FB.DLS\_DEAD\_LOCAL\_STORE

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.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE563 |
| CVE | — |
| Возможные последствия | Присваивание значений локальной переменной, которые затем не используются, приводит к избыточному вычислению и расходу ресурсов. |
| Комментарий | Рекомендуется удалить такие присваивания, чтобы избежать ненужных операций и улучшить читаемость кода. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class MyClass {  void doSomething() {  // Значение переменной `x` нигде не используется  int x = 10;  x = 20;  }  }  MyClass myClass = new MyClass();  myClass.doSomething();  } | public static void correctTest() {  class MyClass {  void doSomething() {  int x = computeValue(); // Значение переменной используется  System.out.println("Computed value: " + x);  }  int computeValue() {  return 42;  }  }  MyClass myClass = new MyClass();  myClass.doSomething();  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Удалите неиспользуемое присваивание:** Если локальная переменная присваивается, но ее значение нигде не используется, просто удалите строку с присваиванием.
2. **Удалите неиспользуемую переменную:** Если локальная переменная объявляется и ей присваивается значение, но она нигде не считывается, удалите объявление и присваивание этой переменной.
3. **Используйте переменную:** Если вы обнаружили, что переменная должна была быть использована, найдите место в коде, где это значение необходимо, и добавьте соответствующее использование.
4. **Проверьте логику:** Иногда неиспользуемое присваивание может указывать на ошибку в логике программы. Возможно, вы забыли использовать вычисленное значение или неправильно организовали поток выполнения.
5. **Обратите внимание на побочные эффекты:** В редких случаях присваивание может выполняться ради побочных эффектов (например, вызова конструктора). Однако такой код обычно считается плохим стилем. Если побочные эффекты действительно необходимы, сделайте это более явным.