TYPO3.Flow - Bug # 12917

Status:	Rejected	**Priority:**	Must have
Author:	Julian Kleinhans	**Category:**	Security
Created:	2011-02-09	**Assigned To:**	Robert Lemke
Updated:	2011-09-09	**Due date:**	
PHP Version:			
Has patch:			
Complexity:			
Affected Flow version:			
Subject:	Access denied by using the HashService Setter injection		

Description

[See https://review.typo3.org/#change,699]

If i use the HashService Setter injection in a protected area as a loggedin user i became an "Access denied"

this is my Policy.yaml

```
resources:
  methods:
    F3_Tutorials_RestrictedAdminArea: 'class(F3\Tutorials\Controller\Admin\.*)'
roles:
  Administrator: []
acls:
  Administrator:
    methods:
      F3_Tutorials_RestrictedAdminArea: GRANT
```

in the Controller\Admin i used a setter injection for the HashService.

```
public function injectHashService(\F3\FLOW3\Security\Cryptography\HashService $hashService) {
  $this->hashService = $hashService;
}
```

and with this method in my code i become the "Access denied" page.. my logs:

```
Successfully re-authenticated tokens for account "test1" [logged in
F3\FLOW3\Security\Aspect\LoggingAspect::logManagerAuthenticate()]

Access denied (0 denied, 0 granted, 1 abstained) to method
F3\Tutorials\Controller\Admin\AccountController::injectHashService(). [logged in
F3\FLOW3\Security\Aspect\LoggingAspect::logJoinPointAccessDecisions()]
```

if i used the property injection

```
/**
* @inject
```

```
    * @var \F3\FLOW3\Security\Cryptography\HashService
    */
    protected $hashService;
```

it works without problems.. only with the setter injection There is NO other ACL in a policy.yaml

History

#1 - 2011-05-31 16:45 - Christopher Hlubek

I had a similar problem that I solved with a different pointcut. I changed the Controller->.*() to Controller->(?!inject)[a-z].*() such that the Policy will not be applied to inject methods.

I consider this as a workaround. The Policy shouldn't be applied to inject methods at all, since the authentication might not be initialized when injecting dependencies.

#2 - 2011-08-04 08:25 - Sebastian Kurfuerst

- Target version changed from 1.0 beta 1 to 1.0 beta 2

we won't manage this for beta1 anymore, postproning

#3 - 2011-09-09 11:21 - Robert Lemke

- Status changed from New to Accepted
- Assigned To set to Robert Lemke

#4 - 2011-09-09 11:30 - Robert Lemke

- Status changed from Accepted to Rejected

I have thought about this one. As I see it, FLOW3 behaves correctly and we shouldn't change the behavior.

The point is, if we'd exclude all inject*() methods from the policies by default, we wouldn't solve the whole problem, because there can still be regular setters (setFoo()) which are used for Dependency Injection. Granting access to any setter by default and even if you explicitly create a policy which doesn't allow access to Class->.* would be dangerous.

Therefore, if the Access Denied exception tells that a method injectSomething() couldn't be called, that should be enough information for either switching to property injection, like Julian did, or adjusting the policy, like Christopher did.

#5 - 2011-09-09 14:17 - Julian Kleinhans

Ok, so we should add this information to the wiki error page