FB.RPC\_REPEATED\_CONDITIONAL\_TEST

Детектор определяет ошибку, в ходе которой код содержит условную проверку, выполняемую дважды, одну за другой (например, x == 0 || x == 0). Возможно, второе вхождение должно быть чем-то другим (например, x == 0 || y == 0).

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Дублирование условных проверок может указывать на логическую ошибку в коде, которая приводит к неправильному поведению программы. В лучшем случае это просто лишняя проверка, но в худшем — ошибка в логике, где разработчик изначально планировал сравнивать разные переменные. |
| Комментарий | Необходимо перепроверить логику условия и исправить возможную ошибку. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(int x, int y) {  if (x == 0 || x == 0) {  // ...  }  } | public static void correctTest(int x, int y) {  if (x == 0 || y == 0) {  // ...  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Тщательно проанализируйте условное выражение, содержащее повторенную проверку.
2. Определите, была ли вторая проверка добавлена по ошибке или является результатом копирования и вставки кода, который не был должным образом отредактирован.
3. Исправьте второе вхождение условия, заменив его на предполагаемую логическую проверку. Это может включать в себя использование другой переменной, другого значения или другого метода.
4. Если повторение условия было намеренным (что крайне маловероятно), пересмотрите логику, чтобы убедиться, что это действительно необходимо и не может быть упрощено.
5. Обратите внимание на любые побочные эффекты в повторяющихся проверках (например, вызовы методов). Даже если логически условие остается прежним, повторное выполнение побочных эффектов может быть нежелательным.