**CS763 Lab 1 Static Application Security Testing**

Donghang He

## Question 1

SpotBugs setting

图形用户界面, 应用程序

描述已自动生成

1. How many bugs are identified by SpotBugs?

64 bugs are identified by SpotBugs

图形用户界面, 文本, 应用程序

描述已自动生成

1. What are the categories of these bugs?

* Security 8 bugs
* Bad practice 14 bugs
* Experimental 7 bugs
* Internationalization 2 bugs
* Dodgy code 28 bugs
* Correctness 1 bug
* Performance 4 bugs

1. What are security bugs identified?

8 security bugs are identified.

One is HTTP response splitting vulnerability.

Two are HTTP cookie formed from untrusted input.

Other five are potential SQL problem.

图形用户界面, 文本

描述已自动生成

1. Modify the configuration to only report the scary bugs. How many are they?

SpotBugs setting

图形用户界面, 应用程序

描述已自动生成

Only 3 bugs are found

文本

描述已自动生成

1. Can you find any false positive?

No false positives, there are only some situations that turn warnings into bugs

## Question 2

1. How many bugs are identified by SonarQube?

62 bugs are identified by SonarQube

图形用户界面, 文本, 应用程序

描述已自动生成

1. What are the categories of these bugs?

SonarQube does not provide specific bug categories, so I made a simple classification based on the content of the bug.

* Return bug



* Try-with-resources or close

图形用户界面, 文本, 应用程序

描述已自动生成

* Front-end HTML/CSS bug

表格

中度可信度描述已自动生成

1. What are security bugs identified?

Most security bugs are about the resource not be closed, and also there is one about the null pointer.

图片包含 日程表

描述已自动生成

Also, in the vulnerability part there are some minor level bugs may also relate to security bug.

表格

低可信度描述已自动生成

1. How many critical security bugs?

In those bugs there are 13 blocker level bugs which more important than critical level.

图形用户界面, 文本, 应用程序

描述已自动生成

1. Can you find any false positive?

Unexpected duplicate “background“: This bug has appeared many times, but this is a false positive. This way of writing is cross-browser adaptation. webkit represents the adaptation to the chrome browser, moz means adaptation to the Firefox browser.

图形用户界面

描述已自动生成

## Question 3

Compare the results by SpotBugs and SonarQube and state your findings.

1. In SpotBugs it found 64 bugs, and in SonarQube it found 62 bugs.
2. SpotBugs categorizes the bugs found in detail to make it easier to modify. The bugs displayed in SonarQube are not categorized, but are graded according to severity.
3. SpotBugs only analyzed the contents of the java folder, but did not analyze the rest of the jsp part of the webapp. In short, it only performed server-side code analysis, not the entire project. The analysis of SonarQube is more comprehensive, but it may not be as detailed as SpotBugs.
4. In summary, combining the two in the actual project development will produce very good results. And SonarQube also showed code smell related issues. However, according to my development experience in the last semester, most of the front-end code problems found in SonarQube may be avoided during development. Because the more and more humanized integrated development environment (IntelliJ IDEA) will already prompt these not-so-important bugs as warnings when programming. Therefore, many related problems will be avoided in the development process. At the same time, most of the other Java-related bugs can be found in SpotBugs, and they are clearly classified for easy modification. So, I think SpotBugs will be more practical than SonarQube for the projects I have been exposed to.

## Question 4

Choose 5 bugs from any report and explain them in more detail. Do you know how to fix them? You may need to do additional research on them. (Optional: If you are familiar with Java and web development, you may review the code manually, and see if you can find any bugs that are not identified by SpotBugs or SonarQube?)

Many duplicate bugs have been found in SpotBugs and SnoarQube, although only the specific explanations and solutions of 5 bugs are listed below. However, they cover about half of the bugs.

1. Unexpected missing generic font family (from SonarQube)

文本

描述已自动生成

This error is mainly due to the failure to define a general font when defining the font. The general font can ensure that the default font is used when the system cannot find the defined font. The code can be modified as follows:

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

At the same time, this problem is also an error that IntelliJ will automatically prompt as I mentioned before. And provides a simple one-key modification文本

描述已自动生成

1. Add “<th>” headers to this “<table>” (SonarQube)

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Here, the HTML specification requires that the first row of the table must be the header row, and a description of the table must be added. However, the form here is just for the password recovery step after forgetting the password, enter the username and security question to obtain the password. Here I tend to use bootstrap input groups文本

描述已自动生成

1. Null pointers should not be dereferenced (SpotBugs and SonarQube)

图形用户界面, 文本, 应用程序

描述已自动生成

Null pointer exception is reported, and the toString function cannot be executed when sb is empty. A very common problem in Java. Empty parameters cannot be converted to string type. The modification method is as follows:

文本

描述已自动生成

1. Confusing method name (SpotBugs bad practice)

图表

描述已自动生成

There are two problems here. The first is that the first letter of the java class name should be capitalized, and the second is that the class name is ambiguous and has no clear definition. Change xxe to Show.

1. Runtime Exception capture (SpotBugs dodgy code)

图形用户界面, 文本

描述已自动生成

Code errors that are not very severe, the official documents provide a modification plan图形用户界面, 文本, 应用程序

描述已自动生成

Modify it accordingly图形用户界面, 文本, 应用程序

描述已自动生成

## Question 5

If you have done any project before, use SpotBugs or SonarQube to examine one of your projects and report your findings.

I use the SpotBugs to examine my project of CS673.

图形用户界面, 文本, 应用程序

描述已自动生成

After detecting with SpotBugs, it is found that a large number of bugs in the code are not adding final before some constants, which is the negligence of a lot of repeated work. In the figure below, you can see that some error messages have been added final, and some of the error messages defined at the beginning of the project are not added final. This is because we mentioned this problem when we optimized the code later but did not make detailed changes because it did not affect the overall operation.文本

描述已自动生成

Since I am mainly responsible for the front-end implementation in this project, I am not very familiar with many back-end codes. So, I tested it again with SonarQube. It may be a problem with the project SonarQube did not analyze the front-end files but found as many as 195code smell related problems. Which includes the final issue I mentioned earlier.图表

低可信度描述已自动生成

Since this course also has a project, and I plan to continue to use the last semester's project for improvement and safety-related optimization. So, we will continue to use this two software to optimize the bugs in the project.

## Summary

The main purpose of this experiment is to learn to use the two software SpotBugs and SonarQube and can optimize and improve the code through the analysis results of these two software. I learned a lot in this experiment, but I am still not very familiar with this two software. I don't know the modification plan for some bugs raised. The difficulty of this experiment was moderate, and the document clearly stated all the points. For some problems that occur during operation, even if they are not mentioned in the document, they can be solved by browsing related web pages. It's just that I haven't found any bug classification for SonarQube, so the last few questions of the second question are somewhat difficult to answer. Both software are very easy to use and will help my future project development.