FB.DMI\_USELESS\_SUBSTRING

Детектор определяет ошибку, в ходе которой код вызывает substring(0) для строки, которая возвращает исходное значение.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE440 |
| CVE | — |
| Возможные последствия | Использование substring(0) не изменяет строку и избыточно расходует процессорное время, снижая производительность. |
| Комментарий | Просто используйте строку напрямую без вызова substring(0), так как этот вызов не выполняет никаких полезных действий. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  String str = "Hello, world!";  String uselessSubstring = str.substring(0);  System.out.println(uselessSubstring);  } | public static void correctTest() {  String str = "Hello, world!";  String correctString = str;  System.out.println(correctString);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Удалите вызов .substring(0):** Найдите все места в коде, где вызывается метод substring(0) у строковой переменной или литерала, и просто замените это выражение на саму строку.
2. **Проверьте логику:** Убедитесь, что вызов substring(0) не был ошибочной попыткой получить какую-либо другую подстроку. Если предполагалось другое поведение, исправьте параметры метода substring().
3. **Сосредоточьтесь на чистоте кода:** Удаление избыточных операций делает код более лаконичным и легким для понимания.