FB.ST\_WRITE\_TO\_STATIC\_FROM\_INSTANCE\_METHOD

Детектор определяет ошибку, в ходе которой метод экземпляра записывает в статическое поле. Это сложно исправить, если манипулируют несколькими экземплярами, и, как правило, это плохая практика.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Запись в статическое поле из метода экземпляра может привести к неконтролируемым побочным эффектам, если манипулируется несколькими экземплярами класса, поскольку все экземпляры будут изменять одно и то же статическое поле. Это нарушает принцип инкапсуляции и может вызвать трудности при многозадачности и тестировании. |
| Комментарий | Чтобы избежать подобных ошибок, следует ограничить запись в статические поля только методами, которые работают с классом, а не с его экземплярами. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  MyClass myObject = new MyClass();  myObject.modifyStaticField();  }  static class MyClass {  private static int staticField = 0;  public void modifyStaticField() {  staticField = 10;  System.out.println("Статическое поле изменено в экземпляре: " + staticField);  }  public static void modifyStaticFieldStatically() {  staticField = 10;  System.out.println("Статическое поле изменено через статический метод: " + staticField);  }  } | public static void correctTest() {  MyClass.modifyStaticFieldStatically();  }  static class MyClass {  private static int staticField = 0;  public void modifyStaticField() {  staticField = 10;  System.out.println("Статическое поле изменено в экземпляре: " + staticField);  }  public static void modifyStaticFieldStatically() {  staticField = 10;  System.out.println("Статическое поле изменено через статический метод: " + staticField);  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Переместите логику записи в статическое поле в статический метод класса.** Создайте статический метод, который будет отвечать за изменение статического поля.
2. **Вызывайте статический метод для изменения статического поля.** Вместо вызова метода экземпляра для этой цели, обращайтесь к статическому методу непосредственно через имя класса (например, MyClass.modifyStaticFieldStatically();).
3. **Если логика изменения статического поля зависит от состояния конкретного экземпляра, пересмотрите архитектуру.** Возможно, статическое поле не является подходящим решением, и следует использовать поле экземпляра или другой механизм для хранения и управления состоянием.
4. **Рассмотрите возможность использования шаблона Singleton, если статическое поле представляет собой единственный экземпляр класса.** В этом случае доступ к состоянию можно контролировать через методы экземпляра Singleton.
5. **В многопоточной среде обеспечьте надлежащую синхронизацию при доступе к статическим полям, даже если запись осуществляется из статических методов.** Используйте блоки synchronized или другие механизмы синхронизации для предотвращения состояния гонки.