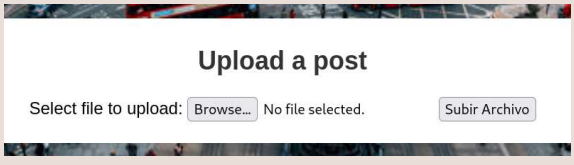
27. ✅Máquina: Picadilly(Fácil)

- *1. Descubrimiento de puertos y servicios con Nmap:*
  - Utilizamos Nmap para descubrir los puertos abiertos y los servicios en ejecución.
    - **Comando**: `nmap -sVC 172.17.0.2 -Pn`
    - **Resultado**: Se encontraron los servicios *HTTP Apache* puerto 80, servicio *(SSL) HTTPS Apache* puerto 443
- *2. Inspección servicio HTTP puerto 80:*
  - Navegando por el servicio HTTP puerto 80 observamos un archivo llamado *backup.txt* que contiene la siguiente información: `/// The users mateo password is //// ---------- hdvbfuadcb ---------- "To solve this riddle, think of an ancient Roman emperor and his simple method of shifting letters." ////////////////////////////////////`
- *3. Cifrado Backup.txt:*
  - Según el mensaje que nos dejan en el archivo *backup.txt*, podemos intuir que se trata de un *Caesar Cipher*, que consisite en mover las posiciones de las letras teniendo en cuenta el orden del abecedario. Usando la herramienta *https://www.dcode.fr/caesar-cipher*, podemos realizar una búsqueda de fuerza bruta de las posibles contraseñas utilizando diferentes valores de *shift* para desencriptarla. Vemos diferentes combinaciones interesantes, sobre todo la *easycrxazy*, que por similitud, con sentido, una buena candidata a contraseña sería *easycrazy*.
  - 

- *4. Inspección servicio HTTPS puerto 443:*
  - En el servicio HTTPS, puerto 443, nos encontramos lo que parece ser un blog. Tenemos la opción de subir archivos, así que podemos aprovecharnos de esto para subir una *reverse shell*.
  - 

- *5. Subida Reverse Shell servicio HTTPS:*
  - Ya que tenemos la opción de subir archivos, subiremos un archivo *.php* de una reverse shell y poniéndonos en escucha en el puerto 444. Crearemos la *Reverse Shell* utilizando la herramienta *Reverse Shell Generator*.
    - *Comando*: Creación Archivo .php Reverse Shell: `PHP Pentest Monkey indicando IP máquina atacante y puerto en escucha (444)` -> Escucha: `nc -lvnp 444`
    - *Resultado*: Recibimos una Shell remota de la máquina víctima como usuarios *www-data*, configurándola para su uso completo sin limitaciones, en la que también observamos el usuario *mateo* mencionado anteriormente en el *backup.txt*

```
www-data@063be3d43b53:/home/mateo$ export TERM=xterm
www-data@063be3d43b53:/home/mateo$ export SHELL=bash
www-data@063be3d43b53:/home/mateo$ whoami
www-data
```

- *6. Conexión Máquina Víctima usuario Mateo:*
  - Navegando por la máquina víctima siendo usuarios *www-data*, podemos observar el usuario existente de *mateo*, que también se mencionaba en el *backup.txt*, y el cual la contraseña estaba encriptada siguiendo el *Caesar Cipher*. Probaremos los diferentes candidatos que habíamos encontrado anteriormente para el texto cifrado: *hdvbfuadcb*.
    - *Comando*: `su mateo` -> contraseña: `candidatos Caesar Cipher Decoder Brute Force`
    - *Resultado*: Hemos tenido éxito con la contraseña *easycrazy* para el usuario *mateo*.

- *7. Verificación permisos usuario Mateo:*
  - Una vez dentro del usuario *mateo* en la máquina víctima con las credenciales encontradas de los candidatos del Caesar Cipher Decoder (contraseña *easycrazy*), vamos a verificar que permisos tiene este usuario sobre la máquina víctima.
    - **Comando**: `sudo -l`
    - **Resultado**: Mateo tiene máximos privilegios para ejecutar *php* sin necesidad de contraseña : `(ALL) NOPASSWD: /usr/bin/php`
- *8. Escalada de privilegios con php:*
  - Podemos aprovechar los máximos privilegios que tiene *mateo* con el binario *php* para realizar una escalada de privilegios. Buscamos en *GTFO Bins/php* el código *sudo* para realizar esta escalada y abrir una bash con privilegios máximos.
    - ***Comando***: `sudo /usr/bin/php -r "system('/bin/bash');"`
    - ***Resultado***: Aprovechando los máximos privilegios en el binario *php* para el usuario *mateo* hemos conseguido escalar con máximos privilegios *root* en la máquina víctima. Fin de la intrusión!

```
mateo@063be3d43b53:/var$ sudo /usr/bin/php -r "system('/bin/bash');"
root@063be3d43b53:/var# whoami
root
root@063be3d43b53:/var# cd /root/
root@063be3d43b53:~# ls -a
.
..
.bashrc
.profile
.ssh
root@063be3d43b53:~# sudo -l
Matching Defaults entries for root on 063be3d43b53:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local
r/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User root may run the following commands on 063be3d43b53:
    (ALL : ALL) ALL
root@063be3d43b53:~# xDaliK
bash: xDaliK: command not found
```