LARGEAU Brandon

Why you should consider using static code analysis

A VET (VEille Technologique) report

2nd Year of Master's degree in computer science

October 22, 2020

1 Introduction

As a developer you may have notice that the world is going faster and faster everyday. This acceleration translates into the fact that when we develop an application, we can notice a certain pressure regarding deadlines. This pressure can lead us to make more and more errors when coding. When we talk about errors, we are talking about professional errors that can have important consequences in the echo-system of an application. Those errors can affect several areas of the application, security, performance, maintainability, etc. However, there are solutions to help solve those kinds of problem. We will see here that there are specialized tools for this.

A static code analysis tool is here to ensure that you don't make mistakes. Static because it is analyzing your code statically which mean that you don't need to run your application to find problems compared to dynamic code analysis tool where you need to your application. If you are not using this type of tool yet, we will show you why it is important to start right now.

Talking about rules, there is an entity called **C**ommon **W**eakness **E**numeration which is here "To improve the quality of software with respect to known security issues within source code"[1]. When the tool detects that there's a violation of one of the rules of the CWE[1], there is a link to this rules explaining what is going wrong.

Another advantage of static code analysis tools is that they allow errors to be detected at the early stage of the development of the application, which means that there is less cost on fixing them since the part of the code that is injured has less dependency on another part of the code.

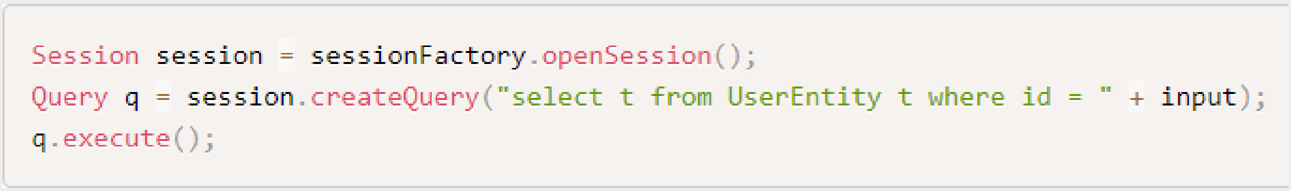
2 Security

When developing an application or a website, you need to ensure the security because you don't want private information to be leak by a security vulnerability. It is hard to know every security issue, this is why you have to follow some rules. One important rule when dealing with database is that you have to sanitize any SQL requests. To help you with this kind of vulnerability,

1 Common Weakness Enumeration

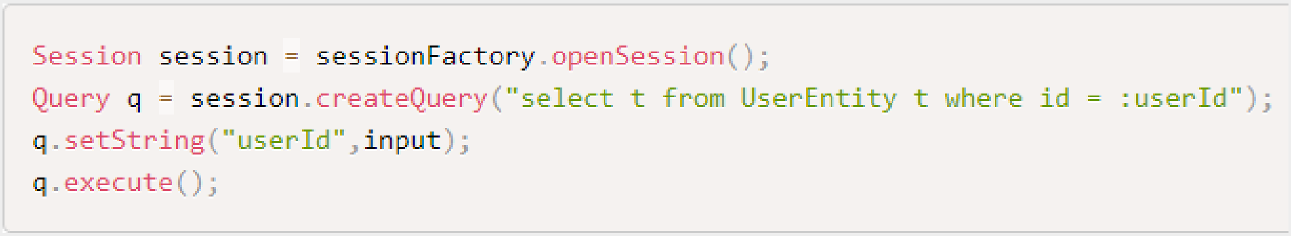
there are dedicated tools such as Find Security Bugs[2] which is a free plugin that can interface into your favorite IDE. It detect up to 135 bug patterns regarding Java web applications and give you all the needed information to fix the problem.

An example of such vulnerability when using Hibernate[3] is the “CWE-564: SQL Injection: Hibernate”[2] vulnerability stating that “SQL statement built with user-controlled input” can be problematic. A good way to fix this is to use bind variables.



```java
Session session = sessionFactory.openSession();
Query q = session.createQuery("select t from UserEntity t where id = " + input);
q.execute();
```

Vulnerable code (https://find-sec-bugs.github.io/bugs.htm#SQL_INJECTION_HIBERNATE)



```java
Session session = sessionFactory.openSession();
Query q = session.createQuery("select t from UserEntity t where id = :userId");
q.setString("userId",input);
q.execute();
```

Solution with a bind variable
(https://find-sec-bugs.github.io/bugs.htm#SQL_INJECTION_HIBERNATE)

It is important to follow rules and use such tools that have built-in bug pattern checker.

3 Coding Standards

A coding standard is there to put everyone on the same level regarding the best practices of programming when creating a collaborative project and generally in a community. It is a set of guidelines to ensure a good programming style. There are many aspects when we talk about coding standards like file organization, in Java a file should only contain one class/interface and the identifier have to start with an uppercase letter. An interface should be adjective starting with

2 https://find-sec-bugs.github.io/

3 Hibernate ORM (or simply Hibernate) is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database.

uppercase letter and if there are multiple words, any new word have to start with an uppercase letter as well. The indentation of the code has to be normalized, are they tab, space, how many spaces, etc. It becomes more and more important to follow those guidelines as the project grow up.

It affect a lot of things such as software quality and “Maintaining software quality is very important especially for critical system.”[3]. When working in a team, a document describing the guidelines is usually defined. This document must be read and accepted when the team starts the programming part of the project. If everyone has their own way of coding, it would be difficult to maintain a large project. The maintainability of a project is the ability to modify part of the code without completely breaking the project and the time needed to do so. There is also the refactoring part, which enables the code to be restructured for better readability.

A static code analysis tool can help you to follow the guidelines. It has the ability to detect the duplication of code. There are solutions that allow you to evaluate your code by giving you a rating, whether in terms of code quality, code duplication, maintainability, etc. You can also see where you made a mistake in the code and a way to correct the error, whether it is a security problem, code smells, coding standards problems, etc.

4 Automatic correction

The use of a tool that allows us to have an overview of the different mistakes we could make is a good solution to implement, but in general, the application of solutions to solve these problems does not happen automatically. Some features of these tools can sometimes give imprecise results in the search for a solution to a rule violation, “in other words, they report warnings that may or may not correspond to an actual mistake”[4]. The generation of the list of detected problems can sometimes overwhelm the developer in regard to the amount of work to be done when the tool is applied for the first time on a large project. However, it is possible to configure the tool to automatically correct an entire category of rule violation. The entire project will then be scanned and the correction applied. It is possible to validate each correction manually, which is preferable as the tool can sometimes make mistakes. What is also useful during development is that the correction of a rule violation can also be automatically applied directly in the IDE[4] after an action on your part. This way it is easier for you to see and fix the problem directly when you face it.

5 Conclusion

In today's fast-paced world, it is important to use all the tools at our disposal to make our lives easier. These tools are not only there to save us time, but also to allow us to correct errors that can be critical both in terms of security and application performance. They allow us to easily find the documentation related to the violation of a rule. The integration of these tools are easily done in our favourite IDEs and makes the work easier. Everything is there at our disposal, why not use them?

4 Integrated Development Environment

References

1: Robert A. Martin, Sean Barnum, Steve Christey, Being Explicit About Weaknesses,
https://cwe.mitre.org/documents/being-explicit/BlackHatDC_BeingExplicit_Slides.pdf, 2007
2: CWE, CWE-564: SQL Injection: Hibernate,
https://cwe.mitre.org/data/definitions/564.html, 2006
3: Qirat Ashfaq, Rimsha Khan, Sehrish Farooq, A Comparative Analysis of Static Code Analysis Tools that check Java Code Adherence to Java Coding Standards,
https://ieeexplore-ieee-org.passerelle.univ-rennes1.fr/stamp/stamp.jsp?tp=&arnumber=8681007, 2019
4: Diego Marcilio, Carlo A.Furia, Rodrigo Bonifácio, Gustavo Pinto, SpongeBugs: Automatically generating fix suggestions in response to static code analysis warnings,
https://www-sciencedirect-com.passerelle.univ-rennes1.fr/science/article/pii/S016412122030128X?via%3Dihub, 2020