FB.NS\_DANGEROUS\_NON\_SHORT\_CIRCUIT

Детектор определяет ошибку, в ходе которой в коде используется некороткая логика (например, & или |), а не упрощенная логика (&& или ||). Кроме того, кажется возможным, что в зависимости от значения левой части вы можете не захотеть оценивать правую часть (поскольку она будет иметь побочные эффекты, может вызвать исключение или может быть дорогостоящей.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Использование некороткой логики (например, & или | вместо && или ||) может привести к выполнению ненужных вычислений. |
| Комментарий | Рекомендуется использовать короткую логику (&& для "и" и || для "или"), чтобы гарантировать, что правая часть выражения не будет оцениваться, если левая часть уже определяет результат. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(int x) {  // Не короткая логика с побочным эффектом  if (x > 0 & expensiveOperation()) {  System.out.println("Condition met!");  }  }  public static boolean expensiveOperation() {  // Например, это может быть запрос к базе данных или сложная вычислительная операция  return true;  } | public static void correctTest(int x) {  // Короткая логика: правая часть не оценивается, если левая уже ложна  if (x > 0 && expensiveOperation()) {  System.out.println("Condition met!");  }  }  public static boolean expensiveOperation() {  // Например, это может быть запрос к базе данных или сложная вычислительная операция  return true;  } |