**Escuela Politécnica Nacional**

**Desarrollo de Software Seguro**

**Examen bimestral**



* Grupo: 2

**Curso:** GR2SW

**Informe análisis estático**

Febrero 2025

Contenido

[Estudiante asignado 3](#_Toc189939610)

[Resumen Ejecutivo 3](#_Toc189939611)

[Código de identificación del hallazgo 3](#_Toc189939612)

[1. Operaciones de monto 3](#_Toc189939613)

[2. Autenticación Endpoints 4](#_Toc189939614)

[3. Token-Required Decorador 5](#_Toc189939615)

[4. Banking Operation Endpoints 5](#_Toc189939616)

[Descripción de la vulnerabilidad 7](#_Toc189939617)

[Severidad 8](#_Toc189939618)

[Recomendación / Mecanismo de mitigación 13](#_Toc189939619)

## Estudiante asignado

|  |  |  |  |
| --- | --- | --- | --- |
| **Empresa** | **Nombre** | **Cargo** | **Contacto** |
| EPN | Rommel Merchán | ISA | [rommel.merchan@epn.edu.ec](mailto:rommel.merchan@epn.edu.ec) |

## Resumen Ejecutivo

Este informe detalla los hallazgos de seguridad identificados en una auditoría de software sobre un sistema de operaciones bancarias. Se han detectado múltiples vulnerabilidades que afectan la integridad, confidencialidad y disponibilidad del sistema, comprometiendo tanto la seguridad de las transacciones como la protección de la información sensible de los usuarios.

Para mitigar estos riesgos, se recomienda implementar prácticas de seguridad como el uso de ORM para evitar inyecciones SQL, el cifrado de datos sensibles, la validación estricta de entradas de usuario, el manejo seguro de sesiones y tokens, y el uso de mecanismos de concurrencia seguros en operaciones financieras.

## Código de identificación del hallazgo

### Operaciones de monto

|  |
| --- |
| **BP-16**: Mala práctica de envío de datos en text claro, se obtiene el string y float en texto claro. |

### Autenticación Endpoints

|  |
| --- |
| **BP-1**: Mala práctica identificada en la línea número 89, 94 y 115, uso de raw querys.Interfaz de usuario gráfica, Texto  El contenido generado por IA puede ser incorrecto. |
| **BP-2**: Falta de cifrado en el login |
| **BP-3**: Falta de tiempo de expiración de tokens. |

### Token-Required Decorador

|  |
| --- |
| **BP-4:** Inserción de los tokens en texto plano en los logs. |
| **BP-5**: Exposición de información sensible en el token del usuario. |
|  |

### Banking Operation Endpoints

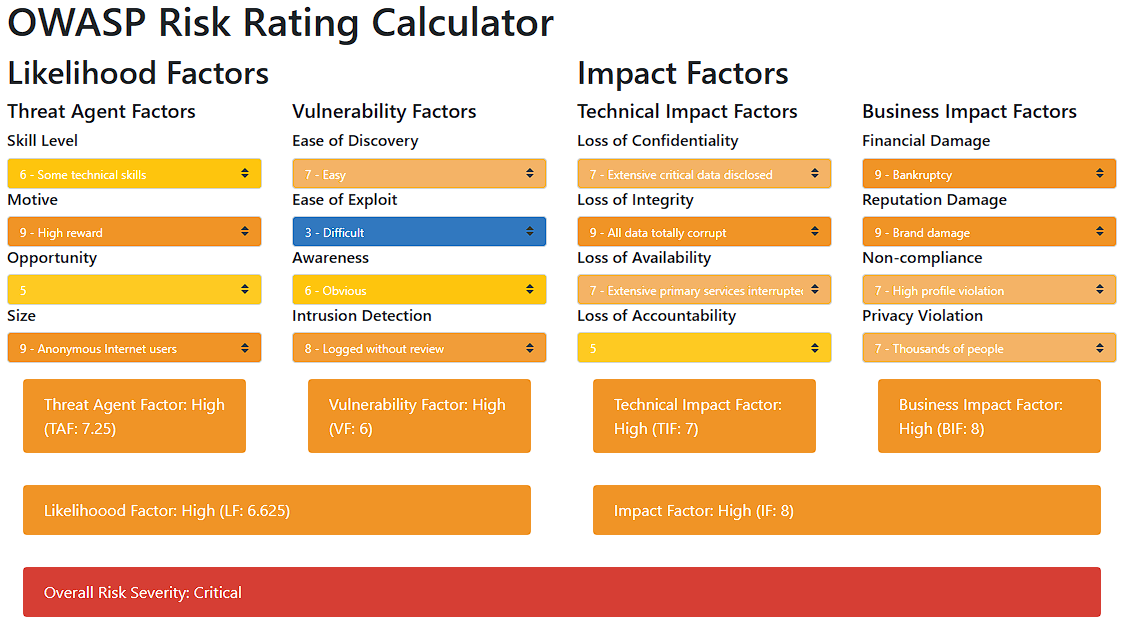
|  |
| --- |
| **BP-6**: Falta de verificación del token entre la pertenecía del token de usuario y el de la cuenta. |
| **BP-7**: If sin ningún tipo de control, comparación entre current\_balance y amount |
| **BP-8:** Posible enumeración de usuarios al tratar de realizar una transferencia |
| **BP-9:** Condición de carrera, posible generación de inconsistencia en la base de datos. |
| **BP-10:** Exposición de detalles (error 500) |
| **BP-11**: Condición de carrera posible generación de inconsistencia en la base de datos |
| **BP-12:** Exposición de detalles (error 500) |
| **BP-13:** Condición de carrera posible generación de inconsistencia en la base de datos |
| **BP-14:** Exposición de detalles (error 500) |
| **BP-15:** Falta de validación del campo amount. |

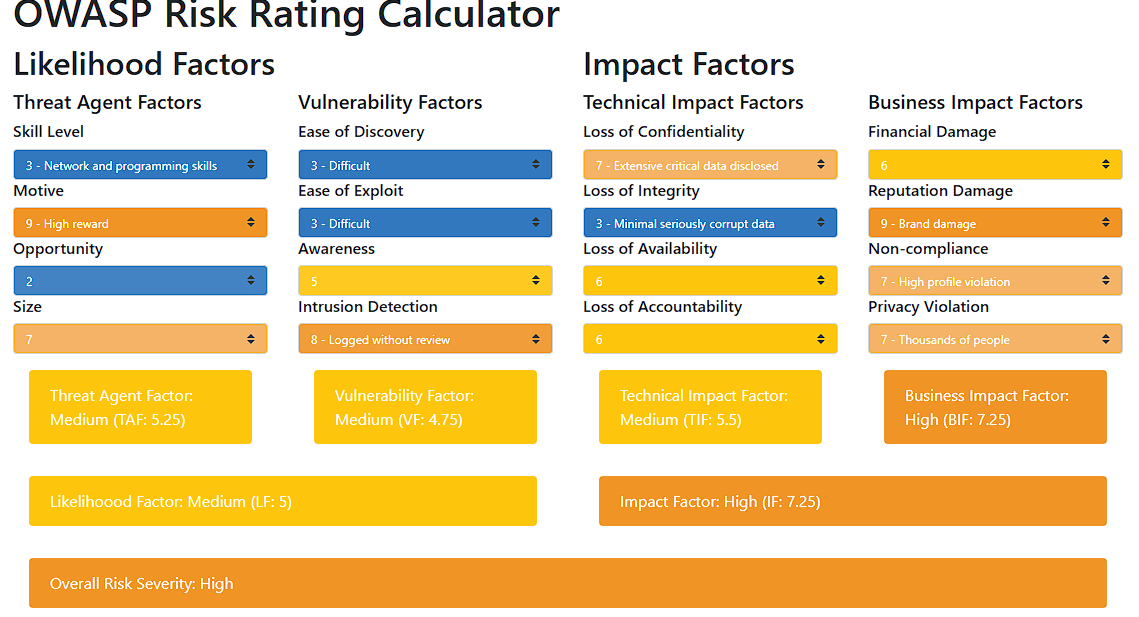
## Descripción de la vulnerabilidad

|  |  |
| --- | --- |
| ID de Vulnerabilidad | Descripción |
| **BP-1** | Uso de raw querys para hacer una consultas y acciones directamente a la base de datos, mostrando información sensible como la tabla , las columnas y debido a que el campo de entrada no esta correctamente sanitizado es susceptible a inyección de caracteres especiales y desbordamiento. |
| **BP-2** | Comparación de la contraseña en texto plano, es decir, el usuario coloca su contraseña y esta se compara tal cual (texto-plano) con la base de datos, violando el principio de confidencialidad. |
| **BP-3** | Falta de tiempo de expiración de tokens, los tokens se guardar dentro de la base de únicamente con su valor, no obstante, no tiene un TTL (Time-To-Life), si ese token no se borra por algún error o robo de token, el token será valido por siempre. |
| **BP-4** | Inserción de los tokens en texto plano en los logs, generando exposición de información sensible, robo de token, modificación de tokens y eliminación de token en específicos, no cumple los principios de integridad y disponibilidad. |
| **BP-5** | Exposición de información sensible en el token del usuario, devuelve información excesiva, innecesaria y privilegiada como puede ser el rol y el email, violando el principio de confidencialidad. |
| **BP-6** | Falta de verificación entre el id del usuario y el de la cuenta, es decir, primero se tiene que ser una verificación entre la cuenta y el usuario para cualquier operación, se puede usar el id de otra cuenta para realizar dicha transacción, faltando al principio de Autenticidad. |
| **BP-7** | If sin ningún tipo de control, comparación entre current\_balance y amount, no existe try y catch en operación de comparación dentro del código, provocando fallos inesperados, provocando exposición de información. |
| **BP-8** | Posible enumeración de usuarios al tratar de realizar una transferencia, el usuario puede intentar las veces que quiere una transferencia con cuentas que el cree que existen, no obstante, puede enumerar usuarios que existen dentro del banco. |
| **BP-9, BP-11, BP-13** | Condición de carrera, posible generación de inconsistencia en la base de datos, existen dos update simultáneos, si se intenta dos o mas transacciones simultaneas, generará una posible inconsistencia. |
| **BP-10, BP-12, BP-14** | Exposición de detalles (error 500), el código 500 generará un error que expondrá detalles del código y arquitectura de la aplicación. |
| **BP-15** | Falta de validación del campo amount, el campo no está sanitizado, es posible insertar string o número excesivamente grandes, lo cual generará un error no controlado debido a la ausencia de try y catch que puedan obtener el error y controlarlo. |
| **BP-16** | En el contexto de banco, cada dato tiende a ser confidencial por tal motivo la mala práctica de envío de datos en text claro, se obtiene el string y float en texto claro y la falta de sanitización de dichos campos genera un ambiente inseguro para los usuarios. |

## Severidad

La severidad será calculada usando la herramienta [OWSAP RICK RATING CALCULATOR](https://owasp-risk-rating.com/)

* **BP-1:** **Crítico**  
  
* **BP-2: High**



* **BP-3: High**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

* **BP-4: Medium**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

* **BP-5: Crítico**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

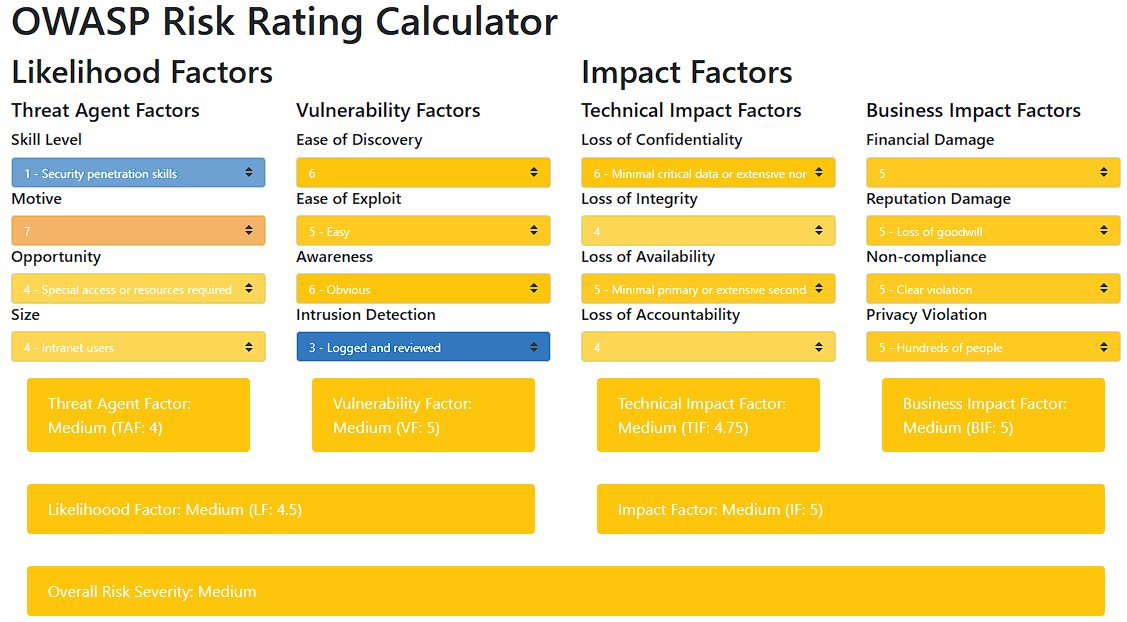
* **BP-6: Medium**Interfaz de usuario gráfica, Aplicación

  El contenido generado por IA puede ser incorrecto.
* **BP-7 Medium**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

* **BP-8: Medium**



* **BP-9, BP-11, BP-13: High**

Interfaz de usuario gráfica, Aplicación

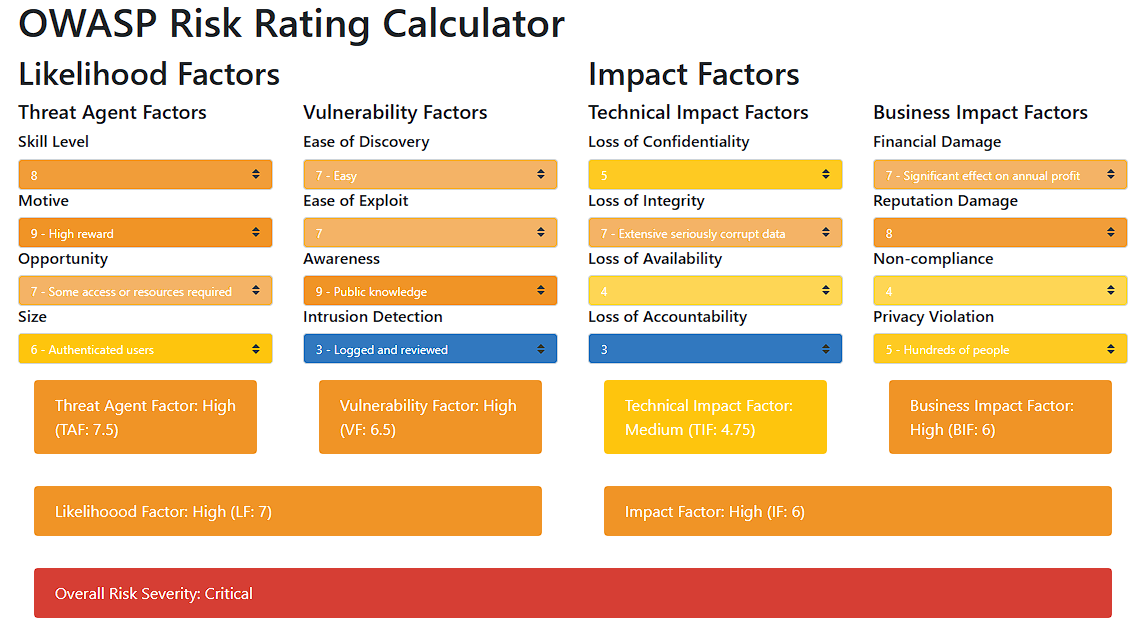
El contenido generado por IA puede ser incorrecto.

* **BP-10, BP-12, BP-14: Medium**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

* **BP-15:**



* **BP-16:**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Recomendación / Mecanismo de mitigación

* BP-1: Uso de raw querys

Utilizar ORM (Object-Relational Mapping) para interactuar con la base de datos en lugar de consultas directas. Implementar consultas preparadas para prevenir inyecciones SQL.

Validar y sanitizar las entradas de los usuarios antes de procesarlas.

* BP-2: Comparación de la contraseña en texto plano

Implementar un algoritmo de hashing seguro como Argon2, bcrypt o PBKDF2 para almacenar contraseñas. Comparar contraseñas utilizando funciones seguras como password\_verify() en PHP o hashlib.compare\_digest() en Python.

Evitar el almacenamiento de contraseñas en texto plano.

* BP-3: Falta de expiración de tokens

Establecer un tiempo de vida (TTL) para los tokens.

Implementar una lista de revocación de tokens.

Usar JWT con claims de expiración (exp).

* BP-4: Inserción de tokens en logs

Evitar registrar tokens de autenticación en logs.

Aplicar técnicas de enmascaramiento o anonimización en logs si es necesario almacenar información relacionada.

* BP-5: Exposición de información sensible en el token

Limitar la información contenida en el token (por ejemplo, no incluir roles ni emails).

Usar claims personalizados solo si son estrictamente necesarios.

* BP-6: Falta de verificación entre el ID del usuario y la cuenta

Implementar validaciones adicionales en el backend para verificar que el ID del usuario coincida con la cuenta en la que intenta operar.

Aplicar autenticación basada en roles (RBAC).

* BP-7: If sin control en la comparación de balances

Implementar estructuras de control robustas con try/catch.

Validar que el saldo sea suficiente antes de ejecutar operaciones.

* BP-8: Posible enumeración de usuarios en transferencias

Implementar respuestas genéricas en caso de error para evitar la enumeración de usuarios.

Limitar la cantidad de intentos de transferencia por IP o usuario.

* BP-9, BP-11, BP-13: Condición de carrera en transacciones

Implementar bloqueo de registros a nivel de base de datos (SELECT ... FOR UPDATE).

Usar mecanismos de transacción (ACID) para garantizar la consistencia.

* BP-10, BP-12, BP-14: Exposición de detalles en errores 500

Configurar mensajes de error genéricos para usuarios finales.

Registrar detalles del error en logs internos sin exponer información sensible.

* BP-15: Falta de validación del campo amount

Implementar validaciones en el backend para asegurar que el campo solo acepte valores numéricos dentro de un rango permitido.

Usar validaciones en el frontend antes de enviar datos al servidor.

* BP-16: Envío de datos en texto claro

Implementar cifrado TLS en todas las comunicaciones.

Evitar el almacenamiento de datos sensibles en texto plano.

Aplicar cifrado en la capa de almacenamiento cuando sea necesario.