

# Static Code Analysis

CODE ANALYSIS ON SYSTEM OF LUONG DUNG

MICHEL BIJNEN, KRISTIAN HANSEN, ALEXANDER LAMBOOIJ, ROEL  
LUCASSEN, YOURI SAMAN, THOMAS VAN SCHIJNDEL

# Introduction

This document has been set-up in order to communicate what vulnerabilities have been found within the git repository of Luong Dung.

This list of vulnerabilities was established by running Sonarqube Scanner analyses and performing manual code reviews.

The severity levels are defined using the CVSS version 3. This system calculates a severity score from 0.1 to 10.0 for every found vulnerability. Severity scores are categorized as follows:

Severity	CVSS V3 Score Range	Definition
<b>Critical</b>	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
<b>High</b>	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
<b>Moderate</b>	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
<b>Low</b>	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
<b>Informational</b>	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

# Table of contents

Introduction	1
Table of contents	2
Code analysis result	3
Entire System	3
01 - No .gitignore	3
02 - No git best practices	3
03 - Bad SOLID practices	3
User_Client	4
01 - SQL Injection	4
02 - Plain Text password	4
03 - Password policy	4
04 - Dangerous function	5
05 - No alternative attribute	5
06 - No curly brackets	6
07 - Overriding variable	7
08 - Unexpected Coercion	7
09 - Unnecessary return	8
10 - Unused Variable	8
11 - Hash data	8
12 - No content	9
13 - Redundant variable	9
14 - <img> no body	9
15 - Logic in Model	9
16 - HTTPMethod POST	10
JavaKafka	11
01 - Readability Demo	11
02 - Useless assignment to local variable	11
03 - Useless assignment to local variable	11
04 - Variable does not match naming convention	12
05 - Variable does not match naming convention	12
06 - Specific type in constructor	13
Test_machine_client	14
01 - SQL Injection	14
02 - Plaintext password	14
03 - Unexpected Coercion	14
04 - Empty return at end of function	15
05 - Business logic in model	15
06 - Using var	15
07 - Hardcode api URL	15
08 - Inconsistent folder or package names	16
Database	17
01 - Plaintext username and password	17
02 - Username equal to password	17
Conclusion	18

# Code analysis result

## Entire System

<b>Low (3.5)</b>	<b>01 - No .gitignore</b>
<b>Location</b>	Every project
<b>Problem</b>	No .gitignore is present.
<b>Recommendations</b>	Add a .gitignore file and ignore maps like .idea, node_modules, .vscode, and target.

<b>Informational</b>	<b>02 - No git best practices</b>
<b>Location</b>	Git repository
<b>Problem</b>	Unreadable repository, not according to git best practices. Multiple projects are stored in one repository. No branches are used for different user stories or tasks.
<b>Recommendations</b>	Refactor git repositories according to best practices.

<b>Informational</b>	<b>03 - Bad SOLID practices</b>
<b>Location</b>	Total code base
<b>Problem</b>	According to SOLID, classes and methods should not be long and only have a single responsibility. This is not done in these classes and thus decreases readability.
<b>Recommendations</b>	Make sure every method has a single responsibility and all 'utils' methods are divided in other classes.

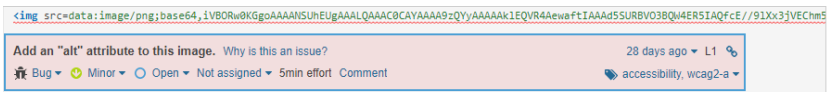
## User\_Client


<b>Critical(9.9)</b>	<b>01 - SQL Injection</b>
<b>Location</b>	User_client -> app -> models -> user.model.js -> lines 22-23/39/61/160/185/223/247/266/288/313
<b>Problem</b>	SQL injection is possible.
<b>Recommendations</b>	Use prepared statements.

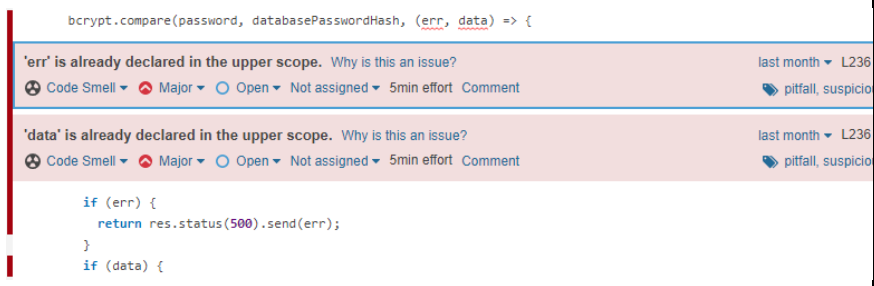
<b>Critical (9.4)</b>	<b>02 - Plain Text password</b>
<b>Location</b>	User_client -> app -> config -> db.config.js  User_client -> app -> config -> hex_db.config.js  User_client -> app -> controllers -> user.controller.js -> line 244/302
<b>Problem</b>	A hardcoded plaintext password is present.
<b>Recommendations</b>	Move the credentials to environmental variables.

<b>High (8.6)</b>	<b>03 - Password policy</b>
<b>Location</b>	User_client -> app -> controllers -> user.controller.js -> line 17  User_client -> server.js -> lines 3-4
<b>Problem</b>	Bad password policy.
<b>Recommendations</b>	Increase maximum length to at least 100 characters (maybe longer). Remove blacklist. Check whether password and username are (partially) NOT the same. Remove specific constraints (uppercase, lowercase) but do force or recommend something. This way password constraints are less predictable.

<b>High(7.2)</b>	<b>04 - Dangerous function</b>
<b>Location</b>	User_client -> app -> models -> user.model.js → line 142-155
<b>Problem</b>	Delete all users should not be in production.
<b>Recommendations</b>	Move development functions to separate projects. This way it cannot be pushed into production.

<b>Low (3.3)</b>	<b>05 - No alternative attribute</b>
<b>Location</b>	User_client -> app -> media -> qrCodeFile.html -> line 1  User_client -> app -> controllers -> user.controller.js -> line 489
<b>Problem</b>	No alternative attribute provided for an image.
<b>Recommendations</b>	Add an alternative attribute for when the image is not available.
<b>Proof</b>	

Informational	06 - No curly brackets
Location	User_Client -> app -> controllers -> user.controller.js -> lines 165/308
Problem	No curly braces or indentation after the else.
Recommendations	<p>In the absence of enclosing curly braces, the line immediately after a conditional is the one that is conditionally executed. By both convention and good practice, such lines are indented. In the absence of both curly braces and indentation the intent of the original programmer is entirely unclear and perhaps not actually what is executed. Additionally, such code is highly likely to be confusing to maintainers.add curly braces or an indentation after using else just like you would with an if statement.</p>
Proof	

Informational	07 - Overriding variable
Location	User_client -> app -> controllers -> user.controller.js -> lines 236/246/253/262/302/395/468  User_client -> app -> models -> user.model.js -> line 197
Problem	Overriding or shadowing a variable declared in an outer scope can strongly impact the readability, and therefore the maintainability, of a piece of code. Further, it could lead maintainers to introduce bugs because they think they're using one variable but are really using another.
Recommendations	<a href="https://wiki.sei.cmu.edu/confluence/display/c/DCL01-C.+Do+not+reuse+variable+names+in+subscopes">https://wiki.sei.cmu.edu/confluence/display/c/DCL01-C.+Do+not+reuse+variable+names+in+subscopes</a> - Do not reuse variable names in subscopes.  <a href="https://wiki.sei.cmu.edu/confluence/display/java/DCL51-J.+Do+not+shadow+or+obscure+identifiers+in+subscopes">https://wiki.sei.cmu.edu/confluence/display/java/DCL51-J.+Do+not+shadow+or+obscure+identifiers+in+subscopes</a> - Do not shadow or obscure identifiers in subscopes.
Proof	 <pre> bcrypt.compare(password, databasePasswordHash, (err, data) =&gt; {   'err' is already declared in the upper scope. Why is this an issue?   Code Smell Major Open Not assigned 5min effort Comment   'data' is already declared in the upper scope. Why is this an issue?   Code Smell Major Open Not assigned 5min effort Comment    if (err) {     return res.status(500).send(err);   }   if (data) { </pre>

Informational	08 - Unexpected Coercion
Location	User_client -> app -> controllers -> user.controller.js -> line 201/260/314  User_client -> app -> controllers -> user.model.js -> lines 46/68/107/128/168/296/321
Problem	Comparison <code>err.info == 'not_found'</code> may cause unexpected type coercion.
Recommendations	Replace <code>'=='</code> with <code>'==='</code> .



Informational	09 - Unnecessary return
Location	User_client -> app -> controllers -> user.controller.js -> lines 454/457/462/471/477  User_client -> app -> models -> user.model.js -> lines 49/54/71/76/130/136/147/152/171/177/201/207/215/233/239/252/258/271/277/298/305/324/329
Problem	Return is unnecessary as the last statement in a function with no return value.
Recommendations	Remove the empty return statement.

Informational	10 - Unused Variable
Location	User_client -> app -> controllers -> user.controller.js -> line 17  User_client -> server.js -> lines 3-4
Problem	An unused variable is present.
Recommendations	Remove unused variables.

Informational	11 - Hash data
Location	User_client -> app -> controllers -> user.controller.js -> lines 55-57
Problem	Email, first name, last name are being hashed with bcrypt.
Recommendations	Do not hash emails, first names, last names or basically anything except passwords, because this is data that is needed in other functions. Hashing is irreversible and makes this data practically useless.

Informational	12 - No content
Location	User_client -> app -> controllers -> user.controller.js -> line 166-168/182-183
Problem	204 means NO CONTENT. This means no content is returned.
Recommendations	Do not return a response body.

Informational	13 - Redundant variable
Location	User_client -> app -> controllers -> user.controller.js -> line 437
Problem	Local variable 'token' is redundant.
Recommendations	Remove local variable 'token'.

Informational	14 - <img> no body
Location	Location: User_client -> app -> controllers -> user.controller.js -> line 489
Problem	Html <img> tag has no body.
Recommendations	Change <img src=""></img> to <img src=""/>.

Information	15 - Logic in Model
Location	User_client -> app -> models -> user.model.js
Problem	Business logic code detected in model. Reduces readability.

<b>Recommendations</b>	Create a logic class for the logic in the model class.
------------------------	--

<b>Informational</b>	16 - HTTPMethod POST
<b>Location</b>	User_client -> app -> routes -> user.routes.js -> line 29
<b>Problem</b>	Updating one value in an object should be a PATCH, and not a POST.
<b>Recommendations</b>	Change linkTest to PATCH.

## JavaKafka

Informational	01 - Readability Demo
Location	JavaKafka
Problem	Make sure that it is clear to anyone who sees this repository knows that this project's classes are for demonstration purposes. This code can never go to production.
Recommendations	Rename JavaKafka to JavaKafkaDemo and add a README to the JavaKafka which says that it has to stay in development.

Informational	02 - Useless assignment to local variable
Location	JavaKafka → ConsumerAssignAndSeek.java → line 21
Problem	Useless assignment to local variable 'groupId'.
Recommendations	Remove the assignment to local variable 'groupId'.
Proof	<pre>String groupId = "my_second_application";</pre> <div>Remove this useless assignment to local variable "groupId". Why is this an issue? 3 months ago L21 </div> <div> Code Smell  Major  Open  Not assigned 15min effort <a href="#">Comment</a>  cert, cwe, unused </div>

Information	03 - Useless assignment to local variable
Location	JavaKafka → ProducerKeys.java → line 29
Problem	Useless assignment to local variable 'key'.
Recommendations	Remove the assignment to local variable 'key'.

<b>Proof</b>	<pre>String key = "id_" + Integer.toString(i);</pre> <p>Remove this useless assignment to local variable "key". Why is this an issue? 3 months ago ▾ L29 🔗</p> <p>🔗 Code Smell ▾ 🚫 Major ▾ 🔓 Open ▾ Not assigned ▾ 15min effort Comment 🔗 cert, cwe, unused ▾</p>
--------------	---

<b>Information</b>	04 - Variable does not match naming convention
<b>Location</b>	JavaKafka→ ConsumerAssignAndSeek.java → line 44
<b>Problem</b>	Local variable does not match the naming convention.
<b>Recommendations</b>	Rename 'NoOfMessageToRead' to 'noOfMessageToRead'.
<b>Proof</b>	<pre>int NoOfMessageToRead = 5;</pre> <p>Rename this local variable to match the regular expression '^a-z[a-zA-Z0-9]*\$'. Why is this an issue? 3 months ago ▾ L44 🔗</p> <p>🔗 Code Smell ▾ 🟡 Minor ▾ 🔓 Open ▾ Not assigned ▾ 2min effort Comment 🔗 convention ▾</p>

<b>Information</b>	05 - Variable does not match naming convention
<b>Location</b>	JavaKafka→ ConsumerAssignAndSeek.java → line 46
<b>Problem</b>	Local variable does not match the naming convention.
<b>Recommendations</b>	Rename 'NoOfMessageReadSoFar' to 'noOfMessageReadSoFar'
<b>Proof</b>	<pre>int NoOfMessageReadSoFar = 0;</pre> <p>Rename this local variable to match the regular expression '^a-z[a-zA-Z0-9]*\$'. Why is this an issue? 3 months ago ▾ L46 🔗</p> <p>🔗 Code Smell ▾ 🟡 Minor ▾ 🔓 Open ▾ Not assigned ▾ 2min effort Comment 🔗 convention ▾</p>

Informational	06 - Specific type in constructor
Location	JavaKafka → ConsumerDemo.java → line 30
Problem	Type specification in this constructor can be less verbose.
Recommendations	Replace the type specification in this constructor call with the diamond operator.
Proof	<pre>KafkaConsumer&lt;String, String&gt; consumer = new KafkaConsumer&lt;String, String&gt;(properties);</pre> <p>Replace the type specification in this constructor call with the diamond operator ("&lt;&gt;").  <small>(sonar.java.source not set. Assuming 7 or greater.) Why is this an issue?</small> 3 months ago ▾ L30 🔗</p> <p>🔗 Code Smell ▾ 🟡 Minor ▾ 🔵 Open ▾ Not assigned ▾ 1min effort Comment 🧑‍🔧 clumsy ▾</p>

## Test\_machine\_client

<b>Critical(9.9)</b>	<b>01 - SQL Injection</b>
<b>Location</b>	Test_machine_client -> Models -> model.test_machine.js -> line 17
<b>Problem</b>	SQL Injection possible.
<b>Recommendations</b>	Use prepared statements.

<b>Critical(9.4)</b>	<b>02 - Plaintext password</b>
<b>Location</b>	Test_machine_client -> config -> db.config.js
<b>Problem</b>	A hardcoded plaintext password is present.
<b>Recommendations</b>	Move the credentials to environmental variables.

<b>Informational</b>	<b>03 - Unexpected Coercion</b>
<b>Location</b>	Test_machine_client -> Controllers -> controller.test_machine.js -> line 10  Test_machine_client -> Models -> model.test_machine.js -> line 25
<b>Problem</b>	Comparison err.info == 'not_found' may cause unexpected type coercion.
<b>Recommendations</b>	Replace '==' with '==='.

Informational	04 - Empty return at end of function
Location	<p>Test_machine_client -&gt; Controllers -&gt; controller.test_machine.js -&gt; line 36</p> <p>Test_machine_client -&gt; Models -&gt; model.test_machine.js -&gt; lines 28/35</p> <p>Test_machine_client -&gt; test_machine.js -&gt; line 13</p>
Problem	Ending a function with an empty return statement.
Recommendations	Remove the empty return statement.

Informational	05 - Business logic in model
Location	Test_machine_client -> Models -> model.test_machine.js
Problem	Business logic code detected in model. Reduces readability.
Recommendations	Create a logic class for the logic in the model.test_machine.js.

Informational	06 - Using var
Location	Test_machine_client -> servers -> server.js -> line 3
Problem	Usage of var.
Recommendations	Use const or let instead of var.

Informational	07 - Hardcode api URL
Location	Test_machine_client -> servers -> server_test_machine.js -> line 8
Problem	Hardcoded api URL.
Recommendations	Move hardcoded URLs to environmental variables or config files.



<b>Informational</b>	08 - Inconsistent folder or package names
<b>Location</b>	Test_machine_client
<b>Problem</b>	Inconsistency naming folders or packages.
<b>Recommendations</b>	Rename the folders or packages.

## Database

<b>Critical (9.4)</b>	<b>01 - Plaintext username and password</b>
<b>Location</b>	Database -> DatabaseCredentials.txt
<b>Problem</b>	Username and password should not be plaintext in the repository.
<b>Recommendations</b>	Move this file to a file on your computer that does not have a connection to the outside world. Also work in a development environment with a local hosted database, so you have a different production database.

<b>High(7.6)</b>	<b>02 - Username equal to password</b>
<b>Location</b>	Database -> DatabaseCredentials.txt
<b>Problem</b>	Username is equal to the password.
<b>Recommendations</b>	Change password to random generated string of at least 20 characters (lowercase letters, uppercase letters, numbers and characters).

# Conclusion

Because of all the vulnerabilities that are found in this project, the system is far from ready for production. These vulnerabilities need to be solved and the developer(s) need to keep in mind to not make the same mistakes as mentioned in this document.

Because there are a lot of code smells (category “Informational”), the code becomes almost unreadable. In some cases, this also creates a security risk.