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

1

Valid Reported on Aug 28th 2022

Description

Input fields allowing Markdown Input are vulnerable to XSS. This requires Superadmin permissions though.

Proof of Concept

Steps to reproduce:

```
1. Log in to the admin account
2. Go to Admin -> General Settings
3. Enter the Payload in the `Login Note` and `Dashboard Message` fields.
4. Go to the Dashboard & confirm the XSS in the dasboard message. Logout ar
```

Payload:

```
[XSS](javascript:alert(document.location))
```

Impact

The impact is JavaScript Code Execution. However, superadmin privileges are required to edit the vulnerable input fields.

Occurrences

dashboard.blade.php L20

According to the Parsedown Readme, to prevent XSS it is required to set t

```
$Parsedown->setSafeMode(true);
```

login.blade.php L31

According to the Parsedown Readme, to prevent XSS it is required to set the safemode:

```
$Parsedown->setSafeMode(true);
```

CVE
CVE-2022-3035
(Published)

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

Severity
Medium (5.9)

Registry
Other

Affected Version
6.0.10

Visibility
Public

Status
Fixed

Found by

vautia
@vautia
master

Fixed by

snipe
@snipe
maintainer

Chat with us

This report was seen 1,685 times.

We are processing your report and will contact the **snipe/snipe-it** team within 24 hours.
3 months ago

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

A **snipe/snipe-it** maintainer has acknowledged this report 3 months ago

snipe validated this vulnerability 3 months ago

vautia has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

snipe marked this as fixed in **v6.0.11** with commit **9cf5f3** 3 months ago

snipe has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

login.blade.php#L31 has been validated ✔

dashboard.blade.php#L20 has been validated ✔

Sign in to join this conversation