FB.CAA\_COVARIANT\_ARRAY\_FIELD

Полю присваивается массив ковариантного типа. Это может привести к ArrayStoreException во время выполнения, если ссылка какого-либо другого типа будет сохранена в этом массиве позже. Стоит рассмотреть возможность изменения типа создаваемого массива или типа поля.

Таблица №1 – Основные характеристики детектора FB.CAA\_COVARIANT\_ARRAY\_FIELD

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Runtime Exception Potential |
| CWE | --- |
| CVE | --- |
| Возможные последствия | * ArrayStoreException при выполнении * Нарушение типобезопасности * Повреждение данных массива |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| private static Object[] fieldArray = new String[5];  public static void incorrectTest() {  try {  fieldArray[0] = 123; // Попытка сохранить Integer в String[]  System.out.println("Element stored: " + fieldArray[0]);  } catch (ArrayStoreException e) {  System.out.println("Caught ArrayStoreException as expected");  }  } | private static String[] safeFieldArray = new String[5];  public static void correctTest() {  safeFieldArray[0] = "safe text"; // Безопасное хранение  System.out.println("Element stored safely: " + safeFieldArray[0]);    // Альтернатива с проверкой типа  Object element = "another safe text";  if (element instanceof String) {  fieldArray[0] = element;  System.out.println("Field array element stored with type check: " + fieldArray[0]);  }  } |