INT\_OVERFLOW.ZERO.WRAP

Указывает на потенциальную ошибку переполнения целых чисел, которая приводит к "заворачиванию" значения обратно к нулю или близкому к нему значению.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Unknown |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE190;CWE191;CWE194;CWE195;CWE196;CWE197 |
| CVE | — |
| Возможные последствия | Переполнение с заворачиванием может привести к тому, что значение переменной будет неожиданно установлено в ноль или близкое к нулю значение, что может вызвать логические ошибки, некорректное поведение программы или сбои. |
| Комментарий | В случае с "wraparound" важно использовать типы данных, поддерживающие более широкий диапазон, или предусматривать дополнительные проверки для предотвращения переполнений. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  int a = Integer.MAX\_VALUE;  // Переполнение и "заворачивание" значения  int result = a + 1;  System.out.println(result);  } | public static void correctTest(int a) {  if (a == Integer.MAX\_VALUE) {  System.out.println("Overflow detected, cannot add 1!");  } else {  int result = a + 1;  System.out.println(result);  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Используйте тип данных с большим диапазоном (например, long вместо int) для хранения результата операции, которая может вызвать переполнение.
2. Перед выполнением операции, которая может привести к переполнению, проверьте, не превысят ли операнды границы допустимого диапазона, чтобы предотвратить заворачивание.