FB.XSS\_REQUEST\_PARAMETER\_TO\_SEND\_ERROR

Этот код напрямую записывает параметр HTTP на страницу ошибок сервера (используя HttpServletResponse.sendError). Повторение этих ненадежных входных данных допускает отражение уязвимости межсайтового скриптинга.

Таблица №1 – Основные характеристики детектора FB.XSS\_REQUEST\_PARAMETER\_TO\_SEND\_ERROR

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Security Vulnerability (Cross-Site Scripting) |
| CWE | CWE81 |
| CVE | --- |
| Возможные последствия | Отраженная XSS-атака через сообщения об ошибках; кража сессионных cookies; подмена содержимого страницы ошибки |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest(HttpServletRequest request, HttpServletResponse response) {  // Некорректно: прямой вывод пользовательского ввода в ошибку  if (request != null && response != null) {  try {  String errorMsg = request.getParameter("errorMsg");  response.sendError(HttpServletResponse.SC\_BAD\_REQUEST,  "Invalid input: " + errorMsg); // Уязвимость XSS!  System.out.println("Error sent (INSECURE)");  } catch (Exception e) {  e.printStackTrace();  }  }  } | public static void correctTest(HttpServletRequest request, HttpServletResponse response) {  // Корректно: экранирование и валидация сообщения  if (request != null && response != null) {  try {  String errorMsg = request.getParameter("errorMsg");  String safeMsg = sanitizeErrorMessage(errorMsg);  response.sendError(HttpServletResponse.SC\_BAD\_REQUEST,  "Invalid input: " + safeMsg);  System.out.println("Error sent securely");  } catch (Exception e) {  e.printStackTrace();  }  }  } |