FB.HRS\_REQUEST\_PARAMETER\_TO\_COOKIE

Этот код создает файл cookie HTTP, используя ненадежный параметр HTTP. Если этот файл cookie будет добавлен в ответ HTTP, это приведет к уязвимости разделения ответа HTTP.

Таблица №1 – Основные характеристики детектора FB.HRS\_REQUEST\_PARAMETER\_TO\_COOKIE

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Security Vulnerability |
| CWE | CWE113 |
| CVE | --- |
| Возможные последствия | Межсайтовая скриптовая атака (XSS); подделка межсайтовых запросов (CSRF); утечка конфиденциальных данных |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest(HttpServletRequest request, HttpServletResponse response) {  // Некорректно: прямое сохранение параметра запроса в cookie  if (request != null && response != null) {  String userInput = request.getParameter("userPref");  Cookie cookie = new Cookie("userPreference", userInput); // Без валидации  cookie.setMaxAge(3600);  response.addCookie(cookie);  System.out.println("Cookie set without validation (INSECURE)");  }  } | public static void correctTest(HttpServletRequest request, HttpServletResponse response) {  // Корректно: валидация и санация входных данных  if (request != null && response != null) {  String userInput = request.getParameter("userPref");  // Валидация и очистка входных данных  String sanitized = sanitizeInput(userInput);  if (isValidPreference(sanitized)) {  Cookie cookie = new Cookie("userPreference", sanitized);  cookie.setHttpOnly(true);  cookie.setSecure(true);  cookie.setMaxAge(3600);  response.addCookie(cookie);  System.out.println("Secure cookie set with validation");  } else {  System.out.println("Invalid input rejected");  }  }  } |