


Reflected XSS when using flashMessages or languageDictionary

Moderate Izychowski published GHSA-jr3j-whm4-9wvm on Jun 4, 2021

Package

 **auth0-lock** (npm)

Affected versions

<= 11.30.0

Patched versions

11.30.1

Description

Overview

Versions before and including 11.30.0 are vulnerable to reflected XSS. An attacker can execute arbitrary code when the library's

- flashMessage feature is utilized and user input or data from URL parameters is incorporated into the flashMessage .
- languageDictionary feature is utilized and user input or data from URL parameters is incorporated into the languageDictionary .

Am I affected?

You are affected by this vulnerability if you are using auth0-lock version 11.30.0 or lower and all of the following conditions apply:

- You are utilizing flashMessage feature.
- User input or data from URL parameters is incorporated into the flashMessage .

An example of a vulnerable snippet where query parameters are used to populate the text property of a flashMessage .

```
var params = new URLSearchParams(location.search);
var errorMessage = params.get('error__message');
var showParams = {};

if (!errorMessage === true) {
  showParams.flashMessage = {
    type: 'error',
    text: 'We were unable to log you in. ' + errorMessage,
  };
};

lock.show(showParams);
```

OR

- You are utilizing languageDictionary feature.
- User input or data from URL parameters is used in languageDictionary properties.

An example of a vulnerable snippet where query parameters are used to populate the socialLoginInstructions property of a languageDictionary .

```
var params = new URLSearchParams(location.search);
var instruction = params.get('instruction');

var options = {
  languageDictionary: {
    emailInputPlaceholder: "something@youremail.com",
    title: "title",
    socialLoginInstructions: instruction
  },
};

var lock = new Auth0LockPasswordless(
  CLIENT_ID,
  DOMAIN,
  options
);

lock.show()
```

How to fix that?

Upgrade to version 11.30.1 .

Will this update impact my users?

The fix uses DOMPurify to sanitise the flashMessage and languageDictionary inputs. If you are including inline JavaScript in these fields, like script tags or onclick attributes, these will be removed.

Severity

Moderate

CVE ID

CVE-2021-32641

Weaknesses

