FB.NS\_NON\_SHORT\_CIRCUIT

Похоже, что в этом коде используется логика, не связанная с коротким замыканием (например, & или |), а не логика короткого замыкания (&& или ||). Логика, не связанная с коротким замыканием, приводит к вычислению обеих частей выражения, даже если результат может быть получен на основе знания левой части. Это может быть менее эффективным и приводить к ошибкам, если левая часть защищает случаи, когда вычисление правой части может привести к ошибке.

Таблица №1 – Основные характеристики детектора FB.NS\_NON\_SHORT\_CIRCUIT

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Non-Short-Circuit Operator Usage |
| CWE | --- |
| CVE | --- |
| Возможные последствия | Лишние вычисления; потенциальные исключения (NullPointerException); неэффективная работа кода |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  String str = null;  if (str != null & str.length() > 0) { // ОШИБКА: & вместо &&  System.out.println("String is not empty");  } else {  System.out.println("This will throw NullPointerException!");  }  } | public static void correctTest() {  String str = null;  if (str != null && str.length() > 0) { // Корректно: &&  System.out.println("String is not empty");  } else {  System.out.println("Safe check with short-circuit");  }  } |