

Máquina Wicca (Vulnynx)

De Ignacio Millán Ledesma Publicado el: 30 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:a0:97:e9	1	60	PCS Systemtechnik GmbH
10.0.2.12	08:00:27:39:63:a9	1	60	PCS Systemtechnik GmbH

- **Kali (Máquina Atacante):** 10.0.2.4
- **Máquina Víctima:** 10.0.2.12

Comprobamos si tenemos conexión con la máquina víctima, para ello ejecutamos el siguiente comando:

```
$ ping -c 1 10.0.2.12
```

```
PING 10.0.2.12 (10.0.2.12) 56(84) bytes of data.
64 bytes from 10.0.2.12: icmp_seq=1 ttl=64 time=0.510 ms

— 10.0.2.12 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.510/0.510/0.510/0.000 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 10.0.2.12 -sVC`

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 01:09 CEST
Nmap scan report for 10.0.2.12
Host is up (0.00032s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2 (protocol 2.0)
|_ ssh-hostkey:
|   256 3a:dc:d6:1d:84:b6:96:c0:8f:96:1e:65:a0:24:0e:fb (ECDSA)
|_  256 de:93:17:fb:3a:19:9c:e0:17:32:2d:a9:73:f7:c5:94 (ED25519)
80/tcp    open  http     Apache httpd 2.4.57 ((Debian))
|_ http-server-header: Apache/2.4.57 (Debian)
|_ http-title: Apache2 Debian Default Page: It works
5000/tcp  open  http     Node.js (Express middleware)
|_ http-title: VulnMyx Lab
MAC Address: 08:00:27:39:63:A9 (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 12.23 seconds
```

Como podemos comprobar la Máquina Víctima tiene abiertos los puertos **22**, **80** y **5000**.

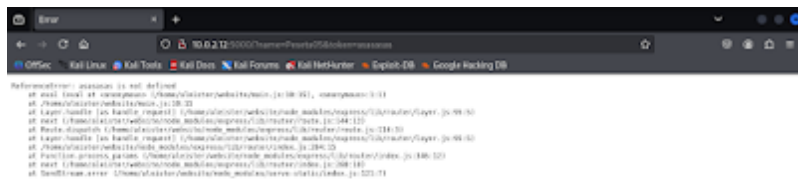
Comprobamos que es lo que corre en el puerto 5000.



Nos pide un texto, por lo tanto lo escribimos y pulsamos enter.

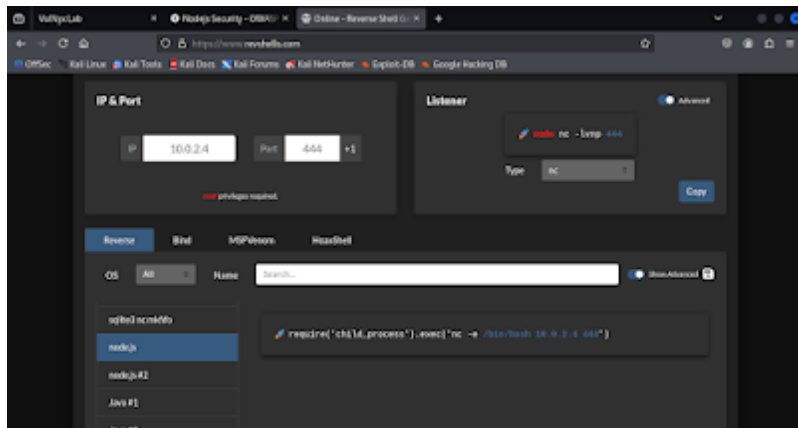


Nos damos cuenta que cambia la url añadiendo un nuevo parámetro "*token*", probamos cambiando esté parámetro por otro **numero** y no pasa nada, pero si lo cambiamos por **letras**, la web crashea.



Este error lo genera la función "`eval()`", investigando un poco nos damos cuenta de que esta función toma un argumento de cadena y lo ejecuta como cualquier otro código fuente de **Javascript**. Combinándolo con la entrada del usuario, este comportamiento conlleva una vulnerabilidad de **ejecución remota de código**.

Por lo tanto, nos generamos una reverse shell en **nodejs**.

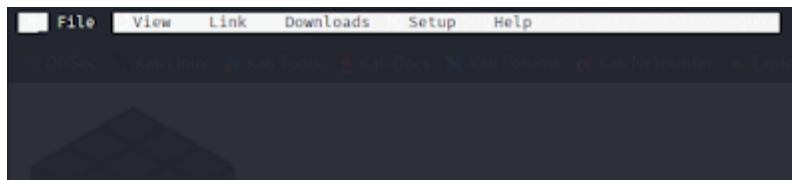


En nuestra Máquina Atacante con la ayuda de la herramienta **netcat(nc)** nos ponemos a la escucha por el puerto **444** por donde vamos a recibir la conexión, para ello ejecutamos el siguiente comando:

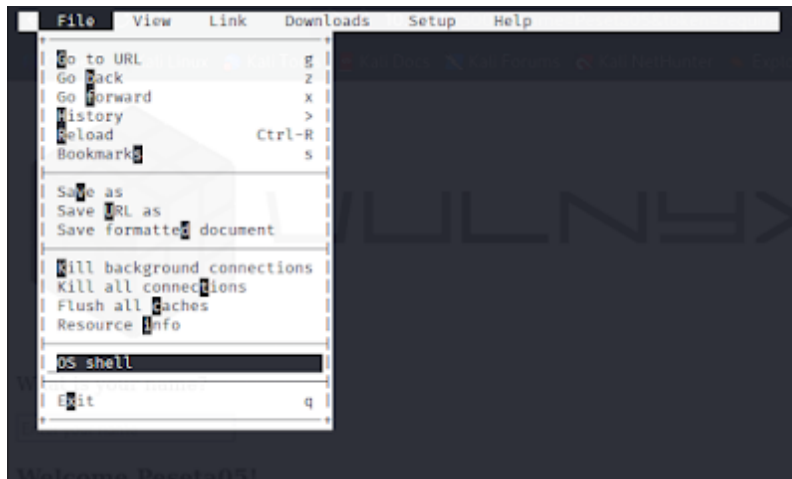
```
$ nc -lvp 444
```

```
listening on [any] 444 ...
```

Pegamos el **one-liner** creado anteriormente en el parámetro "`token`" y le damos enter.



Y con la flechas me desplazo por **File > OS shell**.



¡¡¡ Ya somos **root**!!!

```
root@wicca:/# whoami  
root
```

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

```
root@wicca:/home/aleister# cat user.txt  
VulNyx{d9f213df08ea2b3bf6cc90be28fa827f}  
root@wicca:/home/aleister# cd /root/  
root@wicca:~# cat root.txt  
VulNyx{dab686b0ee76b5edf6fc317c51d6f102}
```