FB.DM\_GC

Детектор определяет ошибку, в ходе которой код явно вызывает сбор мусора. За исключением конкретного использования в бенчмаркинге, это очень сомнительно.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Явный вызов сборщика мусора (System.gc(), Runtime.getRuntime().gc()) не гарантирует немедленного освобождения памяти и может привести к неожиданным задержкам в работе приложения. |
| Комментарий | Рекомендуется избегать явных вызовов сборки мусора, так как JVM самостоятельно управляет памятью наиболее эффективным способом. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  // ...  // явное принудительное освобождение памяти  System.gc();  } | public static void correctTest() {  // Единственный оправданный случай —  // бенчмаркинг перед замером потребления памяти.  System.gc(); // Освобождаем память перед замером  long beforeMemory = Runtime.getRuntime().freeMemory();  byte[] array = new byte[10\_000\_000];  long afterMemory = Runtime.getRuntime().freeMemory();  System.out.println("Memory used: " + (beforeMemory - afterMemory) + " bytes");  } |