Exposure of Sensitive Information to an Unauthorized Actor in ionicabizau/parse-url

0

Valid Reported on Feb 11th 2022

Description

First Assume this example

```
var parseUrl = require("parse-url")
parseUrl("http://firstdomain.com@jflsdk.com:20200@google.com/path/name?foo=
```

that return :

```
protocols: ["http"]
protocol: "http"
port: null
resource: "firstdomain.com@jflsdk.com"
user: ""
pathname: "/20200@google.com/path/name"
hash: "some-hash"
search: "foo=bar&bar=42"
href: "http://firstdomain.com@jflsdk.com:20200@google.com/path/name?foo=bar
```

With the same URL in the URL interface of nodejs we have following result:

```
hash: "#some-hash"
host: "google.com"
hostname: "google.com"
href: "http://firstdomain.com%40jflsdk.com:20200@google.com/
origin: "http://google.com"
```

```
  password: "20200"
  pathname: "/path/name"
  port: ""

  protocol: "http:"
  search: "?foo=bar&bar=42"
  searchParams: URLSearchParams {}
  username: "firstdomain.com%40jflsdk.com"
```

In `parse-url` , You can see that the resource and href have different origins but they have the same origin in the `URL` interface of nodejs/browser.

attack senario

The scenario is simple: developers get a URL from the user in Backend, parse it, and verify its host(resource). The host value of the parsed payload is google.com, and they confirm the host as a reliable host, then use href to get the contents and show them to users, but the aaaaaa.com contents will be delivered to users.
Maybe You say developers often don't use the href and they use the URL instead (after confirming the host value), I say :
They store the created URL object and want to keep it somewhere and use it again and again.
More interesting example :
After storing the URL object, developers want to get content from google.com with the cookies that belong to them, but they think wrong!! And they send the cookies and other authorization headers to aaaaaa.com that cause potential information disclosure damage.

CVE
CVE-2022-0722
(Published)

Vulnerability Type
CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

Severity
Medium (4.8)

Visibility
Public

Status
Fixed

Found by

amammad

@amammad

pro

Fixed by

Ionică Bizău (Johnny B.)

@ionicabizau

unranked

This report was seen 797 times.

We are processing your report and will contact the **ionicabizau/parse-url** team within 24 hours.
10 months ago

We have contacted a member of the **ionicabizau/parse-url** team and are waiting to hear back
9 months ago

We have sent a follow up to the **ionicabizau/parse-url** team. We will try again in 7 days.
9 months ago

Ionică 9 months ago — Maintainer

Thank you for the report! :star:

I am unsure what the difference really is: I can see that the `@` is encoded, but apart from that the `href` seems to be the same, isn't it?

amammad 9 months ago — Researcher

I'm sorry for my bad explanation.
I will answer you in the next three hours, as I don't have access to my computer at this time.

amammad modified the report 9 months ago

amammad 9 months ago

When we want to get content of `http://firstdomain.com@jflsdk.com:20200@google.com/path/name?foo=bar&bar=42#some-hash` with the help of an HTTP library, we will lead to

`http://google.com/path/name?foo=bar&bar=42#some-hash` and many libraries use `firstdomain.com@jflsdk.com:20200` as Auth data of a URL.

They first exclude all information before the last `@` sign and then use all data after the last `@` as the hostname and directories and queries and ...

amammad 9 months ago Researcher

Also in browsers when we want to visit browsers `http://firstdomain.com@jflsdk.com:20200@google.com/path/name?foo=bar&bar=42#some-hash` , we will be led to `http://google.com/path/name?foo=bar&bar=42#some-hash`

amammad 9 months ago Researcher

With URL interface, we have `http://firstdomain.com%40jflsdk.com:20200@google.com/path/name?foo=bar&bar=42#some-hash` that encode the first `@` because it uses `firstdomain.com%40jflsdk.com` as the username of the Auth part.
However, there isn't any problem with that because it has a correct host/hostname (`google.com`) value.

amammad 9 months ago Researcher

Hi Ionică Bizău

Please tell me to provide more profound examples and explanations if I couldn't show the impact properly.

We have sent a second follow up to the **ionicabizau/parse-url** team. We will try again in 10 days. 9 months ago

Ionică Bizău (Johnny B.) validated this vulnerability 9 months ago

amammad has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

We have sent a fix follow up to the **ionicabizau/parse-url** team. We will try again in 7 days. 9 months ago

We have sent a second fix follow up to the **ionicabizau/parse-url** team. We will try again in 10 days. 9 months ago

We have sent a third and final fix follow up to the **ionicabizau/parse-url** team. This report is now considered stale. 8 months ago

Ionică Bizău (Johnny B.) marked this as fixed in **7.0.0** with commit **21c72a** 5 months ago

Ionică Bizău (Johnny B.) has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Sign in to join this conversation