FB.INT\_VACUOUS\_BIT\_OPERATION

Детектор определяет ошибку, в ходе которой целочисленная битовая операция (и, или, или исключающее или), которая не выполняет никакой полезной работы (например, v & 0xffffffff).

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Такие битовые операции не изменяют значение переменной и могут указывать на логическую ошибку или избыточный код, затрудняющий понимание. |
| Комментарий | Следует удалить бесполезную операцию или заменить ее на осмысленный вариант, если она была задумана для маскирования или приведения типа. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  int x = 42;  // результат всегда x  int y = x & 0xFFFFFFFF;  int z = x | 0;  int w = x ^ 0;  System.out.println("x & 0xFFFFFFFF = " + y);  System.out.println("x | 0 = " + z);  System.out.println("x ^ 0 = " + w);  } | public static void correctTest() {  int x = 42;  int y = x & 0xFF; // младший байт числа  int z = x | 0xFF; // устанавливаем младший байт в 1  int w = x ^ 0xFF; // инвертируем младший байт  System.out.println("x & 0xFF = " + y);  System.out.println("x | 0xFF = " + z);  System.out.println("x ^ 0xFF = " + w);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Удалите бесполезную операцию:** Если битовая операция не выполняет никакой полезной работы, просто удалите соответствующую строку кода.
2. **Замените на корректную операцию:** Если операция предназначалась для выполнения определенной задачи (например, маскирования, установки или инвертирования битов), замените ее на правильную битовую операцию с соответствующей маской.
3. **Проверьте логику:** Убедитесь, что удаление или замена операции не нарушит логику программы. Возможно, изначально задумка была другой.
4. **Будьте внимательны при рефакторинге:** При рефакторинге кода обращайте внимание на такие избыточные операции и удаляйте их для улучшения читаемости и потенциальной производительности.