APPS SEGURAS

Caso práctico 1.

Ramiro de Santos

INDICE:

1. **A07:2021-Identification and Authentication Failures**

No se requiere la autenticación de ningún usuario, por lo tanto entiendo que cualquier persona podría acceder a esta página, buscar un DNI y obtener la información del paciente que desee sin ningún tipo de problema. Entiendo que esta debilidad también se podría relacionar con la "A01:2021-Broken Access Control" ya que la conexión a BBDD no requiere de una contraseña, está en blanco.

```php
if(isset($_POST["dni"]))
{
    $dni=$_POST["dni"];
    $mysqli = new mysqli("localhost","root", "", "hospitalcentral");
    if ($mysqli->connect_errno)
    {
        echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->c
    }
    $consulta="SELECT * FROM pacientes  WHERE dni='$dni' ";
    $resultado=$mysqli->query($consulta);
    $row_cnt = $resultado->num_rows;
    if($row_cnt > 0)
```

2. **A06:2021 – Vulnerable and Outdated Components**

Habría que revisar que los links externos no suponen un riesgo para la seguridad de nuestro código.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" re
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.
```

3. **A03:2021-Injection**

La consulta a bbdd se hace sin que se compruebe que los datos introducidos por el usuario en el campo de búsqueda DNI sean realmente un DNI. El usuario podría introducir código que por ejemplo actualizara, borrara o accediera a información confidencial de la bbdd.

```
$dni=$_POST["dni"];
$mysqli = new mysqli("localhost","root", "", "hospitalcentral");
if ($mysqli->connect_errno)
{
    echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->
}
$consulta="SELECT * FROM pacientes  WHERE dni='$dni' ";
$resultado=$mysqli->query($consulta);
$row_cnt = $resultado->num_rows;
if($row_cnt > 0)
```

4. **A02:2021-Cryptographic Failures**

Entiendo que tanto el DNI como la información médica mostrada son datos confidenciales que deberían gozar de una función de cifrado a la hora de transmitirse y almacenarse. Si un usuario accede a la bbdd podría leer toda la información no protegida. El valor del DNI se queda almacenado en el $_POST, no se borra ni está cifrado, pudiendo ser accesible para un atacante.

5. **A04:2021-Insecure Design**

Según la documentación, el diseño seguro es "una cultura y una metodología que evalúa constantemente las amenazas y garantiza que el código esté diseñado y probado de manera sólida para evitar métodos de ataque conocidos**".**

Dicha cultura no está implementada en este código ya, como estamos viendo, hay múltiples fallas en la seguridad que no se han tenido en cuenta ni probado.

6. **A09:2021-Security Logging and Monitoring Failures**

No existe un registro de eventos (por ejemplo,fallas de inicio de sesión o acceso al server) que pueda ser evaluado en busca de cuentas maliciosas o sospechosas.