Alumno: Rodrigo Vila

Ejercicios Módulo 3 Unidad 1

Ejercicio Número 1 Unidad 1

En caso de tener conocimientos de programación WEB, ¿te animas a exponer qué errores se pueden encontrar en esa configuración anterior?:

```
<form action="login_form.php">
  <input id="username" name="username"/>
  <input id="password" name="password"/>
  <input id="rememberMe" name="rememberMe"/>
  <input id="login" name="login"/>
  <input type="submit"/>
</form>
```

Existen implicaciones de seguridad a considerar en éste formulario claramente de inicio de sesión:

El formulario envía datos utilizando el método “GET” (predeterminado para “<form>” sin el atributo “method”). Esto expone el nombre de usuario y la contraseña en la URL, que es visible en el historial del navegador y en los registros del servidor.

También el campo "password" utiliza el tipo de entrada de texto predeterminado. Esto significa que la contraseña se muestra en texto sin formato a medida que el usuario la escribe. Esto no es seguro ya que cualquier persona cercana podría ver la contraseña.

Para mejorar la seguridad de este formulario de inicio de sesión implementaría lo siguiente:

En primera instancia resolveré lo más básico que tengo a simple vista, aunque sería una buena práctica implementar validación del lado del servidor para garantizar que los nombres de usuario y las contraseñas cumplan con criterios específicos (longitud, tipos de caracteres).

Utilizar el método “POST” en la etiqueta “<form>” para enviar datos de forma segura, ocultándolos de la URL.

Agregarle al campo “password” el “type=”password”. Esto enmascara los caracteres a medida que se escriben, lo que mejora la seguridad.

También no puedo dejar de notar que el formulario no tiene títulos ni “labels” y al campo “rememberMe” le falta el “type=checkbox” pero es un detalle que no compete a la seguridad pero hacen que la visualización del código en el front-end sea un desastre. Vamos por partes…

Remarco en color rojo las recomendaciones mencionadas:

```
<!-- Inicio del código -->
<form action="login_form.php" method="post">
        <input id="username" name="username"/>
        <input id="password" type="password" name="password"/>
        <input id="rememberMe" type="checkbox" name="rememberMe"/>
        <input id="login" name="login"/>
        <input type="submit"/>
</form>
<!-- Final del código -->
```

Implementar PHP también proporcionaría algunas buenas prácticas para la seguridad, como protección para ataques de inyección de código, limitando longitud de carácteres, etc. Y para ataques CSRF, generando un token del lado servidor.

Algo de información sobre los ataques CSRF (Cross-site request forgery):

“El CSRF es un tipo de exploit malicioso de un sitio web en el que comandos no autorizados son transmitidos por un usuario en el cual el sitio web confía. Esta vulnerabilidad es conocida también por otros nombres como XSRF, enlace hostil, ataque de un clic, secuestro de sesión, y ataque automático.”

Esto se solucionaría, como mencioné anteriormente, generando un **CSRF Token**. “El CSRF Token es un valor único, secreto e impredecible que genera la aplicación del lado del servidor y se transmite al cliente de tal manera que se incluye en la siguiente solicitud realizada por el cliente.”

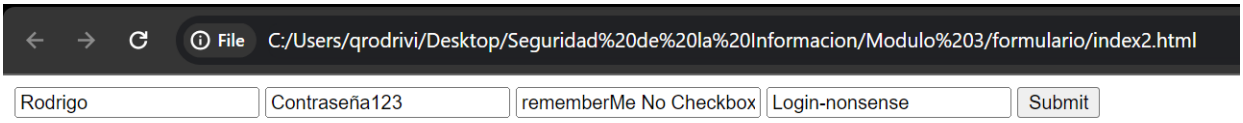
Implementaríamos la función del CSRF Token en el botón “Submit” escondida en el front-end con el “type=”hidden” masomenos de esta manera:

```
<!-- Inicio del código -->
<input type="hidden" name="csrf_token" value="<?php echo generateCSRFToken(); ?>">
<button type="submit">Login</button>
    <?php
```

```
    function generateCSRFToken() {
        return "ejemplo_csrf_token";
    }
    ?>
<!-- Final del código -->
```

Para agregarle un poquito de detalle al ejercicio (y satisfacer mi mal llamado TOC de front-end) adjunto algunas imágenes de la visualización del código crudo y otras con las implementaciones de seguridad mas detalles estéticos como títulos, labels y checkbox:

Crudo:



Modificado:

Ejercicio Número 2 Unidad 1

Crear un documento (DOC/PDF), en el cual este explicado con sus propias palabras, de acuerdo a lo leído en esta unidad, si las medidas son ¿suficientes o no?

Desarrollar: si son suficientes las políticas, justificar el porqué de cada una (no más de 1 página).

Si son insuficientes: explicar que agregaría y justificar el porqué de cada una (no más de 2 páginas).

También se pueden aportar nuevas ideas de medidas.:

Como dicta la consigna del ejercicio, publicare un PDF luego del posteo del presente documento con el desarrollo de la consigna.

Rodrigo Vila.-