FB.BC\_BAD\_CAST\_TO\_ABSTRACT\_COLLECTION

Детектор определяет ошибку, в ходе которой код преобразует коллекцию в абстрактную коллекцию (например, List, Set или Map). Убедитесь, что вам гарантировано, что объект имеет тип, к которому вы выполняете приведение. Если все, что вам нужно, — это иметь возможность перебирать коллекцию, вам не нужно приводить ее к набору или списку.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Приведение коллекции к абстрактному типу, такому как List, Set или Map, может привести к ClassCastException, если объект не является экземпляром соответствующего класса. Это также может снизить производительность, так как не всегда требуется приводить коллекцию к абстрактному типу, если достаточно просто перебирать элементы. |
| Комментарий | Если целью является просто перебор элементов коллекции, лучше использовать общий интерфейс коллекции, такой как Collection, или работать с самими элементами через итератор. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class MyClass {  public void processCollection(Collection<?> collection) {  // Неправильное приведение коллекции к абстрактному типу  // Это может привести к ClassCastException, если collection не является List  List<?> list = (List<?>) collection;  for (Object item : list) {  System.out.println(item);  }  }  }  Collection<String> collection = new HashSet<>();  collection.add("apple");  collection.add("banana");  MyClass myClass = new MyClass();  myClass.processCollection(collection);  } | public static void correctTest() {  class MyClass {  public void processCollection(Collection<?> collection) {  // Просто перебирать коллекцию без приведения ее к абстрактному типу  // Это более безопасно, так как не зависит от типа коллекции  for (Object item : collection) {  System.out.println(item);  }  }  }  Collection<String> collection = new HashSet<>();  collection.add("apple");  collection.add("banana");  MyClass myClass = new MyClass();  myClass.processCollection(collection);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Избегайте ненужного приведения:** Если вам нужно только перебрать элементы коллекции, работайте непосредственно с интерфейсом Collection или используйте цикл for-each.
2. **Проверяйте тип перед приведением:** Если приведение к конкретному абстрактному типу необходимо, убедитесь, что объект является экземпляром этого типа с помощью оператора instanceof перед выполнением приведения.
3. **Пересмотрите логику:** Возможно, необходимость приведения указывает на проблему в дизайне кода. Подумайте, нельзя ли реализовать требуемую функциональность, не завися от конкретного абстрактного типа коллекции.