FB.IL\_INFINITE\_LOOP

Детектор определяет ошибку, в ходе которой цикл не имеет способа завершиться (кроме, возможно, выдачи исключения).

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Бесконечное выполнение программы, зависание потока, отказ в обслуживании (DoS) |
| Комментарий | Ошибка возникает, когда цикл не имеет корректного условия выхода, что приводит к бесконечному выполнению кода и возможному зависанию приложения |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  int i = 0;  while (i < 10) {  // ...  // Без условия завершения или  // с неверным условием  i--;  }  } | public static void correctTest() {  int i = 0;  while (i < 10) {  // С условием  i++;  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Проверьте условие выхода цикла:** Убедитесь, что условие, при котором цикл должен завершиться, является логически правильным и рано или поздно станет истинным (или ложным).
2. **Убедитесь в обновлении переменных условия:** Проверьте, что все переменные, участвующие в условии выхода цикла, правильно обновляются внутри тела цикла таким образом, чтобы условие выхода могло быть достигнуто.
3. **Рассмотрите использование break:** Если выход из цикла зависит от выполнения определенного условия внутри тела цикла, используйте оператор break для принудительного выхода. Убедитесь, что условие для break может быть достигнуто.
4. **Избегайте циклов без явного условия выхода:** Циклы while (true) или for (;;), которые не содержат оператора break или return, с большой вероятностью являются ошибкой (за исключением случаев, когда это намеренное поведение, например, в обработчиках событий или серверных приложениях, где цикл ожидает новых запросов). Если такой цикл необходим, убедитесь, что внутри него есть логика, которая в конечном итоге приведет к выходу из метода или завершению программы.
5. **Тщательно протестируйте циклы:** Протестируйте различные сценарии выполнения цикла, включая граничные случаи, чтобы убедиться, что цикл всегда завершается при ожидаемых условиях.