FB.UWF\_UNWRITTEN\_FIELD

Детектор определяет ошибку, в ходе которой поле никогда не записывается. 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 | Поле никогда не инициализируется, поэтому все его чтения возвращают значение по умолчанию (null, 0, false и т. д.), что может привести к ошибкам в логике программы, включая NullPointerException. |
| Комментарий | Неинициализированное поле может указывать на ошибку в коде или его ненужность. Следует проверить, требуется ли его инициализация, или удалить поле, если оно не используется. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class Test {  private String value;  public Test() {  }  public void printValue() {  // NullPointerException!  System.out.println(value.toUpperCase());  }  }  } | public static void correctTest() {  class Test {  private String value;  public Test(String value) {  this.value = value;  }  public void printValue() {  System.out.println(value.toUpperCase());  }  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Найдите место, где поле должно инициализироваться. Определите, где и когда полю должно быть присвоено значимое значение (например, в конструкторе, сеттере или другом методе).
2. Реализуйте инициализацию поля. Добавьте код для присвоения полю нужного значения в определенном месте.
3. Проверьте, не является ли поле константой. Если поле должно содержать постоянное значение, объявите его как final и инициализируйте при объявлении.
4. Если поле используется только для временных вычислений внутри одного метода, рассмотрите возможность использования локальной переменной. Это уменьшит область видимости и потенциальные ошибки.
5. Если поле никогда не используется (не читается), удалите его. Наличие неиспользуемых полей засоряет код.