XSS vulnerability with default `onCellHtmlData` function in hhurz/tableexport.jquery.plugin 1

Valid Reported on Apr 5th 2022

Description

If you can jam some nasty code into a table-cell, you can force this script to perform arbitrary javascript when someone tries to export the table using this library. An example used against us was:

```
"><iMg SrC="x" oNeRRor="alert(1);">
```

It looks like, if you don't specify an `onCellHtmlData` function, the default one is used here:
https://github.com/hhurz/tableExport.jquery.plugin/blob/986adee1cfa1022e5f8b3d085c333b26782d6aca/tableExport.js#L2079-L2123
That one includes the line:
https://github.com/hhurz/tableExport.jquery.plugin/blob/986adee1cfa1022e5f8b3d085c333b26782d6aca/tableExport.js#L2084
Which, according to the JQuery folks, is definitely XSS-able - https://api.jquery.com/jQuery.parseHTML/ (scroll down to 'Security Considerations').
A user can route around the default implementation of onCellHtmlData by providing their own function for it, but I still think the default implementation should be 'safe' for all uses.
Users of this library who do *not* attempt to export tables of user-provided data are probably immune. But I would figure most table exports are going to be of some kind of dynamic data (why export a static table?), so I suspect that most uses of this library will be vulnerable to these attacks.
(I also think that Bug Bounty researchers are just finding implementations of this library and attacking them, as opposed to letting you know that there might be a problem, but that's neither here nor there).
We were able to route around the problem by setting `htmlContent` to `true` - but our users hate that so I was looking for another workaround (and also trying to explain why it happened in the first place!) and then I figured out the `onCellHtmlData` problem.

Proof of Concept

Proof of Concept

Render a table with a cell with the value `"><iMg SrC="x" oNeRRor="alert(1);">` and then export it as CSV or PDF (and probably a few others).

Impact

Transmitting cookies to third-party servers. Sending data from secure sessions to third-party servers

References

- This is how we found out about the bug in our own application :(

CVE
CVE-2022-1291
(Published)

Vulnerability Type
CWE-79: Cross-site Scripting (XSS) - Stored

Severity
High (7.6)

Registry
Other

Affected Version
1.22.0

Visibility
Public

Status
Fixed

Found by

Brady Wetherington
@uberbrady
maintainer

This report was seen 728 times.

Chat with us

We are processing your report and will contact the hhurz/tableexport.jquery.plugin team within 24 hours. 8 months ago

We created a **GitHub Issue** asking the maintainers to create a `SECURITY.md` 8 months ago

Brady 8 months ago Researcher

Here is an actual exploit: https://live.bootstrap-table.com/code/uberbrady/11033 (this uses Bootstrap Tables, which is a thin wrapper of this library)

hhurz validated this vulnerability 8 months ago

Brady Wetherington has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

hhurz marked this as fixed in **1.25.0** with commit **dcbaee** 8 months ago

The fix bounty has been dropped ✖

This vulnerability will not receive a CVE ✖

Sign in to join this conversation