FB.UWF\_UNWRITTEN\_PUBLIC\_OR\_PROTECTED\_FIELD

Детектор определяет ошибку, в ходе которой в общедоступное/защищенное поле не было обнаружено никаких записей. Все его чтения будут возвращать значение по умолчанию. Проверьте наличие ошибок (нужно ли было его инициализировать?), или удалите, если он бесполезен.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Если в общедоступное или защищенное поле не записывается значение, оно всегда будет содержать значение по умолчанию для его типа (например, null для ссылочных типов или 0 для числовых типов). |
| Комментарий | Если поле должно содержать значение, но оно не инициализируется, это может быть ошибкой, и стоит рассмотреть возможность добавления инициализации. |

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 UWF\_UNWRITTEN\_PUBLIC\_OR\_PROTECTED\_FIELD();  System.out.println("unusedPublicField = " + example.unusedPublicField);  System.out.println("unusedProtectedField = " + example.unusedProtectedField);  } | public int usedPublicField;  protected String usedProtectedField;  public static void correctTest() {  var example = new UWF\_UNWRITTEN\_PUBLIC\_OR\_PROTECTED\_FIELD();  example.usedPublicField = 100;  example.usedProtectedField = "Test";  System.out.println("usedPublicField = " + example.usedPublicField);  System.out.println("usedProtectedField = " + example.usedProtectedField);  } |

Методы устранения уязвимости при срабатывании детектора:

1. Найдите место, где полю должно присваиваться значение. Определите, в каком классе или внешнем коде предполагается запись в это публичное или защищенное поле.
2. Реализуйте запись в поле. Добавьте код для присвоения полю нужного значения в соответствующем месте.
3. Проверьте, должно ли поле быть константой. Если поле должно содержать неизменяемое значение, объявите его как public static final (или protected static final) и инициализируйте при объявлении.
4. Если поле предназначено для использования только внутри класса или его подклассов, рассмотрите возможность сделать его private (или protected для подклассов) и предоставить методы доступа (сеттеры). Это улучшит инкапсуляцию.