FB.IM\_BAD\_CHECK\_FOR\_ODD

Детектор определяет ошибку, в ходе которой в коде используется x % 2 == 1, чтобы проверить, является ли значение нечетным, но это не работает для отрицательных чисел (например, (-5) % 2 == -1). Если этот код предназначен для проверки на нечетность, рассмотрите возможность использования (x & 1) == 1 или x % 2 != 0.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | При проверке числа на нечетность с x % 2 == 1 могут возникнуть ошибки для отрицательных чисел, поскольку остаток от деления может быть -1 (например, -5 % 2 == -1), что приведет к некорректным результатам. |
| Комментарий | Используйте более надежные способы проверки нечетности:  (x & 1) != 0 или x % 2 != 0 вместо x % 2 == 1.  Это обеспечит корректную работу для всех целых чисел, включая отрицательные. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  int[] numbers = {5, -5, 8, -8};  for (int num : numbers) {  // не работает для отрицательных чисел  if (num % 2 == 1) {  System.out.println(num);  }  }  } | public static void correctTest() {  int[] numbers = {5, -5, 8, -8};  for (int num : numbers) {  if ((num & 1) == 1) {  System.out.println(num);  }  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Замените x % 2 == 1 на (x & 1) == 1 (или != 0):** Используйте побитовую операцию И с единицей для проверки младшего бита. Если младший бит установлен (равен 1), число является нечетным.
2. **Замените x % 2 == 1 на x % 2 != 0:** Проверьте, что остаток от деления на 2 не равен нулю. Это условие будет истинным для всех нечетных чисел (положительных и отрицательных).
3. **Проверьте весь код:** Найдите все места, где используется x % 2 == 1 для проверки на нечетность, и замените это выражение на один из корректных вариантов.
4. **Учитывайте контекст:** В некоторых редких случаях, если по логике программы вы работаете только с неотрицательными числами, x % 2 == 1 может быть допустимым, но в общем случае лучше использовать более надежные подходы.