FB.IIO\_INEFFICIENT\_INDEX\_OF

Детектор определяет ошибку, в ходе которой код передает константную строку длины 1 в String.indexOf(). Более эффективно использовать целочисленную реализацию String.indexOf(). ф. е. вызовите myString.indexOf('.') вместо myString.indexOf(".").

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Использование String.indexOf() с константной строкой длины 1 может быть менее эффективным, так как метод должен работать с объектом строки, а не с примитивным символом. |
| Комментарий | Для улучшения производительности вместо передачи строки длиной 1 в метод indexOf(), используйте прямой символ, например: myString.indexOf('.') вместо myString.indexOf("."). |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  String myString = "Hello, world!";  // передача строки длиной 1 в String.indexOf()  int index = myString.indexOf(".");  System.out.println("Incorrect Result: " + index);  } | public static void correctTest() {  String myString = "Hello, world!";  // передача символа в String.indexOf()  int index = myString.indexOf('.');  System.out.println("Correct Result: " + index);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Замените строковые литералы длиной 1 на символьные литералы:** Если аргументом в indexOf() или lastIndexOf() является строковый литерал, состоящий из одного символа (например, ".", "a", ","), замените его на соответствующий символьный литерал (например, '.', 'a', ',').
2. **Используйте переменные типа char:** Если символ для поиска хранится в переменной типа String длиной 1, извлеките символ из строки с помощью метода charAt(0) и передайте этот символ в indexOf() или lastIndexOf().
3. **Проверьте все вызовы indexOf() и lastIndexOf():** Найдите все места в коде, где используются эти методы, и убедитесь, что аргумент является символом (типа char или символьным литералом) вместо строки длиной 1.
4. **Обратите внимание на константы:** Если вы используете константы для поиска одиночных символов, убедитесь, что они определены как char, а не как String.