FB.SE\_READ\_RESOLVE\_IS\_STATIC

Чтобы метод readResolve распознавался механизмом сериализации, его нельзя объявлять как статический метод.

Таблица №1 – Основные характеристики детектора FB.SE\_READ\_RESOLVE\_IS\_STATIC

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Incorrect readResolve() Implementation |
| CWE | --- |
| CVE | --- |
| Возможные последствия | Нарушение контракта сериализации, если readResolve() объявлен как static; неожиданное поведение при десериализации - метод не будет вызван автоматически; потенциальные проблемы с безопасностью, если подмена объекта при десериализации работает некорректно |
| Комментарий | Ошибка возникает, когда метод ошибочно помечен как static, что делает его бесполезным для механизма сериализации. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| static class IncorrectSingleton implements Serializable {  static final IncorrectSingleton INSTANCE = new IncorrectSingleton();    private static Object readResolve() { // Ошибка: метод static  return INSTANCE;  }  } | static class CorrectSingleton implements Serializable {  static final CorrectSingleton INSTANCE = new CorrectSingleton();    private Object readResolve() { // Правильно: не-static  return INSTANCE;  }  } |

Методы устранения уязвимости при срабатывании детектора.

1. Убедитесь, что readResolve() не объявлен как static: метод должен быть экземплярным;
2. Проверьте сигнатуру метода: должна быть точно: private Object readResolve();
3. Протестируйте десериализацию: убедитесь, что метод действительно вызывается и возвращает правильный объект
4. Для синглтонов используйте enum-реализацию. Это наиболее надежный способ гарантировать единственность экземпляра: