FB.RCN\_REDUNDANT\_COMPARISON\_TWO\_NULL\_VALUES

Детектор определяет ошибку, в ходе которой метод содержит избыточное сравнение двух ссылок, обе из которых определенно равны нулю.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE476 |
| CVE | — |
| Возможные последствия | Сравнение двух null-значений бессмысленно и может указывать на ошибку в логике программы. |
| Комментарий | Следует удалить избыточное сравнение или пересмотреть логику кода, если сравнение было добавлено по ошибке. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(Object a, Object b) {  // Избыточное сравнение  // (если метод всегда возвращает null, проверка бессмысленна)  if (a == b) {  System.out.println("Обе переменные равны null");  }  } | public static void correctTest(Object a, Object b) {  if (a == null) {  System.out.println("Переменная a равна null");  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Удалите избыточное сравнение двух переменных, которые гарантированно равны null.**
2. **Пересмотрите логику кода, в котором выполняется такое сравнение.** Подумайте, была ли цель сравнения другой, или же код был добавлен по ошибке.
3. **Если предполагалось проверить, является ли какая-либо из переменных null, выполните проверку для каждой переменной отдельно.**
4. **Убедитесь, что поток выполнения программы действительно приводит обе сравниваемые переменные к null в данном контексте.**