Part 2: SonarCloud – Static Analysis

This part of the assignment is about getting familiar with the static analysis tool ***SonarCloud****.* [1]

For this part of the assignment you can choose to work with these three alternatives:

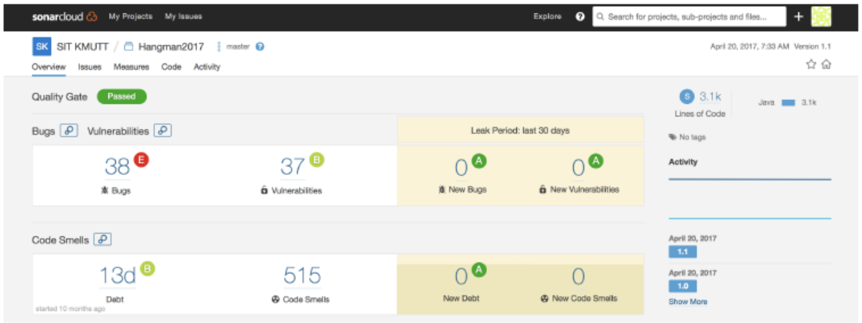
Option 1:
Search and select an already scanned project called "*Hangman2017 by SIT KMUTT*", and solve the following tasks based on the projects results shown **here.**
DIFFICULTY: EASY

Option 2:
Select any another project of your own choosing. Search from SonarCloud any open source project available. (e.g. "*sudoku-solver*").
DIFFICULTY: NORMAL

Option 3:
Run an analysis of your own project/code (either Java, C#, C, C++, Javascript, Python etc.) using the instructions here, and use those results to solve the following tasks.
DIFFICULTY: HARD

OBS: If you choose **Option 3**, SonarCloud requires signing up with a GitHub/Bitbucket account, and it only permits you to scan open source projects (i.e. public project(s)).

Regardless of the chosen option, you should be able to see the following page:


sonarcloud
My Projects
My Issues
Explore
Search for projects, sub-projects and files...
SK SIT KMUTT / Hangman2017 master
Overview
Issues
Measures
Code
Activity

April 20, 2017, 7:33 AM Version 1.1

Quality Gate Passed

Bugs Vulnerabilities

Leak Period: last 30 days

38 E Bugs

37 B Vulnerabilities

0 A New Bugs

0 A New Vulnerabilities

Code Smells

13d B Debt

started 10 months ago

515 Code Smells

0 A New Debt

0 New Code Smells

3.1k Lines of Code

Java 3.1k

No tags

Activity

April 20, 2017 1.1

April 20, 2017 1.0

Show More

Figure 1: SonarCloud Project Overview aka. Dashboard

[1] SonarCube/SonarCloud Documentation – `https://docs.sonarqube.org/display/SONAR/User+Guide`

Task 0.

0.1 **Specify** which of the following options you have chosen. (e.g. *Option 1: "Hangman2017"*)

0.2 If you choose *Option 3*; **specify** the GitHub/Bitbucket URL.

Task 1.

1.1 What does the metrics say about the code/project in whole (e.g. quality gate)?

1.2 **Briefly explain** what the types of issues listed by the analyzer mean?

- 1.2.1 Bugs?
- 1.2.2 Vulnerabilities?
- 1.2.3 Code Smells?
- 1.2.4 Blocker?
- 1.2.5 Coverage?

1.3 For the given project selected, **name the folder(s)** containing the highest number of LOC (Lines of code), and for the given folder?

- 1.3.1 How many bugs are there?
- 1.3.2 How many vulnerabilities?
- 1.3.3 How many code smells?

Hint: *Go to the "Code" tab to see the complete list of files and subfiles in the project.*

Task 2.

2.1 **Select and list** at least (four) 4 issues (e.g. bugs/vulnerability/code smells/. . .), and **briefly explain why** the analyzer has reported the given issues?

2.2 For each of the listed issues, **how would you solve** the problem?

Hint: *Use either textual description, or code/ pseudocode to describe the treatment.*

Task 3.

3.1 **Briefly explain** the following concepts:

- 3.1.1 False-positive (aka. False-fail result)
- 3.1.2 False-negative (aka. False-pass result)

3.2 **Provide** an example of each of the concepts.