FB.DM\_NEW\_FOR\_GETCLASS

Детектор определяет ошибку, в ходе которой этот метод выделяет объект только для вызова метода getClass(), чтобы получить для него объект класса. Проще получить доступ к свойству .class класса.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Сбой программы при попытке модификации неизменяемого кортежа, некорректная работа функций, ожидающих изменяемую структуру данных |
| Комментарий | Невозможность добавить элемент в кортеж вызывает сбой там, где ожидался список |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  // Неоправданное создание объекта  Class<?> clazz = new DM\_NEW\_FOR\_GETCLASS().getClass();  System.out.println("Class name: " + clazz.getName());  } | public static void correctTest() {  Class<DM\_NEW\_FOR\_GETCLASS> clazz = DM\_NEW\_FOR\_GETCLASS.class;  System.out.println("Class name: " + clazz.getName());  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Используйте литерал класса:** Замените создание нового объекта и вызов getClass() на использование литерала класса (ClassName.class). Например, вместо new MyClass().getClass() используйте MyClass.class.
2. **Получите Class от существующего экземпляра (если есть):** Если у вас уже есть экземпляр нужного класса, вы можете вызвать его метод getClass(). Однако, если объекта еще нет и вы создаете его только для получения Class, используйте литерал класса.
3. **Проверьте весь код:** Найдите все места, где создаются новые объекты исключительно для вызова getClass() и замените их на литералы класса.
4. **Улучшите читаемость:** Использование литерала класса делает код более прямым и понятным для разработчиков, показывая, что вы получаете объект Class для определенного типа.