FB.SQL\_BAD\_RESULTSET\_ACCESS

Вызов методов getXXX или updateXXX набора результатов был выполнен там, где индекс поля равен 0. Поскольку поля ResultSet начинаются с индекса 1, это всегда является ошибкой.

Таблица №1 – Основные характеристики детектора FB.SQL\_BAD\_RESULTSET\_ACCESS

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Incorrect ResultSet Access |
| CWE | --- |
| CVE | --- |
| Возможные последствия | Утечки ресурсов, если ResultSet, Statement или Connection не закрываются; SQL-инъекции, если запрос формируется конкатенацией строк; некорректное чтение данных, если колонки выбираются по индексу, а не по имени; проблемы с производительностью из-за неоптимального управления соединениями. |
| Комментарий | Рекомендуется использовать try-with-resources и параметризованные запросы. |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  try {  Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/test", "user", "pass");  Statement stmt = conn.createStatement();  ResultSet rs = stmt.executeQuery("SELECT id, name FROM users");  while (rs.next()) {  int id = rs.getInt(1); // Плохо: доступ по индексу  String name = rs.getString(2); // Плохо: доступ по индексу  System.out.println(id + ": " + name);  }  // Плохо: ресурсы не закрыты!  } catch (SQLException e) {  e.printStackTrace();  }  } | public static void correctTest() {  String query = "SELECT id, name FROM users WHERE id = ?";  try (Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/test", "user", "pass");  PreparedStatement stmt = conn.prepareStatement(query)) {  stmt.setInt(1, 1); // Защита от SQL-инъекций  try (ResultSet rs = stmt.executeQuery()) {  while (rs.next()) {  int id = rs.getInt("id"); // Хорошо: доступ по имени  String name = rs.getString("name"); // Хорошо: доступ по имени  System.out.println(id + ": " + name);  }  }  } catch (SQLException e) {  e.printStackTrace();  }  } |

Методы устранения уязвимости при срабатывании детектора.

1. Всегда закрывать ресурсы (ResultSet, Statement, Connection);
2. Использовать PreparedStatement вместо Statement;
3. Обращаться к колонкам по имени, а не по индексу;
4. Использовать пул соединений (HikariCP, Apache DBCP).