



# EasyPeasy|Informe Tecnico

16/12/2022

—

Alejandro Sánchez Ramos

IPs de la Máquina: 10.10.192.21, 10.10.146.27, 10.10.246.23, 10.10.208.68

## Escaneo previo

Utilizando la herramienta nmap se hace un escaneo de todos los puertos de la máquina y se pregunta por el servicio alojado en ese puerto si está abierto y su versión, obteniendo que solamente estos puertos están abiertos:

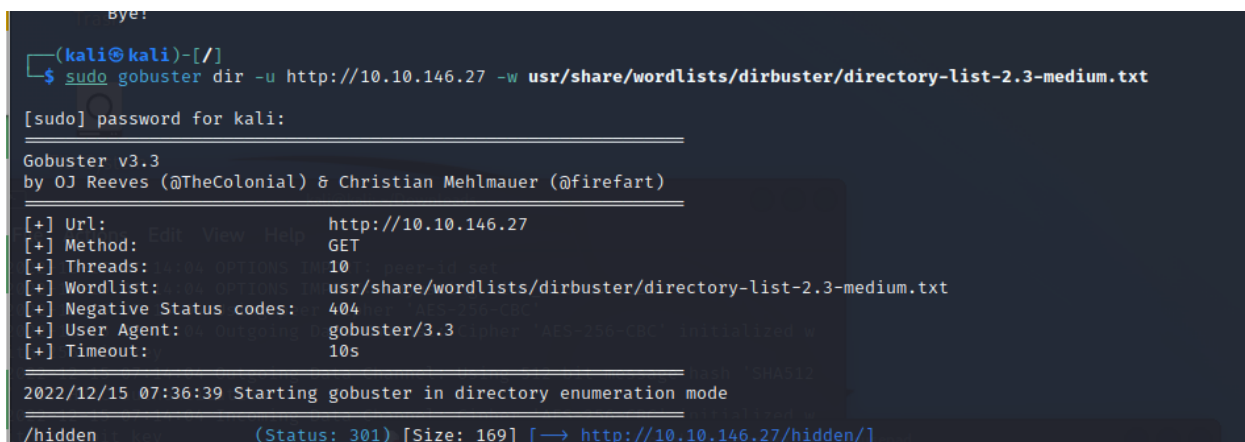
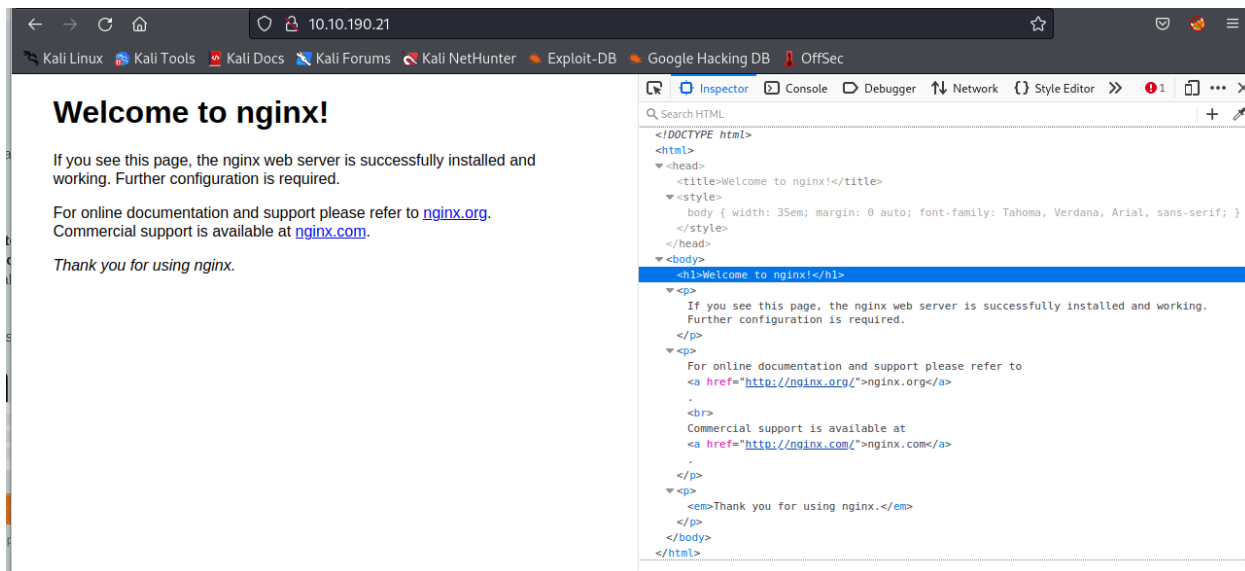
- 80/tcp - nginx v.1.16.1
- 6498/tcp - OpenSSH v.7.6p1 | Ubuntu 4ubuntu0.3
- 65524/tcp - Apache httpd v.2.4.43

```
(kali@kali)-[~]
$ sudo nmap -p- -sC -sV -Pn -T4 10.10.190.21
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-14 02:33 EST
Stats: 0:00:57 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 02:34 (0:00:06 remaining)
Stats: 0:01:01 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.52% done; ETC: 02:34 (0:00:00 remaining)
Stats: 0:01:04 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.76% done; ETC: 02:34 (0:00:00 remaining)
Stats: 0:01:05 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 91.67% done; ETC: 02:34 (0:00:00 remaining)
Nmap scan report for 10.10.190.21
Host is up (0.051s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      nginx 1.16.1
|_ http-robots.txt: 1 disallowed entry
|_ /
|_ http-title: Welcome to nginx!
|_ http-server-header: nginx/1.16.1
6498/tcp  open  ssh       OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 30:4a:2b:22:ac:d9:56:09:f2:da:12:20:57:f4:6c:d4 (RSA)
|   256 bf:86:c9:c7:b7:ef:8c:8b:b9:94:ae:01:88:c0:85:4d (ECDSA)
|_ 256 a1:72:ef:6c:81:29:13:ef:5a:6c:24:03:4c:fe:3d:0b (ED25519)
65524/tcp open  http      Apache httpd 2.4.43 ((Ubuntu))
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.43 (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 66.34 seconds
```

## Búsqueda en el servicio nginx

Primero miramos el código fuente de la aplicación nginx y en el robots.txt sin encontrar resultados notables, por lo cual se decide hacerle fuzzing a la web para comprobar si existen directorios ocultos, encontrando un directorio llamado hidden



```

2 <html>
3 <head>
4 <title>Welcome to ctf!</title>
5 <style>
6     body {
7         background-image: url("https://cdn.pixabay.com/photo/2016/12/24/11/48/lost-places-1928727_960_720.jpg");
8         background-repeat: no-repeat;
9         background-size: cover;
10        width: 35em;
11        margin: 0 auto;
12        font-family: Tahoma, Verdana, Arial, sans-serif;
13    }
14 </style>
15 </head>
16 <body>
17 </body>
18 </html>
19

```

Por lo que hacemos de nuevo un fuzzing al directorio hidden encontrando otro directorio llamado whatever, al comprobar el código fuente de la página se descubre un texto plano codificado en base64, pero sin ninguna información relevante, al no encontrarse más información se decide investigar el servicio apache



```
(kali㉿kali)-[~] Ubuntu 20.04 with NetPlan
$ echo "ZmxhZ3tmMXJzN19mbDRnfQ==" | base64 -d
flag{f1rs7_fl4g}
```

## Búsqueda en el servicio Apache


Primero se comprueba el archivo robots.txt encontrando en el user-agent un texto codificado que al identificarlo se muestra como un posible md5 pero al decodificarlo no se encuentran resultados notables

```
User-Agent:*  
Disallow:/  
Robots Not Allowed  
User-Agent:a18672860d0510e5ab6699730763b250  
Allow:/  
This Flag Can Enter But Only This Flag No More Exceptions
```

```
alejandro@alejandro-HP-Pavilion:/$ hashid a18672860d0510e5ab6699730763b250  
Analyzing 'a18672860d0510e5ab6699730763b250'  
[+] MD2  
[+] MD5  
[+] MD4  
[+] Double MD5  
[+] LM  
[+] RIPEMD-128
```

Md5 hash digest

a18672860d0510e5ab6699730763b250

 Copy Hash

Md5 digest unhashed, decoded, decrypted, reversed value:

flag{1m\_s3c0nd\_fl4g}

Copy Value  Copy Value

[Blame this record](#)

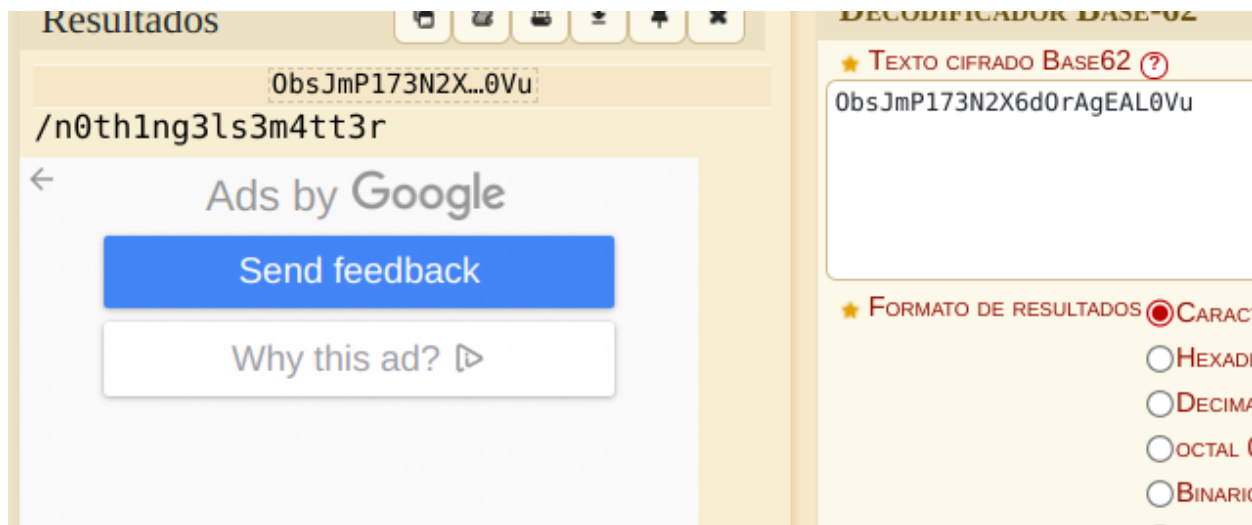
Después de mirar el archivo robots.txt, se comprueba el código fuente del apache y utilizando el buscador del navegador se buscan las palabras hidden y flag encontrando un texto codificado(ObsJmP173N2X6dOrAgEAL0Vu)

```

</li>
<li>
  They are activated by symlinking available
  configuration files from their respective
  Fl4g 3 : flag{9fdafbd64c47471a8f54cd3fc64cd312}
  vailable/ counterparts. These should be managed
  by using our helpers
  <tt>
    a2enmod,
  }
</style>
</head>
<body>
  <div class="main_page">
    <div class="page_header floating_element">
      
        Apache 2 It Works For Me
      <p hidden>its encoded with ba....:ObsJmP173N2X6dOrAgEAL0Vu</p>
    </span>
  </div>
!--
  <div class="table_of_contents floating_element">
    <div class="section_header section_header_grey">
      TABLE OF CONTENTS

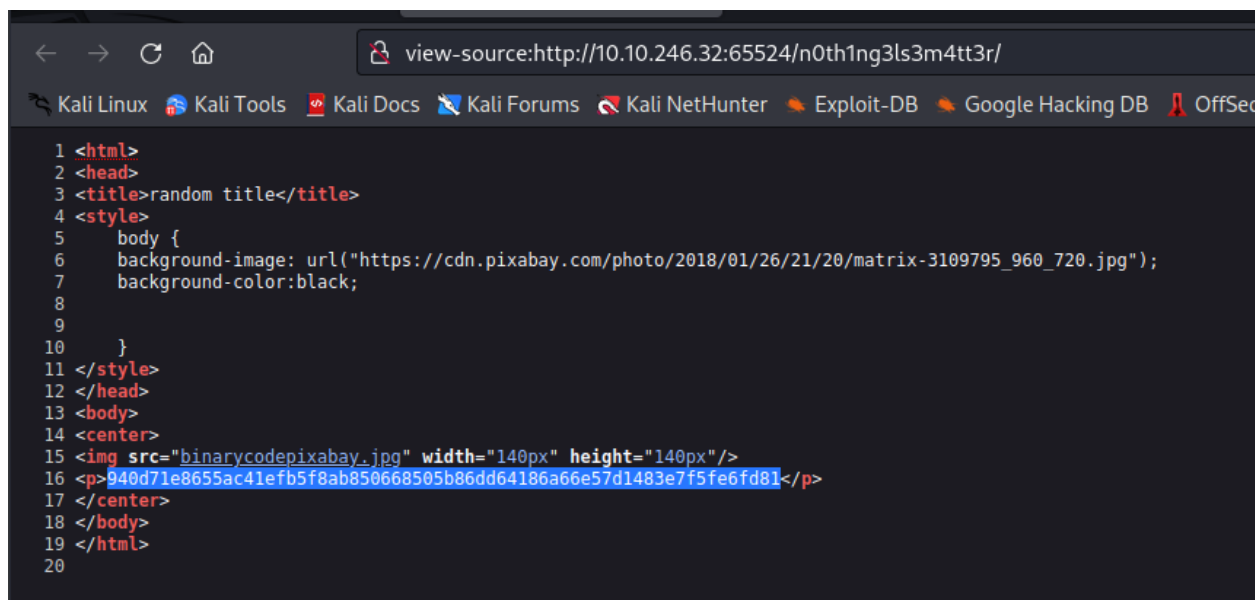
```

Se identifica el texto de hidden y investigando se descubre que es una codificación base 62, que nos da un directorio /n0th1ng3ls3m4tt3r





Al inspeccionar el código fuente del directorio encontrado, se descubre una imagen y un hash de formato desconocido(GOST)



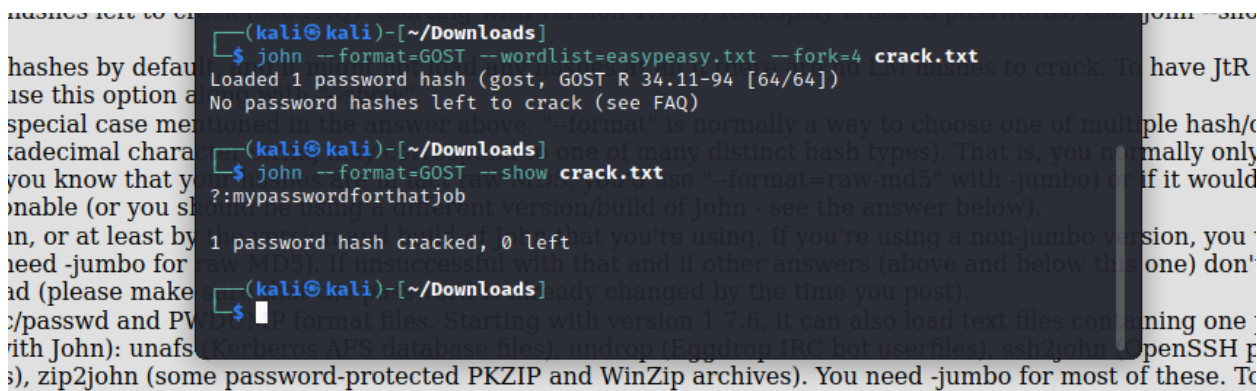
```

1 <html>
2 <head>
3 <title>random title</title>
4 <style>
5   body {
6     background-image: url("https://cdn.pixabay.com/photo/2018/01/26/21/20/matrix-3109795_960_720.jpg");
7     background-color:black;
8   }
9 }
10 </style>
11 </head>
12 <body>
13 <center>
14 
15 <p>940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81</p>
16 </center>
17 </body>
18 </html>
19
20

```

Utilizando john the ripper se intenta descifrar el hash utilizando un diccionario , se termina descifrando lo que parece una contraseña

- mypasswordforthatjob



```

(kali㉿kali)-[~/Downloads]
$ john --format=GOST --wordlist=easypeasy.txt --fork=4 crack.txt
Loaded 1 password hash (gost, GOST R 34.11-94 [64/64])
No password hashes left to crack (see FAQ)

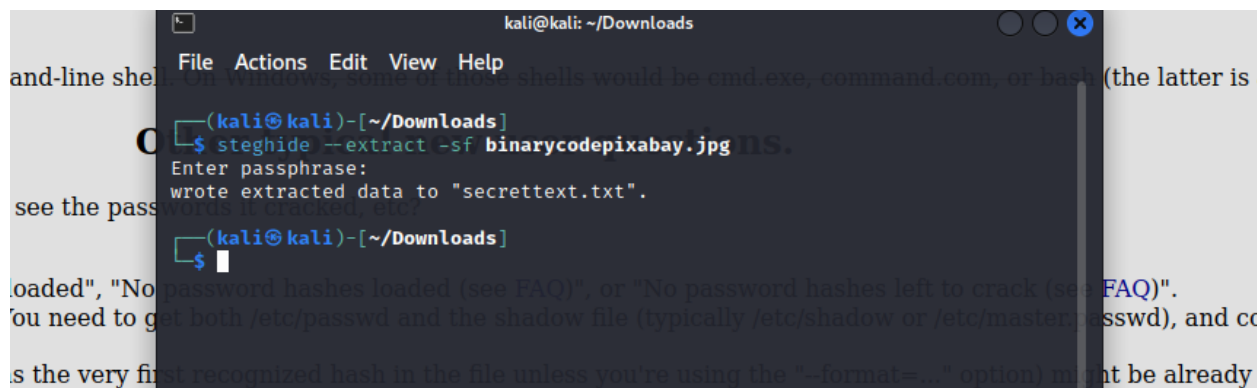
(kali㉿kali)-[~/Downloads]
$ john --format=GOST --show crack.txt
?mypasswordforthatjob

1 password hash cracked, 0 left at you're using. If you're using a non-jumbo version, you
(kali㉿kali)-[~/Downloads]
$ john --format=GOST --show crack.txt
?mypasswordforthatjob

(kali㉿kali)-[~/Downloads]
$ john --format=GOST --show crack.txt
?mypasswordforthatjob

```

Al encontrar una "supuesta" contraseña decido, seguir comprobando la imagen encontrada también en la página utilizando steghide y la contraseña encontrada anteriormente, extrayendo un fichero txt llamado "secrettext.txt"



```
kali@kali: ~/Downloads
File Actions Edit View Help
(kali@kali)-[~/Downloads]
$ steghide --extract -sf binarycodepixabay.jpg
Enter passphrase:
wrote extracted data to "secrettext.txt".
(kali@kali)-[~/Downloads]
$
```

Al inspeccionar el archivo txt se muestra un usuario(boring) y una contraseña en binario, que se intentara utilizar para el logueo en el ssh, y al pasar el binario texto plano se destapa la contraseña(iconvertedmypasswordtobinary)

```
1 username:boring
2 password:
3 01101001 01100011 01101111 01101110 01101110 01100101 01110010 01101000 01100101 01100100 01101101 01111001
01110000 01110001 01110011 01110011 01101111 01100010 01100100 01101000 01101111 01100010 01101001
01101110 01100001 01110010 01111001
```

## Traducción de Binario a Texto

### Texto resultado:

iconvertedmypasswordtobinary

### Binario original:

```
01101001 01100011 01101111 01101110 01101110 01100101 01110010 01101000 01100101
01100100 01101101 01111001 01110000 01100001 01110011 01110011 01101111 01101111
01110010 01100100 01101000 01101111 01100010 01101001 01101110 01100001 01110010
01111001
```



## Conexión a la máquina por ssh

Utilizando los credenciales descubierto con anterioridad nos conectamos a la máquina, descubriendo un fichero llamado user.txt, que parece estar codificado según se menciona en el fichero

```
(kali㉿kali)-[/]
$ ssh boring@10.10.146.27 -p 6498
*****
**          This connection are monitored by government offical          **
**          Please disconnect if you are not authorized                  **
** A lawsuit will be filed against you if the law is not followed        **
*****
boring@10.10.146.27's password:
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!!!
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!!!
boring@kral4-PC:~$
```

```
boring@kral4-PC:~$ ls
user.txt
boring@kral4-PC:~$ cat user.txt
User Flag But It Seems Wrong Like It`s Rotated Or Something
synt{a0jvgf33zfa0ez4y}
boring@kral4-PC:~$ cd ..
```

Al investigar se descubre que es un cifrado llamado ROT13, que consiste en rotar las letras del abecedario , y utilizando la tabla expuesta debajo se descifra el texto obteniendo una frase(flag{n0wits33msn0rm4l}) pero sin importancia para hacer una escalada de privilegios

