FB.RCN\_REDUNDANT\_NULLCHECK\_OF\_NULL\_VALUE

Детектор определяет ошибку, в ходе которой метод содержит избыточную проверку известного нулевого значения на соответствие постоянному нулю.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE476 |
| CVE | — |
| Возможные последствия | Избыточная проверка null == null всегда возвращает true, что делает код излишним и может указывать на логическую ошибку. |
| Комментарий | Следует удалить лишнюю проверку или проверить, не допущена ли логическая ошибка при определении переменной. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(Object text) {  if (text == null) {  System.out.println("text равен null");  }  if (text == null) {  System.out.println("Эта проверка бессмысленна");  }  } | public static void correctTest(Object text) {  if (text == null) {  System.out.println("text равен null");  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Проанализируйте поток выполнения программы. Определите, в какой точке кода переменная гарантированно равна null.
2. Удалите все последующие проверки на null для этой переменной, если они находятся в области, где переменная остается null.
3. Убедитесь, что нет путей выполнения, при которых переменная могла бы стать ненулевой между моментом, когда она была признана null, и избыточной проверкой.
4. Пересмотрите логику кода, если избыточная проверка кажется необходимой. Возможно, существует ошибка в предыдущей логике, или же избыточная проверка была добавлена по недоразумению.