

CSE 5382 Secure Programming Fall 2022

Programming Assignment 1 - Introduction to Static Analysis

Prem Atul Jethwa | paj1810 | 1001861810

Tool Versions

The tools I have used to test the code are:

- FindBugs – 3.0.1, findsecbugs-plugin-1.8.0
- SonarQube

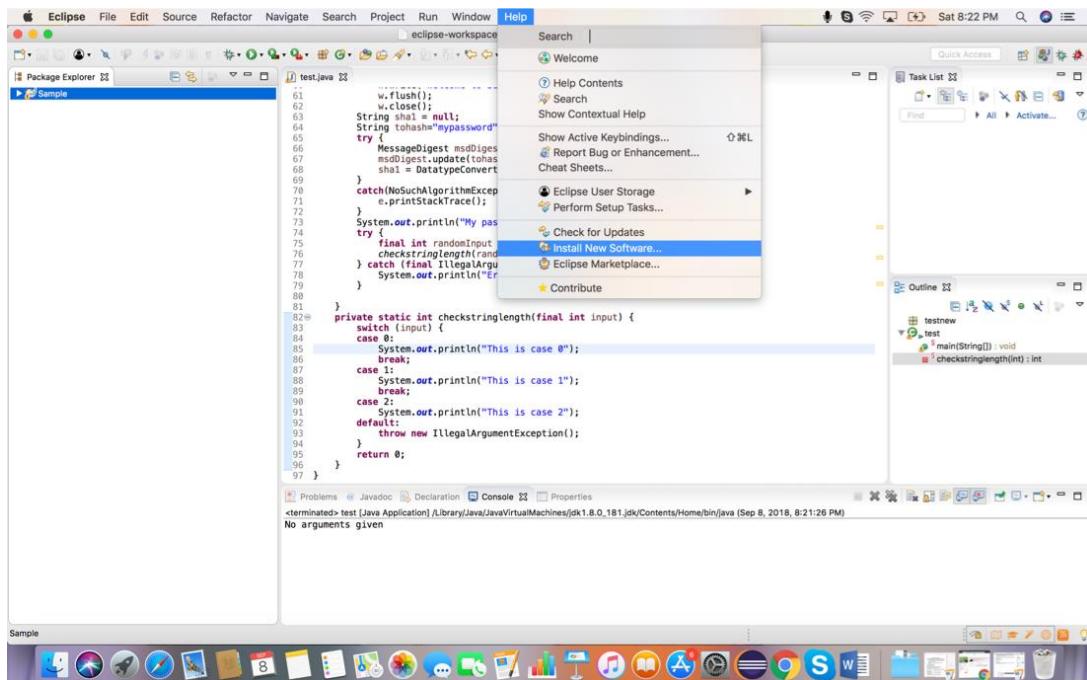
FindBugs – 3.0.1, findsecbugs-plugin-1.8.0

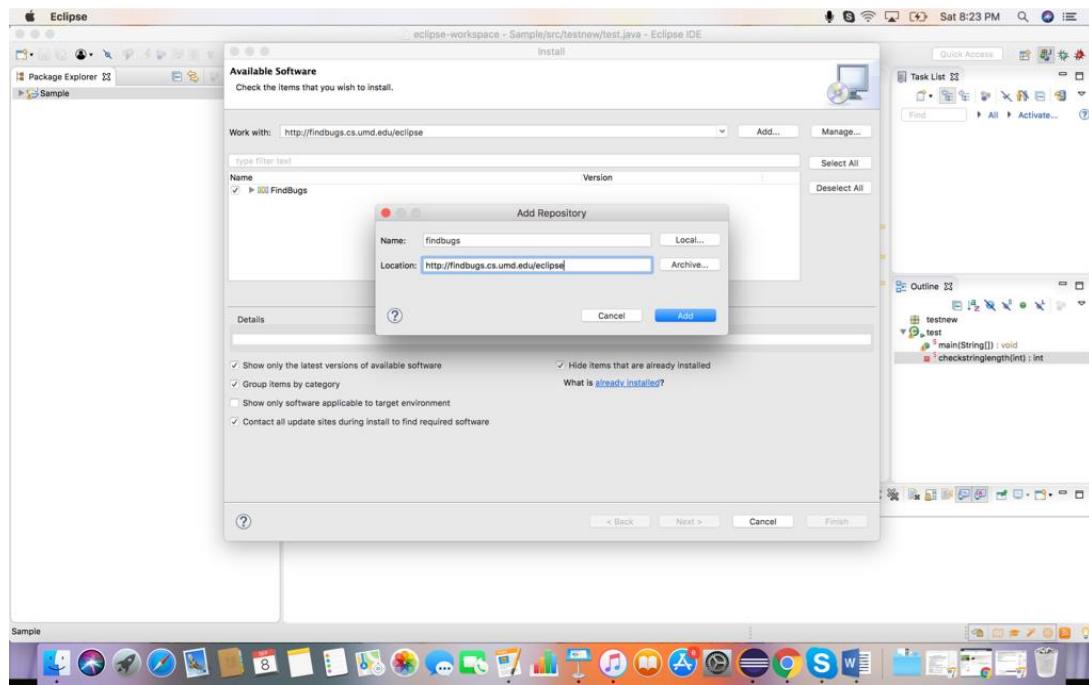
Tool Invocation Process

Eclipse is installed, From Help tab, go to “Install New Software” to add the FindBugs plugin as shown below:

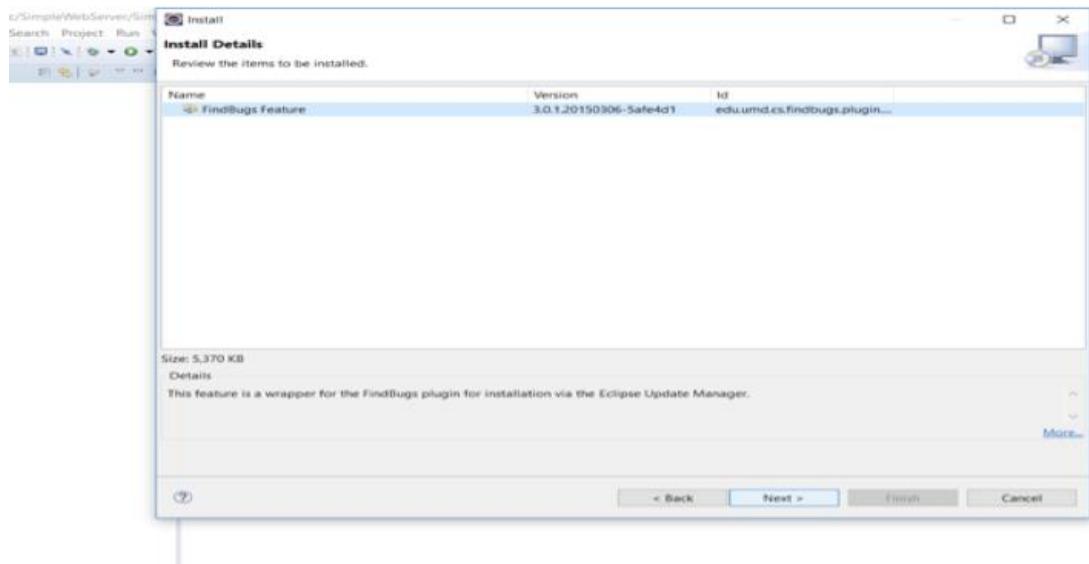
1. Click on “Install New Software”
2. Give the url to get the FindBugs plugin for eclipse as:
<http://findbugs.cs.umd.edu/eclipse> and click on “Add”

Reference: <http://findbugs.sourceforge.net/>

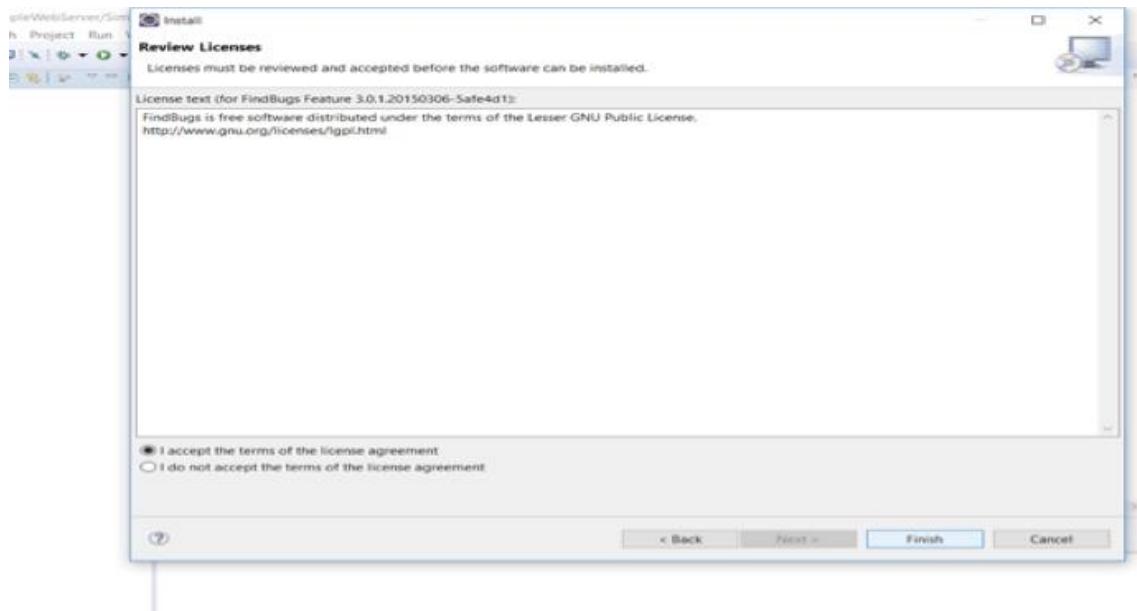




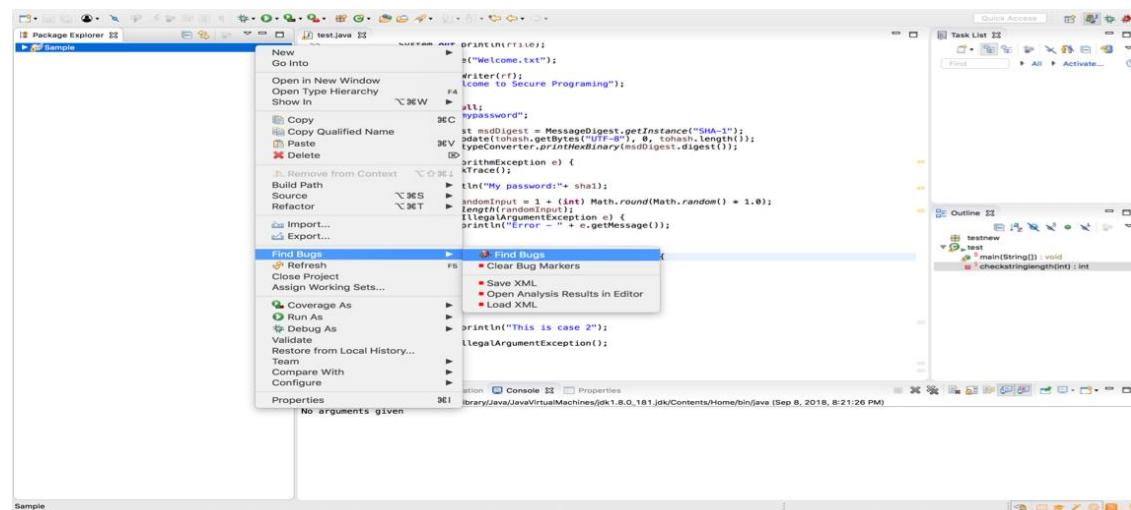
3. Click 'Next' again as shown below:



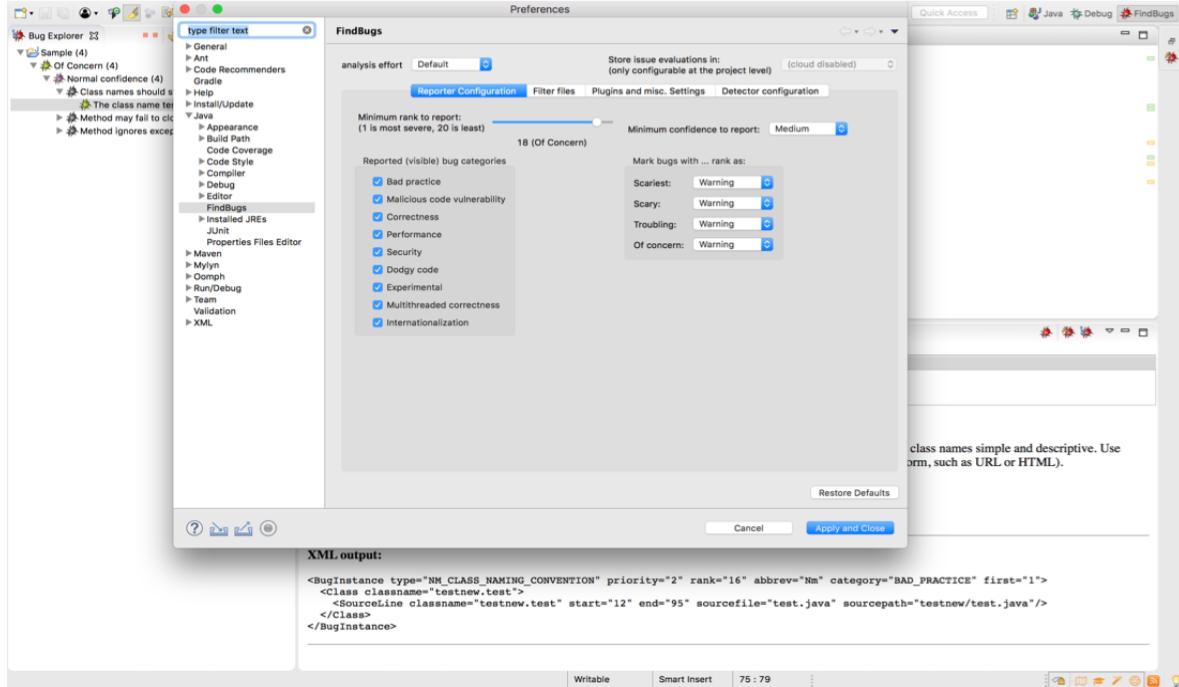
4. Select 'I accept' and click Finish to complete the installation FindBugs plugin for eclipse:



5. It will ask you to restart eclipse, click ok to restart eclipse.
6. To run the tool, select the project in the package explorer and select Find Bugs and in the sub menu, select 'Find Bugs' again.
7. Added the included example in the eclipse



The below Find bugs explorer will display the bugs present in the code. The recommended configuration to use with Find Security Bugs is to limit the scan to Security only bug detectors. Go to Eclipse -> Preferences (Mac) or Window -> Preferences (Windows). Then go to Java -> FindBugs, and make sure only "Security" is checked on the "Reporting configuration" tab's "Reported (visible) bug categories" list.

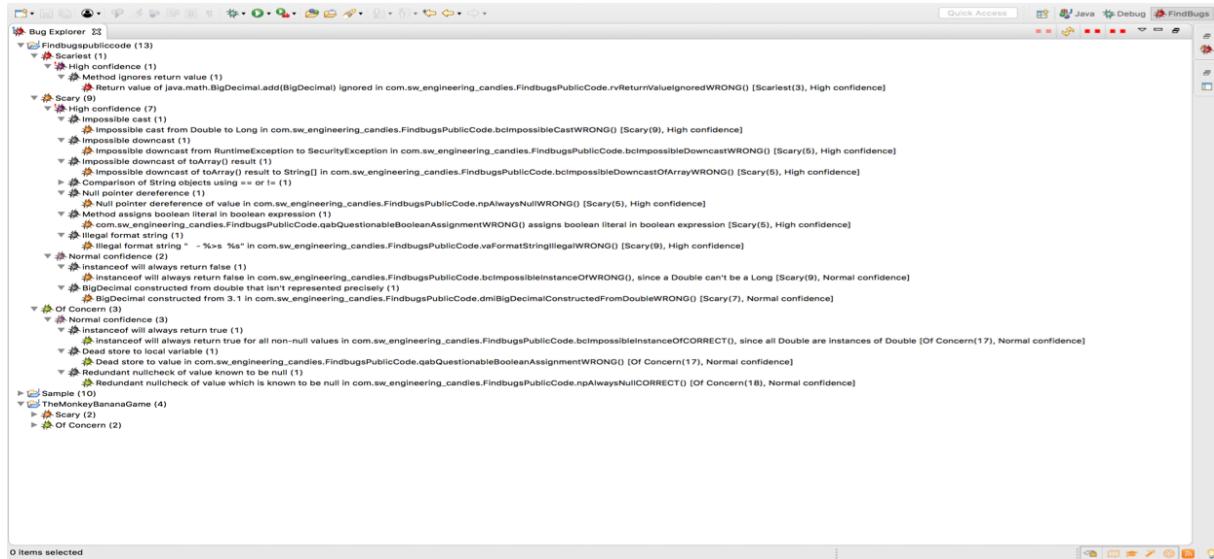


Find Bugs categorizes the bugs as follows:

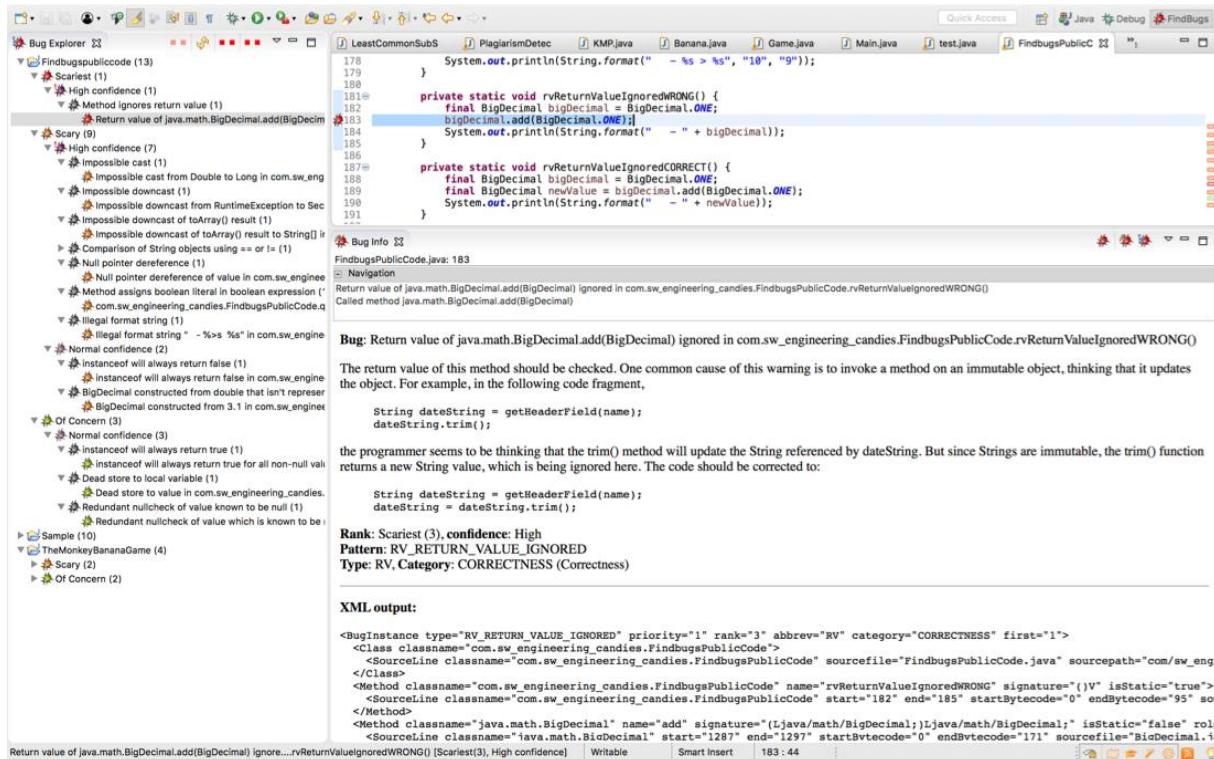
1. Scariest: This type of bug is ranked between 1 & 4
2. Scary: This type of bug is ranked between 5 & 9
3. Troubling: This type of bug is ranked between 10 & 14
4. Of concern: This type of bug is ranked between 15 & 20

With the FindBugs the below code is giving 13 bugs.

Reference: <http://www.sw-engineering-candies.com/blog-1/findbugstmwarningsbysample-parti>



Bug 1



Bug 2

Bug Info:

FindbugsPublicCode.java: 99

Navigation

Bug: Impossible cast from Double to Long in com.sw_engineering_candies.FindbugsPublicCode.bcImpossibleCastWRONG()

This cast will always throw a ClassCastException. FindBugs tracks type information from instanceof checks, and also uses more precise information about the types of values returned from methods and loaded from fields. Thus, it may have more precise information that just the declared type of a variable, and can use this to determine that a cast will always throw an exception at runtime.

Rank: Scary (9), confidence: High
Pattern: BC_IMPOSSIBLE_CAST
Type: BC, Category: CORRECTNESS (Correctness)

XML output:

```
<BugInstance type="BC_IMPOSSIBLE_CAST" priority="1" rank="9" abbrev="BC" category="CORRECTNESS" first="1">
    <Class classname="com.sw_engineering_candies.FindbugsPublicCode">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng/candies/FindbugsPublicCode.java" start="98" end="101" startBytecode="0" endBytecode="15" sourcetext="private static void rvReturnValueIgnoredWRONG() { final BigDecimal bigDecimal = BigDecimal.ONE; bigDecimal.add(BigDecimal.ONE); System.out.println(String.format(" - %s > %s", "10", "9")); }" />
    </Class>
    <Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="bcImpossibleCastWRONG" signature="()V" isStatic="true">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="98" end="101" startBytecode="0" endBytecode="15" sourcetext="private static void rvReturnValueIgnoredWRONG() { final BigDecimal bigDecimal = BigDecimal.ONE; bigDecimal.add(BigDecimal.ONE); System.out.println(String.format(" - %s > %s", "10", "9")); }" />
    </Method>
    <Type descriptor="Ljava/lang/Double;" role="TYPE_FOUND">
        <SourceLine classname="java.lang.Double" start="49" end="1053" sourcefile="Double.java" sourcepath="java/lang/Double.java"/>
    </Type>
    <Type descriptor="Ljava/lang/Long;" role="TYPE_EXPECTED">
        <SourceLine classname="java.lang.Long" start="54" end="1615" sourcefile="Long.java" sourcepath="java/lang/Long.java"/>
    </Type>
    <LocalVariable name="doubleValue" register="0" pc="5" role="LOCAL_VARIABLE_VALUE_OF"/>
    <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="99" end="99" startBytecode="6" endBytecode="6" sourcetext="final BigDecimal newValue = bigDecimal.add(BigDecimal.ONE); System.out.println(String.format(" - %s > %s", "10", "9")); }" />
</BugInstance>
```

Bug 3

Bug Info:

FindbugsPublicCode.java: 111

Navigation

Bug: Impossible downcast from RuntimeException to SecurityException in com.sw_engineering_candies.FindbugsPublicCode.bcImpossibleDowncastWRONG()

This cast will always throw a ClassCastException. The analysis believes it knows the precise type of the value being cast, and the attempt to downcast it to a subtype will always fail by throwing a ClassCastException.

Rank: Scary (5), confidence: High
Pattern: BC_IMPOSSIBLE_DOWNCAST
Type: BC, Category: CORRECTNESS (Correctness)

XML output:

```
<BugInstance type="BC_IMPOSSIBLE_DOWNCAST" priority="5" rank="5" abbrev="BC" category="CORRECTNESS" first="1">
    <Class classname="com.sw_engineering_candies.FindbugsPublicCode">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng/candies/FindbugsPublicCode.java" start="110" end="113" startBytecode="0" endBytecode="18" sourcetext="private static void rvReturnValueIgnoredCORRECT() { final BigDecimal bigDecimal = BigDecimal.ONE; final BigDecimal newValue = bigDecimal.add(BigDecimal.ONE); System.out.println(String.format(" - %s > %s", "10", "9")); }" />
    </Class>
    <Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="bcImpossibleDowncastWRONG" signature="()V" isStatic="true">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="110" end="113" startBytecode="0" endBytecode="18" sourcetext="private static void rvReturnValueIgnoredCORRECT() { final BigDecimal bigDecimal = BigDecimal.ONE; final BigDecimal newValue = bigDecimal.add(BigDecimal.ONE); System.out.println(String.format(" - %s > %s", "10", "9")); }" />
    </Method>
    <Type descriptor="Ljava/lang/RuntimeException;" role="TYPE_FOUND">
        <SourceLine classname="java.lang.RuntimeException" start="51" end="118" sourcefile="RuntimeException.java" sourcepath="java/lang/RuntimeException.java"/>
    </Type>
    <Type descriptor="Ljava/lang/SecurityException;" role="TYPE_EXPECTED">
        <SourceLine classname="java.lang.SecurityException" start="42" end="83" sourcefile="SecurityException.java" sourcepath="java/lang/SecurityException.java"/>
    </Type>
    <LocalVariable name="exception" register="0" pc="10" role="LOCAL_VARIABLE_VALUE_OF"/>
    <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="111" end="111" startBytecode="11" endBytecode="11" sourcetext="System.out.println(String.format(" - %s > %s", "10", "9")); }" />
</BugInstance>
```

Bug 4

Screenshot of the Eclipse IDE showing the FindBugs plugin. The Bug Explorer view displays a list of bugs found in the file `FindbugsPublicCode.java`. A specific bug is selected, showing its code location and details.

```

178     System.out.println(String.format(" - %s > %s", "18", "9"));
179 }
180
181 private static void rvReturnValueIgnoredWRONG() {
182     final BigDecimal bigDecimal = BigDecimal.ONE;
183     bigDecimal.add(BigDecimal.ONE);
184     System.out.println(String.format(" - " + bigDecimal));
185 }
186
187 private static void rvReturnValueIgnoredCORRECT() {
188     final BigDecimal bigDecimal = BigDecimal.ONE;
189     final BigDecimal newValue = bigDecimal.add(BigDecimal.ONE);
190     System.out.println(String.format(" - " + newValue));
191 }

```

Bug Info

`FindbugPublicCode.java: 135`

Description: Impossible downcast of `toArray()` result to `String[]`

Pattern: BC_IMPOSSIBLE_DOWNCASE_OF_TOARRAY

Type: BC, Category: CORRECTNESS (Correctness)

XML output:

```

<BugInstance type="BC_IMPOSSIBLE_DOWNCASE_OF_TOARRAY" priority="1" rank="5" abbrev="BC" category="CORRECTNESS" first="1">
<Class classname="com.sw_engineering_candies.FindbugsPublicCode">

```

Bug 5

Screenshot of the Eclipse IDE showing the FindBugs plugin. The Bug Explorer view displays a list of bugs found in the file `FindbugsPublicCode.java`. A specific bug is selected, showing its code location and details.

```

155     }
156
157     private static void esComparingStringsWithEqWRONG() {
158         final StringBuilder sb1 = new StringBuilder("1234");
159         final StringBuilder sb2 = new StringBuilder("1234");
160         final String string1 = sb1.toString();
161         final String string2 = sb2.toString();
162         System.out.println(" - " + (string1 == string2));
163     }
164
165     private static void esComparingStringsWithEqCORRECT() {
166         final StringBuilder sb1 = new StringBuilder("1234");
167         final StringBuilder sb2 = new StringBuilder("1234");
168         final String string1 = sb1.toString();
169     }

```

Bug Info

`FindbugPublicCode.java: 162`

Description: Comparison of String objects using == or != in com.sw_engineering_candies.FindbugsPublicCode.esComparingStringsWithEqWRONG()

Pattern: ES_COMPARING_STRINGS_WITH_EQ

Type: ES, Category: BAD_PRACTICE (Bad practice)

XML output:

```

<BugInstance type="ES_COMPARING_STRINGS_WITH_EQ" priority="1" rank="9" abbrev="ES" category="BAD_PRACTICE" first="1">
<Class classname="com.sw_engineering_candies.FindbugsPublicCode">
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw/engineering/candies/FindbugsPublicCode.java" start="158" end="162" startBytecode="0" endBytecode="30" soi="soi-Method"/>
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" name="esComparingStringsWithEqWRONG" signature="(V)V" isStatic="true" soi="soi-Method"/>
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="158" end="162" startBytecode="0" endBytecode="30" soi="soi-Method"/>
<TypeDescriptor type="Ljava/lang/String;" role="TYPE_FOUND" start="111" end="3141" sourcefile="String.java" sourcepath="java/lang/String.java"/>
<LocalVariable name="string2" register="3" pc="13" role="LOCAL_VARIABLE_VALUE_OF"/>
<LocalVariable name="string1" register="4" pc="13" role="LOCAL_VARIABLE_VALUE_OF"/>
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="162" end="162" startBytecode="44" endBytecode="44" soi="soi-Method"/>
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="162" end="162" startBytecode="44" endBytecode="44" soi="soi-Method"/>
<Property name="edu.umdc.cs.findbugs.detect.RefComparisonWarningProperty" value="DYNAMIC_AND_UNKNOWN" />

```

Bug 6

Null pointer dereference of value in com.sw_engineering_candies.FindbugsPublicCode.npAlwaysNullWRONG()

A null pointer is dereferenced here. This will lead to a `NullPointerException` when the code is executed.

Rank: Scary (5), confidence: High
Pattern: NP_ALWAYS_NULL
Type: NP, Category: CORRECTNESS (Correctness)

XML output:

```
<BugInstance type="NP_ALWAYS_NULL" priority="1" rank="5" abbrev="NP" category="CORRECTNESS" first="1">
    <Class classname="com.sw_engineering_candies.FindbugsPublicCode">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng/candies/FindbugsPublicCode.java" startBytecode="155" endBytecode="169" startLine="155" endLine="169" startColumn="1" endColumn="24" file="FindbugsPublicCode.java" line="155" column="1" />
        <Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="npAlwaysNullWRONG" signature="()V" isStatic="true">
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="194" end="200" startBytecode="169" endBytecode="175" startLine="194" endLine="200" startColumn="1" endColumn="6" file="FindbugsPublicCode.java" line="194" column="1" />
            <LocalVariable name="value" register="0" pc="11" role="LOCAL_VARIABLE_VALUE_OF"/>
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="195" end="195" startBytecode="175" endBytecode="175" startLine="195" endLine="195" startColumn="1" endColumn="1" file="FindbugsPublicCode.java" line="195" column="1" />
        </Method>
    </Class>
</BugInstance>
```

Bug 7

com.sw_engineering_candies.FindbugsPublicCode.qabQuestionableBooleanAssignmentWRONG() assigns boolean literal in boolean expression [Scary(5), High confidence]

This method assigns a literal boolean value (true or false) to a boolean variable inside an if or while expression. Most probably this was supposed to be a boolean comparison using `==`, not an assignment using `=`.

Rank: Scary (5), confidence: High
Pattern: QBA_QUESTIONABLE_BOOLEAN_ASSIGNMENT
Type: QBA, Category: CORRECTNESS (Correctness)

XML output:

```
<BugInstance type="QBA_QUESTIONABLE_BOOLEAN_ASSIGNMENT" priority="1" rank="5" abbrev="QBA" category="CORRECTNESS" first="1">
    <Class classname="com.sw_engineering_candies.FindbugsPublicCode">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng/candies/FindbugsPublicCode.java" startBytecode="212" endBytecode="218" startLine="212" endLine="218" startColumn="1" endColumn="24" file="FindbugsPublicCode.java" line="212" column="1" />
        <Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="qabQuestionableBooleanAssignmentWRONG" signature="()V" isStatic="true">
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="212" end="218" startBytecode="212" endBytecode="218" startLine="212" endLine="218" startColumn="1" endColumn="6" file="FindbugsPublicCode.java" line="212" column="1" />
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="213" end="213" startBytecode="218" endBytecode="218" startLine="213" endLine="213" startColumn="1" endColumn="1" file="FindbugsPublicCode.java" line="213" column="1" />
        </Method>
    </Class>
</BugInstance>
```

Bug 8

```

<BugInstance type="VA_FORMAT_STRING_ILLEGAL" priority="1" rank="9" abbrev="FS" category="CORRECTNESS" first="1">
    <Class classname="com.sw_engineering_candies.FindbugsPublicCode">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng/Findbugspubliccode(13)" start="174" end="175" startBytecode="0" endBytecode="61" so</SourceLine>
    </Class>
    <Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="vaFormatStringIllegalWRONG" signature="()V" isStatic="true">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="174" end="175" startBytecode="0" endBytecode="61" so</SourceLine>
        <Method>
            <SourceLine classname="java.lang.String" name="format" signature="(Ljava/lang/String;[Ljava/lang/Object;)Ljava/lang/String;" isStatic="t</SourceLine>
            <String value=" - %s %s" role="STRING_FORMAT_STRING"/>
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="174" end="174" startBytecode="19" endBytecode="19" sou</SourceLine>
        </Method>
    </Method>
</BugInstance>

```

Bug 9

```

<BugInstance type="BC_IMPOSSIBLE_INSTANCEOF" priority="2" rank="9" abbrev="BC" category="CORRECTNESS" first="1">
    <Class classname="com.sw_engineering_candies.FindbugsPublicCode">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng/Findbugspubliccode(13)" start="122" end="124" startBytecode="0" endBytecode="13" so</SourceLine>
    </Class>
    <Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="bcImpossibleInstanceOfWRONG" signature="()V" isStatic="true">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="122" end="124" startBytecode="0" endBytecode="13" so</SourceLine>
        <Type descriptor="Ljava/lang/Double;" role="TYPE_FOUND">
            <SourceLine classname="java.lang.Double" start="49" end="1053" sourcefile="Double.java" sourcepath="java/lang/Double.java"/>
        </Type>
        <Type descriptor="Ljava/lang/Long;" role="TYPE_EXPECTED">
            <SourceLine classname="java.lang.Long" start="54" end="1615" sourcefile="Long.java" sourcepath="java/lang/Long.java"/>
        </Type>
        <LocalVariable name="value" register="0" pc="17" role="LOCAL_VARIABLE_VALUE_OF"/>
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="123" end="123" startBytecode="18" endBytecode="18" sou</SourceLine>
    </Method>
</BugInstance>

```

Bug 10

Bug Info:

FindbugsPublicCode.java: 148

Navigation

BigDecimal constructed from 3.1 in com.sw_engineering_candies.FindbugsPublicCode.dmiBigDecimalConstructedFromDoubleWRONG()

Called method new java.math.BigDecimal(double)

Did you intend to invoke java.math.BigDecimal.valueOf(double)

Value 3.1

Value 3.10000000000000005551151231257827021181583404541015625. You probably want to use the BigDecimal.valueOf(double d) method, which uses the String representation of the double to create the BigDecimal (e.g., BigDecimal.valueOf(0.1) gives 0.1).

Bug: BigDecimal constructed from 3.1 in com.sw_engineering_candies.FindbugsPublicCode.dmiBigDecimalConstructedFromDoubleWRONG()

This code creates a BigDecimal from a double value that doesn't translate well to a decimal number. For example, one might assume that writing new BigDecimal(0.1) in Java creates a BigDecimal which is exactly equal to 0.1 (an unscaled value of 1, with a scale of 1), but it is actually equal to 0.10000000000000005551151231257827021181583404541015625. You probably want to use the BigDecimal.valueOf(double d) method, which uses the String representation of the double to create the BigDecimal (e.g., BigDecimal.valueOf(0.1) gives 0.1).

Rank: Scary (7), **confidence:** Normal

Pattern: DMI_BIGDECIMAL_CONSTRUCTED_FROM_DOUBLE

Type: DMI, **Category:** CORRECTNESS (Correctness)

XML output:

```
<BugInstance type="DMI_BIGDECIMAL_CONSTRUCTED_FROM_DOUBLE" priority="2" rank="7" abbrev="DMI" category="CORRECTNESS" first="1">
<Class classname="com.sw_engineering_candies.FindbugsPublicCode">
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng:148">
<Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="dmiBigDecimalConstructedFromDoubleWRONG" signature="()V" isStatic="true" start="148" end="150" startBytecode="0" endBytecode="86" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng:148">
<Method classname="java.math.BigDecimal" name="<init>" signature="(D)V" isStatic="false" role="METHOD_CALLED">
<SourceLine classname="java.math.BigDecimal" start="872" end="873" startBytecode="0" endBytecode="36" sourcefile="BigDecimal.java" sourcepath="java/math:872">
<Method classname="java.math.BigDecimal" name="valueOf" signature="(D)Ljava/math.BigDecimal;" isStatic="true" role="METHOD_ALTERNATIVE">
<SourceLine classname="java.math.BigDecimal" start="1274" end="1274" startBytecode="0" endBytecode="35" sourcefile="BigDecimal.java" sourcepath="java/math:1274">
</Method>
</Method>
</Class>
</BugInstance>
```

Bug 11

Bug Info:

FindbugsPublicCode.java: 128

Navigation

instanceof will always return true for all non-null values in com.sw_engineering_candies.FindbugsPublicCode.bcImpossibleInstanceOfCORRECT(), since all Double are instances of Double

Expected type Double

Bug: instanceof will always return true for all non-null values in com.sw_engineering_candies.FindbugsPublicCode.bcImpossibleInstanceOfCORRECT(), since all Double are instances of Double

This instanceof test will always return true (unless the value being tested is null). Although this is safe, make sure it isn't an indication of some misunderstanding or some other logic error. If you really want to test the value for being null, perhaps it would be clearer to do better to do a null test rather than an instanceof test.

Rank: Of Concern (17), **confidence:** Normal

Pattern: BC_VACUOUS_INSTANCEOF

Type: BC, **Category:** STYLE (Dodgy code)

XML output:

```
<BugInstance type="BC_VACUOUS_INSTANCEOF" priority="2" rank="17" abbrev="BC" category="STYLE" first="1">
<Class classname="com.sw_engineering_candies.FindbugsPublicCode">
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng:128">
<Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="bcImpossibleInstanceOfCORRECT" signature="()V" isStatic="true" start="127" end="129" startBytecode="0" endBytecode="13" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng:128">
<Type descriptor="Ljava/lang/Double;" role="TYPE_FOUND">
<SourceLine classname="java.lang.Double" start="49" end="1053" sourcefile="Double.java" sourcepath="java/lang/Double.java"/>
</Type>
<Type descriptor="Ljava/lang/Double;" role="TYPE_EXPECTED">
<SourceLine classname="java.lang.Double" start="49" end="1053" sourcefile="Double.java" sourcepath="java/lang/Double.java"/>
</Type>
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="128" end="128" startBytecode="18" endBytecode="18" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng:128">
<SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="128" end="128" startBytecode="18" endBytecode="18" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng:128">
</SourceLine>
</Method>
</Class>
</BugInstance>
```

Bug 12

Bug Info

FindbugsPublicCode.java: 213

Navigation

Dead store to value in com.sw_engineering_candies.FindbugsPublicCode.qabQuestionableBooleanAssignmentWRONG()

Local variable named value

Bug: Dead store to value in com.sw_engineering_candies.FindbugsPublicCode.qabQuestionableBooleanAssignmentWRONG()

This instruction assigns a value to a local variable, but the value is not read or used in any subsequent instruction. Often, this indicates an error, because the value computed is never used.

Note that Sun's javac compiler often generates dead stores for final local variables. Because FindBugs is a bytecode-based tool, there is no easy way to eliminate these false positives.

Rank: Of Concern (17), **confidence:** Normal
Pattern: DLS_DEAD_LOCAL_STORE
Type: DLS, Category: STYLE (Dodgy code)

XML output:

```
<BugInstance type="DLS_DEAD_LOCAL_STORE" priority="2" rank="17" abbrev="DLS" category="STYLE" first="1">
    <Class classname="com.sw_engineering_candies.FindbugsPublicCode">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng">
            <Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="qabQuestionableBooleanAssignmentWRONG" signature="(V)" isSta
                <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="212" end="218" startBytecode="0" endBytecode="19" so
            </Method>
            <LocalVariable name="value" register="0" pc="5" role="LOCAL_VARIABLE_NAMED"/>
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="213" end="213" startBytecode="4" endBytecode="4" sou
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="213" end="213" startBytecode="4" endBytecode="4" sou
            <Property name="edu.umd.cs.findbugs.detect.DeadLocalStoreProperty.BASE_VALUE" value="true"/>
            <Property name="edu.umd.cs.findbugs.detect.DeadLocalStoreProperty.LOCAL_NAME" value="value"/>
            <Property name="edu.umd.cs.findbugs.detect.DeadLocalStoreProperty.NO_LOADS" value="true"/>
        </BugInstance>
```

Dead store to value in com.sw.engineering_candies.FindbugsPublicCode.qabQuestionableBooleanAssignmentWRONG() [Of Concern(17), Normal confidence]

Bug 13

Bug Info

FindbugsPublicCode.java: 204

Navigation

Redundant nullcheck of value which is known to be null in com.sw_engineering_candies.FindbugsPublicCode.npAlwaysNullCORRECT()

Value loaded from value

Bug: Redundant nullcheck of value which is known to be null in com.sw_engineering_candies.FindbugsPublicCode.npAlwaysNullCORRECT()

This method contains a redundant check of a known null value against the constant null.

Rank: Of Concern (18), **confidence:** Normal
Pattern: RCN_REDUNDANT_NULLCHECK_OF_NULL_VALUE
Type: RCN, Category: STYLE (Dodgy code)

XML output:

```
<BugInstance type="RCN_REDUNDANT_NULLCHECK_OF_NULL_VALUE" priority="2" rank="18" abbrev="RCN" category="STYLE" first="1">
    <Class classname="com.sw_engineering_candies.FindbugsPublicCode">
        <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" sourcefile="FindbugsPublicCode.java" sourcepath="com/sw_eng">
            <Method classname="com.sw_engineering_candies.FindbugsPublicCode" name="npAlwaysNullCORRECT" signature="(V) isStatic=true">
                <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="203" end="209" startBytecode="0" endBytecode="138" so
            </Method>
            <LocalVariable name="value" register="0" pc="2" role="LOCAL_VARIABLE_VALUE_OF"/>
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="204" end="204" startBytecode="3" endBytecode="3" sou
            <SourceLine classname="com.sw_engineering_candies.FindbugsPublicCode" start="204" end="204" startBytecode="3" endBytecode="3" sou
        </BugInstance>
```

Redundant nullcheck of value which is known to be null in com.sw.engineering.candies.FindbugsPublicCode.npAlwaysNullCORRECT() [Of Concern(18), Normal confidence]

SonarQube

SonarQube, is a self-managed, automatic code review tool that systematically helps you deliver Clean Code. As a core element of our Sonar solution, SonarQube integrates into your existing workflow and detects issues in your code to help you perform continuous code inspections of your projects.

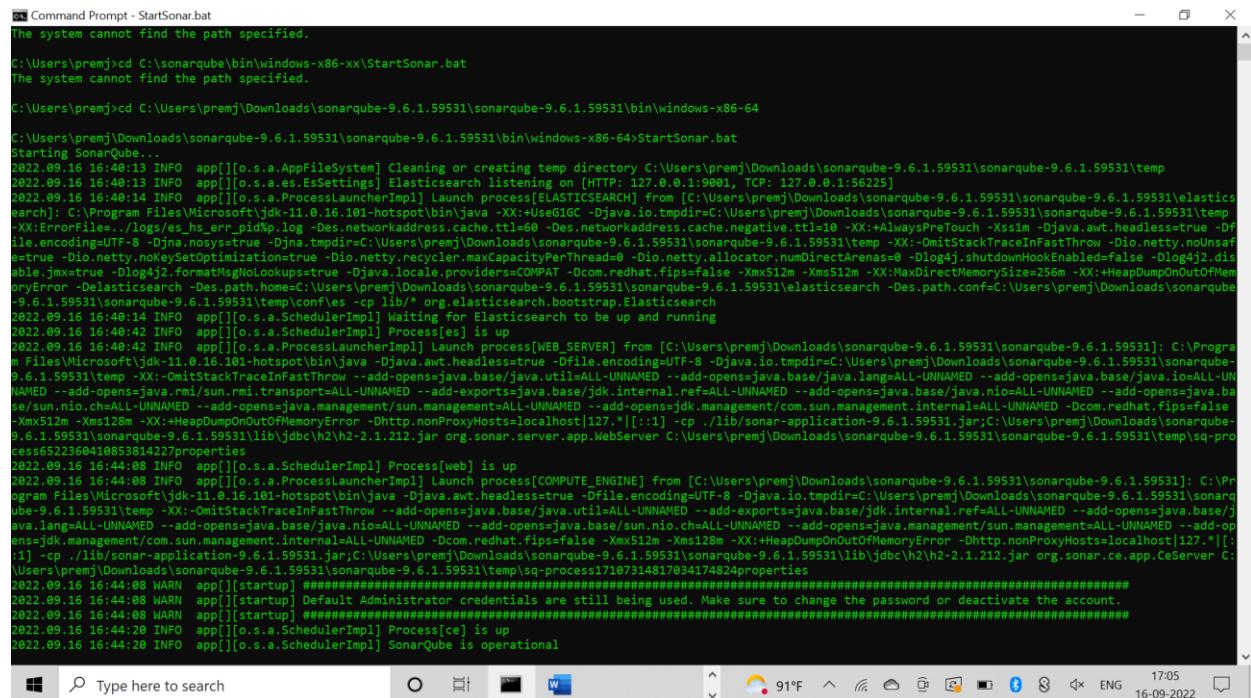
SonarQube is a Static Analysis tool which collects and analyses source code and provides Code Quality Assurance by detecting code vulnerabilities and issues.

Website to download the SonarQube: - <https://www.sonarqube.org/download-community-edition/>

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

Inputs: SonarQube takes entire code (Java) as an input. It will go through each line of the source code to detect bugs, vulnerabilities, and defects.

In the below screenshot I have started the SonarQube using terminal (windows)

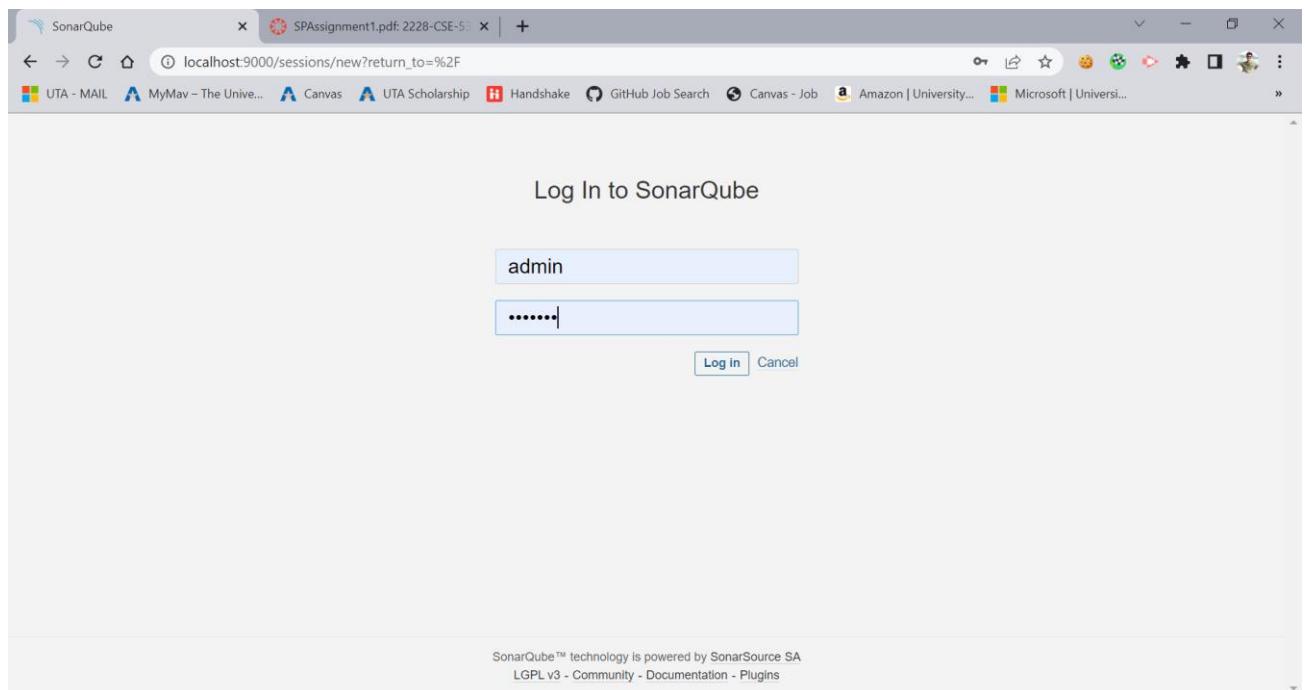


```
Command Prompt - StartSonar.bat
The system cannot find the path specified.

C:\Users\premj>cd C:\sonarqube\bin\windows-x86-xx\StartSonar.bat
The system cannot find the path specified.

C:\Users\premj>cd C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\bin\windows-x86-64\StartSonar.bat
Starting SonarQube...
2022.09.16 16:48:13 INFO [o.s.a.AppFileSystem] Cleaning or creating temp directory C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\temp
2022.09.16 16:48:13 INFO [o.s.a.es.EsSettings] Elasticsearch listening on [HTTP: 127.0.0.1:9001, TCP: 127.0.0.1:56225]
2022.09.16 16:48:14 INFO [o.s.a.ProcessLauncherImpl] Launch process[ELASTICSEARCH] from [C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\elasticsearch]: C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java -XX:+UseG1GC -Djava.io.tmpdir=C:\Users\premj\Downloads\sonarqube-9.6.1.59531\temp -XX:+OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dlogaj.shutdownHookEnabled=false -Dlog4j.disable.jmx=true -Dlog4j.formatMSNoLogups=true -Djava.locale.providers=COMPAT -Dcom.redhat.fips=false -Xmx512m -Xms512m -XX:MaxDirectMemorySize=256m -XX:+HeapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\elasticsearch -Des.path.conf=C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\temp\conf\les -cp lib/* org.elasticsearch.bootstrap.Elasticsearch
2022.09.16 16:48:14 INFO [app][o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
2022.09.16 16:48:42 INFO [app][o.s.a.SchedulerImpl] Process[es] is up
2022.09.16 16:48:42 INFO [app][o.s.a.ProcessLauncherImpl] Launch process[WEB_SERVER] from [C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531]: C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\temp -XX:+OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.rmi=sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.nio.channels=ALL-UNNAMED --add-opens=java.base/java.nio.charset=ALL-UNNAMED --add-opens=java.base/java.nio.charset=UTF-8 -Djava.io.tmpdir=C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\temp -XX:+OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.nio.channels=ALL-UNNAMED --add-opens=java.base/java.nio.charset=ALL-UNNAMED -Dcom.redhat.fips=false -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost[127.*[:]*] -cp ./lib/sonar-application-9.6.1.59531.jar;C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\temp\sq-processes6522360410853814227\properties
2022.09.16 16:44:08 INFO [app][o.s.a.ProcessLauncherImpl] Process[web] is up
2022.09.16 16:44:08 INFO [app][o.s.a.ProcessLauncherImpl] Launch process[COMPUTE_ENGINE] from [C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531]: C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\temp -XX:+OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.nio.channels=ALL-UNNAMED --add-opens=java.base/java.nio.charset=ALL-UNNAMED --add-opens=java.base/java.nio.charset=UTF-8 -Djava.io.tmpdir=C:\Users\premj\Downloads\sonarqube-9.6.1.59531\sonarqube-9.6.1.59531\temp\sq-process17107314817034174824\properties
2022.09.16 16:44:08 WARN [app][startup] #####
2022.09.16 16:44:08 WARN [app][startup] Default Administrator credentials are still being used. Make sure to change the password or deactivate the account.
2022.09.16 16:44:08 WARN [app][startup] #####
2022.09.16 16:44:20 INFO [app][o.s.a.SchedulerImpl] Process[ce] is up
2022.09.16 16:44:20 INFO [app][o.s.a.SchedulerImpl] SonarQube is operational
```

Once the SonarQube is up, the following login page appears at **localhost:9000**



After logging in I created the project, I chose to analyse the project locally.
To analyze it locally I decided to use Maven tool.

A screenshot of the SonarQube dashboard. The top navigation bar includes "Projects", "Issues", "Rules", "Quality Profiles", "Quality Gates", and "Administration". A search bar on the right says "Search for projects..." with a magnifying glass icon. Below the search bar is a "Create Project" button and a home icon. On the left, there is a sidebar with "My Favorites" (highlighted) and "All". Under "Filters", there are buttons for "Clear All Filters", "Passed" (0), and "Failed" (0). There are also sections for "Quality Gate" (Passed 0, Failed 0), "Reliability" (A rating 0, B rating 0, C rating 0, D rating 0, E rating 0), and "Security" (A rating 0, B rating 0, C rating 0, D rating 0, E rating 0). The main content area shows a project named "MyProject1" with a star icon. It says "Project's Main Branch is not analyzed yet." and has a "Configure analysis" button. A message box at the bottom left says "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." At the bottom right, it says "SonarQube™ technology is powered by SonarSource SA" and "Community Edition - Version 9.6.1 (build 59531) - LGPL v3 - Community - Documentation - Plugins - Web API".

SonarQube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects... A

MyProject1 master

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Do you want to integrate with your favorite CI? Choose one of the following tutorials.

- With Jenkins
- With GitHub Actions
- With Bitbucket Pipelines
- With GitLab CI
- With Azure Pipelines
- Other CI

Are you just testing or have an advanced use-case? Analyze your project locally.

- Locally

Manually uploaded the code using Maven 3.8.6

SonarQube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects... A

MyProject1 master

Overview Issues Security Hotspots Measures Code Analysis Project Settings Project Information

1 Provide a token Analyze "MyProject1":sqp_28a5dd66dd85f35f28343cd4409f941834e11be2

2 Run analysis on your project

What option best describes your build?

Maven Gradle .NET Other (for JS, TS, Go, Python, PHP, ...)

Execute the Scanner for Maven

Running a SonarQube analysis with Maven is straightforward. You just need to run the following command in your project's folder.

```
mvn clean verify sonar:sonar \
-Dsonar.projectKey=MyProject1 \
-Dsonar.host.url=http://localhost:9000 \
-Dsonar.login=sqp_28a5dd66dd85f35f28343cd4409f941834e11be2
```

Please visit the [official documentation of the Scanner for Maven](#) for more details.

Is my analysis done? If your analysis is successful, this page will automatically refresh in a few moments.

You can set up Pull Request Decoration under the project settings. To set up analysis with your favorite CI tool, see the tutorials.

Once the analysis is completed the SonarQube launches an analysis report. It lists the issues as well as the severity of the issues.

The screenshot shows the SonarQube interface for the project 'MyProject1'. The top navigation bar includes 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar at the top right contains the placeholder 'Search for projects...'. Below the navigation, a summary indicates 'Last analysis had 1 warning' on 'September 11, 2022 at 10:22 PM Version 0.0.1-SNAPSHOT'. The main area displays a list of issues under the 'Issues' tab. On the left, a sidebar shows the file structure: 'src/main/java/hash.java'. The first issue listed is a 'Code Smell' (Severity: Minor) with the message 'Move this file to a named package.' It includes a 'Bulk Change' button and a detailed view of the issue with its status, assignee, and effort. Other issues listed include 'Rename this class name to match the regular expression "[A-Z][a-zA-Z0-9]*\$"', 'Complete the task associated to this TODO comment.', 'Replace this use of System.out or System.err by a logger.', and 'Replace this use of System.out or System.err by a logger.' (another one). Each issue has a detailed view with its location (line number), severity, status, assignee, effort, and a 'Comment' section.

SonarQube gives suggestions about how the issue can be avoided

This screenshot shows the SonarQube interface for the project 'MyProject1', focusing on the 'Issues' tab. The left sidebar shows the file structure: 'src/main/java/hash.java'. The first issue listed is a 'Code Smell' (Severity: Major) with the message 'Replace this use of System.out or System.err by a logger.' It includes a 'Bulk Change' button and a detailed view of the issue with its status, assignee, and effort. The code snippet shown is 'System.out.println("Remainder = " + remainder);'. Below this, another 'Code Smell' (Severity: Major) with the message 'Replace this use of System.out or System.err by a logger.' is listed, also with a code snippet: 'Map<Integer, String> map = new HashMap();'. Further down, two more 'Code Smell' entries are shown: 'Provide the parametrized type for this generic.' (Severity: Major) with code 'Map<Integer, String> mapp = new HashMap(3);' and 'Provide the parametrized type for this generic.' (Severity: Major) with code 'System.out.println("Inserting the values into the HashMap!!!");'. Each issue has a detailed view with its location (line number), severity, status, assignee, effort, and a 'Comment' section.

SonarQube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects... A

MyProject1 master Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

This block of commented-out lines of code should be removed. Why is this an issue? 5 minutes ago L99 %

Code Smell Major Open Not assigned 5min effort Comment unused

```
assertEquals("1000", Integer.toBinaryString(8));
assertEquals("10100", Integer.toBinaryString(20));

for (int i=0; i<numrows; i++)
    for (int j=0; j<numcols; j++);
    pixels++;
```

Remove this useless assignment to local variable "pixels". Why is this an issue? 5 minutes ago L105 %

Code Smell Major Open Not assigned 15min effort Comment cert, cwe, unused

```
switch (ch) {
    case 1:
        do_something(1); break;
    case 2:
        do_something(2); break;
    case 3:
        do_something(1); break;
    case 4:
        do_something(4); break;
```

Replace this use of System.out or System.err by a logger. Code Smell

Replace this use of System.out or System.err by a logger. Code Smell

Provide the parametrized type for this generic. Code Smell

Provide the parametrized type for this generic. Code Smell

Replace this use of System.out or System.err by a logger. Code Smell

In the following image, SonarQube identifies minor bugs and gives the description

SonarQube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects... A

MyProject1 master Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information 1 / 5 issues 27min effort

vulnerability

Code Smell 5 Bulk Change

Severity MINOR Clear

- Blocker 0 Minor 5
- Critical 1 Info 2
- Major 21

Ctrl + click to add to selection

Scope Resolution Status Security Category Creation Date Language Rule Tag

src/main/java/hash.java

Move this file to a named package. Why is this an issue? 4 minutes ago % convention

Code Smell Minor Open Not assigned 10min effort Comment

Rename this class name to match the regular expression ^[A-Z][a-zA-Z0-9]*\$. Why is this an issue? 4 minutes ago L4 % convention

Code Smell Minor Open Not assigned 5min effort Comment

Rename this method name to match the regular expression ^[a-z][a-zA-Z0-9]*\$. Why is this an issue? 4 minutes ago L73 % convention

Code Smell Minor Open Not assigned 5min effort Comment

Declare "numcols" and all following declarations on a separate line. Why is this an issue? 4 minutes ago L94 % convention

Code Smell Minor Open Not assigned 2min effort Comment

Remove this unused "pixels" local variable. Why is this an issue? 4 minutes ago L94 % unused

Code Smell Minor Open Not assigned 5min effort Comment

5 of 5 shown

Conclusion:

- Both the tools take Java Byte code as an input.
- Following bugs can be found:
FindBugs: Performance, Bad practice, Dodgy , Internationalization, Malicious code, Vulnerability, Bogus random, Noise, Correctness, Multithreaded, Style.
- FindBugs cannot find Security category, so we have FindSecurityBugs with the security mode enabled.
FindSecurityBugs: Security Bugs is to limit the scan to Security only bug detectors on, i.e. when the security category is enabled it is FindSecurityBugs Security.s
- FindBugs/FindSecurityBugs will find based on the suspicious patterns in the byte code and when it hits such byte strings in our byte code, it flags it to vulnerability.
- As **SonarQube** can work as both client and server. So, it is more efficient than **FindBugs**.