FB.PT\_RELATIVE\_PATH\_TRAVERSAL

Программное обеспечение использует параметр HTTP-запроса для создания пути, который должен находиться в ограниченном каталоге, но оно не нейтрализует должным образом последовательности, такие как «..», которые могут разрешаться в местоположение, находящееся за пределами этого каталога.

Таблица №1 – Основные характеристики детектора FB.PT\_RELATIVE\_PATH\_TRAVERSAL

|  |  |
| --- | --- |
| Категория критичности | Minor |
| Надежность | Unknown |
| Источник | SpotBugs |
| Тип ошибки | Security Vulnerability |
| CWE | CWE23 |
| CVE | --- |
| Возможные последствия | Обход ограничений файловой системы; доступ к конфиденциальным файлам; модификация защищенных данных; компрометация системы через относительные пути |

Таблица №2 – Примеры срабатывания детектора и устранения уязвимости

|  |  |
| --- | --- |
| Потенциально небезопасное | Корректная конструкция |
| public static void incorrectTest() {  // Некорректно: использование относительных путей без проверки  String userInput = "reports/../../etc/passwd"; // Может прийти из запроса  File file = new File("data/" + userInput);    if (file.exists()) {  System.out.println("File accessed (INSECURE): " + file.getPath());  }  } | public static void correctTest() {  // Корректно: нормализация и проверка пути  String userInput = "reports/../../etc/passwd";  Path basePath = Paths.get("data/").toAbsolutePath().normalize();  Path resolvedPath = basePath.resolve(userInput).normalize();    if (!resolvedPath.startsWith(basePath)) {  System.out.println("Relative path traversal attempt detected");  return;  }    File file = resolvedPath.toFile();  if (file.exists()) {  System.out.println("File accessed securely: " + file.getPath());  }  } |