Arbitrary code execution via function parsing

Critical oxyno-zeta published GHSA-j3rv-w43q-f9x2 on Aug 13

Package

react-editable-json-tree (npm)

Affected versions	Patched versions
<2.2.2	2.2.2

Description

Impact

Our library allows strings to be parsed as functions and stored as a specialized component, `JsonFunctionValue`. To do this, Javascript's `eval` function was used to execute strings that begin with "function" as Javascript. This was an oversight that unfortunately allows arbitrary code to be executed if it exists as a value within the JSON structure being displayed. Given that this component may often be used to display data from arbitrary, untrusted sources, this is extremely dangerous.

One important note is that users who have defined a custom `onSubmitValueParser` callback prop on the `JsonTree` component should be ***unaffected***. This vulnerability exists in the default `onSubmitValueParser` prop which calls `parse`.

Patches

We have decided on a two-pronged approach to patching this vulnerability:

1. Create a patch update that adds a workaround **which is not enabled by default** to preserve backwards-compatibility
2. On the next major update, **we will enable this workaround by default**

The workaround we have decided on is adding a prop to `JsonTree` called `allowFunctionEvaluation`. This prop will be set to `true` in v2.2.2, so you can upgrade without fear of losing backwards-compatibility.

We have also implemented additional security measures as we know many people may not read the details of this vulnerability, and we want to do the best we can to keep you protected. In v2.2.2, we switched from using `eval` to using `Function` to construct anonymous functions. This is better than `eval` for the following reasons:

- Arbitrary code should not be able to execute immediately, since the `Function` constructor explicitly *only creates* anonymous functions
- Functions are created without local closures, so they only have access to the global scope

This change has brought a *slight* potential for breaking backwards-compatibility if users for some reason were relying on side-effects of our usage of `eval`, but that is beyond intended behavior, so we have decided to go ahead with this change and consider it a non-breaking change.

Workarounds

As mentioned above, there are a few scenarios you must consider:

If you use:

- **Version `<2.2.2`**, you must upgrade as soon as possible.
- **Version `^2.2.2`**, you must explicitly set `JsonTree`'s `allowFunctionEvaluation` prop to `false` to fully mitigate this vulnerability.
- **Version `>=3.0.0`**, `allowFunctionEvaluation` is already set to `false` by default, so no further steps are necessary.

References

None.

For more information

If you have any questions or comments about this advisory:

- Open an issue in the GitHub repo

Severity

Critical 10.0 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Changed

Confidentiality	**High**
Integrity	**High**
Availability	**High**

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

CVE ID

CVE-2022-36010

Weaknesses

CWE-95

Credits

Phanabani

oxyno-zeta