Media Management - Bug # 48044

Status:	Resolved	**Priority:**	Should have
Author:	Frans Saris	**Category:**	
Created:	2013-05-08	**Assigned To:**	Frans Saris
Updated:	2013-05-27	**Due date:**	
Subject:	PHP Fatal error when installing naw_securedl		

Description

PHP Fatal error: require_once(): Failed opening required '..../typo3conf/ext/media/Resources/Private/Php/user_secure_download.php'

Related issues:

related to Media Management - Task # 48047: Release Media 1.0.0	**Resolved**	**2013-05-08**

Associated revisions

Revision ac5a2e04 - 2013-05-27 11:11 - Frans Saris

[BUGFIX] Fix implementation of naw_securedl hook

Hook is now FAL aware and checks against field sys_file.fe_groups
added by Media extension.

The patch also brings some changes related to:

- Fix the default pid, #48302
- Fix some unit tests along the way
- Improve the method processMagicCall from the asset repository
- Fix string escaping in query

Change-Id: I7f5ca4407f7f3eaf654442a0cf0a0ed489087c14
Fixes: #48044
Fixes: #48302
Reviewed-on: https://review.typo3.org/20636
Reviewed-by: Fabien Udriot
Tested-by: Fabien Udriot

History

#1 - 2013-05-08 10:47 - Gerrit Code Review

Patch set 1 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#2 - 2013-05-11 17:48 - Gerrit Code Review

Patch set 2 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#3 - 2013-05-13 21:29 - Gerrit Code Review

Patch set 3 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#4 - 2013-05-13 21:36 - Gerrit Code Review

Patch set 4 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#5 - 2013-05-15 15:50 - Gerrit Code Review

Patch set 5 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#6 - 2013-05-17 10:08 - Gerrit Code Review

Patch set 6 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#7 - 2013-05-17 10:09 - Fabien Udriot

Hi,

Frans, I finally managed to review your patch. It looks to me the approach envisaged try covering (too) many cases. My personal preference would be to stick to the scope of Media and only handle file under its control. It also means letting up to the developer implementing its own security check against files not managed by Media. Basically, I would go with this logic:

1. check if the file can be find in Media
2. **If the answer is no**, don't do anything and let continue further security check (kind of chain of responsibility pattern)
3. **If the answer is yes**, check if permissions applied against the file.

Following this, I am sorry to have removed much of your code in the last patch ;(but I thought it is not lost since it still could be re-used for your own needs.

I have also updated the configuration text from the EM regarding permission. I recommend having a look at it for the sake of clarification of the "concept" of permission. For now, activating the permission requires to set permissions on each file which can be a burden for the user because it will have deny by default policy. Though, in the long term, my wish / goal is to make Media multi-storage aware. When so, you would be able to have a public storage with no permission applied (allowed by default) next to a restricted storage (denied by default policy).

Also, important to mention, that I have tested with master version of EXT:naw_securedl and I had to face a bug which looks to be required applying http://forge.typo3.org/issues/48269.

#8 - 2013-05-17 21:37 - Frans Saris

Hi,

Fabien why did you leave out the check for processed images? That also can be content you want to protect and since there is a link to the original

Sticking to one storage is fine for now. But would love to see media as a tool + api for multi storage handling. The tool is bind to one storage but my intention was to make the api multi storage aware. But will leave that for a next version of Media :)

I hoop to find some time in the next days to test your version. Looked through your changes already and by reading it looked good so far.

In de master of naw_securedl the problem with the ?1234 in the thumbnail url is fixed?

#9 - 2013-05-21 09:14 - Gerrit Code Review

Patch set 7 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#10 - 2013-05-22 14:13 - Gerrit Code Review

Patch set 8 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#11 - 2013-05-22 14:14 - Fabien Udriot

Hi Frans

answering on the ticket because there is more room for long post ;)

> *Fabien do you know if I do something wrong? or is it a bug in the media api?*

I was able to reproduce the bug and see it more as an Extbase bug. However, it is also true that we are not in an 100% Extbase context... To work around we could use $asset->getFrontendUserGroups()->getFirst() which will return null if no record is found.

Though, as let in the comment in Gerrit, the check violates the "deny from all" principle which is the default when activating permission and I would be for avoiding that. On the other hand I totally see your need and I would have the following proposition: for now, we totally skip the support of secure images (but not for other files of course) and postpone it until we have multi-storage capability. This would be a trade-off for not messing up with the concept of permission IMO. On the top of that, image permissions are not all handled in the BE module so there would be some inconstancy and more code required. As info, you can define in the setting of EXT:naw_securedl not to handle images by setting key "filetype" which has the effect not to rewrite URL on the FE. I have updated the documentation accordingly.

I am also keeping for later the code related to the EXT:naw_securedl not supporting "?1234567890" (since we are not supporting secure images as a first go). As a fix, we could use your code which makes sense or, also, open an issue in EXT:naw_securedl and ask them to fix the issue at the source.

Related to the multi-storage implementation, I can tell I have that in my todo list and need it for one of our customer in a close future. It could be the next feature I would like to have but not for 1.0.0, though.

#12 - 2013-05-22 19:02 - Frans Saris

This way you have no option to make asset public available.
When you have media and naw_securedl installed and asset is found but your not loggedin you get 401.

Only skipping the check for images isn't the way to go in my opinion.

I think we should handle this as every other element. The "deny from all" only makes sence for BE users. In FE no group set means public available. In next version we can try to find a "for every login" group option.

This solution is a blocker fot me. Also the sequrity setting from em conf isn't respected here so there is now way to turn this of when both extensions are installed.

#13 - 2013-05-22 19:31 - Fabien Udriot

- Target version set to 1.0.0

> *The "deny from all" only makes sence for BE users.*

Well, why would you want to protect a file in the BE but make it possible to access it from the FE? This looks as a security leak.

I would suggest disabling the hook because it seems the situation is not satisfying but keep it as example. Once we have properly addressed the issue, we can re-enable it. Would it be ok?

#14 - 2013-05-22 20:13 - Frans Saris

> *The "deny from all" only makes sense for BE users.*
>
> *Well, why would you want to protect a file in the BE but make it possible to access it from the FE? This looks as a security leak.*

No, this is the way it work everywhere in TYPO3. You can deny a BE user access to a part of the file tree. But that doesn't mean that part of the file tree isn't available for the public user in the FE.

- For BE access the rule "deny from all" apply's for all non admin BE Users.
- For FE access the rule "available for all" apply's until a group is assigned to a page or content element.

The FE and BE rights aren't linked.

> *I would suggest disabling the hook because it seems the situation is not satisfying but keep it as example. Once we have properly addressed the issue, we can re-enable it. Would it be ok?*

Yes, disabling this hook is the better option for now. Until the whole access rights layer is properly integrated.

Had a quick look at the way FE groups are set for pages and content elements. There the groups with the UIDs -2 and -1 are used for "available for all loggedin users" and "Hidden for loggedin users". The -2 option would make sense for Asset management to but isn't accepted at the moment because there is no FE group record with UID -2.

#15 - 2013-05-23 10:00 - Fabien Udriot

Hi Frans,

> *No, this is the way it work everywhere in TYPO3. You can deny a BE user access to a part of the file tree. But that doesn't mean that part of the file tree isn't available for the public user in the FE.*

But when creating a new page for instance, the default is the page is hidden for the public. I really would like, when having permission, a file is not accessible by default unless you do something explicitly. It sounds as a requirement when dealing with sensible data. Perhaps we could also "play" with the hidden flag or do something with the -2, -1 FE user group as pointed out...

Anyway, the hook will be disabled in the next patch.

#16 - 2013-05-23 10:00 - Gerrit Code Review

Patch set 9 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#17 - 2013-05-23 11:11 - Frans Saris

Fabien Udriot wrote:

> *Hi Frans,*
>
> > *No, this is the way it work everywhere in TYPO3. You can deny a BE user access to a part of the file tree. But that doesn't mean that part of the file tree isn't available for the public user in the FE.*
>
> *But when creating a new page for instance, the default is the page is hidden for the public. I really would like, when having permission, a file is not accessible by default unless you do something explicitly. It sounds as a requirement when dealing with sensible data. Perhaps we could also "play" with the hidden flag or do something with the -2, -1 FE user group as pointed out...*

Think that something for the default indexing settings just like the default category etc.
Would be nice if you can set that for each storage. Or even better overrule it per directory (or mountpoint) like DAM.
But then the media backend module has to support this so you can select a storage or even directory when uploading new files.
Current media backend module is nice for a small set of assets. But when you have a set of a few thousand al in one directory it is not maintainable anymore. Or what's your opinion about this.

#18 - 2013-05-23 13:38 - Fabien Udriot

> *Would be nice if you can set that for each storage. Or even better overrule it per directory (or mountpoint) like DAM.*
> *But then the media backend module has to support this so you can select a storage or even directory when uploading new files.*

Indeed, this is my plan to go in that direction. I have created this one #48485

> *Current media backend module is nice for a small set of assets. But when you have a set of a few thousand al in one directory it is not maintainable anymore. Or what's your opinion about this.*

Since Media is driven by categories and not by directories you shouldn't worry about having tons of files in same directory. Directory is just a way of categorizing assets in a special way and categories looks more flexibility and powerful IMHO. Besides, I also have on my radar #42587 for improving search capabilities and would like implementing something like that but it will take time (and sponsoring...).

#19 - 2013-05-27 11:10 - Gerrit Code Review

Patch set 10 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/20636

#20 - 2013-05-27 11:35 - Frans Saris

- Status changed from Accepted to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:ac5a2e04474e38efbe4e5ae853f9b6659c9c6dd4.