FB.REC\_CATCH\_EXCEPTION

Детектор определяет ошибку, в ходе которой метод использует блок try-catch, который перехватывает объекты Exception, но Exception не генерируется внутри блока try, а RuntimeException не перехватывается явным образом. Распространенной ошибкой является выражение try { ... } catch (Exception e) { Something } как сокращение для перехвата нескольких типов исключений, каждый из блоков catch которых идентичен, но эта конструкция также случайно перехватывает RuntimeException. , маскируя потенциальные ошибки.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE396 |
| CVE | — |
| Возможные последствия | Перехват Exception без необходимости может скрыть ошибки, включая RuntimeException, что затруднит отладку и выявление критических проблем. |
| Комментарий | Следует перехватывать только конкретные исключения, которые действительно могут возникнуть в блоке try, и отдельно обрабатывать RuntimeException, если это необходимо. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  try {  System.out.println("код не выбрасывает Exception");  } catch (Exception e) {  System.out.println("Перехват Exception, но внутри try нет его источника");  }  } | public static void correctTest() {  try {  throw new IllegalArgumentException("Пример RuntimeException");  } catch (IllegalArgumentException e) {  System.out.println("Перехвачена IllegalArgumentException: " + e.getMessage());  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Проанализируйте блок try и определите, какие проверяемые исключения (Checked Exception) могут быть в нем выброшены. Перехватывайте только эти конкретные типы исключений.
2. Если внутри блока try может быть выброшен RuntimeException, рассмотрите возможность явного перехвата наиболее вероятных подклассов (NullPointerException, IllegalArgumentException, ArrayIndexOutOfBoundsException и т.д.) в отдельных блоках catch для их специфической обработки.
3. Избегайте перехвата Exception как общего типа, если в блоке try нет кода, который явно выбрасывает проверяемые исключения.
4. Рассмотрите возможность предотвращения RuntimeException с помощью проверок состояния (например, проверка на null, проверка границ массива) вместо их перехвата.
5. Если необходимо перехватить все оставшиеся исключения, включая RuntimeException, сделайте это явно после перехвата более специфичных исключений.
6. Убедитесь, что перехваченные исключения обрабатываются должным образом (логирование, восстановление состояния, повторное выбрасывание с обертыванием и т.д.). Пустые блоки catch следует избегать.