FB.WA\_NOT\_IN\_LOOP

Ошибка "Wait not in loop" или "Wait not in loop in" указывает на то, что операция wait() используется внутри кода, но без обертывания её в цикл. Обычно, использование wait() должно быть в цикле, чтобы предотвратить "ложные пробуждения" (spurious wake-ups) и правильно обрабатывать условия, по которым поток должен продолжить выполнение.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Использование wait() вне цикла может привести к некорректному возобновлению потока из-за ложных пробуждений (spurious wake-ups) или уведомлений по неподходящему условию. |
| Комментарий | Рекомендуется всегда вызывать wait() внутри цикла с проверкой условия ожидания. Это гарантирует, что поток возобновит выполнение только тогда, когда действительно наступит требуемое состояние. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| class SharedResourceIncorrect {  private boolean ready = false;  public synchronized void waitForReady() throws InterruptedException {  wait(); // вызов wait() без цикла  System.out.println("Condition met, proceeding...");  }  public synchronized void setReady() {  ready = true;  notifyAll();  }  } | class SharedResourceCorrect {  private boolean ready = false;  public synchronized void waitForReady() throws InterruptedException {  while (!ready) { // проверка условия в цикле  wait();  }  System.out.println("Condition met, proceeding...");  }  public synchronized void setReady() {  ready = true;  notifyAll();  }  } |