FB.NM\_WRONG\_PACKAGE

Детектор определяет ошибку, в ходе которой метод в подклассе не переопределяет аналогичный метод в суперклассе, поскольку тип параметра не совсем соответствует типу соответствующего параметра в суперклассе.

Таблица №1 – Основные характеристики детектора FB.NM\_WRONG\_PACKAGE

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Несоответствие типов параметров может нарушить функциональность программы, приводя к ошибкам времени выполнения, невозможности вызвать методы и неправильной логике работы кода. |
| Комментарий | Эта ошибка может привести к непредсказуемому поведению программы, так как несоответствие типов параметров может вызвать проблемы с вызовом метода и его работой на уровне экземпляров классов. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class Parent {  public void method(String str) {  // ...  }  }  class Child extends Parent {  // @Override - error  public void method(int n) {  // ...  }  }  } | public static void correctTest() {  class Parent {  public void method(String str) {  // ...  }  }  class Child extends Parent {  @Override  public void method(String str) {  // ...  }  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Используйте аннотацию @Override:** Всегда добавляйте аннотацию @Override перед методом в подклассе, который вы предполагаете переопределить. Компилятор проверит, действительно ли существует метод с такой же сигнатурой в суперклассе.
2. **Тщательно проверяйте типы параметров:** Убедитесь, что типы параметров в методе подкласса полностью совпадают с типами параметров в методе суперкласса, включая порядок и регистр (для ссылочных типов).
3. **Обратите внимание на примитивные типы и их обертки:** int и Integer, double и Double и т.д. являются разными типами. Если метод в суперклассе принимает примитивный тип, метод в подклассе для переопределения должен принимать тот же примитивный тип.
4. **Учитывайте ковариантность возвращаемых типов (начиная с Java 5):** Хотя возвращаемый тип может быть ковариантным (подклассом возвращаемого типа суперкласса), типы параметров должны быть идентичными для переопределения.
5. **Проверяйте иерархию классов:** Убедитесь, что вы правильно понимаете иерархию классов и какой метод в суперклассе вы пытаетесь переопределить.