New issue

Jump to bottom

Stored XSS in User Instructions Widget #3025

Closed wjmccann opened this issue on May 17, 2020 · 3 comments

Labels bug report

wjmccann commented on May 17, 2020 • edited

Version: 1.4

Within the Edit User Instructions field where you can enter source code you are able to generate scripting that then executes in the user's browser when they click on the instructions page.

POC:

```
<body onload=alert(1)></body>
```

Additionally,

```
<a href=# onmouseover=alert(1)>Text</a>
```

will also execute scripting in the browser.

POC video is available here: https://youtu.be/SpFmM03JI40

wjmccann added the bug report label on May 17, 2020

galaktipus commented on Sep 7, 2020

Any commit fixing the issue?

GaryAllan commented on Sep 7, 2020 Collaborator

@galaktipus
HTML is allowed in the User Instructions field so <script> is implicitly permitted.

This isn't a bug/issue, this is a feature request to limit the User Instructions field to a safe(r) subset of HTML (if such a thing actually exists!)

The ticket doesn't state why this potential XSS is an issue. An adversary would require admin rights to edit the User Instructions field and can therefore already perform all actions in the application.

GaryAllan closed this as completed in c1a610b on Sep 14, 2020

GaryAllan added a commit that referenced this issue on Sep 14, 2020

Bugfix: Stored XSS in instructions widgets. Fixes #3025 70f1e82

GaryAllan commented on Sep 14, 2020 Collaborator

The code did attempt to remove <script> tags so I've updated with a more robust DOM parser.

```
$instructions->instructions = str_replace("<script", "<div class='error'><xmp><script", $instructions->instructions);
$instructions->instructions = str_replace("</script>", "</script></xmp></div>", $instructions->instructions);
preg_replace('#<script(.*?)>(.*?)</script>#is', '', $_POST['instructions']);
```

Assignees

No one assigned

Labels

bug report

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants