FB.SIC\_INNER\_SHOULD\_BE\_STATIC

Детектор определяет ошибку, в ходе которой класс является внутренним классом, но не использует встроенную ссылку на объект, который его создал. Эта ссылка увеличивает размер экземпляров класса и может сохранять ссылку на объект-создатель дольше, чем необходимо. Если возможно, класс следует сделать статическим.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Необоснованное использование нестатического (не static) внутреннего класса приводит к тому, что он неявно хранит ссылку на экземпляр внешнего класса. Это увеличивает потребление памяти, усложняет управление временем жизни объектов и может привести к утечкам памяти, если внутренний класс используется асинхронно или передаётся за пределы внешнего объекта. |
| Комментарий | Если внутренний класс не использует поля и методы внешнего класса, его следует объявить static. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class Outer {  class InnerClass { // Нестатический внутренний класс  public void doSomething() {  System.out.println("...");  }  }  }  Outer outer = new Outer();  // Требуется ссылка на внешний объект  Outer.InnerClass inner = outer.new InnerClass();  inner.doSomething();  } | public static void correctTest() {  class Outer {  // Статический вложенный класс  static class InnerClass {  public void doSomething() {  System.out.println("...");  }  }  }  // Не требует ссылки на внешний объект  Outer.InnerClass inner = new Outer.InnerClass();  inner.doSomething();  } |