

New issue

Jump to bottom

## Store XSS #1212

Closed

3as0n opened this issue on Jun 19, 2020 · 10 comments

3as0n commented on Jun 19, 2020

## Describe your problem

<https://github.com/bludit/bludit/blob/master/bl-kernel/ajax/logo-upload.php>

Logo upload only determines the suffix, but not the content, which causes XSS and the user can inject any javascript and html code in the page

payload

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg">
  <polygon id="triangle" points="0,0,0,50,50,0" fill="#009900" stroke="#004400"/>
  <script type="text/javascript">
    alert(1);
  </script>
</svg>
```

常规 高级 SEO 社交网络(SNS) 图片 语言 Custom fields Logo

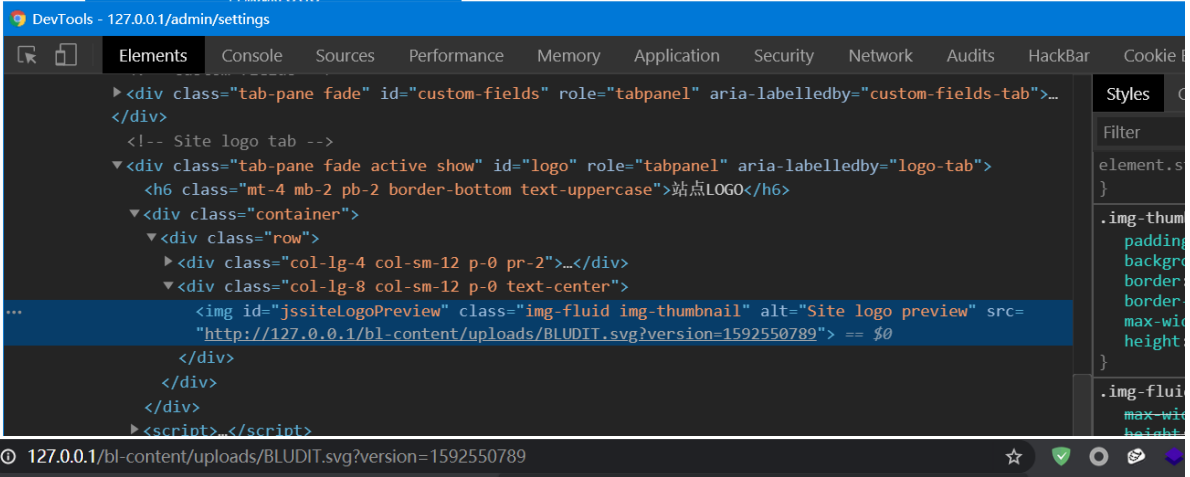
站点LOGO

上传图片

Browse

Site logo preview

删除LOGO



127.0.0.1/bl-content/uploads/BLUDIT.svg?version=1592550789

127.0.0.1 显示

1

确定

galaktipus commented on Jun 25, 2020 · edited

Any known fix for issue yet? Seems like it was assigned to CVE-2020-15006

3as0n commented on Jun 25, 2020

Author

Yes, this is the cve number. I think we can either delete the svg format or add a check mechanism. But it affects performance.  
...

dignajar commented on Jun 25, 2020

Member

I will check, is not a problem for Bludit because the users are trusted by the Administrator, I mean Bludit doesn't have user registration.  
  
Feel free to provide a fix for it.  
  
Regards

3as0n commented on Jun 25, 2020

Author

If people enter the background, they can use this vulnerability to spread or fish malicious JS. If someone's domain name is trusted  
...

ghost commented on Jun 26, 2020 • edited by ghost

Vector Images are really important (especially for responsive design), but many Content Management Systems avoid them because detecting and filtering XSS is a horrible job.

However, losing SVG Support in Bludit shouldn't be the way to go. So how about the following Validation function (can be part of the Valid helper class). It's may not perfect now, but it works on the most common XSS attacks and I also checked the used Regular Expression against [MDN's SVG Attribute's list](#) to avoid false positives.

```
<?php
class Valid
{
    static public function svg($content, $requireXmlTag = false, $requireDoctypeTag = false) {
        if($requireXmlTag && strpos($content, "<?xml") !== 0) {
            return false;
        }
        if($requireDoctypeTag && strpos($content, "<!DOCTYPE") === false) {
            return false;
        }

        // Validate XML Format
        try {
            $xml = @new SimpleXMLElement($content);
        } catch(Exception $e) {
            $xml = false;
        }
        if(!$xml) {
            return false;
        }

        // Check Basic XSS
        if(!preg_match("/(<(\/)?script[on[a-z]+=|(java)?script:\/i", $content)) {
            return false;
        }
        return true;
    }
}
```

~ Sam.

ghost commented on Jun 26, 2020 • edited by ghost

We can also add the following regular expression to avoid URLs except the w3.org standards too, which may leads to false positives if someone adds a URL as comment. This is also not fully tested yet.

```
(http.){(:\/\)}?(www\.)?(?!\w3\.org)
```

working-name commented on Jun 26, 2020

@SamBrishes That regex is a good start but it gets complicated fast: <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>

Especially when you can have whitespace in the word 'javascript:'. Let me know if you're serious about using regex for this, I'd be happy to make one that catches the known methods people use to bypass xss checks.

3as0n commented on Jun 26, 2020

Author

It's a good idea.  
...


ghost commented on Jun 27, 2020 • edited by ghost ▾

Especially when you can have whitespace in the word 'javascript'. Let me know if you're serious about using regex for this, I'd be happy to make one that catches the known methods people use to bypass xss checks.

Feel free to bypass the regex, but keep in mind, that it still need to be a valid XML file according to `SimpleXMLElement`.


Of course, we can also use an (adapted) version of the Ulf Harnhammar's [Kses Library](#) and check each single tag and attribute available in a SVG image only, but I don't think that this is really necessary.

Anyway, Bludit should also just allow admins to upload SVGs to restrict the responsibility and I guess it is also important to check the MIME type of images too, next to the file extension. (For example as I'm using it on my [media](#) plugin).

 ghost mentioned this issue on Jun 28, 2020

MIME Type Check for Issue #1218 and #1212 #1219

[↪ Merged](#)

 dignajar added a commit that referenced this issue on Jun 29, 2020

 Merge pull request [#1219](#) from SamBrishes/patch-010 ... 


4282a97

dignajar commented on Feb 22

Member

Hello, with the new version of Bludit v4.0 rc1, I would like to close the old Github issues. If you feel that your issue is not resolved in the latest version, create a new ticket.

- Help and Support use the Forum <https://forum.bludit.org>
- Bugs and new requests here in Github <https://github.com/bludit/bludit/issues>

 dignajar closed this as completed on Feb 22

Assignees

No one assigned

Labels

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants

