TAINTED.INT\_OVERFLOW.TRUNC

Детектор указывает на переполнение int из ненадежного источника. Trunk указывает, что некоторые данные усекаются или округляются, что приводит к потере точности или искажению информации.

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 некорректной обработке данных, потере значимых битов или неожиданному поведению программы, особенно если значение используется в критических операциях. |
| Комментарий | Рекомендуется проверять диапазон значений и валидировать входные данные перед преобразованием типов или арифметическими операциями. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  File file = new File("large\_file.bin");  // может вернуть значение больше Integer.MAX\_VALUE  long fileSize = file.length();  int bufferSize = (int) fileSize; // УСЕЧЕНИЕ  byte[] buffer = new byte[bufferSize]; // ArrayIndexOutOfBounds или OutOfMemory  System.out.println(buffer.length);  } | public static void correctTest() {  File file = new File("large\_file.bin");  long fileSize = file.length();  if (fileSize > Integer.MAX\_VALUE) {  System.out.println("Файл слишком большой для обработки в массиве int.");  } else {  int bufferSize = (int) fileSize;  byte[] buffer = new byte[bufferSize];  System.out.println(buffer.length);  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Валидируйте входные данные. Перед преобразованием ненадежных данных в тип int проверьте, находится ли значение в допустимом диапазоне для типа int (от Integer.MIN\_VALUE до Integer.MAX\_VALUE).
2. Используйте более широкий тип данных. Если возможно, используйте тип данных большего размера (например, long) для хранения и обработки потенциально больших значений, чтобы избежать переполнения и усечения.
3. Выполняйте явную проверку на переполнение. Перед выполнением арифметических операций с ненадежными int значениями, которые могут привести к переполнению, добавьте проверки, чтобы убедиться, что результат не выйдет за пределы допустимого диапазона int.