FB.SF\_DEAD\_STORE\_DUE\_TO\_SWITCH\_FALLTHROUGH

Детектор определяет ошибку, в ходе которой значение, сохраненное в предыдущем случае переключения, здесь перезаписывается из-за сбоя переключения. Вполне вероятно, что вы забыли поставить перерыв или возврат в конце предыдущего дела.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE484 |
| CVE | — |
| Возможные последствия | Перезапись значения без явного намерения может привести к некорректной работе программы, поскольку используется не то значение, которое ожидалось. Это может вызвать непредсказуемое поведение и логические ошибки. |
| Комментарий | Необходимо проверить, требуется ли выполнение следующего case. Если нет, необходимо добавить break или return, чтобы предотвратить нежелательную перезапись переменной. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(int input) {  switch (input) {  case 1:  // ...  case 2:  // Падение сквозь case  }  } | public static void correctTest(int input) {  switch (input) {  case 1:  // ...  break;  case 2:  // …  break;  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Внимательно изучите оператор switch и определите, является ли намеренным "проваливание" из одного case в другой. Проверьте логику обработки каждого case.
2. Если выполнение не должно переходить в следующий case после завершения текущего, добавьте оператор break; в конце блока кода для этого case. Оператор break прервет выполнение оператора switch и передаст управление следующей за ним инструкции.
3. Если в конце case происходит возврат значения из метода, используйте оператор return; вместо break;.
4. Если в конце case выбрасывается исключение, используйте оператор throw new Exception(...);.
5. В редких случаях, когда "проваливание" является намеренным и логически обоснованным, добавьте комментарий (// fall through) в конце case, чтобы явно указать на это намерение. Это поможет другим разработчикам понять код и избежать случайного добавления оператора break. Однако намеренное "проваливание" следует использовать с осторожностью, так как оно может сделать код менее читаемым.
6. Проверьте, не является ли перезаписываемая переменная важной для дальнейшей логики программы. Убедитесь, что использование перезаписанного значения не приведет к ошибкам.