FB.MTIA\_SUSPECT\_SERVLET\_INSTANCE\_FIELD

Этот класс является наследником класса сервлета и использует переменную-член экземпляра. Поскольку в рамках J2EE создается только один экземпляр класса сервлета и используется в многопоточном режиме, эта парадигма крайне нежелательна и, скорее всего, проблематична. Стоит рассмотреть возможность использования только локальных переменных метода.

Таблица №1 – Основные характеристики детектора FB.MTIA\_SUSPECT\_SERVLET\_INSTANCE\_FIELD

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Concurrency Vulnerability |
| CWE | --- |
| CVE | --- |
| Возможные последствия | Состояние гонки в многопоточной среде; Утечка данных между пользовательскими сессиями; Непредсказуемое поведение сервлета |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| // Некорректно: изменяемое поле экземпляра сервлета  static class UserServletIncorrect extends HttpServlet {  private String currentUser; // Опасное поле экземпляра  @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp)  throws IOException {  currentUser = req.getParameter("user");  resp.getWriter().write("Current user: " + currentUser);  }  }  public static void incorrectTest(HttpServletRequest req, HttpServletResponse resp) {  if (req != null && resp != null) {  try {  new UserServletIncorrect().doGet(req, resp);  } catch (IOException e) {  e.printStackTrace();  }  }  } | // Корректно: использование локальных переменных  static class UserServletCorrect extends HttpServlet {  @Override  protected void doGet(HttpServletRequest req, HttpServletResponse resp)  throws IOException {  String currentUser = req.getParameter("user"); // Локальная переменная  resp.getWriter().write("Current user: " + currentUser);  }  }  public static void correctTest(HttpServletRequest req, HttpServletResponse resp) {  if (req != null && resp != null) {  try {  new UserServletCorrect().doGet(req, resp);  } catch (IOException e) {  e.printStackTrace();  }  }  } |