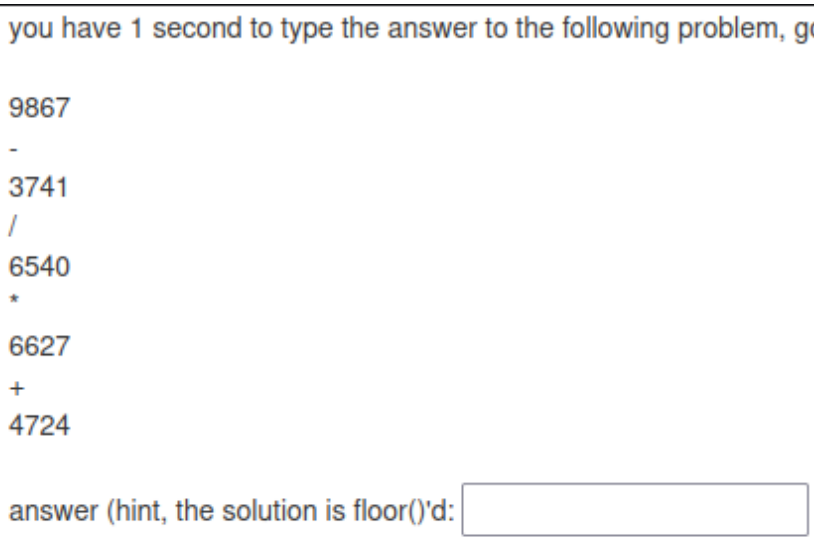
calc

Challenge

We are given a website that gives a math problem, and states that we must solve the math problem within one second of loading the page.



In the source code for the website, I saw the math is stored in a custom HTML element tagged expression.
This can be extracted and evaluated using python.

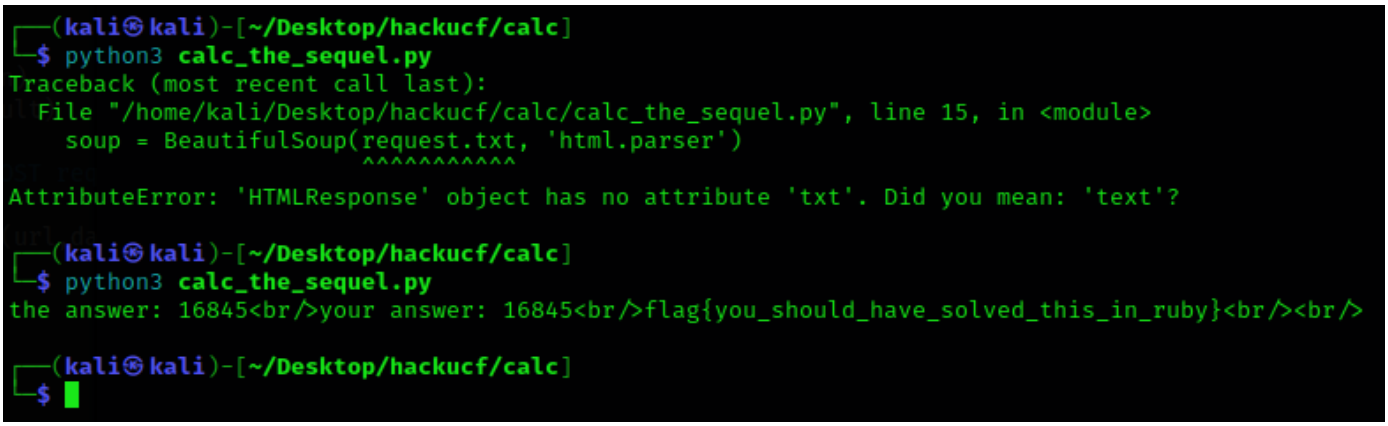
```
<link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">

<div class="container">
you have 1 second to type the answer to the following problem, good luck! if
only there was a better way...<br/></br><expression>9867<br/>-
<br/>3741<br/>/<br/>6540<br/>*<br/>6627<br/>+<br/>4724</expression>
  <form action="calc.php" method="POST">
   <br/> answer (hint, the solution is floor()'d:
     <input type="text" name="answer" class="input" value=""/><br/>
  </form>
    </br>
</div>
```

Solution

Using the BeautifulSoup, math, and HTMLSession libraries I created a script in Python to load, solve, and send the math problem to the server.

```python
from bs4 import BeautifulSoup
import math
from requests_html import HTMLSession

# The URL of the webpage
url = "http://ctf.hackucf.org:4000/calc/calc.php"

# Create session
session = HTMLSession()

# Access calc website
request = session.get(url)

# Parse the HTML content
soup = BeautifulSoup(request.text, 'html.parser')

# Extract the expression
expression = soup.find('expression').text

# Find the answer
result = eval(expression)
result = math.floor(result)

# Send the answer via POST request to the calc website
data= {'answer':result}
response = session.post(url,data=data)

# Print the response
print(response.text)
```

Here, an HTML Session is first created.
Then the source code of the page is dynamically loaded using the BeautifulSoup library to obtain the math problem.
The equation is extracted by finding the custom expression element.
Then the equation is simply solved with the eval() function, and floored using the math library's floor() function.
Using BurpSuite, I intercepted the data sent by the webpage and found that it is delivered via POST request in which the input is stored as "answer".
Thus, I save the result of the calculations as JSON data {"answer" : result} before sending the data to the server.
Finally, the response of the server is printed.



When running the script in the terminal, the URL is loaded, the math is located and evaluated, and then formatted and sent to the server via POST request.

The response the server gives the flag.