☰ View Issue Details

ID	Project	Category	View Status	Date Submitted	Last Update
0030384	mantisbt	security	public	2022-05-25 10:33	2022-06-24 04:05

Reporter	febin	**Assigned To**	dregad		
Priority	normal	**Severity**	major	**Reproducibility**	always
Status	■ closed	**Resolution**	fixed		
Product Version	2.25.4				
Target Version	2.25.5	**Fixed in Version**	2.25.5		

Summary	0030384: CVE-2022-33910: Stored XSS via SVG file upload
Description	MantisBT allows SVG files and that leads to Stored Cross-Site Scripting to account takeover. SVG files are technically XML-based images that can include javascript in them. An attacker can send a maliciously crafted SVG file by attaching it with an issue/bug report and when a user or an admin clicks on the attachment, it will get opened in the browser tab instead of downloading it as a file and the javascript will get executed in his browser, that is capable of doing various stuff like stealing cookies, sending requests on behalf of that user, etc., Severity: HIGH Remediation: Restrict SVG files or sanitize javascript from the SVG data.

Steps To Reproduce

1. Create a new issue report
2. Add the maliciously SVG as an attachment.
3. When a user or admin opens the issue request and clicks on the attachment, the javascript will get executed.

Note: I have attached a sample SVG file with this report as a proof of concept, you can view the SVG file's source code in your browser. The javascript code might not execute because of the CSP that is implemented in this instance, but not all other instances would have CSP implemented and that makes this a valid security issue.

POC (SVG source code):

```
&lt;?xml version=&quot;1.0&quot; standalone=&quot;no&quot;?>
&lt;!DOCTYPE svg PUBLIC &quot;-//W3C//DTD SVG 1.1//EN&quot; &quot;http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd&quot;>

&lt;svg onload=&quot;alert(1);&quot; version=&quot;1.1&quot; baseProfile=&quot;full&quot; xmlns=&quot;http://www.w3.org/2000/svg&quot;>
   &lt;rect width=&quot;300&quot; height=&quot;100&quot; style=&quot;fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)&quot; />

    &lt;script>
alert(&quot;XSS&quot;);
alert(document.cookie);
    &lt;/script>

&lt;/svg>
```

Additional Information

I wish to be credited for the finding and for my name to be included in the CVE report

Name: FEBIN MON SAJI
Email: febinrev811@gmail.com

Tags

No tags attached.

Relationships

related to	0029135	■ closed	dregad	CVE-2022-33910: Unrestricted SVG File Upload leads to CSS Injection

Activities

dregad
2022-05-25 19:17
developer ~0066643

Thanks for the detailed report. I confirm the problem, will look into it.

A similar issue (~~0029135~~) was reported a few months ago using the same SVG attack vector, except that it refers to CSS injection, but I believe it is the same root cause.

atrol
2022-05-26 04:47
developer ~0066644

> but not all other instances would have CSP implemented

CSP headers are standard in MantisBT and active out of the box on any installation.
As long as source code is not changed or dirty 3rd party plugins change the CSP headers, there is no major security issue.

Of course, there is still an issue when

- using outdated browsers that don't support CSP
- opening the file independant from MantisBT after download

I don't see at the moment how this can be prevented, as changing the content itself when uploading or downloading is not an option.

One more option for MantisBT administrators is to use the following options to configure allowed / non-allowed attachment types.

```
/**
 * Files that are allowed or not allowed.  Separate items by commas.
 * eg. 'php,html,java,exe,pl'
 * if $g_allowed_files is filled in NO other file types will be allowed.
 * $g_disallowed_files takes precedence over $g_allowed_files
 * @global string $g_allowed_files
 */
$g_allowed_files = '';

/**
 *
 * @global string $g_disallowed_files
 */
$g_disallowed_files = '';
```

@dregad, we could increase the out of the box security by changing the default settings to something like `$g_allowed_files = 'txt';`
Not sure if this is a good idea, as it will force most of the admins to change the setting after an upgrade.

dregad
2022-06-13 05:23
developer ~0066739

> I don't see at the moment how this can be prevented, as changing the content itself when uploading or downloading is not an option.

I agree that we should not mess with changing file contents.

My initial idea was to simply add `svg` to *$g_disallowed_files* by default, but that does not actually defend against previously uploaded files, and admins who have customized this setting would need to update

So I did some research and found this article https://digi.ninja/blog/svg_xss.php that details the various use cases where users are vulnerable to scripted SVG files. The only one that concerns us is the *direct view* scenario.

Since we channel all attachments downloads through *file_download.php*, it should be fairly straightforward to force SVG files to download instead of being displayed, by means of a `Content-disposition: attachment` header. As it turns out, the script already contains such logic today, but SVG files are currently forced inline so it should be a simple matter of switching the mime type to force them to download instead.

dregad
2022-06-13 06:22
developer ~0066740
Last edited: 2022-06-13 06:22

Actually this prevents scripts execution, but an attacker could still play CSS tricks (~~0029135~~), so I think we also need to disable the ability to upload SVG to be on the safe side.

dregad
2022-06-13 06:31
developer ~0066742

CVE Request 1282365 sent

dregad
2022-06-17 04:55
developer ~0066755

CVE-2022-33910 assigned

dregad
2022-06-17 05:13
developer ~0066757

@febin attached is a proposed patch for review, thanks in advance for your feedback

CVE-2022-33910.patch (2,338 bytes)

febin
2022-06-17 10:42
reporter ~0066760

Svg can include html using <foreignObject> element(works on Firefox only), that can be used for phishing or similar stuff.

dregad
2022-06-19 10:14
developer ~0066768

> Svg can include html using <foreignObject> element(works on Firefox only), that can be used for phishing or similar stuff.

@febin I'm not sure I get your point. The proposed patch completely prevents upload of SVG files by default (can be changed by admin), and as an extra safety measure ensures the SVG files are always downloaded as attachments instead of being opened in a browser tab. I believe other uses cases (display in IMG tags) are not exposed to the vulnerability.

Please clarify and correct me if I'm wrong.

Related Changesets

MantisBT: master-2.25 0d1d7b65 2022-06-13 06:03 dregad Details Diff	Code cleanup: 1 array element per row, sorted Issue ~~0030384~~	**Affected Issues** ~~0030384~~
	mod - file_download.php	Diff File
	Prevent script execution when viewing SVG files	

MantisBT: master-2.25 262ecdde

2022-06-13 06:09

dregad

Details Diff

A cross-site scripting vulnerability allows remote attackers to attach maliciously crafted SVG files to issue reports or bugnotes. When a user or an admin clicks on the attachment, file_download.php will it open the SVG in a browser tab instead of downloading it as a file, causing the javascript to execute. This risk is mitigated by MantisBT's default Content Security Policy, which prevents execution of inline scripts.

This fixes the issue by forcing download as attachment for files of image/svg+xml mime type.

Devendra Bhatla and Febin Mon Saji <febinrev811@gmail.com> both and independently reported this vulnerability.

Fixes ~~0030384~~, CVE-2022-33910

Affected Issues

~~0030384~~

mod - file_download.php	Diff File