FB.UR\_UNINIT\_READ

Детектор определяет ошибку, в ходе которой конструктор считывает поле, которому еще не присвоено значение. Это часто происходит, когда программист по ошибке использует поле вместо одного из параметров конструктора.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Чтение непроинициализированного поля в конструкторе может привести к неопределённому поведению, включая использование значений по умолчанию (например, null, 0 или false), что может вызвать неожиданные ошибки в логике программы. В случае работы с объектами это может привести к NullPointerException при последующем доступе к полю. |
| Комментарий | Поле может быть неинициализировано при чтении, что часто является результатом ошибки, связанной с использованием его вместо параметра конструктора. Следует убедиться, что поле корректно инициализировано перед его использованием. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class Test {  private int value;  public Test(int value) {  // this.value еще не инициализирован  System.out.println(this.value);  this.value = value;  }  }  } | public static void correctTest() {  class Test {  private int value;  public Test(int value) {  // параметр конструктора  System.out.println(value);  this.value = value;  }  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Переместите инициализацию поля выше места его первого чтения в конструкторе. Убедитесь, что полю присвоено значение до того, как оно будет использовано.
2. Проверьте, не была ли допущена ошибка в использовании имени поля вместо параметра конструктора. Убедитесь, что вы используете параметр, переданный в конструктор, если это было задумано.
3. Убедитесь, что все поля, которые используются в конструкторе, имеют явное присваивание значения. Даже если поле инициализируется значением по умолчанию, явное присваивание может сделать код понятнее.
4. Для final полей проследите, что они инициализируются ровно один раз: Либо при объявлении, либо в каждом конструкторе класса до их использования.