FB.HSC\_HUGE\_SHARED\_STRING\_CONSTANT

Детектор определяет ошибку, в ходе которой большая строковая константа дублируется в нескольких файлах классов. Вероятно, это связано с тем, что последнее поле инициализируется строковой константой, а язык Java требует, чтобы все ссылки на последнее поле из других классов были встроены в этот файл классов.

Таблица №1 – Основные характеристики детектора

|  |  |
| --- | --- |
| Категория критичности | MAJOR |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Ошибка разработчика |
| CWE | — |
| CVE | — |
| Возможные последствия | Дублирование большой строковой константы в нескольких файлах классов приводит к увеличению размера байт-кода, росту потребления памяти и возможным проблемам с кешированием. |
| Комментарий | Рекомендуется хранить такие строковые константы в одном общем классе и ссылаться на них из других классов. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Некорректное написание | Корректное написание |
| public static void incorrectTest() {  class A {  public static final String HUGE\_STRING = "Очень длинная строка... (2+ KB)";  public static String getString() {  return HUGE\_STRING;  }  }  class B {  public static final String HUGE\_STRING = "Очень длинная строка... (2+ KB)";  public static String getString() {  return HUGE\_STRING;  }  }  String s1 = A.getString();  String s2 = B.getString();  } | public static void correctTest() {  class Constants {  public static final String HUGE\_STRING = "Очень длинная строка... (2+ KB)";  }  class A {  // ...  public static String getString() {  return Constants.HUGE\_STRING;  }  }  class B {  // ...  public static String getString() {  return Constants.HUGE\_STRING;  }  }  String s1 = A.getString();  String s2 = B.getString();  } |

Методы устранения уязвимости при срабатывании детектора:

1. **Создайте общий класс констант:** Определите отдельный класс (например, с именем Constants, CommonConstants или аналогичным), предназначенный для хранения общих констант.
2. **Переместите большую строковую константу:** Вынесите определение public static final String HUGE\_STRING = ...; из всех классов, где она дублируется, в этот общий класс констант.
3. **Используйте ссылку на константу:** В классах, где ранее использовалась дублированная константа, замените прямое обращение на ссылку через имя общего класса: CommonConstants.HUGE\_STRING.
4. **Перекомпилируйте проект:** После внесения изменений перекомпилируйте весь проект, чтобы изменения отразились в .class файлах.
5. **Проверьте размер .class файлов (опционально):** После перекомпиляции вы можете сравнить размер .class файлов до и после рефакторинга, чтобы убедиться, что дублирование было устранено и размер файлов уменьшился.