INT\_OVERFLOW.LOOP

Детектор указывает на ситуацию, когда происходит переполнение целочисленного типа данных в цикле.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | VeryLow |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE190;CWE191;CWE194;CWE195;CWE196;CWE197 |
| CVE | — |
| Возможные последствия | Переполнение в цикле может привести к непредсказуемым результатам, ошибкам в логике программы и даже к сбоям или нарушению целостности данных. |
| Комментарий | Необходимо внимательно контролировать значения, изменяемые в цикле, и обеспечить проверку на переполнение до выполнения операций, чтобы избежать потери данных или некорректных вычислений. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  int result = 0;  for (int i = 0; i < Integer.MAX\_VALUE; i++) {  result += i;  if (result < 0) {  System.out.println("Overflow detected!");  break;  }  }  } | public static void correctTest() {  // Используем long, чтобы избежать переполнения  long result = 0;  for (int i = 0; i < Integer.MAX\_VALUE; i++) {  result += i;  if (result > Integer.MAX\_VALUE) {  System.out.println("Overflow detected!");  break;  }  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Используйте тип данных с большим диапазоном (например, long) для переменной, которая может переполниться в цикле. Это позволит избежать переполнения, если результат не выходит за пределы long.
2. Перед выполнением операции, которая может привести к переполнению, проверьте, не превысит ли результат максимальное (или не станет ли меньше минимального) значение типа int.
3. Если переполнение является ожидаемым или частью логики (например, циклический счетчик), явно обработайте это поведение с помощью модульной арифметики или других соответствующих операций.
4. В условиях выхода из цикла используйте сравнения, которые не зависят от переполнения. Например, вместо проверки result < 0 после сложения, проверьте, не произошло ли "перескок" через Integer.MAX\_VALUE.