FB.IIL\_ELEMENTS\_GET\_LENGTH\_IN\_LOOP

Метод вызывает NodeList.getLength() внутри цикла, и NodeList создается вызовом getElementsByTagName. Этот NodeList не хранит свою длину, а вычисляет ее каждый раз не самым оптимальным образом. Стоит рассмотреть возможность сохранения длины переменной перед циклом.

Таблица №1 – Основные характеристики детектора FB.IIL\_ELEMENTS\_GET\_LENGTH\_IN\_LOOP

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Performance Issue |
| CWE | --- |
| CVE | --- |
| Возможные последствия | * Снижение производительности * Избыточные вычисления * Увеличение времени выполнения |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  String[] items = {"A", "B", "C", "D", "E"};    // Некорректно: получение длины в каждой итерации  for (int i = 0; i < items.length; i++) {  System.out.println(items[i] + " - " + items.length); // items.length вызывается многократно  }  } | public static void correctTest() {  String[] items = {"A", "B", "C", "D", "E"};    // Корректно: кэширование длины перед циклом  int length = items.length;  for (int i = 0; i < length; i++) {  System.out.println(items[i] + " - " + length);  }    // Альтернатива с for-each  for (String item : items) {  System.out.println(item);  }  } |