Stored XSS and possible RCE/LFI in case of misconfiguration in thorsten/phpmyfaq

4

Valid Reported on Oct 3rd 2022

Description

phpmyfaq has a feature to restore from a backup the entire application. An attacker with admin grant can export the configuration and re-upload the same file bypassing all the backend sanitization and controls.

Proof of Concept XSS

login as admin
go to backup page
Create a backup and download it
Edit or add some query to file
in this case i edited the content of a category in order to fire an XSS on the admin panel or homepage
navigate some page and see the xss (homepage, list categories etc).
PoC-Payload:

```
-- Table: faqcategories
INSERT INTO faqcategories (id,lang,parent_id,name,description,user_id,group_id,active,image,show_home) VALUES (2,'it',0,'test2"><img src=x onerror=alert(document.cookie)>',L
```

#MISCONF
In case of misconfiguration of the SQL service user grant. An attacker could abuse of that by reading/write sensitive file.
Example (read file grant) 1:
Read ssh keys, or passwd etc...

```
SELECT LOAD_FILE('/etc/passwd')
```

Example (write file grant) 2:
write a php shell file in the root of the server web (the path is discovered from the system information-> Server Document Root)

```
SELECT  'some php code '  INTO dumpfile '/sitepath/somefile.php'
```

Impact

This vulnerability allow an attacker to take control of the entire database and in some cases read arbitrary file or execute shell commands by writing malicious php file.

CVE
CVE-2022-3608
(Published)

Vulnerability Type
CWE-79: Cross-site Scripting (XSS) - Stored

Severity
High (7.2)

Registry
Other

Affected Version
3.2

Visibility
Public

Status
Fixed

Found by

Hakiduck
@mike993
pro

Fixed by

Thorsten Rinne
@thorsten
unranked

This report was seen 976 times.

Chat with us

We are processing your report and will contact the **thorsten/phpmyfaq** team within 24 hours.

2 months ago

Hakiduck modified the report 2 months ago

We have contacted a member of the thorsten/phpmyfaq team and are waiting to hear back 2 months ago

Thorsten Rinne 2 months ago Maintainer

Great work @mike993 👌 Could you kindly propose/submit a fix for this vulnerability? Any help is appreciated.

Thorsten Rinne 2 months ago Maintainer

@mike993 An attacker with admin permissions has easier ways of adding malicious code into the application.

Hakiduck 2 months ago Researcher

Thank you @Thorsten. Sure, you are right, with an administrative user it is easier to add malicious code, but unfortunately many applications for this reason are not protected, or are superficially protected.

Often an application administrator is also not the system / server administrator. So the application must prevent the users (also admin) to go out of its context.
In this specific case the backup and restore functionality is a feature, but there is no reason for a user to edit it before upload it.

Currently two solutions come in my mind:

Sanitize also the inbound queries from the restore function (I think it is very time-consuming to develop)

Encrypt the backup file before download and decrypt it after upload. This limits the ability to arbitrarily modify the database structure or bypass the various security checks. (i think is faster to develop)

this is an example: https://www.pakainfo.com/encrypt-and-decrypt-files-using-php/

The second solution i think is better in this case.

i hope i was helpful.

Thorsten Rinne 2 months ago Maintainer

Thanks for the hint, I thought about something like this, too. As this solution would be a breaking change, I would add this feature for the upcoming 3.2 release.

Hakiduck 2 months ago Researcher

you're welcome. If is not a problem for you, can you mark as valid this submission?

feel free to contact me for any help.

Thorsten Rinne validated this vulnerability 2 months ago

Hakiduck has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

We have sent a fix follow up to the thorsten/phpmyfaq team. We will try again in 7 days.
2 months ago

Thorsten Rinne marked this as fixed in 3.2.0-alpha with commit 37123e 2 months ago

Thorsten Rinne has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Thorsten Rinne 2 months ago Maintainer

Please feel free to review the patch. I used libsodium as mcrypt is deprecated in PHP 8. As it's a breaking change for the current 3.1 release cycle I added the verification of backups for the upcoming 3.2 release

Hakiduck 2 months ago Researcher

Hello @Thorsten, i reviewed the last commit code and now the backup/restor
ok. Very good job! In the next few days I will test it directly on my servers.

@maintainer can I ask you if we can go ahead with the CVE request?

@maintainer can I ask you if we can go ahead with the CVE request?
@admin could we get CVE?

Pavlos a month ago Admin

@mike993 just waiting for maintainer approval then we can go ahead with your CVE :)

Thorsten Rinne a month ago Maintainer

You can go ahead

Hakiduck a month ago Researcher

@admin You can go ahead for the CVE.

Pavlos a month ago Admin

Sign in to join this conversation