**FB.NP\_NULL\_ON\_SOME\_PATH\_EXCEPTION**

Здесь разыменовывается ссылочное значение, равное нулю на некотором пути управления исключениями. Это может привести к исключению NullPointerException при выполнении кода. Следует обратить внимание: поскольку SpotBugs в настоящее время не отсекает невозможные пути исключений, это может быть ложным предупреждением.

Также стоит обратить внимание, что SpotBugs считает вариант оператора переключателя по умолчанию путем исключения, поскольку случай по умолчанию часто невозможен.

Таблица №1 – Основные характеристики детектора **FB.NP\_NULL\_ON\_SOME\_PATH\_EXCEPTION**

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Null pointer dereference |
| CWE | CWE-476 |
| CVE | --- |
| Возможные последствия | Возникновение NullPointerException при обработке исключений, что может маскировать исходную ошибку |
| Комментарий (опционально) | Особое внимание следует уделить блокам catch/finally. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectExample(String input) {  String data = null;  try {  data = parseInput(input);  System.out.println("Данные: " + data.length());  } catch (Exception e) {  // Потенциальный NPE, если исключение возникло до инициализации data  System.out.println("Ошибка обработки. Длина: " + data.length());  }  } | public static void correctExample(String input) {  String data = null;  try {  data = parseInput(input);  System.out.println("Данные: " + data.length());  } catch (Exception e) {  // Безопасная обработка  System.out.println("Ошибка обработки" +  (data != null ? ". Длина: " + data.length() : ""));  }  } |

Методы устранения уязвимости при срабатывании детектора

1. Защитное программирование: Всегда проверяйте критические переменные на null перед использованием в блоках catch/finally
2. Разделение логики: Избегайте сложных операций в обработчиках исключений
3. Логирование без разыменования: В блоках catch сначала сохраняйте нужные значения в локальные переменные
4. Явная инициализация: Инициализируйте переменные, которые могут использоваться в блоках обработки исключений