FB.BX\_BOXING\_IMMEDIATELY\_UNBOXED\_TO\_PERFORM\_COERCION

Детектор определяет ошибку, в ходе которой примитивное упакованное значение, созданное и затем немедленно преобразованное в другой тип примитива (например, new Double(d).intValue()). Просто выполните прямое примитивное приведение (например, (int) d)

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE192 |
| CVE | — |
| Возможные последствия | Ненужная упаковка и последующая распаковка примитивного значения увеличивает затраты памяти и процессорного времени, создавая дополнительную нагрузку на систему. |
| Комментарий | Следует избегать ненужного создания объектов-обёрток при преобразовании типов примитивов и использовать прямое приведение. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  double d = 42.5;  // Боксинг Double, затем немедленный анбоксинг в int  // deprecated since version 9  Integer boxed = new Double(d).intValue();  System.out.println(boxed);  } | public static void correctTest() {  double d = 42.5;  // Просто выполняем приведение типа  int value = (int) d;  System.out.println(value);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Используйте прямое приведение типов:** Вместо создания объекта-обертки и последующего вызова метода для преобразования, используйте оператор прямого приведения ((целевойТип) значение). Например, (int) doubleValue, (long) floatValue.
2. **Избегайте промежуточных объектов-оберток:** Не создавайте объекты Double, Integer, Float и т.д., если единственная цель - преобразовать их в другой примитивный тип.
3. **Проверьте необходимость преобразования:** Убедитесь, что преобразование типа действительно необходимо. Возможно, можно работать с исходным типом данных.
4. **Обратите внимание на потерю точности:** При прямом приведении типов (например, double к int) может происходить потеря точности (отбрасывание дробной части). Убедитесь, что такое поведение является ожидаемым.