FB.URF\_UNREAD\_PUBLIC\_OR\_PROTECTED\_FIELD

Детектор определяет ошибку, в ходе которой поле никогда не читается. Поле является общедоступным или защищенным, поэтому, возможно, оно предназначено для использования с классами, не рассматриваемыми как часть анализа. Если нет, рассмотрите возможность удаления его из класса.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Если поле объявлено как public или protected, но никогда не читается, это может указывать на избыточность в коде. |
| Комментарий | Следует проверить, действительно ли поле должно быть доступным. Если оно не используется, его стоит удалить. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public int unusedPublicField;  protected String unusedProtectedField;  public static void incorrectTest() {  var example = new URF\_UNREAD\_PUBLIC\_OR\_PROTECTED\_FIELD();  example.unusedPublicField = 10; // Присваивание, но поле не используется  example.unusedProtectedField = "Hello";  } | public int usedPublicField;  protected String usedProtectedField;  public static void correctTest() {  var example = new URF\_UNREAD\_PUBLIC\_OR\_PROTECTED\_FIELD();  example.usedPublicField = 10;  example.usedProtectedField = "Hello";  System.out.println( example.usedPublicField);  System.out.println(example.usedProtectedField);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Удалите объявление поля**, если вы уверены, что оно не используется нигде за пределами текущего анализа кода. Это самый простой способ устранить избыточность.
2. **Проверьте, действительно ли поле должно быть public или protected.** Если поле используется только внутри класса или его внутренних классов, сделайте его private.
3. **Если поле предназначено для использования в других модулях или библиотеках, которые не анализируются, оставьте его.** Добавьте комментарий, объясняющий его предполагаемое использование за пределами текущего контекста.
4. **Рассмотрите возможность использования геттера (getter) для контролируемого доступа к полю.** Даже если прямое чтение поля не происходит в анализируемом коде, геттер может вызываться извне.