

Improper Access Control in bookstackapp/bookstack

0

Valid Reported on Nov 27th 2021

Description

A user with API access can view any attachment which they do not have read access to because read permissions are not being checked at the API attachments read controller.

Proof of Concept

- 1: From default installation give the "Public" role access to system API
- 2: Upload attachment normally to a private page, attachment is now located at `http://[BOOKSTACK-URL]/attachments/1`
- 3: Logout and access `http://[BOOKSTACK-URL]/api/attachments/1` to find Base64 encoded attachment, if we were to go to `http://[BOOKSTACK-URL]/attachments/1` it says we need to login.
- 4: Trying the above with a user account (let us say viewer) allows one to access `http://[BOOKSTACK-URL]/api/attachments/1` but cannot access `http://[BOOKSTACK-URL]/attachments/1`, also proving that access is not being checked at the API controller.

Impact

This vulnerability is capable of allowing users with API access to access confidential attachment data which the users would not have read access to.

Occurrences

AttachmentApiController.php L78L98

missing this->checkOwnablePermission('page-view', \$page);

CVE
CVE-2021-4026
(Published)


Vulnerability Type
CWE-284: Improper Access Control

Severity
Medium (6.5)

Visibility
Public

Status
Fixed

Found by




haxatron

@haxatron

pro

Fixed by



Dan Brown

@ssddanbrown

maintainer

This report was seen 477 times.

- We are processing your report and will contact the bookstackapp/bookstack team within 24 hours.

a year ago
- haxatron modified the report

a year ago
- haxatron modified the report

a year ago
- haxatron modified the report

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

a year ago
- Dan Brown

a year ago

Thanks once again for reporting @haxatron.

From my testing I think the Attachment-Get endpoint is okay. The [query](#) uses the [visible](#) scope which applies permission control to the fetching of the attachment as can be [seen here](#).

I think the difference in behaviour that you're seeing is due to a wider issue. If the guest user is provided the "Access System API" permission then public viewers (if accessing with a bookstack-session-cookie present) can still access the API even if the "Allow public access" setting is disabled. So this would affect all API endpoints.

Realistically I think it'd be very rare, in actual production use, that an admin has provided the public user with API access where they have "Allow public access" as disabled but it's still possible and likely unexpected following the description for that setting. Definitely something to get patched soon.

Dan Brown validated this vulnerability a year ago

haxatron has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

haxatron a year ago

Researcher

Hmm. That's weird, I still see this difference in behaviour if API system access is granted to the Viewer account

Normal

API

haxatron a year ago

Researcher

Ok I think I have narrowed down the case where the scenario applies. If the book is available to a user and page the attachment is attached to is a draft, then the difference in results will be observed with a user account above.

haxatron a year ago

Researcher

So the improper access control in this case would be users with API system access can access attachments attached to draft pages even though draft pages should remain private

Dan Brown a year ago

Can confirm the above scenario. Have addressed as part of BookStack v21.11.2 in addition to the public API access issue this led me to discover. Thank you @haxatron!

Dan Brown marked this as fixed in 21.11.2 with commit b4fa82 a year ago

Dan Brown has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

AttachmentApiController.php#L78L98 has been validated ✓

Sign in to join this conversation

2022 © 418sec

huntr

home

hacktivity

leaderboard

FAQ

contact us

terms

privacy policy

part of 418sec

company

about

team