FB.XSS\_REQUEST\_PARAMETER\_TO\_SERVLET\_WRITER

Этот код напрямую записывает параметр HTTP в выходные данные сервлета, что допускает отражение уязвимости межсайтового скриптинга.

Таблица №1 – Основные характеристики детектора FB.XSS\_REQUEST\_PARAMETER\_TO\_SERVLET\_WRITER

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Security Vulnerability (Cross-Site Scripting) |
| CWE | CWE79 |
| CVE | --- |
| Возможные последствия | Постоянная/отраженная XSS-атака; выполнение произвольного JavaScript кода; компрометация пользовательских данных |

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  if (request != null && response != null) {  try {  String userContent = request.getParameter("content");  PrintWriter out = response.getWriter();  out.println("<div>" + userContent + "</div>"); // Уязвимость XSS!  System.out.println("Content written (INSECURE)");  } catch (Exception e) {  e.printStackTrace();  }  }  } | public static void correctTest(HttpServletRequest request, HttpServletResponse response) {  // Корректно: экранирование HTML и CSP-заголовки  if (request != null && response != null) {  try {  String userContent = request.getParameter("content");  response.setContentType("text/html");  response.setHeader("Content-Security-Policy", "default-src 'self'");    PrintWriter out = response.getWriter();  out.println("<div>" + escapeHtml(userContent) + "</div>");  System.out.println("Content written securely");  } catch (Exception e) {  e.printStackTrace();  }  }  } |