TYPO3.Flow - Bug # 33055

Status:	New	**Priority:**	Must have
Author:	Patrick Pussar	**Category:**	Security
Created:	2012-01-09	**Assigned To:**	
Updated:	2014-04-17	**Due date:**	
PHP Version:	5.3		
Has patch:	No		
Complexity:			
Affected Flow version:	FLOW3 1.0.0		
Subject:	AccessDeniedException instead of WebRedirect		

Description

After defining some restrictions via ACLs on a controller method and defining a WebRedirect I get an AccessDeniedException instead of a redirect.

After do some dirty hack in RequestDispatchingAspect.php:
//} catch (\TYPO3\FLOW3\Security\Exception\AuthenticationRequiredException $exception) { // PPUSSAR: Hot-Patch to get Web-Redirect working
} catch (\TYPO3\FLOW3\Security\Exception $exception) {

it works as expected.

Whether the catch method is to restrictive or the wrong exception is thrown.

Related issues:

related to TYPO3.Flow - Bug # 33078: No Redirect to Login	**New**	**2012-01-10**

History

#1 - 2012-01-10 19:37 - Johannes K

Did you try to call the protected action manually, or via a Fluid generated link?
I'm asking, because to call protected action you also need to pass a csrfToken in the URL.
If the link is generated by Fluid, the URL contains the csrfToken automatically.

Another option is to annotate the action with @FLOW3\SkipCsrfProtection.

No real documentation for this yet, but here is an explanation:
[[http://media.netlogix.de/community/details/artikel/csrf-protection-in-typo3-phoenix-kindly-provided-by-flow3]]

#2 - 2012-01-16 11:46 - Patrick Pussar

It is the first call to the Site. Meaning a user calls a domain like: http://mydomain
This url is linked to a site, which is under ACL restriction. The idea is that the user becomes redirected to the login in case that he is not logged in or can see the content of the page in case that he is already authenticated.

#3 - 2012-01-26 15:27 - Patrick Pussar

The controller method is already annotated with @FLOW3\SkipCsrfProtection

#4 - 2012-11-04 20:06 - Andreas Wolf

I can confirm this problem on latest Flow master. I think that either throwing an "AccessDeniedException" is wrong here or that the RequestDispatchingAspect should also redirect to the login form in case the access was denied.

As far as I understand the concept, "AccessDeniedException"s should be thrown when there is no hope that the user might gain access with a login, i.e. they are already logged in but don't have the neccessary permissions to view the requested resource.
OTOH, the "AuthenticationRequiredException" is thrown when no user has authenticated, but authentication is required to view the desired resource.

The flaws seem to be at various other locations in the system, where no check is performed to see if a user is authenticated or not before throwing an exception - the exception used would depend on that. I'd like to check with the Flow team what part should be fixed here.

#5 - 2013-03-08 11:43 - Adrian Föder

Will/can this maybe be resolved with https://review.typo3.org/#/c/18695/6 ?

#6 - 2014-01-04 20:54 - Andreas Wolf

Adrian Föder wrote:

> *Will/can this maybe be resolved with https://review.typo3.org/#/c/18695/6 ?*

No, it is not, at leas not for the case I experienced. I had a new version deployed while being logged in, leading to my session being destroyed. I got an **AccessDeniedException**, but I would have expected an **AuthenticationRequiredException** (because no user was authenticated).

I'll dig into this again in the next days.

#7 - 2014-04-17 15:54 - Benjamin Heek

I had the same problem.

I fixed it by setting the `TYPO3.Flow.security.authenticationStrategy` to `allTokens`

Because the AuthenticationProviderManager only throws an `AuthenticationRequiredException` when authenticationStrategy is set to `allTokens`

@see TYPO3\Flow\Security\Authentication\AuthenticationProviderManager->authenticate() line 183:

```
if ($this->securityContext->getAuthenticationStrategy() === Context::AUTHENTICATE_ALL_TOKENS) {
                throw new AuthenticationRequiredException(...);
            }
```