Non Privilege User can Enable or Disable Registered in openemr/openemr

0

Valid Reported on Mar 28th 2022

Vulnerability Type

Insecure Direct Object Reference

Affected URL

https://localhost/openemr-6.0.0/interface/modules/zend_modules/public/Installer/manage

Affected Parameters

“modAction=enabled”

Authentication Required?

Yes

Issue Summary

Non-privilege users (accounting & front-office) can disable and enable Registered modules. This function is not visible to non-privilege users upon login but a non-privilege user can directly send a POST request to the vulnerable end-point to either disable or enable a module.

Recommendation

The openEMR cookie must be checked against the “modAction” parameter sent in the POST request to https://localhost/openemr-6.0.0/interface/modules/zend_modules/public/Installer/manage to ensure that only cookies belonging to Admin or privileged users are allowed to enable/disable registered modules.

Credits

Aden Yap Chuen Zhen (chuenzhen.yap2@baesystems.com)
Rizan Sheikh (rizan.sheikhmohdfauzi@baesystems.com) Ali Radzali

Rizan, Sheikh (rizan.sheikhmohdradzi@baesystems.com) Ali Radzali
(muhammadali.radzali@baesystems.com)

Issue Reproduction

An admin user is able to disable and enable registered modules:

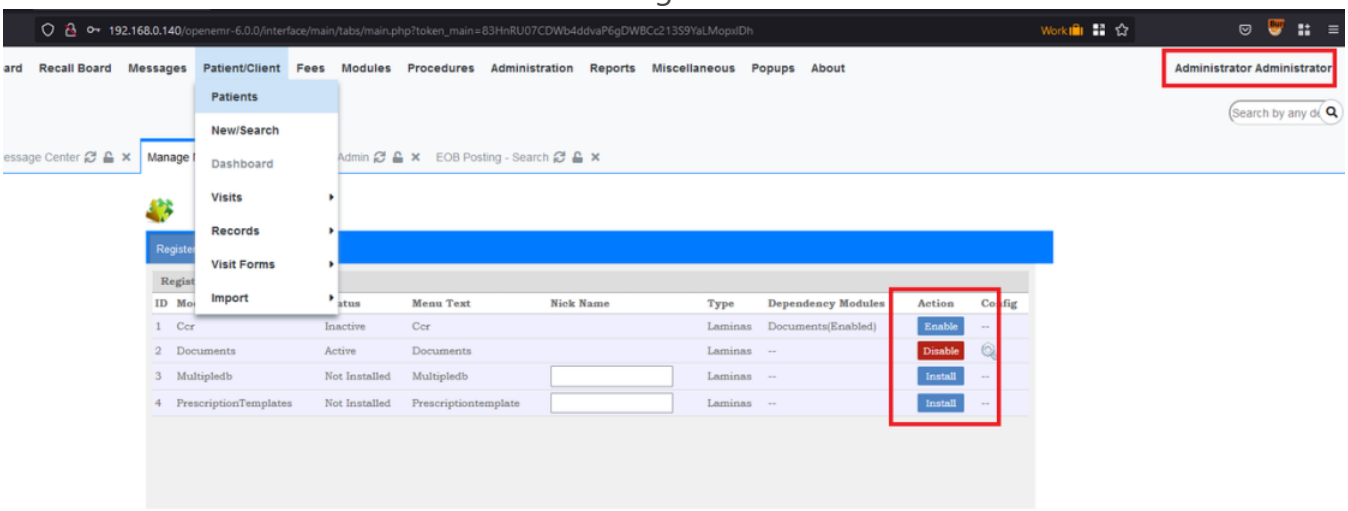

Figure 1: Login as Admin. The Document Module is Currently Disabled.
We used Burp to capture the request of Admin POST request this end point:

```
Host: 192.168.0.140
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/201001
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 53
Origin: http://192.168.0.140
Connection: close
Referer: http://192.168.0.140/openemr-6.0.0/interface/modules/zend_modules/
Cookie: OpenEMR=wa2ubMTCtCWeMvZcad%2CtgbXtvYNdGm%2CTpjZ35HztCGO1Sxd4

modId=2&modAction=enable&mod_enc_menu=&mod_nick_name=
```

Figure 2: Captured Request using Burp. OpenEMR cookie and modAction Parameter are tempered by Non-Privilege User.
We swap out the OpenEMR cookie with a non-privilege user such as Accountant and was still able to enable/disable modules:

able to enable/disable modules.

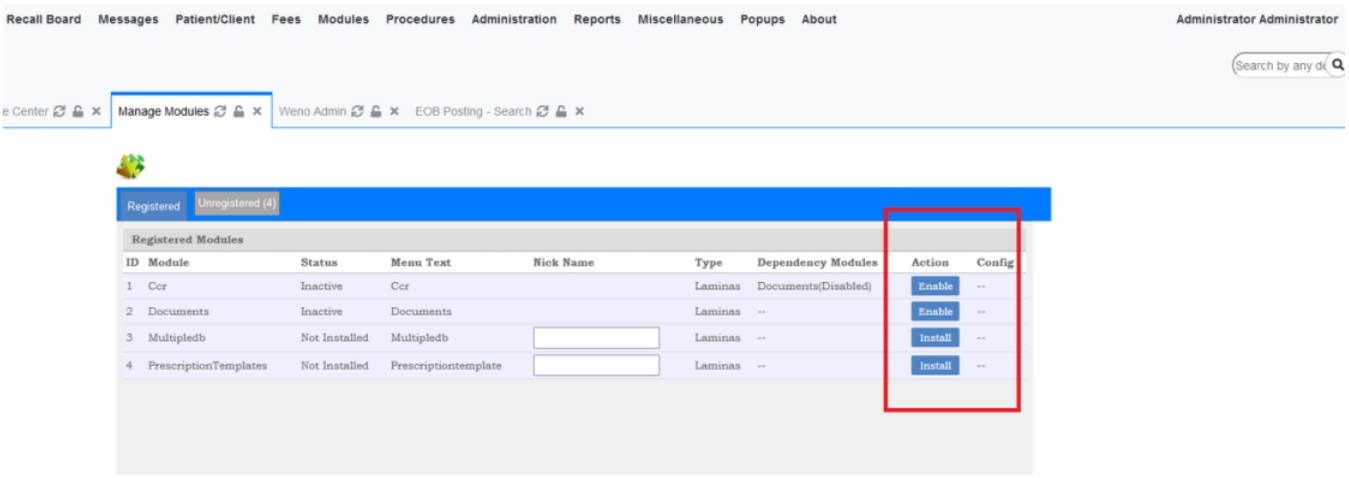

Figure 3: Registered Modules as Seen using Admin account After Non-Privilege User Had Tampered it.

Noticed that the Module function is not visible using non-privilege user. However, by capturing the POST request by Admin in step no 2, we are able to determine the vulnerable end-point to send modifications to the Module function.

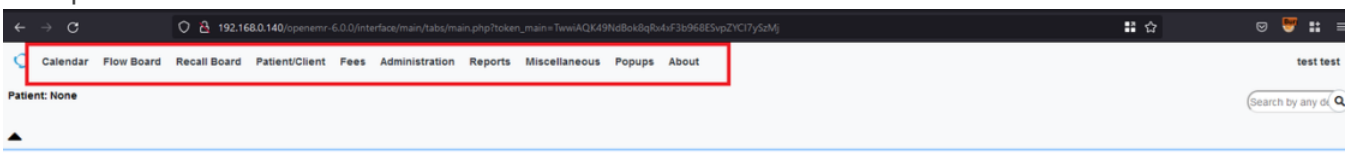

Figure 4: Module Function is Not Visible by Non-Privilege Users

References

- This bug was already reported and fix by Openemr project team. Kindly reach out to Brad in case of questions. Details of patch at: https://www.open-emr.org/wiki/index.php/OpenEMR_Patches

CVE
CVE-2022-1461
(Published)

Vulnerability Type
CWE-1220: Insufficient Granularity of Access Control

Severity
High (8.1)

Visibility
Public

Status
Fixed

Found by

r00t.pgp
@r00tpgp
amateur

This report was seen 510 times.

We are processing your report and will contact the **openemr** team within 24 hours.
8 months ago

We have contacted a member of the **openemr** team and are waiting to hear back 8 months ago

A **openemr/openemr** maintainer has acknowledged this report 8 months ago

A **openemr/openemr** maintainer validated this vulnerability 8 months ago

r00t.pgp has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

A **openemr/openemr** maintainer 8 months ago — Maintainer

A preliminary fix has been placed in the development codebase:
https://github.com/openemr/openemr/commit/619db1d7d7bf5e6a31e7d0489c068998bc9e9327

This fix will be included in the next 6.1.0 patch 1 (6.1.0.1) . After we release 6.1.0 patch 1, then we will confirm the fix at that time.

r00t.pgp 8 months ago — Researcher

Dear @admin I've already ping the maintainer, could you please follow up on the CVE creation?
Tq

Dear @maintainer, could you kindly confirm that CVE can be created for this

A openemr/openemr maintainer 8 months ago Maintainer

Please do not yet make this public yet (I am assuming CVE creation will make it public). I will notify here when we release 6.1.0 patch 1 (in likely 1-2 weeks).

Jamie Slome 8 months ago Admin

Sure, we will wait for your go-ahead on this one 👍

We have sent a fix follow up to the openemr team. We will try again in 7 days. 8 months ago

We have sent a second fix follow up to the openemr team. We will try again in 10 days. 8 months ago

We have sent a third and final fix follow up to the openemr team. This report is now considered stale. 7 months ago

A openemr/openemr maintainer 7 months ago Maintainer

Patch 1 for 6.1.0 (6.1.0.1) has been released, so this fix is now official.

A openemr/openemr maintainer marked this as fixed in 6.1.0.1 with commit 3af1f4 7 months ago

The fix bounty has been dropped ✖

This vulnerability will not receive a CVE ✖

r00t.pgp 7 months ago Researcher

Dear @admin kindly assign cve for thix fix since patch was released. Thank you.

Jamie Slome 7 months ago Admin

Sorted 👍

Sign in to join this conversation