FB.DLS\_DEAD\_LOCAL\_STORE\_SHADOWS\_FIELD

Детектор определяет ошибку, в ходе которой инструкция присваивает значение локальной переменной, но это значение не считывается и не используется ни в одной последующей инструкции. Часто это указывает на ошибку, поскольку вычисленное значение никогда не используется. Существует поле с тем же именем, что и у локальной переменной.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE563 |
| CVE | — |
| Возможные последствия | Присваивание значения локальной переменной, которое не используется, и которое затеняет поле с тем же именем, может привести к путанице и сложностям в дальнейшем сопровождении кода. |
| Комментарий | Рекомендуется избегать такой практики, чтобы улучшить читаемость и предотвратить потенциальные ошибки. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| private String field = "Поле класса";  public void incorrectTest() {  // Локальная переменная с тем же именем, что и поле  String field = "Локальная переменная";  field = null;  // ...  } | private String field = "Поле класса";  public void correctTest() {  // Изменение поля класса напрямую  this.field = "Новая строка";  System.out.println(field);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Измените имя локальной переменной:** Переименуйте локальную переменную так, чтобы ее имя не совпадало с именем поля класса. Это устранит затенение и сделает код более понятным.
2. **Удалите неиспользуемое присваивание:** Если локальная переменная присваивается, но ее значение нигде не используется, просто удалите строку с присваиванием. Если сама переменная больше не нужна, удалите и ее объявление.
3. **Используйте this для доступа к полю:** Если вы действительно хотели работать с полем класса, используйте ключевое слово this для явного указания на поле (например, this.field = ...).
4. **Пересмотрите логику:** Убедитесь, что вы правильно понимаете, с какой переменной вы хотите работать. Затенение часто является признаком ошибки в логике.
5. **Старайтесь избегать затенения:** В целом, рекомендуется избегать ситуаций, когда локальные переменные имеют те же имена, что и поля класса, чтобы предотвратить путаницу и улучшить читаемость кода.