☰ View Issue Details

ID	Project	Category	View Status	Date Submitted	Last Update
0027056	mantisbt	security	public	2020-06-21 02:29	2020-09-11 09:02

Reporter	hanno	**Assigned To**	dregad		
Priority	normal	**Severity**	minor	**Reproducibility**	always
Status	■ closed	**Resolution**	fixed		
Product Version	2.1.0				
Target Version	2.24.2	**Fixed in Version**	2.24.2		

Summary

0027056: CVE-2020-16266: HTML injection (maybe XSS) via custom field on view_all_bug_page.php

Description

The content of the filter variable in the view of the view_all_bug_page.php is not filtered, allowing a thirdparty to inject HTML.

This would usually be a Cross Site Scripting Vulnerability, but the Content Security Policy header blocks executing scripts. However it might still be possible to achieve XSS by invoking functionality from the bundled javascript libraries.

The output should be properly html-escaped.

Steps To Reproduce

1. Create a custom string type field.
2. Create a form that sends HTML code in the custom field 1 value (see poc).

poc:
<form action="https://[hostname]/view_all_set.php?f=3&quot; method="POST">
<input name="custom_field_1[]" value="<h1 style=color:red>INJECTION</h1>">
<input type=submit>
</form>

Tags

No tags attached.

Relationships

related to	0021935	■ closed	cproensa	Filter api refactoring, manage stored filters
related to	0027275	■ closed	dregad	CVE-2020-25288: HTML Injection on bug_update_page.php

Activities

dregad
2020-06-22 03:37
developer ~0064114

Thanks for the bug report, I'll have a look at it.

Did you request a CVE for the issue ? If so, please let us know the ID; otherwise we'll take care of it. How would you like to be credited for the finding ?

dregad
2020-06-22 06:21
developer ~0064116

Confirmed HTML injection it is ! (and potential XSS if CSP settings allow)

dregad
2020-06-22 06:52
developer ~0064118

print_filter_values_custom_field() function seems to be the most appropriate place to add the escaping - @cproensa, what do you think ?

```
diff --git a/core/filter_form_api.php b/core/filter_form_api.php
index 6280cabbb..114deaa0d 100644
--- a/core/filter_form_api.php
+++ b/core/filter_form_api.php
@@ -1855,7 +1855,7 @@ function print_filter_values_custom_field( array $p_filter, $p_field_id ) {
            if( filter_field_is_none( $t_val ) ) {
                $t_strings[] = lang_get( 'none' );
            } else {
-               $t_strings[] = $t_val;
+               $t_strings[] = string_attribute( $t_val );
            }
            $t_inputs[] = '&lt;input type=&quot;hidden&quot; name=&quot;custom_field_' . $p_field_id . '[]&quot; value=&quot;' . string_attribute( >
}
```

hanno
2020-06-22 06:54
reporter ~0064119

One addition: I haven't discovered this myself, this was reported to me because I run a public mantis instance which is covered by a (non-payment) bug bounty.

I just ask the finder if he wants to be publicly credited for this.

dregad
2020-06-22 07:17
developer ~0064120

Problem exists since refactoring of filter display in Mantis 2.1.0 (see ~~0021935~~)

hanno
2020-07-04 02:47
reporter ~0064150

Finder of the vulnerability is Jaime Andrés Restrepo, please credit him accordingly when publishing an update + security advisory.

dregad
2020-08-03 05:32
developer ~0064220

CVE request 938519 sent

dregad
2020-08-03 12:08
developer ~0064222

CVE-2020-16266 assigned.

Related Changesets

MantisBT: master 9ef8f23a 2020-06-22 02:55 dregad Details Diff	Fix XSS in view_all_bug_page.php (CVE-2020-16266) Hanno Boeck reported a stored cross-site scripting (XSS) vulnerability, originally discovered by Jaime Andres Restrepo. Improper escaping on view_all_bug_page.php allowed a remote attacker to inject arbitrary HTML into the page by saving it into a text Custom Field, leading to possible code execution in the browser of any user subsequently viewing the issue (if CSP settings allow it). Prevent the attack by properly escaping the custom field's contents before display. Fixes ~~0027056~~	**Affected Issues** ~~0027056~~
	mod - core/filter_form_api.php	Diff File