FB.DMI\_THREAD\_PASSED\_WHERE\_RUNNABLE\_EXPECTED

Детектор определяет ошибку, в ходе которой объект Thread передается в качестве параметра методу, в котором ожидается Runnable. Это довольно необычно и может указывать на логическую ошибку или вызывать неожиданное поведение.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Передача Thread вместо Runnable может привести к запуску потока дважды или к неожиданному поведению при управлении многопоточностью. |
| Комментарий | Лучше передавать Runnable, так как Thread сам реализует Runnable, но его повторный запуск после завершения приведет к исключению IllegalThreadStateException. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  Thread thread = new Thread(() -> System.out.println("Thread running"));  // Передаем Thread вместо Runnable  Thread newThread = new Thread(thread);  newThread.start();  } | public static void correctTest() {  Runnable runnable = () -> System.out.println("Thread running correctly");  // Передаем корректный Runnable  Thread newThread = new Thread(runnable);  newThread.start();  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Передавайте Runnable:** Вместо передачи объекта Thread, передайте объект, который реализует интерфейс Runnable. Этот объект будет содержать код, который необходимо выполнить в новом потоке.
2. **Создавайте новые Thread объекты с Runnable:** Используйте конструктор Thread(Runnable target) для создания новых потоков, которые будут выполнять код, определенный в переданном Runnable объекте.
3. **Используйте ExecutorService:** Для более сложного управления пулом потоков и выполнения задач используйте java.util.concurrent.ExecutorService. Передавайте объекты Runnable или Callable в методы submit() или execute() ExecutorService.
4. **Разделяйте задачу и механизм выполнения:** Четко разделяйте определение задачи (через Runnable или Callable) и механизм ее выполнения (через Thread или ExecutorService).
5. **Проверьте места передачи аргументов:** Внимательно проверьте все вызовы методов, которые принимают Runnable в качестве аргумента, и убедитесь, что вы передаете именно Runnable, а не Thread.