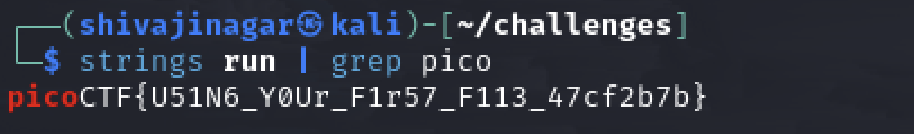
**Reverse Engineering**

1

Graphical user interface, text, application

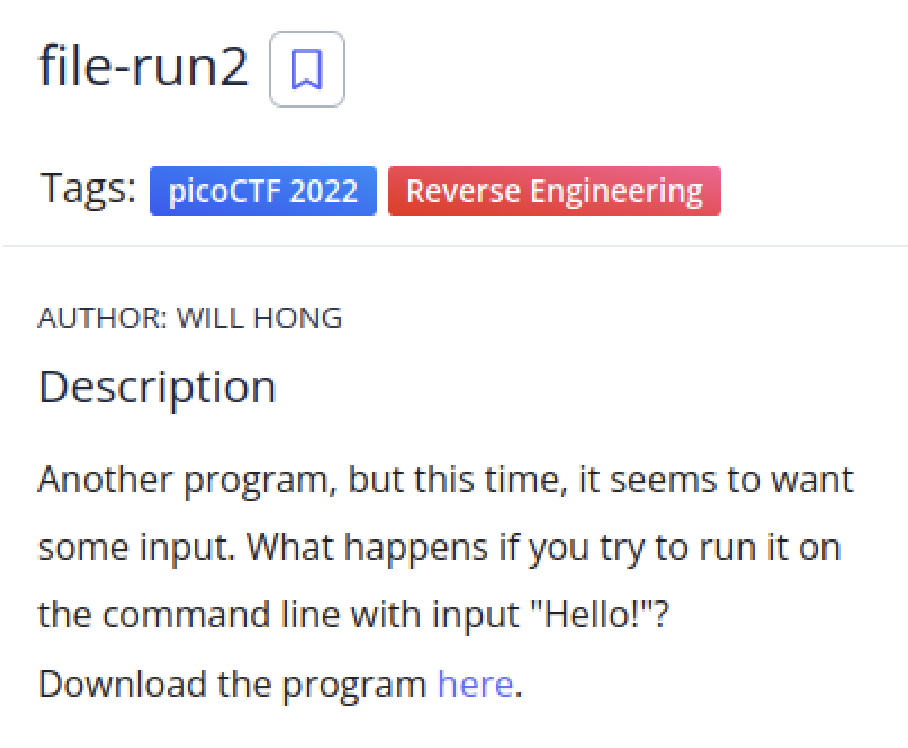
Description automatically generated

Solution

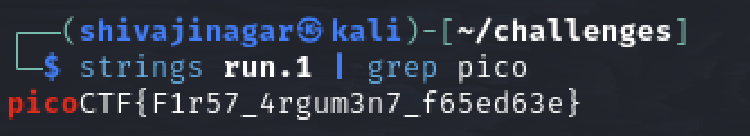


We can find the flag by using the strings command.

2.



Solution

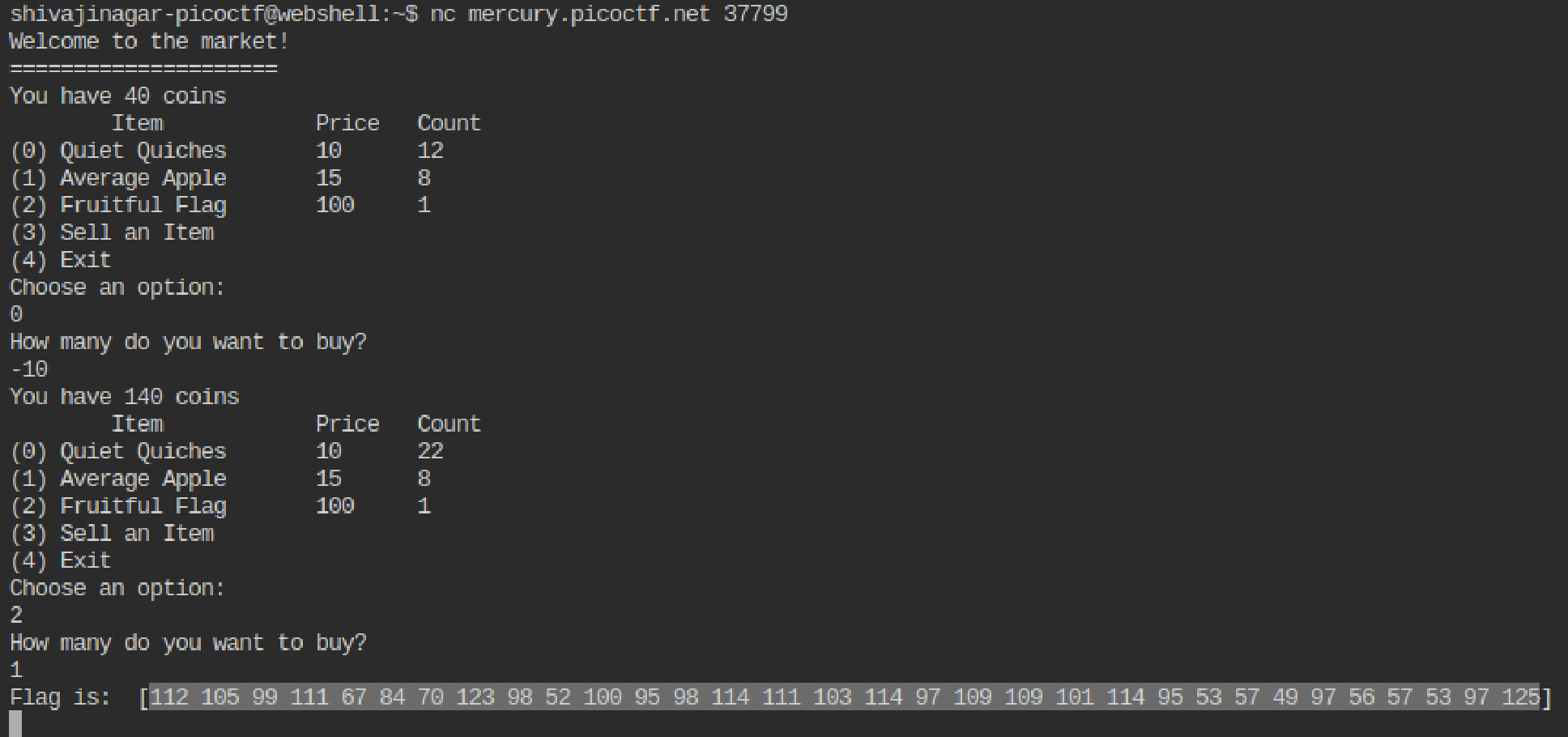


This flag can be found using the strings command

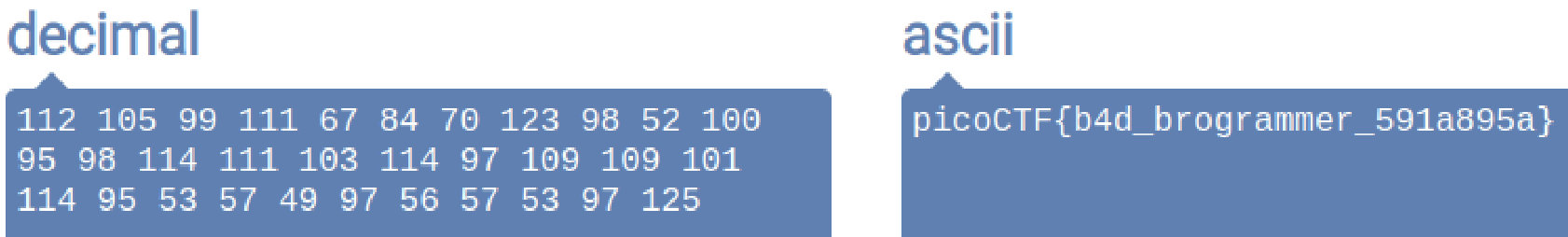
3.



Solution

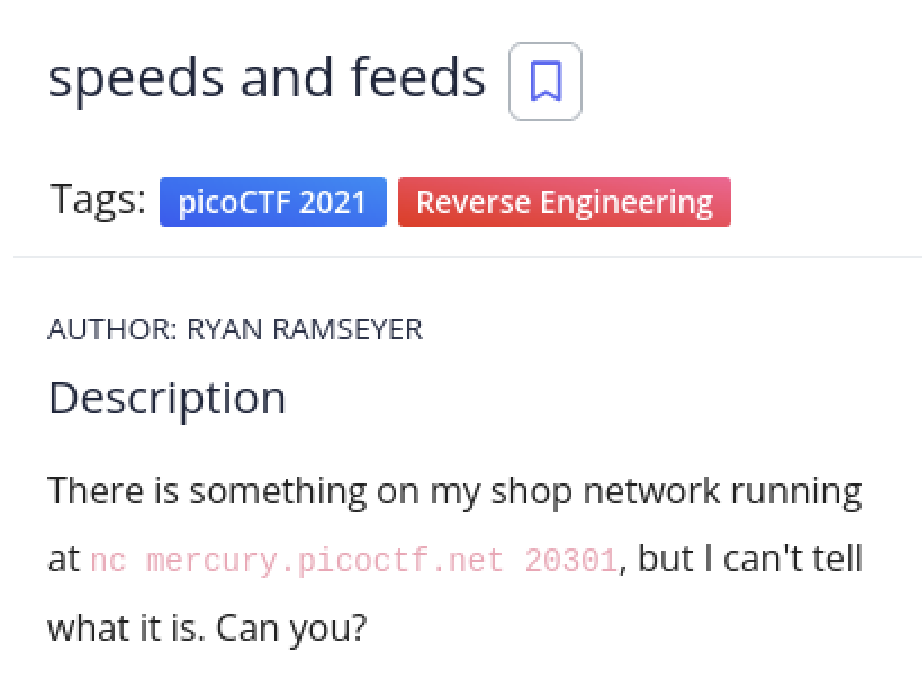


As there are no edge cases, more coins have been credited when we buy -10 quiches. So using the money, we can buy a flag.



Using an online decimal to ASCII converter, we can find the flag.

4.

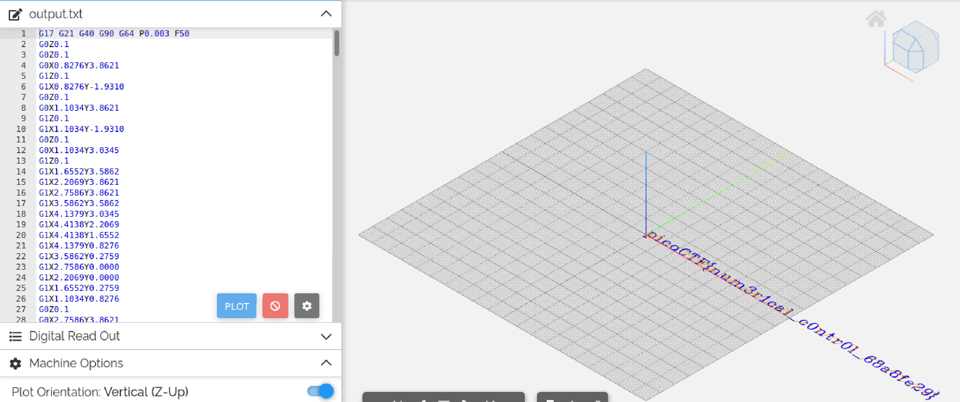


Solution

Text

Description automatically generated

Dumping the output of this command into a file called output.txt



Using an online g-code simulator, we find the file.

5.

Graphical user interface, text, application

Description automatically generated

Solution

Text

Description automatically generated

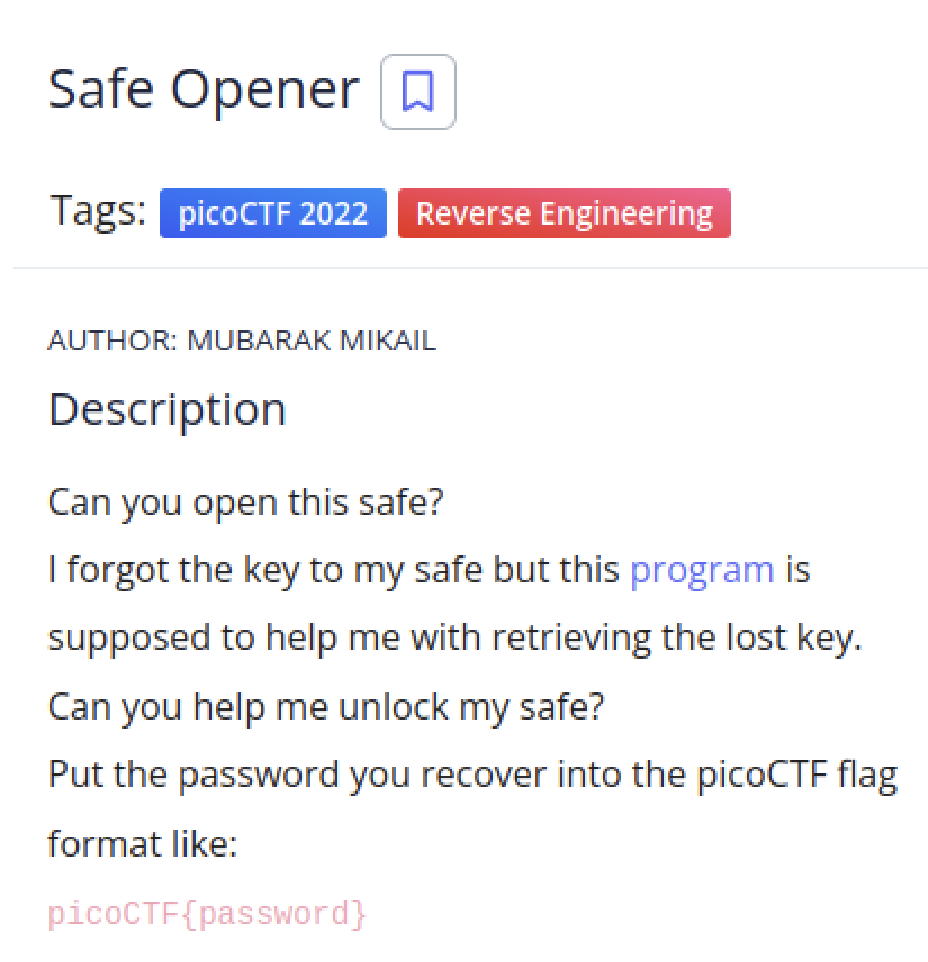
Adding a line to decode the message declared in the python script reveals the flag.

Text

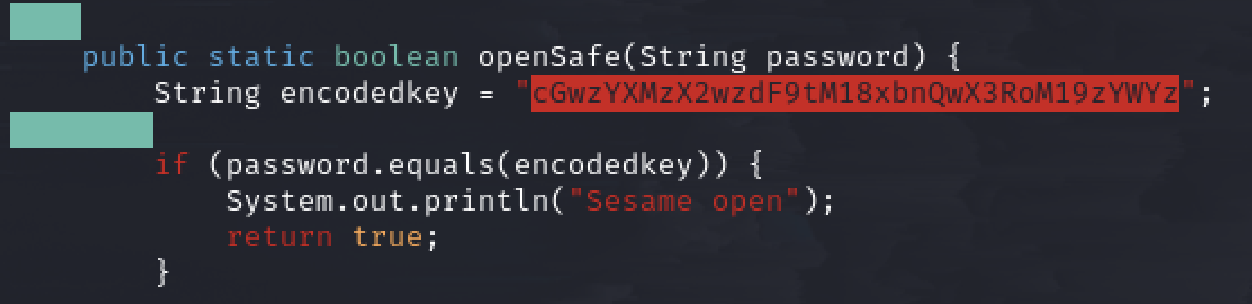
Description automatically generated

This is the flag shown.

6.



Solution



In the openSafe function in the java file, we come across a string called encodedkey which is base 64 encoded and must match with our password. So, if we decode it, we would get the flag.

Text

Description automatically generated

7.

Graphical user interface, text, application

Description automatically generated

Solution



Looking at the java code, we have the characters of the password in a scrambled order.

We have the following flag after descrambling it.

