TYPO3 Core - Bug #94810

Bug # 94787 (Closed): Tracking issue related to HTML sanitization issues

Unable to disable html sanitize

2021-08-11 11:57 - Robert van Kammen

Status:	Closed	**Start date:**	2021-08-11
Priority:	Should have	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
TYPO3 Version:	10	**Complexity:**	
PHP Version:	7.2	**Is Regression:**	
Tags:		**Sprint Focus:**	

Description

Currently it is not possible to disable the html sanitize functionality.
Tested with the <f:format.html>...</f:format.html> function
When I set lib.parseFunc.htmlSanitize = 0 and lib.parseFunc_RTE.htmlSanitize = 0 the html is still sanitized.

This can be testen with:

```
<f:format.html><form action=""><input name="test" /></form></f:format.html>
```

The issue seems to be in the class: TYPO3\CMS\Frontend\ContentObject\ContentObjectRenderer in the function parseFunc.
Line: if ($conf['htmlSanitize'] ?? true) {

History

#1 - 2021-08-11 14:28 - Robert van Kammen

- TYPO3 Version changed from 9 to 10

#2 - 2021-08-11 17:57 - Oliver Hader

Where is that <form action=""><input name="test" /></form> coming from? Did you allow to add HTML forms in CKEditor?!

From my point of view, given <form> example is markup that shall be used as is, thus, it should be <f:format.raw><form action=""><input name="test" /></form></f:format.raw> instead of <f:format.html>@.

Besides that, the very example you used, was encoded in previous versions (before TYPO3 v11.3.2, v10.4.19, v9.5.29), so even without having the HTML sanitizer in place

```
<f:format.html><form action=""><input name="test" /></form></f:format.html>
```

produced in older/previous versions already

```
&lt;form action=&quot;&quot;&gt;&lt;input name=&quot;test&quot; /&gt;&lt;/form&gt;
```

#3 - 2021-08-11 18:00 - Oliver Hader

- Status changed from New to Needs Feedback

#4 - 2021-08-11 18:00 - Oliver Hader

- Description updated

#5 - 2021-08-11 18:00 - Oliver Hader

- Description updated

#6 - 2021-08-11 18:01 - Oliver Hader

- Parent task set to #94787

#7 - 2021-08-12 07:24 - Robert van Kammen

The html is not added by users.
We have some html code generated by a standalone fluid view that is rendered this way.
Using the format raw viewhelper is better in this situation and I will replace the viewhelper.

However the disable problem still seems to be there when changing the htmlSanitize value.

#8 - 2021-08-13 06:18 - Oliver Hader

Robert van Kammen wrote in #note-7:

> The html is not added by users.
> We have some html code generated by a standalone fluid view that is rendered this way.
> Using the format raw viewhelper is better in this situation and I will replace the viewhelper.

Yes, in case the block contains only generated markup (e.g. by Fluid template or view-helper), using <f:format.raw> is the better alternative - in that scenario not sanitization would happen.

> However the disable problem still seems to be there when changing the htmlSanitize value.

I successfully tested the following again (f:format.html uses TypoScript path lib.parseFunc_RTE per default):

```
<f:format.html><b onmouseover="alert(1)">Just testing</b></f:format.html>
// + having TypoScript `lib.parseFunc_RTE.htmlSanitize = 0`
// ... as a result XSS `onmouseover` is not removed
```

However, there was one case with parseFuncTSPath="", when it was not possible to disable HTML sanitization:

```
<f:format.html parseFuncTSPath=""><b onmouseover="alert(1)">Just testing</b></f:format.html>
// in this case NO TypoScript path is used and thus, it cannot be disabled via TypoScript
```

... that's why a change in issue #94786 introduces a new `htmlSanitize` argument for that view-helper - it is still pending at the time of writing this comment.

```
<f:format.html parseFuncTSPath="" htmlSanitize="0"><b onmouseover="alert(1)">Just testing</b></f:format.html>
// ... as a result XSS `onmouseover` is not removed
```

#9 - 2021-08-13 07:10 - Robert van Kammen

Thanks for the information. With the introduction of this new viewhelper this issue is resolved.

#10 - 2021-08-13 10:49 - Henning Lange

Robert van Kammen wrote in #note-9:

> Thanks for the information. With the introduction of this new viewhelper this issue is resolved.

Will this new ViewHelper be included in the upcoming 9.5.30/10.4.20 point release?

Apart from that, we have exactly the same problem. Disabling HTML sanitization globally in Typoscript just doesn't work at all, using <format.raw> doesn't help as well. We still have a couple of tags in our Fluid templates like <script>, <input> or <figure> that are rendered in HTML entities, making some pages unusable and we just don't get it to work with the suggestions given so far! Funny thing is, some templates seem to work, some don't, but we couldn't figure out so far the reason for this...

Couldn't you make the list of blocked tags/elements configurable via Typoscript, just like styles.content.allowTags? Unfortunately, that setting is ignored as well... :-/

(9.5.29 / 10.4.19 seems to me like the most frustrating / work-generating TYPO3 point release update in years, at least if you have to to with Frontend / Fluid templating!)

#11 - 2021-08-13 15:48 - Oliver Hader

- Status changed from Needs Feedback to Closed

Henning Lange wrote in #note-10:

> Robert van Kammen wrote in #note-9:
>
> > Thanks for the information. With the introduction of this new viewhelper this issue is resolved.

> Will this new ViewHelper be included in the upcoming 9.5.30/10.4.20 point release?

Yes, the changes of #94825 and #94786 are target to be released in the next versions (11.3, 10.4, 9.5 & probably ELTS as well).

> Apart from that, we have exactly the same problem. Disabling HTML sanitization globally in Typoscript just doesn't work at all, using <format.raw> doesn't help as well. We still have a couple of tags in our Fluid templates like <script>, <input> or <figure> that are rendered in HTML entities, making some pages unusable and we just don't get it to work with the suggestions given so far! Funny thing is, some templates seem to work, some don't, but we couldn't figure out so far the reason for this...

Having <script> and co sanitized is a signal, that some top-level processing is causing that - probably search for TypoScript parseFunc calls of <f:format.html> wrapping in template files. Besides that, #94837 introduces stack-trace debugging in case tags/attrs get sanitized.

> Couldn't you make the list of blocked tags/elements configurable via Typoscript, just like styles.content.allowTags? Unfortunately, that setting is ignored as well... :-/

Unfortunately not, because e.g. just having allowTags = img did not mitigate cross-site scripting, this setting actually allowed <img src="wrong" onerror="alert(1)">, just try it in previous TYPO3 versions. Besides that attr wild-cards like data-* or aria-* cannot be defined, and there was no checks for attribute values at all.

> (9.5.29 / 10.4.19 seems to me like the most frustrating / work-generating TYPO3 point release update in years, at least if you have to to with Frontend / Fluid templating!)

Believe me, I personally did not expect that many messed up websites. The interesting things is, that there were sites that did not have any problem at all with HTML sanitizer. How do you explain that?