FB.CAA\_COVARIANT\_ARRAY\_ELEMENT\_STORE

Значение сохраняется в массиве, а тип значения не соответствует типу массива. Из анализа известно, что фактический тип массива уже объявленного типа его переменной или поля, а также, что это присвоение не удовлетворяет исходному типу массива. Это назначение может вызвать исключение ArrayStoreException во время выполнения.

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

|  |  |
| --- | --- |
| 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 |
| public static void incorrectTest() {  // Некорректно: ковариантное хранение в массиве  Object[] objectArray = new String[5];  try {  objectArray[0] = 123; // Попытка сохранить Integer в String[]  System.out.println("Element stored: " + objectArray[0]);  } catch (ArrayStoreException e) {  System.out.println("Caught ArrayStoreException as expected");  }  } | public static void correctTest() {  // Корректно: использование правильного типа массива  String[] stringArray = new String[5];  stringArray[0] = "text"; // Безопасное хранение    // Или проверка типа перед сохранением  Object[] objectArray = new String[5];  Object element = "safe text";  if (element instanceof String) {  objectArray[0] = element;  System.out.println("Element stored safely: " + objectArray[0]);  } else {  System.out.println("Invalid type for array");  }  } |