FB.FE\_FLOATING\_POINT\_EQUALITY

Детектор определяет ошибку, в ходе которой операция сравнивает два значения с плавающей запятой на предмет равенства. Поскольку вычисления с плавающей запятой могут включать округление, вычисленные значения float и double могут быть неточными. Для значений, которые должны быть точными, например денежных значений, рассмотрите возможность использования типа с фиксированной точностью, такого как BigDecimal. Для значений, которые не должны быть точными, рассмотрите возможность сравнения на равенство в некотором диапазоне, например: if ( Math.abs(x - y) < .0000001 ).

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Сравнение чисел с плавающей запятой (float и double) на точное равенство может привести к ошибкам из-за потери точности при вычислениях, что может вызвать неожиданные логические ошибки. |
| Комментарий | Используйте BigDecimal для финансовых расчетов или сравнивайте числа с погрешностью, используя Math.abs(x - y) < epsilon, где epsilon — допустимая разница. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  double a = 0.1 \* 3;  double b = 0.3;  // сравнение значений с плавающей запятой на точное равенство  if (a == b) {  System.out.println("Значения равны");  } else {  System.out.println("Значения не равны");  }  } | public static void correctTest() {  double a = 0.1 \* 3;  double b = 0.3;  final double EPSILON = 1e-9;  // проверяем разницу значений в пределах допустимой погрешности  if (Math.abs(a - b) < EPSILON) {  System.out.println("Значения равны");  } else {  System.out.println("Значения не равны");  }  } |