Software Security (502804-3)
Spring 2020
Due: Saturday March 26, 2022, 11:59 pm via Blackboard
SonarQube Lab

1. Overview

The learning objective of this lab is for students to understand how open-source platforms, namely SonarQube, perform automatic reviews with static analysis of code. SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities on 20+ programming languages. SonarQube offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security vulnerabilities.

2. Lab Tasks

This lab covers the following three main tasks.

2.1 Task 1: Installing SonarQube

In this task, we aim to teach the students how to install SonarQube on their personal computers.

Get Started in Two Minutes Guide

Installing from a zip file

1. Download the SonarQube Community Edition.
2. As a **non-root user**, unzip it, let's say in *C:\sonarqube* or */opt/sonarqube*.
3. As a **non-root user**, start the SonarQube Server:

```
# On Windows, execute:
C:\sonarqube\bin\windows-x86-xx\StartSonar.bat

# On other operating systems, as a non-root user execute:
```

```
/opt/sonarqube/bin/[OS]/sonar.sh console
```

If your instance fails to start, check your logs to find the cause.

4. Log in to http://localhost:9000 with System Administrator credentials (login=admin, password=admin).
5. Click the **Create new project** button to analyze your first project.

Using Docker

Images of the Community, Developer, and Enterprise Editions are available on Docker Hub.

1. Start the server by running:

```
$ docker run -d --name sonarqube -p 9000:9000 <image_name>
```

2. Log in to http://localhost:9000 with System Administrator credentials (login=admin, password=admin).
3. Click the **Create new project** button to analyze your first project.

Deliverable: Attach a screenshot of the SonarQube installed on your machine.

2.2 Task 2: Vulnerable Programming · SonarQube

In this task, we aim to teach the students how to use SonarQube to analyze a simple C++ code.

Step 1: Write a simple C++ code. Note you may read online and find different code vulnerabilities.

Step 2: Use SonarQube to evaluate the C++ code you wrote. Explain the code vulnerabilities you have found.

Deliverable:

1. **Send me your "vulnerable" C++ code.**
2. **Screenshot your SonarQube analysis.**

2.3 Task 3: Secure Programming · SonarQube

In this task, we aim to teach the students how to use SonarQube to evaluate a simple C++ code.

Step 1: Make your code secure by eliminating the defined vulnerabilities in the previous task.

Step 2: Use SonarQube again to make sure that your C++ code is now secure and no vulnerabilities.

Deliverable:

1. **Send me your "secure" C++ code.**
2. **Screenshot your SonarQube analysis.**