FB.STCAL\_STATIC\_CALENDAR\_INSTANCE

Несмотря на то, что JavaDoc не содержит никаких намеков на это, календари по своей сути небезопасны для многопоточного использования. Совместное использование одного экземпляра за границами потоков без надлежащей синхронизации приведет к нестабильному поведению приложения.

В версии 1.4 проблемы возникают реже, чем в Java 5, где вы, вероятно, увидите случайные исключения ArrayIndexOutOfBoundsException или IndexOutOfBoundsExceptions в sun.util.calendar.BaseCalendar.getCalendarDateFromFixedDate().

Также могут возникнуть проблемы с сериализацией.

Рекомендуется использовать поле экземпляра.

Таблица №1 – Основные характеристики детектора FB.STCAL\_STATIC\_CALENDAR\_INSTANCE

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Concurrency Vulnerability |
| CWE | CWE366 |
| CVE | --- |
| Возможные последствия | Гонки данных (race conditions) в многопоточной среде; неправильные даты/время в расчетах; трудноуловимые ошибки, проявляющиеся только под нагрузкой; нарушение потокобезопасности в серверных приложениях |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  Runnable task = () -> {  String date = DateUtilsIncorrect.formatCurrentDate();  System.out.println(Thread.currentThread().getName() + ": " + date);  };  // В многопоточной среде могут быть некорректные результаты  new Thread(task, "Thread-1").start();  new Thread(task, "Thread-2").start();  } | public static void correctTest() {  Runnable task = () -> {  String date = DateUtilsCorrect.formatCurrentDate();  System.out.println(Thread.currentThread().getName() + ": " + date);  };  new Thread(task, "Thread-1").start();  new Thread(task, "Thread-2").start();  } |