FB.CAA\_COVARIANT\_ARRAY\_RETURN

Из метода возвращается массив ковариантного типа. Это может привести к ArrayStoreException во время выполнения, если вызывающий код попытается сохранить ссылку какого-либо другого типа в возвращаемом массиве. Стоит рассмотреть возможность изменения типа создаваемого массива или типа возвращаемого значения метода.

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

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

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| private static Object[] getStringArray() {  return new String[5];  }  public static void incorrectTest() {  Object[] array = getStringArray();  try {  array[0] = 123; // Попытка сохранить Integer в String[]  System.out.println("Element stored: " + array[0]);  } catch (ArrayStoreException e) {  System.out.println("Caught ArrayStoreException as expected");  }  } | private static String[] getSafeStringArray() {  return new String[5];  }  public static void correctTest() {  String[] array = getSafeStringArray();  array[0] = "safe value";  System.out.println("Element stored safely: " + array[0]);    // Альтернатива с проверкой типа  Object[] objArray = getStringArray();  Object element = "checked value";  if (element instanceof String) {  objArray[0] = element;  System.out.println("Array element stored with check: " + objArray[0]);  }  } |