FB.SA\_LOCAL\_DOUBLE\_ASSIGNMENT

Детектор определяет ошибку, в ходе которой метод содержит двойное присвоение локальной переменной.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Двойное присваивание локальной переменной приводит к неэффективному использованию памяти и снижает читаемость кода. |
| Комментарий | Не следует присваивать переменной значение, если оно не используется перед следующим присваиванием. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| def multiply(x, by):  return x \* by  def square\_root(x):  return x\*\*0.5  x = [1, 3, 7],  x\_squared = [square\_root(multiply(x\_i, 3)) for x\_i in x] | def multiply(x, by):  return x \* by  def square\_root(x):  return x\*\*0.5  x = [1, 3, 7]  x\_squared = [square\_root(multiply(x\_i, 3)) for x\_i in x] |

Методы устранения уязвимости при срабатывании детектора:

1. **Проанализируйте код метода, содержащего двойное присваивание локальной переменной.** Определите, используется ли значение, присвоенное переменной в первый раз, до второго присваивания.
2. **Удалите первое присваивание, если его значение нигде не используется до того, как переменная перезаписывается.** Это сделает код более чистым и эффективным.
3. **Если первое присваивание было сделано с определенной целью, убедитесь, что эта цель достигается.** Возможно, существует логическая ошибка, и вместо прямого перезаписывания переменной необходимо использовать первое значение каким-либо образом.
4. **Проверьте, не является ли двойное присваивание результатом ошибки копирования и вставки кода.** Убедитесь, что обе операции присваивания предназначены для одной и той же переменной и выполняются с нужными значениями.
5. **Рассмотрите возможность рефакторинга кода, чтобы избежать двойного присваивания, если это улучшит читаемость и логику метода.** Например, можно инициализировать переменную нужным значением только один раз.
6. **Если двойное присваивание является частью сложной логики, тщательно проверьте порядок операций и убедитесь, что промежуточное значение переменной не является важным.**