FB.DM\_STRING\_VOID\_CTOR

Детектор определяет ошибку, в ходе которой создание нового объекта java.lang.String с использованием конструктора без аргументов тратит впустую память, поскольку созданный таким образом объект будет функционально неотличим от пустой строковой константы "". Java гарантирует, что идентичные строковые константы будут представлены одним и тем же объектом String. Поэтому вам следует просто использовать константу пустой строки напрямую.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Создание нового объекта String с помощью конструктора без аргументов (new String()) приводит к ненужному выделению памяти и снижению производительности, так как этот объект эквивалентен строковому литералу "", но не использует пул строк. |
| Комментарий | Рекомендуется использовать строковый литерал "" вместо new String(), поскольку Java автоматически кэширует строковые литералы, что позволяет избежать лишних аллокаций и повысить эффективность работы программы. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  String emptyString = new String();  System.out.println(emptyString);  } | public static void correctTest() {  String emptyString = "";  System.out.println(emptyString);  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Замените new String() на "":** Везде, где вы видите создание новой пустой строки с помощью конструктора без аргументов, замените это на использование пустой строковой константы "".
2. **Проверьте весь код:** Найдите все места, где используется new String() и выполните замену.
3. **Понимайте пул строк:** Строковые литералы в Java автоматически помещаются в пул строк. Использование "" гарантирует, что вы работаете с объектом из этого пула, что способствует повторному использованию объектов и экономии памяти.
4. **Избегайте ненужного создания объектов:** В целом, старайтесь избегать создания новых объектов, если уже существует эквивалентный объект, который можно использовать повторно, особенно для часто используемых неизменяемых объектов, таких как строки.