DEREF\_AFTER\_NULL.LOOP

Детектор указывает на ошибку, что связанную с попыткой разыменования (то есть обращения к значению, на которое указывает ссылка) нулевой ссылки внутри цикла.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | Normal |
| Надежность | Average |
| Источник | SvEng |
| Тип ошибки | Ошибка разработчика |
| CWE | CWE476 |
| CVE | — |
| Возможные последствия | Приводит к NullPointerException или аналогичной ошибке во время выполнения, особенно при большом числе итераций, затрудняя отладку. |
| Комментарий | Следует убедиться, что ссылка, используемая в теле цикла, проверена на null перед каждым обращением или надёжно инициализирована. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest(String[] values) {  for (String value : values) {  System.out.println(value.length());  }  } | public static void correctTest(String[] values) {  for (String value : values) {  if (value != null) {  System.out.println(value.length());  } else {  System.out.println("Null value skipped");  }  }  } |

Методы устранения уязвимости при срабатывании детектора:

1. Добавьте явную проверку на null перед каждым обращением к потенциально нулевой ссылке внутри цикла. Используйте оператор if (reference != null) для безопасного доступа.
2. Если логика позволяет, убедитесь, что ссылка инициализируется ненулевым значением перед входом в цикл. Это может включать проверку входных данных или инициализацию внутри цикла.