

Core - Bug # 52016

Status:	Closed	Priority:	Must have
Author:	Franz Holzinger	Category:	Install Tool
Created:	2013-09-15	Assigned To:	
Updated:	2014-03-18	Due date:	
TYPO3 Version:	6.2		
PHP Version:	5.3		
Complexity:			
Is Regression:	No		
Sprint Focus:			
Subject:	wrong recommendations for typo3temp permission		
Description			
<p>The typo3temp folder has the full permissions set: 777</p> <p>drwxrwsrwx 18 username groupname 4096 Sep 15 11:50 typo3temp</p> <p>However the folder structure in the Install Tool shows this message:</p> <p>/typo3temp has wrong permission</p> <p>Target permission are 2770 but current permission are 2777</p> <p>This recommendation is wrong. And other messages are wrong as well. If I set the rights back to 2770 then the Install Tool would fail to execute on the Welcome page and end up in endless PHP error messages:</p> <p>-----</p> <p>Fatal error: Uncaught exception 'TYPO3\CMS\Install\Exception' with message 'Could not create session folder in typo3temp/. Make sure it is writeable!' in</p> <p>/home/myuser/public_html/subfolder/typo3_src-6.2.0alpha2/typo3/sysex/install/Classes/Service/SessionService.php:140 Stack trace:</p> <p>#0 /home/myuser/public_html/subfolder/typo3_src-6.2.0alpha2/typo3/sysex/install/Classes/Service/SessionService.php(124):</p> <p>TYPO3\CMS\Install\Service\SessionService->ensureSessionSavePathExists('/home/porte-bon...') #1</p> <p>/home/myuser/public_html/subfolder/typo3_src-6.2.0alpha2/typo3/sysex/install/Classes/Service/SessionService.php(84):</p> <p>TYPO3\CMS\Install\Service\SessionService->getSessionSavePath() #2</p> <p>/home/myuser/public_html/subfolder/typo3_src-6.2.0alpha2/typo3/sysex/core/Classes/Utility/GeneralUtility.php(4112):</p> <p>TYPO3\CMS\Install\Service\SessionService->__construct() #3 [internal function]:</p> <p>TYPO3\CMS\Core\Utility\GeneralUtility::makeInstance('TYPO3\CMS\Insta...') #4 /home/myuser/public_html/subfolder/typo3_sr in</p> <p>/home/myuser/public_html/subfolder/typo3_src-6.2.0alpha2/typo3/sysex/install/Classes/Service/SessionService.php on line 140</p>			
Related issues:			
related to Core - Task # 52668: Install Tool: Remove permission checking and ...		Resolved	2014-03-26
related to Core - Bug # 51732: Permissions Issue in 6.2alpha2 install tool "d...		Closed	2013-09-04

History

#1 - 2013-09-15 21:33 - Chris topher

What you describe is not a TYPO3 problem, but a unix issue.

Assigning 777 permissions to a folder and thus making it world-read- and -writable is a bad idea. The proper way to fix this is to set the correct group and user as owner of all files and folders of your TYPO3 installation.

#2 - 2014-02-07 10:59 - Kai Blaschke

There rather is a problem with the hard-coded permission TYPO3 uses in 6.2, and in no way an "issue" in UNIX-like systems. I've got the same problem on my own server, which i **could** solve myself by using different UNIX permissions so TYPO3 would work, but this would actually introduce security issues, instead of fixing them. I'll explain why.

Many hosting companies run PHP using FastCGI, either using Apache or Lighttpd. To make the environment secure, each hosting user gets his own UID/GID, and each PHP-CGI process runs with those IDs. The webserver, in contrast, will use a different UID/GID combination, let's use "www" in this example, and having the customers using user and group names like "web1". The server will have a directory tree like this:

```
drwxr-xr-x root root /var/www
drwxr-x--- web1 www /var/www/web1
drw-rw-r-- web1 web1 /var/www/web1/index.html
drwxr-x--- web2 www /var/www/web2
drw-rw-r-- web2 web2 /var/www/web2/index.php
drw-rw-r-- web2 web2 /var/www/web2/typo3temp/gjsdgsdgsdg.gif
```

Using this structure, only web1 and www can access any files inside web1's web root, but web2 cannot (e. g. by using system() to circumvent open_basedir restrictions), because he isn't a member of the "www" group nor using the UID web1. But, and this is the problem, any files uploaded by the user via FTP or created by a PHP script will belong to the UID *and* primary GID of the web user. Setting the permissions to those enforced by TYPO3 6.2, the tree of web2 will look like this:

```
drwxr-xr-x root root /var/www
drwxr-x--- web1 www /var/www/web1
drw-rw-r-- web1 web1 /var/www/web1/index.html
drwxr-x--- web2 www /var/www/web2
drw-rw---- web2 web2 /var/www/web2/index.php
drw-rw---- web2 web2 /var/www/web2/typo3temp/gjsdgsdgsdg.gif
```

The index.php will still execute normally, and the script has access to everything just as it should. But all static files created in typo3temp, including the folder itself, are now inaccessible to the web server, as static content is delivered by a process using the web server's user/group "www". You'll get the correct HTML page, but all resources fail to load with a 403 forbidden error.

I often read "Why should a file be world-readable? It's totally insecure to do so!" in the issue tracker and forums discussing the problem. But, even if files inside the user's home directory are flagged as being world-readable, they actually aren't, because the parent folder permissions limit access to a single user and a single group.

My suggestion to fix this issue would be to differentiate between script, folder and static content permissions, and let the user choose the best option for each. On hosting environments, where a single UID/GID combination is used to execute scripts and deliver static content, the currently hard-coded permissions would be the right ones. In a FastCGI-based environment, one would choose 770 (or 660) for PHP scripts, 775 for folders and 664 for simple files. Given the above example, this would still be a safe set of permissions. Additionally, the TYPO3 root folder permissions shouldn't be enforced, because the user may install the CMS in his home directory, or in a subdirectory, where different permissions are required.

#3 - 2014-03-18 10:26 - Ernesto Baschny

- Category set to *Install Tool*
- Status changed from *New* to *Closed*
- Target version set to *6.2.0*

This was fixed in #52668. New defaults are 0664 and 2775.