TYPO3 Core - Bug #54833

Check default salting method first when determining salting method

2014-01-08 13:49 - Christoph Dörfel

Status:	Closed	**Start date:**	2014-01-08
Priority:	Should have	**Due date:**	
Assignee:		**% Done:**	100%
Category:	Authentication	**Estimated time:**	0.00 hour
Target version:	next-patchlevel		
TYPO3 Version:	6.2	**Complexity:**	easy
PHP Version:		**Is Regression:**	No
Tags:		**Sprint Focus:**	

Description

When SaltFactory::determineSaltingHashingMethod($saltedHash) gets called, we iterate over all available hashing methods, not prioritising the default salting method: SaltedPasswordsUtility::getDefaultSaltingHashingMethod($mode = TYPO3_MODE).
This can result in unnecessary password updates when the salting method "isValidSaltedPW($saltedHash)" returns TRUE for similar hashing implementations.

Associated revisions

Revision 1b74cb49 - 2014-03-27 00:39 - Markus Klein

[BUGFIX] Check default salting method first

Prioritise default salting hashing method when determining
the salting hashing method of a given salted hash.

Fixes rare cases when the method "isValidSaltedPW()" returns TRUE
for similar salting implementations.

Resolves: #54833
Releases: 6.2
Change-Id: I58eb214f171de9f285a7818edebd925eb8164888
Reviewed-on: https://review.typo3.org/26692
Reviewed-by: Anja Leichsenring
Tested-by: Anja Leichsenring
Reviewed-by: Jigal van Hemert
Reviewed-by: Stefan Neufeind
Tested-by: Stefan Neufeind

History

#1 - 2014-01-08 14:17 - Gerrit Code Review

- Status changed from New to Under Review

Patch set 1 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.
It is available at https://review.typo3.org/26692

#2 - 2014-01-08 15:14 - Markus Klein

I don't get the problem.
isValidSaltedPW() returns TRUE we found a valid hashing algorithm, so why do we care about the default at all?

#3 - 2014-01-08 23:07 - Christoph Dörfel

Markus Klein wrote:

> I don't get the problem.
> isValidSaltedPW() returns TRUE we found a valid hashing algorithm, so why do we care about the default at all?

Yes, but SaltFactory::determineSaltingHashingMethod($saltedHash) also saves the first valid hashing algorithm in SaltFactory::$instance; Most of the time, the default algorithm will also be the valid one. This saves some unnecessary instancing, but there is another point:

The fe_login extension compares the default hashing method with the one that SaltFactory::determineSaltingHashingMethod($saltedHash) stores as

SaltFactory::$instance. When the salting instances aren't equal, the users password is being updated after login.

Now consider you made an extension that builds upon the BlowfishSalt (as en example). The password hash in the database will read $2y$-something for PHP >= 5.4 and $2a$ for PHP < 5.4. Then BlowfishSalt:: isValidSaltedPW() will return TRUE in PHP < 5.4 and your extension (that created the hash in the first place) is never checked. The instance in SaltFactory::$instance is now different from your hashing algorithm. This is wrong. Although you can not safely determine the correct salting algorithm for when two implementations return TRUE for isValidSaltedPW, you should at least prefer the default hashing algorithm.

edit:
Reason:
I made a salting instance that uses the password_XX() methods of PHP >= 5.5 or emulates them for PHP < 5.5. PHP 5.5 now features a constant PASSWORD_DEFAULT that references the currently strongest algorithm. This constant is designed to change over time as new and stronger algorithms are added to PHP. This means that you can't be sure that two hashing implementations won't both return TRUE.

#4 - 2014-03-06 23:53 - Gerrit Code Review

Patch set 2 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.
It is available at https://review.typo3.org/26692

#5 - 2014-03-06 23:54 - Gerrit Code Review

Patch set 3 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.
It is available at https://review.typo3.org/26692

#6 - 2014-03-08 11:01 - Gerrit Code Review

Patch set 4 for branch **master** of project **Packages/TYPO3.CMS** has been pushed to the review server.
It is available at https://review.typo3.org/26692

#7 - 2014-03-27 01:30 - Markus Klein

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset 1b74cb499d687002761ec1b0797766ef59f35b72.

#8 - 2018-10-02 12:07 - Benni Mack

- Status changed from Resolved to Closed