INT\_OVERFLOW.TRUNC.UNDER\_BITMASK

Детектор указывает на потенциальную ошибку переполнения целых чисел, которая приводит к "обрезанию" значения под битовой маской.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Average |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE190;CWE191;CWE194;CWE195;CWE196;CWE197 |
| CVE | — |
| Возможные последствия | Может привести к потере значимых битов, некорректной обработке данных и нарушению логики работы программы, особенно в операциях с низкоуровневой манипуляцией битами. |
| Комментарий | Следует обеспечить корректность применения битовых масок и использовать явное приведение типов с проверкой границ значений, чтобы избежать нежелательного поведения. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  int largeValue = 0x1FF; // 511, больше, чем 1 байт  byte truncated = (byte) (largeValue & 0xFF); // маска обрезает до младших 8 бит  // это "wrap around" (выведет -1 (т.к. 0xFF == -1 в byte))  System.out.println("Truncated value: " + truncated); // Выведет -1 (т.к. 0xFF == -1 в byte)  } | public static void correctTest(int largeValue) {  if (largeValue <= 0xFF) {  byte result = (byte) (largeValue & 0xFF);  System.out.println("Safe truncated value: " + result);  } else {  System.out.println("Value too large to safely truncate under bitmask!");  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Перед применением битовой маски и усечением значения до меньшего типа, проверьте, находится ли исходное значение в допустимом диапазоне целевого типа.
2. Если исходное значение может выходить за пределы диапазона целевого типа, обработайте этот случай соответствующим образом. Это может включать вывод предупреждения, выброс исключения или использование альтернативной логики.
3. Используйте явное приведение типов только после проверки диапазона, чтобы избежать неожиданного "wrap-around" или потери данных из-за переполнения.