FB.ME\_MUTABLE\_ENUM\_FIELD

Изменяемое общедоступное поле определяется внутри общедоступного перечисления, поэтому оно может быть изменено вредоносным кодом или случайно из другого пакета. Хотя изменяемые поля перечислений могут использоваться для ленивой инициализации, открывать их для внешнего мира — плохая практика. Стоит рассмотреть возможность объявления этого поля окончательным и/или частным для пакета.

Таблица №1 – Основные характеристики детектора FB.ME\_MUTABLE\_ENUM\_FIELD

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Design Flaw |
| CWE | --- |
| CVE | --- |
| Возможные последствия | * Нарушение инвариантов enum * Непредсказуемое поведение программы * Проблемы с безопасностью |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  // Некорректно: enum с изменяемым полем-коллекцией  enum Department {  IT(new ArrayList<>()), HR(new ArrayList<>());    private List<String> employees;    Department(List<String> employees) {  this.employees = employees;  }    public List<String> getEmployees() {  return employees; // Возвращает изменяемую коллекцию  }  }    Department.IT.getEmployees().add("John"); // Модификация enum!  System.out.println("IT Employees: " + Department.IT.getEmployees());  } | public static void correctTest() {  // Корректно: защитная копия и неизменяемая коллекция  enum Department {  IT(List.of()), HR(List.of());    private final List<String> employees;    Department(List<String> employees) {  this.employees = new ArrayList<>(employees); // Защитная копия  }    public List<String> getEmployees() {  return new ArrayList<>(employees); // Возвращаем копию  }    // Или лучше - возвращать неизменяемое представление  public List<String> getEmployeesUnmodifiable() {  return List.copyOf(employees);  }  }    // Попытка модификации будет неудачной  List<String> itEmployees = Department.IT.getEmployeesUnmodifiable();  try {  itEmployees.add("Mary"); // Выбросит UnsupportedOperationException  } catch (UnsupportedOperationException e) {  System.out.println("Properly protected enum field");  }  } |