FB.BX\_BOXING\_IMMEDIATELY\_UNBOXED

Детектор определяет ошибку, в ходе которой примитив упаковывается, а затем сразу же распаковывается. Вероятно, это связано с ручным боксом в месте, где требуется неупакованное значение, что вынуждает компилятор немедленно отменить работу бокса.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Необходимость упаковки и последующей распаковки примитива приводит к избыточным операциям, увеличению нагрузки на сборщик мусора и снижению производительности. |
| Комментарий | Следует избегать ненужного бокса примитивных типов и использовать примитивные значения напрямую, если упаковка не требуется. Это позволит уменьшить накладные расходы и повысить эффективность работы программы. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| // Принимаем Integer вместо int  public static int sum(Integer a, Integer b) {  // Авто-распаковка  return a + b;  }  public static void incorrectTest() {  // Боксинг и немедленный анбоксинг внутри метода  int result = sum(10, 20);  System.out.println(result);  } | // Принимаем Integer вместо int  public static int sum(Integer a, Integer b) {  // Авто-распаковка  return a + b;  }  public static void correctTest() {  int result = sum(1, 2);  System.out.println(result);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Используйте примитивные типы напрямую:** Если нет явной необходимости в использовании объектов-оберток (например, при работе с коллекциями, параметризованными типами или рефлексией), используйте примитивные типы (int, double, boolean и т.д.) напрямую.
2. **Измените сигнатуры методов:** Если метод принимает или возвращает объекты-обертки, но внутри работает с примитивными значениями, рассмотрите возможность изменения сигнатуры метода для использования примитивных типов.
3. **Избегайте ручного боксинга, если это не требуется:** Не создавайте объекты-обертки вручную (Integer.valueOf(), new Integer()) там, где можно использовать примитивные значения. Автоматический боксинг/анбоксинг может скрыть такие избыточные операции.
4. **Проанализируйте контекст:** Разберитесь, почему происходит упаковка и немедленная распаковка. Возможно, проблема кроется в другом месте кода, где требуется объект-обертка, но в данном конкретном участке кода достаточно примитивного типа.