FB.QF\_QUESTIONABLE\_FOR\_LOOP

Детектор определяет ошибку, в ходе которой другая переменная инициализируется и проверяется циклом for.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Использование переменной в цикле for, которая инициализируется и проверяется внутри этого цикла, может затруднить понимание логики программы и повлиять на читаемость кода. |
| Комментарий | Рекомендуется явно инициализировать и проверять переменные за пределами цикла for, если они не служат исключительно для итерации. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  for (int i = 0, j = 10; i < 10; i++, j--) {  System.out.println("i = " + i + ", j = " + j);  }  } | public static void correctTest() {  int j = 10;  for (int i = 0; i < 10; i++, j--) {  System.out.println("i = " + i + ", j = " + j);  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Ограничьте использование цикла for одной основной переменной, контролирующей итерацию.
2. Инициализируйте и обновляйте дополнительные переменные вне цикла или внутри его тела, если их связь с основной переменной не является тривиальной.
3. Используйте цикл while для более сложной логики управления, включающей несколько переменных.
4. Рассмотрите использование итераторов или других средств для обхода коллекций и структур данных.
5. Сделайте код максимально читаемым и понятным, даже если исходный вариант с несколькими переменными в for работает корректно.