FB.IIL\_PATTERN\_COMPILE\_IN\_LOOP

Метод вызывает Pattern.compile внутри цикла, передавая константные аргументы. Если шаблон необходимо использовать несколько раз, нет смысла компилировать его для каждой итерации цикла. Следует переместить этот вызов за пределы цикла или даже в статическое конечное поле.

Таблица №1 – Основные характеристики детектора FB.IIL\_PATTERN\_COMPILE\_IN\_LOOP

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Performance Issue |
| CWE | --- |
| CVE | --- |
| Возможные последствия | * Значительное снижение производительности * Избыточное создание объектов Pattern * Увеличение времени выполнения |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  String[] inputs = {"item1", "item2", "item3"};    // Некорректно: компиляция в каждой итерации  for (String input : inputs) {  Pattern pattern = Pattern.compile("item\\d"); // Компиляция в цикле  Matcher matcher = pattern.matcher(input);  if (matcher.matches()) {  System.out.println("Found: " + input);  }  }  } | public static void correctTest() {  String[] inputs = {"item1", "item2", "item3"};    // Корректно: однократная компиляция перед циклом  Pattern pattern = Pattern.compile("item\\d");  for (String input : inputs) {  Matcher matcher = pattern.matcher(input);  if (matcher.matches()) {  System.out.println("Found: " + input);  }  }  } |