

New issue

Jump to bottom

Admin dashboard vulnerable to DOM XSS #969

Closed

0xnibbles opened this issue on Jan 13, 2021 · 0 comments

0xnibbles commented on Jan 13, 2021

Opened this issue because there is not a security advisory or a response from the repo maintainers after sending a report by email. The POC described here used the docker continuous deployment instance (<https://ci.simplcommerce.com>).

POC

The following POC is just an example of many others that are prone to DOM XSS. To better understand and see a more detailed explanation of what is, please see the following link: [https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_\(XSS\)](https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_(XSS))

Using the user register page and the input is the Full Name field at the register page (<https://ci.simplcommerce.com/register>).

Payload - `<script>alert(1)</script>`

Register

Create a new account.

Email	<input type="text" value="testingXSS@simplcommerce.com"/>
Full name	<input type="text" value="<script>alert(1)</script>"/>
Password	<input type="password" value="****"/>
Confirm password	<input type="password" value="****"/>
	<input type="button" value="Register"/>

An user with the name `<script>alert(1)</script>` is registered and stored to the database. This payload appears to not be triggered while browsing through the site but when admin needs to remove this user, the payload is executed.

Below is shown an example where to trigger our malicious payload at <https://ci.simplcommerce.com/admin#!/users>.

SimplCommerce Admin					
Users					
Email	Full Name	Role	Customer Groups	Created On	Actions
		From			
		At	At		
217 records found					
testingXSS@simplcommerce.com	<script>alert(1)</script>	customer		Sep 01, 2020 10:02:28 AM	<input type="button" value="Remove"/>

We can see the registered user and the payload in the full name field. Nothing anormal happens when loading the page. But when trying to **remove the user** and **after clicking the remove button** the script of the malicious payload is executed and the alert box appears.

This behavior confirms the XSS vulnerability. It **requires** to click the remove button, but as soon as is clicked, the exploit is executed. Looking to the inspector from the dev tools, we can see the injected script in the **Bootbox modal body**.

```

<div class="modal-dialog">
  <div class="modal-content">
    <div class="modal-body">
      <button class="bootbox-close-button close" type="button">
        <div class="bootbox-body">
          Are you sure you want to delete this user?
          <script>alert(1)</script>
        </div>
      </div>
    <div class="modal-footer">
      </div>
  </div>
</div>

```

In the SimplCommerce code, this happens in the following line of code at

<https://github.com/simplcommerce/SimplCommerce/blob/master/src/Modules/SimplCommerce.Module.Core/wwwroot/admin/user/user-list.html#L65>

```

17  <div ng-model="user" to="vm.user">
18    <div class="form-group">
19      <input type="text" value="{{user.email}}"/>
20      <input type="text" value="{{user.fullName}}"/>
21      <input type="text" value="{{user.role}}"/>
22      <input type="text" value="{{user.customerGroups}}"/>
23      <input type="text" value="{{user.createdOn}}"/>
24      <input type="button" value="Remove"/>
25    </div>
26  </div>
27  <div ng-click="vm.deleteUser(user)" class="btn btn-danger">
28    <div class="bootbox">
29      <div class="bootbox-body">
30        Are you sure you want to delete this user?
31      </div>
32    </div>
33  </div>

```

When clicked the **delete button**, the function `vm.user(delete)` is executed. If we look for the `user-list.js` file

(<https://github.com/simplcommerce/SimplCommerce/blob/master/src/Modules/SimplCommerce.Module.Core/wwwroot/admin/user/user-list.js>) we can see of the function is doing.

```

26  vm.deleteUser = function deleteUser(user) {
27    bootbox.confirm('Are you sure you want to delete this user: ' + user.fullName, function (result) {
28      if (result) {
29        userService.deleteUser(user)
30          .then(function (result) {
31            vm.getUsers(vm.tableStateRef);
32            toastr.success(user.fullName + ' has been deleted');
33          })
34          .catch(function (response) {
35            toastr.error(response.data.error);
36          });
37      }
38    });
39  };

```

At line 27, the value of `user.fullName` field is added directly to the bootbox modal without proper sanitization making it vulnerable to XSS.


Also, at line 33 a toast is launched when we confirm deleting the user and again `user.fullName` is added to the toast. The same payload can be triggered twice if we follow this use case.

Fixing the vulnerability

As the bootstrap.js maintainer does not fix the XSS vulnerability, to protect the users of Simplicommerce from being attacked is recommended to sanitize input **before adding it to any bootstrap modal or dialog**. The following link explains the approach to validate user data <https://docs.microsoft.com/en-us/aspnet/core/security/cross-site-scripting?view=aspnetcore-3.1#where-should-encoding-take-place>.

The next functions are just suggestions to validate input data. Upon your goals, you can choose the one(s) that best fit to your project:

- `HttpUtility.HtmlEncode` - <https://docs.microsoft.com/en-us/dotnet/api/system.web.httputility.htmlencode?view=netcore-3.1>
- `WebUtility.HtmlEncode` - <https://docs.microsoft.com/en-us/dotnet/api/system.net.webutility.htmlencode?view=netcore-3.1>

 **thiennn** added a commit that referenced this issue on Jan 13, 2021


 **#969** fixed xss

aa0d0fe

 **thiennn** added a commit that referenced this issue on Jan 13, 2021

 **#969** fixed xss (#970)

✖ 8bb0b10

 **thiennn** closed this as completed on Jan 18, 2021

 **afernandes** pushed a commit to afernandes/SimplCommerce that referenced this issue on May 20, 2021

 **simplcommerce#969** fixed xss (simplcommerce#970)

bc711bc

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

