Challenge: Spookifier

Challenge Description :

There's a new trend of an application that generates a spooky name for you. Users of that application later discovered that their real names were also magically changed, causing havoc in their life. Could you help bring down this application?

Context :

- For This one, you need to download the source code, analyze each function, and find a way to get the flag, (Hint: SSTI)

Notes :

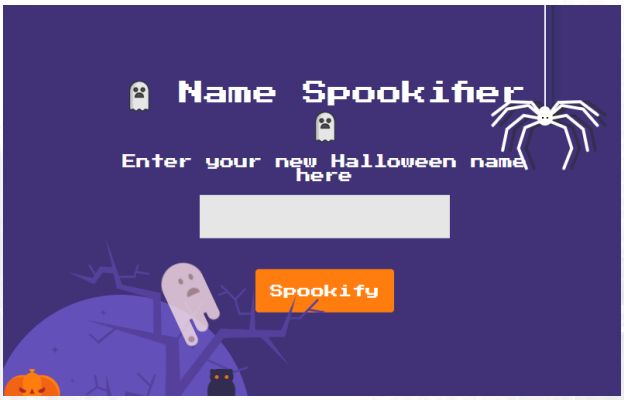
- Tools:
    - Browser - A browser like chrome, firefox, brave, librewolf.
    - VsCode - For browsing the code, you could also just use notepad or other alternatives.

Challenge :

- Basic analysis on a web-page.

Flag :

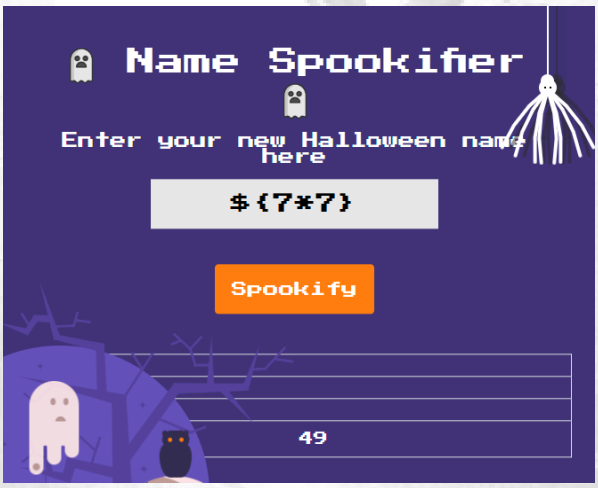
- Start the instance, go to the website and open the Inspect the page, find what you can, meanwhile try and check the source code to link it to the web-page to get a better understanding of how it works.



- Looking at the code we see that the function 'Generate_render' looks vulnerable, with no input sanitation or any checks we can possibly abuse this to get the flag.

```
}

def generate_render(converted_fonts):
    result = '''
        <tr>
            <td>{0}</td>
        </tr>

        <tr>
            <td>{1}</td>
        </tr>

        <tr>
            <td>{2}</td>
        </tr>

        <tr>
            <td>{3}</td>
        </tr>

    '''.format(*converted_fonts)

    return Template(result).render()
```

- We also check what framework / engine is running this web-application and it seems to be Mako that is used to render the HTML with the Input that is given in the text-box.
- Now we know it is running on python3 and using Mako for the input we can test the input sanitation theory with a basic math equation.



- When we input 7x7 we get the answer making it correct the theory of no input sanitization, (You can just check the code more than once to know for sure), this works because the python engine is used to compute the equation.
- Now for the flag, we will try to run some basic linux commands within some python code, to enter in the input that will run on the engine to render on the webpages html.
- The POC is

  " ${open('/flag.txt').read()} "

- Then we get the flag : HTB{t3mpl4t3_1nj3ct10n_C4n_3x1st5_4nywh343!!}