FB.UMAC\_UNCALLABLE\_METHOD\_OF\_ANONYMOUS\_CLASS

Детектор определяет ошибку, в ходе которой анонимный класс определяет метод, который не вызывается напрямую и не переопределяет метод в суперклассе. Поскольку методы других классов не могут напрямую вызывать методы, объявленные в анонимном классе, кажется, что этот метод не вызывается. Метод может быть просто мертвым кодом, но также возможно, что метод предназначен для переопределения метода, объявленного в суперклассе, и из-за опечатки или другой ошибки метод фактически не переопределяет метод, для которого он предназначен.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Метод, определённый в анонимном классе, не будет вызван, что может привести к неработающему функционалу или наличию мёртвого кода. В худшем случае, если метод должен был переопределять метод суперкласса, но из-за ошибки этого не делает, программа может работать некорректно или не обрабатывать определённые сценарии. |
| Комментарий | Рекомендуется проверить необходимость метода: если метод не используется, его следует удалить, а если он должен переопределять метод суперкласса, убедиться в корректности его объявления. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  Runnable runnable = new Runnable() {  @Override  public void run() {  // ...  }  void execute() {  // нигде не вызывается и  // не переопределяет метод суперкласса  }  };  } | public static void correctTest() {  Runnable runnable = new Runnable() {  @Override  public void run() {  execute();  // ...  }  void execute() {  // ...  }  };  } |