FB.UW\_UNCOND\_WAIT

Ошибка "unconditional wait" (безусловное ожидание) указывает на то, что в вашем коде присутствует операция ожидания (например, использование метода wait() в Java), которая выполняется без проверки какого-либо условия. Это может привести к блокировке (замораживанию) выполнения программы, если не выполнены определенные условия для продолжения работы.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Безусловное ожидание без проверки условия может привести к зависанию потока, если notify() или notifyAll() никогда не будут вызваны. |
| Комментарий | Рекомендуется использовать wait() внутри цикла с проверкой условия, чтобы избежать зависания потока. Например, применяя while (condition) { wait(); }, можно гарантировать, что поток продолжит выполнение только при выполнении необходимых условий. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() throws InterruptedException {  class Test {  private final Object lock = new Object();  public void work() throws InterruptedException {  synchronized (lock) {  // ожидание без условия  lock.wait();  }  }  }  Test test = new Test();  test.work(); // Ошибочный вызов (может зависнуть)  } | public static void correctTest() throws InterruptedException {  class Test {  private final Object lock = new Object();  private boolean ready = false;  public void doWork() throws InterruptedException {  synchronized (lock) {  // Проверка условия перед ожиданием  while (!ready) {  lock.wait();  }  // ...  }  }  public void markReady() {  synchronized (lock) {  ready = true;  lock.notifyAll();  }  }  }  Test test = new Test();  Thread worker = new Thread(() -> {  try {  test.doWork();  } catch (InterruptedException e) {  Thread.currentThread().interrupt();  }  });  worker.start();  Thread.sleep(1000);  test.markReady(); // wait() корректно завершится  worker.join();  } |