INT\_OVERFLOW

Обычно возникает, когда результат арифметической операции превышает максимальное или минимальное допустимое значение для типа int.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE190;CWE191 |
| CVE | — |
| Возможные последствия | Может привести к неожиданным результатам, нарушению логики программы, потенциальным сбоям или уязвимостям, особенно при использовании переполненных значений в условиях или циклах. |
| Комментарий | Следует использовать защиту от переполнения — например, применять методы безопасной арифметики или использовать типы с расширенным диапазоном, если это необходимо. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(int a, int b) {  // результат может стать отрицательным  int result = a + b;  System.out.println("Результат (переполнен): " + result);  } | public static void correctTest(int a, int b) {  if ((long) a + b > Integer.MAX\_VALUE) {  System.out.println("Переполнение при сложении!");  } else {  int result = a + b;  System.out.println("Результат: " + result);  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Используйте тип данных с большим диапазоном для хранения результата операции, которая может вызвать переполнение. Например, если вы складываете или умножаете два int, сохраните результат в long.
2. Перед выполнением потенциально переполняющейся операции проверьте, не превысят ли операнды границы допустимого диапазона. Например, при сложении a + b, проверьте, что a <= Integer.MAX\_VALUE - b.