UNREACHABLE\_CODE.GLOBAL

Детектор указывает на то, что существует некоторый код, который никогда не будет выполнен из-за ошибки с глобальной переменной.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Average |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Недостижимый код из-за глобальных переменных может привести к сбоям, неопределенному поведению программы и затруднить отладку. Это также может нарушить логику программы, особенно в многозадачных или многопоточечных приложениях. |
| Комментарий | Рекомендуется избегать чрезмерного использования глобальных переменных и использовать локальные переменные или другие структуры данных, чтобы повысить предсказуемость и управляемость кода. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| // Глобальная переменная  private static boolean flag = true;  public static void incorrectTest() {  if (flag) {  System.out.println("Start test");  return;  }  System.out.println("This will never be printed");  } | // Глобальная переменная  private static boolean flag = true;  public static void correctTest(boolean change) {  flag = change;  if (flag) {  System.out.println("Condition met, but flag is false now.");  } else {  System.out.println("This line is reachable because flag is false.");  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Проанализируйте использование глобальной переменной. Определите все места в коде, где изменяется значение глобальной переменной, влияющей на недостижимый код.
2. Пересмотрите условия, зависящие от глобальной переменной. Изучите условные операторы или циклы, которые используют эту глобальную переменную. Убедитесь, что логика этих условий корректна и позволяет достигать ранее недостижимый код при определенных значениях глобальной переменной.