Let a bot deal with your static analysis warnings backlog

EclipseCon 2022
27/10/2022
# TABLE OF CONTENTS

/01 /STATIC ANALYSIS WARNINGS

/02 /SORALD

/03 /DEMO

/04 /EXPERIMENTS
/WHO ARE WE?

Research Engineer @ KTH
amansha@kth.se
KTH, Stockholm, Sweden

PHD Student @ KTH
khaes@kth.se
KTH, Stockholm, Sweden
“A potential bug in the program found without executing it”
Calling .equals on a can result in null pointer exception

```java
String a = null;
return a.equals("foo");

String a = null;
return "foo".equals(a);
```
HOW TO FIND THEM?

Method invocation 'equals' will produce 'NullPointerException'

```java
src/main/java/Violation.java
3     String a = null;
4
5     return a.equals("foo");
6   }
7 }
```
IT IS ALSO A BUG!
“LOTS OF OTHER THINGS TO DO RIGHT NOW”

Something more actionable is required.
SpoonLabs/sorald

Automatic repair system for static analysis warnings from SonarQube's SonarJava
It ain't much, but it's honest work.
Abstract Syntax Tree (AST): tree to represent the structure of the program.

```c
int main() {
    return 0;
}
```
/SPOON

/AST TRANSFORMATION

main
---
int
---
return
---
0

main
---
int
---
return
---
1

/HIGH FIDELITY PRETTY-PRINTING

int main() {
    return 1;
}

27-10-2022 | Aman Sharma | Khashayar Etemadi
/HOW SORALD WORKS: DEEP DIVE

Hey, Sorald! Repair this for me, please?

On it. @sonarqube, do you find violations?

Yes. I find S1132 at line 3 column 8.

Let me do the AST transformation now :)

@spoon, please make it pretty. :*

Sure thing. @user, here you go!

So grateful to all of you <3
1 String a = null;
2 return a.equals("foo");
```java
@Override
protected void repairInternal(CtInvocation<?> element) {
    // Get executable reference of method to be called
    CtExecutableReference<?> ctExecutableReferenceToMethodToBeCalled = element.getExecutable();

    // Create new invocation in which the receiver and argument are swapped
    CtInvocation<?> newInvocation =
        getFactory()
            .createInvocation
                .element().getArguments().get(0),
                ctExecutableReferenceToMethodToBeCalled,
                element.getTarget());

    // Replace old invocation by the new one
    element.replace(newInvocation);

    // Delete null check on the variable if it exists
    deleteNullCheckIfExists(newInvocation);
}
```
/WAYS TO USE SORALD

Choose your favourite:

$ java -jar sorald.jar repair

$ mvn sorald:repair

fix: Fix violation of Sonar rule 2097 #3972
/DEMO (STEPS)

/Mining
We use Sorald’s miner to find violations

/Repairing
We use Sorald’s repair to fix violations

/Re-mining
We mine violations again to check if they are fixed

/Compile
We compile to ensure we have not introduced an error
Let’s see how Sorald works in action:

- We repair a project by SonarSource: Sonar-Scanner-Cli@5c518d6 https://github.com/SonarSource/sonar-scanner-cli

- We use Sorald’s maven plugin

- We fix violations of S1132

  ```
  $ mvn se.kth.castor:sorald:repair -DruleKey=S1132
  ```

- We re-mine violations and compile the project after the repair

- S1132: “Strings literals should be placed on the left side when checking for equality”
  - Solution: Move the literal to the left
/EXPERIMENTING IN THE WILD

/161 Popular Repos

We select 161 Github repos:
- More than 50 stars
- PR & CI friendly
- Active & Maven
- Healthy (mvn compile)

/10 Sonar Rules

We experiment with 10 rules:
- Bug type
- Common in past experiments

27-10-2022 | Aman Sharma | Khashayar Etemadi
1. `Thread.run()` should not be called directly.
2. Synchronization should not be based on Strings or boxed primitives.
3. Resources should be closed.
4. `BigDecimal(double)` should not be used.
5. `hashCode` and `toString` should not be called on array instances.
6. `InterruptedException` should not be ignored.
7. Math operands should be cast before assignment.
8. `toString()` and `clone()` methods should not return null.
9. `Iterator.next()` methods should throw `NoSuchElementException`.
10. Strings and Boxed types should be compared using `equals()`.
Violations fixed in 161 projects
54 patches for 80 newly introduced violations in 350 days
Number of merged PRs: 17
Number of closed PRs: 10
Number of pending PRs: 2

*As of the date of publication*
Stevehu commented on Dec 4, 2021

@khaes-kth This is a very good tool and it would help a lot of open-source projects. The PR looks pretty good and easy to understand with a detailed description. I like the reference links if users want to explore the extra details.

Regarding the coding style, I found the first file has some indentation issues with some extra spaces. Other than that, looks good.

Again, thanks a lot for your help.
There are other analyzers to be plugged in:
- Qodana
- SpotBugs
- ...

Unacceptable patches can be filtered out:
- False positive warnings
  Eg. str1==str2
- Uncertain fixes
AUTOMATIC REPAIR TOOL

AST TRANSFORMATIONS

PART OF DEVELOPMENT PROCESS
/THANKS!

amansha@kth.se
khaes@kth.se

Link to GitHub repository:
https://github.com/SpoonLabs/sorald

White Paper:

Voting Link

White Paper: