Writeup P2 Tarea 1

Writeup P2 Tarea 2

Nombre: Benjamín Aguilar Osorio
Equipo: L05 1NCR31BL35

Persona a cargo de la pregunta **Joaquin Oportus**

Descripción general del problema

Se trata de un problema de web, donde se presenta una pagina web con una interfaz de busquedas que tiene implementado un estado de aceptacion asociado a la cantidad de aciertos en la pagina, donde los aciertos estan determinados por las respuestas correctas a cada "pregunta" que hace la pagina

Metodología de resolución

A continuación se darán los **pasos** que se llevaron a cabo para resolver el problema:

1. Para entender el funcionamiento del servidor, se parte por analizar el inspector de la página web, revisando inicialmente el HTML de la página.
2. Al revisar el HTML, se puede descubrir un valor "hidden" que expresa la pregunta que se debe contestar en la página para que sea considerada correcta.
3. Este valor oculto puede ser modificado y transformado en un valor vacío, de manera que, sin importar el valor de la pregunta que se entregue, se considere como una respuesta correcta.
4. Teniendo esto en consideración, se decidió ejecutar la página varias veces, modificando el valor oculto para aumentar la cantidad de adivinaciones. Sin embargo, después de leer en el foro que eso no se debe hacer, se continuó investigando el inspector de la página web.
5. Al seguir investigando el inspector, en particular el almacenamiento de la página, se pudo encontrar que el caché de la página se guarda de forma local y que este puede ser modificado. Además, este valor corresponde con la cantidad total de adivinaciones.
6. Para continuar con la solución, se consultó con el compañero a cargo del problema, quien indicó que se revisaran los encabezados de las páginas web presentes en la pestaña de RED. Esto se hizo con el fin de determinar el framework que se utiliza internamente en la página.
7. Al analizar los encabezados de las páginas solicitadas, se puede observar que, internamente, la página funciona con Python, por lo que posiblemente utiliza Flask.
8. Para probar si el servidor utiliza Flask, se realiza la consulta {{1+1}}. Si esto se resuelve correctamente, se puede determinar que trabaja con Flask. Efectivamente, se determina que trabaja con Flask.

9. Considerando esto, se decidió realizar una inyección de Flask, siguiendo las instrucciones de la siguiente página web: https://kleiber.me/blog/2021/10/31/python-flask-jinja2-ssti-example/.
10. Siguiendo estas instrucciones se llegó hasta

```
{{'abc'.__class__.__base__.__subclasses__()[96].__subclasses__()
[0].__subclasses__()}}
```

Para continuar, se consultó al compañero a cargo por una instrucción adicional. Él indicó que se consultara por los archivos más comunes de los programas Flask, en particular, se consultó por el archivo app.py.
11. Este archivo se consultó mediante la siguiente instrucción:

```
{{'abc'.__class__.__base__.__subclasses__()[96].__subclasses__()
[0].__subclasses__()[0]("app.py").read()}}
```

12. Al realizar esta consulta usando la instrucción mencionada, se obtiene el contenido del archivo app.py, el cual se ingresó a ChatGPT para hacer un código más legible.
13. Al leer el código entregado, se pudo determinar la existencia de un HTML que podría contener la flag de la página.
14. Para acceder al HTML de la página, se utilizó el siguiente "comando":

```
{{'abc'.__class__.__base__.__subclasses__()[96].__subclasses__()
[0].__subclasses__()[0]("templates/home.hmtl").read()}}
```

15. Después de consultar el código de la instrucción anterior, se obtiene un HTML ilegible, por lo que se ingresó a ChatGPT para obtener un código legible.
16. En este código, se puede determinar que la condición para entregar la flag está relacionada con la cantidad de adivinaciones correctas. En particular, cuando se llega a **757757757** adivinaciones correctas, se entrega la flag.
17. Finalmente, para alcanzar la cantidad de adivinaciones, se modifica la cantidad en el caché de la página inicial, asignándole el valor de **757757756**, y cambiando el valor oculto correspondiente a la pregunta de la adivinación.
18. Por último, se envía una solicitud nuevamente para alcanzar el número de adivinaciones. De esta forma, se consigue la flag **cc5325{reg3x_g3x_g3x_4r3_fun}**