

## TYPO3.Flow - Bug #46636

### Authentication does not work any longer without redirects

2013-03-25 09:58 - Marco Falkenberg

<b>Status:</b>	Resolved	<b>Start date:</b>	2013-03-25
<b>Priority:</b>	Must have	<b>Due date:</b>	
<b>Assignee:</b>	Bastian Waidelich	<b>% Done:</b>	100%
<b>Category:</b>	Security	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.0	<b>Complexity:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>	No		

**Description**

After applying the patch [#46352](#) authentication via HTTP-Basic (PersistedUsernamePasswordProvider & UsernamePasswordHttpBasic-Token) throws

#1222268609: Access denied (0 denied, 0 granted, 1 abstained)

#### Associated revisions

##### Revision e06e0f2d - 2013-05-07 17:42 - Bastian Waidelich

[BUGFIX] Authentication does not work any longer without redirects

This fixes a regression that made the authenticated roles only available in the security context after a redirect following authentication.

Background:

This is a regression introduced with the 1st level cache added in [ld256b168ff9c6aa4cac8da8957ada237f9236c71](#) but the actual problem is that the PersistenceQueryRewritingAspect initializes the security context if it was not initialized before (since change [l44838de1503cbe49cf3fee51921b731bfaa0cfc5](#)) when intercepting QOM queries setting the context roles to "Anonymous" and "Everybody".

This change adds a new method Context::withoutAuthorizationChecks() that allows you temporarily disable authorization related interceptors e.g. PolicyEnforcement and PersistenceQueryRewriting aspects in order to be able to circumvent authorization in low level operations (for example to fetch the current account in an AuthenticationProvider).

Usage::

```
$this->securityContext->withoutAuthorizationChecks (
    function ($accountRepository, $username, $providerName, &$account) {
        // this will disable the PersistenceQueryRewritingAspect for this one call
        $account = $accountRepository
            ->findActiveByAccountIdentifierAndAuthenticationProviderName($username, $providerName)
    }
);
```

Change-Id: [lb31cd6bcf10504670439d4c700dda0b14e512d80](#)

Related: [#46352](#)

Fixes: [#46636](#)

Releases: master, 2.0

##### Revision b964e06b - 2013-05-07 17:46 - Bastian Waidelich

[BUGFIX] Authentication does not work any longer without redirects

This fixes a regression that made the authenticated roles only available in the security context after a redirect following authentication.

Background:

This is a regression introduced with the 1st level cache added in [ld256b168ff9c6aa4cac8da8957ada237f9236c71](#) but the actual problem is

that the PersistenceQueryRewritingAspect initializes the security context if it was not initialized before (since change l44838de1503cbe49cf3fee51921b731bfaa0cfc5) when intercepting QOM queries setting the context roles to "Anonymous" and "Everybody".

This change adds a new method Context::withoutAuthorizationChecks() that allows you temporarily disable authorization related interceptors e.g. PolicyEnforcement and PersistenceQueryRewriting aspects in order to be able to circumvent authorization in low level operations (for example to fetch the current account in an AuthenticationProvider).

Usage::

```
$this->securityContext->withoutAuthorizationChecks (
    function ($accountRepository, $username, $providerName, &$account) {
        // this will disable the PersistenceQueryRewritingAspect for this one call
        $account = $accountRepository
            ->findActiveByAccountIdentifierAndAuthenticationProviderName($username, $providerName)
    }
);
```

Change-Id: lb31cd6bcf10504670439d4c700dda0b14e512d80

Related: #46352

Fixes: #46636

Releases: master, 2.0

## History

---

### #1 - 2013-04-25 10:32 - Marco Falkenberg

After some debugging I could locate the problem. The thing is that the Security\Context never will be informed about a new authenticated token, and under some circumstances Security\Context->getRoles() will be called **before** the token is authenticated.

This happens e.g. when you use a PersistedUsernamePasswordProvider. It tries to fetch a user from persistence. Then the PersistenceQueryRewritingAspect hooks in which calls Security\Context->getRoles() and... bam: The default roles Everybody and Anonymous are locked. And when UsernamePasswordHttpBasic-authentication is used, the authentication and access to the protected resource afterwards occurs in one request.

### #2 - 2013-04-25 10:36 - Marco Falkenberg

A solution would be to write a new slot which unsets the locked roles and connect it with the authenticatedToken signal of the AuthenticationProviderManager.

### #3 - 2013-04-29 18:59 - Bastian Waidelich

- Subject changed from Broken Authentication via HTTP-Basic to Authentication does not work any longer without redirects
- Priority changed from Should have to Must have
- Target version set to 2.0

Marco, this one cost me some hours too..

The issue is not only true for HTTP authentication but for most authentication providers! Usually the issue doesn't matter though, because of a redirection to some other action.

The problem is that the **PersistenceQueryRewritingAspect** calls **SecurityContext::getRoles()** for all **Query::execute()** invocations. That means that the following code

```
$account = $this->accountRepository->findActiveByAccountIdentifierAndAuthenticationProviderName($credentials[
'username'], $this->name);
```

in **PersistedUsernamePasswordProvider** implicitly initializes the security context with the roles **Everybody & Anonymous**

### #4 - 2013-04-30 08:47 - Bastian Waidelich

- Status changed from New to Accepted
- Assignee set to Bastian Waidelich

### #5 - 2013-04-30 09:42 - Gerrit Code Review

- Status changed from Accepted to Under Review

Patch set 1 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/20346>

#### #6 - 2013-04-30 10:02 - Christopher Hlubek

What about using some context to disable the PersistenceQueryRewritingAspect in the AccountRepository? This is a feature we would also need for policy enforcement. This would allow to execute queries (or call methods) without security in an explicit way. This is also useful for example in commands where usually no account is authenticated or in low-level tasks.

Pseudocode:

```
$this->securityContext->withoutAuthorization(function() use ($accountRepository, $credentials, &$amp;account) {  
    $account = $accountRepository->findActiveByAccountIdAndAuthenticationProviderName($credentials['username'], $this->name);  
});
```

The aspect could check for a global flag in SecurityContext and have an early return.

Bastian Waidelich wrote:

The problem is that the **PersistenceQueryRewritingAspect** calls **SecurityContext::getRoles()** for all **Query::execute()** invocations.

#### #7 - 2013-04-30 10:18 - Karsten Dambekalns

IIRC we ignored security in the aspect if the SecurityContext cannot be initialized - that used to work fine in the past. Hm.

#### #8 - 2013-04-30 10:57 - Bastian Waidelich

Karsten Dambekalns wrote:

IIRC we ignored security in the aspect if the SecurityContext cannot be initialized - that used to work fine in the past. Hm.

There's even a test **rewriteQomQueryDoesNotRewriteQueryIfSecurityContextCannotBeInitialized** in **PersistenceQueryRewritingAspectTest** ;)

#### #9 - 2013-04-30 11:00 - Bastian Waidelich

Bastian Waidelich wrote:

Karsten Dambekalns wrote:

IIRC we ignored security in the aspect if the SecurityContext cannot be initialized - that used to work fine in the past. Hm.

..But this has been changed with the aim to "enforce Query Rewriting more reliably": <https://review.typo3.org/#/c/16106/>

So maybe Christophers suggestion would be the way to go

#### #10 - 2013-04-30 11:18 - Bastian Waidelich

Bastian Waidelich wrote:

So maybe Christophers suggestion would be the way to go

I just gave this a quick try and it seems to work fine! I'll push a WIP

#### #11 - 2013-04-30 16:38 - Gerrit Code Review

Patch set 2 for branch **master** has been pushed to the review server.  
It is available at <https://review.typo3.org/20346>

#### #12 - 2013-05-07 12:30 - Gerrit Code Review

Patch set 3 for branch **master** has been pushed to the review server.  
It is available at <https://review.typo3.org/20346>

#### #13 - 2013-05-07 14:08 - Gerrit Code Review

Patch set 4 for branch **master** has been pushed to the review server.  
It is available at <https://review.typo3.org/20346>

#### #14 - 2013-05-07 16:52 - Gerrit Code Review

Patch set 5 for branch **master** has been pushed to the review server.  
It is available at <https://review.typo3.org/20346>

**#15 - 2013-05-07 17:42 - Gerrit Code Review**

Patch set 6 for branch **master** has been pushed to the review server.  
It is available at <https://review.typo3.org/20346>

**#16 - 2013-05-07 17:48 - Gerrit Code Review**

Patch set 1 for branch **2.0** has been pushed to the review server.  
It is available at <https://review.typo3.org/20586>

**#17 - 2013-05-07 18:35 - Bastian Waidelich**

- *Status changed from Under Review to Resolved*
- *% Done changed from 0 to 100*

Applied in changeset [e06e0f2dd6eb565f00ae535c780ab13b74de8f92](https://review.typo3.org/20586).