FB.PATH\_TRAVERSAL\_IN

Ошибка "Method reads a file whose location is specified by user input" указывает на потенциальную уязвимость в коде, связанную с чтением файла, путь к которому указывается пользовательским вводом. Это может стать источником опасности, таких как чтение конфиденциальных файлов, выполнение нежелательных операций или атаки вида "Path Traversal" (перемещение по пути).

Таблица №1 – Основные характеристики детектора FB.PATH\_TRAVERSAL\_IN

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Path Traversal |
| CWE | CWE22 |
| CVE | --- |
| Возможные последствия | Несанкционированный доступ к файлам за пределами целевого каталога, утечка конфиденциальных данных, возможность модификации системных файлов |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest(String fileName) {  try {  File file = new File("/var/www/uploads/" + fileName);  String content = Files.readString(file.toPath());  System.out.println("File content: " + content);  } catch (IOException e) {  e.printStackTrace();  }  } | public static void correctTest(String fileName) {  try {  Path basePath = Paths.get("/var/www/uploads/").normalize().toAbsolutePath();  Path userPath = Paths.get(fileName).normalize();  Path resolvedPath = basePath.resolve(userPath).normalize();    if (!resolvedPath.startsWith(basePath)) {  throw new SecurityException("Attempted path traversal attack");  }    String content = Files.readString(resolvedPath);  System.out.println("Safe file content: " + content);  } catch (IOException | SecurityException e) {  e.printStackTrace();  }  } |