Client request: We need a security scan report of chatbot platform covering the Top-10 Owasp security vulnerabilities.

Top 10 vulnerabilities

<https://owasp.org/www-project-top-ten/>

1. Broken access control. ...
2. Cryptographic failures. ...
3. Injection (or [SQL injections](https://www.spiceworks.com/it-security/application-security/articles/what-is-sql-injection/))
4. Insecure design. ...
5. Security misconfigurations. ...
6. Vulnerable and outdated components. ...
7. Identification and authentication failures. ...
8. Software and data integrity failures
9. Security logging and monitoring failures
10. Server-side request forgery (SSRF)

SonarQube:

static code analysis is a collection of algorithms and techniques used to analyze source code in order to automatically find potential errors or poor coding practices. The idea is similar in spirit to compiler warnings (which can be useful for finding coding errors), but to take that idea a step further and find bugs that are traditionally found using run-time debugging techniques such as testing.

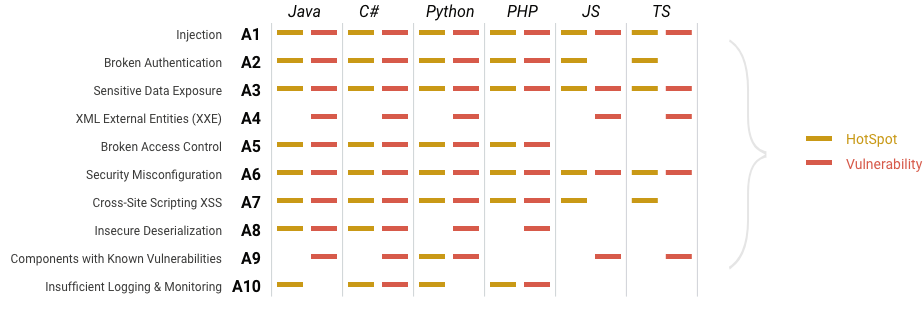
Static code analysis, also commonly called "white-box" testing, looks at applications in non-runtime environments. It is the only proven method to cover the entire code base and identify all the vulnerable patterns. Static code analysis is also considered as a way to automate code review process.

**Main Focus : Security Vulnerabilities**

**SonarQube**

SonarQube collects and analyzes source code, measuring quality and providing reports for your projects. It combines static and dynamic analysis tools and enables quality to be measured continuously over time.  Everything that affects our code base, from minor styling details to critical design errors, is inspected and evaluated by SonarQube, thereby enabling developers to access and track code analysis data ranging from styling errors, potential bugs, and code defects to design inefficiencies, code duplication, lack of test coverage, and excess complexity.

<https://www.sonarqube.org/features/security/owasp/>



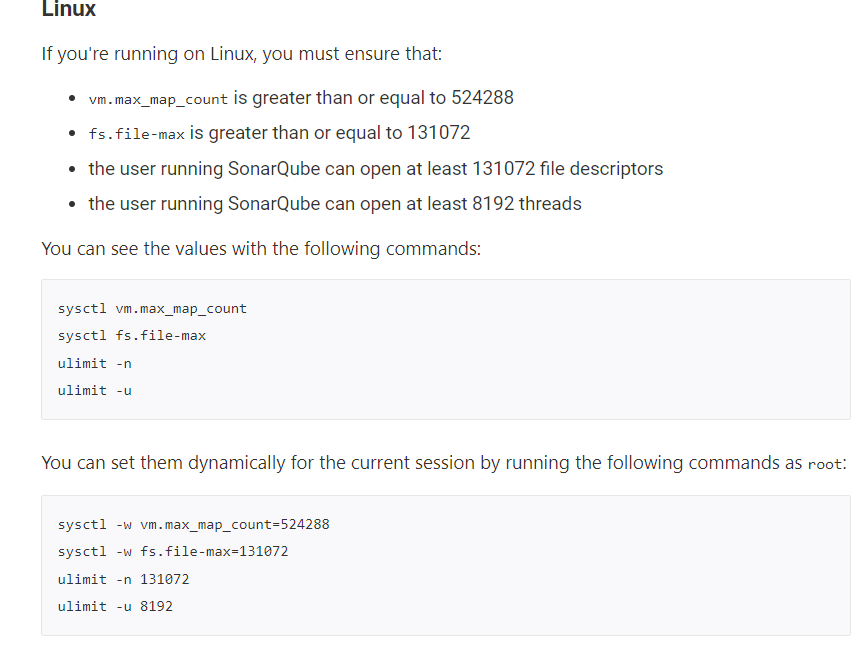
SonarQube pre-request:

<https://docs.sonarqube.org/latest/requirements/requirements/>

Runtime environment JAVA 11 or 17

One of following database:

1. Postgres
2. Oracle
3. Microsoft SQL server



SonarQube code scanner:

<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>

## Configuring your project

Create a configuration file in your project's root directory called sonar-project.properties

#Configure here general information about the environment, such as SonarQube server connection details for example

#No information about specific project should appear here

#----- Default SonarQube server

#sonar.host.url=http://localhost:9000

#----- Default source code encoding

#sonar.sourceEncoding=UTF-8

#----- Default SonarQube server

#sonar.host.url=http://localhost:9000

#sonar.login=sqp\_7dc687bc13391cb886ae789876f02b25577f9b81

#sonar.projectKey=test

#sonar.projectName=test

#sonar.projectVersion=1.0

#sonar.sources=C:\\Users\\GOWTHAM\\Desktop\\app\\

#sonar.language=py

#sonar.sourceEncoding=UTF-8

#Sonar.scm.disabled=True