FB.CI\_CONFUSED\_INHERITANCE

Этот класс объявлен финальным, но объявляет поля, подлежащие защите. Поскольку класс является окончательным, его нельзя получить из него, а использование protected сбивает с толку. Модификатор доступа к полю должен быть изменен на частный или общедоступный, чтобы отображать истинное использование поля.

Таблица №1 – Основные характеристики детектора FB.CI\_CONFUSED\_INHERITANCE

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Design Flaw |
| CWE | --- |
| CVE | --- |
| Возможные последствия | Нарушение принципа подстановки Барбары Лисков; Неочевидное поведение подклассов; Трудности в поддержке кода |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| // Некорректная иерархия наследования  static class Bird {  void fly() {  System.out.println("Flying");  }  }  static class Penguin extends Bird { // Пингвин не может летать!  @Override  void fly() {  throw new UnsupportedOperationException("Penguins can't fly!");  }  }  public static void incorrectTest() {  Bird bird = new Penguin();  bird.fly(); // Выбросит исключение  } | // Корректная реализация через композицию  static interface Bird {  void move();  }  static class FlyingBird implements Bird {  public void move() {  System.out.println("Flying");  }  }  static class Penguin implements Bird {  public void move() {  System.out.println("Swimming");  }  }  public static void correctTest() {  Bird bird1 = new FlyingBird();  Bird bird2 = new Penguin();    bird1.move(); // Flying  bird2.move(); // Swimming  } |