FB.MS\_MUTABLE\_COLLECTION

Экземпляр изменяемой коллекции присваивается конечному статическому полю, поэтому он может быть изменен вредоносным кодом или случайно из другого. Стоит рассмотреть возможность переноса этого поля в Collections.unmodifyingSet/List/Map/etc. чтобы избежать этой уязвимости.

Таблица №1 – Основные характеристики детектора FB.MS\_MUTABLE\_COLLECTION

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Security Vulnerability |
| CWE | CWE218 |
| CVE | --- |
| Возможные последствия | * Несанкционированная модификация коллекции * Нарушение инкапсуляции * Потенциальные race conditions |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  // Некорректно: возврат изменяемой коллекции  class Team {  private List<String> members = new ArrayList<>();    public List<String> getMembers() {  return members; // Опасное возвращение ссылки  }  }    Team team = new Team();  team.getMembers().add("Alice");  team.getMembers().add("Bob");    // Внешний код может модифицировать внутреннюю коллекцию  team.getMembers().clear();  System.out.println("Team members (insecure): " + team.getMembers());  } | public static void correctTest() {  // Корректно: защитные копии и неизменяемые коллекции  class Team {  private List<String> members = new ArrayList<>();    public List<String> getMembers() {  return new ArrayList<>(members); // Возвращаем копию  }    // Или лучше - возвращать неизменяемое представление  public List<String> getMembersUnmodifiable() {  return Collections.unmodifiableList(members);  }    // Метод для добавления членов с контролем  public void addMember(String name) {  members.add(name);  }  }    Team team = new Team();  team.addMember("Alice");  team.addMember("Bob");    try {  team.getMembersUnmodifiable().clear(); // Выбросит исключение  } catch (UnsupportedOperationException e) {  System.out.println("Properly protected collection");  }  } |