FB.DL\_SYNCHRONIZATION\_ON\_BOXED\_PRIMITIVE

Детектор определяет ошибку, в ходе которой код синхронизируется с помощью упакованной примитивной константы, например целого числа.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Синхронизация на объекте-обертке примитивного типа (например, Integer или Double) может привести к непредсказуемому поведению, поскольку такие объекты кэшируются и могут быть разделяемыми между разными частями кода. Это может привести к ложному взаимному исключению, состоянию гонки или взаимоблокировке. |
| Комментарий | Для безопасной синхронизации следует использовать отдельный объект-монитор (private final Object lock = new Object();), а не объекты-обертки примитивных типов, чтобы избежать проблем, связанных с их кэшированием и совместным использованием. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class Test {  // автоупаковка примитива  private final Integer lock = 42;  public void method() {  // синхронизация на объекте-обертке  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 объектов-оберток примитивных типов 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 объект-обертку на синхронизацию по созданному приватному объекту-монитору.
3. **Избегайте синхронизации на результатах valueOf():** Не используйте Integer.valueOf(value), Double.valueOf(value) и другие подобные методы для получения объекта-монитора, так как эти методы могут возвращать кэшированные объекты.
4. **Проверьте все места синхронизации:** Убедитесь, что во всем коде, где требуется синхронизация, используются выделенные приватные объекты-мониторы.
5. **Будьте осторожны с автобоксингом:** Автоматическое преобразование примитивов в обертки может скрыть случаи, когда происходит синхронизация на кэшированных объектах. Внимательно анализируйте код, где используются synchronized блоки и объекты-обертки.