FB.DL\_SYNCHRONIZATION\_ON\_SHARED\_CONSTANT

Детектор определяет ошибку, в ходе которой код синхронизируется по строковому литералу.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Синхронизация на строковом литерале может привести к неожиданному поведению, так как строковые литералы кэшируются и могут быть разделяемыми между разными частями кода. Это может привести к ложному взаимному исключению, состоянию гонки или взаимоблокировке при работе нескольких потоков. |
| Комментарий | Для безопасной синхронизации следует использовать отдельный объект-монитор (private final Object lock = new Object();), а не строковые литералы или другие кэшируемые константы, чтобы избежать проблем с совместным использованием объектов. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class Test {  // строковый литерал интернируется  private static final String LOCK = "LOCK";  public void method() {  // возможное разделение блокировки между разными классами  synchronized (LOCK) {  System.out.println("Executing critical section");  }  }  }  } | public static void correctTest() {  class Test {  // независимый объект  private final Object lock = new Object();  public void method() {  // Гарантированно корректная синхронизация  synchronized (lock) {  System.out.println("Executing critical section");  }  }  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Создайте приватный объект-монитор:** Вместо использования строковых литералов для синхронизации, создайте приватный экземпляр Object специально для этой цели: private final Object lock = new Object();.
2. **Используйте созданный объект в synchronized блоках:** Замените синхронизацию на строковый литерал на синхронизацию по созданному приватному объекту-монитору.
3. **Избегайте синхронизации на строковых литералах:** Никогда не используйте строковые литералы в качестве аргумента для synchronized.
4. **Будьте осторожны с константами:** Даже если строковая константа не является литералом (например, new String("LOCK")), но является static final, рассмотрите возможность непреднамеренного совместного использования, особенно если она доступна другим классам. Лучшим решением является использование приватных объектов-мониторов.
5. **Проверьте все места синхронизации:** Убедитесь, что во всем коде, где требуется синхронизация, используются выделенные приватные объекты-мониторы.