FB.DC\_DOUBLECHECK

Этот метод может содержать экземпляр блокировки с двойной проверкой. Эта идиома неверна с точки зрения семантики модели памяти Java.

Таблица №1 – Основные характеристики детектора FB.DC\_DOUBLECHECK

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Concurrency Vulnerability (Race Condition) |
| CWE | CWE609 |
| CVE | --- |
| Возможные последствия | Возможность создания нескольких экземпляров singleton-объекта; частично инициализированные объекты в многопоточной среде; непредсказуемое поведение программы; нарушение гарантий singleton-паттерна |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  new Thread(() -> {  SingletonIncorrect instance = SingletonIncorrect.getInstance();  System.out.println("Thread 1 instance: " + instance);  }).start();    new Thread(() -> {  SingletonIncorrect instance = SingletonIncorrect.getInstance();  System.out.println("Thread 2 instance: " + instance);  }).start();  } | public static void correctTest() {  new Thread(() -> {  SingletonCorrect instance = SingletonCorrect.getInstance();  System.out.println("Thread 1 correct instance: " + instance);  }).start();    new Thread(() -> {  SingletonBest instance = SingletonBest.getInstance();  System.out.println("Thread 2 best instance: " + instance);  }).start();  } |