FB.INT\_VACUOUS\_COMPARISON

Детектор определяет ошибку, в ходе которой существует целочисленное сравнение, которое всегда возвращает одно и то же значение (например, x <= Integer.MAX\_VALUE).

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Такие сравнения не оказывают никакого влияния на логику программы и могут быть признаком лишнего или ошибочного кода. |
| Комментарий | Проверьте условие сравнения. Если оно всегда возвращает одно значение, возможно, оно лишнее, и код следует упростить, удалив это сравнение. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(int x) {  if (x <= Integer.MAX\_VALUE) {  System.out.println("x <= Integer.MAX\_VALUE is always true.");  }  if (x >= Integer.MIN\_VALUE) {  System.out.println("x >= Integer.MIN\_VALUE is always true.");  }  x = 42;  if (x != 42) {  System.out.println("This will never happen.");  }  } | public static void correctTest(int x, int y) {  if (x < Integer.MAX\_VALUE) {  System.out.println("x is less than Integer.MAX\_VALUE.");  }  if (x != y) {  System.out.println("x is not equal to y.");  }  if (x > 10 && x < 100) {  System.out.println("x is between 10 and 100.");  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Удалите избыточное сравнение:** Если сравнение всегда возвращает одно и то же значение, и это не является частью более сложной логики, просто удалите оператор if (или другое условное выражение) и оставьте только код, который должен выполняться (или не выполняться) всегда.
2. **Пересмотрите логику:** Подумайте, что вы на самом деле хотели проверить. Возможно, вы допустили ошибку в условии сравнения или использовали неправильные константы.
3. **Проверьте контекст:** Рассмотрите, как значение переменной определяется и изменяется в данном участке кода. Возможно, в другом контексте сравнение имело бы смысл.
4. **Будьте внимательны при рефакторинге:** При изменении кода обращайте внимание на такие сравнения, которые выглядят подозрительно, и убедитесь, что они действительно необходимы.