FB.DLS\_DEAD\_LOCAL\_STORE\_IN\_RETURN

Детектор определяет ошибку, в ходе которой оператор присваивает значение локальной переменной в операторе возврата. Это назначение не имеет никакого эффекта. Пожалуйста, убедитесь, что это утверждение соответствует действительности.

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 непосредственно перед оператором возврата не оказывает никакого эффекта, так как переменная сразу же используется в возвращаемом значении. |
| Комментарий | Рекомендуется избавиться от такого присваивания, чтобы повысить эффективность и читаемость кода. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class MyClass {  int computeValue() {  int x = 10;  // не имеет эффекта  return (x = 20);  }  }  MyClass myClass = new MyClass();  System.out.println(myClass.computeValue());  } | public static void correctTest() {  class MyClass {  int computeValue() {  int x = 10;  x = 20;  return x;  }  }  MyClass myClass = new MyClass();  System.out.println(myClass.computeValue());  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Удалите присваивание в операторе return:** Если вы присваиваете значение локальной переменной непосредственно в операторе return, вынесите это присваивание на отдельную строку перед оператором return.
2. **Верните непосредственно значение:** Если нет необходимости сохранять присваиваемое значение в локальной переменной для дальнейшего использования в методе, верните непосредственно вычисляемое значение.
3. **Проверьте логику:** Убедитесь, что такое присваивание не было попыткой выполнить какое-либо действие после возврата значения (что невозможно). Если такая логика предполагалась, необходимо пересмотреть структуру метода.
4. **Сделайте код более читаемым:** Избегайте сложных выражений в операторе return, особенно с побочными эффектами, такими как присваивание. Разбейте сложные операции на несколько строк для лучшей читаемости.