Cross-Site Scripting through Fluid view helper arguments

Moderate **ohader** published **GHSA-hpjm-3ww5-6cpf** on Nov 17, 2020

Package

php **typo3fluid/fluid** (Composer)

Affected versions	Patched versions
2.0.0-2.0.7, 2.1.0-2.1.6, 2.2.0-2.2.3, 2.3.0-2.3.6, 2.4.0-2.4.3, 2.5.0-2.5.10, 2.6.0-2.6.9	2.0.8, 2.1.7, 2.2.4, 2.3.7, 2.4.4, 2.5.11, 2.6.10

Description

Meta

- CVSS: `CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N/E:F/RL:O/RC:C` (5.7)
- CWE-79

Problem

Three XSS vulnerabilities have been detected in Fluid:

1. TagBasedViewHelper allowed XSS throug maliciously crafted `additionalAttributes` arrays by creating keys with attribute-closing quotes followed by HTML. When rendering such attributes, TagBuilder would not escape the keys.
2. ViewHelpers which used the `CompileWithContentArgumentAndRenderStatic` trait, and which declared `escapeOutput = false`, would receive the content argument in unescaped format.
3. Subclasses of AbstractConditionViewHelper would receive the `then` and `else` arguments in unescaped format.

Solution

Update to versions 2.0.8, 2.1.7, 2.2.4, 2.3.7, 2.4.4, 2.5.11 or 2.6.10 of this `typo3fluid/fluid` package that fix the problem described.

Updated versions of this package are bundled in following TYPO3 (`typo3/cms-core`) releases:

- TYPO3 v9.5.23 (using typo3fluid/fluid v2.6.10)
- TYPO3 v10.4.10 (using typo3fluid/fluid v2.6.10)

The specific vulnerabilities are prevented by:

1. Explicitly escaping keys found in the `additionalAttributes` array passed to a TagBasedViewHelper before using them as attribute names.
2. Detecting "content argument" on ViewHelpers using the trait CompileWithContentArgumentAndRenderStatic and escaping it based on the state of `escapeChildren` when `escapeOutput` is toggled off. Escaping still will not occur if explicitly disabled by an enclosing ViewHelper. This homogenises escaping behavior of "content arguments" so the same strategy is used whether the "content" argument is passed as argument or child content.
3. Explicitly defining the `then` and `else` arguments on AbstractConditionViewHelper subclasses as escaped and applying escaping in all cases where escaping is not explicitly disabled by an enclosing ViewHelper.

Affected cases

1. The fix for TagBasedViewHelper does not affect any valid use cases; it only prevents use of maliciously crafted attribute/value arrays passed as `additionalAttributes`.
2. Any case where a ViewHelper with a "content argument" and which defines `escapeOutput = false` is used with the content argument instead of passing variables as child node - e.g. `<v:h content="{variable}" />` instead of `<v:h>{variable}</v:h>` to intentionally circumvent escaping of any HTML in `{variable}`.
3. Any case where a condition ViewHelper is used with `then` or `else` arguments to render a variable containing HTML, excluding cases where the variable is intentionally unescaped - e.g. `<f:if condition="1" then="{variable -> f:format.raw()}" />`, and excluding any cases where a ViewHelper is used as argument value and the ViewHelper intentionally disables escaping - e.g. `<f:if condition="1" then="{f:render(section: 'MySection')}" />` does not escape the `then` argument because `f:render` disables output escaping.

Cases 2 and 3 can be mitigated to allow variables with HTML to not be escaped, by intentionally disabling escaping by chaining the variable used in the argument with `f:format.raw` as described in case 3. Note that this constitutes a potential security issue, for which the template author is solely responsible. Example: `<f:if condition="1" then="{intentionalHtmlVariable}" />` can allow HTML in `{intentionalHtmlVariable}` by adding `-> f:format.raw()` - to become `<f:if condition="1" then="{intentionalHtmlVariable -> f:format.raw()}" />`.

Custom ViewHelpers which use `CompileWithContentArgumentAndRenderStatic` can alternatively pass a 6th argument with value `false` to the call to `registerArgument` which registers the "content argument", which explicitly disables escaping of the argument value: `$this->registerArgument('arg', 'string', 'My argument', false, null, false);`. Note that this constitutes a potential security issue for which the ViewHelper author is solely responsible. **Variables containing HTML should only be allowed after taking great care to prevent XSS through other means, e.g. sanitising the variable before it is assigned to Fluid or only allowing such variables to come from trusted sources.**

Credits

Thanks to Jonas Eberle and Sinan Sekerci (Dreamlab Technologies) who reported this issue and to TYPO3 core merger Claus Due who fixed the issue.

References

- TYPO3-CORE-SA-2020-009

Severity

Moderate

CVE ID

CVE-2020-26216

Weaknesses

No CWEs

Credits

 NamelessCoder

 jonaseberle