FB.SS\_SHOULD\_BE\_STATIC

Детектор определяет ошибку, в ходе которой класс содержит последнее поле экземпляра, которое инициализируется статическим значением времени компиляции. Рассмотрите возможность сделать поле статическим.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Если поле инициализируется значением, которое не зависит от конкретного экземпляра класса, его хранение как нестатического приводит к избыточному расходу памяти, так как каждому экземпляру класса будет присвоено собственное значение этого поля. |
| Комментарий | Если поле не зависит от состояния экземпляра и всегда инициализируется статическим значением, следует сделать его статическим. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class InnerClass {  // Поле экземпляра  private final String staticValue = "value";  public void printValue() {  System.out.println(staticValue);  }  }  InnerClass inner = new InnerClass();  inner.printValue();  } | public static void correctTest() {  class InnerClass {  // Статическое поле  private static final String staticValue = "value";  public void printValue() {  System.out.println(staticValue);  }  }  InnerClass inner = new InnerClass();  inner.printValue();  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Проверьте, действительно ли поле инициализируется статическим значением времени компиляции.** Убедитесь, что присваиваемое значение является константой, литералом или результатом статического выражения.
2. **Добавьте ключевое слово static к объявлению поля.** Это сделает поле полем класса, а не полем экземпляра.
3. **Убедитесь, что код, использующий это поле, продолжает работать корректно после того, как оно стало статическим.** Поскольку статические поля доступны через имя класса (например, ClassName.fieldName), а также через экземпляры класса, в большинстве случаев изменение не повлияет на существующий код. Однако если вы полагались на то, что поле является уникальным для каждого экземпляра (что противоречит условию обнаружения этой ошибки), вам потребуется пересмотреть эту логику.
4. **Рассмотрите возможность сделать поле public static final, если его значение является константой и предназначено для общего доступа.** Это является стандартной практикой для определения констант в Java.