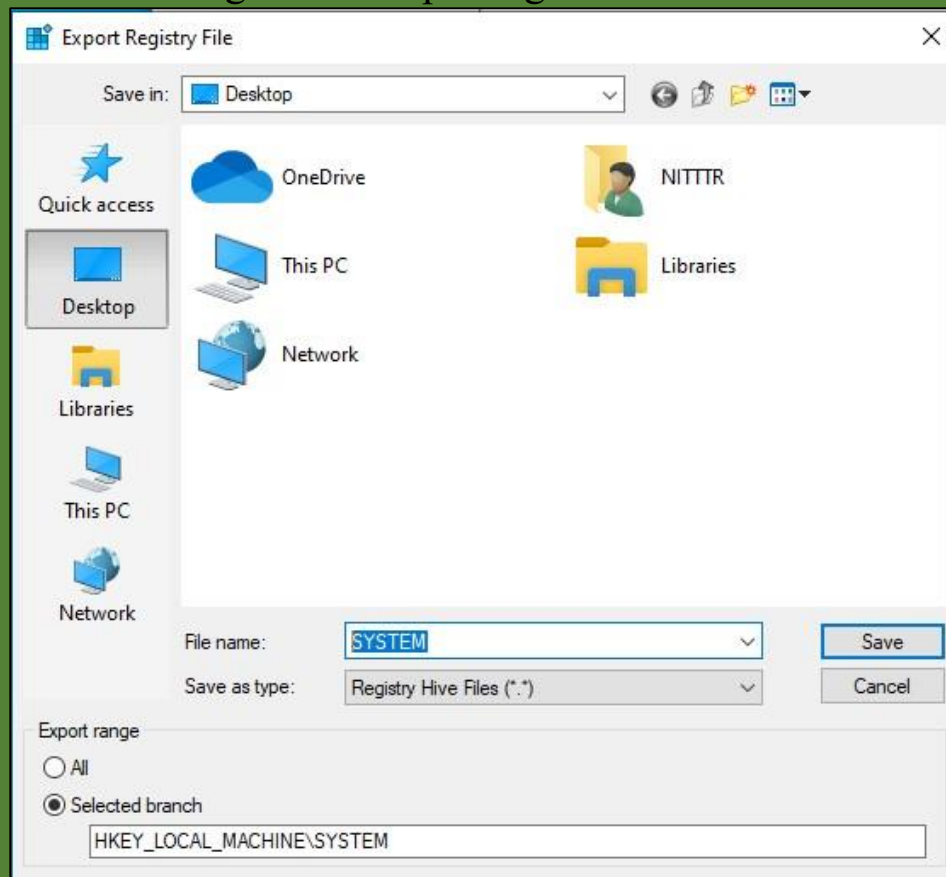


Figure 12: Saving the SYSTEM file

7: Download the “Mimikatz” tool by clicking the “mimikatz_trunk.zip” file from GitHub website as shown in Figure 13 and Figure 14.

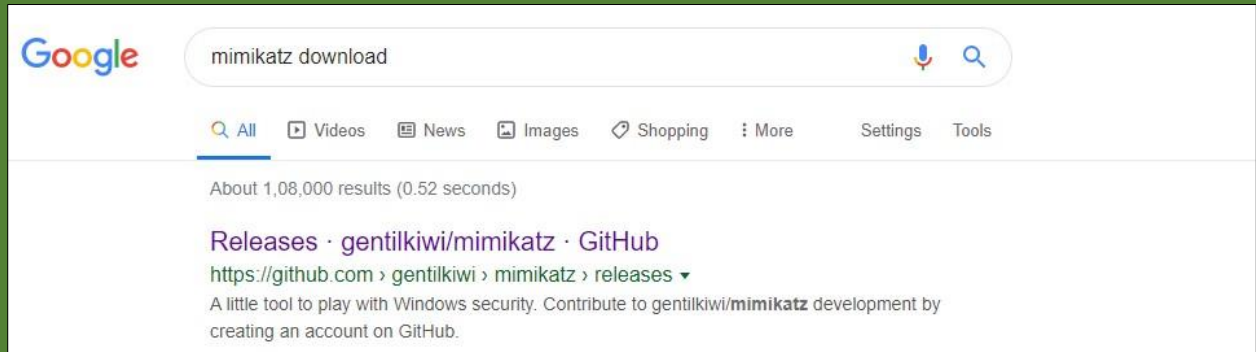


Figure 13: Search Mimikatz tool

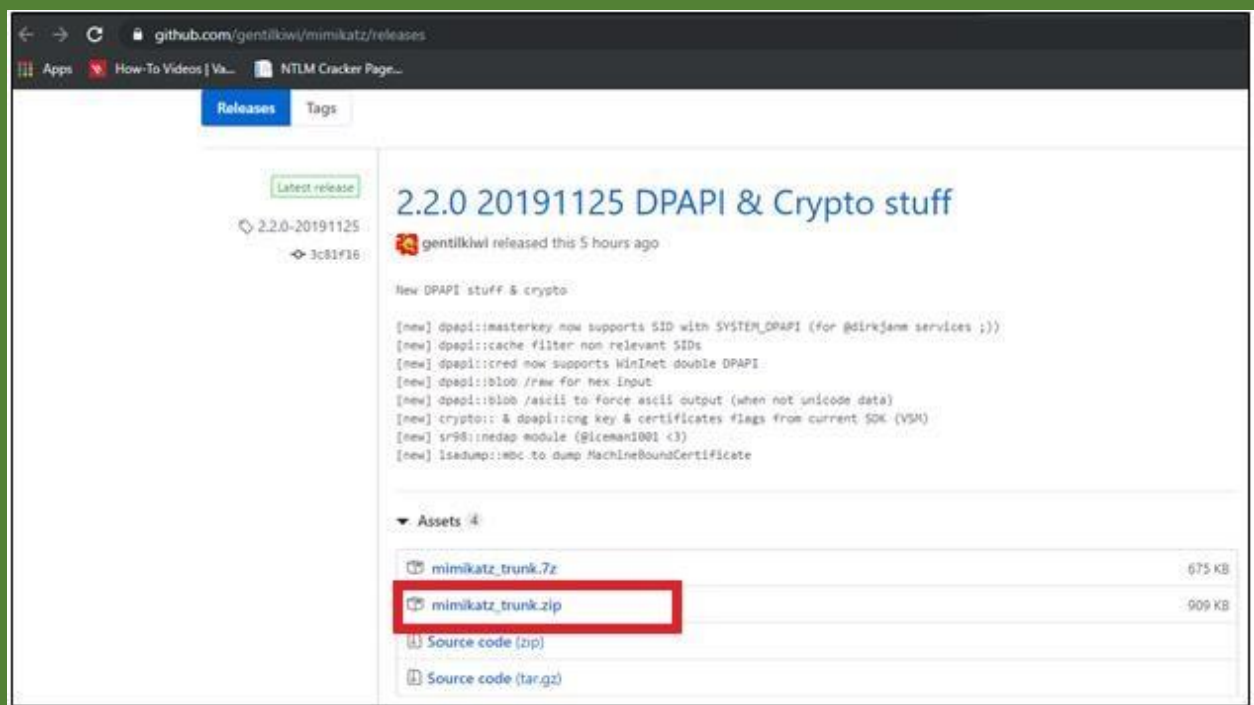


Figure 14: Download Mimikatz_trunk file from GitHub

Step

8: After downloading the file, unzip the “mimikatz_trunk.zip” file. Now go to: “C:/Downloads/mimikatz_trunk/x64/mimikatz” and left click twice on mimikatz file as shown in Figure 15.

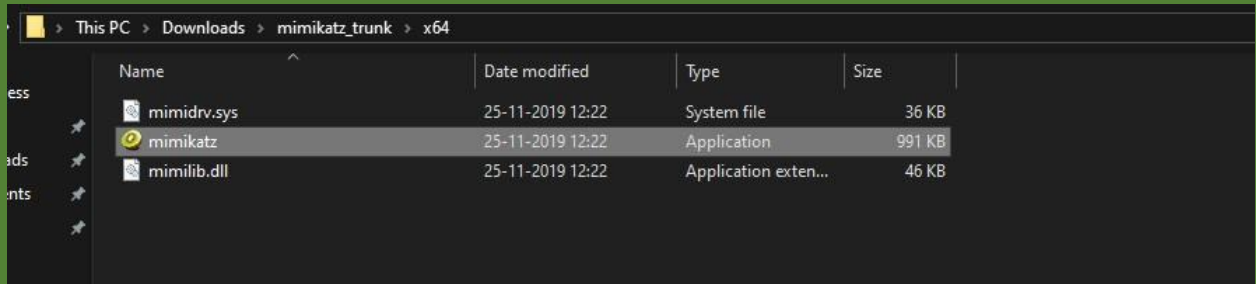


Figure 15: Downloaded file of mimikatz_trunk

Step 9: A prompt with a security warning will open as shown in Figure 16. Click on “Run” button to run the tool.

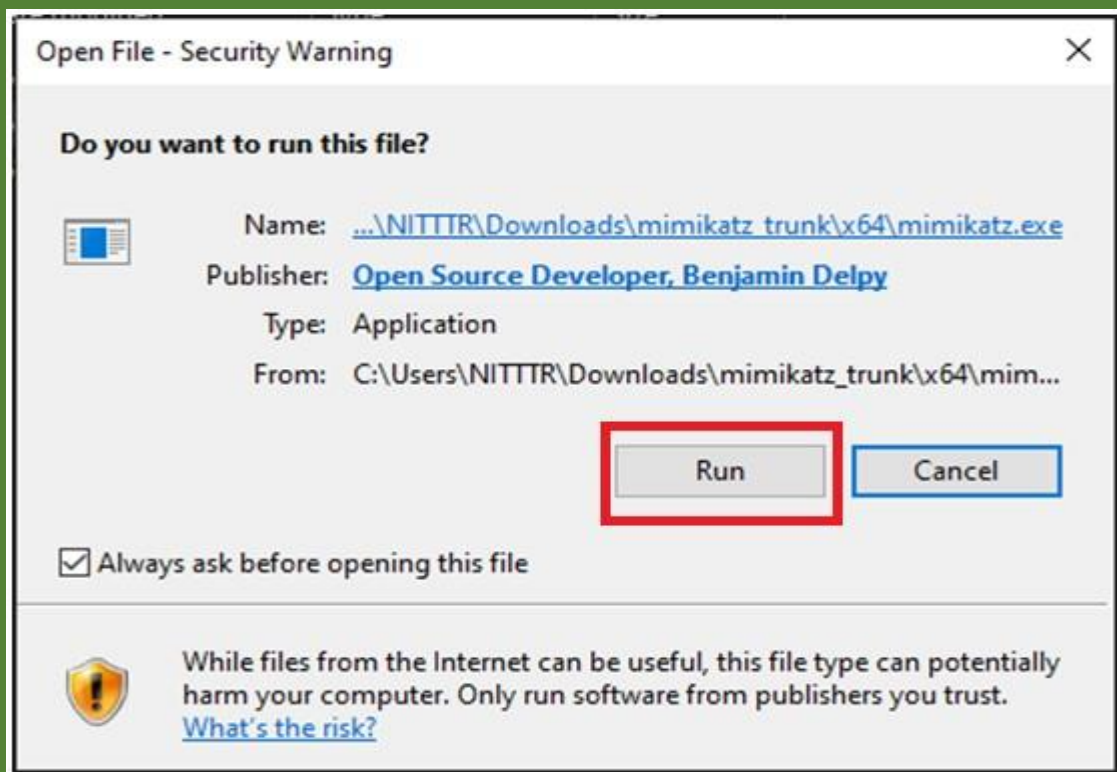
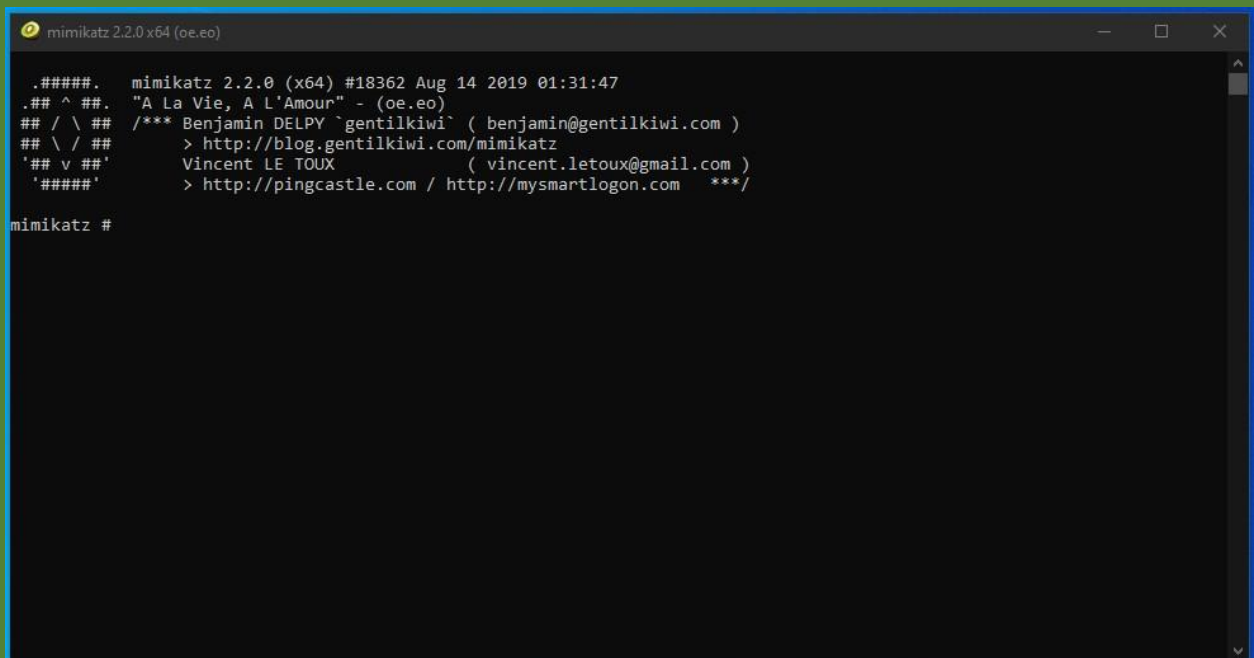


Figure 16: Click “Run” button

10: A command line prompt of Mimikatz tool will open as shown in Figure 17.



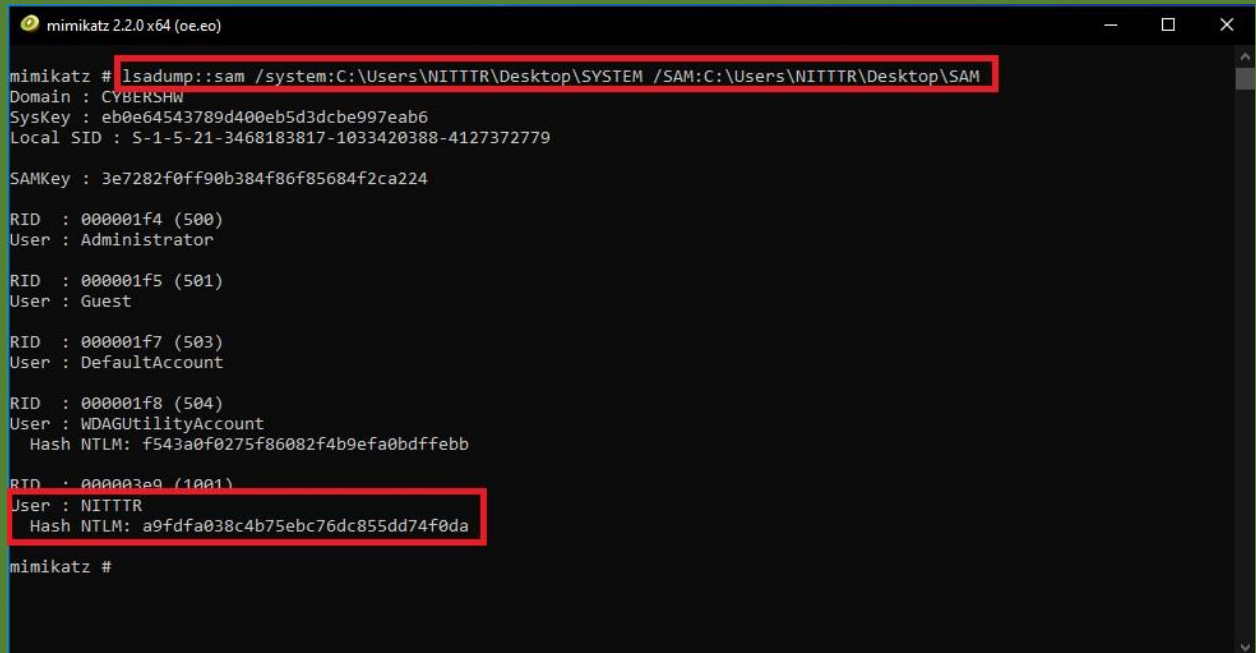
```
mimikatz 2.2.0 (x64) #18362 Aug 14 2019 01:31:47
.#####. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi' ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz #
```

Figure 17: Mimikatz command line

Step 11: Type “*lsadump::sam /system:C:\Users\NITTTR\Desktop\SYSTEM /SAM:C:\Users\NITTTR\Desktop\SAM*” command in command line prompt of Mimikatz tool. Press Enter. The command will show NTLM hash password of Windows operating system as shown in Figure 18.

Step



```
mimikatz 2.2.0 x64 (oe.eo)
mimikatz # lsadump::sam /system:C:\Users\NITTTT\Desktop\SYSTEM /SAM:C:\Users\NITTTT\Desktop\SAM
Domain : CYBERSHW
SysKey : eb0e64543789d400eb5d3dcbe997eab6
Local SID : S-1-5-21-3468183817-1033420388-4127372779

SAMKey : 3e7282f0ff90b384f86f85684f2ca224

RID : 000001f4 (500)
User : Administrator

RID : 000001f5 (501)
User : Guest

RID : 000001f7 (503)
User : DefaultAccount

RID : 000001f8 (504)
User : WDAGUtilityAccount
Hash NTLM: f543a0f0275f86082f4b9efa0bdf9ebb

RID : 000003e9 (1001)
User : NITTTT
Hash NTLM: a9fdfa038c4b75ebc76dc855dd74f0da

mimikatz #
```

Figure 18: Typing the command and getting NTLM hash

TOOL II: HASHCAT TOOL

- Hashcat tool [2] is an advanced password recovery tool.
- It had a proprietary code base until 2015, but now it is an open source tool and available in Kali Linux operating system.
- It recovers and provides plaintext of various hash such as LM, NTLM, MD5, SHA, and so on.

RECOVERING PLAINTEXT FROM NTLM HASH WITH HASHCAT TOOL

The Mimikatz tool provides NTLM hash of Windows operating system password. The password in plaintext from NTLM hash can be recovered with hashcat tool with following steps:

Step 1: Open Kali Linux operating system as shown in Figure 19.

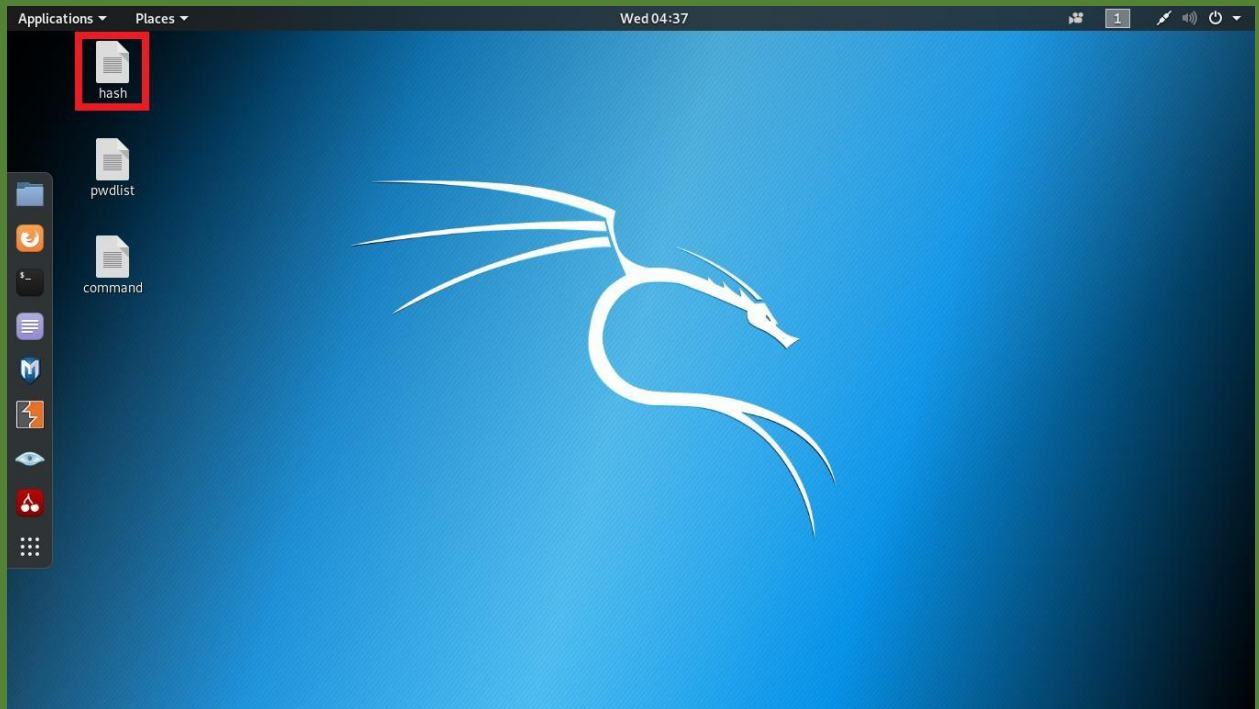


Figure 19: Kali Linux operating system

Step 2: Copy the NTLM hash (recovered with Mimikatz tool, refer Figure 18) and store it in a file on Desktop as shown in Figure 20. Also, multiple NTLM hash can be stored in a file to get plaintext as shown in Figure 21.

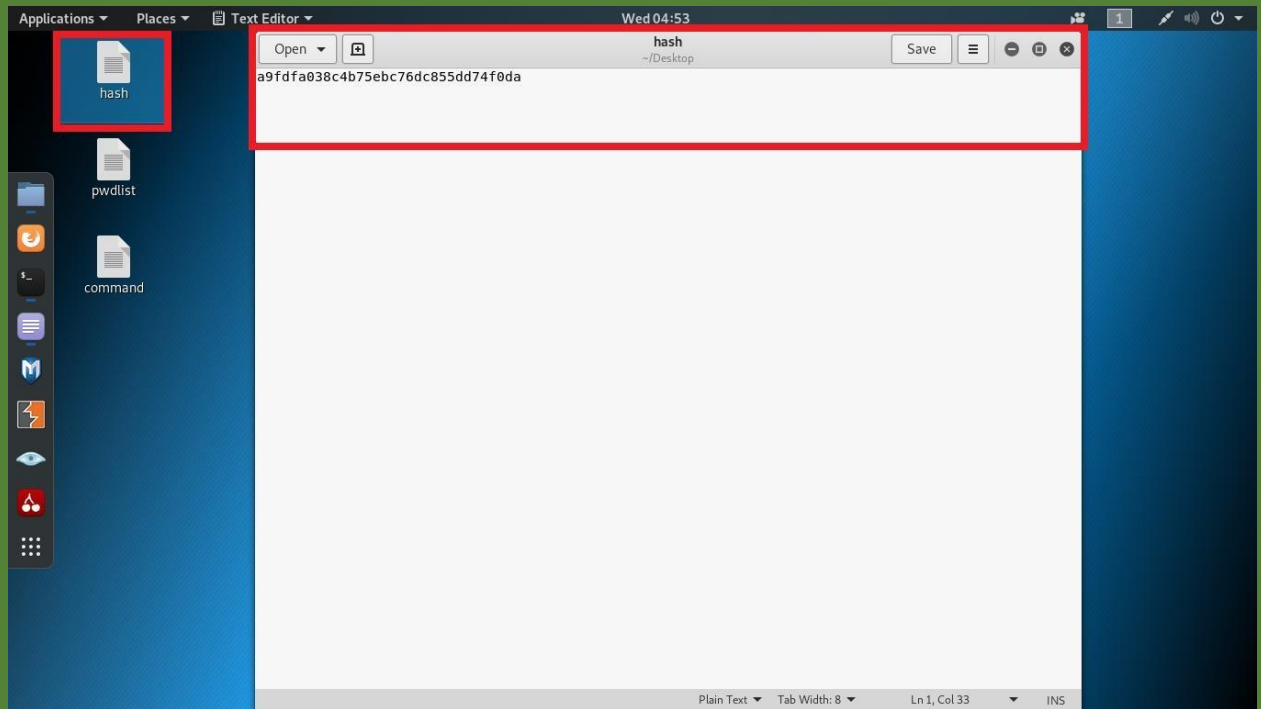


Figure 20: NTLM hash in a file

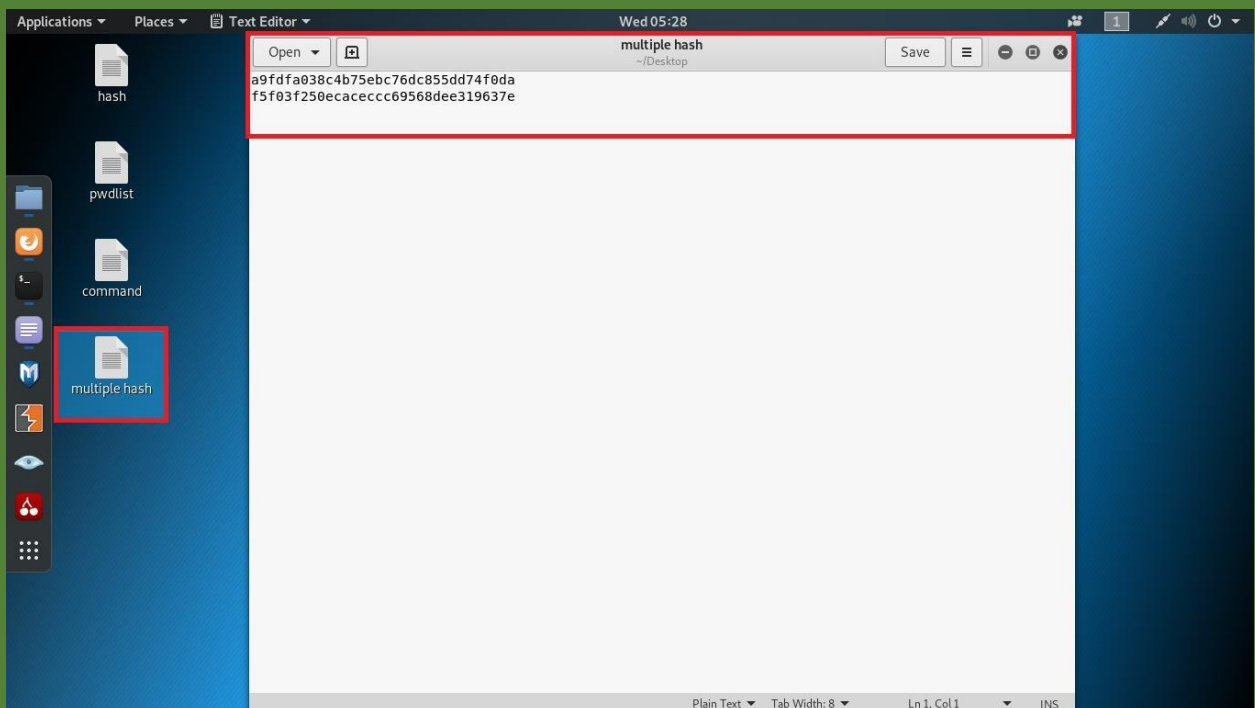


Figure 21: Multiple NTLM hash in a file

Step 3: Search the password wordlist by browsing Google search engine as shown in Figure 22. Open the GitHub website and download the ZIP file as shown in Figure 23.

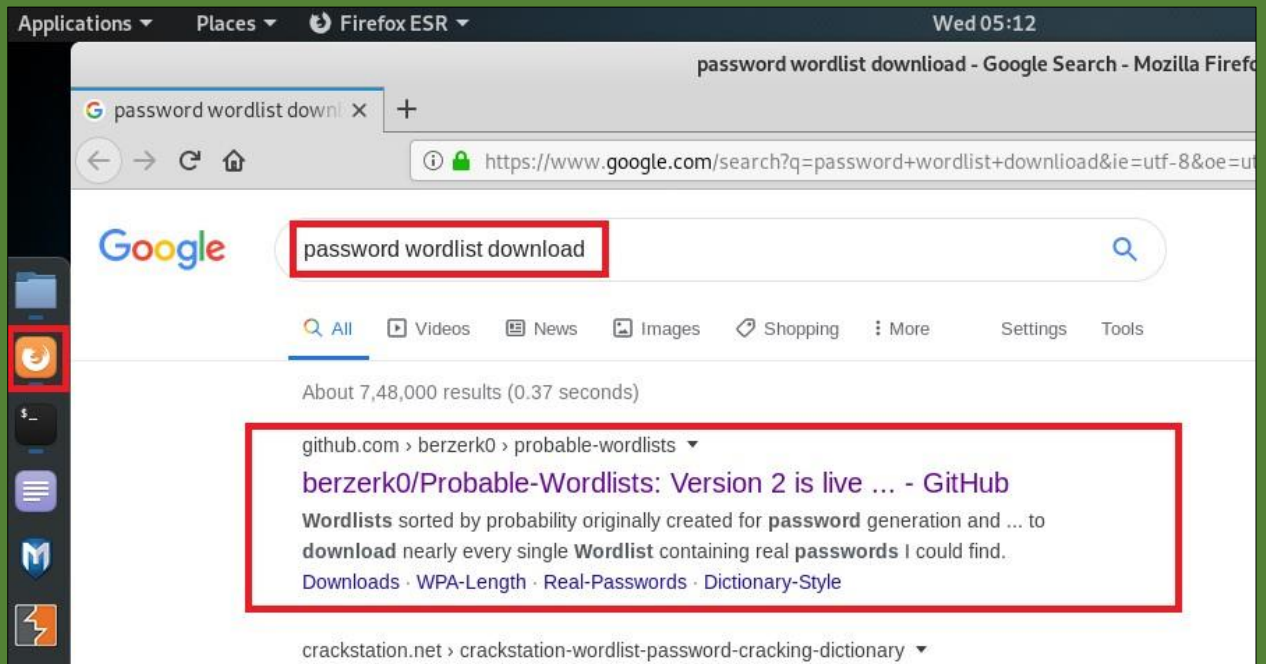


Figure 22: Search password wordlist



Figure 23: Download password wordlist

Step 4: Save and open the downloaded file as shown in Figure 24. Open the “Real-Passwords” folder to see the passwords wordlist as shown in Figure 25.

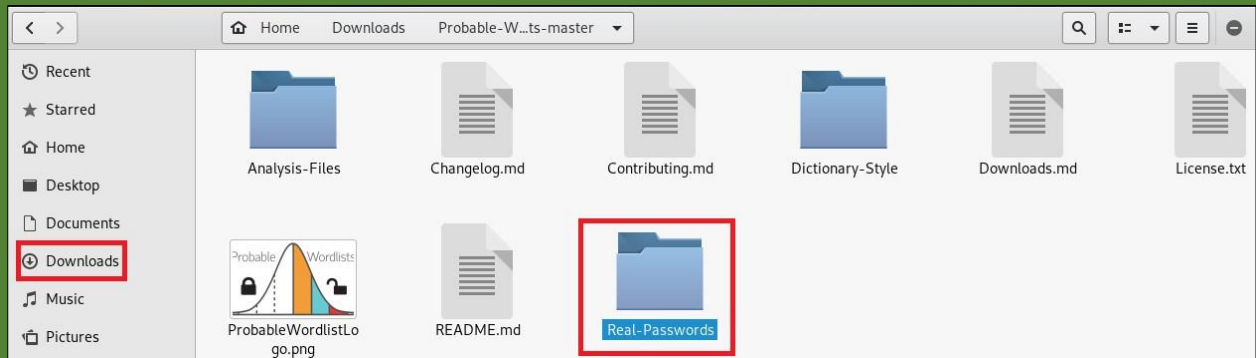


Figure 24: Password folder in downloaded file

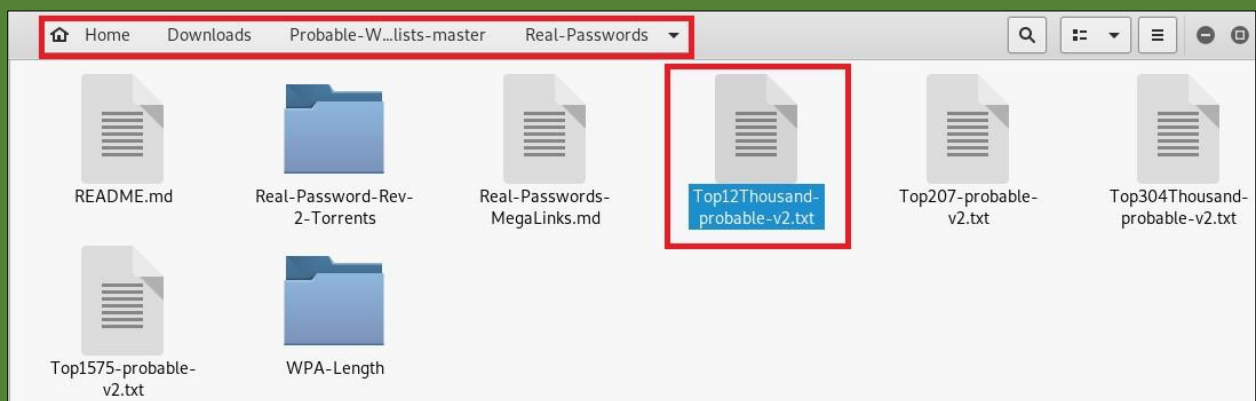


Figure 25: Password wordlist

Step 5: Open any password wordlist (e.g., Top12Thousandprobable-v2.txt file) as shown in Figure 26. Copy the wordlist file on Desktop and rename as “pwdlist” as shown in Figure 27.

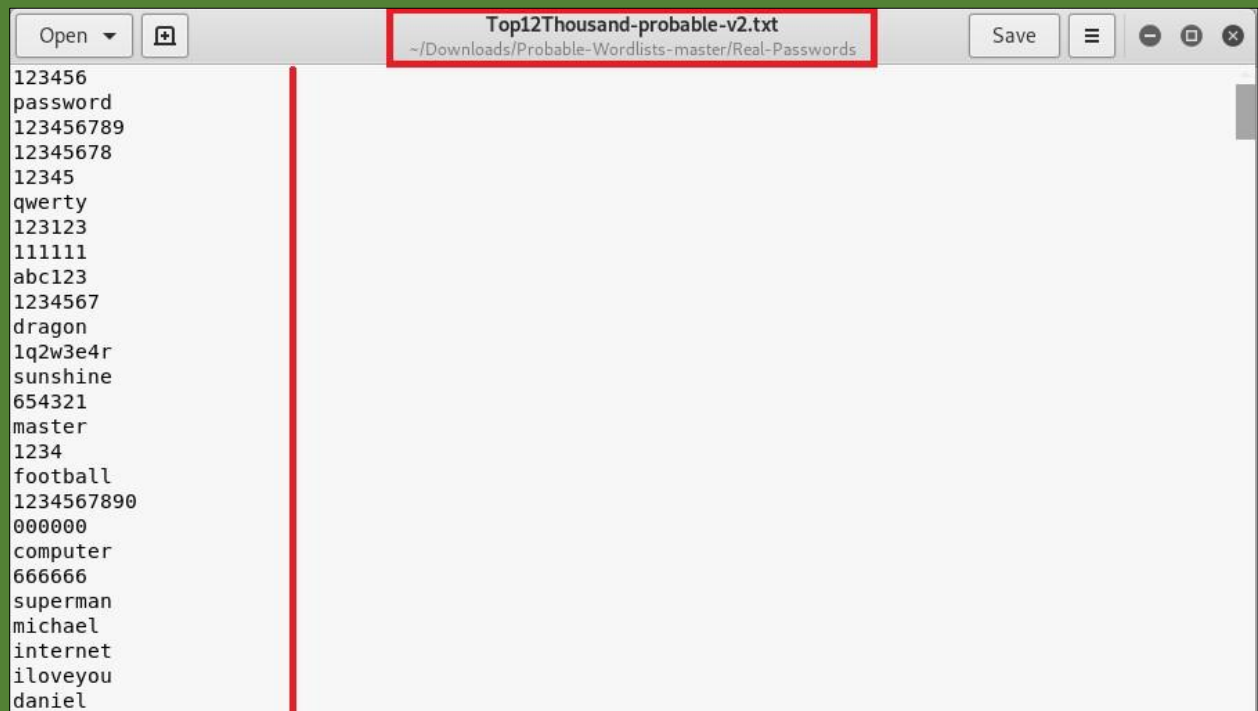


Figure 26: Top 12 thousand most frequently used passwords

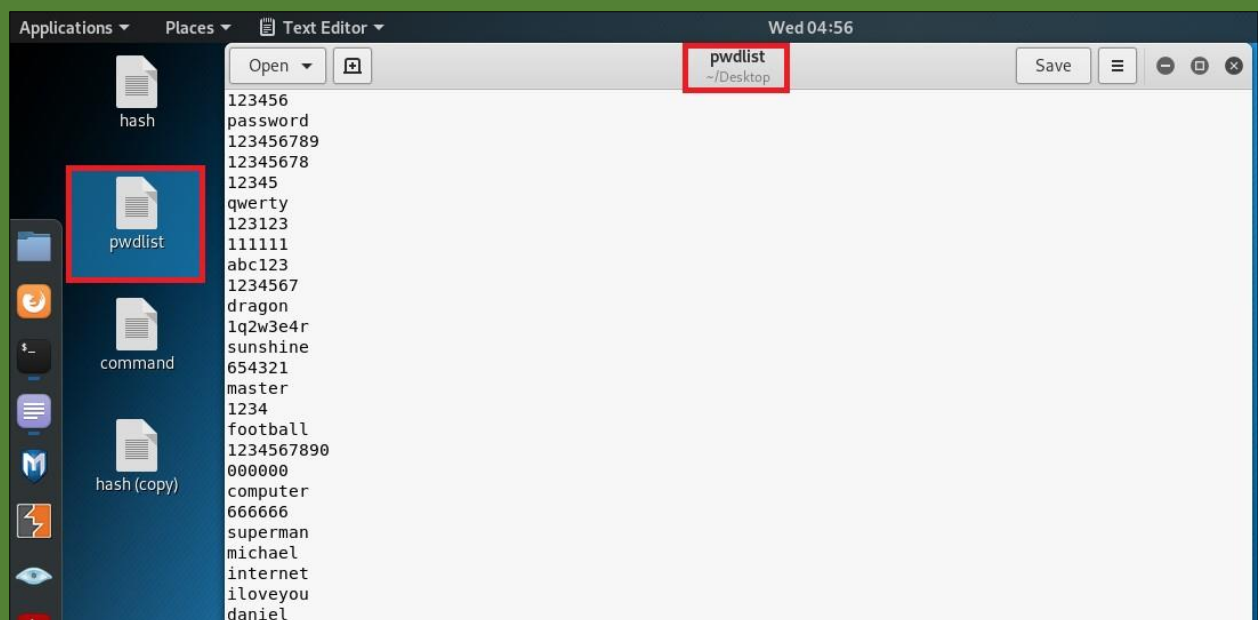


Figure 27: Copy the wordlist file on Desktop

Step 6: In Kali Linux operating system, open the hashcat tool. Go to Applications-> Password attacks-> hashcat as shown in Figure 28.

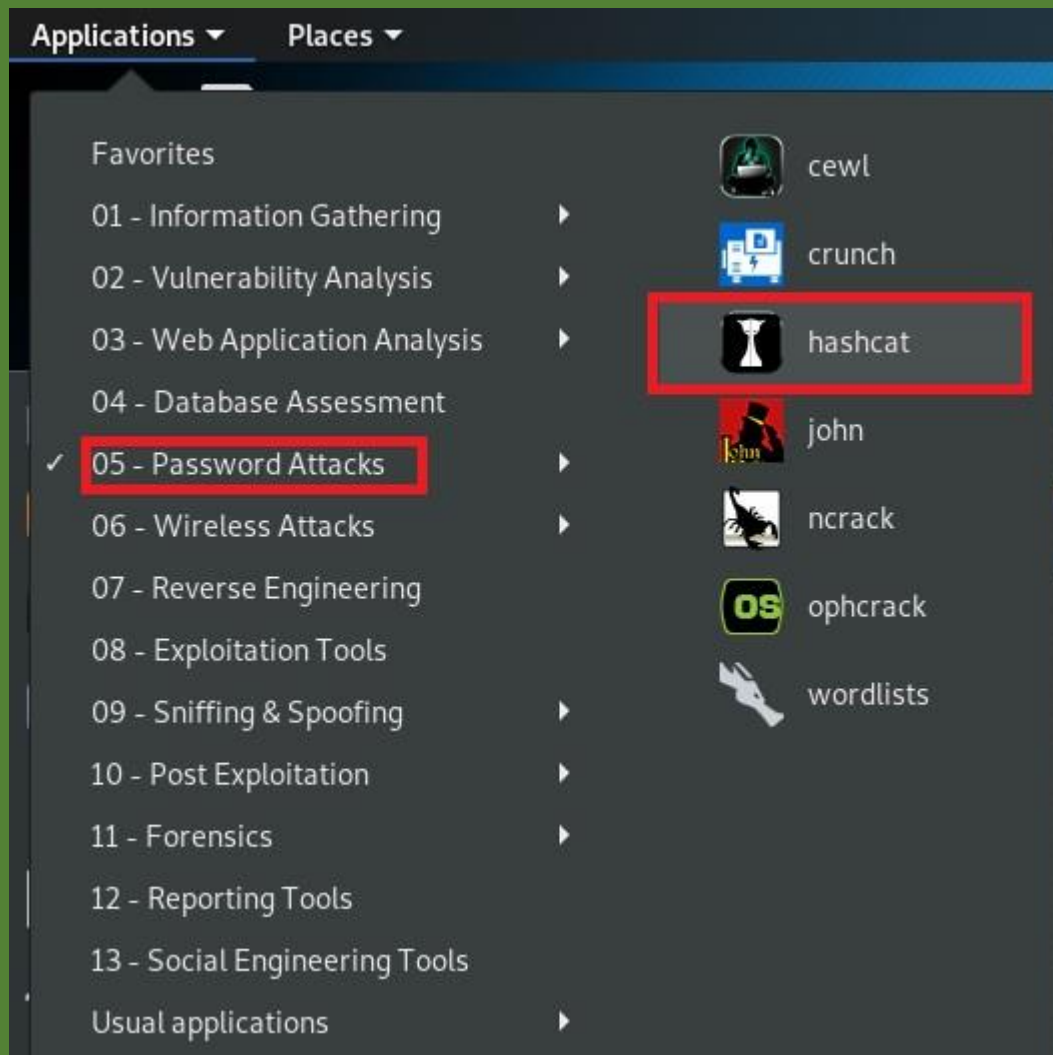


Figure 28: Opening hashcat tool

Step 7: A terminal with usage of hashcat tool will open as shown in Figure 29. The tool states various hash modes which can be recovered as shown in Figure 30 and Figure 31. The NTLM hash has ID of 1000 as shown in Figure 31. The tool also shows various attack modes as shown in Figure 32.

```
root@kali: ~
File Edit View Search Terminal Help
hashcat - advanced password recovery

Usage: hashcat [options]... hash[hashfile|hccapxfile [dictionary|mask|directory]
...

- [ Options ] -

Options Short / Long      | Type | Description
      | Example
=====+=====
-m, --hash-type           | Num  | Hash-type, see references below
      | -m 1000
-a, --attack-mode         | Num  | Attack-mode, see references below
      | -a 3
-V, --version             |      | Print version
-h, --help                |      | Print help
--quiet                   |      | Suppress output
--hex-charset             |      | Assume charset is given in hex
--hex-salt                |      | Assume salt is given in hex
```

Figure 29: Hashcat terminal

```
root@kali: ~
File Edit View Search Terminal Help
th commas | --brain-session-whitelist=0x2ae61ldb

- [ Hash modes ] -

# | Name | Category
=====+=====
900 | MD4 | Raw Hash
0 | MD5 | Raw Hash
5100 | Half MD5 | Raw Hash
100 | SHA1 | Raw Hash
1300 | SHA2-224 | Raw Hash
1400 | SHA2-256 | Raw Hash
10800 | SHA2-384 | Raw Hash
1700 | SHA2-512 | Raw Hash
17300 | SHA3-224 | Raw Hash
17400 | SHA3-256 | Raw Hash
17500 | SHA3-384 | Raw Hash
17600 | SHA3-512 | Raw Hash
17700 | Keccak-224 | Raw Hash
17800 | Keccak-256 | Raw Hash
17900 | Keccak-384 | Raw Hash
18000 | Keccak-512 | Raw Hash
600 | BLAKE2b-512 | Raw Hash
```

Figure 30: Hash modes

```

root@kali: ~
File Edit View Search Terminal Help
16400 | CRAM-MD5 Dovecot | HTTP, SMTP, LDAP Se
rver
15000 | FileZilla Server >= 0.9.55 | FTP Server
11500 | CRC32 | Checksums
3000 | LM | Operating Systems
1000 | NTLM | Operating Systems
1100 | Domain Cached Credentials (DCC), MS Cache | Operating Systems
2100 | Domain Cached Credentials 2 (DCC2), MS Cache 2 | Operating Systems
15300 | DPAPI masterkey file v1 | Operating Systems
15900 | DPAPI masterkey file v2 | Operating Systems
12800 | MS-AzureSync PBKDF2-HMAC-SHA256 | Operating Systems
1500 | descript, DES (Unix), Traditional DES | Operating Systems
12400 | BSDi Crypt, Extended DES | Operating Systems
500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5) | Operating Systems
3200 | bcrypt $2*$, Blowfish (Unix) | Operating Systems
7400 | sha256crypt $5$, SHA256 (Unix) | Operating Systems
1800 | sha512crypt $6$, SHA512 (Unix) | Operating Systems
122 | macOS v10.4, MacOS v10.5, MacOS v10.6 | Operating Systems
1722 | macOS v10.7 | Operating Systems
7100 | macOS v10.8+ (PBKDF2-SHA512) | Operating Systems
6300 | AIX {smd5} | Operating Systems
6700 | AIX {ssha1} | Operating Systems
6400 | AIX {ssha256} | Operating Systems
6500 | AIX {ssha512} | Operating Systems

```

Figure 31: Hash modes displaying NTLM hash

```

root@kali: ~
File Edit View Search Terminal Help
3 | Original-Word:Finding-Rule
4 | Original-Word:Finding-Rule:Processed-Word
- [ Attack Modes ] -
# | Mode
====+=====
0 | Straight
1 | Combination
3 | Brute-force
6 | Hybrid Wordlist + Mask
7 | Hybrid Mask + Wordlist
- [ Built-in Charsets ] -
? | Charset
====+=====
l | abcdefghijklmnopqrstuvwxyz
u | ABCDEFGHIJKLMNOPQRSTUVWXYZ
d | 0123456789
h | 0123456789abcdef
H | 0123456789ABCDEF
s | !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
a | ?l?u?d?s

```


Figure 32: Attack modes

Step 8: The basic examples regarding the usage of hashcat tool is shown in Figure 33.

```

root@kali: ~
File Edit View Search Terminal Help

=====
 1 | Low          | 2 ms   | Low          | Minimal
 2 | Default       | 12 ms  | Economic     | Noticeable
 3 | High          | 96 ms  | High         | Unresponsive
 4 | Nightmare     | 480 ms | Insane       | Headless

- [ Basic Examples ] -

Attack-      | Hash-      |
Mode         | Type       | Example command
=====
=====
Wordlist      | $P$       | hashcat -a 0 -m 400 example400.hash example.dict
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
Combinator    | MD5       | hashcat -a 1 -m 0 example0.hash example.dict example.dict

If you still have no idea what just happened, try the following pages:

* https://hashcat.net/wiki/#howtos_videos_papers_articles_etc_in_the_wild
* https://hashcat.net/faq/
root@kali:~#

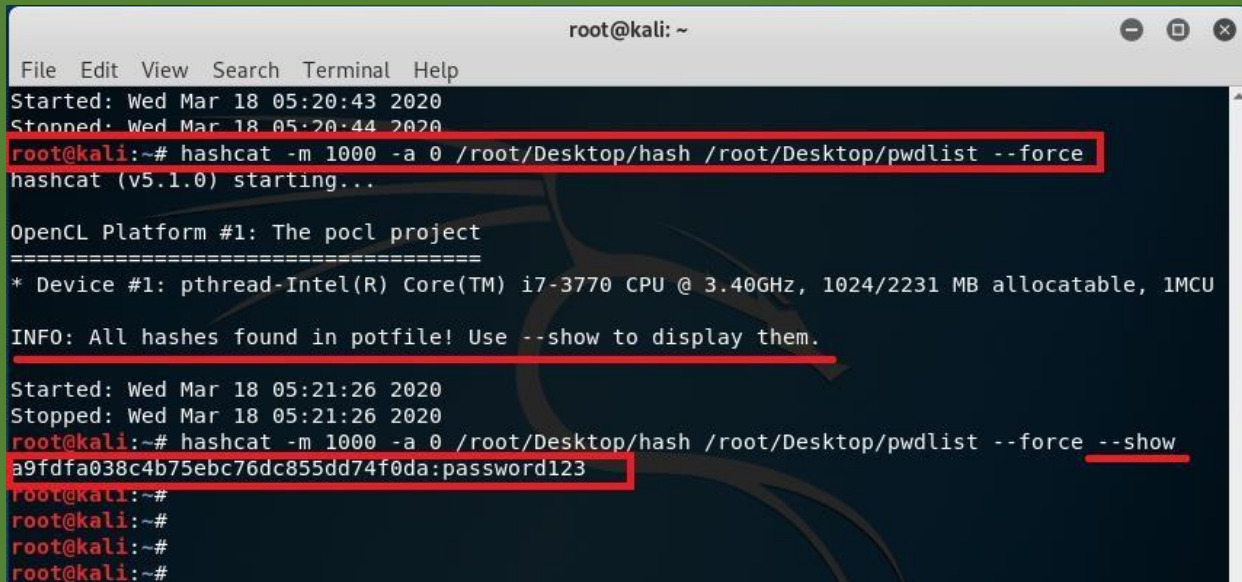
```

Figure 33: Example of hashcat

Step 9: Write the command “*hashcat -m 1000 -a 0 /root/Desktop/hash /root/Desktop/pwdlist --force*” to recover the hash and “*hashcat -m 1000 -a 0 /root/Desktop/hash /root/Desktop/pwdlist --force --show*” to display the plaintext of NTLM hash as shown in Figure 34.

In this command, -m stands for hash mode (e.g., 1000 stands for NTLM hash, refer Figure 31) and -a stands for attack mode (e.g., 0 stands for straight attack, refer Figure 32). The path to the hash file and wordlist file is also given in the command.

The plaintext of the NTLM hash is displayed in the Figure 34 and highlighted in red rectangular box. The plaintext of the NTLM hash is “password123”.



```
root@kali: ~  
File Edit View Search Terminal Help  
Started: Wed Mar 18 05:20:43 2020  
Stopped: Wed Mar 18 05:20:44 2020  
root@kali:~# hashcat -m 1000 -a 0 /root/Desktop/hash /root/Desktop/pwdlist --force  
hashcat (v5.1.0) starting...  
  
OpenCL Platform #1: The pocl project  
=====
```

* Device #1: pthread-Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, 1024/2231 MB allocatable, 1MCU

```
INFO: All hashes found in potfile! Use --show to display them.  
Started: Wed Mar 18 05:21:26 2020  
Stopped: Wed Mar 18 05:21:26 2020  
root@kali:~# hashcat -m 1000 -a 0 /root/Desktop/hash /root/Desktop/pwdlist --force --show  
a9fdfa038c4b75ebc76dc855dd74f0da:password123  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#
```

Figure 34: Recover plaintext of NTLM hash

Step 10: Similarly, hashcat can recover plaintext of multiple hash file. Write the command “*hashcat -m 1000 -a 0 /root/Desktop/multiplehash /root/Desktop/pwdlist --force*” to recover the hash and “*hashcat -m 1000 -a 0 /root/Desktop/multiplehash /root/Desktop/pwdlist --force -show*” to display the plaintext of multiple NTLM hash as shown in Figure 35.

The plaintext of the multiple NTLM hash is displayed in the Figure 35 and highlighted in red rectangular box. The plaintext of the NTLM hash is “shweta123” and “password123” respectively.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# hashcat -m 1000 -a 0 /root/Desktop/multiplehash /root/Desktop/pwdlist --force
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
=====
* Device #1: pthread-Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, 1024/2231 MB allocatable, 1MCU

INFO: All hashes found in potfile! Use --show to display them.

Started: Wed Mar 18 05:45:39 2020
Stopped: Wed Mar 18 05:45:39 2020
root@kali:~# hashcat -m 1000 -a 0 /root/Desktop/multiplehash /root/Desktop/pwdlist --force --show
f5f03f250ecaceccc69568dee319637e:shweta123
a9fdfa038c4b75ebc76dc855dd74f0da:password123
root@kali:~#
root@kali:~#
root@kali:~#
```

Figure 35: Recover plaintext of multiple NTLM hash

COUNTERMEASURES

The following countermeasures must be followed:

- **Strong Passwords:** Establish strong password using special characters, numbers, and lower and upper case alphabets. □ **Minimum Password Length:** The length of the password should be set to at least 14 characters. The long passwords are harder to crack than the short ones.
- **Dictionary words:** Do not use dictionary words such as password, qwerty, abc123, etc. These passwords can be cracked easily with tools.

Do not rely on similar looking characters such as: 3 □ E , 5 □ S , ! □ 1. These words are also stored in dictionary.

- **Minimum Password age:** The users must change the password after some time (30 days). This will reduce the risk of password cracking.
- **Stronger authentication method:** Use stronger authentication methods such as enable Gmail one time password feature to login in a new device.
- **Different passwords:** Use different passwords for different device or websites.
- **Sharing passwords:** Do not share passwords with anyone or change password immediately after usage, if shared.
- **Storing passwords:** Avoid storing passwords in an unsecured location such as desktop or mobile phones. An attacker can access those passwords by hacking the device. Try to remember the passwords.
- **Personal Information:** Do not use personal information such as date of birth, pet names, vehicle number, etc. An attacker can easily guess the password by knowing personal details through social engineering.

REFERENCES

- [1] B. DELPY, "github," [Online]. Available: <https://github.com/gentilkiwi/mimikatz/releases>. [Accessed 21 February 2020].
- [2] Atom. [Online]. Available: <https://tools.kali.org/password-attacks/hashcat>. [Accessed 27 March 2020].

