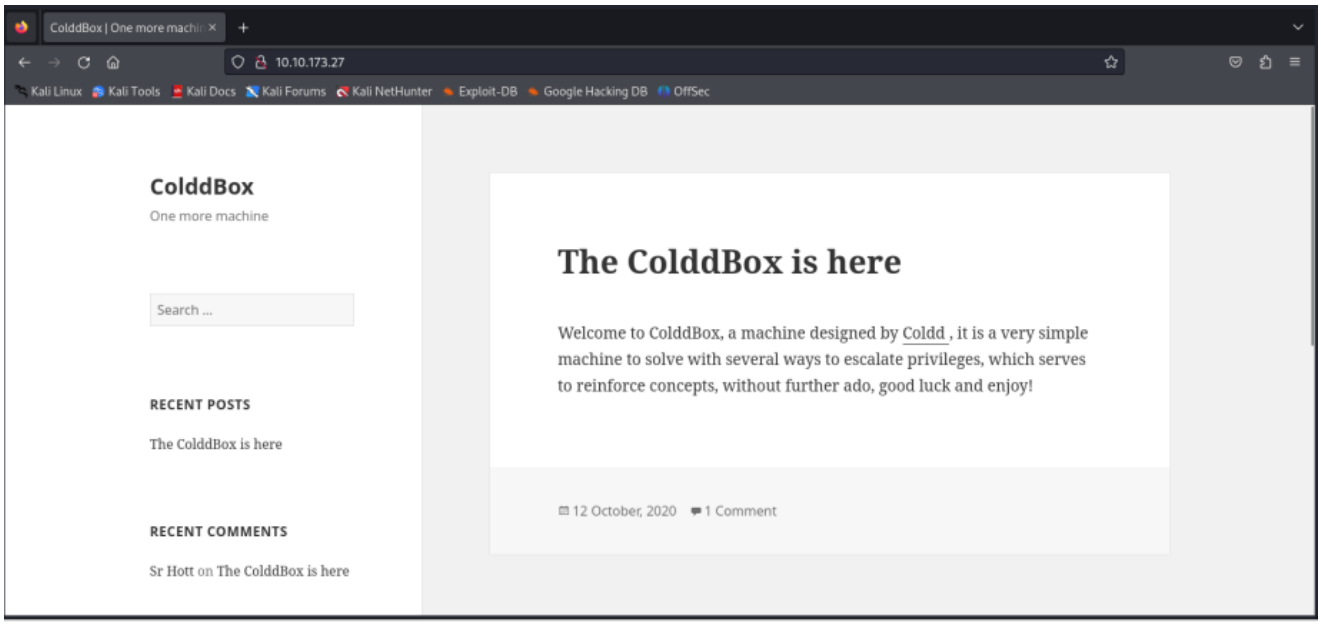
Maquina ColddBox - Tryhackme

En este primer escenario se ha elegido una maquina con sistema operativo Linux que contiene un servidor web con un CMS Wordpress ya que este es bastante usado en la actualidad y creo que sería útil incluirlo en el proyecto, ya que muchas empresa lo no lo suelen actualizar mucho y creo sería importante mostrar que podría pasar si no actualizas este CMS.

Como maquina voy a usar la de ColddBox que se encuentra en la pagina llamada Tryhackme, una maquina donde contiene diferentes maquinas preparas para ser explotadas sin corregir el riesgo de que pueda afectar a entornos reales, si no que son escenarios simulad

1. ## Enumeración y Reconocimiento
2. En primer lugar vamos a proceder a escanear todos los puertos de la maquina objetivo, ya que solo tenemos la dirección ip, pero previamente podemos insertar esa ip en el navegador para ver si existe algún sitio web.



Vemos que si que existe una página web, pero vamos a lanzar un escaneo con **nmap** para ver si tiene algún otro puerto.
Para ello utilizaremos el siguiente comando:

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p- --open --min-rate 8000 10.10.216.153 -Pn
```

Y nos mostraría el siguiente resultado:

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-11 12:27 EST
Nmap scan report for 10.10.216.153
Host is up (0.55s latency).
Not shown: 40825 closed tcp ports (conn-refused), 24708 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
4512/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

He usado un escaneo un poco más avanzado ya que un simple escaneo de nmap mostraba solo el puerto 80 como muestro en esta captura.

Por eso he usado los siguientes parámetros para hacer el escaneo un poco más avanzado

```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.216.153
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-11 12:23 EST
Nmap scan report for 10.10.216.153
Host is up (0.55s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
80/tcp open  http
```

- **-sV** → Esta opción habilita la detección de versiones de servicios en los puertos de nmap. Es decir, nmap intentara determinar qué servicio se está ejecutando en cada puerto abierto y que versión de software utiliza.
- **-p-** → esta opción indica que se deben escanear todos los puertos TCP. Es decir, que nmap intentara conectarse a cada puerto TCP en el rango de 1 a 65535.
- **--open** → esta opción indica que solo se deben mostrar los puertos que este abiertos, es decir que nmap solo nos mostrara los que respondan a las solicitudes de conexión.
- **--min-rate** → Esta opción establece la velocidad mínima de paquetes en 8000 paquetes por segundo. Pero esto es una opción no recomendad para entornos de producción ya que suele ser muy agresivo.
- **-Pn** → esta opción indica que se deben ignorar los hosts que no responden a las solicitudes de ping. Es decir, Nmap no intentará determinar si el host está activo antes de realizar el escaneo.

Ahora vamos a profundizar un poco y usando la opción *(-p)* para especificar los dos puertos y *(-sCV)* para
algo más de información más detallada acerca de los puertos que tiene abiertos esta máquina. Tras analizar la información devuelta por NMAP se confirma que los puertos TCP 80 y 4512 están abiertos y en ellos corren dos servicios cuyas versiones ya se conocen.

Con esta información obtenemos el siguiente resumen:

- **TCP 80** → Servicio HTTP → Aplicación Apache versión 2.4.18
- **TCP 4512** → Servicio SSH → Aplicación OpenSSH versión 2.10 Ubuntu, Protocolo 2.0

En primer lugar se procede a investigar el servicio HTTP tras el puerto TCP80 y se usaran las herramientas whatweb y nikto para obtener algo más información adicional:

HTTP (TCP80)

```
┌──(kali㉿kali)-[/shared/THM/colddbox]
└─$ whatweb http://10.10.252.91 | more
http://10.10.252.91 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.18 (
Ubuntu)], IP[10.10.252.91], JQuery[1.11.1], MetaGenerator[WordPress 4.1.31], PoweredBy[WordPress,WordPress,], Script[
text/javascript], Title[ColddBox | One more machine], WordPress[4.1.31], x-pingback[/xmlrpc.php]
```

```
┌──(kali㉿kali)-[/shared/THM/colddbox]
└─$ nikto -h http://10.10.252.91
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.10.252.91
+ Target Hostname:    10.10.252.91
+ Target Port:        80
+ Start Time:         2023-11-01 17:19:50 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
lnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /hidden/: This might be interesting.

+ /xmlrpc.php: xmlrpc.php was found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
^C
```

Gracias a estas dos herramientas podemos confirmar que realmente se trata de un servidor web Apache con la versión 2.4.18 y usando wordpress en la versión 4.1.31 con lo cual esto significa que esta versión de wordpress podría ser vulnerable ya que muchas versiones antiguas de este CMS puede tener vulnerabilidades.

Buscando la versión de wordpress en el navegador nos lleva a esta página del NIST, la cual nos lleva al siguiente CVE el 2020-4046 que si nos fijamos en la descripción afectaría a nuestra versión de wordpress. “Según **INCIBE**, en las versiones afectadas, los usuarios aunque carezcan de privilegios como colaboradores y autores puede usar el bloque incorporado de determinada manera para inyectar código HTML no filtrado en el editor de bloques. Cuando las publicaciones afectadas son vistas por un usuario con mayores privilegios, esto podría conllevar a una ejecución de script en el archivo editor/wp-admin"

CVE-2020-4046 Detail

Description

In affected versions of WordPress, users with low privileges (like contributors and authors) can use the embed block in a certain way to inject unfiltered HTML in the block editor. When affected posts are viewed by a higher privileged user, this could lead to script execution in the editor/wp-admin. This has been patched in version 5.4.2, along with all the previously affected versions via a minor release (5.3.4, 5.2.7, 5.1.6, 5.0.10, 4.9.15, 4.8.14, 4.7.18, 4.6.19, 4.5.22, 4.4.23, 4.3.24, 4.2.28, 4.1.31, 4.0.31, 3.9.32, 3.8.34, 3.7.34).

QUICK INFO

CVE Dictionary Entry:
CVE-2020-4046
NVD Published Date:
06/12/2020
NVD Last Modified:
01/27/2023
Source:
GitHub, Inc.

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

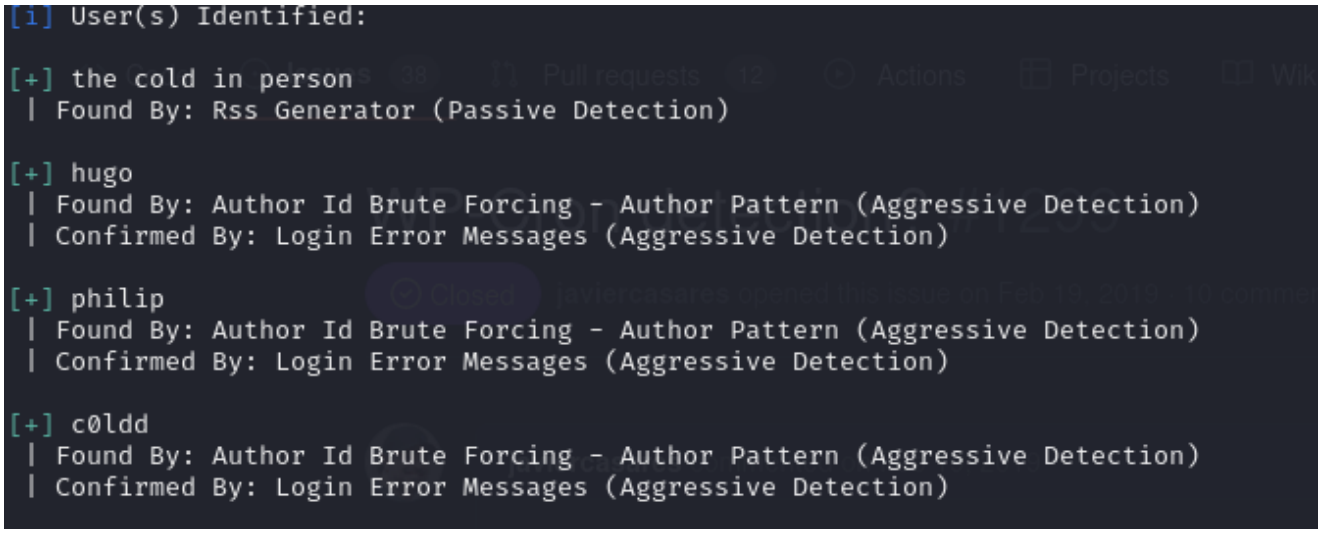
CNA: GitHub, Inc. **Base Score:** 5.4 MEDIUM **Vector:** CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N

Ahora tratándose de wordpress, esta CMS tiene una página llamada wp-admin, la cual contiene un login que nos llevara al panel de administración del sitio web el cual nos va a servir para explotar esa vulnerabilidad. Pero el primer paso sería saber que usuarios puede acceder al panel de administración.

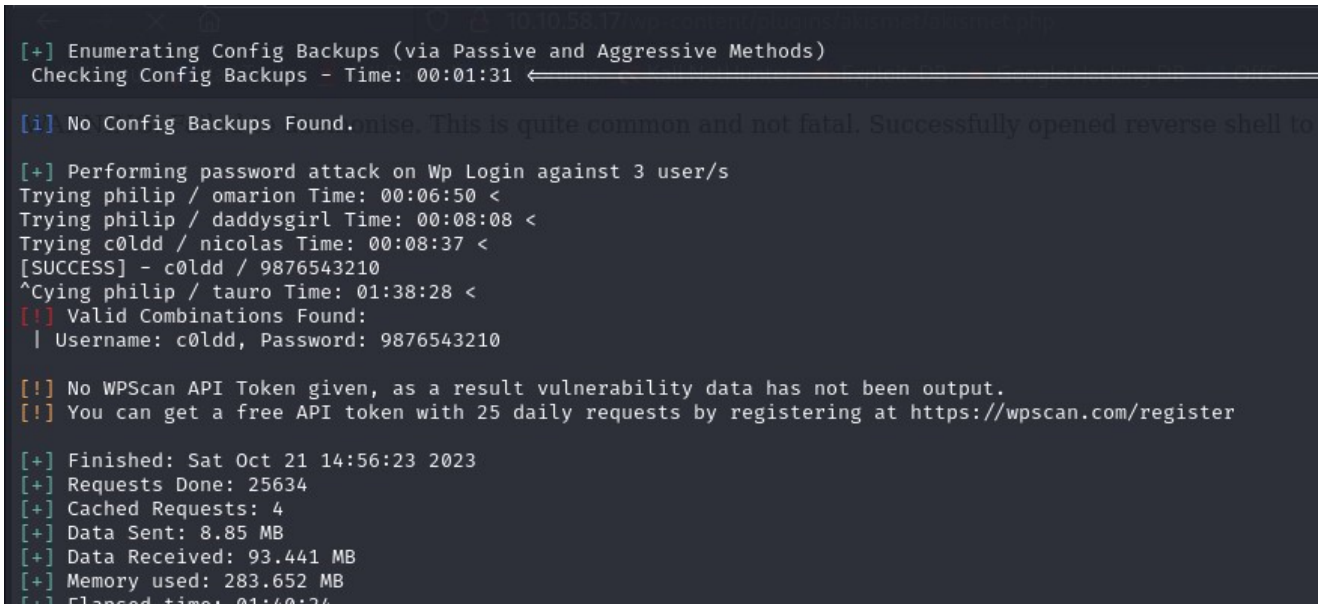
Para ello lanzaremos el siguiente comando,

```
Wpscan –url_ [_http://10.10.21.213_](http://10.10.21.213/) _–enumarate u_
```

Lo que hará este comando es con el parámetro –url nos cogerá la url del sitio web wordpress el –enumerate u nos enumerara solo los usuarios que encontré.



Vemos que nos muestra tres usuarios, de los cuales, nos creamos un pequeño diccionario con estos tres nombres y usando la misma herramienta de Wpscan junto con el diccionario por defecto rockyou intentaremos mediante fuerza bruta conseguir dicha contraseña para acceder al panel de control.



Como podemos ver nos saco la contraseña para el usuario c0ldd que seria 9876543210, entonces con esto datos ya podemos proceder a acceder y pasar a la parte de explotación.

SSH (TCP 4512)

En segundo lugar, se procede a investigar el servicio ssh tras el puerto TCP4512 y se comprueba que no se permite login sin contraseña o con la contraseña por defecto de root que sería toor.

```
┌──(kali㉿kali)-[/shared/THM/colddbox]
└─$ ssh root@10.10.64.31 -p 4512
The authenticity of host '[10.10.64.31]:4512 ([10.10.64.31]:4512)' can't be established.
ED25519 key fingerprint is SHA256:4Burx9DOSmBG9A0+DFqpM7rY4cyqpq59iluJwKx690c.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.64.31]:4512' (ED25519) to the list of known hosts.
root@10.10.64.31's password:
Permission denied, please try again.
root@10.10.64.31's password:
Permission denied, please try again.
root@10.10.64.31's password:
root@10.10.64.31: Permission denied (publickey,password).
```

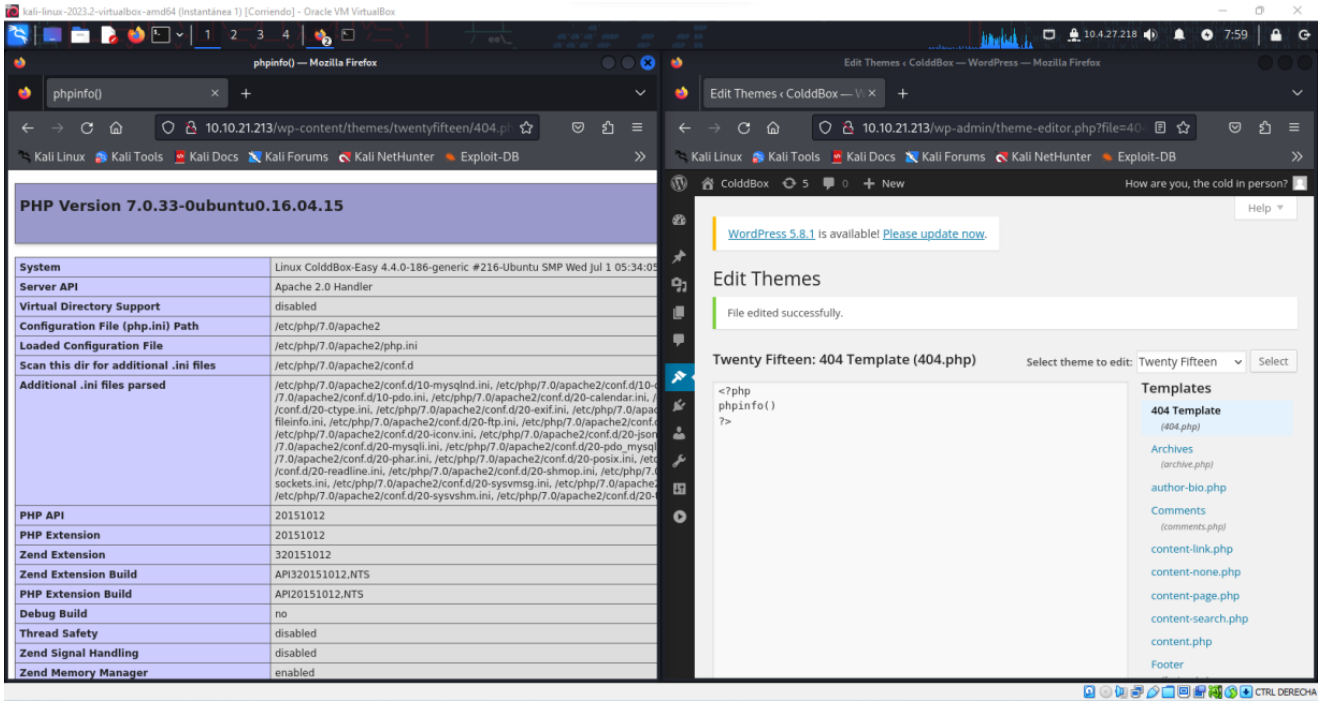
Análisis de Vulnerabilidades

En la siguiente fase como ya hemos conseguido, obtener el usuario y contraseña para poder acceder al panel de control y así poder explotar la vulnerabilidad del editor wordpress la cual nos permitirá obtener un Shell reverso insertando código php en el editor de wordpress usando la página 404.php ubicada en la sección themes > nombredeltema y editor.

Pero primero vamos a realizar una serie de pruebas insertando pequeños fragmentos php para comprar bien que podemos insertar código php.

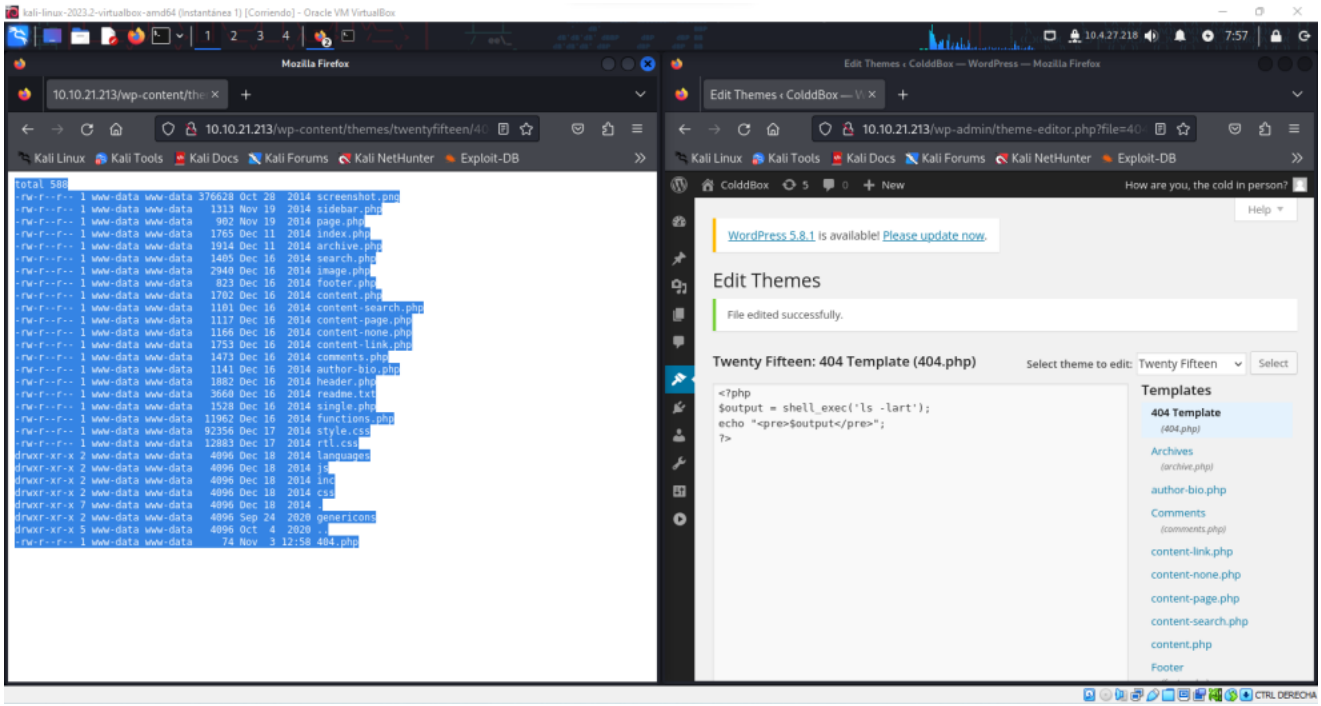
TEST 1

El primer test que realizaremos será el de insertar una pequeña función php llamada phpinfo () la cual nos va a mostrar información detallada sobre php como puede ser su versión y poco más.



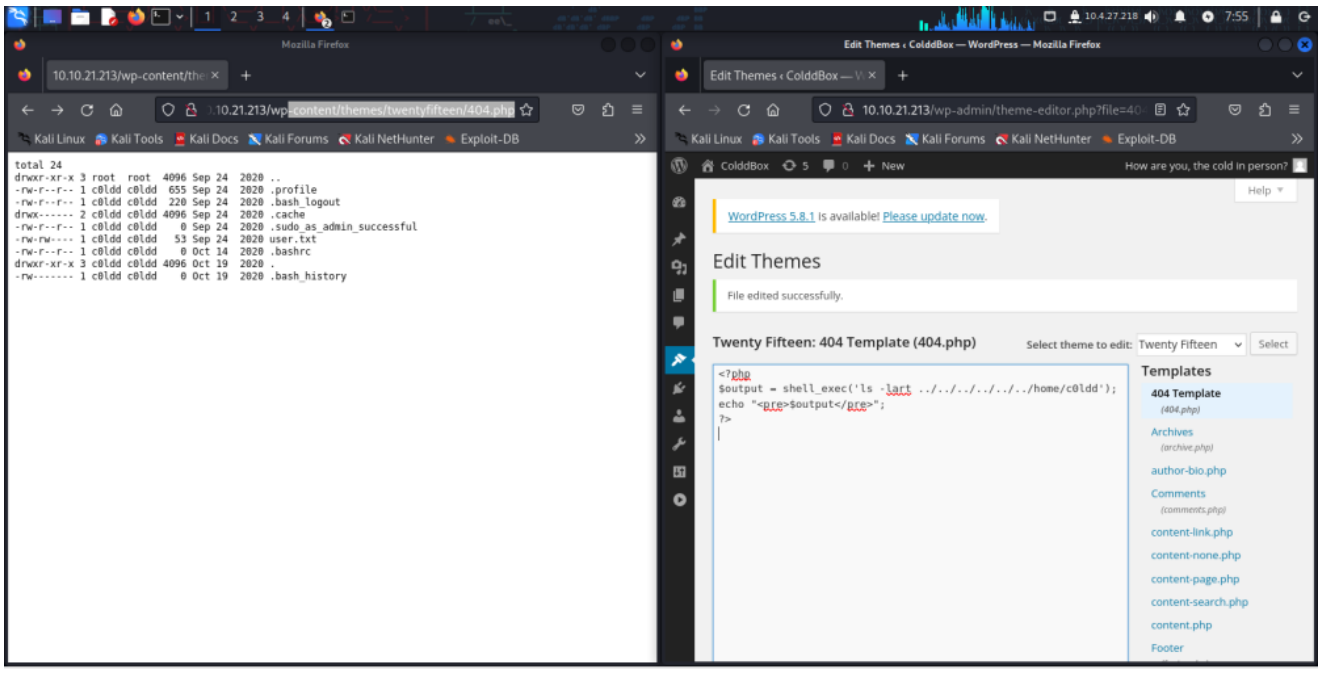
TEST 2

El segundo test costara de intentar lanzar un simple comando a través de una sentencia php, la cual nos va a servir para verificar en que directorio estamos y si realmente podemos usar comandos del servidor a través de wordpress.

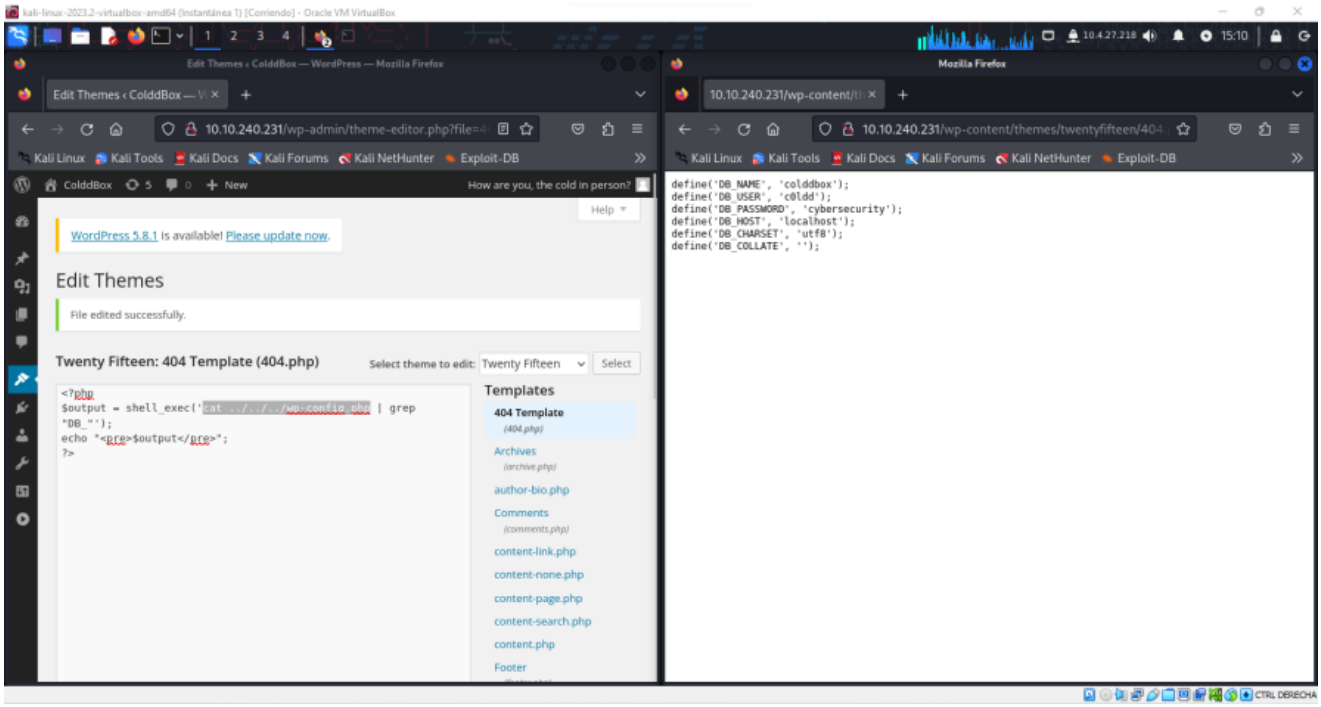


TEST 3

Debido a que podemos listar los directorios actuales del servidor y sabiendo ya en carpeta nos encontraremos usando la secuencia *'..'*. Que en Linux significa el directorio anterior entonces intentaremos listar si existe el user.txt y que permisos tiene.



Ahora ya sabemos que podemos usar comandos del sistema, así que como nosotros somos el www-data que es el usuario web en apache2, podemos intentar ir a la carpeta donde se encuentran los ficheros de configuración de wordpress e ir al fichero wp-config.php el cual es donde nosotros escribimos los datos de la base de wordpress y coger el usuario y contraseña y probar de conectarnos vía ssh.



Aquí podemos ver que mediante el siguiente comando *CAT ../../../wp-config.php,* el cual nos muestra por pantalla el fichero wp-config el cual vemos el nombre de usuario y contraseña

Usados en la para configurar la base de datos de wordpress, entonces lo que voy a hacer es insertar *el db_username y db_password* para acceder a la maquina vía ssh por el puerto 4512 que es el que encontramos en la fase de reconocimiento.

Aquí en esta otra captura voy a mostrar que he podido acceder vía ssh usando los credenciales encontrados en el fichero wp-config.php, el cual deberían pertenecer a la base de datos.

Y ahora ya sí que podríamos ir a user.txt y leer la flag del usuario y ya tendríamos una parte que es la de conseguir acceder con un usuario del sistema.

```
┌──(kali㉿kali)-[~]
└─$ ssh c0ldd@10.10.240.231 -p 4512
The authenticity of host '[10.10.240.231]:4512 ([10.10.240.231]:4512)' can't be established.
ED25519 key fingerprint is SHA256:4Burx9DOSmBG9A0+DFqpM7rY4cyqpq59iluJwKx690c.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.240.231]:4512' (ED25519) to the list of known hosts.
c0ldd@10.10.240.231's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-186-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


Pueden actualizarse 129 paquetes.
92 actualizaciones son de seguridad.


Last login: Mon Nov  8 13:20:08 2021 from 10.0.2.15
c0ldd@ColddBox-Easy:~$
```

```
c0ldd@ColddBox-Easy:~$ ls
user.txt
c0ldd@ColddBox-Easy:~$ cat user.txt
RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==
c0ldd@ColddBox-Easy:~$
```

Explotación

En este punto, ahora que ya hemos analizado la vulnerabilidad y obtenido acceso con un usuario del sistema, para poder acceder a la siguiente flag debemos acceder mediante la **escalada de privilegios** y para ello hay diferentes puntos a revisar de la maquina:

- Se revisa la configuración y el contenido de los ficheros **passwd y shadow**: los únicos usuarios con el shell bash asignada son “c0ldd” y “root”. El fichero /etc/shadow se encuentra protegido y solo es legible por root.
- El usuario c0ldd, haremos un **sudo -l** para ver si hay algún programa con permisos root que pueda ejecutar el usuario.
- Se buscan ficheros con permisos especiales **SUID y SGID**, no se encuentran ficheros con permisos especiales fuera del estándar para el correcto funcionamiento de Linux
- Revisar tareas programadas, para localizar si existe alguna que ejecute algún script con altos privilegios, pero no existe ninguna.
- Y **revisar la versión del kernel** ya que puede haber alguna de las versiones más antiguas de kernel de Linux que pueda ser vulnerable, pero no es nuestro caso.

En nuestro caso el fallo de seguridad se encontrara en que al ejecutar sudo -l, nos llevara a tres binarios de los cuales pertenecen a root pero el usuario tiene permiso para ejecutarlas usando sudo.

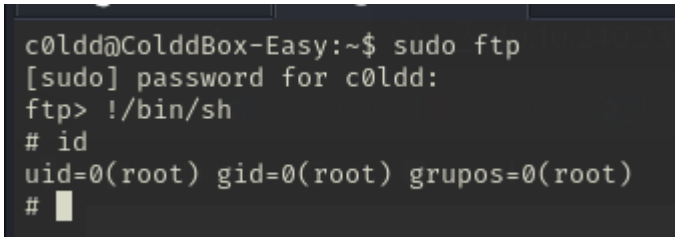
```
c0ldd@ColddBox-Easy:~$ sudo -l
[sudo] password for c0ldd:
Coincidiendo entradas por defecto para c0ldd en ColddBox-Easy:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

El usuario c0ldd puede ejecutar los siguientes comandos en ColddBox-Easy:
    (root) /usr/bin/vim
    (root) /bin/chmod
    (root) /usr/bin/ftp
c0ldd@ColddBox-Easy:~$
```

Se trata de los binarios de vim (un editor de texto, que se usa en Linux para editar ficheros en modo terminal), la herramienta chmod (Que sirve para otorgar permisos UGO en Linux) y el FTP (el cual es un protocolo de transferencia de archivos).

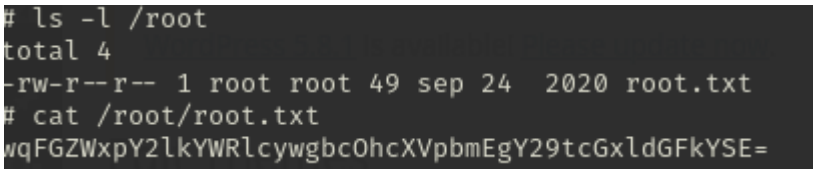
Estos tres binarios podrían ser nos de utilidad para escalar privilegios ya que gracias a la página de GTFoBins, podemos encontrar cierto comando para ejecutar ciertos ficheros que podrías dar nos acceso root.

En mi caso voy a usar ftp, el cual sería bastante sencillo ya que lo que podemos hacer será con sudo delante ejecutar *ftp* de forma normal y cuando nos abra una *shell* con el prompt de *ftp>* introduciremos! /*bin/sh.*

Pero si este mismo comando lo ejecutamos sin sudo delante, lo que nos ara será mostrar una shell normal, eso es debido a que sudo es lo que hace que pueda ejecutar ftp como usuario privilegiado.



Y ahora ya tendrías la flag de root



Escenario ColddBox-Easy

- Flag del **usuario**: RmVsaWNpZGFkZXMsIHByaW1lciBuaXZlbCBjb25zZWd1aWRvIQ==
- Flag de **root**: wqFGZWxpY2lkYWRlcywgbcOhcXVpbmEgY29tcGxldGFkYSE=

Post-Explotación

Como tarea de **post-explotación** voy a nombrar algunas nociones básicas de seguridad para poder arreglar estos pequeños fallos que nos han hecho conseguir acceso al host y poder escalar privilegios.

- En primer lugar con respecto al acceso al panel de control de wordpress, abría que usar contraseñas un mas fuertes o que no se encuentre en los diccionarios comunes de contraseñas, ya que usando un simple diccionario común que encontramos en kali Linux hemos podido acceder al CMS.
- Por otro convendría actualizar la versión de wordpress ya que estos posibles fallos como es el que podamos insertar comandos del servidor en el editor de código en versiones más recientes se encuentra ya parcheado. Es por eso que es muy importante usar siempre la última versión.
- Y también convendría no usar el mismo usuario y contraseña para el usuario del servidor que para el usuario de la base de datos de wordpress sobre si es de un usuario con privilegios de sudo.

Reporte y Mitigación

VULNERABILIDAD

CVE: CVE el 2020-4046

SERVICIO: Wordpress Versiones 5.4 a anteriores.

INFORMACIÓN: En las versiones de wordpress afectadas, usuarios con pocos permisos (como colaboradores o autores) pueden aprovecharse de un bloque llamado “embed” para introducir código HTML sin restricciones en el editor de bloques. Cuando un usuario con permisos más altos ve estos mensajes más altos, podría permitir la ejecución de scripts en el editor o en la parte de administración de Wordpress (wp-admin).

MITIGACIÓN: y para poder mitigar este problema lo más recomendable seria actualizar la versión de wordpress ya que si por algo hay versiones más recientes es por eso para corregir este tipo de fallos de seguridad. Además de monitorear y restringir los privilegios que se le aplican a los usuarios sobre el sistemas. Así como existen muchos plugins de seguridad que pueden detectar y prevenir este tipo de ataques como pueden ser Wordfence Security, Sucuri Security, iThemes Security, BulletProof Security, All In One WP Security & Firewall, Security Ninja, MalCare Security, Cerber Security, Antispam & Malware Scan.

INFORMACIÓN ADICIONAL:

https://nvd.nist.gov/vuln/detail/CVE-2020-4046

https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2020-9046