FB.RC\_REF\_COMPARISON

Детектор определяет ошибку, в ходе которой метод сравнивает два ссылочных значения с помощью оператора == или !=, при этом правильный способ сравнения экземпляров этого типа обычно заключается в метод еquals(). Можно создавать отдельные экземпляры, которые равны, но не сравниваются как ==, поскольку это разные объекты. Примерами классов, которые обычно не следует сравнивать по ссылке, являются java.lang.Integer, java.lang.Float и т. д. RC\_REF\_COMPARISON охватывает только типы-оболочки для примитивов.

Таблица №1 – Основные характеристики детектора FB.RC\_REF\_COMPARISON

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Неправильное использование оператора сравнения может привести к логическим ошибкам в программе, так как объекты, которые должны быть равны, могут быть признаны неравными из-за различия в их ссылках. |
| Комментарий | Использование оператора == или != для сравнения ссылочных типов может привести к неожиданным результатам, так как он проверяет ссылки на объекты, а не их фактические значения, что нарушает логику сравнения в некоторых классах, например, в типах-оболочках для примитивов. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  Integer a = 1000;  Integer b = 1000;  System.out.println(a == b);  } | public static void correctTest() {  Integer a = 1000;  Integer b = 1000;  System.out.println(a.equals(b));  } |

Методы устранения уязвимости при срабатывании детектора:

1. Замените операторы == на метод equals() для сравнения объектов ссылочных типов (особенно типов-оболочек) на предмет равенства их значений.
2. Замените операторы != на отрицание результата метода equals() (!a.equals(b)) для проверки неравенства значений.
3. Убедитесь, что сравниваемые объекты не являются примитивными типами (например, int, float), для которых операторы == и != корректно сравнивают значения. Детектор RC\_REF\_COMPARISON специально нацелен на типы-оболочки.
4. Помните, что для некоторых классов (помимо типов-оболочек), где логическое равенство определяется состоянием объекта, также необходимо использовать метод equals() (который должен быть правильно переопределен в этих классах). Однако RC\_REF\_COMPARISON фокусируется именно на типах-оболочках для примитивов.
5. Используйте операторы == и != только тогда, когда вам действительно нужно проверить, являются ли две переменные ссылками на один и тот же объект в памяти (например, при сравнении с null или при проверке идентичности объектов-синглтонов).