INT\_OVERFLOW.TRUNC.COND

Указывает на ситуацию, когда происходит переполнение целочисленного типа данных из-за операции усечения, и это переполнение происходит в условном выражении.

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 | Может привести к неверной оценке условий, ошибочным переходам в логике программы и потере значимости данных, особенно при приведении типов с большей разрядностью к меньшей. |
| Комментарий | Рекомендуется внимательно анализировать приведение типов и использовать защитные проверки при работе с условными конструкциями, чтобы избежать непредсказуемого поведения из-за потери данных. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(long a) {  long bigValue = Integer.MAX\_VALUE + a;  // Приведение к int вызывает переполнение  if ((int) bigValue > 100) {  System.out.println("Truncated value is greater than 100");  } else {  System.out.println("Truncated value is NOT greater than 100");  }  } | public static void correctTest(long a) {  long bigValue = Integer.MAX\_VALUE + a;  // Используем переменную как есть, без усечения  if (bigValue > 100) {  System.out.println("Long value is greater than 100");  } else {  System.out.println("Long value is NOT greater than 100");  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Избегайте усечения (приведения к меньшему типу) значений, которые могут привести к переполнению, непосредственно в условных выражениях. Вместо этого, либо работайте с исходным типом большей разрядности, либо выполните проверку диапазона перед усечением.
2. Если необходимо привести значение к меньшему типу для сравнения в условии, сначала убедитесь, что значение находится в пределах допустимого диапазона целевого типа.
3. Используйте более широкий тип данных для выполнения сравнения в условном выражении, если это возможно. В примере, сравнивая long bigValue с 100, нет необходимости приводить bigValue к int.
4. Если усечение неизбежно, и вам нужно сравнить усеченное значение, рассмотрите возможность сравнения с границами целевого типа перед усечением, чтобы предсказать результат сравнения усеченного значения.