Hoja de Trabajo - Reto de Seguridad Flask: "Protege tu App"

Instrucciones Generales

Trabaja en equipo (3 personas máximo). Abre el archivo `app.py` entregado por el docente. Deberás analizarlo, descubrir errores de seguridad y aplicar mejoras. No se permite preguntar al docente sobre teoría. Ustedes deben investigar, experimentar y documentar lo aprendido.

Pistas del Reto

1. **¿Dónde se guarda la contraseña? ¿Está protegida?**

   En el código, la contraseña aparece escrita directamente, como texto plano (`admin123`). Esto es peligroso porque **cualquiera que vea el código podría acceder al sistema** sin permiso, por otro lado, **NO** se utiliza ninguna técnica de protección como hashing o cifrado.

2. **¿Qué pasa si ingreso `" OR "1"="1` como contraseña?**

   Aunque la aplicación no se conecta a una base de datos relacional ni realiza consultas SQL, esta entrada puede generar una **excepción no controlada**, ya que no hay un sistema que valide correctamente los datos de entrada. En esos casos el sistema podría comportarse de forma inesperada si no se validan bien los datos, lo cual nos hace reflexionar sobre lo importante que es validar la entrada del usuario.

3. **¿Qué mensaje muestra cuando algo falla? ¿Da demasiada información?**

   El mensaje de error muestra el nombre de usuario ingresado incluso cuando es inválido, lo que revela información importante. Esto **NO** es recomendable porque le da pistas a un posible atacante. En lugar de eso, lo ideal es mostrar mensajes más genéricos como “Información incorrecta”, sin dar más información.

4. **¿Qué pasa si dejo los campos vacíos?**

   No hay validación de campos vacíos. Esto puede provocar errores no controlados, o incluso que la aplicación se detenga (caída de la página). Esto no solo afecta la experiencia del usuario, sino que también abre puertas a errores más graves si se conecta a bases de datos u otros servicios.

5. **¿Podría alguien ver la contraseña original si entra a la base de datos?**

   En este caso, como no hay conexión a una base de datos, la contraseña es aún más vulnerable, ya que se encuentra literalmente en el código fuente. Por eso, una de las primeras reglas de la seguridad es: nunca guardar contraseñas tal cual como se escriben.

Mejoras realizadas (escribir aquí las modificaciones y por qué las hicieron)

- ### Implementación de Hashing o cifrado de contraseñas

Usamos `werkzeug.security` para generar una versión cifrada de la contraseña. Así, aunque alguien vea esa versión en la base de datos, no podrá saber cuál es la original.

- ### Validación de Entradas

Ahora el sistema revisa que los campos no estén vacíos antes de enviar el formulario. Si alguno lo está, muestra un mensaje claro:

“Todos los campos son obligatorios.”

- ### Manejo de Errores Mejorados

Cambiamos los mensajes para que no revelen información sobre el usuario o la contraseña. Ahora, si algo está mal, simplemente dice:
“Información incorrecta.”

Reflexión grupal

(Escribir una conclusión conjunta sobre lo aprendido en este reto. ¿Qué fue lo más difícil? ¿Qué les sorprendió? ¿Qué entendieron sobre seguridad?)

Lo más difícil de este trabajo para todos nosotros fue el tema de python ya que no sabemos usarlo bien por ende nos tocó ver tutoriales y dejarnos llevar por devdoc y MDN para la documentación de python y que nos sorprendió es que hay programas muy malos y programadores malos eso es para que vean que lo malo sale caro por eso se recomienda una buena gestión y usar metodologías ágiles y hacer reuniones scrum y que haya más comunicación y lo que más entendimos fue que se nos expande más el conocimiento respecto a hashing , y más el documento de apoyo fue más útil para reconocer todo lo de seguridad por eso hay que tener muy presentes cada cosa y ser atento y una frase “Confiar es bueno pero no confiar es mejor ”

Integrantes del equipo:

- Nombre 1: Samuel Alzate Echeverrí
- Nombre 2: Juan Jose Rivera Vergara
- Nombre 3: Maria Camila Rojas Ospina

Fecha: 15/04/2025

REFERENCIAS

- https://devdocs.io/
- https://developer.mozilla.org/en-US/
- https://youtu.be/Uyy3kAfm-1w?si=ZVNXARWstvOJqmhd
- https://youtu.be/nKPbfIU442g?si=LV2TqEGUG3rRona3