FB.NP\_GUARANTEED\_DEREF\_ON\_EXCEPTION\_PATH

На пути исключения существует оператор или ветвь, выполнение которой гарантирует, что значение в этот момент равно нулю, и это значение гарантированно будет разыменовано (за исключением прямых путей, включающих исключения во время выполнения).

Таблица №1 – Основные характеристики детектора [FB.NP\_GUARANTEED\_DEREF\_ON\_EXCEPTION\_PATH]

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Null pointer dereference on exception path |
| CWE | CWE-476 |
| CVE | — |
| Возможные последствия | Сбой приложения, утечки данных, неожиданное завершение работы. |
| Комментарий (опционально) | Обязательна проверка на null перед использованием объекта, особенно в блоках кода, где могут возникать исключения. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void correctExample() {  String data = null;  try {  data = getData();  } catch (Exception e) {  e.printStackTrace();  }  if (data != null) {  System.out.println(data.length());  } else {  System.out.println("Data is null due to exception!");  }  } | public static void incorrectExample() {  String data = null;  try {  data = getData(); // Может выбросить исключение  } catch (Exception e) {  e.printStackTrace();  }  System.out.println(data.length()); // NP\_GUARANTEED\_DEREF\_ON\_EXCEPTION\_PATH сработает здесь  } |

Методы устранения уязвимости при срабатывании детектора

1. Всегда проверяйте переменные на null перед их использованием.
2. Используйте аннотации @Nullable и @Nonnull для явного указания допустимости null.
3. Применяйте методы Optional для безопасной работы с потенциально нулевыми объектами.
4. Используйте статический анализ кода для автоматического обнаружения подобных ошибок.