FB.ICAST\_INTEGER\_MULTIPLY\_CAST\_TO\_LONG

Ошибка "result of integer multiplication cast to long" указывает на то, что результат умножения целых чисел затем приводится к типу long. Это предупреждение может появиться, когда результат умножения целочисленных значений, которые могут поместиться в типе int, приводится к типу long.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE192 |
| CVE | — |
| Возможные последствия | Приведение результата целочисленного умножения к типу long может быть лишним и неэффективным, если результат умножения умещается в диапазоне типа int. |
| Комментарий | Если результат умножения целых чисел не выходит за пределы диапазона типа int, нет необходимости приводить его к типу long. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  int a = 1000;  int b = 2000;  // результат умножения целых чисел приводится к типу long  long result = (long) (a \* b);  System.out.println("Incorrect Result: " + result);  } | public static void correctTest() {  int a = 1000;  int b = 2000;  // результат умножения целых чисел не нуждается в приведении к long  long result = a \* (long) b;  System.out.println("Correct Result: " + result);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Приведите один из операндов к long перед умножением (для предотвращения переполнения):** Если существует риск, что результат умножения может выйти за пределы диапазона int, приведите хотя бы одну из целочисленных переменных к типу long *до* выполнения операции умножения. Например: long result = (long) a \* b; или long result = a \* (long) b;.
2. **Удалите избыточное приведение (если переполнение int исключено):** Если вы уверены, что результат умножения всегда будет помещаться в диапазоне int, нет необходимости явно приводить результат к long. Java автоматически выполнит неявное расширение типа int до long при присваивании.
3. **Проанализируйте диапазоны входных значений:** Определите максимальные возможные значения целочисленных операндов и убедитесь, что их произведение не превышает Integer.MAX\_VALUE, если вы не приводите к long до умножения.
4. **Будьте внимательны к порядку операций:** Порядок приведения типов и выполнения арифметических операций имеет значение. Приведение результата после целочисленного умножения не решает проблему переполнения.