

Inspections/

Quality Issues:

1. “Vulnerabilidad de Seguridad: Evitar construcción de consultas de base de datos desde datos controlados por el usuario”.

The screenshot shows the SonarQube interface with the following details:

- Filters:**
 - Clean Code Attribute:** Consistency (0), Intentionality (2), Adaptability (0), Responsibility (1).
 - Software Quality:** Security (3), Reliability (21), Maintainability (35).
 - Severity:** High (3), Medium (0), Low (0).
- Issues List:**
 - Issue 1:** Responsibility. Make sure this MongoDB database password gets changed and removed from the code. (30min effort, 2 months ago).
 - Issue 2:** Intentionality. Change this code to not construct database queries directly from user-controlled data. (30min effort, 1 month ago).
 - Issue 3:** Intentionality. Change this code to not construct database queries directly from user-controlled data. (30min effort, 1 month ago).

The screenshot shows the detailed view of a quality issue with the following details:

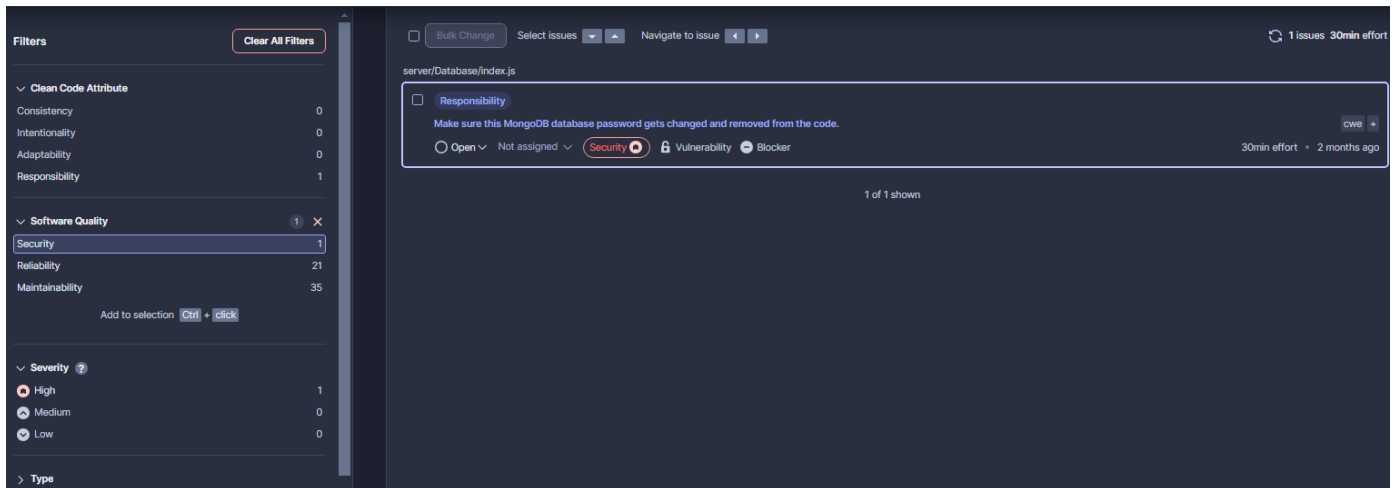
- Issue Details:**
 - Issue 1:** Intentionality | Not complete. Change this code to not construct database queries directly from user-controlled data. (30min effort, 1 month ago).
 - Description:** NoSQL operations should not be vulnerable to injection attacks [jsecuity:SS147](#).
 - Software qualities impacted:** Security.
 - Tags:** CWE, L40.
 - Line affected:** L40.
 - Effort:** 30 min.
 - Introduced:** 1 month ago.
- Execution Flow:**
 - 2 steps execution flow:**
 - SOURCE:** a user can craft an HTTP request with malicious content.
 - SINK:** this invocation is not safe; a malicious value can be used as argument.
- Code Snippet:**

```
server/Routes/user.js
27 piperson.. if (dv == 11) return '0';
28 if (dv == 10) return 'K';
29 return dv.toString();
30 };
31
32 piperson.. router.post('/register', async (req, res) => {
33   try {
34     const { name, rut, password, cargo } = req.body;
35
36     if (!validateRut(rut) || rut.length < 11) {
37       return res.status(401).json({ message: 'Invalid RUT format' });
38     }
39
40     const existingUser = await User.findOne({ rut });
41
42     if (existingUser) {
43       return res.status(401).json({ message: 'User already exists' });
44     }
45
46     const hashedPassword = await bcrypt.hash(password, 10);
47     const newUser = new User({ name, rut, password: hashedPassword, cargo });
48     await newUser.save();
49   } catch (err) {
50     return res.status(500).json({ message: 'Internal server error' });
51   }
52 });
```

- Descripción del Error: El código construye una consulta usando datos controlados directamente por el usuario lo que puede provocar posibles inyecciones además de comprometer la seguridad del software, lo cual es una vulnerabilidad de seguridad crítica que debe ser solucionada de inmediato.
- Manera de Abordar: La consulta fue parametrizada en una variable llamada query, la cual almacena los datos y la convierte a string brindando así solución a la vulnerabilidad.

```
let query = { rut: rut.toString() }  
const existingUser = await User.findOne(query);
```

- Resultados: Los errores ya no figuran en el servicio Sonarqube.



2. “La sentencia “If” no debe ser la única sentencia en el bloque “else””.

The screenshot displays the SonarQube web interface. On the left, a sidebar lists various issues, with the selected issue being "If statement should not be the only statement in 'else' block". The main panel shows the code for `client/src/components/RegisterPage.js`. The issue is highlighted at line 32, where an `if` statement is the only statement within an `else` block. The interface includes a top navigation bar with tabs for "Where is the issue?", "Why is this an issue?", "Activity", and "More info". The right sidebar provides metadata about the issue, including its tags, line affected (L32), effort (5 min), and introduction date (1 month ago).

```
1  pipexson... import React, { useState, useEffect } from 'react';
2  pipexson... import { Link } from 'react-router-dom';
3  pipexson... import axios from 'axios';
4  pipexson... import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
5  pipexson... import { faUser, faKey } from '@fortawesome/free-solid-svg-icons';
6  pipexson... import { faIdBadge } from '@fortawesome/free-solid-svg-icons';
7
8  const RegisterPage = () => {
9    const [name, setName] = useState('');
10   const [rut, setRut] = useState('');
11   const [dv, setDv] = useState('');
12   const [password, setPassword] = useState('');
13   const [confirmPassword, setConfirmPassword] = useState('');
14   const [cargo, setCargo] = useState('');
15
16   const registerUser = async (userData) => {
17     try {
18       const response = await axios.post('http://localhost:5000/user/register', userData);
19       return response.data;
20     } catch (error) {
21       console.log(error);
22     }
23   };
24
25   const handleRegister = async (e) => {
26     e.preventDefault();
27     if (name === '' || rut === '' || dv === '' || password === '' || confirmPassword === '' || cargo === '') {
28       window.alert('Por favor rellena todos los campos');
29     } else if (password !== confirmPassword) {
30       window.alert('Las contraseñas no son iguales');
31     } else {
32       if (password === confirmPassword) {
33         const response = await registerUser({ name, rut: `${rut}-${dv}`, password, cargo });
```

- Descripción del Error: Dentro de un bloque de código `else` no es una buena opción el tener únicamente una sentencia `if` ya que esto genera un impacto en la mantenibilidad del software, en nuestro caso un impacto medio.
- Manera de abordar: Para solucionar el problema reestructuramos el código de modo que los “if” anidados dentro de “else” se convirtieran en “else if”.

```
const handleRegister = async (e) => {
  e.preventDefault();
  if (name === '' || rut === '' || dv === '' || password === '' || confirmPassword === '' || cargo === '') {
    window.alert('Por favor rellena todos los campos');
  } else if (password !== confirmPassword) {
    window.alert('Las contraseñas no son iguales');
  } else {
    try {
      const response = await registerUser({ name, rut: `${rut}-${dv}`, password, cargo });
```

- Resultados: Los errores ya no figuran en el servicio Sonarqube.

Filters

Clear All Filters

Clean Code Attribute

Consistency0

Intentionality1

Adaptability0

Responsibility0

Software Quality

Security0

Reliability0

Maintainability1

Severity ?

1 X

High1

Medium1

Low32

Add to selectionCtrl + Click

Type

Resolution

Status

Security Category

Creation Date

Bulk Change

Select Issues

Navigate to Issue

1 issues1min effort

server/Database/index.js

Intentionality

Remove this useless assignment to variable "conn".

CWEunused

OpenNot assignedMaintainabilityCode SmellMajor

1min effort · 2 months ago

1 of 1 shown