FB.IS2\_INCONSISTENT\_SYNC

Доступ к полям этого класса осуществляется непоследовательно с точки зрения синхронизации. В этом отчете об ошибке указано, что детектор шаблонов ошибок определил, что класс содержит смесь заблокированных и разблокированных доступов, он не аннотирован как javax.annotation.concurrent.NotThreadSafe, по крайней мере один заблокированный доступ был выполнен одним из собственных методов класса, и количество несинхронизированных обращений к полям (чтение и запись) составляло не более трети всех обращений, при этом вес операций записи в два раза превышал вес операций чтения.

Типичная ошибка, соответствующая этому шаблону, — забывание синхронизировать один из методов в классе, который должен быть потокобезопасным.

Можно выбрать узлы с надписью «Несинхронизированный доступ», чтобы отобразить места кода, в которых детектор считает, что доступ к полю осуществлялся без синхронизации.

Следует обратить внимание, что в этом детекторе существуют различные источники погрешностей; например, детектор не может статически обнаружить все ситуации, в которых удерживается замок. Кроме того, даже если детектор точно различает заблокированный и незаблокированный доступ, рассматриваемый код все равно может быть правильным.

Таблица №1 – Основные характеристики детектора FB.IS2\_INCONSISTENT\_SYNC

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Concurrency Vulnerability (Race Condition) |
| CWE | CWE366 |
| CVE | --- |
| Возможные последствия | Состояние гонки (race condition) при доступе к общим ресурсам; частично видимые или поврежденные данные; нарушение инвариантов класса; трудноуловимые ошибки, проявляющиеся только под нагрузкой |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  BankAccountIncorrect account = new BankAccountIncorrect();  account.deposit(1000);    new Thread(() -> {  // Может прочитать устаревшее значение баланса  System.out.println("Incorrect balance: " + account.getBalance());  }).start();  } | public static void correctTest() {  BankAccountCorrect account = new BankAccountCorrect();  account.deposit(1000);    new Thread(() -> {  System.out.println("Correct balance: " + account.getBalance());  }).start();  } |