FB.NO\_NOTIFY\_NOT\_NOTIFYALL

Этот метод вызывает notify(), а не notifyAll(). Мониторы Java часто используются в различных условиях. Вызов notify() пробуждает только один поток, а это означает, что пробуждённый поток может быть не тем, который ожидает условия, которое только что выполнил вызывающий объект.

Таблица №1 – Основные характеристики детектора FB.NO\_NOTIFY\_NOT\_NOTIFYALL

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Concurrency Vulnerability |
| CWE | --- |
| CVE | --- |
| Возможные последствия | Потерянные уведомления (missed notifications); Deadlock-и из-за неразбуженных потоков; голодание (starvation) некоторых потоков; непредсказуемое поведение в сложных условиях гонки. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  TaskProcessorIncorrect processor = new TaskProcessorIncorrect();    // Создаем несколько ожидающих потоков  for (int i = 0; i < 3; i++) {  new Thread(() -> {  try {  processor.waitForCondition();  System.out.println("Thread " + Thread.currentThread().getId() +  " awakened (may miss with notify())");  } catch (InterruptedException e) {  Thread.currentThread().interrupt();  }  }).start();  }    // Сигнализируем - может разбудить только один поток  new Thread(() -> {  processor.signalCondition();  System.out.println("Signaled with notify()");  }).start();  } | public static void correctTest() {  TaskProcessorCorrect processor = new TaskProcessorCorrect();    for (int i = 0; i < 3; i++) {  new Thread(() -> {  try {  processor.waitForCondition();  System.out.println("Thread " + Thread.currentThread().getId() +  " correctly awakened");  } catch (InterruptedException e) {  Thread.currentThread().interrupt();  }  }).start();  }    new Thread(() -> {  processor.signalCondition();  System.out.println("Signaled with notifyAll()");  }).start();  } |