Prototype Pollution in alvarotrigo/fullpage.js

2

✓ Valid Reported on Feb 26th 2022

Description

`fullPage` utils are available to developers using `window.fp_utils`. They can use these utils for their own use-case (other than fullPage) as well. However, one of the utils `deepExtend` is vulnerable to Prototype Pollution vulnerability.
Javascript is "prototype" language which means when a new "object" is created, it carries the predefined properties and methods of an "object" with itself like `toString`, `constructor` etc.
By using prototype-pollution vulnerability, an attacker can overwrite/create the property of that "object" type. If the victim developer has used that property anywhere in the code, then it will have severe effect on the application.
For e.g.:

```
var obj = {};
console.log(obj.A); // undefined
obj["__proto__"].A = 1;
console.log(obj.A);  // 1
var new_obj = {};
console.log(new_obj.A); // 1  -> exploit
```

Proof of Concept

STEP 1: Visit https://alvarotrigo.com/fullPage demo.
STEP 2: Run the following code in dev tools console
NOTE: I am asking to run this in `console` for PoC purpose only. The real-world exploitation scenario may vary.

```
var o = {};
o.toString();
var obj = window.fp_utils.deepExtend({},{"constructor": {"p
```

STEP 3: Call `toString` prototype function

```
o.toString();
```

and you will see an alert pop-up showing XSS exploitation.

Impact

Prototype pollution can be used to create/overwrite predefined properties and methods of object type. It can lead to XSS, change code logic, DoS etc. based on the application code.

References

- Prototype Pollution in AngularJS

CVE
CVE-2022-1295
(Published)

Vulnerability Type
CWE-1321: Prototype Pollution

Severity
High (7.3)

Visibility
Public

Status
Fixed

Found by

Rohan Sharma
@r0hansh
unranked

Fixed by

Álvaro
@alvarotrigo
unranked

This report was seen 804 times.

We are processing your report and will contact the **alvarotrigo/fullpage.js** team within 24 hours. 9 months ago

We created a **GitHub Issue** asking the maintainers to create a `SECURITY.md` 9 months ago

Álvaro 9 months ago Maintainer

Thanks for that!

Replacing the deepExtend function for the following one seems to fix the bug:

```
 function deepExtend( a, b ) {
    a = a || {};

    for ( var prop in b ) {
      a[ prop ] = b[ prop ];
    }
    return a;
}
```

I'll be releasing fullPage.js version 4 this week with this change (and many others)

Álvaro 9 months ago Maintainer

Ok, this function won't do exactly the same, It will fail to merge deep objects.

Any proposed solution to this vulnerability?

Álvaro 9 months ago Maintainer

Ok, Seeing the references I did this and it seems to have solve it:
https://jsfiddle.net/7c9uo3yn/

```
if (!obj.hasOwnProperty(key) || key == '__proto__' || key == 'constructor'){
```

```
if (!obj.hasOwnProperty(key) || key == '__proto__' || key == 'constructor'){
                continue;
}
```

Although... I'm not sure if I should do:

```
key == '__proto__'
```

Or

```
key == 'prototype'
```

Any hints on that?

Rohan Sharma 9 months ago Researcher

yes, as mentioned. you just need to check for keys `__proto__` and `constructor`

Rohan Sharma 9 months ago Researcher

no need to check for `prototype`

Jamie Slome 8 months ago Admin

@maintainer - would you kindly mark the report as valid and confirm the fix now that a patch has been released in v4?

You can do these actions using the dropdown below ⬇️

We have contacted a member of the **alvarotrigo/fullpage.js** team and are waiting to hear back 8 months ago

Álvaro validated this vulnerability 8 months ago

Rohan Sharma has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Álvaro marked this as fixed in 4.0.2 with commit bf6249 8 months ago

Álvaro has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

Álvaro 8 months ago Maintainer

Thanks for reporting it! ;)
The issue has been fixed in version 4 (4.0.2 is the latest)

Jamie Slome 8 months ago Admin

Great work all! 👍

Sign in to join this conversation

terms

privacy policy

Chat with us